

信息安全技术丛书

互联网企业安全高级指南

赵彦 江虎 胡乾威 编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

互联网企业安全高级指南 / 赵彦, 江虎, 胡乾威编著. — 北京: 机械工业出版社, 2016.8

(信息安全技术丛书)

ISBN 978-7-111-54301-5

I. 互… II. ① 赵… ② 江… ③ 胡… III. 网络公司-企业安全-指南 IV. F276.6-62

中国版本图书馆 CIP 数据核字 (2016) 第 166015 号

本书由业内多位顶级安全专家倾力打造, 分享了他们十多年的安全行业经验, 特别是对大型企业 (包括国内 TOP10 互联网公司在内) 的安全架构实战经验, 对如何打造企业级的网络安全架构与信息安全管理体系进行了系统化的总结。从技术到管理, 从生产网络到办公网络, 从攻防对抗到业务风控, 涉及安全领域的各个维度, 包括三十多个重要话题, 为企业实施符合互联网特性的安全解决方案提供了实用指南。本书分为三大部分: 理论篇、技术篇、实践篇, “理论篇” 包括安全大环境与背景、安全的组织、甲方安全建设方法论、业界的模糊地带等, “技术篇” 包括防御架构原则、基础安全措施、网络安全措施、入侵感知体系、漏洞扫描、移动应用安全、代码审计、办公网络安全、安全管理体系、隐私保护等, “实践篇” 包括业务安全与风控、大规模纵深防御体系设计与实现、分阶段的安全体系建设等。



互联网企业安全高级指南

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 吴 怡

责任校对: 董纪丽

印 刷:

版 次: 2016 年 8 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 19.25

书 号: ISBN 978-7-111-54301-5

定 价: 69.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

本书赞誉

(人名按照姓氏拼音排序)

本书是安全大 V 们多年安全从业经验的精华凝聚，是互联网企业安全从业人员的必读经典。

——陈建，平安集团信息安全资深总监

长期以来，系统而全面讲述互联网企业安全文章都极少，图书就更少了。目前大多数文字内容是描述各种攻防细节、攻击技巧，或者是理论味道十足、高大上无比的安全体系架构理论；不是缺乏企业安全整体架构以点代面，就是缺乏企业实际场景难以落地。对于互联网企业安全管理者而言，最有价值的并不是具体的攻防技巧，也不是那些安全理论标准，而是一套可落地、低成本、行之有效的最佳实践。本书作者凭借在甲方和乙方十多年摸爬滚打的实际经验，梳理出了这样一套满满都是经验的互联网企业安全建设最佳实践，帮助企业进行安全建设时少走弯路，少花冤枉钱；帮助想往企业安全方向发展的专业人士拓宽思路，完整视野，丰富经验；帮助 CSO 们归纳总结，审视发展。

——陈洋 (cy07)，小米 CSO

互联网甲方所面临的安全挑战，是与传统企业完全不同的。甲方和乙方，攻击和防守，角色转换并不容易。本书内容翔实，对有志参与互联网企业安全建设的人士来说，是一本必备宝典。

——CoolQ，阿里巴巴安全部资深安全专家

大部分 IT 管理者思想上很重视安全，但是缺少安全管理理念和执行，因为大家都没有一个正确的安全观，或者说安全体系概念，不知道怎么去落地执行。本书作者赵彦在甲方和乙方都待过，他的眼界是可以覆盖到更多的安全层面，这本书很好地从多个视角阐述了安全体系的架构，并且已经给出了执行步骤，希望对你的企业有用。

——窦喆，运维帮创始人

甲方安全工作知不易，行更难，本书结合多位业界顶尖安全专家的经验从管理和技术维度全面深入剖析了甲方安全的场景；是企业安全相关从业人员的通关秘籍。

——方勇，UCloud 安全中心总监

很高兴看到赵彦等同学的大作问世，本书没有高深的理论知识，没有高大上的概念，全部内容来自十几年的实战经验和思考总结，相信可以给在互联网企业从事安全相关工作的同学提供全面、系统、接地气的指导和帮助。

——郭添森，去哪儿网安全总监

大型互联网企业的在线业务在触达了全网用户的同时也给所有攻击者暴露了攻击面，且攻击者又拥有近乎海量的资产，量变引起质变，企业的安全防护问题早已不仅仅是技术问题，更多的是工程和管理问题。如何做好这里的工作，是令人头疼的事情。更为遗憾的是，目前市面上的信息安全类书籍要么是讲具体某类安全技术，要么就是放之四海而皆准却又无法落地的理论指导，很少有理论结合企业实际工作情况的探讨，这本书的出版弥补了这个遗憾。本书作者赵彦是安全圈老前辈，在甲方乙方都工作过，有丰富的经验；江虎过去有五年时间是我的同事，为腾讯的安全体系建设做出了重要贡献。本书从理论、技术、实践三个部分入手，基本涵盖了互联网企业安全的方方面面，是企业安全人员的案头必备参考书籍。强力推荐。

——胡珀 (lake2)，腾讯安全平台部总监

一直在互联网公司从事互联网安全攻防的工作，在这个过程中也一直尝试寻找一些相对稳定的答案。本书的价值不仅在于手把手告诉你网络安全怎么做，业务安全怎么做，系统攻防怎么做，虽然书中在这方面也有足够精到的论述，它的价值更在于让你身处一个纷繁复杂、日新月异的互联网大环境中，从外部到内部、从宏观到微观多重视角去思考你所服务企业的信息安全的方方面面，让你理解你的企业当前处于什么样的安全位置，并运用恰当的方法论、安全管理手段和安全技术，找到属于你服务企业的答案。这本书是我见过的国内第一本系统化剖析整个互联网企业安全的书籍，如果早几年出现这样的书籍，一定可以让我少走几年弯路，期待这本书能够让更多的安全同行早日找到属于自己的答案。

——黄眉，阿里巴巴安全部安全总监

本书是少见的横贯信息安全“南”“北”之作，并且十分难得地分享了互联网公司安全实践的经验，不论是从事“传统安全”还是“互联网安全”，均值得一读。同时，本书包括的内容也相当丰富、全面，涵盖了从组织建立、建设方法论、原则直至具体实践，不论是

希望加入信息安全行业的爱好者还是行业老兵，都可以从中得到启发。

——金湘宇 (NUKE)，前信息技术、信息安全资深顾问，现私募投资人

当今世界的技术发展日新月异，安全工作的复杂性急剧增高，网络与信息安全事故层出不穷，若不能有效应对会给各行各业和广大用户带来极大风险。非常幸运我所工作的公司都对安全非常重视，拥有业界的一批安全精英，华为公司的安全专家赵彦就是其中的出众代表，他不仅具有坚实的攻防基础和安全战略思维，还能够洞察传统安全与互联网安全的差异，对部门、公司、行业的安全发展提出有高度价值的建议。本书内容来源于实践，经过了深入的分析和提炼，升华成为有益的指导和参考，相信不论是 CSO 还是刚入行的从业者，不论是安全专业人员还是想了解安全的业务人员，都会从本书中获益良多。

——李雨航，Huawei Global Chief Scientist/Technologist (Cyber Security)，网络安全实验室 CTO 兼国际 CSO，CSA 大中华区主席，XJTU Fellow/Professor

终于看到一本真正意义上企业安全方面的书籍，作者兼顾广博与精深的专业功底，横跨甲乙方、传统安全与互联网安全的多个行业领域的视角与知识层次，相信每一位读者都会有所收获。从攻防以外的视角，体系化地介绍安全，对企业安全的负责人、从业者以及乙方机构咨询与技术人员都有很好的实用指南作用。

——吴树鹏，滴滴出行首席安全顾问

很高兴在这个时间看到这样一本书的面世，在试读样章时，产生了很多的共鸣。企业做安全很难，尤其是在以开放和不断变化著称的互联网领域尤甚。业务形态的差异决定了各个企业的安全目标不尽相同。“方向错了，前进便是倒退”。企业安全的不同阶段如何合理地设定目标，选择切实可行的方案，以及如何在有限的时间、人力和资源下前行，是安全管理者们必须面对的问题。安全建设必然伴随着不断的取舍、权衡，乃至博弈。希望阅读此书后能让每个安全人员在大到安全规划、策略制定，小到每个安全漏洞修补、策略上线，乃至安全事件的应急追查中，都能认清自己所做事情在企业安全蓝图中的位置，而不是为了做事而做事，更好地“搞定”事情。

——xi4oyu，百度安全实验室 Xteam 负责人

相比技术篇，本书的理论篇更加精彩。互联网安全从业者最大的挑战并非技术，而是如何建立正确的行业格局观，并且能够与业务团队、安全团队管理层、公司管理层建立良好的沟通机制，取得信任和支持。本书的几位作者在互联网行业有着丰富的安全管理和从业经验，强烈推荐有志于从事互联网安全行业的朋友阅读。

——薛锋，微步在线 CEO，原亚马逊中国 CISO

当今的互联网对企业安全老大的需求远远大过供给。不客气的说，现在很多互联网的安全负责人都是被赶鸭子上架，在工作中摸索学习做 CSO 的。有些聪明的、注重学习的人，到岗后能逐步建成一只有一定实力并且能逐步成长的团队。另外一些就利用 CTO 不懂安全，靠一张嘴皮子欺上瞒下，拉了一票自己人搞办公室政治。而真正有实战经验、有创新力、能系统性地写这样一本书的极少数人，大都正忙于建设自己企业的安全团队或者在创业路上，没有时间或者没有动力写。本人曾经有幸在美国做过互联网企业的甲方和乙方，并且在三个中国互联网企业分别从头建设了三只安全团队。虽然我也动过这个心思写本书总结一下经验教训以及中间的有趣的人和事，但也是打算退休以后才做。另外我从市场的角度考虑，真正关心这个问题的人少的可怜，写完了我估计想看的人也不会多，可能最终就是个自娱自乐罢了。赵彦是这个极少数的人群中的一个极少数派，居然现在就有时间有兴趣写出这样一本书出来。当然，每个企业不同，每个 CSO 的思想方法不同，这本书里的有些方法或者说法 CSO 们并不见得会完全赞同，但是这本书给 CTO 或者准备当 CSO 的人提供了一个比较完整的视角，把 CSO 具体都应该解决什么问题讲清楚了，而不是以前笼统的一句“反正安全出了事就都是他负责”。除了介绍问题以外，本书也介绍了一些系统方法帮助新上任的 CSO 整理一下总体思路。总之，这是一本当下业界急缺的书。

——杨更，前小米、美团、亚马逊中国 CSO

此书凝结了赵彦对互联网企业安全体系建设的思考和经验提炼，覆盖了管理和解决方案，在宏观面阐述了自己的理解和安全观，对安全领域从业者是一份很好的参考资料。

——杨勇 (Coolc)，腾讯云副总裁，腾讯安全平台部负责人

干货！是落地实践还是纸上谈兵，是美女还是野兽，作者抽丝剥茧、层层展开，分享了他们在互联网企业安全领域中丰富的实战心得，甲方和乙方都非常值得阅读参考。

——赵粮博士，绿盟科技首席技术官

对于绝大多数的企业来说，安全是信息技术建设中薄弱而又很难提高的环节，而这个环节上发生的问题又会深刻地影响到整个企业，本书深入浅出地介绍企业信息安全体系从建设到实施的一套完整过程，强烈建立企业信息安全技术团队的负责人都能阅读此书，一定能让你受益匪浅。

——郑歆炜 (cnhawk)，支付宝高级安全专家

前言

在互联网+的进程中，一方面互联网企业越来越多，另一方面由外部环境推动的或自发的安全意识越来越强，对安全建设的需求也越来越多，很多企业都开始招聘安全负责人，不乏年薪上百万元和几百万元的安全负责人职位，但事实是很多公司常年用高薪，都招不到合适的安全负责人，其中的原因有很多，比较客观的一条就是这个行业所培养的有互联网整体安全视角的人实在寥寥无几，而这些人都不缺高薪，也不缺职位。我在“信息安全行业从业指南 2.0”一文中曾经写过自 2014 年开始，安全行业的大部分高端人才都在互联网行业，为什么还是有那么多缺口？细想了一下有几个方面的因素：

- ❑ 过去，安全并不太受重视，安全从业者的职业发展瓶颈明显，很多拥有整体安全视角的人离开了安全行业转去做其他的事情了。
- ❑ 早年在乙方安全公司的人经历了给客户从零开始建设安全体系的过程，而后来进入乙方的毕业生，接触客户时大都已经有安全体系，无法体验从 0 到 1 的过程并从中学到完整的方法论，很多方法论甚至已“失传”，有些方法论原本就只集中在公司总部的一圈人手里，分支机构除了文档得不到“大象”。
- ❑ BAT 这类企业安全也早已经过安全建设期，后来者分工很细，除了几个早已身居总监职位的老员工之外，大多数人无法得知安全体系的全貌，很多人离开了 BAT 也说不清自己责任之外的事情该怎么做。
- ❑ 二三线互联网企业中，很多团队所持有的安全体系并不完整，也不是业内最佳实践，有些甚至就是救火型团队，更难有体系化的积累。
- ❑ 当下的很多乙方安全公司，在互联网大潮的冲击下也面临着转型的挑战，要提供符合时代趋势的解决方案仍需努力。
- ❑ 在社区，目前大家都热衷攻防，而不是企业安全建设。攻击者、乙方、甲方之间仍然存在较大的鸿沟，彼此互相不屑，从社区里也多半只能挖掘到一些单点型的防御

手法，即便好学的人订阅了所有的安全站点和微信公众号，恐怕也难以学会企业整体安全建设的方法。

基于以上种种原因，我明显感觉到有一堵墙存在于业界、社区、甲方、乙方、想学习的人和信息的不对称之间，我决定动手推倒这堵墙，所以就有了这本书。

首先本书聚焦于互联网行业的企业级安全解决方案、架构、方法论和建设思路，关于单点技术，市面上已经有很多书，所以本书内容大多不会围绕单点技术来讲，而是希望读者看完之后找到企业安全整体建设的那种感觉。即便是一个甲方安全工程师，也能从中学到互联网公司安全负责人的知识和视野，并以此为导航，逐步积累自己所需要的知识和技能，向更高的层级发展。

对于乙方安全公司的从业者而言，顾问们或许可以从中了解甲方的需求和工作，从而提供更加接地气的交付方案。对于产品设计和研发人员，本书展示的互联网安全架构有助于拓展传统安全的思路，不一定能直接复用，但也许能有所启发。

对于黑客技术爱好者，比较安全的道路仍然是从事安全，不可避免地也要学习这些，高级的渗透和攻击技巧都需要绕过防御手段，了解防御者思维是必须跨过的门槛。

本书同样适用于想了解企业安全建设的 CTO、运维总监、研发总监、架构师。但本书内容都假设读者有一定的基础，对一些比较基础的名词和技术没有做太多的解释。

在出版这本书时由于时间和信息披露方面的限制，写出来的部分与我们原先期望的仍有不少差距，但另一方面我们希望像互联网产品的迭代方式一样，不追求完美，但求尽快面世，因此本书也难免带着各种 bug 上线，也欢迎各位读者的意见或建议。此外，我们计划在本书的第 2 版中进一步展开某些话题，并且更加系统化，同时会在业界最佳实践方面挑战另一个维度。

最后特别感谢本书的编辑吴怡自我在 360 工作时便找到我，建议我将网络中的文字写成书，让更多的人能读到。感谢另外两位作者江虎（ID：xti9er）和胡乾威（ID：Rayxcp）帮我分担了很多压力，使得此书能尽快面世。同时感谢 netxfly@ 小米、职业欠钱 @ 腾讯、clyde@ 电信云堤、终极修炼师 @ 唯品会、laintoday@ 爱奇艺提供的帮助。

