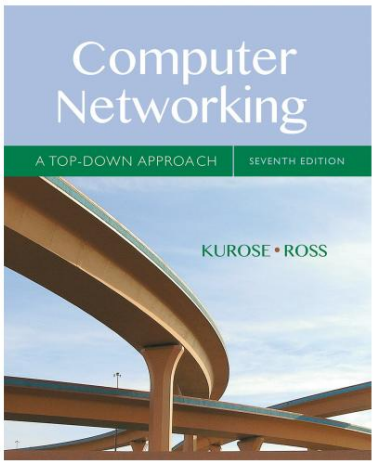
Wireshark 实验室:HTTP v7.0

计算机网络补充:自上而下的方法,第 7版， JF Kurose 和 KW Ross



“告诉我,我就忘了。给我看,我记得。让我参与进来,我理解。”中国谚语

在介绍性实验室中使用 Wireshark 数据包嗅探器弄湿了我们的脚后,我们现在准备使用 Wireshark 来调查运行中的协议。在本实验中,我们将探索 HTTP 协议的几个方面:基本的 GET/响应交互、HTTP 消息格式、检索大型 HTML 文件、检索带有嵌入对象的 HTML 文件以及 HTTP 身份验证和安全性。在开始这些实验之前,您可能需要查看文本的第 2.2 节。[1]

1.基本的HTTP GET/response交互

让我们通过下载一个非常简单的 HTML 文件开始我们对 HTTP 的探索 一个非常短且不包含嵌入对象的文件。请执行下列操作:

1. 启动您的网络浏览器。
2. 启动 Wireshark 数据包嗅探器,如介绍实验室中所述（但尚未开始数据包捕获）。在 display-filter-specification 窗口中输入“http”（只是字母,而不是引号）,以便稍后在 packet-listing 窗口中仅显示捕获的 HTTP 消息。 （我们这里只对HTTP协议感兴趣,不想看到所有抓包的杂乱无章）。
3. 等待一分钟多一点（我们很快就会知道原因）,然后开始 Wireshark 数据包捕获。
4. 在浏览器中输入以下内容 http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html 您的浏览器应该显示非常简单的单行 HTML 文件。
5. 停止 Wireshark 数据包捕获。

---

[1] 对图表和章节的引用来自我们文本的第 7版,计算机网络,一种自上而下的方法,第 7版,JF Kurose 和 KW Ross,Addison-Wesley/Pearson,2016。

