第 6 章 组件与恶意软件分发

恶意软件与其他良性软件一样，都要经过常规的开发流程。目前，恶意软件的开发并不逊色于全球性的软件公司。甚至有的团伙除了恶意软件开发团队，还拥有完整的 QA 流程。从开发视角来看，几乎与正常软件开发没有分别。从受害者的视角看，攻击者也希望确保恶意软件能够尽可能多地在各种终端上运行，这一点与良性软件也是一致的。攻击者总是希望确保恶意软件能够造成更大的影响，无论是影响范围还是受影响目标的重要程度，这样才能获得更高的投资回报。为了扩大影响，攻击者编写了针对各种操作系统的恶意软件：Windows、macOS、Linux 与 Android。攻击目标也在不断扩展，从台式机、笔记本到服务器、移动设备、工控设备、POS 设备与物联网设备。

可无论恶意软件攻击的目标平台是什么、其开发语言是什么、最终运行在什么类型的设备上，几乎所有恶意软件都包含几个基本组件。在本章中，将会简要介绍构成恶意软件的这些组件，以及恶意软件开发完成后如何分发给受害者。

恶意软件组件

从较高的层级来看，大多数恶意软件都能够被分为如图 6- 1 所示的几个组件。

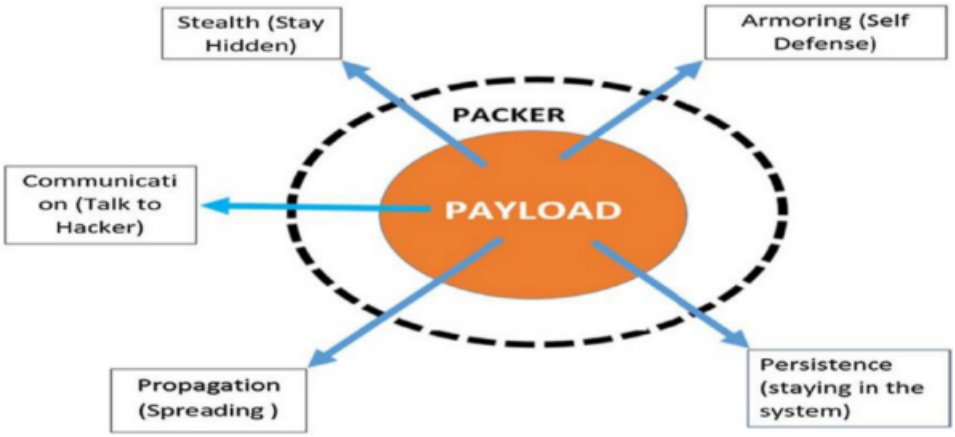

图 6- 1 高层级划分恶意软件组件

一般恶意软件都包含 Payload、壳、持久化、自我保护、维持隐蔽、通信与传播这样几个基础组件。在接下来的章节中，将要逐一介绍这些组件以及如何使用各种分析工具对这些组件进行动静态分析。

Payload

Payload 是恶意软件最重要、最必需的组件。恶意软件感染通常是由多个二进制文件（可执行/不可执行）共同完成的，攻击者组合使用这些二进制文件达成自己的目标。攻击链中的每个二进制文件都可以被称为 Payload，但严格来说只有那些样本中包含并负责实现攻击者真正意图的那部分功能是 Payload。作为恶意软件分析人员，在对恶意软件进行分析时会尽可能地收集完整攻击链中的各个恶意文件。对恶意软件的命名与分类主要是围绕 Payload 进行的，而不是根据攻击链中各个样本的表现。

此处列举了部分 Payload 的种类：

- 窃密恶意软件（Password stealer，PWS）：从浏览器、FTP 客户端与其他程序中窃取受害者密码等凭据信息。
- 银行恶意软件（Banking malware）：旨在窃取受害者的银行凭据。
- 勒索软件（Ransomware）：对受害者的敏感数据与资源进行加密，在获取赎金后释放这些资源返还给受害者。
- 广告软件（Adware）：向受害者展示并不需要的广告。
- POS 恶意软件（Point of Sale malware）：通过与 POS 设备连接的系统窃取信用卡信息。

业界主要是根据 Payload 对恶意软件进行分类，但在分类时并不考虑一些较小的 Payload，如 Dropper、Downloader、Wiper 等。在第 15 章中将会更加深入探讨如何识别 Payload

以及对恶意软件进行分类。

壳

良性软件与恶意软件都可以使用壳，恶意软件中壳通常位于 Payload 的外层，起到压缩与混淆的作用。通常来说，加壳的目的是压缩，但也间接实现了混淆。因为此时内部的 Payload 对外不再可见，反病毒软件的静态分析与静态签名都难以检测加壳的恶意软件。换句话说，恶意软件通过加壳来隐藏 Payload，从而隐藏其真实意图。

在对恶意软件进行逆向工程时，为了查看恶意软件的 Payload 或者功能需要将壳脱掉（脱掉壳代码的过程被称为脱壳）。反病毒软件在尝试脱壳二进制样本文件时会用到实现好的脱壳算法，但安全厂商很难为已有的、成千上万的加壳程序一一编写脱壳程序。第 7 章中将会更加详细地讨论加壳与脱壳，其中会涉及到各种脱壳方法。

持久化

各种恶意软件都希望能够在失陷主机上持续存在，即使多次重启或登录仍能驻留在系统中。恶意软件维持持续驻留的技术被称为持久化，攻击者通常会选择利用操作系统的特性来保持驻留。以下列出了恶意软件需要持久化的部分原因：

- 银行恶意软件旨在窃取银行凭证信息，攻击者希望恶意软件能够始终保持运行，以便在用户打开浏览器登录银行网站时窃取凭证信息。
- 远控恶意软件旨在监控受害者的活动并将信息回传给攻击者，只有在用户使用系统时恶意软件能够保持运行状态才能做到这一点。
- 勒索软件不仅要加密系统上的现有文件，还要加密用户在重启后创建的新文件。