目 录

前言

理论篇

第 1 章 安全大环境与背景2

1.1 切入“企业安全”的视角.....2

1.2 企业安全包括哪些事情.....5

1.3 互联网企业和传统企业在安全建设
中的区别.....9

1.4 不同规模企业的安全管理.....12

1.5 生态级企业 vs 平台级企业安全建设
的需求.....13

1.6 云环境下的安全变迁.....16

第 2 章 安全的组织17

2.1 创业型企业一定需要 CSO 吗.....17

2.2 如何建立一支安全团队.....19

第 3 章 甲方安全建设方法论22

3.1 从零开始.....22

3.2 不同阶段的安全建设重点.....24

3.3 如何推动安全策略.....26

3.4 安全需要向业务妥协吗.....28

3.5 选择在不同的维度做防御.....29

3.6 需要自己发明安全机制吗.....33

3.7 如何看待 SDL34

3.7.1 攻防驱动修改.....36

3.7.2 SDL 落地率低的原因37

3.7.3 因地制宜的 SDL 实践38

3.7.4 SDL 在互联网企业的发展39

3.8 STRIDE 威胁建模.....40

3.9 关于 ISO2700142

3.10 流程与“反流程”43

3.11 业务持续性管理.....45

3.12 关于应急响应.....47

3.13 安全建设的“马斯洛需求”层次.....48

3.14 TCO 和 ROI.....50

第 4 章 业界的模糊地带52

4.1 关于大数据安全.....52

4.2 解决方案的争议.....55

技术篇

第 5 章 防御架构原则60

5.1 防守体系建设三部曲.....60

5.2 大规模生产网络的纵深防御架构.....62

5.2.1 互联网安全理念.....62

5.2.2 攻击者视角.....63

5.2.3 防御者模型.....63


5.2.4 互联网安全架构设计原则.....	66	第 8 章 入侵感知体系	123
第 6 章 基础安全措施	70	8.1 主机入侵检测.....	123
6.1 安全域划分.....	70	8.1.1 开源产品 OSSEC.....	123
6.1.1 传统的安全域划分.....	70	8.1.2 MIG.....	129
6.1.2 典型的 Web 服务.....	71	8.1.3 OSQuery.....	131
6.1.3 大型系统安全域划分.....	72	8.1.4 自研 Linux HIDS 系统.....	135
6.1.4 生产网络和办公网络.....	74	8.2 检测 webshell.....	144
6.2 系统安全加固.....	75	8.3 RASP.....	149
6.2.1 Linux 加固.....	75	8.3.1 PHP RASP.....	149
6.2.2 应用配置加固.....	81	8.3.2 Java RASP.....	153
6.2.3 远程访问.....	83	8.4 数据库审计.....	159
6.2.4 账号密码.....	83	8.5 入侵检测数据分析平台.....	162
6.2.5 网络访问控制.....	84	8.5.1 架构选择.....	162
6.2.6 补丁管理.....	86	8.5.2 功能模块.....	163
6.2.7 日志审计.....	86	8.5.3 分析能力.....	164
6.3 服务器 4A.....	87	8.5.4 实战演示.....	167
第 7 章 网络安全	89	8.6 入侵检测数据模型.....	169
7.1 网络入侵检测.....	89	8.7 数据链生态——僵尸网络.....	174
7.2 T 级 DDoS 防御.....	95	8.7.1 僵尸网络传播.....	174
7.2.1 DDoS 分类.....	95	8.7.2 僵尸网络架构.....	175
7.2.2 多层防御结构.....	100	8.7.3 应对僵尸网络威胁.....	179
7.2.3 不同类型的企业.....	108	8.8 安全运营.....	181
7.2.4 不同类型的业务.....	109	第 9 章 漏洞扫描	182
7.2.5 服务策略.....	109	9.1 概述.....	182
7.2.6 NIPS 场景.....	110	9.2 漏洞扫描的种类.....	183
7.2.7 破防和反制.....	111	9.2.1 按漏洞类型分类.....	183
7.2.8 立案和追踪.....	112	9.2.2 按扫描器行为分类.....	190
7.3 链路劫持.....	113	9.3 如何应对大规模的资产扫描.....	197
7.4 应用防火墙 WAF.....	117	9.4 小结.....	198
7.4.1 WAF 架构分类.....	117	第 10 章 移动应用安全	200
7.4.2 WAF 安全策略建设.....	118	10.1 背景.....	200
7.4.3 WAF 性能优化.....	121	10.2 业务架构分析.....	200

10.3	移动操作系统安全简介	201	13.7	开放与合作	246
10.4	签名管理	202	第 14 章 隐私保护 248		
10.5	应用沙盒及权限	203	14.1	数据分类	250
10.6	应用安全风险分析	204	14.2	访问控制	250
10.7	安全应对	205	14.3	数据隔离	251
10.8	安全评估	206	14.4	数据加密	253
10.9	关于移动认证	206	14.5	密钥管理	258
第 11 章 代码审计 207			14.6	安全删除	258
11.1	自动化审计产品	207	14.7	匿名化	259
11.2	Coverity	208	14.8	内容分级	259
第 12 章 办公网络安全 216			实践篇		
12.1	文化问题	216	第 15 章 业务安全与风控 264		
12.2	安全域划分	217	15.1	对抗原则	264
12.3	终端管理	218	15.2	账号安全	265
12.4	安全网关	221	15.3	电商类	270
12.5	研发管理	222	15.4	广告类	274
12.6	远程访问	224	15.5	媒体类	274
12.7	虚拟化桌面	224	15.6	网游类	274
12.8	APT	226	15.7	云计算	275
12.9	DLP 数据防泄密	227	第 16 章 大规模纵深防御体系设计与实现 276		
12.10	移动办公和边界模糊化	228	16.1	设计方案的考虑	276
12.11	技术之外	229	16.2	不同场景下的裁剪	281
第 13 章 安全管理体系 230			第 17 章 分阶段的安全体系建设 283		
13.1	相对“全集”	234	17.1	宏观过程	283
13.2	组织	235	17.2	清理灰色地带	285
13.3	KPI	236	17.3	建立应急响应能力	286
13.4	外部评价指标	239	17.4	运营环节	288
13.5	最小集合	240	附录 信息安全行业从业指南 2.0 290		
13.5.1	资产管理	240			
13.5.2	发布和变更流程	241			
13.5.3	事件处理流程	241			
13.6	安全产品研发	245			



理论篇



- 第1章 安全大环境与背景
 - 第2章 安全的组织
 - 第3章 甲方安全建设方法论
 - 第4章 业界的模糊地带
- 

安全大环境与背景

如果从一个很微观的角度切入企业安全这个话题，那么大多数人会像一叶孤舟跌进大海茫茫然找不到方向，所以本章从安全领域整体环境入手，以便于读者找到系统性的那种感觉。尽管笔者没有致力于提供关于企业安全的一个非常完整的“上帝视角”，但也尽可能地兼顾了这方面的需求。

1.1 切入“企业安全”的视角

目前安全行业中“二进制”和“脚本”流派广为人知，虽然他们是安全行业的主力军，但除了微观对抗之外，安全是一个很大的工程，比如企业安全管理，实际上并不属于上述两类。确切来说，应归入另外一个群体：CSOs，加了s表示他们是一个群体，这个群体从生态链的顶端联接着绝大多数从业者和安全厂商。在这个描述中我并没有发明新的名词，没有新建一个诸如气宗剑、宗这样的词，只是在知识结构和职业背景上做一定的区分，用于后续对这本书的扩展。

企业安全是不是发现漏洞然后修复漏洞，再设置一下防火墙之类的工作？假如你的公司只有一个产品、两台服务器、3个程序员，我认为这个说法不能算错。不过在绝大多数情况下，企业安全远不止于此。渗透性测试和对抗能不能算企业安全？在一个过于纸上谈兵的企业我觉得这是不错的切入点，不过局部对抗发生于企业安全的各个场景中，它只能算

是缩影，不是全貌。企业安全是什么？对传统乙方安全公司，对新兴的业务安全公司、移动安全公司，对甲方的互联网公司，对甲方的传统公司，对咨询顾问，对漏洞研究者，对活跃于各大 SRC 上的白帽子们来说，诠释肯定都不一样。

先说一下笔者的经历，以便了解是从什么角度来阐述这一问题的。学生时代跟现在的很多白帽子一样玩玩渗透，玩玩二进制，在过去叫幻影（Ph4nt0m）的组织里认识了很多大 V，大学毕业后即进了绿盟做渗透测试、安全服务和咨询，这是乙方中离甲方安全最近的职位，接受了绿盟对传统安全体系和方法论的教育，有些 10 年前的东西放到今天看都还会觉得完全不过时。

现在的安全行业里除了显得有些务虚的安全理论之外，要么就是一边倒的攻防，要么就是过于超前、浮在表面没有落地方案的新概念，这些声音对企业安全实操都缺乏积极的意义。有些方法论是有实操意义的，并不像攻防研究者声称的是纯务虚的东西，纯粹是位置决定想法的问题。还有些流行的概念在解决实际问题上的效果有待验证，并不像市场鼓吹的那么好。技术很重要，但攻防只解决了一半问题，安全的工程化以及体系化的安全架构设计能力是业内普遍的软肋，多数人不擅此道。对市场上的各种观点，我认为可能需要一个相对客观的评价：即某项技术或管理在全生命周期的各个环节中，在不同的行业、不同的场景下有什么样的价值，而不是很随意地贴标签。很多概念 10 多年前就有了，发明一个新概念讲与过去一样的事情，再给自己贴一个发明者的标签对行业没有积极的意义。纵深防御之类的概念在 ISS 没被 IBM 收购之前就有了，为什么现在有的人觉得这个词很新？因为过去没重视，或者说缺少实践。

在绿盟最大的便利并不是下班路上随便都能找到能聊 exploit 技术的大牛，而是视野：从金字塔视角看到安全的整体解决方案，囊括组织、管理和技术 3 方面的东西，覆盖全行业全价值链过程的技术方案，算上第三方的话几乎涵盖市面上所有的安全产品和解决方案。有人看到这些会问：这不是传统安全那一套吗？且不急，本书后面讲的都是围绕互联网企业安全的，并不打算在传统安全上花很多篇幅，只是需要区分一下企业安全实际的状况和某些厂商为了兜售自己的产品而宣扬的概念是有所不同的，大多数厂商都会避开自己的弱项而在市场活动及软文上专注地强调自己擅长的概念。

离开绿盟后我去了甲方，一家大型网游公司，2008 年将近万台的物理服务器分布于三十多个 IDC 的规模似乎比当时搜狐全站的 IDC 规模还要大一些。跟现在一样，那时候也是普遍缺少安全负责人的时代，我也有幸组建了一支属于自己的安全团队，成为当时极少

数的安全总监之一。团队中的人现在遍布互联网行业的半壁江山，且都是安全部门独当一面的骨干。在这段时间我亲身经历了从乙方到甲方视角的过渡，从零开始建立安全体系，真正把乙方玩的东西在一家甲方公司落地了。我的方法思路过程跟某些互联网公司不太一样，因为那时候 BAT 的安全人员大多是毕业后直接去的甲方，一开始就做甲方安全，而我则是从乙方到甲方，所以实践上更多是参考了乙方的方法论，再自己摸石头过河，除了攻防之外，多线并行，直接建立较完整的安全体系。

后来在安全行业不太景气的那个年代我好像碰到了安全行业的天花板。之后跟在“信息安全的职业生涯”一文中所述的那样，我实践了里面所说的最后一跳，做了一家网游公司的技术负责人，社会俗名 CTO，由安全转向全线技术管理。说实话，在这段时间里我并不是特别重视安全，一方面跟自己是安全出身有关，另一方面这确实是位置决定想法的事情，不是安全不重要，而是有很多事情比安全更重要。老实说，安全这个事情跟金钱关系密切，当你有 100 万元的时候拿出 2 万元买个保险箱装它们你觉得值，但你只有 2 万元的时候要拿出 8 千元买保险箱，大多数人都会不愿意。可参见我在知乎上回答的那个问题：“为什么做安全一定要去大公司”。我窃以为很多公司的 CEO、CTO 对安全的认识，翻译过来应该是：被黑是一件很负面的事情，所以找个人筹建团队打包了，只要不出事就行。他们不是真的认为安全非常重要，也不会把安全当成一种竞争力。现在说这句话并不是在影射过去，当下国内很多企业的观念仍然停留在这个水平上。

之后我去 360 经历了短暂的时光，再次以乙方的身份拜访了企业级客户，很偶然地发现大多数乙方安全公司的顾问或工程师其实都没有企业安全管理的真正经验。虽不能把这些直接等价于纸上谈兵，不过确实是乙方的软肋。在甲方企业高层的眼中，攻防这档子事可以等价于我花点钱让安全公司派几个工程师给我做渗透测试然后修复漏洞，不像大型互联网公司那样上升为系统化和工程化的日常性活动。离开数字公司后，我到了全球化的公司（华为）从事产品线安全，负责两朵云：公有云和终端云。产品线安全属于甲方安全，又跟很多甲方安全不太一样，比传统意义上的甲方安全介入得更深，覆盖率更高的 SDLC，直接导向产品设计的源头。对绝大多数甲方而言，你也许在用 OS 的 Dep&ASLR，也许在用各种容器，但你很少会自己去发明轮子，你也许会自己造一个 WAF 这样的工具，但你可能很少会像微软那样要自己去搞一个 EMET 这种涉及安全机制层面的东西。但在产品线安全里，这一切都会更进一步，不只是像互联网企业那样关注入侵检测、漏洞扫描等，而是从设计和威胁建模的角度去看整体和细节的安全。这又拓展了我从 R&D 的视角看待以及分析安全问题的眼界。因此，我可以站在一个较全面、客观、中立的立场来说安全，我不会说

某些方式属于纸上谈兵，也不会把攻防捧得至高无上。

接下来，切入本章正题，企业安全是什么？我认为它可以概括为：从广义的信息安全或狭义的网络安全出发，根据企业自身所处的产业地位、IT 总投入能力、商业模式和业务需求为目标，而建立的安全解决方案以及为保证方案实践的有效性而进行的一系列系统化、工程化的日常安全活动的集合。怎么感觉有点咬文嚼字？实际上，这里的每一个项都会决定你的安全整体方案是什么，哪怕同是中国互联网 TOP10 中的公司，安全需求也完全不一样。

有人也许会觉得 CSO 干的活有点虚，但凡偏管理都是纸上谈兵。我不直接回答这个问题，我只举一个例子。大多数身在这个行业的人都知道社工库遍地都是，入侵者甚至站在了大数据的维度，国内的数据库绝大多数除了 password 字段加盐值存储之外，其余信息都以明文保存。而在欧美等地隐私保护是有明确的法律规定的，映射到数据持久化这个细节，就是需要满足一定强度以上的加密算法加密存储。CSO 就是需要制定这些策略的人，难道说这些都是形而上学无用的安全措施吗？在互联网公司，安全负责人会较多地介入到日常技术性活动中，但随着组织规模的扩大和行政体系的加深，CSO 不再可能像白帽子一样专注于攻防对抗的细节，这也是一个无法回避的现实问题。是不是一定要说出诸如 CSRF 时 IE 和其他浏览器的区别，才算是合格的 CSO？我觉得这要看具体场景，对于国内排名 TOP10 以后的互联网企业，我觉得这个要求也许勉强算合理范畴，但对于规模非常庞大的企业而言，这个要求显然太苛刻了，比如我所在公司，CSO 属于法务类职位而不是技术类职位。

不想当将军的士兵不是好士兵，虽然有人想走纯技术路线，但是仍有很多人想过要当 CSO。CSO 这个职位跟某些大牛表达的不完全一致，所以下面的篇幅会继续写，至少在技术层面，CSO 不会只停留在微观对抗上，而是会关注系统性建设更多一点。至于跟董事会建立沟通桥梁，虽然也重要，不过关注的人就更少了，本书将不会涉及。

1.2 企业安全包括哪些事情

企业安全涵盖 7 大领域，如下所示：

1) **网络安全**：基础、狭义但核心的部分，以计算机（PC、服务器、小型机、BYOD……）和网络为主体的网络安全，主要聚焦在纯技术层面

2) **平台和业务安全**：跟所在行业和主营业务相关的安全管理，例如反欺诈，不是纯技术层面的内容，是对基础安全的拓展，目的性比较强，属于特定领域的安全，不算广

义安全。

3) 广义的信息安全：以 IT 为核心，包括广义上的“Information”载体：除了计算机数据库以外，还有包括纸质文档、机要，市场战略规划等经营管理信息、客户隐私、内部邮件、会议内容、运营数据、第三方的权益信息等，但凡你想得到的都在其中，加上泛“Technology”的大安全体系。

4) IT 风险管理、IT 审计 & 内控：对于中大规模的海外上市公司而言，有诸如 SOX-404 这样的合规性需求，财务之外就是 IT，其中所要求的在流程和技术方面的约束性条款跟信息安全管理重叠，属于外围和相关领域，而信息安全管理本身从属于 IT 风险管理，是 CIO 视角下的一个子领域。

5) 业务持续性管理：BCM (Business Continuity Management) 不属于以上任何范畴，但又跟每一块都有交集，如果你觉得 3) 和 4) 有点虚，那么 BCM 绝对是面向实操的领域。最近，有网易、中有支付宝、后有携程，因为各种各样的原因业务中断，损失巨大都属于 BCM 的范畴。有人会问：这跟安全有什么关系？安全是影响业务中断的很大一部分可能因素，例如 DDoS，入侵导致必须关闭服务自检，数据丢失，用户隐私泄露等。又会有人问：这些归入安全管理即可，为什么要跟 BCM 扯上关系，做安全的人可以不管这些吗？答案自然是可以不管，就好像说：“我是个 Java 程序员，JVM、dalvik (ART) 运行原理不知道又有什么关系，完全不影响我写代码！”事实上，BCM 提供了另一种更高维度、更完整的视角来看待业务中断的问题。对于安全事件，它的方法论也比单纯的 ISMS 更具有可操作性，对业务团队更有亲和力，因为你知道任何以安全团队自我为中心的安全建设都难以落地，最终都不会做得很好。