您的 Wireshark 窗口应该类似于图 1 中所示的窗口。如果您无法在实时网络连接上运行 Wireshark,您可以下载在执行上述步骤时创建的数据包跟踪。2

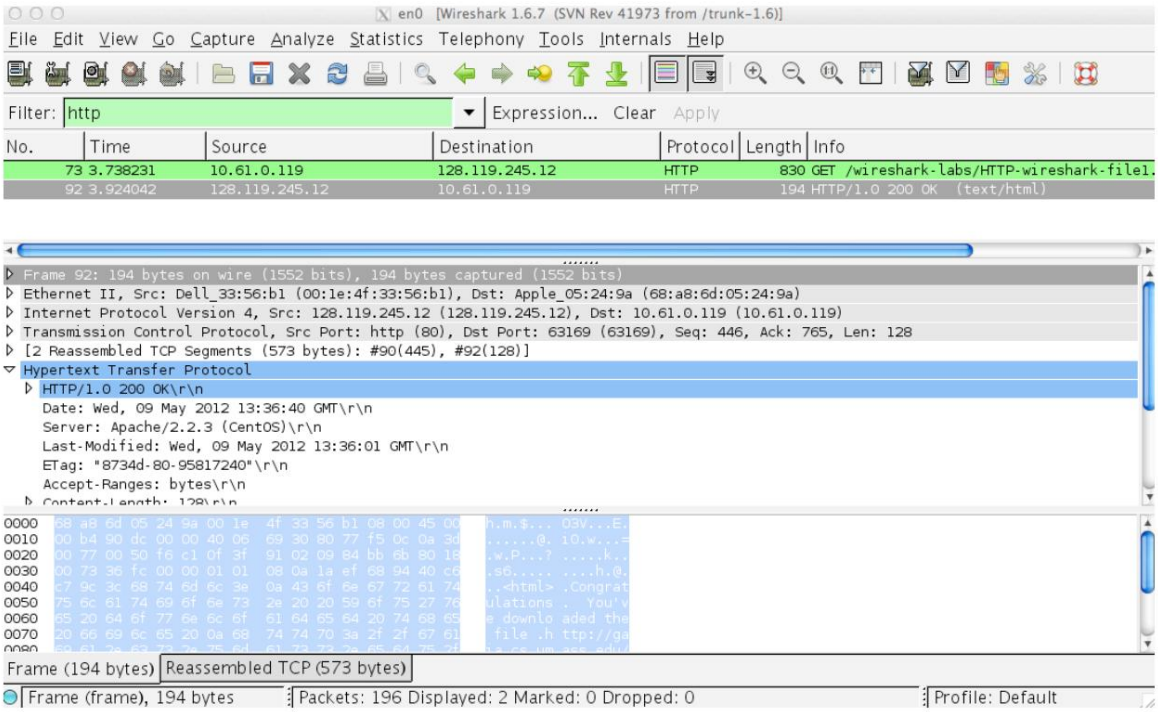


图 1:浏览器检索到 http://gaia.cs.umass.edu/wireshark-labs/ HTTP wireshark-file1.html 后的 Wireshark 显示

图 1 中的示例显示在数据包列表窗口中捕获了两条 HTTP 消息:GET 消息（从您的浏览器到 gaia.cs.umass.edu Web 服务器)和从服务器到您的浏览器的响应消息。数据包内容窗口显示所选消息的详细信息（在本例中为 HTTP OK 消息,在数据包列表窗口中突出显示）。回想一下,由于 HTTP 消息是在 TCP 段中携带的,而 TCP 段是在 IP 数据报中携带的,而 IP 数据报是在以太网帧中携带的,Wireshark 也会显示帧、以太网、IP 和 TCP 数据包信息。我们希望尽量减少显示的非 HTTP 数据量（我们在这里对 HTTP 感兴趣,并将在以后的实验中研究这些其他协议）,因此请确保 Frame、Ethernet、IP 和TCP 信息有一个加号或一个右三角（表示有隐藏的、未显示的信息）,而 HTTP 行有一个减号或一个下三角（表示显示有关 HTTP 消息的所有信息）。

2 下载 zip 文件http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip并解压缩文件 http-ethereal-trace-1。此 zip 文件中的跟踪是由在作者的一台计算机上运行的 Wireshark 收集的,同时执行 Wireshark 实验室中指示的步骤。下载跟踪后,您可以将其加载到 Wireshark 并使用“文件”下拉菜单查看跟踪,选择“打开”,然后选择 http-ethereal-trace-1 跟踪文件。生成的显示应该类似于图 1。

（Wireshark 用户界面在不同操作系统和不同版本的 Wireshark 中显示略有不同）。

（注意:您应该忽略对 favicon.ico 的任何 HTTP GET 和响应。如果您看到对此文件的引用,则您的浏览器会自动询问服务器它（服务器)是否有一个应该显示在旁边的小图标文件在您的浏览器中显示的 URL。在本实验中,我们将忽略对这个讨厌的文件的引用。）。

通过查看 HTTP GET 和响应消息中的信息,回答以下问题。在回答以下问题时,您应该打印出 GET 和响应消息（有关如何执行此操作的说明,请参阅介绍性 Wireshark 实验室)并指出您在消息中的何处找到了回答以下问题的信息。当您提交作业时,请在输出中添加注释,以便清楚您在输出中获得答案信息的位置（例如,对于我们的课程,我们要求学生用钢笔标记纸质副本,或使用彩色字体的文本）。

1. 您的浏览器运行的是 HTTP 1.0 版还是 1.1 版?什么版本的 HTTP 是
   服务器运行?
2. 您的浏览器表明它可以接受哪些语言（如果有）
   服务器?
3. 你电脑的IP地址是什么？ gaia.cs.umass.edu 服务器的?
4. 服务器返回给浏览器的状态码是什么?
5. 您检索的 HTML 文件最后一次在服务器上修改是什么时候?
6. 多少字节的内容被返回到您的浏览器?
7. 通过检查数据包内容窗口中的原始数据,您是否看到数据中没有显示在数据包列表窗口中的任何标头?如果有,请命名
   一。

在您对上述问题 5 的回答中,您可能会惊讶地发现您刚刚检索到的文档在您下载该文档之前的一分钟内被最后一次修改。这是因为（对于这个特定文件）,gaia.cs.umass.edu 服务器将文件的最后修改时间设置为当前时间,并且每分钟这样做一次。因此,如果您在两次访问之间稍等片刻,该文件似乎最近被修改过,因此您的浏览器将下载该文档的“新”副本。

