安全技术经典译丛

车联网渗透测试

[美] 艾丽萨·奈特(Alissa Knight) 著

梆梆安全研究院 译

清华大学出版社

北 京

北京市版权局著作权合同登记号 图字：01-2021-5737
Alissa Knight
Hacking Connected Cars: Tactics, Techniques, and Procedures
EISBN：978-1-119-49180-4

图书在版编目(CIP)数据

车联网渗透测试/ (美)艾丽萨·奈特(Alissa Knight) 著；梆梆安全研究院译. —北京：清华大学出版社，2021.10
(安全技术经典译丛)
书名原文：Hacking Connected Cars：Tactics, Techniques, and Procedures
ISBN 978-7-302-59249-5

Ⅰ. ①车… Ⅱ. ①艾… ②梆… Ⅲ. ①汽车－物联网－测试 Ⅳ. ①U469-39

中国版本图书馆 CIP 数据核字(2021)第 191819 号

责任编辑：王 军
封面设计：孔祥峰
版式设计：思创景点
责任校对：成凤进
责任印制：朱雨萌

出版发行：清华大学出版社
网 址：http://www.tup.com.cn，http://www.wqbook.com
地 址：北京清华大学学研大厦 A 座 邮 编：100084
社 总 机：010-62770175 邮 购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：三河市国英印务有限公司
经 销：全国新华书店
开 本：148mm×210mm **印 张**：8.75 **字 数**：227 千字
版 次：2021 年 10 月第 1 版 **印 次**：2021 年 10 第 1 次印刷
定 价：59.80 元

产品编号：086618-01

译 者 序

网联汽车作为汽车与信息、通信产业跨界融合的典型应用，在为生活带来便利的同时，也引入了各种各样的网络安全风险，包括远程攻击、数据窃取、信息欺骗等，这些风险不仅影响财务安全，还可能危及人的生命安全。

如今，国内外已经越发重视网联汽车的信息安全问题。在译者翻译本书的过程中，我国发布了《智能网联汽车生产企业及产品准入管理指南(试行)》(征求意见稿)、联合国欧洲经济委员会发布了UNECE R155法规。二者不约而同地对汽车生产企业及产品的网络安全防护提出了要求，网络安全已成为整个网联汽车产业亟须解决的重点问题。

正所谓未知攻，焉知防，要提升产品的网络安全防护能力，除了要研究先进的网络安全防护技术，还需要知悉常见的网络安全攻击手段、策略和技术。本书就重点介绍了当前主流的网联汽车破解策略、技术和步骤。了解这些知识既可帮助企业更有效地制定相应的安全防护策略，又可方便车企及安全人员对车辆进行有效的渗透测试，这也是我们选择翻译这本书的初衷。

本书共分为两部分：第一部分根据渗透测试执行标准按阶段重点介绍了网联汽车渗透测试的前期准备、信息搜集、威胁建模、漏洞分析、漏洞利用、后渗透阶段过程中所需的技术、策略和步骤，第二部分则重点介绍了风险管理、风险评估框架、汽车 PKI、报告

等安全管理相关的技术、步骤。具体的渗透技术涵盖了当前热门的伪基站搭建及利用、蓝牙攻击、WiFi 攻击、PKI 攻击等。

今天，网联汽车的功能日新月异，相应的渗透测试技术也在发展、变化，希望读者朋友可以以本书为立足点进行更广泛、更深入的学习研究。以风险管理为例，ISO 和 SAE 联合制定的 ISO/SAE 21434《道路车辆-网络安全工程》也对相关内容进行了讨论。如本书的作者 Alissa Knight 所言，网联汽车受到的安全挑战是对整个社会的挑战，因此整个社会应该一起努力解决这个问题。

本书由梆梆安全研究院翻译，在研究院院长卢佐华女士的带领与统筹安排下，整个团队为保证翻译质量付出了不懈的努力。参与本书翻译的人员有卢佐华、袁森、王晓霞、王远远、周鑫飞、王亚东、陈建全、朱鹏。由于本书的专业性强，相关术语生僻，译者们在翻译的过程中查阅、参考了大量背景资料，在不偏离原意的前提下尽可能保证信达雅，对于其中翻译不到位的地方，还请读者见谅。

梆梆安全研究院成立于 2016 年 1 月，发布了《智能网联汽车信息安全白皮书》《智能交通网络安全实践指南》等白皮书，翻译了《物联网安全》([美]李善仓(Shancang Li)和许立达(Li Da Xu)著)。本书所述的渗透测试技术具有很强的实操性，我们在翻译的过程中也对其中的关键技术进行了验证，涉及的对象包括比亚迪、北汽、五菱、江淮、宇通客车等厂商的部分关键零部件。

感谢清华大学出版社的编辑们，没有他们的帮助和鼓励，本书不可能顺利付梓。

译　者

作 者 简 介

Alissa Knight 在网络安全领域工作了20多年。在过去的10年中，她将漏洞研究的重点放在了为美国、中东、欧洲和亚洲的客户破解网联汽车、嵌入式系统和IoT设备方面。她持续地与世界上一些大的汽车制造商和OEM厂商合作，以制造更安全的网联汽车。

Alissa是Brier & Thorn集团的CEO，也是Knight Ink的管理合伙人。她将黑客技术与文字内容创作和视觉内容创作结合起来，贡献给网络安全领域的参与者和市场领导者。作为一名连续创业者，Alissa曾是Applied Watch和Netstream的CEO。她曾在并购交易中将公司出售给国际市场上的上市公司。

她的职业热情在于会见世界各地的佼佼者们并向他们学习，分享她对重塑全球市场的颠覆性力量的看法。Alissa 的长期目标是帮助尽可能多的组织制订和执行其战略计划，并专注于其风险较高的领域，弥合孤岛效应，有效地管理跨组织边界的风险，并使他们能明智地承担风险，以实现长期价值创造。

致　　谢

我要感谢我一生中来来往往的许多人，以及在整个过程中帮助我更好地理解漏洞研究这一神秘领域的人们。在许多方面，我与他们的合作为本书的内容奠定了基础。特别感谢 Robert Leale、The Crazy Danish Hacker、“Decker”、Solomon Thuo、Karsten Nohl 博士(密码学专家)、Ian Tabor、Graham Ruxton 以及一路走来的每个人，他们在成书过程中教会了我很多，并在无数个日日夜夜里支持我撰写本书。

我还要向我的父亲 Sojourn 致以深深的敬意，他一直没能出版他自己的图书。虽然他去世得很早，但他的生活却比那些活了 100 年的人还要充实得多。

我还要感谢我的儿子 Daniel，他一直是我的灵感来源，也是我每天早上醒来的理由，他将永远是我最大的成就。我的姐姐和妈妈是我认识的最坚强的女人，而且她们也知道如何不受约束地去爱。我最好的朋友 Emily 和我的商业伙伴兼朋友 Carlos Ruiz 教我如何真正生活并让我成为最好的自己。

最后，我要感谢我生命中最爱的人，我最好的朋友、爱人，也是我最忠诚的粉丝——Melissa。“我一只手就能征服世界，只要你牵着我的另一只手。”

序　言

汽车网络安全也许是人类有史以来最独特、最具挑战性的安全问题。载着人和重要货物的几千磅重的机器在高速行驶，周围都是其他完全联网的、自动化的甚至与周围环境通信的类似机器。随着大量新技术进入汽车领域来提升这些新功能和新特性，普通汽车可能也有 10～100+万行的代码，并且需要管理多种协议。随着汽车的复杂性不断上升，很容易想象，在任何给定的车辆中都可能存在很多潜在的安全漏洞。

作为菲亚特-克莱斯勒汽车公司车辆安全保障项目的前全球负责人(2017—2019 年)，我每天都需要利用多种工具来应对这一复杂的挑战。其中，我使用的功能最多的工具之一是行业推广计划。通过该计划，我与独立研究人员建立了联系，以鼓励和促进针对我们系统的安全性研究。也是通过这个计划，我第一次遇到了 Alissa Knight。Alissa 的努力和著作填补了汽车行业公司和其他研究人员在教育和意识方面的巨大空白。通过阅读 Alissa 的著作，我个人已经成长为专业的黑客。

这种安全挑战是对社会的挑战。因此，不只是生产产品的企业，整个社会都应该努力解决这个问题。Alissa 是安全意识的倡导者和最佳的实践者，她为我们所有人推动了一个更安全的未来。我希望这本书的内容，以及 Alissa 的其他几本著作，能帮助你成为一个更有安全意识的人。负责任地使用本书的内容，加入当地的安全研究

小组，以 Alissa 为榜样，回馈社会，让大家都能受益。

Thaddeus Bender
全球车辆安全保障项目经理
菲亚特-克莱斯勒汽车公司

信任——一种使我们人类能了解周围世界的必要情感。这是一种最原始的需求。当我们吃饭时，我们必须相信食物不会毒死我们。味觉和嗅觉的发展，只是为了让我们信任我们的饭菜。我们走路时需要知道下一步不会掉下悬崖，也不会撞上大橡树。所以，我们培养了视力，以免让周围的环境要了我们的命。我们必须信任与我们交往的人。所以，我们培养了自己的猜疑心和幽默感。

信任是我们的生存之道，是我们生活中的必需品。它蕴含在我们每一个有意识和无意识的决定中——每个人都是如此。因此，当我们吃饭、走路、睡觉，甚至开车时，我们必须相信，使我们移动的传感器和系统不会让我们过早地赴死。这也是未来出行的关键。车辆需要被信任。自动驾驶汽车必须赢得我们的信任。虽然自动驾驶技术尚不完美，但人们有可能过度信任该系统。

2016 年，第一起自动驾驶汽车死亡事件发生了。车辆驾驶员 Joshua Brown 相信，他的自动驾驶系统不会允许他的车辆全速撞上半挂车。他的系统运行正常。问题在于，当时的半挂车是白色的，在明亮的天空下，车辆的物体检测算法无法将挂车与周围环境区分开来。不过，该系统的运转确实跟宣传的一样。只是自动驾驶系统并非为应对所有情况而开发，所以用户必须时刻关注路况。本案中用户对系统的信任度太高了。在自动驾驶成功案例的海报频频出现的某个地方，Joshua 由于过度信任他的自动驾驶系统而付出了惨重代价。

在不久的将来，下一代自动驾驶汽车将到来，并且这些系统将被宣传为不需要用户干预即可运转的系统。车辆的驾驶员实际上将

在系统启动后变成忽略车辆的速度、轨迹或周围环境的乘客。这些系统将需要操作者用生命来信任构成自主驾驶系统的众多电子控制模块、车载网络、数百万行的代码和电子传感器。最重要的是，车载 WiFi、远程信息处理控制器和车对车通信等新技术提高了复杂性且扩大了攻击领域。

要确保这些系统不被恶意篡改，需要警惕、机智聪明、有条理、有才华的人才，以保证联网的自动驾驶汽车获得信任。而这正是 Alissa Knight 的闪光点所在。她是汽车网络安全的坚定支持者。她不仅希望建立一个网络安全工程师社区，还希望确保汽车制造商及其零部件供应商努力保障其软件、硬件和传感器的安全。

我第一次见到 Alissa 是在德国，她在那里生活和工作就是为了这个目标。在我们第一次见面时，她用一个拥抱迎接我，同时说道："我是个拥抱者。"直觉上，她明白什么是信任。她知道，拥抱将有助于培养一种纽带，帮助我们为当前和未来的项目共同努力。

Alissa 的才华不止于此。她继续致力于讲授和谈论如何保障车辆系统的安全，她在网上讲解如何通过建立和测试蜂窝网络基站来测试远程信息传输系统，并讲授许多其他相关主题的在线课程。

能够认识 Alissa Knight，并与她一起进行多个项目，致力于保护汽车电子系统的未来，我深感荣幸。Alissa，我祝愿你出书顺利，并在今后的生活中能有更多的作品。谢谢你的信任和拥抱!

Robert Leale

CanBusHack Inc.总裁

前　言

战略需要思考，战术需要观察。

——*Max Euwe*

2002 年 5 月 7 日，Bennett Todd 在一个漏洞开发邮件列表中提到，他在审查无线网络时偶然发现了一个 UDP 端口，该端口是用来远程调试 VxWorks 操作系统的，VxWorks 是 Wind River Systems 公司开发的一款实时操作系统(Real-Time Operating System，RTOS)，现已被英特尔收购。他审查的无线网络产品中有一些默认开启了这个端口。Todd 还不知道他发现的这个 17185 UDP 端口会导致一个更广泛的漏洞，将影响到大量运行着 VxWorks 操作系统的不同网联设备。

2010 年 8 月，在 Todd 公开他的发现 8 年后，HD Moore 在 Defcon 第 23 届大会上，向观众们展示了他在 Todd 2002 年研究的基础上，针对每一台 VxWorks 设备进行详尽测试所得到的成果。

在 Wind River Systems 公司于 2010 年 8 月 2 日发布的漏洞说明中，这个端口被证实为它的 WDB 目标代理，这是一个在目标中驻留的运行时工具，开发期间主机工具需要通过这个工具与 VxWorks 系统进行连接。WDB 调试代理的访问是不安全的，Moore 通过抓取内存漏洞，发现部署 VxWorks 的系统存在一个巨大的安全漏洞，允

许远程攻击者在无需有效凭证的情况下对内存中的数据进行读写。

发现漏洞时，Todd 在他的文章中只提到了无线接入点受到的影响，并没有意识到 VxWorks 是嵌入式系统的实时操作系统，它用途广泛，而不只是被无线接入点使用。Wind River 还被用在其他设备中，包括 Thales 公司的 Astute 级潜水艇潜望镜、波音 AH-64 阿帕奇攻击直升机、NASA 的 Mars Rover，甚至宝马 2008 年以后生产的车型使用的 iDrive 系统等，这里仅列举了几个例子。

在病毒学中，一种病毒被引入一个新的宿主物种中，并通过新的宿主种群进行传播的过程，被称为外溢或跨物种传播(Cross-species Transmission，CST)。同样的事情也会发生在信息安全中，目标设备或产品的漏洞被公布，会意外导致病毒外溢到其他产品中。

1996 年，德国的 Rohde & Schwarz 公司开始销售第一款 IMSI 捕捉器(GA 090)，该捕捉器允许用户强迫身份不明的移动用户传输其 SIM 卡的 IMSI。后来，在 1997 年，该捕捉器还支持监听用户呼出的电话。