6) 安全品牌营销、渠道维护：CSO 有时候要做一些务虚的事情，例如为品牌的安全形象出席一些市场宣介，presentation。笼统一点讲，现在 SRC 的活动基本也属于这一类。

7) CXO 们的其他需求：俗称打杂。这里你不要理解为让安全团队去攻击一下竞争对手的企业这样负面向的事情，而是有很多公司需要做，但运维开发都不干，干不了或者不适合干的事情，安全团队能力强大时可以承包下来的部分，事实上我的职业生涯里就做了不少这样的事情。

基础的网络安全是在甲方的绝大多数安全团队能覆盖的事情，不管你的安全团队能力如何，在公司里有无影响力，这个是必须要做的，因为这是把你招过来的初衷。再往后的发展，是否止于此则看个人的想法。对于沉醉攻防技术的人，其实不需要往后发展了，这些足够了。但如果你的安全团队富有活力和想法，即便你想止于此他们也不干，把部门做大做强是这些人的愿望，只有这样才能给安全团队更大的空间。这点跟乙方是不一样的，对于乙方而言，

你可以在某个单点领域上无限深挖，而不会遇到天花板，因为你始终是在满足主营业务的需求，即使你成为骨灰级的专家，公司也会对你在某方面创新有所期待而给你持续发展的可能性。但是在甲方，安全不是主营业务，归根结底，安全是一个保值型的后台职能，不是一个明显能创造收益的前台职能，是一个成本中心而非盈利中心。安全成本的大小跟业务规模以及公司盈利能力相关，公司发展时预算和人员编制都会增加，业务停滞时安全做得再好也不会追加投入，因为无此必要。反面的例子也有：做得不好反而追加投入的，那是一种政治技巧而非现实需要。在乙方，无论你的漏洞挖掘技能多厉害，公司都不会跳出来讲“你已经超出我们需求了，你还是去更强大的公司吧”（通常情况下）。但是在甲方，假设是在一个国内排名大约 TOP5 以后的互联网企业，养一个漏洞挖掘的大牛也会令人很奇怪，他是在给企业创造价值还是在自娱自乐是会受到质疑的，CSO 也会被质疑是否花了大价钱挖来的人不是出于业务需要而是用于扩大自己团队在业内影响力这种务虚的事。假如公司到了 Google 这种级别，有一大堆产品，储备大牛则是顺利成章的，业务上显然是有这种需求的。不过还要看产出是否对主营业务有帮助，工作成果不能转化为主营业务竞争力的尝试性活动在公司有钱的时候无所谓，在公司收紧腰带时则其存在价值就有争议。

以狭义的安全垂直拓展去发展甲方安全团队的思路本质上是个不可控的想法，筹码不在 CSO 手中，甚至不在 CTO 手中，而是看主营业务的晴雨表，甲方安全是要看“脸”的，这个脸还不是指跨部门沟通合作，而是在最原始的需求出发点上受限于他们。因此有想法的安全团队在网络安全方面做得比较成熟时会转向平台和业务安全，平台和业务安全是一个很大的领域，发展得好，安全团队的规模会扩大 2 倍，3 倍，并且在企业价值链中的地位会逐渐前移，成为运营性质的职能，结合 BCM 真正成为一个和运维、开发并驾齐驱的大职能。

BCM 在很多人眼里就是 DR（Disaster Recovery，灾难恢复），DR 其实只是 BCM 中的一个点，属于下层分支。不过这对技术领域的人而言是最直观的部分，DR 在互联网企业里由基础架构部门或运维主导。不过强势的甲方安全团队其实也是能参与其中的，而 BCP（Business Continuity Plan，业务持续性计划）中的很大一部分跟安全相关，我之前也主导过 BCP&DRP（Disaster Recovery Plan，灾难恢复计划），受益于绿盟那个年代的教育不只是攻防，而是完整的信息安全和风险管理。有兴趣的读者可以看一下 BS25999（BCM 的一个标准）。

广义的信息安全，比较直观的映射就是 ISO2700x 系列，行业里的绝大多数人都知道 ISO27001 和 BS7799，这里就不展开了，对真正有安全基础的人而言，都是很简单的东西。

在企业里能否做到广义的安全，主要看安全负责人和安全团队在公司里的影响力，对上没有影响力，没有诠释利害关系和游说的能力，自然也就做不到这些。另一方面，狭义安全主要对接运维开发等技术面公司同僚，但是广义安全会对接整个公司的各个部门，对于沟通面的挑战来说，又上了一个新的台阶，在我看来这主要取决于安全的领队人物自己拥有什么样的知识结构以及他的推动能力如何。

在企业完全涉及的 7 大领域中，对于第 4) 条，如果你所在的组织有这方面的需求，安全职能自然也会参与其中，是否刻意去发展他则看自己需求，对我朋友中某些做过 IT 治理和风险咨询的人，相信是有能力一并吃下的，如果是技术派，不建议去尝试。

第 6) 条属于水到渠成的事情，到了那一步你自然需要考虑，就算你不想，公司也会让你去，就像我现在明明做技术活，却也不知道为什么会跟这一类事情挂上钩。

第 7) 条有人看时自动过滤了，不过安全负责人自身是否有瓶颈，能否在企业里发展起来跟这条有很大关系，甚至有很多从 1) 发展到 2)、3) 的人都需要借助 7) 这个渠道，点到为止，不多说了。

对于互联网公司，我建议做 1)、2)、5)；对于传统行业，我建议做 1)、3)、4)、5)。

在互联网行业，我觉得安全工作可以概括为以下几个方面：

- ❑ **信息安全管理**（设计流程、整体策略等），这部分工作约占总量的 10%，比较整体，跨度大，但工作量不多。
- ❑ **基础架构与网络安全**：IDC、生产网络的各种链路和设备、服务器、大量的服务端程序和中间件，数据库等，偏运维侧，跟漏洞扫描、打补丁、ACL、安全配置、网络和主机入侵检测等这些事情相关性比较大，约占不到 30% 的工作量。
- ❑ **应用与交付安全**：对各 BG、事业部、业务线自研的产品进行应用层面的安全评估，代码审计，渗透测试，代码框架的安全功能，应用层的防火墙，应用层的入侵检测等，属于有点“繁琐”的工程，“撇不掉、理还乱”，大部分甲方团队都没有足够的人力去应付产品线交付的数量庞大的代码，没有能力去实践完整的 SDL，这部分是当下比较有挑战的安全业务，整体比重大于 30%，还在持续增长中。
- ❑ **业务安全**：上面提到的 2)，包括账号安全、交易风控、征信、反价格爬虫、反作弊、反 bot 程序、反欺诈、反钓鱼、反垃圾信息、舆情监控（内容信息安全）、防游戏外挂、打击黑色产业链、安全情报等，是在“吃饱饭”之后“思淫欲”的进阶需求，在基础安全问题解决之后，越来越受到重视的领域。整体约占 30% 左右的工作量，

有的甚至大过 50%。这里也已经纷纷出现乙方的创业型公司试图解决这些痛点。

对整体介绍的部分在前面的篇幅讲得比较多，主要目的是希望“视野”部分不缩水，这些概念在后面篇幅都不打算再展开了。

1.3 互联网企业和传统企业在安全建设中的区别

总体来看，传统企业偏重管理，有人说是“三分技术，七分管理”；而互联网企业偏重技术，我认为前面那个三七开可以倒过来。其实这种说法也是不准确的，到底什么算技术，什么算管理，这些都没有明确的定义。安全领域大部分所谓管理不过是组织技术性的活动而已，充其量叫技术管理。

先说一下传统企业和互联网企业在安全建设需求上的差异。

传统企业安全问题的特征如下：

- 1) IT 资产相对固定。
- 2) 业务变更不频繁。
- 3) 网络边界比较固定。
- 4) IDC 规模不会很大，甚至没有。
- 5) 使用基于传统的资产威胁脆弱性的风险管理方法论加上购买和部署商业安全产品（解决方案）通常可以搞定。

大型互联网企业需要应对如下问题：

- ❑ 海量 IDC 和海量数据。
- ❑ 完全的分布式架构。
- ❑ 应对业务的频繁发布和变更。

同时架构层面需要关注：高性能、高可用性、（水平）扩展性、TCO（ROI）。

在规模不大的互联网公司，传统企业的风险管理方法论是可以沿用的。但在大型互联网公司，传统企业的方法论可能会失效，因为你可能连基础架构上跑什么业务都搞不清，想理清所有系统接口间的调用关系以及数据流去检视设计风险以及设置细粒度的访问控制就是件不现实的事情。产品线极多时，业内没有任何一个团队敢说自己的能力支持全产品

线，对于高速发展的业务，当你理清了你想要的时，说不定架构又发生了变化了。只有对占公司整体营收比较主要的以及培育性质的战略级的核心业务，才有必要去深入调研并随之更新，其他的主要依靠自动化手段。

1. 传统企业的安全建设

从安全建设上来看，传统企业的安全建设是：在边界部署硬件防火墙、IPS/IDS、WAF、商业扫描器、堡垒机，在服务器上安装防病毒软件，集成各种设备、终端的安全日志建设 SOC，当然购买的安全硬件设备可能远不止这些。在管理手段上比较重视 ISMS（信息安全管理体系）的建设，重视制度流程、重视审计，有些行业也必须做等级保护以及满足大量的合规性需求。

2. 互联网企业的安全建设

互联网可分为生产网络和办公网络，即便最近 Google 声称取消内网也是针对办公网络而非生产网络。互联网行业的大部分安全建设都围绕生产网络，而办公网络的安全通常只占整体的较小比重。但是某些传统企业可能完全没有生产网络而只有办公网络，那么网络安全也就变成办公网络的网络安全。但我推测，随着社会“互联网+”进程的加速，很多传统企业也会有自己的生产网络，最终都变成和互联网公司一样的形态。所以对于那些在给传统企业客户提供咨询和解决方案的乙方的工程师，如果不学习互联网安全，也迟早会陷入困境。

互联网企业的生产网络中，安全解决方案基本上都是以攻防为驱动的，怕被黑、怕拖库、怕被劫持就是安全建设的最直接的驱动力。互联网公司基本不太会考虑等保、合规这种形而上的需求，只从最实际的角度出发，这一点是比传统企业更务实的地方。曾遇到过一个例子，说要在服务器上装防病毒软件，推测就知道是传统企业的思路，不是没有真正实践过互联网企业安全就是没被业务线挑战过。在大型互联网企业，仅是性能损耗、运维成本和软件成本这几条就能分分钟把这种需求干掉，更不用进入对于服务器防护这种更实际的话题了。很多标准说到底都是各厂商参与编写，博弈并达成妥协，有利于自己产品销售的代言白皮书，并不是完全站在建设性的角度的，作为乙方给政企客户写解决方案建议书无可厚非，但在互联网公司做企业安全，生搬硬套某些标准就会闹出笑话来。

3. 从量变到质变

对于超过一定规模的大型互联网公司，其 IT 建设开始进入自己发明轮子的时代，安全解决方案开始局部或进入全部自研的时代。例如不会购买硬件防火墙，而是用服务器+Netfilter 的方式自建，不会部署硬件 IDS/IPS，而是用其他方式来解决这个问题。其实不难理解，规模小的时候买台硬件防火墙放在最前面，省事。但是规模大了，难道去买 1000 台硬防放在 IDC 机房？成本上就没法通过批准。再说，基于分布式系统的 CAP 理论和 Map-Reduce 衍生的一系列互联网架构，本质上都具有“无限”的扩展能力，而对于传统的硬件盒子式的解决方案，其设计大多源于对小型网络体系架构的理解，基本不具备扩展能力，完全不能适应大规模的互联网架构。在这种情况下甲方安全团队自己动手去打造完全围绕自身业务的解决方案也就成必然趋势。

4. 大型互联网企业安全建设的方法论

自研或对开源软件进行二次开发+无限水平扩展的软件架构+构建于普通中低端硬件之上（PC 服务器甚至是白牌）+大数据机器学习的方式，是目前大型互联网公司用来应对业务持续性增长的主流安全解决方案。是否真的到了机器学习阶段这个有点难说，但是安全进入大数据时代则是肯定的。

与办公网络和雇员信息安全管理相比，互联网公司的文化比较开放，一般不太会维持激进的安全政策（对雇员做太多信息安全方面的管制和限制），这点也是跟传统企业差别比较大的地方。

也有一些灰色地带，比如 TCO 较高的安全方案，一开始没感觉，但实质是吸毒，随着 IDC 的规模扩张，安全的成本越来越大，最后被自己巨大的成本“毒死”。对于做惯传统行业解决方案且客户手里都有大把预算的顾问来说，对这一点是没感觉的，几千万元的安全整体方案信手拈来，但是放到互联网中，一旦业务规模成倍增长，这些方案最终都会走入死胡同。不止是成本，如果不能做到兼顾宿主的性能，安全架构随整个业务架构水平扩展，保证高可用性，最终安全措施都会走进死胡同。

以安全集成为自身职业亮点的人如果不积极学习，会有很大贬值风险，因为以后不需要堆硬件盒子式的解决方案了，就算堆也不再是原来的堆法。

1.4 不同规模企业的安全管理

1. 创业型公司

对于创业型公司而言，安全不是第一位的，我在唱反调吗？应该只是大实话而已。安全建设的需求应该是：保障最基本的部分，追求最高性价比，不求大而全，映射到技术实现应该是做如下事情：

- ❑ 基本的补丁管理要做。
- ❑ 漏洞管理要做。
- ❑ L3 ~ L7 的基本的访问控制。
- ❑ 没有弱密码，管好密码。
- ❑ 账号认证鉴权不求各种基于条件的高大上的实时风控，但求基本功能到位。
- ❑ 办公网络做到统一集中管理（100 人以上规模）和企业防病毒，几个人的话，就完全不用考虑了，APT 什么的就听一听拉倒了。
- ❑ 流程什么的就没必要去搞了，有什么需求，口头约束一下。
- ❑ 找两篇用到的开发语言的安全编程规范给程序员看看，安全专家们说的 SDL 就不要去追求了，那个东西没有一堆安全“准”专家玩不起来。
- ❑ 系统加固什么的，网上找两篇文章对一下，确保没有 root 直接跑进程，chmod 777，管理后台弱密码对外这种低级错误，当然有进一步需求，也可以参考后面的技术篇中的措施。
- ❑ 实惠一点的，找个靠谱的白帽子兼职做一下测试，或者众测也行。

是不是觉得简陋了一点？我估计很多乙方提供的服务除了卖产品之外也不会比这个效果更好了。不过，现在不少创业公司是拿了不小的风投的，也没那么寒酸。另外一个很重要的特征是，创业公司基本都上公有云了，不会自己再去折腾 IDC 那点事，所以相对而言以上措施中的一部分可以等价于：

- 1) 使用云平台提供的安全能力，包括各种抗 DDoS、WAF、HIDS 等。
- 2) 使用市场中第三方安全厂商提供的安全能力。

具体怎么选怎么用就不展开讲了，不过不要因为迷信云平台关于安全能力的广告而自己不再去设防，毕竟针对租户级的通用型的安全方案其覆盖面和防御的维度是有限的。

2. 大中型企业

这个层次对应市场上大多数公司的安全需求，它的典型特征是，业务营收的持续性需

要安全来保障。公司愿意在 IT 安全上投入固定的成本，通常小于 IT 总投入的 10%，不过这已经不错了。这时候开始有专职的安全人员或安全团队，建设上重视效果和 ROI，会具备初步的纵深防御能力。对应技术上的需求为：

- L2 ~ L7 中的每一层拥有完整的安全设计。
- 对所有的服务器、PC 终端、移动设备，具有统一集中的状态感知、安全检测及防护能力。
- 应用层面细粒度控制。
- 全流量入侵检测能力。
- 无死角 1 天漏洞发现能力。
- 在安全等级较高的区域建立纵深防御和一定的 0 天发现能力。
- 初具规模的安全专职团队。
- 对应用交付有自主的评估和修补能力。
- 从 IT 服务层面建立必要的流程、业务持续性以及风控应急措施。
- 对业务安全形成自己的风控及安全管理方法论。
- 将难以自己实现的部分外包。

好像跟前面的描述比一下子抽象了？是的，这个层级的安全需求没办法很具体地量化。不同的业务模式对应的安全实现还是有很大的不同，加上这个区间里的公司营收规模可以差很大，对应的安全投入也可以差很多。那什么样的公司归入这个区间呢？比如四大行，国内 TOP10 甚至 TOP5 以后的互联网公司，大量的电信、金融、政府、能源等企业都属于这个区间，但我为什么不把 BAT 归入这个区间？这样的分类岂不是不科学？我之所以这样分类完全是从安全建设的复杂度上去考量的，而不是从安全建设的资金投入上去归类的。很多行业（比如金融）的安全投入很大，但这些解决方案大都是靠花钱就能买来的，而 BAT 的方案花钱不一定能买得到，很多都需要自己动手去打造，对安全团队的定位、能力和需求完全不一样。那 BAT 以外较大的互联网公司呢？我觉得他们会玩些各自特点的小花样，但是不会进入大范围的复杂的安全建设，这是由公司的整体面和业务决定的，不需要过分保障和超前业务的安全建设。