Windows 平台上，恶意软件通常会篡改注册表中与系统启动和启动程序有关的配置来维持

持久化，在第 8 章中也将更加深入地探讨持久化。

C&C 通信

不论动机是出于上传窃密数据还是接收攻击者命令，大多数恶意软件都会通过命令与控制（C2C/CnC/C2/C&C）服务器与攻击者进行通信关系。十年前的 C&C 通信十分简单，如 IRC 通信或者简单的 HTTP 通信。随着 IDS、IPS 与下一代防火墙等网络安全产品的发展，增强了对恶意软件通信的检测拦截。这都促使恶意软件升级更为复杂的通信机制，如使用 HTTPS、DNS 隧道、DGA、Tor 等，在第 9 章中将对该内容进行详细讨论。

传播

恶意软件都会期望能够感染更多的目标机器，除了获取更多的受害者外还有其他的原因。例如 APT 的攻击目标通常不在能够轻松触达的位置，Autorun Worms 就是通过 USB 闪存驱动器穿透气隙隔离，从一台主机传播到另一台主机。另一个臭名昭著的 Wannacry，则是使用“永恒之蓝”漏洞通过网络感染其他主机。

另一种传播机制是通过 PE 文件感染，通常会被病毒或者文件感染恶意软件使用。病毒将自己的代码插入到另一个良性文件中劫持其执行流，使得每次被执行都会感染更多良性文件，以此实现传播。如果受感染的文件被拷贝到无感染环境中执行，也会感染无感染环境中的其他良性文件。

恶意软件还可以通过各种渠道（如 SMB 和共享文件夹）在网络上传播。众所周知，攻击者和恶意软件会利用各个软件使用的默认凭据进行传播、也会利用软件存在的漏洞进行传播，如 WannaCry。在第 9 章中会对这些内容进行介绍。

自我保护

恶意软件肯定不希望被反病毒软件检测到，也不希望被恶意软件分析人员进行分析。恶意软件会使用反调试、反杀软、反虚拟机、反沙沙盒、反分析工具等技术进行自我保护。

恶意软件分析人员通常会在安装了各种分析工具（如 Process Hacker、OllyDbg、IDA Pro、Wireshark、ProcMon 等）的虚拟机中对恶意软件进行分析。所以，恶意软件通常会检查虚拟机在系统上保留的特征判断是否运行在虚拟机中，以及系统中是否安装了各种分析工具。当恶意软件检测到在分析环境中运行时，恶意软件可能会终止执行或者表现出良性行为来欺骗恶意软件分析人员。

除了恶意软件分析人员之外，以反病毒软件与沙盒为代表的各种安全产品也都是恶意软件防范的重点。通过系统上的文件、进程与注册表项可以检测反病毒软件，而在内存中可以通过匹配与虚拟机和沙盒组件相关的特征来检测基于虚拟机的沙盒。在检测发现安全软件存在的情况下，恶意软件可能会表现得相当正常，以避免被检测到。

为了突破这种“全副武装”的恶意软件，研究人员可以对恶意软件进行逆向工程，手动跳过/绕过恶意软件使用的这些自我保护代码。或者，也可以使用二进制插桩技术自动检测并跳过此类代码，使恶意软件去执行真实代码。在后续的章节中也将会对这两部分内容进行介绍。

维持隐蔽

恶意软件为了不被用户发现、不被反病毒软件检出，需要将自身隐藏起来。除勒索软件外，大多数恶意软件都更希望能够隐蔽执行。隐蔽技术多种多样，从简单的更改文件属性进行隐藏，到复杂的代码注入、Rootkit 与 Process Hollowing 等。隐蔽性是银行木马、远控木马和其他部分恶意软件最重要的特性，在第 10 章与第 11 章中将会对各种隐蔽技术与 Rootkit 进行介绍。

分发机制

恶意软件需要分发到其他主机上以便进行感染，恶意软件的分发与开发同样困难。以下是攻击者在分发恶意软件时想要达成的一些目标：

- 确保在分发恶意软件时无法追踪攻击者。
- 能够有效地投递恶意软件并感染目标主机。
- 针对某个国家、地区或者公司进行的定线攻击，分发机制必须确保不会感染预定目标以外的其他受害者。
- 能够绕过基于网络与主机的安全产品。

大多数分发机制都严重依赖社会工程学，鱼叉邮件就是一种典型的社会工程学分发渠道，这是攻击者借助电子邮件这种古老但有效的攻击手段发起攻击的方式。受害者点击邮件中的恶意链接或者执行恶意附件，触发恶意软件感染。另一种典型的分发机制是路过式下载（drive-by download），在受害者并不知情的情况下完成恶意软件感染。

总的来说，分发机制可分为三大类：

- 物理分发：使用跨设备共用的 USB 闪存驱动器与硬盘驱动器进行传播与感染。
- 网站分发：恶意软件部署在网站上，受害者访问到此类网站时就会被感染。这些恶意网站的地址通过电子邮件或者恶意广告进行传播，甚至是入侵合法网站。网站上部署了恶意软件或者漏洞利用工具包，在受害者毫不知情且无需任何形式的交互的情况下进行感染与传播。本章后续也将对漏洞利用工具包进行详细介绍。
- 邮件分发：这是最古老也最常用的技术，通过电子邮件直接发送恶意软件作为附件或者将指向恶意网站的链接嵌入电子邮件中。攻击者经常将 Microsoft Office 文档、PDF 文档和其他脚本文件作为附件，由这些附件作为 Downloader 或者 Dropper 下载其他恶意软件与 Payload。

现在可以看下这三类分发机制中一些典型的代表技术：

漏洞利用与漏洞利用工具包

程序员在编写代码时不可避免地会犯错误，在程序中以 Bug 的形式体现。其中一些 Bug 可能非常严重，攻击者利用这些 Bug 甚至可以控制程序执行甚至控制系统，此类 Bug 被称为漏洞。如果存在漏洞的应用程序部署在服务器上，随着应用程序执行被控制，服务器也可能被攻陷。

假定我们有一个存在漏洞的进程，攻击者如何控制该进程呢？攻击者会编写被称为漏洞利用的一小段代码，将其作为程序的输入触发漏洞。存在漏洞的进程接收到包含漏洞利用的输入并对其进行处理时，漏洞利用会针对漏洞进行攻击，攻击成功后使 CPU 执行漏洞利用代码从而控制进程。整体过程如图 6- 2 所示：