在 2001 年 4 月举行的 Blackhat Briefings Asia 大会上，Emmannuel Gadaix 公布了第一个已知的 GSM 漏洞，通过中间人(Man-In-The-Middle，MITM)攻击和取消注册拒绝服务(Denial of Service，DoS)攻击影响移动电话的使用。

2010 年，Karsten Nohl 发布了一款名为 Kraken 的破解工具，以破解用于保护 GSM 流量的 A5/1 加密技术，该工具利用彩虹表来破解 A5/1 加密技术，之后被称为“柏林表”。Nohl 的工具后来在同一年被 Kristen Paget 修改，后者在 Defcon 18 大会上透露了如何使用伪基站(Base Transceiver Station，BTS)或 IMSI 捕捉器来拦截移动电话和短信，而根本不需要破解。

今天的网联汽车、自动驾驶车辆的 OTA(Over-The-Air，空中下载)更新和其他功能非常依赖与后端的通信。虽然那些关于 GSM 的漏洞最初是针对移动电话及其用户的，但它们后来导致漏洞外溢到

汽车领域中。

Paget 在演讲中使用了一个价值约 1500 美元的通用软件无线电外设(Universal Software Radio Peripheral，USRP)，该设备比之前的 GA 090 便宜数十万美元，她还提出一个这样的想法：与其通过嗅探 GSM 电话和短信进行离线破解，不如选择另一个策略。Paget 使用手机创建了一个连接到她笔记本电脑的基站，从而能完全禁用 A5/1 加密，使得离线破解变成了多余的操作。

Paget 后来开始为特斯拉工作——无疑是将她之前对移动网络黑客技术的研究应用到了网联汽车的安全问题上。现在她作为一名黑客为 Lyft 工作。Paget 在会议期间观察到，GSM 规范本身要求在网络加密(A5/0)功能被禁用时向用户发出警告通知，而该警告在蜂窝网络上被故意禁用，这一点尤其令人震惊，凸显了汽车制造商在其远程通信基础设施方面依赖的手机运营商的系统性问题。

就在 2015 年举办的 DEFCON 23 上，Charlie Miller 和 Chris Valasek 演示了对一辆未经改装的乘用车的远程利用，与他们的第一次演示不同的是，这次并不需要物理访问汽车及其诊断接口。Miller 和 Valasek 演示了汽车主机(HU)的一个漏洞，该漏洞允许他们在未经身份验证的情况下与 TCP/6667(dbus)进行通信，因而他们可通过连接 HU 的 WiFi 热点向主机系统发送指令。更糟的是，由于移动运营商的蜂窝网中的防火墙太弱，他们可通过远程信息处理控制单元(Telematics Control Unit，TCU)的 GSM 接口访问 dbus 端口，执行相同的攻击。通过修改从网上下载的固件并重新刷入瑞萨 V850 微处理器，他们能重新编写微处理器的代码，将 CAN 消息直接发送至 HU 所连接的 CAN 总线，并实际控制汽车，如踩刹车、转动方向盘、熄火、开启雨刷和控制音响。

这是首次公开发布的针对网联汽车的远程黑客攻击。之前公开的利用技术都要求对汽车的 OBD-II(调试)接口进行物理访问或连接。

自 2015 年以来，越来越多的漏洞被公布，这些被利用的漏洞存在于众多品牌和型号的网联汽车的内部组件中，而不仅是 HU 上。其中一些被利用的漏洞是由于原始设备生产商(Original Equipment Manufacturers，OEM)使用未签名的固件而造成的，未签名的固件允许研究人员在其中植入后门并重新刷入微处理器，之后，研究人员可将数据发送到 CAN 总线上，从而实现对车辆的物理控制。

这种漏洞外溢的情况不仅影响 GSM，还会影响蓝牙、WiFi 和汽车 OEM 厂商们使用的其他嵌入式操作系统。

从现代交通工具的角度来看软件代码的数量，F-35 联合攻击战斗机大约需要 570 万行代码来操作它的系统。如今，高级网联汽车拥有近 1 亿行代码，运行在汽车整个车内网络中的 70～100 个基于微处理器的电子控制单元(Electronic Control Units，ECU)上。随着网联汽车和自动驾驶汽车复杂性的提高，根据 Frost & Sullivan 的估计，在不久的将来，汽车将需要 2 亿～3 亿行代码，而当前召回的汽车中，60%～70%的问题都是由电子故障造成的。

一个无可争议的事实就是，网联汽车和自动驾驶汽车已不再是无法实现的未来，而是今天的现实。2020 年，路上行驶的网联汽车和自动驾驶汽车的总数已超过 1000 万辆。

当然，汽车行业的技术进步无疑将有助于提高效率和增加收入。“追求舒适、便利”的这一代人在成长过程中期待着与电子邮件、网络和社交网络随时连接。KPMG UK 估计，从 2014 年到 2030 年，自动驾驶汽车可减少 2500 起重大汽车安全事故；这份大胆的声明得到了本田研发部的美国负责人的支持，他为公司制定了到 2040 年实现零事故的目标。

虽然许多 OEM 厂商仍采用 CAN 总线等众多较旧的技术，但它们已经开始将 ECU 集成到汽车中，并使用基于以太网的 TCP/IP 进行通信。需要指出的是，在 2015 年，一辆汽车中的 ECU 数量最多约为 80 个，而今天，由于成本降低和整体重量下降，一辆豪车中

ECU 的数量可超过 150 个。

如今我们正处在第三次工业革命中，无人驾驶、自动驾驶汽车正迅速成为现实，而道德黑客/渗透测试人员也越来越重视研究如何识别和利用汽车中的漏洞。

正如 Garth Brooks 所述，无人驾驶汽车的出现，就像“我们曾经推迟到明天的事情现在变成了今天”。但汽车技术发展中的军备竞赛已经形成新的威胁格局，在这种威胁下，攻击的结果不再局限于被破坏的网站或被盗用的信用卡号码，而是潜在的生命损失。事实上，网联汽车不再仅被看作一堆由内燃机驱动的金属，通过转动曲轴来移动车轮，而黑客对其一无所知。现在，汽车不过是带有轮子的计算机，由多个 CPU、嵌入式操作系统和可通过蓝牙、WiFi 和 GSM 进行通信的应用程序组成的技术堆栈，由出价最低的竞标者支付并制造。

术语和定义

考虑到最近关于网联汽车网络安全的新闻报道，媒体对术语的淡化、自媒体和平台对术语的误解等，现在我们有必要就一些基本术语定义达成一致。

车间通信(Inter-Vehicle Communications，IVC) 是指车辆之间、车辆与移动网络以及车辆与路侧单元(Road Side Units，RSU)之间建立的外部通信，因此不包含车辆自身网络内 ECU 之间的任何通信。

车载自组网(Vehicular Ad-Hoc Network，VANET) 与 IVC 是同义词，常可互换使用，但 VANET 更多是指道路上两辆车之间动态建立的自组网络，而不是车辆与基础设施 RSU 之间建立的网络。VANET 的一个例子就是两辆车之间建立的无线自组网络，用于共享前方即将发生的道路危险信息，如坑洞等。

智能交通系统(Intelligent Transportation System，ITS) 是今天很常用的一个术语，指的是 IVC，并迅速成为它的代名词。有趣

的是，那些没有在汽车行业工作过的人在尝试使汽车更智能之前，曾试图使运输系统(如道路)更智能(但失败了)，他们不尝试让业内的OEM 标准化协议，如 IEEE 802.11，这是一个被称为智能车辆-高速公路系统(IVHS)的术语。

车对车(Vehicle to Vehicle，V2V)、车对基础设施(Vehicle to Infrastructure，V2I)、和车对一切(Vehicle to Everything，V2X) 是业界常用的术语，用于描述车辆与另一个节点(如车辆或基础设施本身)之间的通信终端(通俗地说，有些人把 car 和 vehicle 互换使用，提出 C2C、C2I、C2X，但这种情况很少见)。

IEEE 802.11，正如计算机行业的朋友们所知道的那样，这是无线局域网(WLAN)技术及其修订版的标准，其中包括 802.11a、802.11b、802.11g 和较新的 802.11ac。它已被用于 HU 和 TCU 之间的通信以及车间通信。由于某些方面的原因，原有的 802.11 标准缺失了一些功能，IEEE 802.11p 是为了解决 IVC 的这些不足而开发的，明确界定在 5.9GHz 范围内，因传输距离短而很少在消费类家庭网络中使用。

脆弱性评估，又称脆弱性分析，是指通过人工或自动化手段对系统、网络或通信基础设施中可能影响系统机密性、完整性或可用性的安全缺陷进行识别、定义和分类。脆弱性能否被利用对于能否将其归类为脆弱性并不重要。

渗透测试是对系统或网络进行的模拟攻击，旨在识别和利用目标中的漏洞。这些测试展示了可成功利用目标的现实世界的攻击场景，以便更好地保护目标，使之免受现实世界的攻击。

杀戮链，或称杀戮链模型(KCM)，是军方最初设想的一系列预先设定的步骤，以描述攻击的结构。这个术语(就像其他网络安全术语一样)已被军方运用于网络安全领域，Lockheed Martin 公司将其正式定义为“网络杀戮链模型”。步骤描述：①侦察；②武器化；③投送；④使用；⑤安装；⑥命令和控制(Command and Control，C2);

⑦针对目标的行动。人们可能认为安装和 C2 不可能在 TCU 或 HU 上实现，但我将在本书中证明，根据 HU 或 TCU 的架构，这实际上是可以做到的。

风险，特别是 IT 领域的风险，是指某个特定的威胁利用资产或资产组中的漏洞的可能性，采用发生的可能性和影响来衡量。

对于非汽车专家

汽车机电一体化是关于汽车工程中的机械和电子技术的研究。因为这个工程领域涉及的范围很广，且业界有关于它的专著，所以本书只介绍汽车机电一体化中与汽车网络安全最相关的领域，以及你从事这项工作时要多了解的内容。

这里有一个简单的要求，希望你忘掉你对汽车的一切想法，并记住一件重要的事情：汽车在过去 15 年里已经发展为车轮上的计算机网络。此处说网络，是因为车辆本身就是一个由运行微处理器的电子控制单元(Electronic Control Units，ECU)、Linux 或 Android 等操作系统组成的车内网络。要知道，最新制造出的汽车甚至用车载以太网作车内网络。在车内网络上运行的 ECU 现在甚至可通过 TCP/IP 进行通信。以太网总线可能连接到一个连接着 CAN 总线的网关上。需要注意的是，较新的汽车需要利用以太网提供的较大的 MTU(Maximum Transmission Unit，最大传输单位)，而不是 CAN 的带宽限制。这并不是说，随着车载以太网的出现，其他通信技术就不存在了，因为将更小、更便宜的 ECU 迁移到以太网上是没有意义的。然而，对于那些负责执行时间敏感任务的功能更多、更丰富的 ECU 而言，车载以太网还是有市场的。

本书将尽可能用最通俗的语言来解释汽车的机电一体化，从你可能遇到的不同网络拓扑结构开始，然后是不同的协议，最后是 ECU

本身。在此需要注意的是，本书将从浅显的层面来解释所有这些技术，这样你就可理解你在目标环境中的工作，而不需要理解如何自己构建 ECU。如果你想在这些领域中的某一个进行拓展，建议你从众多关于网联汽车的书籍中，或者从博世汽车工程指南中挑选一本，它们对这些主题进行了详细的分解。

汽车网络

你必须开始把汽车看成一个由节点(ECU、执行器等)组成的半独立网络，这些节点都是通过网络相互通信的，无论这个网络是 CAN 总线、以太网、MOST、FlexRay，还是近几十年来可能出现或消失的其他技术。这里说半独立，是因为有一个进入车内网络的入口，比如 TCU 的 GSM 接口或者 HU 上运行的 WiFi 接入点等。这一内容在后面的章节里有更详细的介绍，所以此处不再赘述。在一些广告中，当汽车急转弯时，汽车的前大灯会朝道路的方向转动，或者可自动并线停车，如果你看过这些广告，那么你要明白，这些事情发生的唯一途径就是前大灯、方向盘等都在彼此之间发送和接收数据——实际上是相互“对话”。在此处的大灯例子中，驾驶员在转动方向盘时，方向盘实际上是在与 ECU 进行通信，将数据发送给控制前大灯的 ECU，因此，汽车转弯时，前大灯会随之转动。大灯并不是因为准确预测了驾驶员将要做什么而自动转动。很遗憾，人工智能爱好者们，能够读懂司机心思的方向盘恐怕还是虚构的，但这并不是说以后也不可能出现。

车内通信

现在，汽车中几乎每个组件(从锁、门把手到前大灯和刹车灯)

都由连接到车载网络的 ECU 控制，因此，它们可向车中的其他 ECU 发送和接收信号，这些 ECU 接收数据并作出适当的响应。事实上，只是打左转向灯就用到了不少于 8 个嵌入式系统。因此，如今影响汽车的故障中 90%以上都与电气问题有关。ECU 只是运行微处理器和嵌入式操作系统的嵌入式系统，用来接收来自传感器或触发执行器的数据。ECU(不包括那些较小的设备，如电源锁)从闪存中启动，这要求它们具有预编译的固件。本书后面的内容将演示如何利用它。

剧透警告：本书甚至会告诉你，一个漏洞研究人员最近演示了一种方法，只需要将汽车的前大灯拆掉，就可直接访问 CAN 网络并获得对 CAN 总线的完全读写权限。把 CAN 总线看成一个常规渗透测试的内部网络，相当于你已经接入了内网。这就像前文中刚描述的那样，一旦有了在 CAN 总线上发送和接收信号的能力，就可控制汽车的物理属性，从转动方向盘，到踩刹车、油门踏板，甚至是启动或关闭汽车。因此，访问 CAN 总线(网络)实际上就是在 Windows 域中获得超级用户级(企业管理员)的访问权限。与网络上的服务器不同，设备之间可能没有进一步的认证，这意味着你可向 CAN 总线发送消息，告诉汽车关闭，该过程不会提示你输入用户名或密码，也不会出示一个公钥，要求你用私钥进行认证。对 HU 或 TCU 进行渗透测试时，你会遇到不同的网络。虽然网络拓扑结构本身并不重要，但有必要了解现存的一些技术。

最近汽车的发展推动了以太网的使用，如自动驾驶辅助系统(Automated Driver-Assistance Systems，ADAS)的现代化，ADAS 现使用车载网络不同域的数据，这对数据交换速率提出了很高的要求，同时有低延迟和严格同步的要求，以减少或消除缓存的需要。延迟对自适应巡航控制(Adaptive Cruise Control，ACC)之类的系统可能是毁灭性的，因为自适应巡航依赖于多种数据源，如里程表、高分辨率视频、雷达和激光雷达(Light Detection and Ranging，LIDAR)等。