再往上一层是大型互联网企业。

1.5 生态级企业 vs 平台级企业安全建设的需求

生态级企业和平台级企业之间的安全建设需求不仅仅是量的差别而是质的差别。

1. 差别的表象

主要差别表现在是否大量地进入自己造轮子的时代，即安全建设需要依托于自研，而非采用商业解决方案或依赖于开源工具的实现。

那么平台级公司和生态级公司的区别又在哪里？从表象上看，生态级公司的大型基础架构如果用传统的安全方案几乎都无法满足，所以会大量的进入自研。而平台级公司则会依赖开源工具更多一些，不会对所有解决方案场景下的安全工具进行自研。如果有预算也会优先投在“业务安全”侧，比如反欺诈平台等等，而不会自己去搞入侵检测。当然，这有可能是个伪命题，有可能随着时间的推移，乙方公司也开始提供具有可扩展性、能应对分布式架构的方案，或者当时间尺度拉得长一点，平台级公司每年在自研上投入一点点，多年之后也具备了 BAT 级别的安全能力也并非完全不可能。不过这些都是理想状况，现实总是受到多方面因素制约的。

2. 差别的成因

第一个因素是技术驱动在底层还是在应用层驱动业务。表象上，平台级公司和生态级公司都是以 PC 端 Web 服务为入口的平台应用和以移动端 APP 入口的移动应用。有的依赖于一些 PC 客户端或移动端偏底层的 APP，但在技术实现方式上，平台级公司更多地直接使用或少量修改开源软件，而生态级公司的 IT 基础设施则会类似于 Google 的三篇论文一样，不仅仅停留在使用 and 少量修改，而是会进入自己造轮子的阶段。其中所造的轮子是否对业界有意义这种问题暂时不去评价，但对应的安全建设则反映出平台级公司的安全主要围绕应用层面，而生态级公司的安全会覆盖基础架构和应用层面两块。直接使用开源工具的部分交给社区去处理，自己跟进打补丁就行了，但如果是自己开发的，那么就需要自己去解决一揽子的安全问题，比如 Google 造了 Android 这个轮子，那 Android 一系列的安全问题 Google 需要自己解决，比如阿里自己去搞了一个 ODPS，那阿里的牛人也需要解决这个，再比如华为在物联网领域造了 LiteOS 这个轮子，自然也要去处理对应的问题，而这些偏底层的问题显然早已超出应用安全的范畴，也不是一般的甲方安全团队有能力应对的。其实有些平台级公司也是发明了一些轮子的，比如自动化运维工具，比如一些 NOSQL，不过 IDC 规模两者之间仍然差得比较远，上层的业务复杂度也有差距，支持的研发团队的规模也有差距，对安全工具的自动化能力和数据处理规模仍然存在阶梯级的差别，这一点也决定了为什么要自研。安全其实只是 IT 整体技术建设的一个子集，当整体技术架构和实现方式进入自产自销阶段时，安全建设也必然进入这个范畴。对于很多实际上依靠业务和线下资源驱动而非技术驱动的互联网公司而言，安全建设去做太多高大上的事情显然是没有必要的。

第二个因素是钱，钱也分为两个方面：1) 成本；2) ROI。假设安全投入按 IT 总投入的固定 10% 算，又假设生态级公司的安全建设成本是平台级公司的 5 ~ 10 倍，这个成本除了带宽、IDC 服务器软硬件之外，还有技术团队，加起来才是总拥有成本 TCO。10 个人的安全团队和 100 个人的安全团队能做的事情相差太大，具体可以参考我在知乎上的帖子“为什么从事信息安全行业一定要去大公司？”。还有一方面则类似于“去 IOE”，上面提到目前对于大型互联网没有合适的安全解决方案，即使有，这个成本可能也会无法接受，所以假如乙方公司能推出既能支撑业务规模，又具有性价比的方案，我认为甲方安全团队真的没有必要再去造轮子了。

第三个因素是人。安全团队的人员数量也只是一个很表面的数字，安全团队的构成才是实力，能围到大牛的安全团队和由初级工程师组成的安全团队显然是不一样的，首先前者的成本不是所有的公司都能接受，其次，平台不够大即使大牛来了也未必有用武之地。大多数平台级公司中安全团队的知识和经验集中在 Web/App、应用层协议、Web 容器、中间件和数据库，生态级公司则扩展至系统底层、二进制、运行时环境和内核级别，能力积累也存在差别。这里并无褒贬之意，仅在说明业务对技术的需求不一样。

3. 平台级公司的“止步”点

那么，平台级公司在安全建设上是不是就没有乐趣呢？其实这类公司也玩一些小花样，比如修改 SSHD、LVS，加入一些安全特性。可能也会自己定制一个 WAF，或者搞搞日志的大数据分析。但比如涉及 DPI、全流量入侵检测、SDN、内核级别的安全机制，基本上都不会介入，对于一个规模不是特别大的平台级公司的甲方安全团队而言，这些门槛还是有点高。

4. 生态级公司的竞技场

生态级公司是不是全领域进军自己造轮子呢？也不是，主要工作还是在入侵检测、WAF、扫描器、抗 DDoS、日志分析等领域。在 SDL 环节上可能也会自己研发些工具，但很与比直接使用商业工具更短平快。阿里钱盾、腾讯管家、百度杀毒这些都跟 360 客户端一样与生产网络没什么关系，就不去讲了。另外自研工具有一个原则：都限定在“民用”领域，不会自己去发明一个 RSA 算法这样的东西。

国内的平台级公司里也有一个例外：数字公司，因为其主营业务是信息安全，所以就没有安全投资固定占比理论的影响。

1.6 云环境下的安全变迁

云计算的本质是改变企业需求方通过传统的渠道获取 IT 资源的形式。传统的方式是一个企业假如要构建信息化的能力，必须要采购硬件，采购软件，维护一个较大的 IT 团队，TCO 很高。但是，到了云计算时代，这一切你都不需要，你只要轻点鼠标就可以获取大量的计算、存储和网络资源，并且不再需要专门的人员去 IDC 机房维护服务器，不需要大量的运维人员，甚至某些通用的应用开发都省了，你可以将手头的 IT 预算用于最需要的部分——完全聚焦于自己的业务，而不用费大量的精力维护基础设施，甚至资源的获取变得弹性：需要时轻易获取大量甚至海量的计算资源，用完后可以及时释放，不用担心资产闲置和老化。在 IT 产业的销售模式中被颠覆的一环是，传统企业不再直接购买服务器存储商业数据库，而是通过云计算平台获取。同样地安全到了云计算时代，企业客户会更希望通过云化的方式来获取和整合安全的解决方案，例如 SaaS（Security as a Service），这就要求安全产品或解决方案本身需要支持虚拟化、软件化、分布式、可扩展性，并且利用大数据和人工智能，利用云端无限的计算和存储能力，缓解传统安全解决方案中数据的离散、单点的计算能力不足，信息孤岛和无法联动等问题。

1. 云的租户

对使用云的租户而言，云平台自身以及应用超市 Marketplace（类似 APP store）集成了各个安全厂商的云安全产品以及可托管的安全专家服务。如果自身没有太强烈地要主导安全实践的意愿，通常情况下通过应用超市和云平台免费提供的安全功能就可以快速构建基础的安全能力。如果希望得到进阶的安全防护，则必须自己进一步动手。

2. 云安全提供商

在传统的方案中，安全厂商以提供硬件安全产品和安全服务为主，而在云环境下，硬件形式的安全方案会越来越不合拍，与之相比，把竞争力构建在软件层面的安全方案会成为云上的主流。相比过去面对面提供安全服务，现在则转变为在租户侧部署各种安全传感器，通过在云端汇聚安全度量数据，结合威胁情报或由专家解读数据来提供可管理的一站式安全服务。



安全的组织

很多人忽略组织，但实际上组织是比技术和流程更重要的东西，“人”是所有问题的决定性因素。本章就讲一下安全组织中的人。

2.1 创业型企业一定需要 CSO 吗

1. 招聘方的诉求

当下不少创业型公司都在找 CSO，也有找到我的，就顺带分享一下我对这个问题的思考。首先这个问题即便是对同一个人而言可能答案也会因时而变，其次 CSO 只是一个代表性的称呼，大家不要过于纠结一定要做什么事才算 CSO，CSO 和 CISO 又有什么区别之类的，在这个语境下用来指代招聘方想找拥有全局安全管理经验的人，甚至最好是资深的从业者，他能带一支哪怕是几个人的安全团队，并能把安全相关的事全部揽过去。

2. 不同阶段的需求

对于尚在天使融资阶段，找个 CSO 大多就是去忽悠投资人的，通俗一点理解就是凑一支履历光鲜的队伍去“骗钱”。能不能做事情这个很难说，也不能说人家一定做不了事情，这种只能事后诸葛亮的盖棺定论，随便说人不靠谱也是不负责任的，不过我想对于大多数

有不错岗位的人而言应该是不会去的。

对于拿到 A 轮、B 轮投资的阶段，业务方向已被证明，业务量不大但进入快速成长期，问题层出不穷。CEO 和 CTO 觉得头疼并且想把这部分压力转嫁出去，理论上是应该找一个懂安全的人的，但是现在这个时间点（也就是 2016 年）真的不是一个好时间，在当下很多公司用数百万也经常找不到合适的 CSO，创业型公司要在这个时间点上争夺安全人才真的是很累的一件事。建议找 2 ~ 3 个靠谱的安全工程师，然后 CTO（技术总监）、运维 Leader、开发的架构师这几个角色快速补一点安全的知识和方法论，哪怕是充当救火队长赶鸭子上架，大家配合一下，应该也能把安全撑起来，不一定非要找一个大牛级的 CSO，因为有时候业务量没那么大，不一定需要很高级别的人，且创业型公司为了保持自己的鲜活也不需要套用大公司的流程和方法，做好基础的运维安全，代码安全，加强一些安全意识，不出问题的时候腾出时间来研究一下下一步怎么做，跟时间和业务增长速度赛跑，跑着跑着，应该就会越来越轻松了。当然，作为创业者，如果你有一个安全领域的朋友愿意帮你出谋划策，偶尔回答一下安全建设如何做这类问题，其实那 CSO 的需求也就解决了，代价只是请吃几顿饭。

对于拿到 C 轮、D 轮、E 轮投资的阶段，业务初具规模，开始朝更高的目标冲刺，同时能给人画更多的“饼”，如果之前已经招到靠谱的安全工程师，那这个时候应该已经成长起来，承担起 Leader 的角色，通常也不需要从外部招 CSO 了。这个时候想从外部招 CSO 的都是之前主观的或被迫的不太重视安全，出了问题才想要头痛医头脚痛医脚。这个时候的业务形态跟大公司比应该是“麻雀虽小五脏俱全”，安全管理上应该是有很多相似的地方了，对于初具规模的业务而言安全管理的经验很重要，可以直接从大公司挖人。当然了级别较高的人估计还是挖不动的，能挖动的主要就是骨干那一层的人吧。通常你需要容忍他们业务上有深度但不一定面面俱到，很可能情商和管理能力也差强人意，不过能解决你的问题就行，面面俱到的人才毕竟还是太贵了。对于那些估值超过 100 亿美元的创业型公司，还是建议找高端人才。

对于准备冲 IPO 的阶段，这个时候想必不会囊中羞涩，如果你是这类公司的 CXO，还为找不到人感慨万千，我想也许是心态还不够开放，亦或许是给人的“诚意”还不够，对此我也给不了太多建议了，毕竟市场上高级人才只有那么一小撮人，来不来，去不去取决于那一小撮人的意愿，被打动了自然来，没被打动的自然不去。

3. 创业型企业的挑战

对选择转会的 CSO 去创业公司是否有挑战？有，也没有。首先做的事情往往还是从头建团队，把过去建设安全体系的过程从零开始再做一遍，从这个角度讲重复过去做的事情尽管业务有所不同但过程和结果上未必有挑战。有“挑战”的反而是如何在创业公司的条件下招募成员，维系团队，在很多流程及界限不分明的情况下推动事情落地，仅此而已。对 CSO 本人而言得到的实践机会未必有大公司多而广，因为安全是一个随企业规模而复杂化的工作。但是，如果你是 CSO 的跟班小弟，那你可能是成长最快得到锻炼最多的人，因为你经历了安全建设从无到有飞速发展，从简单到自动化的过程，这个过程不是人人都能经历的，如果你进入 BAT 在一个很细的分支上耕耘几年未必能积累到这种“全局视野”，相反在这种环境下才能快速积累。所以除非公司 IPO，CSO 都不是团队里收益最大的人，但跟班小弟却是最大的赢家。

2.2 如何建立一支安全团队

如果要去一家公司领衔安全建设，第一个问题就是如何建立安全团队。上面提到不同的公司状况对应的安全建设需求是不一样的，需要的安全团队也是不一样的，所以我按不同的场景来深入分析这个问题。

在目前国内的市场中，BAT 这种公司基本是不需要组团队的，对安全负责人有需求的公司大约是从准生态级互联网公司、平台级互联网公司、大型集团的互联网+，到千千万万的互联网创业型公司。

1. 极客团队

如果你在一个小型极客型团队，例如 Youtube 被 Google 收购前只有 17 个人，在这样的公司里你自己就是安全团队，俗称“one man army”。此时一切头衔皆浮云，需要的只是一个全栈工程师。

2. 创业型企业

对于绝大多数创业型企业而言，就像之前所说的，CSO 不一定需要，你拉两个小伙伴

一起去干活就行了，今天 BAT 的安全总监们，当年也都是干活的工程师小伙伴，10 多年过去了，工程师熬成了“CSO 大叔”。当你的公司变成 BAT 时，只要你成长得够快，也许下一个“CSO 大叔”就是你自己。

3. 不同的能力类型

搞安全的人现在其实不好招，很多企业都招不到安全负责人，所以会有一堆没有甲方安全实操经验或者没有整体安全经验的人被推上安全团队领导的岗位。对绝大多数公司而言，安全建设的需求都是聚焦于应用的，所以安全团队必然也是需要偏网络和应用的人。大牛显然是没必要的，而懂渗透，有一定网络系统应用攻防理论基础的人是最具培养价值的。除此之外乙方的咨询顾问、搞安全标准的、售前售后等在这个场景下的培养成本都很高，不具有短期 ROI，所以都不会是潜在的候选者。在安全技术领域里，其实只有两类人会有长期发展潜力：第一类是酷爱攻防的人，对绕过与阻断有着天生的兴趣；第二类就是可能不是很热爱安全，但是 CS 基础极好的程序员，这一类人放哪里都是牛人，第一类则跟行业相关。粗俗一点儿的说法找几个小黑客就能做企业安全了？这种观点是不是有人会觉得偏科的厉害了。确实我也认为会渗透跟做企业安全的系统性建设之间还是有比较大的鸿沟的。仅仅是说在有限选择的情况下，假如你不像某些土豪公司一样随便就能招到高手，那么可选的替代方案中最具可行性和性价比的是什么呢，之所以选会渗透懂攻防的人，那是因为这类人具备了在实践层面而不是理论层面真正理解安全工作的基础，在此基础上去培养是非常快的，策略、流程、标准、方法论可以慢慢学，因为这些都不是救火时最需要的技能，而是在和平年代且规模较大的公司才需要的东西。看有些公司的招聘工程师的要求里还写着要证书什么的，不禁感慨一下，很多能做事的“少年”其实根本没有证书，有证书的人通常适合去做乙方的售前而不是甲方的安全工程师。甲方招聘要求证书的，基本上都是传统企业，互联网企业这么写的应该怀疑一下那里的整体水平。当然，这个说法对安全负责人的招聘不成立，因为国内的 CTO 很少有懂安全的，招聘者其实也不知道安全总监到底需要哪些技能，随便拷贝了一个也很正常，高端职位的 JD（职位描述）很多时候都是模糊的，除了一个头衔之外，其他都要聊了才知道，这时候你就不要去嫌弃 JD 怎么写的这么差，毕竟老板也不懂这事该怎么做，找你就是为了解决这个问题。

单纯攻防型的人在前期培养比较快，但当安全团队随着公司规模和业务快速成长时，思维过于单点的人可能会出现“瓶颈”。后面会提到安全建设实际上是分阶段的，而且是系统性的，视野和思路开阔的人会从工程师中脱颖而出，成为安全团队的领导。

