

▣ 知道创字研发技能表v2.2



2014/3/9 发布

by @知道创字 (www.knownsec.com) @余弦 & 行之

知道创字是国内Geek十足且普遍被认为特别有前途的互联网安全公司，

动态请关注微信公众号：[Lazy-Thought](#)。

▣ 说明

- 本技能表为知道创字研发工程师必备技能
- 聪明的人会根据每个tip自驱动扩展
- 不聪明的，坐等别人手把手的人请绕行，不仅不适合知道创字，也不适合任何有Geek精神的公司
-  附件标志是我们推荐的附加资源，感谢资源提供者
- [知道创字研发技能表v2.2离线版打包下载](#) 

▣ 通用技能

▣ 公司与个人

- 公司是盈利性组织
- 个人和公司必须双赢

▣ 工作

- 在认同公司理念且能够给公司创造足够价值的基础上，为个人发展而工作

▣ 沟通、反馈、责任

- 一个无沟通能力的人，要么是天才，要么是不可爱的人

▣ 反馈要及时

- 避免出问题不反馈，影响进度

▣ 方式

- 正式的：邮件
- 临时的：QQ等即时通信
- 礼貌的：给个电话、短信

- 工作有大小，责任心无大小

▣ 周报的透明

- 意义：大家互相了解工作与心得，有利于自己的判断与成长




- 不是单纯的给领导汇报工作
- 任务四象限，决定优先级
 - 紧急重要
 - 重要不紧急
 - 紧急不重要
 - 不紧急不重要
- 新事物的敏感性
 - 保持好奇心
 - 不要局限在自己的圈子，适当跨界吸收灵感
- 订阅国内外优秀博客/资源，鲜果RSS阅读不错
 - 英文很重要
- 选择性参与一些必要的会议，听必要的主题，讨论必要的话题
- 成长
 - 对知识的渴望程度决定了前进动力的大小
 - ⚠ BT常说：「做人一定要狂热！」
 - ⚠ 我说：「一定得贪婪！」
 - 不要矫情，不要浮夸！
 - 和比你厉害的人在一起，和一流的人工作
 - 指点往往是精华
 - ⚠ 杜绝笨蛋爆炸
 - 二流的人招进来的人不太可能是一流的
- 思考
 - 批判性思考
 - 换位思考
- 💡 提问的智慧
 - 遇到问题先独立思考，尽最大努力后再提问
 - <http://www.wapm.cn/smart-questions/smart-questions-zh.html>
- 💡 小事心态
 - 越基础的事越关键，需要越细心
 - 不要一味盲目追求「高级感」，而忽视「小事」/「简单事」/「基础事」

- 基础不牢、地动山摇
- 小事做不好，别提大事

☐ 💡 任务拆分

- 成长过程会经历：能力越大、责任越大、事情越多

☐ 思路

- 拆分细化为多个点
- 排好优先级 
- 加入时间维度：何时完成什么 
- 是否需要寻求帮助，谁能帮你，自己单干？ 
- 任务是否可以切换/并发
- 自己欠缺什么，立马发现