未来汽车的发展将包括合作式自适应巡航控制(Cooperative Adaptive Cruise Control，CACC)，它将在严格的实时约束下，通过 VANET 无线接收来自附近其他车辆的数据。随着 BYOD 和其他售后市场的定制化，消费者要求更高的吞吐量、更高的传输速率，因此，消除不同域和总线系统正迅速成为当下的需求，从而推动 ECU 通过车载以太网向单一、统一、高速率传输的总线系统迁移。注意，较小、较便宜的 ECU 不需要迁移到以太网，仍然可通过 MOST 或 FlexRay 运行，而以太网作为另一条总线连接到车载网络的中央网关。

最近，为了解决现代汽车线束越来越重的问题，无线网络被引入汽车中。在如今的现代汽车中，线束的重量很容易超过 30kg。除了成本外，断线问题也是一直以来都备受关注的一个问题，而车载无线网络的实施解决了这一问题。

无线技术还没有得到广泛的应用，这很可能是小型、廉价的 ECU 在成本上的限制所致，在这些 ECU 上，这种技术是没有意义的。无线技术在 HU 与 TCU 的连接中得到了应用。车辆中的 BYOD 也需要无线连接，消费者对车内热点的需求日益增长。此外，消费者更倾向于使用手机的 GPS 进行导航，而不是使用汽车出厂时 HU 内置的 GPS，以利用更智能的导航系统，通过提供众包数据(如 Waze)的应用程序识别实时道路危险或交通状况。通过 TCU 提供的互联网接入，还可使用车载应用程序进行购物。

CAN (Controller Area Network，控制器局域网络)是于 1983 年开发的第一个车载网络总线标准。CAN 作为一种通信机制，是为了满足一个个独立子系统的 ECU 的需求而开发的。子系统可能需要控制一个执行器或接收来自传感器的反馈，而创建 CAN 正是为了满足这些需求。CAN 总线上的所有节点都通过两线系统连接。本书后面的章节将讨论黑客入侵 CAN 总线的问题，并且将通过截图来进一步演示。CAN 协议中并不具备固有的安全功能，因此需要制造商实施密码、加密和其他安全控制措施，以免节点容易遭到中间人攻击和其

他类型的报文注入攻击。

FlexRay 最早出现在 2006 年，旨在解决早期技术的不足，它提供了完全确定、低延迟、高速的传输方式，支持多种灵活的总线系统类型，如无源总线和混合型、有源星状拓扑结构，每一种都用双通道和两级星状/总线级联混合型。

MOST(Media Oriented Systems Transport，媒体导向系统传输)是由 MOST 公司开发的，专门用作多媒体和信息娱乐总线系统，因此 MOST 需要提供高数据速率和低抖动，并支持售后市场的各种多媒体和信息娱乐系统。MOST 被设计成一个单向环状拓扑结构，支持连接 64 个 ECU 和 1 个主控 ECU。

图 0-1 是车载网络的一个例子，其中的 ECU 可连接到一个网络，甚至可连接到两个不同的网络之间。不同类型的总线通过网关连接。

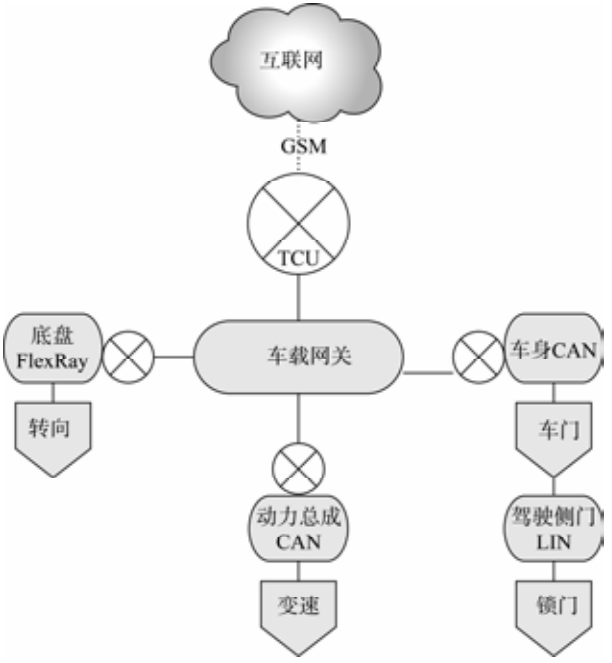


图 0-1　车载网络

车间通信

车间通信(IVC)定义了一个网络，在该网络中，车辆和路侧单元(RSU)是相互通信的节点，可相互提供安全关键警告和交通信息等。

IVC 中存在几种可能的通信范式，包括 RSU、全球定位系统(Global Positioning Systems，GPS)、停放的车辆，甚至广泛部署的蜂窝网络。

交通信息系统(Traffic Information Systems，TIS)是依赖 IVC 的应用程序的最佳例子；具体而言，导航系统在广义上是如何获取关于交通堵塞、道路危险、拥堵、事故等方面的动态更新的。该信息是从导航系统(如 TomTom)以及智能手机应用(如 Google Maps)使用的中央服务器收集的。这些交通信息由中央交通信息中心(Traffic Information Center，TIC)存储和共享，如图 0-2 所示。

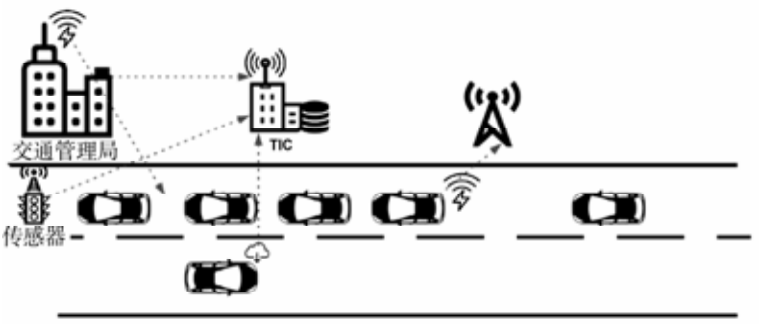


图 0-2　交通信息系统

这是集中式 TIS 的一个例子，还有另一种通信机制，在这种机制中，车辆在道路上经过彼此时直接交换交通信息，形成一个分布式特设网络，车辆之间建立临时连接。这是一种众包式的交通信息交换，也被称为浮动车数据(Floating Car Data，FCD)，如图 0-3 所示。

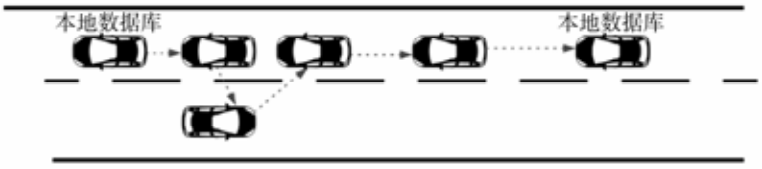


图 0-3 浮动车数据

目前的通信协议都是利用 3G 或 4G 的数据网络——很快就会迁移到 5G，为车内信息上传到 TIC 提供足够的容量。

在 V2V 通信中，WiFi(具体来说是车载网络的衍生品，IEEE 802.11p)被用于支持车辆之间的数据传输，同时正应用于集中式 TIS 架构。这个概念也被称为车载自组网(Vehicle ad-hoc 联网或 VANET)。

目标读者

本书主要面向熟悉网络安全但不是很了解车载机电一体化，并且希望掌握网联汽车网络安全所需的工具和知识的读者。本书也可为需要进行渗透测试或车辆 ECU 风险评估的车载机电一体化专家提供参考。

本书虽然不适合没有传统网络渗透测试经验的读者阅读，但它简要讲解了渗透测试背后的方法论。因此，那些不具备网联汽车渗透测试经验的人应该从本书之外补充一些关于车辆机电一体化和车联网方面的知识。

为了照顾专业知识如此广泛的读者群体，我总结了每一章的要点(因为我本人在阅读书籍时从中受益颇深)，同时提供了一个单独的章节进行定义，以解决汽车机电一体化中比较混乱的术语，供那些从未进行过网联汽车渗透测试的资深渗透测试人员参考。

鉴于此，本书不对车间和车内联网以及车辆机电一体化、应用和协议的基础知识进行深入研究。我把这些内容留给博世公司的专

家和其他在这些领域出版过优秀图书的人士。

本书是一本实践手册，将我自己对黑客入侵网联汽车以及对亚洲、欧洲和美国一些最大的 OEM 的网联汽车移动应用程序、HU 和 TCU 进行风险评估的 10 年研究进行了整理，可用于了解如何为 TCU 和 HU 的微型平台建立可操作的渗透测试实验室。

本书的组织结构

本书根据工作范围细分为两部分。第Ⅰ部分包括渗透测试的策略、技术和步骤。第Ⅱ部分涵盖如何进行风险管理。第Ⅰ部分中的每一章都是根据渗透测试执行标准(PTES)按渗透测试的阶段来组织的。虽然存在多种风险评估框架，但第Ⅱ部分将风险评估和威胁建模的各个章节分解为各自的阶段。本书分为以下章节。

第Ⅰ部分：策略、技术和步骤

第 1 章“前期准备”涵盖了前期准备阶段的行动，通常包括定义利益干系人和其他项目管理步骤，以便为项目做准备，并确保在项目开始前明确界定测试规则和工作范围。

第 2 章“情报收集”涵盖了本阶段需要完成的事项，包括收集工程文档，与利益干系人会面，并确保获得了测试范围内的所有材料和系统的访问权限。

第 3 章“威胁建模”涵盖了不同的威胁建模框架，以及如何在渗透测试过程中执行威胁建模。

第 4 章“漏洞分析”包括主动和被动漏洞分析，甚至包括 CVE 文档审查和供应商的建议，这些建议和审查适用于被测目标的单个部件、软件。

第 5 章“漏洞利用”涵盖了前一阶段可利用漏洞的利用步骤。

第 6 章“后渗透”包括在目标上获得立足点，以及可利用漏洞的后期利用步骤，例如从 HU 目标中反弹 shell。

第 II 部分：风险管理

第 7 章“战略性风险管理”介绍了风险管理过程、风险评估时要涵盖的不同框架、风险处理中要包括的不同阶段，以及在风险评估时对威胁模型进行的简要回顾。

第 8 章“风险评估框架”介绍了现有的各种风险评估方法，以便确定特定项目的最佳框架以及最适合使用的方法。

第 9 章“车辆中的 PKI”讨论了不同的密码分析攻击选项以及在以前的渗透测试中发现的其他漏洞。

第 10 章“报告”涵盖了测试的最后一个重要阶段——报告，其中详细介绍了报告的不同部分，以及如何最合理地展示测试数据。

网站内容

读者可扫描封底二维码下载本书中的参考文件。表 0-1 可作为免费下载的模板，供你在项目中进行网联汽车渗透测试时使用。

表 0-1 模板

标　题	描　述
渗透测试范围文件	用于定义渗透测试范围的模板，其中还包括测试规则
测试规则	定义渗透测试规则的模板。这份文档的最终版本应该由客户签署/执行
RACI 图表	定义团队成员在项目中的角色、职责和责任的模板样本
WBS	工作分解结构样本(WBS)，用作项目管理文档包的一部分，定义了分配给项目团队中每个人的工作
项目章程	用作项目管理文档集的一部分。可下载示例项目章程模板，用于管理渗透测试项目
项目进度表	用于管理渗透测试中的重要里程碑和交付日期的示例项目进度表
风险评估表	用于风险评估的风险评估表样本
风险处理计划	进行风险评估时要使用的风险处理计划样本

需要注意的是，这些模板都是我在自己的项目中给客户的真实交付品的衍生品，为了保护客户的匿名性，很多内容都被删除或编辑了。其中的所有内容都可能是缺乏深度的，但读者应该能自己决定如何在项目中“整理和复用”每个模板。

总结

我最初决定踏上撰写本书的旅程时，希望将过去 10 年来对网联汽车进行渗透测试和风险评估的研究内容进行编纂，从而对信息安全和汽车机电一体化的融合产生持久的影响。我相信本书将帮助世界各地的 OEM 制造出更安全、更有保障的客运车辆。我过去 10 年来有幸在欧洲和亚洲与汽车安全领域知名研究人员合作，他们和我的知识共同奠定了本书的基础。

本书的初稿经受住了安全从业者和汽车工程师的同行评审，我对本书在开创汽车网络安全漏洞研究的新领域中所发挥的作用(无论其作用大小)感到满意。本书已被翻译成北美、欧洲和亚洲主要汽车市场的多种语言，并将成为全球主要的 OEM 厂商关于建立更安全的网联汽车的推荐读物，他们应将其中的隐含知识内化并应用于其中。

最终，本书为解决网联汽车网络安全问题而开辟的道路将变成一个学术领域。这套学术研究将以某种方式冲击这个万物互联世界中的专业知识、人员、项目、社区、挑战、探索和研究的极限。

本书旨在促进全球网络安全界的讨论，并在世界各地的研究人员的讨论中形成丰富的观点，他们会把这本书和自己从工作中获得的知识结合起来，在此基础上进行讨论。此外，我非常乐观地认为，有一天，我将看到网联汽车网络安全成为漏洞研究人员研究的一个突出领域，并成为希望了解并进入网络安全这一深厚领域的全球安全工程师的一个研究领域。

本书所产生的知识体系在某种程度上(无论是赞成还是反对)是非常令人欣慰的，特别是当我看到在这个新的漏洞研究领域里有许多杰出的研究人员(我有幸与他们中的一些人一起工作)实现了我的心愿，即切实影响汽车漏洞研究这个深奥的领域，并为全球的讨论作出贡献。

本书试图提供一个丰富的框架，以理解和实施对 HU 和 TCU 进行渗透测试、威胁建模和风险评估的步骤，同时理解那些随着时间的推移创建的不同框架的丰富性和异构性。

信息安全从未像现在这样成为汽车制造商的核心议题。本书的时效性从未像现在这样强，因为汽车制造商正在努力了解车载网络，该网络以前从未与外界连接，现在却很容易受到威胁，影响到资产的机密性、完整性、可用性和安全性，以及车辆的运行。