4. 大型企业

对于比较大型的平台级公司而言，安全团队会有些规模，不只是需要工程师，还需要有经验的 Leader，必须要有在运维安全，PC 端 Web 应用安全以及移动端 App 安全能独当一面的人，如果业务安全尚有空白地带的话，还需要筹建业务安全团队。

5. 超大型企业

准生态级公司建安全团队这种需求比较少，但因为笔者曾被问及这样的问题，所以就思考的结果写出来。对于这种级别的公司，由于其业务线比较长，研发团队规模通常也比较庞大，整个基础架构也构建于类似云计算的底层架构之上（姑且称之为私有云吧），光有应用安全的人是不够的，安全的领头人必须自己对企业安全理解够深，Leader 这一级必须对系统性的方法论有足够的了解。随便举些例子，1）在出安全事件时如果 Leader 的第一反应是直接让人上机器去查后门的；2）对运维系统变更风险不了解的；3）对在哪一层做防御性价比最高不熟悉的；4）不明白救火和治病的区别的（这种思路会一度体现在提的安全整改建议上），诸如此类的状态去担任 Leader 就会比较吃力（Leader 上面的安全老大自然也会很吃力）。另外 Leader 的跨组织沟通能力应该比较高，在这种规模的公司，不是你的安全策略提的正确就一定会被人接受的。团队里还应该有 1 ~ 2 个大牛级人物，所以带队人自己应该是在圈内有影响力的人，否则这些事情实践起来都很难。

实际上当你进入一个平台级公司开始，安全建设早已不是一项纯技术的工作，而技术管理上的系统性思路会影响整个安全团队的投入产出比。

甲方安全建设方法论

那些国际安全标准、模型和理论已经讲了千万遍，方法论的东西也许是好东西，但是一般人可能都没时间去读完，有时候也会觉得理论脱离实践，所以本章讲的方法论都从实操出发的。

3.1 从零开始

本篇谈一下 CSO 上任之初要做什么吧。很多没做过甲方安全的人也许都没有头绪，或者你只是接触甲方安全的一个细分领域而不是全貌，也许我说的能为你省点脑汁，因为开始是最难的，等过了这个阶段找到了感觉，后面的路就平坦了。

1. 三张表

上任之初你需要三张表。第一张表：组织结构图，这些是开展业务的基础，扫视一下组织结构中每一块安全工作的干系人。例如行政、HR、财务部门是跟公司层面信息安全管理的全局性制度的制定和发布相关的部门，内部审计也跟其强相关；基础架构的运维团队，运维安全相关的要跟他们合作；研发团队，可能在组织结构中分散于各个事业部、各产品线，不一定叫研发，但本质都是产品交付的团队，应用安全和基础服务器软件安全相关的要找他们，也是 SDL 实施的主要对象；运营、市场、客服类职能他们可能没有直接的

系统权限，但是会有一些诸如 CMS 的后台权限（被社工的对象），广告的引入发布（挂马的 iframe，黑链）等乱七八糟的安全问题的关联者，他们也是某些重大安全事件上升到社会影响的危机管理的公关部门；（大）数据部门，因为安全也要用到大数据，是复用一套技术架构还是自己搞，这个取决于每个公司的实际状况，有的自己搞，有的则复用；产品部门，一些跟业务安全和风控相关的安全建设要跟他们合作；CXOs：这里泛指组织中的决策层，什么问题要借助他们自己拿捏吧，双刃剑。

第二张表：每一个线上产品（服务）和交付团队（包括其主要负责人）的映射。这张图实际就是缩水版的问题响应流程，是日常安全问题的窗口，漏洞管理流程主要通过这些渠道去推动，一个安全团队的 Leader 通常需要对应于一个或若干产品的安全改进。不过这里也要分一下权重，比如支撑公司主要营收的产品需要一个主力小团队去负责其 SDL 全过程，而边缘性的产品一个小团队可以并发承接好几个甚至 10 个以上的产品，粒度相对粗一点过滤主要的安全问题即可。通常这样做符合风险管理方法论，但对于深知大公司病又创业过的我来说，还是稍微有些补充的看法，很多成长中的业务，出于起步阶段，没有庞大的用户群，可能得不到公共职能部门的有力扶持，例如运维、安全等，明日之星的业务完全可能被扼杀在摇篮里，这种时候对有责任心的安全团队来说如何带着 VC 的眼光选择性的投入是一件很有意思的事。在一个公司里是安全团队的话语权大还是支柱产品线的话语权大？当然是支柱产品，等产品成长起来了再去补安全的课这种事后诸葛亮的事情谁都会做，等业务成长起来后自己都能去建安全团队了，不一定再需要公共安全团队的支持。锦上添花还是雪中送炭，业务团队的这种感受最后也会反馈给安全团队。

第三张表：准确地说应该是第三类，包括全网拓扑、各系统的逻辑架构图、物理部署图、各系统间的调用关系、服务治理结构、数据流关系等，这些图未必一开始就有现成的，促成业务团队交付或者自己去调研都可以，以后的日常工作都需要这些基础材料。如果运维有资产管理也需要关注一下。

2. 历史遗留问题

到了这里你是不是跃跃欲试，想马上建立完整的安全体系了？估计有人恨不得马上拿扫描器去扫一遍了，别急，就像那首儿童歌曲唱的“葡萄成熟还早得很呐！”，你现在的角色还是救火队长，离建设还早，这跟你的能力和视野没关系，这是客观情况决定的，一个安全没有大问题的公司通常也不会去找一个安全负责人。找安全负责人的公司意味着都有

一堆安全问题亟待处理。这里就引申出一个问题，一般情况下都是出了比较严重的安全问题才去招聘安全负责人和建立专职的安全团队的，就是说这些系统曾经被渗透过，或现在正在被控制中，没有人可以确定哪些是干净的，哪些是有问题的，而你加入的时间点往往就是安全一片空白还不确定是不是正在被人搞。有人说系统全部重装，那你不如直接跟老板说全部系统下线，域名注销，关门算了，那样子显然是行不通的，所以防御者不是时时处处都占上风。这个问题只能灰度处理，逐步建立入侵检测手段，尝试发现异常行为，然后以类似灰度滚动升级的方式去做一轮线上系统的后门排查。

3. 初期三件事

一开始的安全不能全线铺开，而是要集中做好三件事，第一件是事前的安全基线，不可能永远做事后的救火队长，所以一定要从源头尽可能保证你到位后新上线的系统是安全的；第二件是建立事中的监控的能力，各种多维度的入侵检测，做到有针对性的、及时的救火；第三件是做好事后的应急响应能力，让应急的时间成本更短，溯源和根因分析的能力更强。

一边熟悉业务，一边当救火队长，一边筹建团队基本就是上任后的主要工作了。如果团队筹建得快，这个阶段 2 ~ 3 个月就可以结束了，但以目前招聘相对难的状况来看可能需要 4 ~ 6 个月。

3.2 不同阶段的安全建设重点

1. 战后重建

救火阶段过去之后会进入正式的安全建设期。第一个阶段是基础的安全建设，这一期主要做生产网络和办公网络的网络安全的基础部分。也就是在前面 1.4 节“不同规模企业的安全管理”介绍的大中型企业对应的那些需求（当然也包括中小企业的那些）。完成的标志：一方面是所提的那些点全都覆盖到了，另一方面是在实践上不落后于公司的整体技术步伐，比如运维侧在用 Puppet、SaltStack 之类的工具实现了一定程度的自动化运维，那你的安全措施也不好意思是纯手工的对不对，如果产品团队交付已经在用持续集成了，那你是不是也至少提供个带点自动化的代码检查工具，而不是纯肉眼去 Ctrl+F？这一部分其实是很多

人眼中甲方安全的全部内容，不过我觉得远不能止于此。如果这个场景切换到准生态级公司，也许要变化一下，直接向全线工具自动化看齐，一开始就同步自研必要的工具。

2. 进阶

以上算是解决了安全的温饱问题，第二阶段就是要向更广的方向拓展。一是广义的信息安全，以前是在忙于解决不被黑而抽不出身，现在安全相关的事情都要抓起来，从只对接内部 IT，运维和研发部门扩展到全公司，跟安全相关的环节需要加入必要的流程，以前下线的硬盘不消磁的现在要重视起来了，以前雇员可以随意披露公司的信息以后就不可以了，以前雇员离职的账号不回收的现在开始不可能了，以前 DBA 可以给数据库插条记录然后去电商上卖装备的，那种事从此开始要一刀切断，诸如此类的事情还有很多。其实这个时候你可以把 ISO27001 拿出来看看了。二是业务安全，比如用户数据的隐私保护，之前安全只是作为保障而不是一种前台可见的竞争力，但现在安全需要封装起来对用户可见，对产品竞争力负责，如果公司已经发展到一个很大的平台，盗号问题都解决不了的，我觉得真的需要考虑一下自己的乌纱帽问题。这一部分对安全圈人士而言可能并不高大上，可能没太多值得拿出来炫技的部分，但是我认为这些是务实的安全负责人需要考虑的问题，这些属于经营管理者视角下的一揽子安全问题，如果这些问题不解决而去发明 WAF 发明 HIDS 去，尽管可以拿到安全圈来发两篇文章炫耀一下，但从职责上看属于本末倒置，直接影响公司营收的问题需要先解决。之所以把业务安全放在第二阶段而不是去优化安全基础架构是因为投入产出的边际成本，投在业务安全上，这一部分产出会比较直观，对高层来说安全从第一阶段到第二阶段一直是有明显可见的产出，而如果此时选择去优化基础安全能力，这种产出受边际成本递增的影响，效果会极其不确定，而这时候业务安全问题频发，就会被倒逼至两难的境地，一则优化基础安全的工作做了一半，一则又要考虑是否中途转去做点救火的事情，而安全产出是安全团队对公司高层影响力的所在，只有看到持续的产出才会影响力增加，才会有持续的投入，尤其在老板不是技术出身的公司，他也许很难理解你去发明 WAF 的价值，他只会问盗号这么严重怎么不解决。这个问题从工程师的视角和管理者的视角得出的结论可能完全不同，安全对高层的影响力是安全团队在公司内发展壮大基础，这是很多甲方安全团队之痛，你可以对比一下自己所在的环境，安全团队的负责人对大方向的把控上是不是做到了可持续发展，好吧，这个问题有点尖锐。

3. 优化期

第三个阶段会感到开源工具不足以支撑业务规模，进入自研工具时代。其实做攻防和研发安全产品完全是两码事，存在巨大的鸿沟，如果拿做攻防的团队直接去做安全工具开发，恐怕挫折会比较多，即便有些研究员擅长做底层的东西，但对于高并发生产环境的服务器工具而言，还是有很大的门槛。另一方面做攻防和做研发的思路也截然不同，此时其实是在交付产品而不是在树立安全机制，所以要分拆团队，另外招人。

4. 对外开放

第四个阶段，安全能力对外开放，成为乙方，不是所有的甲方安全团队都会经历这个阶段，故而此处不展开。不过我想最重要的区别是，经营意识，成本意识，运营，整体交付，2B 和 2C 的区别，线下最后一公里。

3.3 如何推动安全策略

这是一个在安全负责人的面试中经常被提及的问题，也是在现实生活中甲方团队天天面对的问题。如果你不是正巧在面试，那怎么回答这个问题其实不重要。

1. 公司层面

首先，推动安全策略必须是在组织中自上而下的，先跟高层达成一致，形成共同语言，对安全建设要付出的成本和收益形成基本认知，这个成本不只是安全团队的人力成本和所用的 IDC 资源，还包括安全建设的管理成本，流程可能会变长，发布链条会比过去更长，有些产品可能会停顿整改安全，安全特性的开发可能会占用正常的功能迭代周期，程序员可能会站起来说安全是束缚，这些都是需要跟各产品线老大达成一致的，他们要认同做安全这件事的价值，你也要尽可能的提供轻便的方法不影响业务的速度。在规模较大的公司，只有自上而下的方式才能推得动，如果你反其道行之，那我估计安全团队多半在公司是没有地位的，顶多也就是在微博或者技术博客上有些外在的影响力。往下攻略去影响程序员和 SA/DBA 的难度肯定比往上攻略去影响 CXO/VPs 的难度小，但如果一开始就选择一条好走的路，实际对安全团队来说是不负责任的，作为团队领导你必须直面困难，否则安全团队就只能做些补洞、打杂、救火队长的事。

2. 战术层面

在我过去的文章“CSO的生存艺术”<http://bbs.chinaunix.net/forum.php?mod=viewthread&tid=1163970>中提到一些因势利导的方法，现在回头看这些方法固然值得一用，但也不是最先应该拿出来的。很多时候我认为甲方安全团队思路受限的地方在于：总是把安全放在研发和运维的对立面上，认为天生就是有冲突的。不信回顾一下开会时是不是经常有人对着研发和运维说“你们应该如何如何……应该这么做否则就会被黑……”诸如此类的都反映出意识形态中安全人员觉得研发就是脑残，运维就是傻叉。为什么我之前用了“合作”一词，其实换个角度，你真的了解开发和运维吗，是不是找到个漏洞就心理高高在上了？你是在帮助他们解决问题，还是在指使他们听你行事，如果你是产品研发的领头人，听到下面的程序员对安全修改怨声载道会怎么想？我的建议是从现在开始不要再用“你们”这个词，而改用“我们”，自此之后便会驱动你换位思考，感同身受，真正成为助力业务的伙伴。其实有些问题处理的好，真正让人感到你提的建议很专业，研发和运维人员不仅会接受，而且会认为自己掌握了更好的编码技能或者安全配置技能而产生正向的驱动力。再通俗一点，如果安全跟研发的人际关系是好的，提什么建议都能接受，即如果我认可你这个人，那么我也认可你说的事；反之，如果人际关系不好，那不管你提的对不对，我就是不愿意改，仅仅是迫于CTO的压力不得不改，但我心理还是有怨气，我还是想在代码里留个彩蛋。利用高层的大棒去驱动可能是一种屡试不爽的技巧，但我认为不是上策。

安全策略的推动还依赖于安全建设的有效性，如果大家都看到了安全策略的成效，都认为是有意義的，那么会支持进一步推动安全策略在整个公司的覆盖率和覆盖维度；反之，如果大家都觉得你只不过是在玩些救火的权宜之计，心理可能会觉得有点疲劳，后续自然也不会很卖力帮你推，因为没有认同感。所以安全的影响力是不是完全依赖于高层的重视，我觉得有关系，但也跟自己的表现有很大的关系。CTO肯定要平衡开发、运维、安全三者的关系，不会一直倾向性为安全撑腰，而运维和研发的头肯定都是希望有一个强有力的做安全的外援。在别人心中是不是符合需求且值得信赖这个只有自己去评估了。

至于程序员鼓励师，我姑且认为那是一种实施层面的权宜之计，同时反映出安全行业比较缺少既懂技术且情商又高的人。

3.4 安全需要向业务妥协吗

1. 业界百态

在安全行业 5 年以下的新人得到的灌输基本都是“安全不可或缺”，老兵们可能也有点“看破红尘”的味道，觉得高层重不重视安全也就那么回事。对乙方来说高声呼吁安全的重要性哪怕是强调的有点过头也可以理解，因为是赖以生存的利益相关者，靠它吃饭，影响股价。而对于甲方，实际上要分几种，第一类认为安全压倒一切，且心口一致。持有这类想法的实际又可以细分为两种人，第一种对安全行业涉猎不深，还停留在原始的执念阶段，第二种人的思想可以表达为“业务怎么样与我无关，只要不出安全问题，业务死了都无所谓”，这两种从表象上看都属于第一类，但本质上不同；而第二类人口头唱安全重要，但心里还是会妥协。可见甲方安全团队是形形色色的。

2. 安全的本质

撇开上述业界百态，先看安全管理的本质是什么？安全的本质其实是风险管理，绝对的安全可能吗，说绝对安全本身就是个笑话。就像知名黑客袁哥说的，哪怕是 Fireeye 这样的公司也一样会被 APT，原因是攻防不对等，防御者要防御所有的面，而攻击者只要攻破其中一个面的一个点就可以了，公司几千人的客户端行为不是安全管理员能决定和预测的。在所有的面上重兵布防可不可以，理论上可以，但实际绝对做不到。接近于绝对安全的系统是什么样的？尽可能的不提供服务，提供服务也只提供最单调的数据交互模型，尽可能少的表现元素，那样的话还是互联网吗，还有用户体验可言吗？而且安全和成本永远要追求一个平衡。假设一个大中型互联网公司的安全建设成本从 0 ~ 60 分需要 1000 万，60 ~ 80 分需要 2000 万，80 ~ 90 分需要 5000 万，90 ~ 95 分需要 2 亿，这种边际成本递增是很多公司无法承受的，只能追求最佳 ROI，虽然最佳 ROI 难以衡量，但绝大多数人不会拿出收入的 50% 去投安全建设。