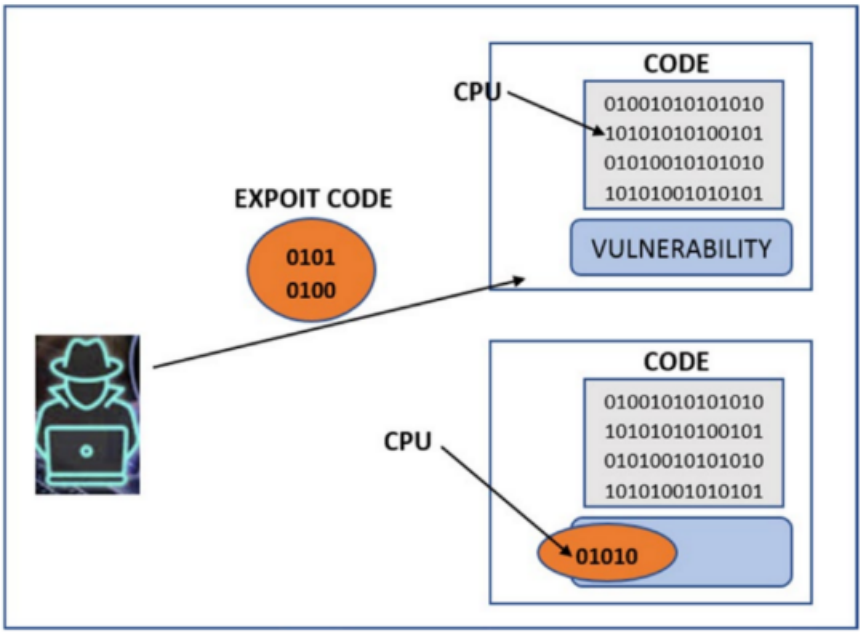

图 6- 2 利用漏洞获取进程控制权过程

如今，大多数漏洞利用代码都会承担下载、执行恶意软件的职责。任何类型的软件或者应用程序都可能存在漏洞并且能够被利用，从 Apache 和 Nginx 等 Web 服务器到 Postfix 等 SMTP 服务器都不能避免。此外，计算机中常用的程序都被发现过存在漏洞。如 Internet

Explorer、Firefox 和 Chrome 等网络浏览器；Adobe Flash 和 Silverlight 等浏览器插件；Adobe Acrobat 和 Foxit 等 PDF 阅读器以及 Microsoft Word 等 Microsoft Office 应用程序。甚至，在 Windows 与 Linux 的操作系统内核中也发现存在漏洞。接下来的几节中，将会解释一些与漏洞相关的术语。

通用漏洞披露（CVE）

在新漏洞被发现时，可以根据固定的命名约定将其报告给为漏洞提供通用名称（CVE-ID 或 CVE 名称）的组织，将有关信息保存在公开披露漏洞的通用数据库中。读者可以自行访问 www.cvedetails.com 并搜索已知的任何厂商/软件的相关漏洞，例如 Internet Explorer，网站上会显示其披露的漏洞列表。通过网站可以浏览一些漏洞的相关信息，了解 CVE 漏洞数据库中如何对漏洞进行描述。值得注意的是，CVE 漏洞数据库中仅包含公开披露的漏洞信息。

补丁：修复漏洞

很多程序都存在漏洞，防御者要在攻击者开始利用前就发现并且修复这些漏洞。安全研究人员通常会积极地发现程序中存在的漏洞，他们这样做可能是出于想要回馈社区并改善安全状况、可能是出于想要获取丰厚的漏洞赏金，也可能是出于名声与声誉考虑。安全研究人员会通过很多种方式发现漏洞，在确认发现漏洞后，有些人会公开发布漏洞描述信息，有些人也会私下联系厂商进行负责任地信息披露，避免漏洞被攻击者滥用。

不论何种情况，厂商都会针对存在漏洞的软件提供新版本进行漏洞修复。厂商将提供的漏洞修复称为补丁，通常以更新的方式推送到各个终端进行部署升级。大多数情况下，厂商在得知存在漏洞后会立刻着手编写并发布补丁程序。但从得知漏洞到编写并发布补丁通常需要几

天甚至几个月的时间，在此期间如果漏洞的相关信息被对外披露，攻击者就可以编写针对该漏洞的漏洞利用进行攻击。因此，安全研究人员通过一定渠道进行负责任地信息披露告知厂商是非常重要的。

近年来，漏洞赏金计划是一个热门且获利丰厚的领域。发现新漏洞将其报告给厂商的安全研究人员将会获得赏金奖励，作为负责任地漏洞披露的奖励。请记住，寻找漏洞的不仅有安全研究人员，攻击者也会不断发现新漏洞以利用这些漏洞进行攻击和获利。甚至于，部分安全研究人员也会向市场上愿意支付合适价格的买家出售漏洞与对应的漏洞利用程序。

0day 漏洞

如前所述，厂商会针对它已知的漏洞通过软件更新发布补丁来修复漏洞。但对于那些未知的漏洞，厂商就无能为力了。0day 漏洞存在于未修复的软件中，厂商可能并不知道软件中存在该漏洞，又或者知道该漏洞但并未进行修复。总之，没有被修复的漏洞被称为 0day 漏洞。如果攻击者发现了一个 0day 漏洞，这会为其带来技术优势，可以顺利攻击那些使用存在该漏洞的软件的用户。

攻击者如何利用漏洞

大多数漏洞都是由于程序对输入缺乏验证导致的。例如，程序需要一个名称作为输入，程序预期输入为一个字母序列，而用户输入了数字。如果程序没有验证输入是否只包含字母，最后就会接收到无效输入并且产生意想不到的后果。

不同的程序以不同的方式接收各种输入。例如 Web 服务器以 HTTP 请求的形式接受输入，而处理输入的后端可能是各种框架和各种语言（如 Ruby、Django、Python、NodeJS 等）。同样的，各种互联网的网页也是用户端程序的输入。例如，当用户使用 Internet Explorer、