事实上，到 2020 年，道路上的汽车已有 1000 多万辆是自主驾驶的，因此，网联汽车的网络安全已成为这个时代的永恒主题。

也许本书将在网联汽车网络安全领域中，在那些观点截然相反的人之间激起持续讨论和对话，为新制定的标准作出贡献，并使人们认识到在开发阶段将安全纳入系统开发生命周期(System Development Life Cycle，SDLC)的重要性，而不是事后再考虑。

原始设备制造商面临的战略和战术网络安全问题日益普遍并且日趋严重，而且人们重新认识到，有必要确保网络安全不再局限于公司内部 IT 安全策略的孤岛，而应使其延伸到其连接的产品线。

我怀着谦虚的态度和极大的雄心，将本书作为行业的基石，开始在第三次工业革命，即万物互联的物联网中构建更安全的联网设备。

而且，我怀着极大的热情和远见，希望本书能在汽车制造商更广泛的生产线中占据一席之地，以推动识别和处理网联汽车中影响人们生命财产安全的 IT 风险。

目　　录

第 I 部分　策略、技术和步骤

第 1 章　前期准备……3
1.1　渗透测试执行标准……4
1.2　范围定义……6
1.2.1　架构……7
1.2.2　完整信息披露……7
1.2.3　版本周期……8
1.2.4　IP 地址……8
1.2.5　源代码……8
1.2.6　无线网络……9
1.2.7　开始日期和结束日期……9
1.2.8　硬件唯一序列号……9
1.3　测试规则……10
1.3.1　时间表……11
1.3.2　驻场测试……11
1.4　工作分解结构……12
1.5　文档收集和审查……13
1.6　项目管理……14
1.6.1　构思和发起……16

1.6.2 定义和规划……22
1.6.3 启动或执行……24
1.6.4 绩效/监督……25
1.6.5 项目完结……26
1.7 实验室布置……26
1.7.1 所需的硬件和软件……26
1.7.2 笔记本电脑的设置……29
1.7.3 Rogue BTS 方案 1：OsmocomBB……30
1.7.4 Rogue BTS 方案 2：BladeRF + YateBTS……34
1.7.5 设置 WiFi Pineapple Tetra……38
1.8 本章小结……39

第 2 章 情报收集……41
2.1 资产登记表……42
2.2 侦察……44
2.2.1 被动侦察……44
2.2.2 主动侦察……61
2.3 本章小结……64

第 3 章 威胁建模……67
3.1 STRIDE 模型……69
3.1.1 使用 STRIDE 进行威胁建模……71
3.1.2 攻击树模型……75
3.2 VAST……81
3.3 PASTA……83
3.4 本章小结……91

第 4 章 漏洞分析……93
4.1 被动和主动分析……94

4.1.1　WiFi……97
4.1.2　蓝牙……108
4.2　本章小结……113

第 5 章　漏洞利用……115
5.1　创建伪基站……117
5.1.1　配置 PC 内部网络……117
5.1.2　让伪基站联网……120
5.2　追踪 TCU……122
5.2.1　当知道 TCU 的 MSISDN 时……122
5.2.2　当知道 TCU 的 IMSI 时……123
5.2.3　当不知道 TCU 的 IMSI 和 MSISDN 时……123
5.3　密钥分析……127
5.3.1　加密密钥……128
5.3.2　证书……128
5.3.3　IV……129
5.3.4　初始密钥……131
5.3.5　密钥有效期……131
5.3.6　密钥存储不安全……131
5.3.7　弱证书密码……133
5.3.8　冒充攻击……133
5.3.9　启动脚本……136
5.3.10　后门 shell……141
5.4　本章小结……142

第 6 章　后渗透……143
6.1　持久访问……144
6.1.1　创建反弹 shell……144
6.1.2　Linux 系统……147

6.1.3 在系统中部署后门……147
6.2 网络嗅探……147
6.3 基础设施分析……149
6.3.1 检查网络接口……149
6.3.2 检查 ARP 缓存……150
6.3.3 检查 DNS……152
6.3.4 检查路由表……153
6.3.5 识别服务……154
6.3.6 模糊测试……155
6.4 文件系统分析……160
6.4.1 历史命令行……160
6.4.2 核心转储文件……161
6.4.3 日志调试文件……161
6.4.4 证书和凭据……161
6.5 OTA 升级……162
6.6 本章小结……163

第Ⅱ部分 风险管理

第 7 章 战略性风险管理……167
7.1 框架……168
7.2 建立风险管理计划……170
7.2.1 SAE J3061……171
7.2.2 ISO/SAE AWI 21434……175
7.2.3 HEAVENS……176
7.3 威胁建模……179
7.3.1 STRIDE……181
7.3.2 PASTA……184

7.3.3 TRIKE……188
7.4 本章小结……190

第 8 章 风险评估框架……191
8.1 HEAVENS……192
8.1.1 确定威胁级别……192
8.1.2 确定影响级别……195
8.1.3 确定安全级别……199
8.2 EVITA……200
8.3 本章小结……204

第 9 章 车辆中的 PKI……205
9.1 VANET……207
9.1.1 车载单元(OBU)……208
9.1.2 路侧单元(RSU)……208
9.1.3 VANET 中的 PKI……208
9.1.4 VANET 中的应用程序……209
9.1.5 VANET 攻击向量……209
9.2 802.11p 的兴起……210
9.3 密码技术……210
9.3.1 公钥基础设施……211
9.3.2 V2X PKI……212
9.3.3 IEEE 美国标准……213
9.4 证书安全……214
9.4.1 硬件安全模块……214
9.4.2 可信平台模块……214
9.4.3 证书固定……215
9.5 PKI 的糟糕实施……216
9.6 本章小结……216

第 10 章 报告……217
10.1 渗透测试报告 ……218
10.1.1 摘要 ……218
10.1.2 执行摘要 ……219
10.1.3 范围 ……221
10.1.4 方法 ……221
10.1.5 限制 ……223
10.1.6 叙述 ……224
10.1.7 使用的工具 ……225
10.1.8 风险等级 ……226
10.1.9 测试结果 ……228
10.1.10 缓解措施……230
10.1.11 报告大纲……230
10.2 风险评估报告 ……232
10.2.1 简介 ……232
10.2.2 参考资料 ……233
10.2.3 功能描述 ……234
10.2.4 HU……234
10.2.5 系统接口 ……235
10.2.6 威胁建模 ……236
10.2.7 威胁分析 ……237
10.2.8 影响评估 ……238
10.2.9 风险评估 ……238
10.2.10 安全控制评估……240
10.3 风险评估表示例 ……243
10.4 本章小结 ……244

第1章 前期准备

给我六小时砍倒一棵树，我会先花四小时磨斧。

——亚伯拉罕·林肯

本章列出实际渗透测试前必要的准备步骤，以开启我们的破解汽车之旅。花费大量时间来准备渗透测试看似本末倒置，但准备不足会导致许多问题。我将根据本书所选的渗透测试框架，按照项目管理知识体系(PMBOK)，介绍定义测试范围的重要性、测试规则(ROE)、应向利益干系人索取哪些工程文件以及项目管理的各个阶段。

在本章的最后，我将介绍在进行远程信息处理控制单元(Telematics

Control Units，TCU)和信息娱乐系统的渗透测试时，你的实验室中会用到的硬件和软件。

开始测试时，你的第一反应是直接打开 bash shell 开始“攻击”，但回想一下本杰明·富兰克林的一句古老格言：“不做准备，就是在准备失败。”

繁杂的准备工作看起来单调乏味，但它对成功完成渗透测试非常重要，以免你和测试团队的其他成员陷入无休止的范围扩大和管理混乱的泥潭，对团队和利益干系人而言都是如此。在进行风险评估时，准备工作也非常关键，因为使用不同的风险评估方法会得到截然不同的结果。

渗透测试框架旨在确保渗透测试中的所有步骤都有条不紊，并按照正确顺序进行，以产生最佳、最全面的结果。

1.1　渗透测试执行标准

渗透测试执行标准(Penetration Testing Execution Standard，PTES)定义了一个 7 阶段的模型，该模型不仅定义了一种方法及其相关步骤，还包括每个阶段所使用的工具。

PTES 致力于渗透测试执行过程的标准化。PTES 包括以下 7 个阶段。

- **阶段 1：前期交互**——此阶段包括首次利益干系人会议，目的是定义渗透测试范围、ROE 以及资料收集和审查。
- **阶段 2：情报收集**——在此阶段，你将执行被动和主动侦察，还包括服务和应用程序的遍历以及评估对象(Target of Evaluation，TOE)的信息收集。
- **阶段 3：威胁建模**——在这个阶段，你将建立资产和攻击者之间的二分法关系模型(威胁代理/社区分析)。
- **阶段 4：漏洞分析**——在这个阶段，你将通过被动分析(如

审查源代码和阅读说明书)或主动分析(使用工具和人工测试)来识别系统和应用中的缺陷。

- **阶段 5：漏洞利用**——在这个阶段，你将绕过安全控制和/或利用前面的漏洞分析阶段中确定的漏洞，来建立对 TOE 的访问。
- **阶段 6：后渗透**——在这个阶段，你将通过创建的后门通道建立对 TOE 的持久访问，并确定车载网络中各系统之间的网络拓扑结构以便进行内网渗透。
- **阶段 7：报告**——这个阶段与前几个阶段一样重要，甚至更重要。报告是你作为测试人员向利益干系人传达风险信息的地方。最后，利益干系人关心的不是你所使用的是什么零日漏洞，他们更关心超出了可接受范围的，与业务、安全相关的风险，以及如何能清晰、简明地向其他利益干系人传达这些风险。

图 1-1 显示了 PTES 的阶段 2~6。

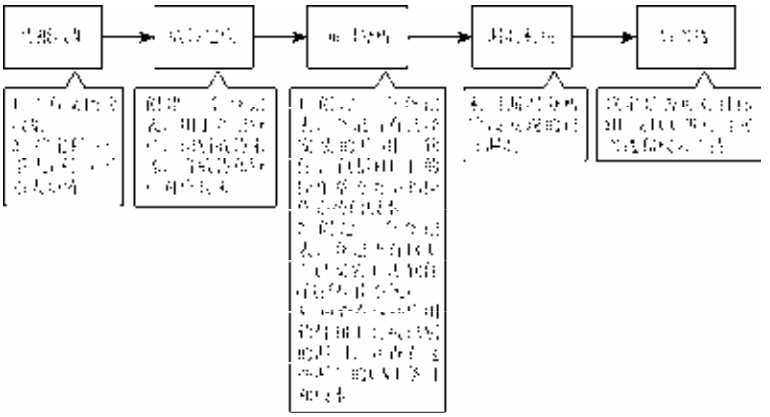


图 1-1　渗透测试执行标准及 TCU/HU 渗透测试中的相关任务

注意：可在项目主页上找到更多关于 PTES 的信息，网址为 http:/www.pentest-standard.org。

渗透测试的前期准备是本章的主题，也是 PTES 框架中的第一个阶段。前期交互阶段的重点是与 TOE 干系人进行一对一讨论，以确定渗透测试的边界，并确保与所有已经确定的关键干系人沟通过。测试规则(ROE)明确规定渗透测试团队在测试过程中哪些事允许做、哪些事禁止做；如果被要求进行白盒式渗透测试，则要从干系人那里拿到所有工程文档和源代码并加以综合利用。

前期交互阶段的重中之重是了解客户的预期结果或交付成果。这通常不会在原始设备制造商(Original Equipment Manufacturer，OEM)和汽车制造商之间的招标文件(Request for Proposal，RFP)中进行阐述。一般情况下，汽车制造商会在 TheStreet 网站发布 RFP，并为汽车内的特定系统征询方案，包括信息娱乐系统和 TCU，这些都是 OEM 投标的一部分。在授予合同前，汽车制造商的 RFP 越来越多地将渗透测试作为考察投标方的硬性要求之一。

很多情况下，OEM 和汽车制造商之间的 RFP 中已经预先明确列出对交付物的要求，RFP 详细描述了汽车制造商希望在最终报告中看到的内容。要确保渗透测试的预期成果输出符合最终报告的模板，以免因未能达到汽车制造商的既定目标而遭到丢弃。虽然你可能在为 OEM 进行渗透测试，但实际上，渗透测试报告最终是提供给汽车制造商的。因此，很多情况下，OEM 让我直接向汽车制造商提交测试结果。但要记住，你的客户是 OEM，而不是汽车制造商，当然，除非汽车制造商雇用了你。

1.2 范围定义

渗透测试的成功完成，最重要的是确保在整个项目中测试范围是受控的。例如，在对车载主机(Head Unit，HU)的渗透测试中范围定义是很重要的，以免耗费大量时间去测试与 TCU 相关的漏洞。超

出范围(扩大范围)是很常见的，最终会让渗透测试人员付出更多时间和金钱。许多情况下，这样做费力不讨好，客户对你并不满意，因为你测试出的这些漏洞影响的是另一个业务部门，对客户本身并不重要。

在许多机构中，计算机信息通信部门是一个与负责 HU 的部门完全分开的业务部门。如果没有适当地划分范围，那么大部分工作(甚至全部工作)都可能不在你所负责的部门的范围内。

在本书的配套网站 www.wiley.com/go/hackingcars 上，可免费下载 HU 和 TCU 渗透测试的范围界定模板。

本节讨论了在确定渗透测试的范围时，需要弄清楚的最重要细节。

1.2.1　架构

目标系统的架构是什么？了解底层的嵌入式操作系统(Operating System，OS)是至关重要的，尤其是涉及影响特定平台和版本的漏洞分析时。例如，它是不是 NVIDIA Linux？是不是 Android？如果是，是哪个版本？是否对内核进行了修改？使用的是哪种微处理器？如果你想修改固件并刷新 TOE，以编译出正确的二进制代码，创建永久连接的后门通道，那么当你在线访问固件时，上述问题都很重要。例如，当应该使用 ELF 格式的 Meterpreter 攻击载荷(payload)时，使用 Python 脚本格式的攻击载荷是无效的。

1.2.2　完整信息披露

与利益干系人一起确定你将有多大机会接触到工程文档、源代码等。源代码的获取会有一定难度，因为与传统的渗透测试不同，TCU 和 HU 的整个源码库通常混合了不同供应商提供的底层驱动代码或库文件。供应链中的其他供应商很少会在渗透测试中与你合作并提供这部分源代码。但反编译器(如 IDA Pro)或其他类型的二进制分析反编译器，是非常有用的工具，可进行静态代码分析和逆向工