3. 妥协的原则

既然安全建设的本质是以一定的成本追求最大的安全防护效果，那一定是会有所妥协的。于是反过来揣摩一下那些说宁可业务死也要做安全的观点的初衷是什么呢，也许你猜到了，怕担责任！因为业务死了安全团队不承担责任，他们可以说“你看安全不是没出问题

嘛!”这固然是一种保护自己的方法，但是从公司的角度看这就有待商榷了。我认为安全本质还是为业务服务，如果业务死了，即使安全做得再好也没价值，更准确一点说，安全需要为业务量身定制，如果业务要轻装上阵，你给他重甲也不行，只能穿防弹背心。安全做得过于重度都是不合适的。相对而言第二类人拥有更加积极的心态，坚持原则又懂得给业务让路，只是要把握好分寸，避免自己的好心被人利用，成为安全问题不整改的免责窗口，那样就事与愿违了。

安全做得不称职的表现，除了“无视业务死活”，还包括：用户体验大打折扣，产品竞争力下降；公司内部流程大幅增加，严重影响工作效率；限制太多员工满意度严重下降，人员流失；规章制度太多，以至于公司文化显得不近人情……这些都属于安全做过头的表现。

有的人出发得太久，以至于忘了初衷是什么。

那么哪些可以妥协，哪些必须坚守呢？高危漏洞，有明显的利用场景，不能妥协。重要的安全特性，比如公有云中的 VPC，底层缺少一个安全特性，直接会导致安全建设的上层建筑失去了“地基”，整个都不牢靠了，这种还是要坚持，可以不精致，但必须有。

对于不痛不痒的漏洞，以及待开发的安全功能，如果开发周期很长，受众群体很少，使用该功能的用户比例极少，边缘性产品，只影响某个中间版本到下个版本被其他机制完全取代了，诸如此类的情况可以考虑酌情妥协。当然这还会涉及另一个话题在不同的维度解决问题，这个之后再展开。

妥协并非退让，而是大局观，试想公司业务没有竞争力时，做安全的一样面临窘境，无论如何都要看主营业务的脸色，与其被动跟随不如快出半个身位。

最后补充一点：妥协不应该发生在工程师层面，而是应该在 Leader 和安全负责人这个层面。如果在安全工程师自己提的整改方案这个层面上，自己主动开始妥协了，那后面很多事情就没法做了。

3.5 选择在不同的维度做防御

攻击的方法千千万万，封堵同一个安全风险的防御方法往往不止一种，如何选择性价

比最高的手段是甲方安全从业者需要权衡的。

1. 技术实现维度场景

在纵深防御的概念中（参考后面的“技术篇”）企业安全架构是层层设防，层层过滤的，常见漏洞如果要利用成功需要突破几层限制，所以退一步对防御者而言有选择在某一层或某几层去设防和封堵的便利，比如 SQL 注入，治本的方法当然是代码写对，治标的方法 WAF 过滤，中间的方法 SQL 层过滤，从效果上说治本的方法固然最好，但在现实中总会遇到各种各样的问题而无法全部选择最安全的解，最后是退而求其次，还是选择某一层或者某几层去防御，需要整体考虑。比如 SQL 注入如果在 HTTP 层面去解决无论是静态规则还是机器学习都需要对抗 HTTP 编码的问题，不是解决这个问题 ROI 最佳的点，但是在 SQL 层面，一切 SQL 语句都是真实的。现实生活中唯一的问题只是你能不能在最佳的点上推动解决方案。

2. “一题多解” 的场景

假如同一个问题有 >1 种解决方案，可能会因场景不同而面临选择。比如对于 SSH 蠕虫的暴力破解，你可以选择：1）使用证书；2）选择外部防火墙上关闭 22 端口，只通过堡垒机登录；3）修改 SSHD 监听非标准端口；4）修改 sshd 源代码对源限制，只接受可信的客户端地址；5）使用类似 fail2ban 这样的工具或自己写 shell 脚本，或者所谓的 HIPS 的功能。乍一看有的做法比较小众，有的则属于偏执狂式的。1），2），5）属于大多数人都认同的普遍的做法，4）和 5）看上去都比较小众，有的人认为可能不适合用作生产网络，其实要看场景。对于大型互联网公司内部自用而言：4）和 5）其实都成立，尽管偏离了业界标准，但只要在公司内部的自治生态里做到“一刀切”就可以，业界普遍是 SSHD 跑在 22 端口，你可以让公司内部的全都跑在 50022 端口，只要公司内部的服务端全都维持这个统一策略，但是这个方案价值不大的地方在于这种信息不对称很容易打破，你花了那么大力气让运维们都去连 50022 端口了，但攻击方很快就能知道，然后努力就白费了。而对于开源软件的修改，如果基础架构研发比较强大，Linux、Nginx、SSHD、MySQL 这些全都可以是改过的私房菜，加点有意义的安全功能也未尝不可。不过中小型公司不需要考虑这一点，自研毕竟是有门槛和成本的。假如场景切换到公有云给租户用的环境，这样干就不合适了，你还是应该提供跟业界兼容的标准运维环境，在标准运维环境之上提供额外的保护。

3. 跨时间轴的场景

对于涉及跨时间维度的防护，典型的场景包括 shellshock 这样的漏洞公布时，各厂商在第一时间分析漏洞评估影响，这种影响是多层面的，不只是说可能拿到什么权限，还包括影响线上系统的哪些组件，这些组件的实时在线要求，修复漏洞会不会导致关联服务不可用。每一个漏洞修复的过程都是攻防双方和时间赛跑，攻击者尽量在厂商没有修复之前寻找利用点并发动渗透，而厂商尽可能在没出 POC 时赶紧把洞补了。在这个时间窗口中，甲方安全团队需要考虑的就是跨时间维度上的布防。出补丁之前是不是就干等不作为呢，显然不是的，细心的人会发现老牌安全公司的漏洞通告的解决方案里通常都会有一条，叫作“临时规避措施”，经验不足的团队是不会写上这条的。修不了漏洞时可以采用一些治标的补救性措施，比如对有漏洞的页面做访问控制，只允许有限的 src.ip 访问，比如在前端的 WAF/IPS 设备上加规则过滤对应的恶意请求，或者临时性的去掉一些权限，或者干脆关掉某些功能，但凡你想得到的通常总能找到临时规避方案，即便是有了补丁升级也不是立即完成的，在大型互联网生产网络里，全网打完一个补丁是需要不少时间的，有可能一个礼拜都弄不完，而且修复过程中要考虑服务可用性需要使用灰度和滚动升级的方法，比如修复前先把负载均衡上的流量切换到备用节点，然后对主节点的服务器打补丁，打完再把流量切回去，然后对备用节点的服务器打补丁……打完补丁后把临时防御措施再“回滚”掉（有价值的保留，核心设备上不建议留太多臃肿的规则），然后把特征加入全流量和主机 IDS。回顾整个时间轴的防护措施依次是：临时性规避措施—push 补丁 / 根治措施—取消临时性措施—添加常态性的特征检测措施—检测到漏网之鱼—继续上述过程，这个过程离最佳实践实际上还差了一个环节，不过这里只是用来说明开头提到的那个问题故而不展开了，后面会介绍对于一个漏洞修复是否需要上升层次的问题。

4. 风险和影响的平衡

假如你遇到一个安全问题是这样的：vlan 数目不够，vxlan 又不可用，提交问题后研发的反馈的方案 A 是如果要彻底修复则需要新增一个 dhcpd 的安全功能大约包含 10000（loc）即 1 万行代码，此时产品线又处于整体加班加点赶工大版本的状态，有人提出了方案 B 做 IP/MAC 双向绑定的缓解性措施，但这样的结果很可能是客户觉得太麻烦，而且配置一多容易出错，此时你想到了一个折中的办法 方案 C：给大客户单独 vlan，若干小客户共享一个 vlan，这样的好处是不需要太多成本，风险降低到可控，客户可以接受。如果选择方案 A 是不是更好？这个要看，假如这 1 万行代码只是用来临时的解决这个问题，显然 ROI 比

较低，但是如果后续的版本本来就要加入这个功能，不妨考虑一下。又如果后续版本不需要这个小众的功能，研发心里其实本不打算去开发的，说不定下次告诉你说要 2 万行代码，然后你到 CTO 那里也说不清风险到底多么严重，就会陷入僵局。**风险缓解的原则**是在以下三者之间做最大平衡：1）风险暴露程度；2）研发运维变更成本；3）用户体验的负面影响。

5. 修复成本的折中

一个安全漏洞的修复如果研发说要开发一周，另外一个方案是运维改一个服务器配置，而你其实心里知道在 WAF 上加条规则就能过滤，只不过你怕被绕过心里对这个措施不是特别有底气。对于这个场景我也不打算直接给出答案，但通常情况下改产品的成本是最高的，成本最高的往往不容易推动，推不动就无法落地，最后就是一堆安全问题。

Amazon 有一个研发理论，用一种 T-Shirt Size 估计的方式来做项目。产品经理会对每一条需求评估上业务影响力的尺寸，如：XXXL 影响一千万人以上或是可以占到上亿美金的市场，XXL 影响百万用户或是占了千万金级别以上的市场，后面还有 XL、L、M、S，这样逐级减小。开发团队也一样，要评估投入的人员时间成本，XXXL 表示要干 1 年，XXL 干半年，XL 干 3 个月，L 干两个月，M 干一个月，S 干两周以下。等等。

于是，可以这样推理：

- ❑ 当业务影响力是 XL，时间人员成本是 S，这是最高优先级。
- ❑ 当业务影响力是 M，时间人员成本是 M，这是低优先级。
- ❑ 当业务影响力是 S，时间人员成本是 XL，直接砍掉这个需求。因为是亏的。
- ❑ 当业务影响力是 XXL，时间人员成本是 XXL，需要简化需求，把需求简化成 XL，时间人员成本变成 M 以下。

安全其实也类似，风险和修复成本去比较，在坚守底线的基础上选择最优解。

综上所述，大家可以发现最优解往往不一定是安全的解，市场上乙方公司渗透测试报告中提的修复方案有些也是无法实施的，很多批判企业安全做得不好的帽子们，有机会真应该到企业里体验一下，企业安全岂是找洞补洞这么简单的事。

参考资料

陈皓的“加班与效率”(<http://coolshell.cn/articles/10217.html#more-10217>)。

3.6 需要自己发明安全机制吗

1. 安全机制的含义

首先解释一下发明安全机制这句话的意思。安全机制包括：常见的对称和非对称加密算法，操作系统自带的 RBAC 基于角色的访问控制，自带的防火墙 Netfilter，Android 的基于 appid 隔离的机制，kernel 支持的 DEP（数据段执行保护），以及各种 ASLR（地址空间随机映射），各种安全函数、服务器软件的安全选项，这些都属于已经存在的安全机制，注意我用的词是“已经存在”，而这个话题是针对是不是要在已有的安全机制上再去发明新的安全机制，比如三星手机的 KNOX，就是在 Android 基础上自己造了个轮子。

2. 企业安全建设中的需求

企业安全的日常工作是不是也会面临自己去发明安全机制的需求？会，但是不常见。实际上，在日常中发生的绝大多数问题都属于对现有安全机制的理解有误、没有启用或没有正确使用安全机制而导致的漏洞，而不是缺少安全机制，所以绝大多数场景都不需要去发明安全机制。发明安全机制是需要成本的，且需要有足够的自信，否则不健全的安全机制消耗了开发的人力又会引入新的安全问题，但此话不绝对。

3. 取舍点

那什么情况下应该发明安全机制呢，这其实非常考验判断者的技术实力。之前也提过对于很多安全漏洞的修复是否要上升层次的问题，首先要判断这是单个问题还是属于一类问题，如果是前者，用救火的方式堵上这个洞就好，没必要再去考虑更多。但假如这是一类问题，而你又没提出通杀这一类问题的手段就会永远处于救火之中，疲于奔命。如果是一类问题，分几种情况。第一种归入安全编程能力不足导致的安全问题，这类问题不需要通过导入新机制解决，而是通过加强 SDL 的某些环节，加强培训教育去解决。第二种情况则是属于在相应的领域还没有成熟的安全解决方案或者现有的安全机制对抗强度太弱，则可以考虑自己去造轮子。

比如有一个函数存在整形溢出，但只有在极特殊的情况下才能触发，平时开发过程中已经大量的使用了安全函数，启用了编译的安全选项，除了给这个函数加一个条件判断修复这个 bug 外是不是还要考虑更进一步的防护呢？大多数情况下显然是没必要的，假如这

是一个公共函数，那你可以选择把修复后的代码封装成安全的 API，避免其他程序员自己实现的时候发生同类问题。

换个问题，如果公司产品的某个私有协议总是被人频繁地解密和利用，而这种解密对产品的影响又较大，假设就是游戏客户端跟服务端通信的指令都能被破解和仿冒，那这种情况下就需要考虑是否更改或创建安全机制，即有没有必要通过实现更强的通信协议加密或提高客户端反调试的对抗等级来缓解这一问题。

如果你说新建安全机制也是补洞的话，其实也没错，就像 DEP 相对于用户态的程序而言是一种机制，而对于操作系统和冯·诺依曼体系结构而言是一个洞。当你过于勤奋地在很微观的细节上补洞却总是补不完的时候，不妨停下来看看能否在更高更抽象的层次上打个补丁。

安全工程师如果要晋升为 Leader 很重要的一点就是对安全事件和安全漏洞的抽象能力，没有抽象就谈不上 PDCA，就意味着更高的管理者对安全 KPI 在你手上能否改进不一定有信心。在纵深防御体系向中高阶段发展时，实际上会比较多的遇到是否要创新安全机制的问题，但是这个场景大多数公司未必会遇到。

3.7 如何看待 SDL

SDL（安全开发生命周期）优化模型如图 3-1 所示。

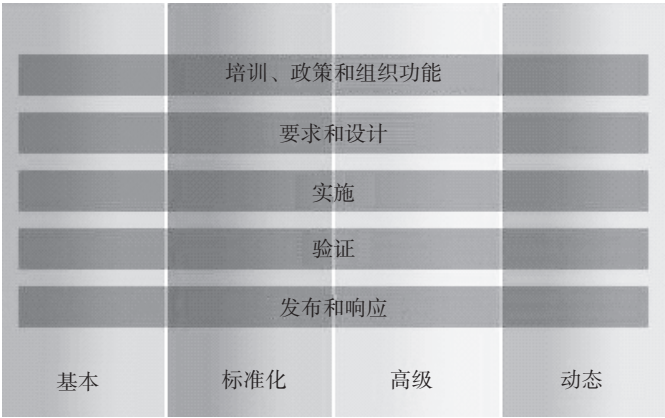


图 3-1 SDL 优化模型

SDL 起源于微软，2004 年将 SDL 引入其内部软件开发流程中，目的是减少其软件中的漏洞数量和降低其严重级别。SDL 侧重于长期维护、流程改进并能够帮助开发过程应对不断变化的威胁状况。早些年微软的产品安全问题比较多，微软在某一年甚至下令所有产品线开发计划停止半年，全部用于整顿安全问题。起初 SDL 适用于传统的瀑布模型和螺旋式开发，到了 2010 年 SDL 增加了敏捷的部分改进，用于应对互联网下的 Web 开发，目前 SDL 的“全貌”如图 3-2 所示。



图 3-2 SDL 整体框架

基本软件安全培训应涵盖的基础概念如下所示。

❑ 安全设计，包括以下主题：

- 减小攻击面
- 深度防御
- 最小权限原则
- 安全默认设置

❑ 威胁建模，包括以下主题：

- 威胁建模概述
- 威胁模型的设计意义
- 基于威胁模型的编码约束

❑ 安全编码，包括以下主题：

- 缓冲区溢出（对于使用 C 和 C++ 的应用程序）
- 整数算法错误（对于使用 C 和 C++ 的应用程序）
- 跨站点脚本（对于托管代码和 Web 应用程序）
- SQL 注入（对于托管代码和 Web 应用程序）
- 弱加密

❑ 安全测试，包括以下主题：

- 安全测试与功能测试之间的区别
- 风险评估
- 安全测试方法
- 隐私，包括以下主题：
 - 隐私敏感数据的类型
 - 隐私设计最佳实践
 - 风险评估
 - 隐私开发最佳实践
 - 隐私测试最佳实践
- 高级概念方面的培训，包括但不限于以下方面：
 - 高级安全设计和体系结构
 - 可信用户界面设计
 - 安全漏洞细节
 - 实施自定义威胁缓解

先看这份培训列表。能把这些彻底讲明白的人其实还是资深工程师以上的人。有人可能觉得我说的夸张了，原因在于大部分互联网公司的研发环境主要是 Web，有很多人能把 SQL 注入、XSS、CSRF 这些讲清楚，但问题是这样就算 SDL 了么？非也，当下热闹的安全大会各种讲攻防的议题，但这些离真正的产品安全设计还差很远，行业的普遍现状是能做入侵检测，能把漏洞修补原理说清楚，但很少有人能把安全架构设计非常体系化的讲清楚。很多人认为 SDL 在互联网公司无法完全落地的原因是因为 DevOps 模式下的频繁交付导致 SDL 显得过于“厚重”，我觉得这只能说对了一半，根据大多数互联网公司现行的模式，我加了一个帽子，姑且就叫“攻防驱动修改”吧。