☐ 💡 方法论

- 完成一件事有好几条途径，优秀的人的途径最短
- 任务拆分很容易得出做事的方法论

☐ 好的「方法论」会让你具备更强的「创造力」！

- 💡 时刻问自己：「是否具备创造力？」

☐ 牛人姿态

- 即使现在不是牛人，也得具备这样的姿态
- 这种感觉只能意会
- 没有一定扎实内功与远见的人很少有这样的姿态

☐ ⚠️ 完成的定义

☐ 比如写个POC

- 1. 搞懂了目标Web应用漏洞的原理
- 2. 熟练运用Python各相关模块与机制

☐ 3. 熟练了解了HTTP协议

- HTTP请求
- HTTP响应

- 4. 代码写得够规范，让人看起来就是爽

☐ 5. 程序经过足够的测试

- 黑测试
- 白测试

- 6. 及时反馈进度
 - 我遇到困难了
 - 我搞定了
- 7. 更新相关文档，沉淀
- ⚠️ 熟练的定义
 - 比如熟练SQL注入
 - SQL语句这门“语言”能脱离文档顺手写出
 - 主流数据库的SQL特有函数、存储过程、机制我都了如指掌
 - MySQL
 - MSSQL
 - Oracel
 - Postgre
 - Access
 - SQLite
 - ...
 - 牛逼的工具我不仅用的顺其自然，源码还读过几遍，我能修改
 - sqlmap
 - ...
 - 我具备创造性，而不仅仅是跟在大牛身后
 - 研究出了几个不错的技巧
 - 发了几篇不错的Paper
 - 对外会议/沙龙等进行了几次分享
 - 写出了自己的相关工具，爽
 - 我实战了N回，遇到了很多奇葩环境，我有足够的信心绕过
 - 以上这些之后，这才叫熟练！其他同理
- 好书推荐
 - 推荐理由
 - 打通任督二脉的书，怎能不看？
 - 任何科学研究最终必须至少到哲学层面，触碰到上帝的脚
 - 具体技术类书籍请见“专业技能”相关部分
 - 鸡汤类

▣ 黑客与画家

▣ 印象深刻：设计者的品味

▣ 好设计是简单的设计

- 抓住本质

▣ 好设计是永不过时的设计

- 如果解决方法是丑陋的，那就肯定还有更好的解决方法，只是还没有发现而已

- 好设计是解决主要问题的设计
- 好设计是启发性的设计
- 好设计通常是有点趣味性的设计
- 好设计是艰苦的设计
- 好设计是看似容易的设计
- 好设计是对称的设计
- 好设计是模仿大自然的设计
- 好设计是一种再设计
- 好设计是能够复制的设计
- 好设计往往是奇特的设计
- 好设计是成批出现的
- 好设计常常是大胆的设计

▣ 浪潮之巅

- 感受IT帝国的崛起与没落，我们现在站在又一个互联网浪潮之巅

▣ 洁癖类

- 重构
- 代码整洁之道
- 代码大全2
- 数学之美

▣ 敏捷类

▣ Rework中文版

- 37signals团队的敏捷经验
- 高效程序员的45个习惯

▣ 产品类

- 人人都是产品经理

- 结网

☐ 神书

- 自私的基因
- 失控
- 万物由来

- ...

☐ 专业技能

☐ 💡 原则

- 至少完整看完与练习好一本书
- 至少过一遍官方文档

☐ ★ 基础必备

☐ HTTP抓包与调试

☐ Firefox插件

☐ Firebug

- 抓包与各种调试

☐ Tamper Data

- 拦截修改

☐ Live Http Header

- 重放功能

☐ Hackbar

- 编码解码/POST提交

☐ Modify Headers

- 修改头部

☐ Fiddler

- 浏览器代理神器
- 拦截请求或响应
- 抓包
- 重放
- 模拟请求
- 编码解码

▣ 第三方扩展

▣ Watcher

- Web前端安全的自动审计工具

▣ Wireshark

- 各种强大的过滤器语法

▣ Tcpdump

- 命令行的类Wireshark抓包神器

▣ Python

▣ urllib2

▣ 打开请求响应调试

- 编辑urllib2的do_open里的h.set_debuglevel
- 改为h.set_debuglevel(1)，这时可以清晰看到请求响应数据，包括https

▣ 什么是跳转

▣ 服务端跳转

▣ 302

- `<?php header("Location: 3.php"); ?>`

▣ 301

- `<?php header("HTTP/1.1 301 Moved Permanently"); header("Location: 2.php"); ?>`

▣ u=urllib2.urlopen(url)后，u.url能得到服务端跳转后的地址

- urllib2自己的特性
- 所谓的会跟进去

▣ 客户端跳转

▣ `<meta http-equiv="refresh" content="0; url=http://www.evilcos.me" />`

- htmlparse解析就行了

▣ `location.href="http://evilcos.me";`

- 正则（弱），js引擎（王道）

▣ Python编码规范

-  [PythonCodingRule.pdf](#) 