程。访问工程文档也可获得非常多的信息。

1.2.3 版本周期

软件发布生命周期描述了应用程序从最初开发到最终发布的整个过程。

在我参加过的很多渗透测试中，我发现很多安全控制措施(如沙盒、CGROUPS 或防火墙规则)都还没有在我测试的版本中实施。在整个渗透测试过程中，很多时候会有新版本提供给你。要注意当前测试的是什么版本；当前测试的硬件和软件应用应该是最新稳定版本。确认问题后，在编写最终报告时适当提及发现问题的具体版本，以防问题在新版本中已经被更正。

1.2.4 IP 地址

HU 和 TCU 容易受到许多潜在的中间人(Man-In-The-Middle，MITM)攻击。如果在 HU 和 TCU 之间使用一个隐藏的无线网络进行通信，通常会分配给 TCU 和 HU 的无线网口静态 IP 地址。虽然有很多方法可识别正在使用的 IP 地址，但若询问客户以提前知道这些 IP 地址是什么，可极大地缩短测试时间。

1.2.5 源代码

你要确定客户是否会向你提供源代码。虽然 OEM 或汽车制造商可能只能提供他们自己的应用程序的源代码，但这值得你去问一下。因为有了源代码，你就能进行静态和动态代码分析。如果客户没有提供源代码，你可使用 gdb、BARF 或 IDA Pro 等工具对二进制文件进行逆向工程。

1.2.6　无线网络

你要确定哪些无线网络已经启用。许多 HU 会充当无线网络接入点，以将 Internet 连接分配给其他控制器。但通常情况下，会由 TCU 充当无线客户端。在高性能的 HU 中，会有两个无线网络接口卡(Network Interface Cards，NIC)：一个作为 WiFi 接入点供车内的乘客使用，另一个则作为隐藏的无线通信设备用于 TCU 连接的网络。如果这是一个白盒渗透测试，你要向客户了解所有正在运行的无线网络的 SSID，同时要查询其他隐藏无线网络的 SSID 以及每个网口卡的 IEEE MAC 地址。在黑盒或灰盒渗透测试下要自己去搜索无线网络的 SSID；第 4 章将演示如何自己搜索无线网络的 SSID，而不需要客户提供。

1.2.7　开始日期和结束日期

务必明确规定渗透测试的开始和结束日期，以免项目的结束时间远远超过你预期的完成日期。如果没有指定结束日期，利益干系人可能会多次驳回，而且如果你作为外包顾问承包了这项工作，随着客户不断地驳回并要求重新测试或变更，最终的付款日期可能会持续被推后。

1.2.8　硬件唯一序列号

几乎在我所做的每一次渗透测试中，我的测试平台范围内都有多个 HU 和 TCU。如果你在现场进行渗透测试，在同一办公室或建筑内常有其他开发人员或工程师在对类似的 HU 或 TCU 进行测试。我经常看到一排排的办公桌上摆放着一模一样的微电子工作台，供我进行测试使用。为正确地识别这些被测单元，我建议注意一下国际移动用户号码(International Mobile Subscriber Number，IMSI)、网卡的 MAC 地址、国际移动设备标识(International Mobile Equipment

Identity，IMEI)以及你正在测试的硬件的所有其他重要唯一标识符。我无法告诉你我有多少次空欢喜，因为我以为我的目标 TCU 连上了我布置的 Evil Twin，结果却发现它不是我的微电子工作台中的 TCU(#Mondays)。

1.3 测试规则

测试范围定义了测试边界，而测试规则(Rules of Engagement，ROE)规定了如何进行这项工作。用军事术语来说，ROE 规定了对使用武力和使用某些军事能力的授权和/或限制。虽然测试规则没有明确规定如何实现结果，但它明确规定了哪些措施是不可实施的。

可从多个有利的切入点进行 HU 和 TCU 的渗透测试。HU 或 TCU 被放在微电子工作台上时，通常处于开发模式，而不是量产模式。因此可访问 Android 调试器(ADB)甚至是以太网端口之类的服务，这些服务在处于量产模式的车辆中可能是不可用的。例如：我曾参加过多次渗透测试，其中以太网端口在微电子工作台中是可以接触到的，但安装在量产型号上时被焊接封闭，就不可访问了。ROE 阶段需要确定在定义范围后如何进行测试。例如，是否允许 hook 以太网端口来测试其他系统级的控制，以发现攻击者可能会利用到的系统漏洞？如果攻击者进入系统中，会有哪些功能可被利用？所以请记住，成功的渗透测试的定义并不是简单地从公共网络获取 root 权限。说来有趣，我在开发模式下在 HU 上建立了一个 shell(在量产模式下是关闭的)之后，发现了一个高危漏洞，这个漏洞比我在公共无线接口或 GSM 中发现的高危漏洞更令 OEM 担心。

在制定 ROE 时，有必要与利益干系人制定协议，确定哪些有利的切入点允许测试，以及测试开始后能接受/不接受的测试步骤(攻击链上哪些步骤)。

另外，你可从本书的配套网站上下载 ROE 模板示例。

1.3.1　时间表

项目成功的关键是确保时间范围明确规定了项目开始和结束的日期。汽车制造商对 OEM 的时间要求非常严格，没有回旋余地，生产线和组装车间的进度表已经明确规定了时间进度。

车载 ECU 的 OEM 厂商所遵循的软件开发过程受制于它专有的规范以及若干国际标准。其中相关度最高以及最重要的标准是汽车 SPICE(Software Process Improvement and Capability dEtermination)、J3061 和 ISO 26262。

作为渗透测试人员，你若在软件发布至生产的过程中发现并公开漏洞，很容易造成软件发布周期的严重延迟，导致错过最后期限。所以，尽管传统渗透测试中漏洞仅在最终报告提交时才会公布，汽车渗透测试中发现的漏洞要即时公布，以便开发人员在发现高危漏洞时马上进行补救。

1.3.2　驻场测试

能否成为一名成功的顾问在很大程度上取决于你是否愿意到客户所在地进行驻场测试。我见过很多大大小小的公司都不愿意在客户的场所进行现场测试，结果他们仅因为想远程工作而失去了合同。CPV 的渗透测试(3~6 个月)当然比传统的网络渗透测试要长得多。

现场工作需求对这个行业来说是一个巨大挑战，不仅要努力适应对汽车内部连接有要求的现代消费者，还要适应千禧一代劳动力的不断变化，因为年轻人希望远程工作，而不是朝九晚五的办公室工作。

因此，你是否愿意去主要的 OEM 和汽车制造商进行现场测试(很可能是到欧洲和亚洲)将决定你在这个领域的成功与否。几乎没有 OEM 会因为你想在家工作，就把整个微电子工作台运到你家。别想远程访问他们的网络，因为那通常是封闭的，很少有人能获得批准。

1.4　工作分解结构

工作分解结构(Work Breakdown Structure，WBS)以分层结构图的形式说明了一个项目中的任务与负责任务交付的相关团队成员之间的关系。将 WBS 看成把整体工作任务分解成可管理的、可交付的任务，并说明各项任务各由谁来负责的结构图。

在 CPV 测试中，仅由一个人执行渗透测试的情况是非常少见的。通常情况下，由一个团队进行测试，通过分配资源来确定谁做什么，这样就可确定角色、责任、问责和权限。在时间跨度较长的渗透测试中，我们甚至在 Excel 表格中创建了一个 RACI 图表(见图 1-2)。

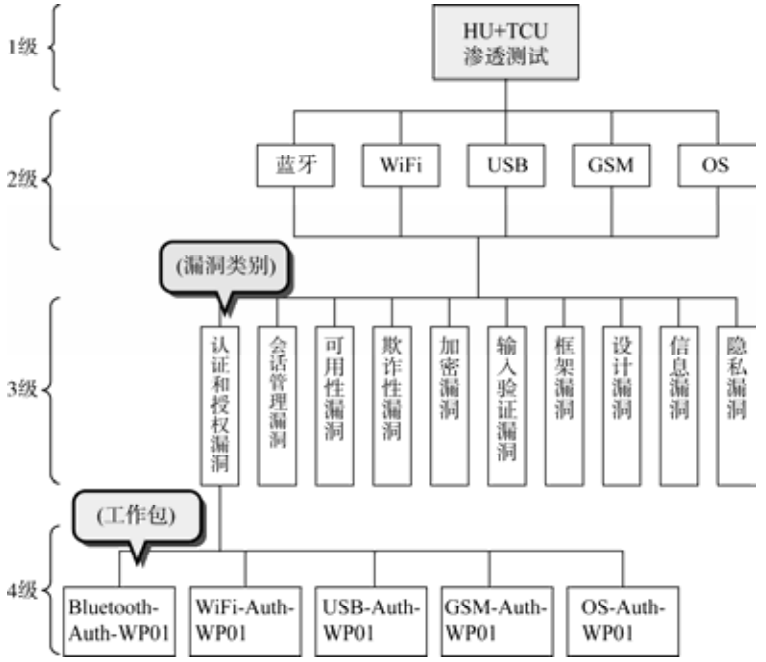


图 1-2　HU+TCU 渗透测试工作包的工作分解结构

虽然这一切都可由一个人完成，但我从未见过精通所有这些攻击面的渗透测试人员，这也证明了一个古老的公理：“同时执行多个任务有可能同时搞砸多件事情。”

1.5　文档收集和审查

从传统的网络渗透测试到 CPV 的渗透测试，其中有一点是我没有预料到的，那就是后者的工程文档比前者多得多。在进行网络渗透测试时，我大概一只手就可数出我收到的准确的和定期更新的网络图的数量。为保持 PCI 合规性，这些企业必须通过年度合格安全评估师(Qualified Security Assessor，QSA)审核。PCI-DSS 审核要求提供网络和应用程序流程图。你现在要做好心理准备，假如要对 TCU 或 HU 进行渗透测试，你将收到大量工程文档。然而，为进行一次彻底的渗透测试或风险评估，所有这些都是必要的。

文档示例

你应该收到的文件包括但不限于以下内容：

- 自定义协议和消息的定义，例如用于汽车制造商后端 OTA 更新的协议和消息。
- 功能列表。
- 概要设计(High-Level Design，HLD)文档。
- 以往的风险评估报告。
- 以往的渗透测试报告。
- IP 架构。
- 固件文件(第三方)。
- 用于 CAN 诊断的收发矩阵。
- 多媒体板、主板、国标板(Country-Specific Board，CSB)等的示意图。

每次会议都可要求利益干系人提供不同的可用文件。然而，如果因为你不询问而导致客户没有提供已有文件，那么除了你自己，你不能责怪任何人。因为文档可能分散在不同的文档所有者手里或文档控制管理(Document Control Management，DCM)系统中，所以在前期准备阶段就应该尽早请求提供文档，而不是在渗透测试开始后才要求提供。这样测试开始后，你和利益干系人可省去很多奔波的时间。在到达现场执行渗透测试工作之前，你也有时间查看所有文档并突出标记在测试期间可能有用的重要信息。

像往常一样，你需要确保所用的文档是最新版本。

1.6　项目管理

根据项目管理协会(Project Management Institute，PMI)的说法，一个项目是阶段性的，有始有终，有一个明确界定的范围，需要确定在这个范围内需要哪些资源来完成。

对于渗透测试项目来说，尤其重要的是确保项目得到专业的管理，以便按时、按预算交付客户需要的结果。所以项目管理是将知识、技能、工具和技术应用到渗透测试或风险评估活动的不同阶段，这是项目成功的必要条件。

虽然我知道“项目管理”这个词可能会让人感到压力很大，你的直觉反应是跳过这一节，但任何渗透测试项目或风险评估都不应该在没有项目管理的情况下开始。要明确的是，这意味着，项目任务书和项目进度表应在整个项目生命周期内建立和监控。我们在之前的渗透测试中使用过的这两种文档的示例也可从本书的配套网站下载。

本节将介绍每个项目阶段的要素，它适用于管理 HU 或 TCU 渗透测试，并说明每个项目阶段的主要成果是什么。

任何正确规划和管理的项目都包括以下要素：①必须完成哪些

工作；②必须产生和审核哪些交付品；③谁必须参与；④如何控制和批准每个阶段。这些要素为渗透测试提供一个系统的、及时的、可控的过程，以保障整个渗透测试成功，最终使项目的利益干系人受益。

表 1-1 根据 PMBOK 映射到 PTES 模型的各个阶段列出了项目的 5 个阶段。

表 1-1 项目阶段

PMBOK 阶段	PTES 阶段	活动
构思和发起		项目任务书
		项目启动
定义和规划	前期准备	项目范围和预算
	情报收集	项目工作结构分解
	威胁建模	甘特图
		参与人员沟通计划
		项目风险管理
启动或执行	漏洞分析	状态和跟踪
	漏洞利用	KPI指标
	后渗透	质量
		预测
绩效/监督		考核目标
		高质量交付物
		工作量和成本跟踪情况
		业绩
项目完结	工作汇报	项目总结
		项目竣工查核清单
		工作汇报

在本节中，我们将为一家亚洲的信息娱乐系统和 TCU 制造商进行一次假想的渗透测试，我们称这个制造商为 AsiaOEM。该公司已

经赢得了为一家亚洲大型汽车制造商设计和制造 HU 和 TCU 的 RFP，我们将大型汽车制造商称为 AsiaCar。

AsiaOEM 的首席信息安全官(Chief Information Security Officer，CISO)必须满足从 AsiaCar 得到的 RFP 中的信息安全要求，所以她请你对这两个产品进行渗透测试。

中标项目的 RFP 明确要求报价人对产品进行渗透测试，并在交付时提交一份报告，说明测试后发现并修复的漏洞。CISO 已经派出了她的项目管理办公室(Program Management Office，PMO)，由其指派一名项目经理负责项目，项目经理首先在项目的构思和发起阶段编写了一个项目构思文件，确定项目的总体目的、时间表和预算。

1.6.1 构思和发起

构思和发起阶段标志着项目的开始，目的是在广泛的层面对项目进行定义，并将商业案例提交给高级管理层审查和批准。

1. 范围

项目范围文档对项目的成功至关重要，因为它确定了项目的参数，定义了项目的目标、可交付成果、任务和其他细节，确保所有成员都了解自己在项目团队中的角色和责任。

范围声明还为渗透测试团队提供了在项目期间对变更请求做出决策的指南。这里要回答的问题包括：渗透测试的实际范围是什么？如果是HU的渗透测试，TCU是否也在范围内？后端远程服务器是否在范围内？如果TCU在范围内，团队应确保发现的影响TCU的漏洞被正确记录，因为很可能会有一个单独的小组负责修复。是否提供shell访问权限，允许进行本地系统测试？目标设备是在开发模式下还是在量产模式下被锁定？所有这些东西都需要确定范围，包括无线网卡、蓝牙或串行数据(如CAN总线、有线以太网、LIN子总线)