Chrome 和 Firefox 等浏览器浏览网页时，HTML 页面就是浏览器的输入。如果攻击者知道目标机器上使用哪些软件，并且知道这些软件中存在哪些漏洞，就可以针对这些漏洞专门开发特定的漏洞利用代码。

攻击者是如何攻击部署了 Web 网站的服务器？在攻击 Web 网站之前，攻击者会确定 Web 服务使用的软件名称，此外还需要找到 Web 服务的确切版本。攻击者可以通过探测获取指纹信息或者使用互联网公开的信息获知 Web 服务的名称与版本。确定后，攻击者就可以利用该软件该版本中存在的、尚未修补的任何已知漏洞，创建特定的漏洞利用发送 HTTP 请求进行攻击。

与 Web 服务相比，攻击普通桌面用户会更困难。服务器对外暴露了部署的软件，公众可以通过网络与 Web 服务进行通信交互，攻击者也可以很容易地进行攻击尝试。普通桌面用户则不运行任何对外暴露的软件，或者其他任何人可以在任何位置进行访问的服务。因此，为了攻击普通桌面用户，攻击者提出了一种被称为漏洞利用工具包 (Exploit Kit) 的攻击方式，使用该方式攻击并感染普通桌面用户。下一节将要重点讨论漏洞利用工具包。

漏洞利用工具包（Exploit Kit）

互联网服务器上部署对公众开放的服务与软件，攻击者都可以直接连接与之通信。这种情况下，攻击者就可以直接发起对服务器的攻击，漏洞利用成功时即可控制服务器。而如果是普通桌面用户，攻击者没有直接访问的通路，毕竟计算机隐藏在家庭网络网关之后。

为了攻击与感染普通桌面用户，攻击者转变攻击策略为“守株待兔”。攻击者通过恶意服务器在互联网上部署诱饵和陷阱，等待毫无戒备之心的用户访问这些恶意服务器，从而感染这些用户。

攻击者在设置陷阱时可能会面临很多问题，很多情况都可能会导致陷阱无法正常工作。最主

要的原因就是攻击者无法确定用户使用软件的准确版本，其次是用户使用的软件可能不容易受到攻击。为了填补这一市场的空白，攻击者开发了漏洞利用工具包。

漏洞利用工具包中包含一个漏洞利用，而是包含一系列漏洞利用，是针对各种软件与版本中的漏洞开发的各种漏洞利用。实际上，所有的用户不会使用相同的软件来上网，用户可以选择使用 Internet Explorer、Chrome、Firefox、Safari、Edge 等各种浏览器访问恶意网站，甚至是各种不同版本的浏览器。漏洞利用工具包则将各种情况都考虑进来，其中包含针对各种浏览器与各个版本的浏览器开发的漏洞利用。

漏洞利用工具包通常部署在攻击者控制的 Web 服务器上，但漏洞利用工具包并不是通过让受害者直接访问部署漏洞利用工具包的恶意网站来发挥作用的。相反，漏洞利用工具包常常会利用被称为登陆页面（landing page）的网页来实现中转。

如图 6- 3 所示，登陆页面是漏洞利用工具包的“门面”，作为前置过滤将有效的漏洞利用发送给受害者。登陆页面通常是一个带有 JavaScript 代码的网页，网页接收到用户的连接请求后，会获取浏览器名称、浏览器版本、浏览器安装的插件、操作系统名称、操作系统版本以及机器上安装的其他软件等信息。得到相关信息后，登陆页面会确定用户是否安装了存在漏洞的浏览器或者其他软件。确认存在后，登陆页面从漏洞利用工具包中选择一个可用的漏洞利用发送给受害者，对受害者进行感染。漏洞利用投递的详细步骤如下：