▣ 入门书

☐ Python核心编程2

☐ 第4章 Python对象

- 完整熟练

☐ 6.8 Unicode

- 完整熟练

☐ 8.11 迭代器和iter()函数

- 完整熟练

☐ 第9章 文件的输入和输出

- 完整熟练

☐ 第10章 错误和异常

- 完整熟练

☐ 第11章 函数和函数式编程

- 完整熟练

☐ 第12章 模块

- 完整熟练

☐ 第14章 执行环境

- 完整熟练

☐ 第15章 正则表达式

- 💡 完整熟练

☐ 第18章 多线程编程

- 完整熟练

☐ 20.2 使用Python进行Web应用：创建一个简单的Web客户端







- 完整熟练

☐ Office能力

- Word文档编写，看去要专业，尤其对外的
- Excel里面大量的统计、图表功能，需要善于使用
- PPT演讲、培训等必备，如何做好PPT？百度一下……

☐ 😊 进一步

- yEd
- Visio

- FreeMind
 - 本技能表就是这个制作
- 熟练VIM
 - 实战至少3回合: <http://coolshell.cn/articles/5426.html> 
- 算法
 - 快排
 - 二分
- 正则表达式
 - 调试工具
 - 🤖 Kodos
 - 💡 RegexBuddy
 - 支持多种语言
 - 支持调试优化
 - 😊 <http://www.regexper.com/> 
 - 正则图解
 - 正则表达式30分钟入门教程: <http://deerchao.net/tutorials/regex/regex.htm> 
 - <http://wiki.ubuntu.org.cn/Python正则表达式操作指南> 
 - 📎 [regex/regularexpressions.pptx](#) 
 - 📎 [regex/正则表达式引擎浅析.txt](#) 
- 研发能力
 - 瀑布模型
 - 需求->需求分析->设计->开发->测试->上线->运维/运营
 - 💡 需求分析能力
 - 给你一个需求, 如何给出一个优美的执行思路——方法论
 - 这个能力非常非常非常的关键
 - 调试能力
 - 只要定位出, 就没有解决不了的Bugs
 - 肉眼看到的都是假象
 - 一定要专业的工具与经验配合
 - Bugs在哪出现, 最终就在哪进行真实模拟调试

▣ 缩小范围

▣ 构建自己的测试样例

- 排除网络复杂未知情况
- 关联模块一个个排除

▣ Python单步调试

- `import pdb;pdb.set_trace()`
- 在需要单步调试的地方加上面这句，运行程序后中断在此，然后查看指令进行一步步细细调试
- 粗暴调试: `print`

▣ 敏捷思想

- 快速迭代
- 任务拆细
- v1原则: 定义好v1的目标，快速完成v1为优先
- 习惯Wiki记录，利于沉淀与分享

▣ 翻墙

- 💡 <http://code.google.com/p/goagent/> 

▣ SSH隧道

- <http://www.ibm.com/developerworks/cn/linux/1-cn-sshforward/index.html> 

▣ 本地转发

- `ssh -L <local port>:<remote host>:<remote port> <SSH hostname>`

▣ 远程转发

- 反弹
- `ssh -R <local port>:<remote host>:<remote port> <SSH hostname>`

▣ 动态转发

- `ssh -D <local port> <SSH Server>`

▣ Web安全

▣ Web服务组件

- 8+1: 一图胜千言哎:) 

▣ 钟馗之眼

- 网络空间搜索引擎
- 大数据，懂的人懂，不懂的人不懂

- <http://www.zoomeye.org> 
- 组件具有影响面，越底层的组件影响面可能越大
- ▢ 安全维度
 - 漏洞
 - 风险
 - 事件
- ▢ Web安全标准
 - OWASP
 - WASC
 - 我们内部Wiki
- ▢ 实战环境
 - ▢ XSS
 - ▢ ks-xsslab_open（内部虚拟机）
 - ▢ 可以搞通
 - XSS
 - CSRF
 - ClickJacking
 - ▢ <http://xss-quiz.int21h.jp/> 
 -  答案: [xss/xss_quiz.txt](#) 
 - ▢ SQL
 - ▢ <https://github.com/Audi-1/sqli-labs> 
 - SQLI-LABS is a platform to learn SQLI
 - 500多个WSL靶场
 - ▢ 渗透虚拟机/BT5/Kali
 - 海量各类型黑客工具
 - ▢ 书
 - 黑客攻防技术宝典（Web实战篇）
 - 白帽子讲Web安全
 - ▢ Web前端黑客技术揭秘
 - 我和xisigr自己出品

- SQL注入攻击与防御

☐ papers

- <http://www.exploit-db.com/papers/> 
- BlackHat/Defcon/国内各安全沙龙等Papers需要持续跟进

☐ 研发清单

☐ 编码环境

- pip
- Vagrant
- tmux/screen
- vim
- zsh + oh-my-zsh
- Python2.7

☐ >Django1.4

- <http://djangobook.py3k.cn/2.0/> 

- web.py
- node.js
- Ubuntu/Gentoo/Centos
- ipython

☐ 版本控制

- git/svn
- gitlab

- Nginx+uWSGI

☐ Python







☐ 官方手册

- 至少过一遍，这都没过一遍，视野会局限
- 行之说：「我没看过Python的书，却熟读官方手册……」

☐ Linux

☐ 书

-  [Bash新手指南.pdf](#) 
-  [高级Bash脚本编程.pdf](#) 