接口是否也在测试范围内。

2. 利益干系人

项目的利益干系人可以是受到结果影响的个人或组织，无论其受到的影响是积极的还是消极的。

这里，你要确定积极的和消极的利益干系人、执行发起人和项目发起人。你要记录在不同业务领域中的联系人，无论是你正为其执行渗透测试的部门，还是其他可能参与支持测试的部门。务必记下所有利益干系人的姓名、电子邮件地址和电话号码，以便在整个测试过程中联系，并适当地标记出积极的或消极的利益干系人。这实际上会成为你的利益干系人矩阵。

项目利益干系人将进行尽职调查，以决定项目是否合理，如果所有标星号的都符合要求，将授予你一份已签署的工作声明(Statement of Work，SOW)。理想情况下，客户会向你提供项目任务书或项目启动文件，概述项目的目的和要求。你可能会看到商业需求、涉及的利益干系人和商业案例(很可能是 OEM 和汽车制造商之间的 RFP 要求)。

需要注意，执行这个阶段的是客户，而不是作为渗透测试者的你。相反，你可能收到这个阶段的输出物。

项目构思文件是关于拟议项目的文件，通常包括可行性研究(如技术或财务)、详细的设计图纸、计划和说明、项目成本的详细估算等。

必须注意的是，项目构思文件通常是由 RFP 中的要求驱使的，即汽车制造商要求 OEM 必须完成的或者需要在最终交付物中实现的结果——例如，向负责设计和制造 HU 的 OEM 发出的 RFP。汽车制造商越来越多地要求 OEM 厂商在生产前进行渗透测试和风险评估。RFP 中的这些要求通常会推动项目构思阶段，也解释了你的工作职责。

下面显示了 AsiaOEM 公司的项目构思文件，用于 HU 和 TCU 的渗透测试。虽然这通常是公司内部两个独立的项目，但为了简洁

起见，我们将把它们合并在一起。该模板可从本书的配套网站下载。

项目构思

HU 和 TCU 渗透测试

编制	Jane Doe，项目经理	**邮件**	Jane.doe@eurocorp.de
部门	信息娱乐系统和远程信息处理	**电话**	+49(0)111 222 3344

项目发起人*(为项目提供资金或资源的人)*

Hans Doe - EVP Connected Car Hans.doe@asiaoem.co.jp +49(111)222-3333	

客户(决定项目启动，确定目标是否达到，接收项目交付物的人。他决定项目是否成功)

Jill Doe - EVP Automaker Jill.doe@asiacar.co.jp (111)222-3333	John Doe - EVP Automaker John.doe@asiacar.co.jp (111)222-3333

项目说明*(项目的总体目标)*

项目说明应简明扼要地描述项目的总体目标。该说明应该明确项目的总体目标、时间框架和资源参数，并应包含以下格式：行动、最终结果、项目完成日期、预计工时。

例如：“对 TCU 和 HU 进行渗透测试和风险评估，并在 2019 年 11 月 1 日之前提交报告，费用不超过 275 000.00 欧元。”

背景*(与项目需求相关的简要背景)*

AsiaOEM 获得了 AsiaCar 的一个项目，为其 2020 年新款智能网联汽车设计和制造 HU 和 TCU。根据 RFP 第 13.2 条的要求，EuroCorp 必须在量产前对 HU 和 TCU 进行渗透测试和风险评估，并向汽车制造商提供最终报告。渗透测试和风险评估的需求已得到明确的定义，并将被提供给执行该任务的供应

(续表)

商。Jane Doe 已被指定为项目经理，她将与选定的供应商一起管理测试的范围，确保最终报告符合 AsiaCar 的要求。

问题的定义*(需要解决的事项)*

为满足 RFP 的要求，AsiaOEM 必须聘请外部顾问为 AsiaCar 进行 HU 和 TCU 的渗透测试。

必须对 TCU 和 HU 的所有通信接口进行“白盒”渗透测试，包括蓝牙、WiFi、串行数据接口和 GSM 接口。此外，还必须对系统上运行的证书交换、加密、ADB shell 操作系统以及二进制文件进行渗透测试。

必须对 HU 进行风险评估，必须记录系统中的所有资产。风险评估应确定信任边界、数据流、入口点和特权代码，以记录系统的潜在漏洞。确定威胁后，还要确定应对每个威胁的解决方案和实施方法。最终结果必须是包含所有已识别的威胁以及相应的解决方案和实施方法的文件。文件简述中必须包括确定潜在安全风险及其对系统造成破坏的风险评估，以及完整的资产清单和记录潜在攻击的威胁模型。

期望状态*(项目顺利完成后的情况)*

描述项目完成后情况会有何不同。这是你想通过解决问题以达到的状态。通常情况下，期望的状态是现有问题已被解决。

- AsiaOEM 产品系列的 HU 和 TCU 将拥有每个系统的完整资产登记册，以及与每项资产相关的风险评级和相应的风险处理计划。
- AsiaOEM 将在其系统开发生命周期(SDLC)中实施安全保护，为客户生产更安全的产品，同时增加消费者对 AsiaOEM 品牌的信任。
- 提前(在公司以外的人发现漏洞之前)发现产品中的漏洞，因为这些漏洞若被公开，可能造成负面影响——削弱股东和消费者对 AsiaOEM 品牌的信任。
- AsiaOEM 将与客户一起定义企业可接受的风险水平，使其能更快速地识别出哪些风险是企业不能接受的，从而处理这些风险。务必采取基于风险评估的方法来处理已确定的漏洞和风险。

(续表)

- EuroCorp 将更好地了解其 TCU 和 HU 产品的潜在攻击面，以及可实施哪些主要和次要控制措施来减少攻击面并处理相关风险。

项目的目标*(要达到的目标或条件)*

项目目标(AKA 标准、目标等)是指描述项目完成时需要达到的指标(数量和/或质量)的说明。目标应该是具体的、可衡量的、约定好的、可实现的、有时间限制的(SMART)，并应反映出关键领域的重要性和价值(如收入利润率、客户满意度等)。

- AsiaOEM 将为 AsiaCar 制作一份最终报告，其中包含在渗透测试中发现的所有漏洞的清单。
- AsiaOEM 将编制一份风险评估和风险处理表，其中包含资产和基于场景的风险评估结果，如果发现的风险在评估后超过了企业可接受的风险水平，将有一个相关的风险处理计划。
- 顾问将在预算范围内按时编制渗透测试和风险评估的最终报告。
- 消费者会对 AsiaOEM 品牌充满信心，因为 AsiaOEM 非常重视车辆中使用其产品的人的安全保障。
- AsiaCar 将有机会审查已识别的风险，并与 AsiaOEM 一起决定哪些风险是企业无法接受的。

预期成果和结果*(优势和后果)*

描述贵公司在成功完成项目后可能获得的收益(改进)。无论这些结果是正面还是负面的，都应该具体说明一下。涵盖结果的衡量标准。衡量标准可记录为要实现的目标，在基线上的量化改进，或与基准目标作比较。

- 通过与顾问的合作，软件开发人员和硬件工程师将获得信息安全意识，这将有助于将网络安全意识植入 AsiaOEM 的文化中。
- AsiaOEM 将为日益关注网联汽车的网络安全问题的消费者们带来更安全的产品，以增加客户对 AsiaOEM 品牌产品的信心，从而提高市场占有率。
- 汽车制造商将对 AsiaOEM 品牌产品安全性更有信心，从而增加 AsiaOEM 产品在市场上的安装量。

(续表)

- AsiaOEM 将满足或超出产品的网络安全测试的最低要求，以便在本招标书中交付。

事实*(优势和后果)*

逐项列出对项目具有消极或积极影响的事实情况。项目要求必须基于事实，项目成功的必要条件应作为事实列在这里。

- 由于 TCU 在欧洲以外地区的 GSM 连通性的限制，以及将微电子工作台运往异地的物流问题，顾问必须能在 AsiaOEM 设备现场工作。
- 在过去 5 年中，许多关于信息娱乐系统和 TCU 的脆弱性和风险的研究已经公开了。
- 最近，微电子工作台出现了供应短缺，可能给顾问带来问题。
- 软件的最新 R500 版本尚未完成。需要在测试过程中向顾问提供多个版本，以实现待测控制。
- AsiaOEM 办公室需要在有限空间中预留一个会议室，持续时间为 3 个月，需要高级管理层的批准才能在测试期间为顾问提供座位。
- 由于时间紧迫，顾问现在必须收到投标书，以给测试分配足够的时间。
- 从内部利益干系人那里得到的信息是有限的，这可能影响到顾问开展工作或访谈的时间安排。

假设*(认为是真实情况，但没有记录的证据)*

列出能使项目顺利完成的基础假设。这些假设不应是使项目成功或失败的问题，而应是你确定项目能完成的前提条件，在这些假设被证实并成为事实之前，可暂时使用这些假设进行测试。

- 在确定渗透测试服务团队之前，AsiaOEM 至少会收到三家专门从事网联汽车渗透测试的 IT 安全公司的投标，供其考虑。
- 至少给顾问预留 4~6 个月的时间，以便在交付期限内对已确定的漏洞进行测试和修复，并处理风险。
- 拨给这个项目的预算应满足顾问收取的费用。
- 内部利益干系人应预留时间来配合顾问的工作，以推进项目的成功。

(续表)

预期的资金来源和预算限制

确定预期的资金来源，并提供尽可能高的项目预算。就项目的优点提供简要的成本/效益分析。

- 该项目的资金将由车载信息服务部门和信息娱乐部门各承担132 000 欧元。

受影响的部门

列出受本项目影响的部门，并简要说明这些部门的参与情况或受到的影响。

- 车载信息服务部门：该团队将负责更新与 TCU 或汽车制造商后端 OTA 相关的漏洞和风险。
- 信息娱乐部门：该团队将负责处理与 HU 相关的漏洞和风险。

修订和批准历史

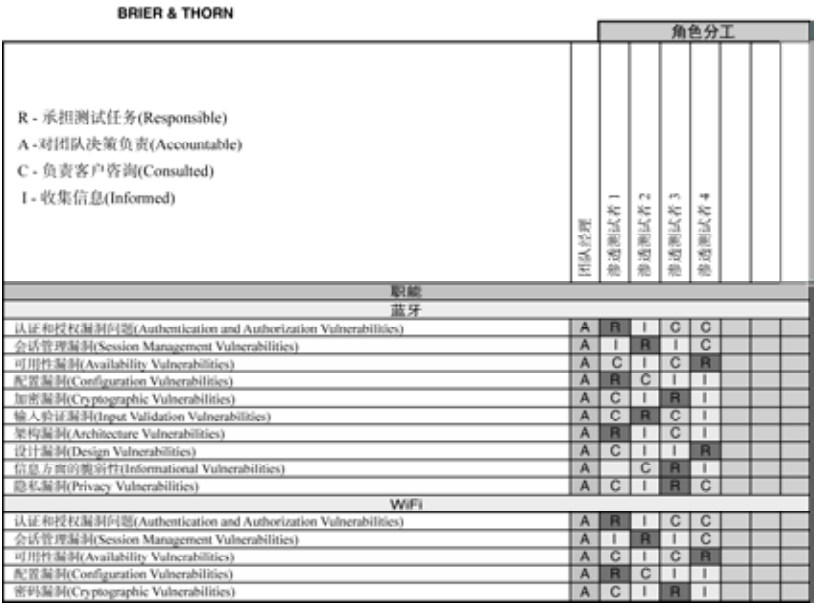
日期	版本	修订者	描述	发起人/利益干系人接受日期

1.6.2 定义和规划

在定义和规划阶段，要定义项目的范围和项目管理计划(如前文所述)。这包括确定渗透测试所涉及的成本、质量、可用资源和实际的时间表。在这个阶段，应该为测试团队的每个人分配角色和任务，确保他们都知道自己的角色、责任和义务。甚至可考虑创建一个 RACI 图表(见图 1-3)。本书的配套网站上还提供了 RACI 图的示例。

在这一阶段，项目经理将创建以下项目文件。

- **范围声明：**明确界定业务需求、项目的收益、目标、可交付的成果和关键进度。
- **工作分解结构：**这是一个可视化的表示方式，将项目的范围分解成可管理的小部分，供测试团队安排分工。
- **里程碑：**确定甘特图中包括了整个项目需要实现的高级别目标。



BRIER & THORN

R - 承担测试任务(Responsible) A - 对团队决策负责(Accountable) C - 负责客户咨询(Consulted) I - 收集信息(Informed)	团队内测	渗透测试人员1	渗透测试人员2	渗透测试人员3	渗透测试人员4			
	角色分工							
职能								
蓝牙								
认证和授权漏洞问题(Authentication and Authorization Vulnerabilities)	A	R	I	C	C			
会话管理漏洞(Session Management Vulnerabilities)	A	I	R	I	C			
可用性漏洞(Availability Vulnerabilities)	A	C	I	C	R			
配置漏洞(Configuration Vulnerabilities)	A	R	C	I	I			
加密漏洞(Cryptographic Vulnerabilities)	A	C	I	R	I			
输入验证漏洞(Input Validation Vulnerabilities)	A	C	R	C	I			
架构漏洞(Architecture Vulnerabilities)	A	R	I	C	I			
设计漏洞(Design Vulnerabilities)	A	C	I	I	R			
信息方面的脆弱性(Informational Vulnerabilities)	A		C	R	I			
隐私漏洞(Privacy Vulnerabilities)	A	C	I	R	C			
WiFi								
认证和授权漏洞问题(Authentication and Authorization Vulnerabilities)	A	R	I	C	C			
会话管理漏洞(Session Management Vulnerabilities)	A	I	R	I	C			
可用性漏洞(Availability Vulnerabilities)	A	C	I	C	R			
配置漏洞(Configuration Vulnerabilities)	A	R	C	I	I			
密码漏洞(Cryptographic Vulnerabilities)	A	C	I	R	I			

图 1-3　RACI 图样本

- **甘特图：**可视化的时间线，可用来规划任务或可视化项目进度表。
- **风险管理计划：**识别所有可预见的项目风险，常见的风险包括不切实际的时间安排、成本估算、客户审查周期、新软件发布测试，以及由于微电子工作台无法使用、系统强化(打补丁、升级)、缺乏承诺的资源等问题导致的项目启动时间推迟。