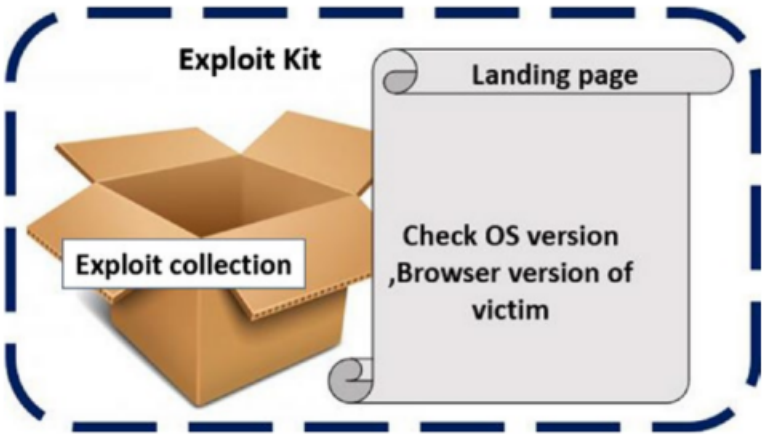

图 6- 3 登陆页面是漏洞利用工具接收用户请求的前置过滤

漏洞利用工具包攻击流程

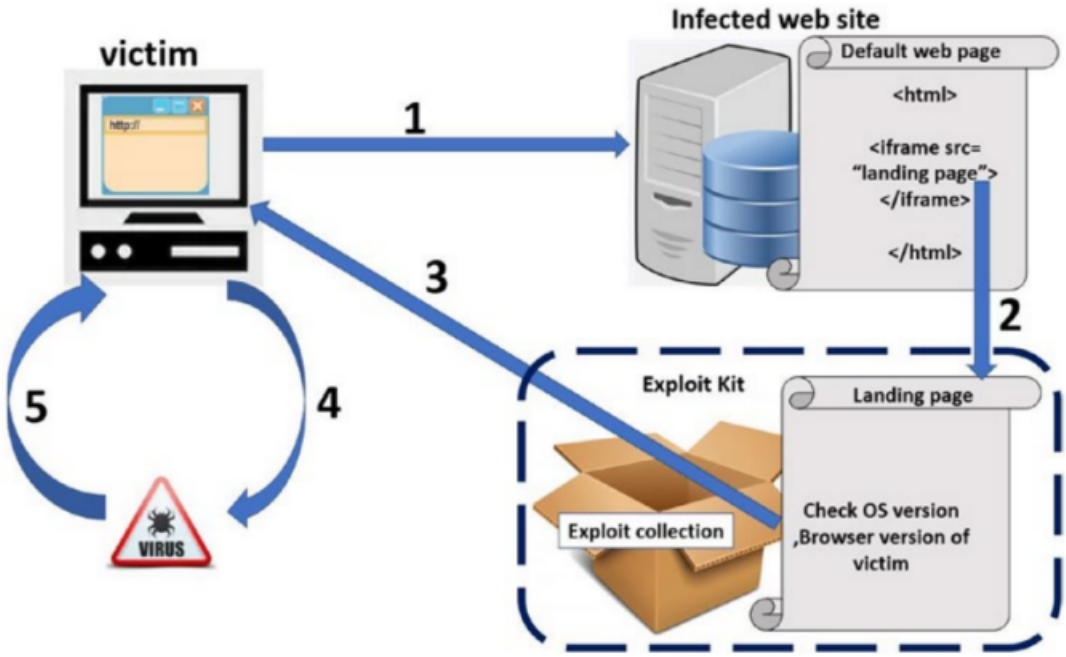

图 6- 4 漏洞利用工具包在攻击后下载恶意软件的完整流程

图 6- 4 展示了漏洞利用工具包的完整攻击流程，漏洞利用工具包的投递步骤如下所示:

1、受害者访问攻击者控制的恶意网站。恶意网站可能是攻击者自己搭建的服务器，其中包含恶意内容并将其返回给任何访问网站的人。更常见的情况是，攻击者攻陷了一个正常网站并篡改了网页的内容，将恶意内容返回给本欲访问正常网站的人。

攻击者如何入侵一个正常网站呢？网站通常是使用 Web 框架与对应编程语言编写的，而在编程语言以及 Apache、Nginx 与 IIS 等 Web 服务软件中，漏洞屡见不鲜。攻击者利用这些 Web 服务软件中的漏洞发起攻击，攻陷服务器。此外，许多 Web 服务软件也存在配置错误缺陷，如管理员使用默认凭据（admin/admin123 等）登录服务器。此时，攻击者就可以通过暴力破解或者字典猜测来实现登录与控制。

攻击者在控制服务器后，对网站原有的内容进行篡改。被篡改后，网页就包含了能够攻击与感染访问者的恶意内容。常见的恶意内容是利用隐藏的 iframe 将超链接指向漏洞利用工具包的登陆页面，这种恶意 iframe 注入的攻击方式在许多网站中都被发现过，这些网站通常

都拥有大量的访问者。尽管网页包含了隐藏的 iframe，但这对访问者来说是不可见的，并不会因为插入隐藏的 iframe 改变页面结构，因此很难被注意到。

2、当普通桌面用户访问被感染的网站时，被插入网页的隐藏 iframe 会自动请求并加载登陆页面。尽管受害者的浏览器访问并加载了登陆页面的内容，但并不需要用户点击任何内容即可触发。另外，由于在隐藏的 iframe 中加载，登陆页面对用户也是不可见的。

3、登陆页面的 JavaScript 代码运行，获取访问者使用的浏览器与各种软件相关信息。得到信息后，如果访问者使用的软件存在漏洞，登陆页面会选择一个合适的漏洞利用并发送给访问者。

4、包含漏洞利用的网页内容被返回给受害者，当这些内容被浏览器加载时就会触发漏洞利用。漏洞利用代码被执行，在攻击成功后就能够控制浏览器的代码执行了，如图 6- 2 所示。

5、攻击者可以控制受害者达成自己的意图，例如回连攻击者控制的其他恶意服务器、下载并执行恶意软件。

将漏洞利用工具包用于恶意软件分发

漏洞利用只是一小段代码，本身能承载的功能有限。攻击者通常希望实现的功能无法通过漏洞利用代码实现，这也是漏洞利用常常作为初始攻击向量与分发机制的原因。

一旦漏洞利用代码被执行，就可以通过互联网上的一个恶意服务器下载功能更完整、更强大的恶意软件。执行恶意软件后就可以完成感染，漏洞利用代码本身就实现了高效与隐蔽的恶意软件分发。

漏洞利用工具包案例研究

各种攻击团伙创建了数十个漏洞利用工具包,其中大多数在 2016 年到 2018 年间最为活跃。最活跃的漏洞利用工具包是 RIG、Sundown、Blackhole 和 Magnitude 等。

网站 www.malware-traffic-analysis.net 持续跟踪活跃的漏洞利用工具包，接下来以 Magnitude 漏洞利用工具包为例进行分析。在提供的示例样本库中，文件 Sample-6-1.txt 与下载 Cerber 勒索软件的 Magnitude 漏洞利用工具包有关。

文件 Sample-6-1.txt 中包含下载 PCAP 文件 2017-08-02-Magnitude-EK-sends-Cerber-ransomware.pcap 的链接地址，下载后就可以使用在第 2 章中介绍过的 Fiddler 打开该文件。Fiddler 是一个非常有用的工具，可以将 HTTP 数据交互可视化，大大简化分析。如图 6- 5 所示，将 PCAP 文件拖动到 Fiddler 图标上，就可以加载该文件。