-  Bash快捷操作.txt 
-  screen最佳实践.pdf 
-  crontab格式详解.pdf 

▢ 前端

▢ 书

- JavaScript DOM编程艺术

▢ 了解DOM

- 这同样是搞好前端安全的必要基础

▢ 库

▢ jQuery

- 优秀的插件应该体验一遍，并做些尝试
- 官方文档得过一遍

▢ ECharts

- 来自百度

■ Google API

▢ ZoomEye Map组件

- ZoomEye团队自己基于开源的打造

▢ AngularJS

- Google出品的颠覆性前端框架

▢ Bootstrap


- 应该使用一遍

▢ 爬虫进阶

▢ 代理池

- 爬虫「稳定」需要

▢ 网络请求

- wget/curl
- urllib2/httpplib2/requests
-  scrapy

▢ 验证码破解

- pytesser

☐ 调度

- crontab是最原生的定时调度
- 基于redis实现的分布式调度
- 基于rpyc实现的分布式调度
- celery/gearman等调度框架

☐ 并发

☐ 线程池

- 进程内优美的并发方案

☐ 协程

- 进程内另一种优美的并发方案

☐ 多进程

- os.fork
- 💡 multiprocessing

☐ 数据结构

- JSON
- cPickle
- protobuf

☐ 数据库

- MySQL
- MongoDB
- Cassandra
- Hadoop体系
- Redis
- Sqlite
- bsddb

☐ DevOps

- SSH证书
- Fabric
- SaltStack
- puppet
- pssh/dsh

☐ 调试

- pdb
- logging
- Sentry
- strace/ltrace
- lsof
- ▢ 性能
 - ▢ Python内
 - timeit
 - cProfile
 - Python性能分析指南: <http://www.oschina.net/translate/python-performance-analysis> 
 - ▢ Python外
 - top/htop/free/iostat/vmstat/ifconfig/iftop...
- ▢ 算法
 - 分词
 - ▢ 贝叶斯
 -  [algorithm/贝叶斯.txt](#) 
 - 神经元
 - 遗传算法
 - 聚类/分类
 - ...
- ▢ 持续集成
 - ▢ 自测试
 - nose
 - Jenkins
- ▢ 协作
 - 类似Trello的在线协同平台
 - 微信
 - 立会
- ▢ 设计思想
 - 人人都是架构师: 具备架构思想是一件多酷的事
 - 实战出真知

▣ 如何设计

-  [任务架构设计变迁.pdf](#) 

- 松耦合、紧内聚
- 单元与单元属性
- 生产者与消费者

▣ 结构

- 队列
- LRU

▣ 分布式

- 存储
- 计算

▣ 资源考虑

- CPU
- 内存
- 带宽

▣ 粗暴美学/暴力美学

- 大数据，先考虑run it，然后才能知道规律在哪
- 「run it优先」能快速打通整体，洞察问题
- 「run it优先」能摆脱细节（繁枝末节）的束缚
- 「run it优先」能快速迭代出伟大的v1

▣ 一个字总结

- 美

▣ 优质资源

- 知乎周刊: <http://zhuanlan.zhihu.com/Weekly> 
- 码农周刊: <http://weekly.manong.io/> 
- Pycoder's Weekly: <http://pycoders.com/archive/> 
- Hacker News: <https://news.ycombinator.com/> 
- Startup News: <http://news.dbanotes.net/> 
- 极客头条: <http://geek.csdn.net/> 
- InfoQ: <http://www.infoq.com/cn> 
- Stack Overflow: <http://stackoverflow.com/> 
- GitHub: <https://github.com/> 
- FreeBuf: <http://www.freebuf.com/> 

- WooYun: <http://drops.wooyun.org/> 

[Expand](#) - [Collapse](#)

☐ 牛人1, 2, 3

- 1研究: 研究东西, 有足够洞察力, 研究水准不错
- 2研发: hack idea自己有魄力实现, 不懂研发的黑客如同不会游泳的海盗
- 3工程: 研发出来的需要实战、需要工程化, 否则只是玩具, 而不能成为真的武器