渗透测试将产生以下可交付成果。

- **威胁模型：**由影响目标的威胁因素创建的威胁模型。
- **工程资料：**在信息收集阶段收集到的所有工程资料。
- **工作分解结构：**WBS 应定义渗透测试中要执行的所有任务，还应包括尽可能细分的可交付成果/工作包，并且应该像前面描述的那样进行分级。工作分解结构图的示例可从本书

的配套网站下载。

1.6.3　启动或执行

在启动或执行阶段，相关工作开始展开并形成可交付成果。这个阶段通常是项目的核心，是投入实战的重要时刻，通常包括定期召开项目进度会议(建议每周一次)，团队成员执行 WBS 中的任务，对新的软件版本再次进行测试(因为它们被强化了)，并在添加新的防火墙规则时进行复测。

本阶段的项目交付成果如下。

- **会议记录：**包含每次会议的情况说明和决策记录。
- **对项目进度表进行例行更新：**请记住，项目进度表是一份可修改的文件。虽然主要的里程碑式的日期很可能不会改变，但在最后一刻想到的新测试或测试过程中发现的其他攻击路径常被添加到项目进度表或 WBS 中。
- **与利益干系人的沟通：**这对项目的成功很重要。在发现漏洞时，必须及时将发现传达给工程团队，尤其是接近项目结束时。不要太晚与利益干系人沟通这些漏洞，以免他们在距离截止生产期限太近时要求修复漏洞，这可能会造成与修复漏洞相关的 bug。保证你在项目定义和计划阶段的前期准备步骤中，明确规定了测试过程中应该什么时间披露以及每隔多久披露一次发现的漏洞。

在执行阶段，以下可交付物是渗透测试的成果。

- **脆弱性：**这是一份漏洞列表，包含在应用程序版本、操作系统以及运行的服务(甚至是专有服务)中识别的所有漏洞。通过协议模糊测试和使用反编译器(如 IDA Pro)或车辆网络工具(如 Vehicle Spy)对二进制文件进行逆向挖掘，可确定专有协议和服务的漏洞。

- **截图：**我不知道有多少次看到渗透测试人员因为忘了截图而无法给出漏洞利用或成功利用漏洞后的证据。你应该通过截图收集尽可能多的证据，因为这些截图能很好地呈现在最终报告中。

1.6.4　绩效/监督

绩效/监督阶段确保项目的进展和执行情况符合预期。应从始至终持续监督项目的进展，并定期(建议每周)与利益干系人举行会议。这个阶段对项目的成功至关重要，因为它让你有机会展示风险评估的结果，以确保项目正在按计划进行，并达到预期目标。至于这一阶段的输出物，渗透测试员将提供最新的漏洞发现，而利益干系人将提供当前发布计划变更的最新信息，以及需要重新测试的新版本的信息。此外，务必让监督整个项目的项目经理感到任务更新是定期沟通的，以保证项目进度的更新。这也可确保你快速发现和减轻测试范围的变化。

一个常见问题是，因为没有跟踪会议记录而无法确保在下一周的会议中继续跟进正在执行的项目。目前有几种不同的会议记录模板，以及一些新的云应用，如 MeetingSense.com 等。我的建议是，你可尝试使用不同的平台将会议记录集中存储下来，甚至还可采用一个项目管理平台。我建议采用基于云的平台，因为现在的项目团队是分散的，分布在不同地区。

另一个建议是务必采用云存储服务，如 box.com 或 dropbox.com (最好是能实现数据静态加密的平台)，因为存储在这些文件夹中的文档是高度敏感的。通过使用基于云存储的服务，你能向客户和/或利益干系人提供远程登录的权利，以便他们将工程文档和其他文件上传到所有项目团队成员共享的云盘中。

1.6.5 项目完结

每个管理得当的项目都有一个明确的开始和结束日期。项目的收尾阶段非常重要，因为它确保项目的所有目标都已实现，可交付成果已完成并提交给客户和项目的利益干系人。项目收尾阶段通常包括以 PowerPoint 演示文稿的形式向客户和利益干系人展示渗透测试团队的成果，并附上完整的报告。

在客户审查和认可了之前的草稿后，才提交最后的报告。

1.7 实验室布置

本节将详细介绍渗透测试实验室中应该具备的硬件和软件。从笔记本电脑上运行的操作系统到你要从制造商那里订购的硬件，都将在本节详述。

WiFi Pineapple 是可选择是否购买的，因为 Evil Twin 和其他无线攻击可通过软件单独进行。而 ValueCAN 适配器、Vehicle Spy 和 RTL-SDR 这些硬件是必须购买的。

1.7.1 所需的硬件和软件

要执行 TCU 和 HU 的渗透测试，需要特定的硬件和软件，你的 jump kit 可能已经包含了这些工具。渗透测试实验室所需的硬件将涵盖你的 jump kit 和包含目标硬件的微电子工作台。

1. 硬件

你的 jump kit 应包括以下设备，你可直接从制造商处购买。列出的价格是截至本书撰写时的最新价格。价格和供应的情况可能会随时间发生变化。

WiFi Pineapple Tetra	$200	https://www.wifipineapple.com/pages/tetra
Value CAN 4	$395	https://www.intrepidcs.com/products/vehicle-network-adapters/valuecan-4/
Vehicle Spy3 Pro	$2795	https://www.intrepidcs.com/products/software/vehicle-spy/

2. RTL-SDR 硬件设置

RTL-SDR(Software Defined Radio，软件定义的无线电)是一种物理设备，可作为计算机上的无线电扫描器，用于接收和传送所在区域的无线电信号(具体取决于硬件情况)。SDR 由无线电组件组成，如调制器、解调器和调谐器，这些组件传统上由硬件实现，现在改由软件实现。RTL-SDR 的频率范围各有不同，从 42~2200 MHz(不包括 1100~1250 MHz)的 Eleonics E4000 加密狗到 BladeRF(BladeRF 的频率为 300MHz~3.8GHz)，并能同时发送和接收无线电信号。

BladeRF 2.0 xA4	$480	https://www.nuand.com/blog/product/bladerf-x40/	全双工 300 MHz~3.8 GHz
HackRF One	$317	https://greatscottgadgets.com/hackrf/ 在 Amazon 上也可用	半双工 30 MHz~6 GHz
USRP B210	$1 216		全双工(2x2 MIMO) 70 MHz~6GHz

3. 软件

本节介绍基站收发台(Base Transceiver Station，BTS)的软件 YateBTS，以及其他需要安装在实验室笔记本电脑上的软件。YateBTS 是 GSM、GPRS 无线接入网络的软件实现，可让你自己的基站塔(伪基站)与 TCU 连接，使你可在 TCU 和制造商后端之间禁

用加密和拦截消息。

YateBTS	基站软件	https://yatebts.com/
OpenBTS	基站软件	https://openbts.org
GNU Radio	软件定义的无线电	https://www.gnuradio.org/
Gqrx	软件定义的无线电	http://www.gqrx.dk
HostAPD	802.11 开源无线接入点	https://w1.fi/hostapd

4. 微电子工作台

虽然每个微电子工作台都不一样，但在任何实验室环境中，你都需要使用以下基本组件来连接大多数 HU 和 TCU。

- 汽车/发动机模拟软件(该软件可模拟开启/关闭车辆的发动机，通常由 OEM 提供)
- Vector 1610 CAN 适配器：CAN FD/LIN USB 适配器
- USB hub
- UART USB 转换器(USB 转串口)
- 车载主机(HU)
- 远程信息处理控制单元(TCU)
- 电源供应
- 以太网交换机

在渗透测试开始之前，需要安装“攻击”主机的操作系统，并编译第三方工具或安装相应的软件包来构建你的 jump kit。目前有一些专门为渗透测试人员提供的 Linux 发行版，它们是专门为渗透测试工作而设计的——如 Kali Linux 或 ParrotOS。如果你决定使用 Kali 等发行版，在尝试对本书中列出的一些工具进行源代码编译之前，务必了解预装了哪些库或工具。

例如，如果你已通过软件包管理器安装了 GNU Radio，之后又从源代码安装 GNU Radio，可能造成库路径的问题。此外，有些发行版提供的 GNU Radio 包可能已经过期了，所以需要检查正在安装

的版本是不是最新的。在基于 Ubuntu 的发行版上搜索已安装的软件包(gnuradio)的命令如下。

```
$ apt search gnuradio
```

有时旧版本的 GNU Radio 也会被收入安装包管理器中。你的发行版的版本不应比当前的 GNU Radio 版本老很多，至少应该是相同的次要版本，也就是说，第二位数应该是相同的。

下面将带领你了解你在 jump kit 中所需要的工具的安装和配置方法。如果你想获得更多信息、错误报告或帮助，请直接向软件开发者或工具供应商提出。

图 1-4 描绘了一个物理网络结构图，说明了实验室中的每个组件应该如何连接。

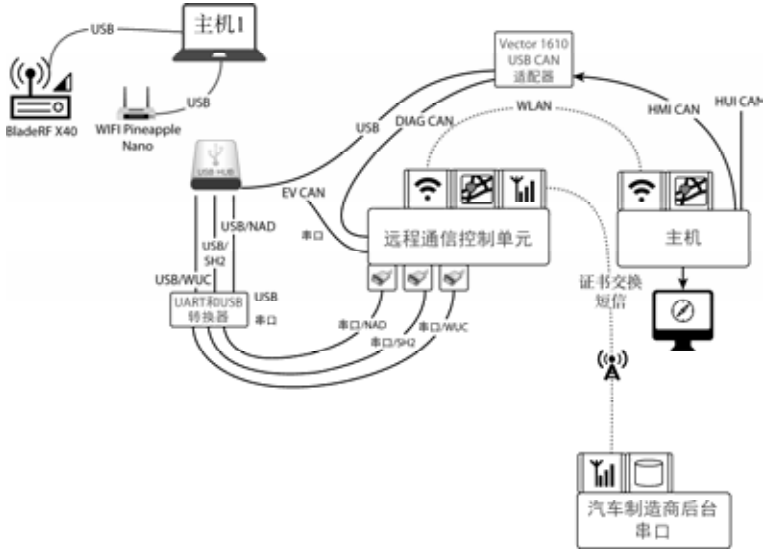


图 1-4　BladeRF 的实验室网络概览

1.7.2　笔记本电脑的设置

在一个独立工作站上打开网络浏览器，然后导航到

http://www.kali.org，这是 Kali Linux 发行版的官方网站(或你喜欢的任何其他发行版)。本书使用的操作系统是 Kali Linux 发行版，截至编写本章时，Kali Linux 发行版的版本为 2018.2。从下载页面下载最新的 ISO 文件。

下载好 ISO 文件后，使用 Linboot(Windows)或 Etcher(Mac)为系统创建一个可引导的闪存镜像。关于安装和设置 Kali 的步骤说明，不在本书的讨论范围内。不过话说回来，既然你正在读本书，我相信你应该不需要安装 Linux 的教程。

安装完 Linux 发行版后，务必运行 apt-get update/upgrade 命令来获取最新版本的软件包和发行版：

```
# apt update ; apt upgrade ; apt dist-upgrade
```

安装好 Kali 后，可下载渗透测试所需要的工具来组建 jump kit，并开始设置第三方设备。

本章的其余部分将罗列下载和安装这些工具的步骤。对于创建伪基站，我提供了两个不同选项。不过，本书后文中关于通过 GSM 黑客入侵 TCU 的章节使用了与 BladeRF 有关的方案 2。

注意：在执行这些步骤前，你有责任了解你所在国家/地区的法律是否禁止搭建伪基站。我和 John Wiley & Sons 公司都不对因执行本书中的步骤而违反当地的联邦通信法律的行为负责。

1.7.3 Rogue BTS 方案 1：OsmocomBB

本节将介绍在我亲爱的朋友兼同事 Solomon Thuo 的研究中，如何使用支持 OsmocomBB 的手机创建一个伪基站。你可在他的博客中找到更多关于建立 OsmocomBB 伪基站的信息，网址是 http://blog.0x7678.com。

要建立一个由 OsmocomBB 驱动的伪基站，需要以下硬件。在这个购物清单中最难购买的商品是支持 OsmocomBB 的手机。我很

幸运地在 eBay 上找到了它们！

- 最新发布的 OsmocomBB：https://www.osmocom.org
- 支持 OsmocomBB 的 GSM 手机：https://osmocom.org/projects/baseband/wiki/Phones
- CP2102 数据线：http://shop.sysmocom.de/products/cp2102-25
- 笔记本电脑 + Linux

OsmocomBB 是一个开源的 GSM 基带软件工具。它的目标是完全取代专有的 GSM 基带软件，可用来创建伪基站。

移动通信系统(Systems for Mobile Communications，SYSMOCOM)的 CP2102 数据线用于连接你的笔记本电脑和手机的 UART 接口，也可用于访问 SIMtrace 来调试 UART。

购买所有必要的硬件设备后，接下来应该开始下载相应软件，配置并连接好这些设备。按照以下步骤在你的笔记本电脑上设置并运行 OsmocomBB。

(1) 确保你没有其他的 USB 线/设备插到你的笔记本电脑上，保证操作系统会给设备驱动程序预留 ttyUSB0。然后用 CP2102 数据线连接手机和笔记本电脑。如果你不确定哪个设备驱动程序被分配给手机，只需要在连接到手机的笔记本电脑上运行以下命令即可：

```
$ dmesg |grep tty
```

(2) 从 OsmocomBB 主页下载 OsmocomBB。

(3) 通过执行以下命令将自定义的 OsmocomBB 固件上传到你的手机：

```
$ sudo ./osmocon -d tr -p /dev/ttyUSB0 -m c123xor –c
../../target/firmware/board/compal_e88/rssi.highram.bin
```

(4) 固件加载完毕后，关闭手机电源。

(5) 在手机关机的情况下，简短地按一下电源键。你的笔记本电脑屏幕看起来应该类似于图 1-5。你现在可开始设置和运行伪基站了。在此之前，一定要给手机充满电，因为电源线会干扰通信。

(6) 将 CP2102 数据线的一端插到手机上，再将另一端插到笔记本电脑上，务必找到 Linux 分配的设备驱动名称，如 ttyUSB0。然后简短地按一下手机上的电源键，加载 OsmocomBB 应用。

(7) 运行 OsmocomBB smqueue 工具：

```
$ cd /rf/public/smqueue/trunk/smquue
$ sudo ./smqueue
```