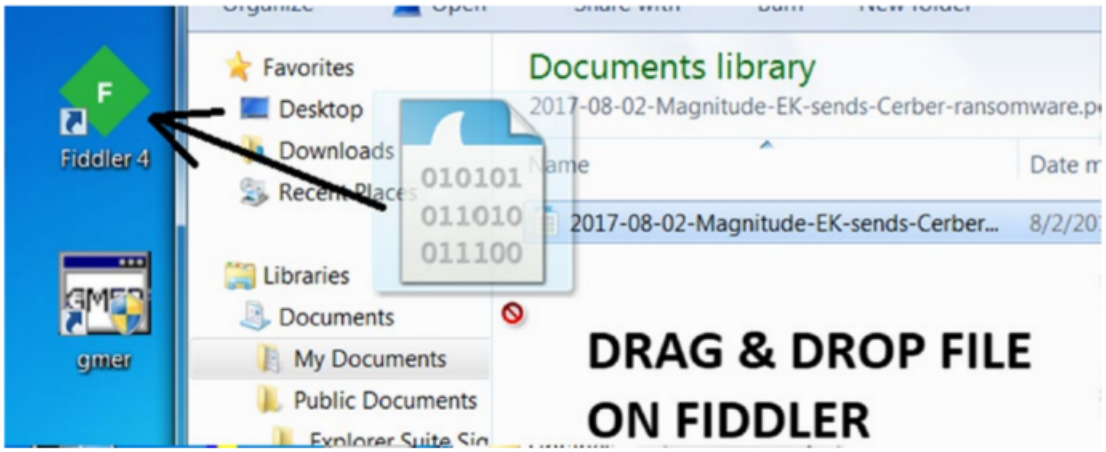

图 6- 5 拖拽加载 PCAP 文件到 Fiddler

Fiddler 以非常直观的方式列出了 PCAP 文件中的 HTTP 数据交互,包括 HTTP 请求与响应。

漏洞利用工具包针对存在漏洞的浏览器与浏览器插件，通过漏洞利用发起攻击。

Adobe Flash 一直都是被广泛使用的浏览器插件之一。如图 6- 6 所示，HTTP 请求-响应中的 5、6 和 7 都显示了服务器回传给用户的 Adobe Flash 文件。在第 9 行的服务器响应中，能够发现服务器回传了一个 PE 文件（通过 MZ 的 Magic Number 确认，在第 3 章中

有介绍过)。现在，PE 文件很少作为 HTTP 请求的响应被直接下载。先是下载了 Flash 文件，紧接着又下载了 PE 文件，推测应该是通过 5、6、7 中的数据包对 Adobe Flash 进行了成功的漏洞利用攻击。

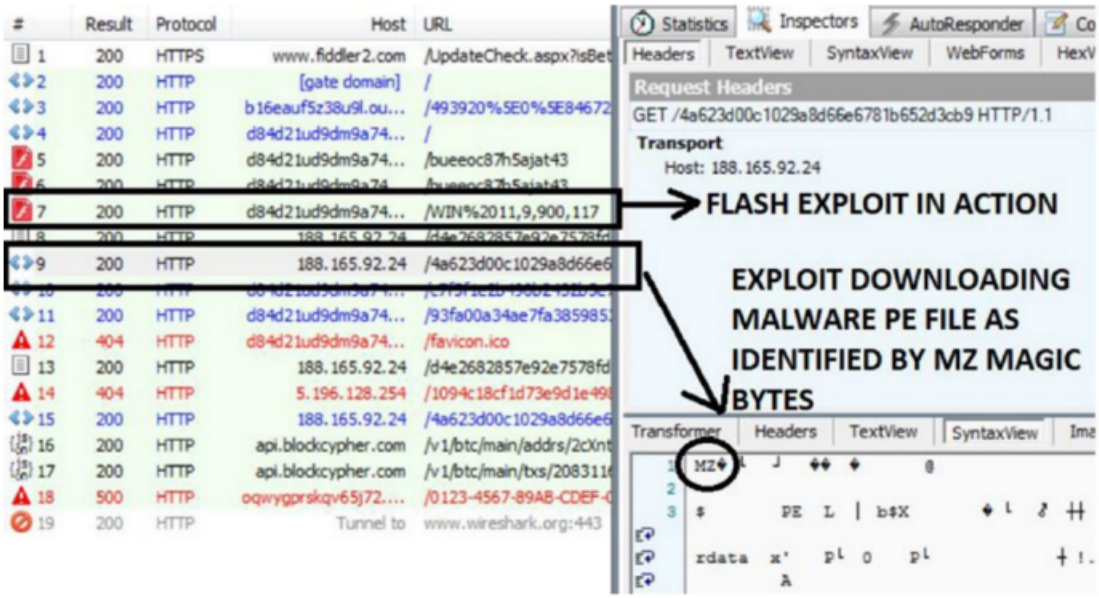

图　6- 6 HTTP 通信中成功完成漏洞利用并下载恶意软件

垃圾邮件

垃圾邮件是最古老的、最常见的恶意软件分发机制。攻击者将电子邮件发送给大量用户，这些邮件并非都是恶意的，大多数垃圾邮件都是为了发送广告。但其中一些垃圾邮件中也会包含指向恶意网站的恶意链接，这些网站可能部署了恶意软件与能够感染计算机的漏洞利用工具包。如今，大多数邮件服务提供商与各种基于网络的反病毒产品都提供了良好的垃圾邮件过滤功能，这可以大大减少垃圾邮件的数量。有些垃圾邮件仍然可以逃避检测，如图　6- 7 所示通过钓鱼窃取用户信息。