3.7.1 攻防驱动修改

大多数甲方安全团队所做的工作实际上处于这个维度。通过对已知的攻击手段，例如 SQL 注入，XSS 等建立事前的安全编码标准，并在发布前做代码审计、渗透测试和提出漏洞修补方案。这种模式的显著优点是针对性比较强，直入主题，见效快。

简单的流程 + 事件驱动型构成了这种日常行为的本质，简单的流程通常包括：

- 事前基线：Web 安全编码标准，各公司内部范围流传的 APP 应用安全设计文档，这

个文档的质量水平通常可以差很远，当然文档永远只是文档，可能就是开发部门不强制不考试 800 年都不看的东西。

❑ 事中措施：代码审计，发布前过一轮扫描器 + 渗透测试。

❑ 事后机制：HTTP 全流量 IDS，Web 日志大数据分析，等等。

❑ 事件驱动：发现了新的安全问题就“事后诸葛亮一把”，做点补救性措施。

从整个过程看，攻防驱动修改比较偏“事后”，相对于完整的 SDL，威胁建模等工作而言，它似乎不用发散精力投入太多就能覆盖已知的攻击点，而且在研发侧不用面对比较大的“推动 SDL 落地的压力”

但是它的缺点也是显而易见的，由于过程方法论的施力点的比较偏事后，所以在从源头上发现和改进问题的能力不足，弱于在产品内建的安全机制上建立纵深防御，被绕过的可能性比较大，事后的 bug 率到一定程度就很难再改善，只能通过不断的攻防对抗升级去事后修补。笼统一点讲就是考虑不够系统性。这跟当前安全行业缺少真正的安全架构设计人才有关，攻防的声音铺天盖地，跟国外比一下在设计和工程化方面差距不小。

事后修补是不是总是有效的，缝缝补补的感觉你觉得会如何呢，对于公司的边缘性产品你可以希望它早日归入历史的尘埃，而对于公司的支柱型产品你只能寄希望于某一个大版本更新时把某些机制推倒重新来过。

3.7.2 SDL 落地率低的原因

1. DevOps 的交付模式

互联网频繁的迭代和发布，不同于传统的软件开发过程。如果一个软件要一年交付，那么在前期抽出 2 ~ 4 周做安全设计也可以接受，但在互联网交付节奏下，可能一周到一个月就要发布版本，你可能没有足够的时间去思考安全这件事。对于 SDL 会拖慢整个发布节奏这个问题上，安全团队去推动也会直面公司管理层和研发线的挑战。不过当你有经验丰富的安全人员和自动化工具支持时，SDL 在时间上是可以大大缩短的。

2. 历史问题

99% 的甲方安全团队的工作都是以救火方式开始，SDL 从来都不是安全建设第一个会想到的事情，而且业内心照不宣的一个原因是，“事前用不上力，偏事后风格的安全建设”

贯穿于大多数安全团队的主线。之所以如此，原因是第一有火必救，有的团队救火上瘾，有的则能抽身转向系统性建设；第二想在事前用力，需要自己足够强大，能摆平研发，不够强大就会变成庸人自扰，自讨没趣，还不如回避。

3. 业务模式

大多数平台级互联网公司的开发以 Web 为主，超大型互联网公司才会进入底层架构造轮子的阶段，而对于以 Web 产品为主的安全建设，第一是事后修补的成本比较低，屡试不爽；第二是部分产品的生命周期不长，这两点一定程度上会让很多后加入安全行业的新同学认为“救火”=“安全建设”。但是在甲方待久了的人一定会发现，哪怕是 Web，只要系统比较大，层层嵌套和不同子系统间的接口调用，会使得某些安全问题的修补成为疑难病症，可能就是设计之初没有考虑安全，致使问题不能得到根治。时间一久，技术债越积越多，大家最后一致默认这个问题没法解决。

4. SDL 的门槛

其实 SDL 是有门槛的，而且还不低。最重要有两点：第一点是安全专家少，很多安全工作者懂攻防但未必懂开发，懂漏洞但未必懂设计，所以现实往往是很多安全团队能指导研发部门修复漏洞，但可能没意识到其实缺少指导安全设计的积累，因为安全设计是一件比漏洞修复门槛更高的事。看业内很多技术不错的安全研究者，写的文章，往往前半篇漏洞分析很给力，但到了安全建议环节好像就觉得少了点什么。第二点是工具支持少，静态代码扫描、动态 Fuzz 等，工欲善其事必先利其器，Facebook 宣称其最好的程序员是投入到工具开发的，而对于国内很多安全团队而言，最好的人都不会用在工具开发上，而是奋斗在攻防第一线做救火队长。稍微好一点的情况是这帮人投入在做安全机制建设上，而业务部门也不会来帮助安全团队开发安全工具。这还跟公司整体上是否重视自动化测试有关，如果公司在测试领域的实践没有做到很前沿，那么安全的黑白盒测试也不会注重工具化建设，代码覆盖率和路径深度等更加不会有人去关注了。实际上不一定要在这个场景下自己去造轮子，用商业工具是不错的选择。

3.7.3 因地制宜的 SDL 实践

1. 重度的场景

对于公司内研发的偏底层的大型软件，迭代周期较长，对架构设计要求比较全面，后

期改动成本大，如果安全团队人手够的话，这种场景应该尽量在事前切入，在立项设计阶段就应该进行安全设计和威胁建模等工作。相比在事后贴狗皮膏药，这种事前的时间投入是值得的，门槛主要还是人。

对于较大软件的“大版本”，包括每个产品初始版本，还比如标杆产品的 1.0 到 2.0 类似这种里程碑式的版本发布，修改和增加了很多功能点，甚至修改了底层的通信协议，这种也需要较完整的 SDL，当然这种版本跳跃有时候只是对外的一种营销手段，不一定是技术上的大修改，这个就要看实际情况了。

2. 轻度的场景

对于架构简单、开发周期短、交付时间要求比较紧的情况，显然完整的 SDL 就太重度了，这个时候，攻防驱动修改就足以解决问题。

其他的诸如小版本发布，技术上没有大的修改，也没必要去跑全量 SDL，否则就太教条和僵化了。

3.7.4 SDL 在互联网企业的发展

目前 SDL 在大部分不太差钱的互联网企业属于形式上都有，但落地的部分会比较粗糙。通常只有一两个环节。最主要的瓶颈还是人和工具的缺失。以前互联网企业只生产 Web，攻防驱动修改得以应付，但是现在大型的互联网企业不再只生产 Web，而是会自己生产诸如分布式数据库、浏览器、手机操作系统这样的大型软件，单纯的攻防驱动修改已经日渐乏力，没有足够的安全设计能力将无法应对未来的威胁。因此推测以后的安全行业中，设计方面的人才严重缺失，大部分甲方安全团队仍然游离在设计的大门之外，只有一些大型厂商正在借研究之名来做一些改进安全设计的工作，期待这些大型厂商们能带一带团队，给这个行业培养一些生力军。

SDL 除了最早基于传统瀑布模型版本，以及为 DevOps 优化的版本，实际上在实践阶段还可以优化成极速发布的版本，或者干脆不追求 SDL 而从其他的维度来弥补 SDL 不健全的问题，其实现的本质是原来的 SDL 对研发流程的修改有点像“阻塞式 IO 模型”，而现在可以通过工具和技术手段使其变成“异步 IO 模型”，从更高维度补贴 SDL 的思路在这里不再展开，后续会在笔者博客上专题分享。

参考资料

微软 SDL 白皮书：<https://www.microsoft.com/zh-cn/download/details.aspx?id=12379>

3.8 STRIDE 威胁建模

STRIDE 是微软开发的用于威胁建模的工具，或者是说一套方法论吧，它把外部威胁分成 6 个维度来考察系统设计时存在的风险点，这 6 个维度首字母的缩写就是 STRIDE，分别为：Spoofing（假冒）、Tampering（篡改）、Repudiation（否认）、Information Disclosure（信息泄露）、Denial of Service（拒绝服务）和 Elevation of Privilege（权限提升），如表 3-1 所示。

表 3-1 STRIDE 威胁建模

属性	威胁	定义	例子
认证	Spoofing（假冒）	冒充某人或某物	假冒 billg、microsoft.com 或 ntdll.dll
完整性	Tampering（篡改）	修改数据和代码	修改一个 DLL，或一个局域网的封包
不可抵赖性	Repudiation（否认）	宣称未做过某个行为	“我没有发送 email”“我没有修改文件”“我肯定没有访问那个网站”
机密性	Information Disclosure（信息泄露）	暴露信息给未经授权的访问者	允许某人阅读 Windows 源代码；将客户列表发布在网站上
可用性	Denial of Service（拒绝服务）	使对服务对用户拒绝访问或降级	发送数据包使目标系统 CPU 满负荷或发送恶意代码使目标服务崩溃
授权	Elevation of Privilege（权限提升）	未经授权获取权限	远程用户执行任意代码，普通用户可以执行管理员私有的系统指令

STRIDE 如何使用？先画出数据流关系图（DFD）用图形方式表示系统，DFD 使用一组标准符号，其中包含四个元素：数据流、数据存储、进程和交互方，对于威胁建模，另外增加了一个元素，即信任边界。数据流表示通过网络连接、命名管道、消息队列、RPC 通道等移动的数据。数据存储表示文本、文件、关系型数据库、非结构化数据等。进程指的是计算机运行的计算或程序。然后对每一个节点元素和过程进行分析判断是否存在上述 6 种威胁，并制定对应的风险缓解措施。例如图 3-3 所示的情况。

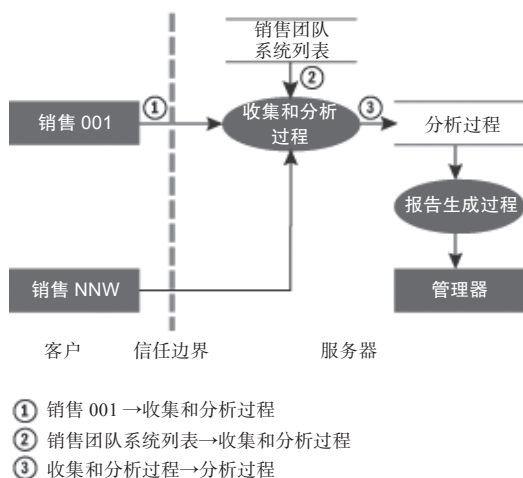


图 3-3 STRIDE 示例

图中，①、②、③其实都存在假冒、篡改、拒绝服务的风险，所以在这些环节都需要考虑认证、鉴权、加密、输入验证等安全措施。但根据风险的不同，过程③在内网的服务器被拒绝服务的风险较小，而在 Internet 上传输的过程①所受到的监听和篡改的风险更大，所以在每个环节上采取的风险削减措施的力度会不一样，这实际上也是什么安全措施一定要落地、什么安全措施可以适当妥协的一个参考视角。很多安全从业者所接受的安全认知往往是进入一家企业后，拿到一份名为应用开发安全标准的文档，里面描述了访问控制、输入验证、编码过滤、认证鉴权、加密、日志等各种要求，久而久之就变成了一种惯性思维，实际上之所以要这么做是因为在系统设计的某个环节存在 STRIDE 中的一种或几种风险，所以在那个设计关注点上要加入对应的安全措施，并不是在所有的地方都要套用全部的或千篇一律的安全措施。否则就会变成另外一种结果：“过度的安全设计”。威胁建模的成果跟工作者自身的知识也有很大的关系，有攻防经验的人比较容易判断威胁的来源和利用场景，如果缺少这方面的认知，可能会发现到处是风险，有些风险的利用场景很少或利用条件非常苛刻，如果一味地强调风险削减措施也会变成有点纸上谈兵的味道，虽然从安全的角度没有错，但从产品交付的整体视角看，安全还是做过头了。

总体上看，STRIDE 是一个不错的参考视角，即便有丰富攻防经验的人也不能保证自己在面对复杂系统的安全设计时考虑是全面的，而 STRIDE 则有助于风险识别的覆盖面。

以上的例子是 high level 的威胁建模，low level 的威胁建模需要画了时序图后根据具体的协议和数据交互进行更进一步的分析，细节可以参考威胁建模相关的方法论，但不管是

high level 还是 low level 都比较依赖于分析者自身的攻防技能。

参考资料

威胁建模：使用 STRIDE 方法发现安全设计缺陷（<http://msdn.microsoft.com/zh-cn/magazine/cc163519.aspx>）。

STRIDE 图表（<http://blogs.microsoft.com/cybertrust/2007/09/11/stride-chart/>）。

3.9 关于 ISO27001

1. 重建对安全标准的认知

虽然标题用了 ISO27001，但实际上这里可以指代所有的安全标准和安全理论。木桶理论安全界的人都知道，但用到实际工作中，没太大用，说到底就是给外行解释安全这件事的一个通俗比喻而已。业内有些声音认为安全标准堵不住漏洞，所以安全标准都是没用的“废物”，这种论据显然是有问题的，首先安全标准的制定就不是为了堵漏洞，所以安全标准跟漏洞没关系，完全两个层面的东西，不能拿来说事，堵漏洞有具体的技术手段，但安全建设并不只有堵漏洞这种微观对抗。

那安全标准到底有什么用，我用最通俗的语言解释一遍，安全标准归根结底是为了给你一个参考和指引，当你把基础的技术防护手段实施之后，过了上任之初的救火阶段之后，就需要停下来思考一下整个企业安全范畴中，哪些事情是短板，哪些领域尚且空白，需要在哪些点上继续深挖才能覆盖公司整体的安全建设，而安全标准的价值就是告诉你，在安全建设的领域里可能有那么 100 件事情是需要做的，但具体选择只做 80 件还是 99 件还是 100 件全做是你自己的事情，它只告诉你 100 件事情是什么，但是这 100 件事情怎么实现，对应的技术方案或流程是什么它不会告诉你，实现和落地是需要自己去想的，它本质上是用于开拓视野，跟堵不堵漏洞完全没冲突，换句话说它是一本书的目录，但对于每个章节怎么写则取决于你自己，你可以买 WAF 也可以加固容器，也可以像偏执狂一样地做代码审计，至于堵漏洞那只是每个章节里的一段文字而已。

2. 最实用的参考

对互联网公司而言，我认为有几个非常刚需的参考：

- ITIL(BS15000/ISO20000)——绝大多数互联网公司的运维流程都是以 ITIL 为骨架建立的，甚至连内部的运维管理平台，监控系统上都能一眼看出 ITIL 的特征。而偏运维侧的安全，基础架构与网络安全，这部分的安全建设是以运维活动为主干，在运维活动上添加安全环节来实现安全管理的。所以想在运维侧建立安全流程必须熟悉 ITIL，把安全环节衔接到所有的发布、变更、配置、问题和事件管理之上，而不是打破原来既有的运维流程，再去独创一个什么安全流程。
- SDL——研发侧的安全管理，绝大多数公司都借鉴了微软的 SDL，即便是再有想法的甲方安全团队也离不开它，所以无论如何必须掌握 SDL。
- ISO27001——企业安全管理领域的基础性安全标准，所谓基础就是不能比这个更加精简了，你可以不碰那些高大上的，但是 ISO 27001 则相当于入门水准，就好像高等数学线性代数你可以不会，但是如果你连 9×9 乘法表都背不出来，那只能永远呆在家里不出门了，因为你连买 10 个苹果找你多少钱都算不来。ISO 27001 总体上提供了一个框架性的认知。

3. 广泛的兼容性

学习攻防技术和学习少数几个国际标准一点都不冲突，南向北向都是人为划分的，除非坐地画圈，否则完全不存在这种天然障碍。一个优秀的甲方工程师就是应该系统化又熟悉技术细节的，对于开篇提到的 CSO 而言，没有视野的人绝对当不了 CSO。

4. 局限性

方法论的作用是解决企业整体安全从 30 分走向 50 分的问题，这个阶段需要具备普适性的有助于改善基本面全方位提升的东西。但是到了中后期，这些就不太管用了，如果你想从 60 分上升到 80 分，不能再依赖于方法论，而是进入安全特性改进的贴身肉搏战状态，很多竞争力也许只有几十条规则，但是这些从表面上是看不出来的，只有依靠专业人士的技能和资源的集中投入才能有所产出。

3.10 流程与“反流程”