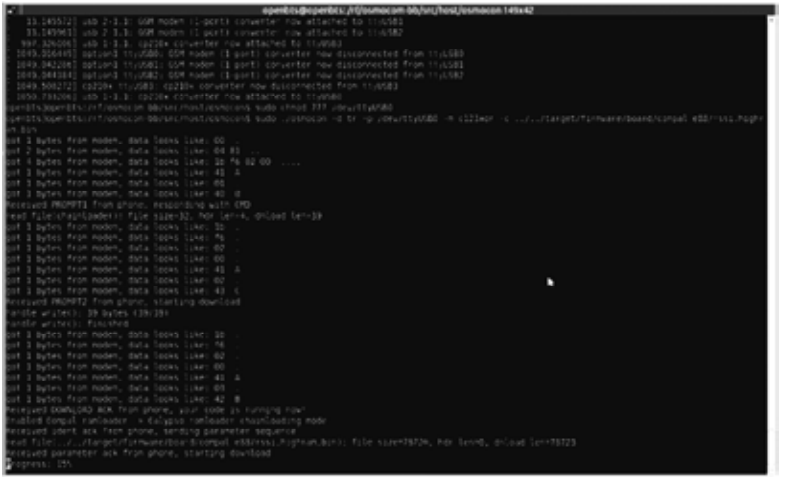


图 1-5　将固件加载到手机上

(8) 运行 OsmocomBB sipauthserve 工具：

```
$ cd /rf/public/subscriberRegistry/trunk
$ sudo ./sipauthserve
```

(9) 启动伪基站工具：

```
$ cd /rf/public/openbts/trunk/apps
$ sudo ./OpenBTS
```

(10) 为你所在地区信号最强的网络运营商确定一个在当地合法的 MCC(Mobile Country Code，移动设备国家代码)和 MNC(Mobile Network Code，移动设备网络代码)：

```
$ cd /rf/public/openbts/trunk/apps
$ sudo ./OpenBTS
```

MCC/MNC 数组的组合是 MCC 与 MNC 结合起来使用，可通过它在 GSM 网络识别每个移动网络运营商。无线电话网络(GSM、CDMA、UMTS 等)使用 MCC 来识别移动用户所属国家。为了唯一地识别移动用户网络，便将 MCC 与 MNC 结合起来使用。MCC 和 MNC 的组合被称为家庭网络标识(Home Network Identity，HNI)，将两者组合为一个字符串(例如，MCC=262，MNC=01，那么 HNI 为 26201)。如果将 HNI 与移动用户识别号(Mobile Subscriber Identification Number，MSIN)相结合，结果就是集成移动用户识别码(Integrated Mobile Subscriber Identity，IMSI)。还可在 www.mcc-mnc.com 找到各运营商的 MCC 和 MNC 的最新名单。

如果未设置 MCC 和 MNC 并使用默认值，你将看到默认的网络名称为 TEST、RANGE 或 SAFARICOM。

你可在手机上进行本地手机信号塔的本地搜索，从而测试伪基站。基站应该被列在网络列表中。加入该网络，并寻找你的伪基站的欢迎信息，如图 1-6 所示。

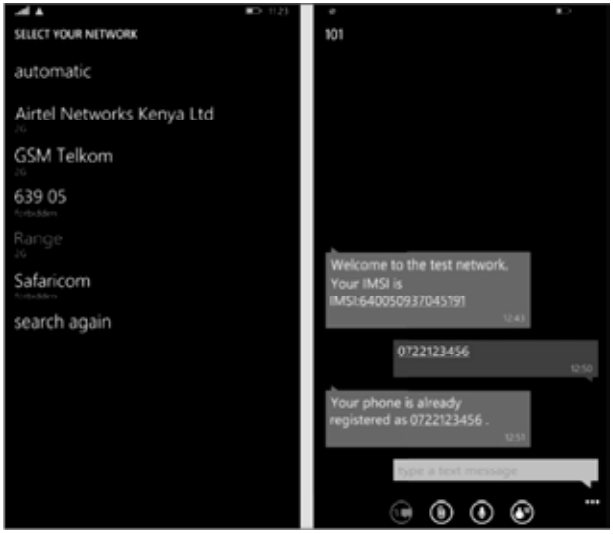


图 1-6　加入新的 OsmocomBB 伪基站的手机

恭喜你!伪基站现已设置完毕,可在实验室里接受 TCU 的连接了。

现在我们已经用 OsmocomBB 设置了一个伪基站，下一节将介绍如何使用 YateBTS 的 BladeRF 来搭建一个伪基站。这是本节手机+OsmocomBB 的替代方案。

1.7.4　Rogue BTS 方案 2：BladeRF + YateBTS

在如今 Google Pixels 和 iPhone 的时代,要找到一部支持 OsmocomBB 的手机是很困难的，所以方案 1 对你来说可能难以实现。此外，本书的其余部分都以使用 BladeRF 为基础，所以即使你能找到方案 1 支持的手机，也可能更愿意使用这个方案。本节将详细介绍设置和升级 BladeRF 最新固件的步骤，以及安装所需的驱动程序的步骤。可直接从 Nuand 网站 www.nuand.com 购买到不同型号的 BladeRF。建议顺便购买安装板的塑料保护壳，因为购买的 BladeRF 没有附带保护壳。

购买 BladeRF 并通过提供的 USB 电缆插入你的笔记本电脑后，需要在实验室设置中执行以下操作：

- BladeRF tools/PPA(https://github.com/Nuand/bladeRF/wiki/Getting-Started:-Linux)
- 笔记本+ Linux

个人软件包存档(Personal Package Archive，PPA)是一种简单的软件分发方法，取消了通过主 Ubuntu 存储库分发软件的过程，允许开发人员将其作为单个包传送。

(1) 从 PPA 下载并安装 Linux 软件包，并为 BladeRF 刷入最新版本的固件，从而设置你的新 BladeRF：

```
$ sudo add-apt-repository ppa:bladerf/bladerf*
$ sudo apt update
$ sudo apt install bladerf libusb-1.0-0-dev
$ sudo apt install gr-gsm
```

我撰写本书时,Kali-Rolling apt 仓库包含了 bladeRF 和 libbladerf

包。如果使用 Kali Linux 2018.1 或更高版本的 Kali Linux，则不需要添加 apt 仓库。可直接跳转到步骤(3)中的第三个命令来安装 BladeRF 和 libbladerf-dev 包。

(2) 安装 BladeRF 头文件(可选)：

```
$ sudo apt install libbladerf-dev
```

(3) 给 BladeRF 刷入最新版本的固件，具体执行哪个命令取决于你购买的 BladeRF 的版本：

```
# For the bladeRF x40:
$ sudo apt-get install bladerf-fpga-hostedx40

# For the bladeRF x115:
$ sudo apt-get install bladerf-fpga-hostedx115

# Load the firmware
$ bladeRF-cli -l /usr/share/Nuand/bladeRF/hostedx40.rbf
```

(4) 使用 BladeRF-cli 工具来测试 BladeRF 的基本功能，从而验证固件升级是否成功：

```
$ bladeRF-cli -p
```

这个命令应该会返回类似于图 1-7 的输出。另外，尝试运行下面的命令：

```
$ bladeRF-cli -e version ; bladeRF-cli -e info
```

图 1-7　bladeRF-cli -p 命令的输出

你的 BladeRF 现在应该将所有 LED 灯都点亮；这些灯常亮，颜色为绿色。

恭喜你！你的新 BladeRF 已经可与 YateBTS 一起使用了。

(5) 下载和编译 YateBTS：

```
$ apt install subversion
```

```
$ apt install autoconf
$ apt install gcc
$ apt install libgcc-6-dev
$ apt install libusb-1.0-0-dev
$ apt install libgsm1-dev
$ cd /usr/src
$ svn checkout http://voup.null.ro/svn/yatebts/trunk yatebts
$ cd yatebts
```

如果你收到错误信息，指出某个软件包不存在，可能是你使用的 Kali 和本书中使用的 Kali 版本不一样。可使用 apt 搜索命令查找相应的软件包及其当前的版本号。

另外注意，截至本章撰写时，libgcc 最新的版本为 6。这使本书撰写时当前版本的 YateBTS 出现了问题。Yate 开发团队为此制作了一个补丁，这里提供了关于如何打补丁的说明，以免你在编译过程中遇到错误。未来版本的 YateBTS 可能不需要这个补丁，因为这个补丁的内容将在未来的版本中实现。

(6) (可选)尝试安装 YateBTS 时，如果你在之前的步骤中收到错误信息，请下载并应用 YateBTS 的 libgcc 6 补丁：

```
Download the patch from: http://yate.rnull.ro/mantis/view.ph?id=416
Copy the patch file yatebts-5.0.0-gcc6.patch to the root directory
of yatebts in /usr/local/etc/yatebts.

  $ svn patch –strip 1 yatebts-5.0.0-gcc6.patch
  $ make clean
  $ ./autogen.sh ; ./configure ; make install
```

(7) 安装并运行 YateBTS NIPC(PC 中的网络)：

```
  $ cd/var/www/html
  $ ln -s/usr/src/yatebts/nipc/web nipc
  $ chmocl atrw/usr/local/etc/yate;chown www-data */usr/
local/etc/yate
```

注意：PC 中的网络是一个系统中的完整 GSM 网络，实现了用户注册和 GSM 网络内外呼叫路由的必要应用。

(8) 启动 Apache 并浏览到新的 NIPC 安装：

```
$ service apache2 restart
```

(9) 打开网页浏览器，查看新的 NIPC 管理页面：

```
http://localhost/nipc
```

随着 NIPC 的运行，现在可使用刚才安装的 NIPC 图形界面来配置 YateBTS 了。这里，你需要按照前面的说明配置 MCC、MNC 和频带。

要获取 ARFCN / UARFCN / EARFCN，需要让手机进入“现场测试模式”。具体步骤在不同的手机之间差异很大。

绝对无线频道编号(Absolute Radio-Frequency Channel Number，ARFCN)是 GSM 中的一个术语，定义了移动无线电系统中提供上行和下行信号的一对物理无线电载波。

(10) 配置 YateBTS。

由于我不知道你的具体参数，这里给出了我的配置以供参考：

BTS Configuration > GSM > GSM

Radio.Band: PCS1900

Radio.C0: #561 1940MHz downlink/1860MHz uplink

Identity.MCC: 310

Identity.MNC: 410

TAPPING:

注意，利用这些设置，你可使用 Wireshark 来捕获由 Yate 发送到本地环回地址的所有数据包。

[x] GSM

[x] GPRS

TargetIP: 127.0.0.1

SUBSCIBERS:

Country Code: 1

SMSC: .*

(11) 启动 YateBTS：

```
$ cd /usr/src/yate
```

以调试/详细模式启动：

```
$ yate -vvvv
```

以守护进程模式启动：

```
$ yate -d
```

以常规前台模式启动：

```
$ yate -s
```

恭喜你！你现在正在运行一个基于 BladeRF 和 YateBTS 的伪基站。现在可等待并接受 TCU 的连接了。

1.7.5 设置 WiFi Pineapple Tetra

由 Hak5 制造的 WiFi Pineapple 是一个模块化的无线审计平台，通过一个简易的 Web 用户界面提供了一些功能。

通过扫描功能，可从仪表板上识别出本地的接入点(隐藏或未隐藏)和攻击。不同于较小的同系列产品 NANO，Pineapple TETRA 能同时支持 2.4GHz 和 5GHz 通道。因此，不建议购买 NANO 来进行 CPV 的渗透测试。Pineapple 能通过一套 WiFi 中间人工具(专用于目标资产收集)对获取的客户信息进行无线截获，本书中会用到这套工具。

Pineapple 以 Hak5 的 PineAP 工具为核心，是一个集侦察、中间人等攻击工具为一体的集合，可用来攻击无线接入点和客户端。虽然配置使用无线网卡的 Linux 和其他免费的开源工具可达到同样的效果，但这里想演示一下如何使用一个商业化的现有工具(Commercial Off-The-Shelf，COTS)，你可考虑把它当作一种替代方案。

这里将使用 Linux 的指令来配置 WiFi Pineapple，因为 Linux 是本书中使用的平台。

(1) 最新固件可从网站 https://www.wifipineapple.com/downloads/

tetra/latest 下载。

(2) 使用随附的 USB y-cable 将 Tetra 连接到电脑上。

(3) 如果所有连接都正常，你现在应该有一个新的网络接口，IP 地址分配到 172.16.42 子网。

(4) 打开 Web 浏览器，并登录到 http://172.16.42.1:1471 来连接 Pineapple(只有 Chrome 和 Firefox 官方支持)。

(5) 按下 Tetra 背面的复位按钮，将 Tetra 重置。

(6) 单击升级链接升级 Tetra，然后等待。蓝灯亮时表示固件升级成功。

(7) 按照指示完成升级过程。

(8) 通过下载并运行 wifipineapple.com 提供的 wp6 脚本，就可让 Tetra 通过你的笔记本电脑共享互联网。要实现这个功能，请运行以下命令：

```
$ wget wifipineapple.com/wp6.sh
$ chmod 755 wp6.sh
$ sudo ./wp6.sh
```

也可通过将一根以太网线连接到 Tetra 上的以太网端口来接入互联网。这是一个替代方案。

(9) 重新登录到 Tetra 的 Web 用户界面中。如果网络连接正常，可在登录后在登录页面上看到公告栏下的最新新闻。

现在，你的 WiFi Pineapple TETRA 应该已经可启动并运行了，接下来我们将在第 4 章中使用它。

1.8　本章小结

本章介绍了项目管理在执行 HU 和 TCU 渗透测试中的重要性，以及 PMBOK 结构化项目的 5 个阶段：构思和发起、规划、执行、绩效/监督和项目完结。项目的各个阶段与渗透测试执行标

准(PTES)的情报收集、侦察、漏洞分析、漏洞利用和后渗透阶段相一致。本章还介绍了 WBS 的要素，以及与利益干系人定义测试范围、创建 ROE 列表的重要性。

本章讨论了在渗透测试开始阶段可能需要的重要工程文档，以及这些文档中通常包含的内容。

最后，建立了一个基于 Kali Linux 工作站的实验室，以及两种搭建伪基站的方案：YateBTS 和需要老式摩托罗拉手机运行的 OsmocomBB。我们还设置了 Hak5 WiFi Pineapple TETRA。

现在你不仅知道了渗透测试的不同阶段，还为进行渗透测试而构建了新的实验室。下面的章节将进入下一个阶段：情报收集。