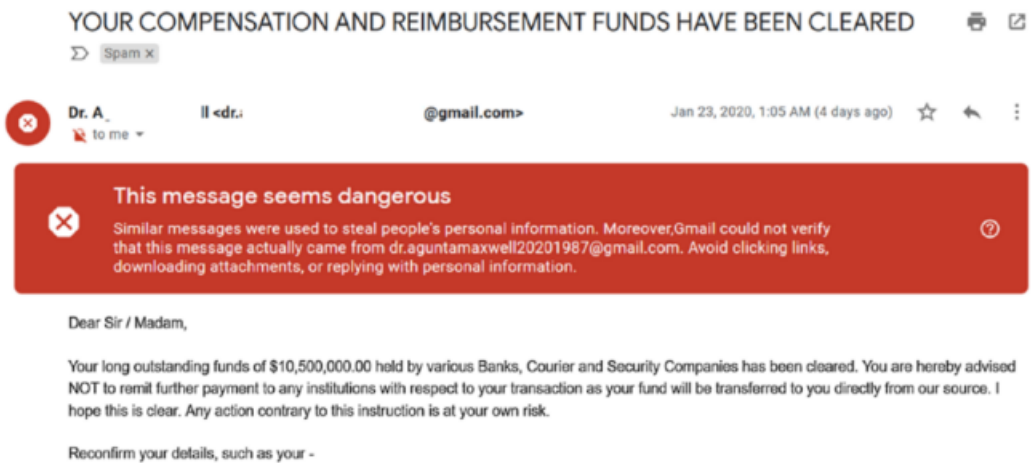

图 6- 7 试图通过钓鱼窃取用户信息的垃圾邮件示例

大多数垃圾邮件都经过精心设计，采用能够吸引用户阅读并点击内容、下载并运行附件的相关主题。例如，攻击者常用的主题有：

- 发票
- 退款
- 彩票/奖金
- 同事/老板/经理发送的邮件
- 朋友/配偶发送的邮件

垃圾邮件有许多变种，基本都是为了收集信息并将恶意软件安装到失陷主机上。其中一些被称为 phishing、whaling、spear phishing、clone phishing，读者可以自行搜索有关这些攻击方式的信息，了解这些垃圾邮件变种与常规垃圾邮件的差异。

受感染的存储设备

受感染的存储设备是恶意软件的一种分发机制，此攻击方式对于气隙网络（air-gapped）中的设备特别有用。使用这种分发机制的恶意软件中，最典型的就是震网（Stuxnet），该恶意软件对伊朗的核计划造成了严重破坏。

当设备的使用场景中，互联网不是设备间交换数据的主要途径，而是通过 USB 驱动器、CD

驱动器和硬盘驱动器承担数据共享的功能时，该攻击方式受到攻击者的青睐。

原因在于 Windows 操作系统提供了一种机制，当连接磁盘或存储设备（如 USB 驱动器或 CD 驱动器）时，操作系统会自动在连接的存储设备中运行名为 autorun.inf 的脚本文件。提供该功能是为了方便用户，在将 DVD 插入时自动播放影音内容。此外，该功能的另一个常见用途是在插入 CD 时自动运行安装程序以安装软件。

这种机制已经被攻击者滥用，攻击者将恶意软件或者恶意脚本存放在 USB 驱动器、硬盘驱动器等同类存储介质中，并在驱动器中放置 autorun.inf 文件利用该机制进行攻击。在驱动器插入计算机时，操作系统就会自动执行恶意脚本或恶意软件，如代码 6- 1 所示。

代码 6- 1 执行存储介质中 malware.exe 文件的 autorun.inf 文件

```
[autorun]
open=malware.exe
```

将带有 autorun.inf 文件的磁盘驱动器插入计算机时，操作系统会自动执行 autorun.inf 文件中的命令，该命令会运行同一磁盘驱动器中的恶意文件 malware.exe。

微软后续在 Windows 7 中禁用了自动运行功能，并且通过软件更新在旧版本的 XP 与 Vista 操作系统中也禁用了此功能。但是在 IT 行业、医疗行业以及其他各种行业和小型企业仍在运行着启用该功能的旧版本 Windows 操作系统，这使得它们很容易受到这种恶意软件分发机制的攻击。

当然，恶意软件并不一定必须通过自动运行功能才能在气隙网络中进行传播。已经入侵失陷主机的恶意软件可以等待 USB 驱动器或磁盘驱动器连接到系统上时，将自身复制到驱动器中。这些驱动器在与其他用户进行共享或是被其他用户使用时，这些用户可能会不小心点击这些样本文件，从而触发感染。

恶意广告

互联网时代，企业触达消费者最常见的方式就是在线广告，像谷歌这样的大型广告公司能够

确保企业投放的广告被数百万用户阅读与点击。广告是许多网站的主要收入来源，运营者通过在线广告对网站内容进行变现。在有用户访问网站时，相关的广告就会自动展现给用户。在线广告也可以作为一种分发机制，企业投放良性广告的渠道也可以被攻击者滥用投放恶意广告，攻击者能够通过多种手段使得广告服务提供商使用恶意内容替换良性广告。通常来说，广告服务提供商并不会验证投放的广告内容是良性的还是恶意的，这就会使攻击者投放的恶意广告分发给数以百万计的用户。

恶意广告中的恶意内容各有不同，从指向部署恶意软件的下载链接到指向漏洞利用工具包的登陆页面或是指向其他失陷网站的链接。

“搭便车”下载（Drive-by Download）

“搭便车”下载也是一种恶意软件分发机制，恶意软件在受害者不知情且许可的情况下被下载到受害者的设备上执行。使用这种分发机制的首先是此前介绍的漏洞利用工具包，会在用户毫不知情的情况下下载并安装恶意软件。此外，还有一些通过“搭便车”下载这种恶意软件分发机制的案例。

互联网上有很多声称可以清理电脑并提高电脑运行速度的工具，如果下载并安装了这些工具，很可能就会被安装其他类型的恶意软件。此类工具通常都与某些恶意软件捆绑在一起，在安装主要工具的同时，“搭便车”的恶意软件也会在不通知用户的情况下就被自动安装到计算机上。该分发机制主要被广告恶意软件与窃密恶意软件所使用，它们通常会搭其他程序的“便车”，在未经受害者同意的情况下直接安装到计算机上。

Downloaders

恶意软件攻击链通常由一组旨在感染系统的恶意二进制文件组成，通常是起始于被称为

Downloader 的恶意软件。Downloader 是一种通用的恶意软件，通常是投递给受害者的第一个恶意软件，其主要任务是下载恶意软件或者 Payload。Downloader 可以以各种形式存在，例如 PE 文件、Microsoft Word 文档文件、PDF 文档文件、脚本文件等。

弱口令登陆

攻击者分发恶意软件最常见的方法就是直接登录到具有弱口令或者没有身份认证的服务器，再下载并执行恶意软件。全世界部署的很多软件和工具都可以通过互联网访问，各种形式的弱口令与没有身份认证的情况广泛存在。

- 许多工具都带有默认凭据，通常在初始设置时需要更改，但很多人在部署时都忘了更改。
- 某些工具没有默认凭据，通常在初始设置时需要设置，但很多人在部署时忘了设置。
- 某些工具没有配置默认身份验证方案，通常在初始设置时需要设置，但很多人在部署时忘了设置。
- 某些用户使用类似于 admin/admin123 类的弱口令，使得攻击者很容易猜到。例如，许多 SSH 服务都使用强度不高、可以猜测的默认密码，这为攻击者入侵留下了极为方便的途径。