1. 人的问题

在传统安全领域一直是强调流程的，但是互联网行业有一点反流程，甚至像 Facebook

这样的公司还表示除非万不得已否则不会新建一条流程来解决问题。那安全建设到底要不要流程。首先有流程肯定能解决问题，但流程化是不是最佳实践则不确定。

于是先解决第一个问题：有没有可能没有流程，什么情况下可能很少或接近于没有流程。假如公司的人很少，从工位上站起来就能看到全公司的人，要发布版本吼一嗓子全员都能听到，这种情况下确实不需要什么流程，不只是安全流程，其他的流程也没必要。

如果组织比较大，单纯一个发布行为会涉及很多跨部门的人，甚至地域上都是分布式的团队，参与活动的人员行为即便是都挂在公司内部的 IM 工具上一致性也无法保证，那这个时候为了尽可能规避人犯错，就会制定一些流程。随着组织越来越大，流程会越来越多，并且流程大都不是“进取和开放”型的，而是“错误规避”型的，整个组织的流程都会表现为在某个方向上高度优化，从而进入基因决定理论的影响范畴，流程的制作者为了保护既有权益，大多会僵化执行流程，开始走向自己的反面。作为安全负责人，必须周期性地审视安全和 IT 治理的流程是不是太冗余了，是否可以精简一下，或者在公司业务扩张、新建业务线的时候考虑一下原有的流程是否适用于新的领域。如果安全负责人对这些问题比较漠视，要么对业务不敏感，要么自身提前进入不受激励的保乌纱帽状态了。

2. 机器的问题

流程是用来解决人容易犯错的问题，而不是用来解决机器犯错的问题，如果把流程用于解决机器犯错的问题那就会闹出笑话来。比如程序发布前，需要有一个安全检查的环节，如果不约定流程，很容易漏掉，这是在解决人步骤错误的问题。但是 10000 台机器打同一个补丁，有 9990 台都打上了没问题，剩下 10 台补丁有问题，这种问题应该通过技术手段解决，而不是通过流程来解决，如何让系统和程序返回人所期望的结果是技术需要解决的问题，跟流程没关系，当然有人说跟程序的执行流程（routine）有关系，是的此流程非彼流程。通过人为的流程来解决，人的工作会越来越多，忙于救火之中不堪重负，通过技术途径解决，组织的自动化程度会越来越高，生产力会越来越强，两条路最终会使团队走向两极分化，选哪一极就看团队的基因了。在前面的例子中为了衡量流程的执行结果，我们通常会引入一些技术的自动化手段来检测，例如被动式扫描，有些开发和运维漏掉安全审计环节直接上线了，也能把这些程序的 URL 找到抓下来扫，当然它并不能替代流程本身的作用。

3.11 业务持续性管理

业务持续性管理（BCM）是一个较高层次的管理机制，通常对应到公司层面，它使企业认识到潜在的危机和相关影响，制订响应、业务和连续性的恢复计划，其总体目标在于提高企业的风险防范能力，有效地响应非计划的业务破坏并降低不良影响。以 2015 年中国互联网的标志性事件为例，网易、携程、支付宝先后发生大规模服务中断，这个问题就是业务持续性管理的场景。

BCM 的全生命周期方法论如图 3-4 所示，其中 BIA（Business Impact Analysis，业务影响分析）、Recovery Strategy（恢复策略），实施以及测试和演练都是很重要的环节。BIA 的例子：如果 QQ 的数据库被拖库了对腾讯有什么影响，如果支付宝被拖库了对阿里有什么影响？这两个问题可能有点极端，再举几个可能性更大一点的：如果公司官网遭受 DDoS 攻击，流量一度超过防御的最大值会有什么影响？如果公司一个主站页面被挂了会有什么影响？好像这些问题都比较头疼，再举一个轻微一点的，如果一个客服的论坛账号被盗了会有什么影响，尽管这个问题在入侵者那里可能会玩成蝴蝶效应，甚至变成 APT，但是一般情况下这个还是比之前的例子要轻微得多，不能 getshell 的话，改一下口令就完事了。从这里也可以看出 BIA 跟基于资产权重的风险管理方法论类似，跟威胁建模也有点异曲同工的味道，只不过威胁建模关注的是具体的系统，而 BIA 关注的是公司的业务。



图 3-4 BCM 的生命周期

对于风控策略，大多数人会想到 DB 审计，纵深防御 OS 防护，应用层防止 SQL 注入，部署 WAF 等，这些东西原先放在安全体系里“貌似”完整，而如今放到 BCM 里一下子就

不完整了，为什么？显然你仍然没有回答上述问题：假如被拖库了怎么办。有的人认为管理是无用的，BCM 也无助于降低程序的安全漏洞率，但是只会攻防，只会堵洞显然也解决不了企业安全中的那许许多多问题。这个问题很现实，作为安全负责人，老板一定会问你假如被拖库了怎么办，有人会说“我引咎辞职”，这样的回答似乎很有责任感，但对面的人可能就会想：“你是骗子么，公司都快垮了，你走人了？”

一个典型的 BCP（Business Continuity Plan）如表 3-2 所示。

表 3-2 典型的 BCP 示例

业务持续性计划	灾难恢复	业务恢复	业务继续	持续性规划
目标	关键电脑、应用	关键业务流程	流程还原	流程变通方案
聚焦	数据恢复	流程恢复	返回正常状态	将就使用
示例事件	大型机故障	实验室泛	建筑物火灾	应用丢失
解决方案	热备份站点恢复	烘干和重启	新的装备和建筑物	使用手工流程

这里面定义了一系列的视角维度和与之对应的措施，不具体展开了，对于安全团队而言，通常需要根据公司所有的业务简单做 BIA，然后根据所有的 IT 资产权重分类分级，并制定对应的保护策略以及关键安全事件发生后的应急预案。应急预案不只是纸面上的流程，还包括了一系列的技术性措施，例如你的账号保护体系。现实生活中有的人比较反感理论派的原因是只制定纸面的策略，而忽略技术环节的 PDCA，不关注如何优化纵深防御来缓解保护失效后持续缩小攻击面和影响面的问题，不关注阻断 kill chain，不重视安全事件发生后的应急体系工具建设，以及关键系统的功能中是否支持有损服务策略等，这些问题导致了技术派认为管理是无用的，甚至连安全标准都背了黑锅，其实标准是无罪的，BCM 是好东西。

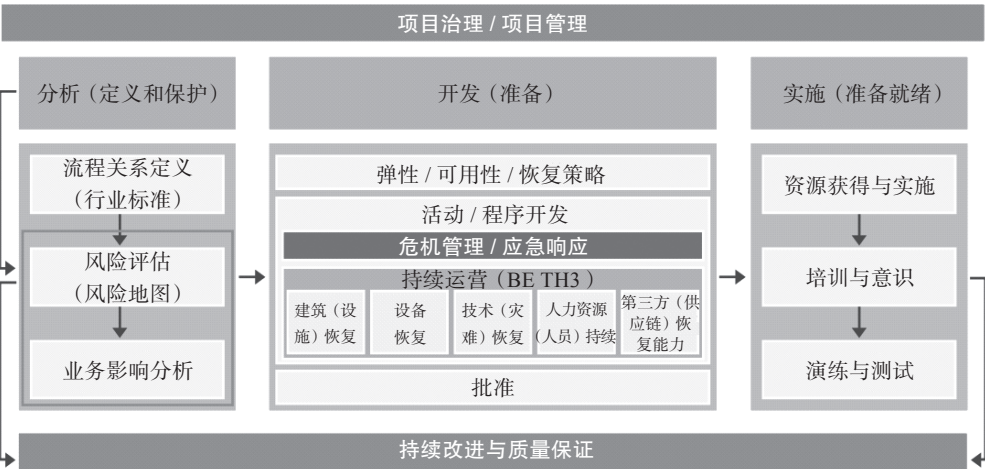


图 3-5 德勤的 BCM 方法论实施路径

图 3-5 是德勤的 BCM 方法论实施路径图，更多关于 BCM 的内容可以上网自己搜索，这里不继续展开了。

参考资料

“德勤业务连续性计划和管理”(<http://www.davislogic.com/bcm.htm>)。

3.12 关于应急响应

很多人认为应急响应就是 SSH 连上被黑的机器去查 rootkit，直到今天我也基本不漏读那些思路清晰的入侵分析的 paper，只不过工程师关注的跟 CSO 关注的还是有些不一样。10 年前，我是乙方工程师时去客户机器上查后门，虽然没有犯过明显的大错误，但有时候还是很怕把系统搞垮了，毕竟人家也没备份数据，所以总归心里不踏实。如果你在 SRC 接到白帽子报漏洞，打开一看人已经入侵到系统了，然后就突然心里一沉，慌慌张张就要了个 root 账号 SSH 连上去了，这种状态其实很不好，搞不好反添乱，一个不小心就发生了点小意外把什么东西搞挂了，然后莫名其妙自己就变成罪魁祸首了。

应急响应流程

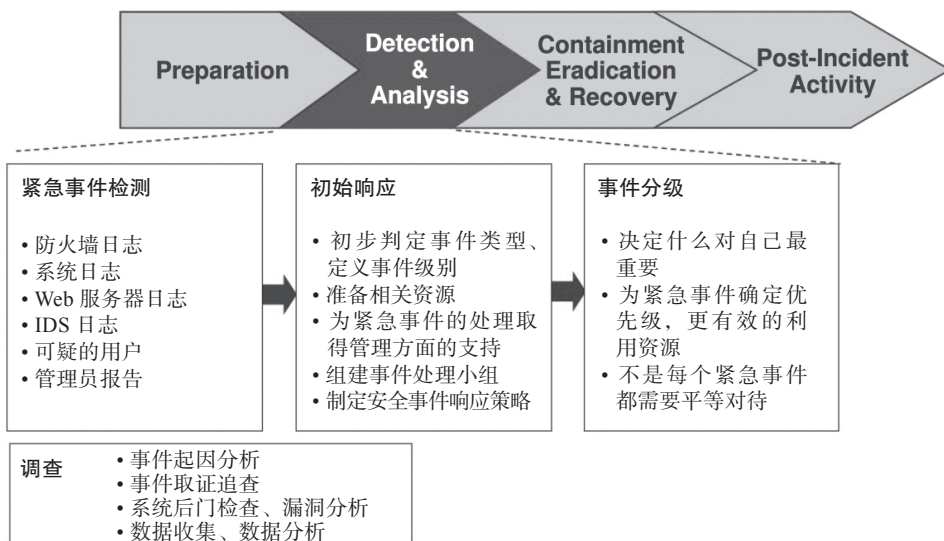


图 3-6 应急响应的 PDCERF 模型

应急响应有一个 PDCERF 模型（也可以参考 NIST SP800-61），如图 3-6 所示。简单说来就是一连串步骤。P 是指 Preparation（准备），之前要做各种准备工具，具体一点的比如应急工具：静态编译的 ls、ifconfig、ps 等总归是要事前准备好。D 是指 Detection（诊断），初步诊断发生了什么类型的问题，有助于后续工作展开，同样是大规模流量，L4 DDoS，CC 还是蠕虫爆发，对应的应急手段不一样。C 是指 Containment（抑制），这是被大多数人忽略的一步，首先应该是抑制受害范围，隔离使受害面不继续扩大，再追求根治，而不是一边任其蔓延，一边去查后门，到头来你查完这台机器，入侵者在这时间档里又搞到了另外两台，然后你就干瞪眼吧。这跟 ITIL 的思路是一样的，问题发生时首先是用事件管理以平息事件，恢复 SLA 为目标，至于到底是什么原因可以先放一放，等恢复服务了，再用问题管理来解决根因的事情。接下来就是大多数人最熟悉的那一步，E 是指 Eradication（根除），寻找根因，封堵攻击源。R 是指 Recovery（恢复），把业务恢复至正常水平。最后，F 是指 follow-up（跟踪），监控有无异常，报告，管理环节的自省和改进措施，现在俗称安全运营的持续改进环节。

ITSM 的思路贯穿于企业安全建设的方方面面，了解攻防技术只是说你具备了参与企业安全建设的技术理论基础，但并不代表你具备了企业安全建设的正确方法。之前说到抑制的环节，首先要了解业务、数据流、各服务接口调用关系，这些都是日常的积累，否则随便一个隔离又把什么服务搞挂了。反过来倒推，如果安全团队平时连个数据流图都没有的，发现单点出现问题症状时大致的系统间影响和潜在的最大受害范围都估计不出来的，那安全建设即便是救火也肯定是一团糟啦。

3.13 安全建设的“马斯洛需求”层次

可按照“马斯洛需求”层次来描述安全建设的需求，如图 3-7 所示。

其实这张图里少了一个级别，就是 LV0，即没有安全措施。

LV1 通过一些较为基础的安全措施做到了基础的访问控制，并且交付的系统不含有明显的高危漏洞。但对于复杂的安全事件自身没有独立处理的能力，必须依赖于外部厂商。

LV2 有专职的安全团队，有攻防技术能力，能做到有火必救，不依赖于外部厂商。但在安全建设上缺乏思路，大多依赖商业产品或照搬已有的安全模式。

LV3 安全建设呈现初步的系统化，覆盖全生命周期，开发和运维环节都有必要的控制流程，主要的系统在架构上都会考虑安全方案，检测和防护手段能因地制宜。

LV4 除了基础架构，应用，数据等技术层面安全能做到全生命周期系统化建设，业务层面的安全问题也有系统化解决方案。安全此时不只关注技术层面的攻防对抗，也关注业务形式的安全以及黑产的对抗。

LV5 安全建设进入最佳实践阶段，不依赖于现有的安全机制，也不依赖于厂商的安全产品，虽说自己造轮子不代表最佳实践，不过对于互联网公司攻防和业务层面的安全问题，如果想做的好一点，不自己发明轮子似乎不太可能，至少现在市场上缺少很多针对互联网的解决方案。在这个阶段，严重的安全事件几乎很少发生，大多数精力都会用于优化现有系统的检测和拦截率。



图 3-7 安全建设的“马斯洛需求”层次

3.14 TCO 和 ROI

企业安全建设本质上不是一个可以通过完全量化的指标来衡量建设得好与坏以及能力成熟度的事情。安全是一个永无止境的投入，永远都没有办法提出一个清单，说这里的条目你都做到了，安全就很好了。也是为什么等级保护、ISO27001、PCI-DSS 这类企业认证可以用来装点“门面”，在广告、营销、融资、上市公司内控等环节告诉公众自己是有那么一点安全能力的，但是永远都不可能说我是无懈可击。某些老板在台上走秀时吹牛的情节请自动忽略。安全建设的好坏不只是依赖于安全团队的强弱，还跟安全建设本身所消耗的金钱和资源有关，大多数企业都不是土豪级的公司，拿业界最佳实践来衡量都缺少前提条件，业界最佳实践可以是安全团队自己的追求，但不是安全团队工作成果的评价标准，因为现实中不可能消耗那么多钱用于安全建设，而一定是根据企业所处的阶段投入到应景的程度适可而止，如果这个时候某些砖家一定要拿什么能力成熟度模型来评估，说结果不好看，你可以很理直气壮的告诉对方从现实的角度出发无可厚非，什么阶段就是应该做什么事，拿个互联网行业千亿美金市值的公司来“套”大多数公司纯属无脑行为。

可用于评价安全建设做的好与坏的唯一标准就是 ROI，用什么样资源（TCO）产出什么样的安全效果。能力平庸的 CSO 可能建了很大的安全团队也还是频出安全事件，团队中不乏张口就是“七分管理，三分技术”的那类人，而面向实操干活的人却是少数且没有地位的工程师，即便有学习能力不错的工程师也没法系统化的组织他们的工作，任其自生自灭。还有的公司不缺干将，但是招了牛人却只做救火之用，到头来的产出和一支没有牛人的团队相差无几。思路清晰的 CSO 即便自己不充当工程师的角色，也能以一支精锐小团队覆盖企业中绝大多数安全环节。

在互联网公司，安全负责人最可能影响 ROI 的几个因素如下：

- ❑ 缺少系统化蓝图，对安全建设的全局以及分解后的轻重没有感性认知，这是最可能导致头痛医头脚痛医脚的原因。
- ❑ 对管理体系和工具链建设没有整体认知，选择在错误的时间点做正确的事。
- ❑ 把安全工作当成 checklist 而不是风险管理工作，凡事要求绝对安全而不能接受相对风险，对下属求全责备，大概只有外星人能满足这种需求。
- ❑ 弱于判断安全建设在实践环节的好与坏，搞不清楚某个安全机制在业界属于落后水平，平均水平，还是领先水平，以及某种安全机制对于削减某类风险的作用强弱，

容易被执行者忽悠。带来的结果就是貌似很苦很努力，但收效甚微。

❑ 个人的职场步调与公司发展的阶段不一致，公司处于快速扩张时期，而安全负责人本人则采取保守谨慎的策略。

❑ 只务内勤，不管外联，不重视生态关系建设。最后导致小问题，大影响。

作为一个风险管理型的角色，而不是直接产生利润的职能，其管理向上沟通可能会有一定的难度，在风险取舍方面跟上下左右达成一致也需要提前沟通好。