由于很多都存在没有身份验证方案或者存在弱口令的问题，攻击者不断在互联网上寻找能够容易登录的服务器。登录后，攻击者可以使用 wget 等系统上原生的应用程序下载并执行恶意软件。攻击者通常会将整个过程自动化，从而加快在互联网上寻找错误配置服务器的速度。名为《Container Malware: Miners Go Docker Hunting in the Cloud》的文章中介绍了攻击者通过网络上配置错误的 Docker 服务来进行恶意软件分发的案例。

共享文件夹

Windows 操作系统提供了通过网络共享文件夹的功能，系统使用 SMB 协议来实现该功能。可以使用 Windows 文件资源管理器查看网络上其他设备的共享文件夹，如图 6- 8 所示。

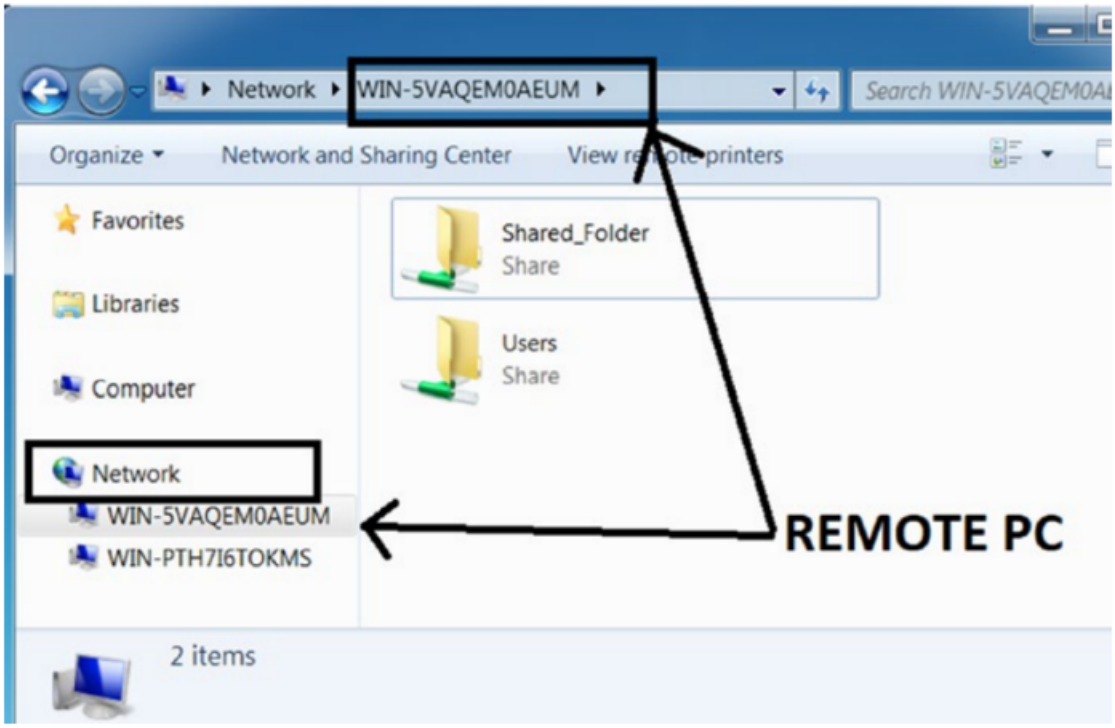

图 6- 8 Windows 文件资源管理器查看网络上其他设备的共享文件夹

选择共享的文件夹对网络上其他用户是可见的。在较新版本的 Windows 操作系统中，默认不可以直接查看另一台远程计算机的共享文件夹，需要使用该计算机的凭据进行身份验证才能够访问。还是有很多用户甚至是系统管理员会在不需要远程用户进行身份验证的前提下启用共享文件夹，他们都没有意识到这可能产生严重的后果。

当用户启用共享文件夹并开启了写入权限时，网络上的任何人都可以将任何文件写入共享文件夹，这样会更加危险。众所周知，恶意软件通常会寻找具有写入权限的共享文件夹来入侵网络上的其他计算机。通过此类共享文件夹，恶意软件将自身复制到文件夹中。恶意软件也会尝试窃取其他用户或者管理员的域权限，使用该权限访问网络上其他计算机的共享文件夹。有时候，用户在远程计算机上发现一个没有见过的新文件会尝试单击查看其功能，共享文件

夹中的恶意软件就会借机执行。为了诱使用户点击这些文件，通常会在共享文件夹中放置恶意 Word 文件与恶意 PDF 文件等此类容易被用户点击的文件。

恶意软件一旦复制到共享文件夹中并被执行，就可以使用各种方式强制远程计算机执行文件，部分方式如下所示：

- 使用 sc.exe 在远程计算机上注册服务，此前在第 5 章中进行了介绍，如代码 6- 2 所示。
- 分析人员可以使用 ProcMon 等动态分析工具进行发现，确定恶意软件是否在网络上进行传播，调用的参数中会使用双反斜杠\\引用远程计算机，可以借此区分是在本地计算机与远程计算机上注册服务。
- 使用 API Miner 与 API 日志确定恶意软件是否使用对应 API，为了区分 API 是在远程计算机还是本地计算机上使用，需要进一步检查传递给 API 的参数。
- 使用 PsExec 执行，分析人员可以使用 ProcMon 等动态分析工具进行发现。

代码 6- 2 sc.exe 在远程计算机上注册服务

```
sc.exe \\<Remote_Machine> create newservice binpath= C:\Path\To\
Shared_Malware.exe start= auto obj= <username> password= <passwor
d>
```

总结

恶意软件与良性软件其实也没有什么区别，都是使用各种技术进行开发的程序。本章介绍了构成恶意软件的各种组件，这些组件为恶意软件提供了所需要的各种功能支持。创建恶意软件只是攻击者工作的一部分，恶意软件也需要最终分发给受害者，本章中对各种投递与感染受害者的分发机制也进行了介绍。