2. HTTP CONDITIONAL GET/response 交互

回想一下本文的第 2.2.5 节,大多数 Web 浏览器执行对象缓存,因此在检索 HTTP 对象时执行条件 GET。在执行以下步骤之前,请确保您的浏览器缓存为空。 （要在 Firefox 下执行此操作,请选择工具->清除最近历史记录并选中缓存框,或者对于 Internet Explorer,选择工具->Internet 选项->删除文件;这些操作将从浏览器的缓存中删除缓存的文件。)现在做以下：

- 启动您的网络浏览器,并确保清除浏览器的缓存,如上所述。
- 启动 Wireshark 数据包嗅探器
- 在浏览器中输入以下 URL
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html
  您的浏览器应该显示一个非常简单的五行 HTML 文件。

· 再次在浏览器中快速输入相同的 URL（或只需选择浏览器上的刷新按钮）

· 停止Wireshark 数据包捕获,并在display-filter-specification 窗口中输入“http”,以便稍后在数据包列表窗口中仅显示捕获的HTTP 消息。 · （注意:如果您无法在实时网络连接上运行 Wireshark,您可以使用 http-ethereal-trace-2 数据包跟踪来回答以下问题;请参阅脚注 1。此跟踪文件是在执行上述步骤时收集的在作者的一台计算机上。）

回答以下问题：

8. 检查从浏览器到服务器的第一个 HTTP GET 请求的内容。您是否在 HTTP GET 中看到“IF-MODIFIED-SINCE”行?
9. 检查服务器响应的内容。服务器是否明确返回
   文件的内容?你怎么知道?
10. 现在检查从浏览器到服务器的第二个 HTTP GET 请求的内容。您是否在 HTTP GET 中看到“IF-MODIFIED-SINCE:”行?如果是这样，“IF-MODIFIED-SINCE:”标头后面有什么信息?

11. 响应第二个 HTTP GET 从服务器返回的 HTTP 状态代码和短语是什么?服务器是否明确返回了文件的内容?
    解释。

3. 检索长文档

到目前为止,在我们的示例中,检索到的文档都是简单而简短的 HTML 文件。接下来让我们看看当我们下载一个长的 HTML 文件时会发生什么。请执行下列操作：

· 启动您的网络浏览器,并确保清除浏览器的缓存,如上所述。

· 启动 Wireshark 数据包嗅探器
· 在浏览器中输入以下 URL
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html
  您的浏览器应该会显示相当长的美国权利法案。
· 停止Wireshark 抓包,并在display-filter-specification 窗口中输入“http”,这样只会显示捕获到的HTTP 消息。 · （注意:如果您无法在实时网络连接上运行 Wireshark,您可以使用 http-ethereal-trace-3 数据包跟踪来回答以下问题;请参阅脚注 1。此跟踪文件是在执行上述步骤时收集的在作者的一台计算机上。）

在数据包列表窗口中,您应该会看到您的 HTTP GET 消息,然后是对您的 HTTP GET 请求的多数据包 TCP 响应。这种多包响应值得解释一下。回想一下 2.2 节（参见正文中的图 2.9）,HTTP 响应消息由一个状态行、后跟标题行、后跟空行和实体正文组成。对于我们的 HTTP GET，

响应中的实体主体是整个请求的 HTML 文件。在我们这里的例子中,HTML 文件相当长,并且 4500 字节太大而无法放入一个 TCP 数据包中。因此,单个 HTTP 响应消息被 TCP 分成几个部分,每个部分包含在一个单独的 TCP 段中（参见文本中的图 1.24）。在最近版本的 Wireshark 中,Wireshark 将每个 TCP 段表示为一个单独的数据包,并且单个 HTTP 响应在多个 TCP 数据包中分段这一事实由 Wireshark 显示的 Info 列中的“重新组装的 PDU 的 TCP 段”指示.早期版本的 Wireshark 使用“Continuation”短语来表示 HTTP 消息的全部内容在多个 TCP 段中被破坏。我们在这里强调 HTTP 中没有“Continuation”消息！

回答以下问题：

12. 您的浏览器发送了多少 HTTP GET 请求消息?跟踪中的哪个数据包编号包含 Bill 或 Rights 的 GET 消息？
13. 跟踪中哪个包号包含相关的状态码和短语
    对 HTTP GET 请求的响应？
14. 响应中的状态码和短语是什么？
15. 承载单个 HTTP 需要多少个包含数据的 TCP 段
    回应和权利法案的文本？

4. 带有嵌入对象的 HTML 文档

现在我们已经了解了 Wireshark 如何显示捕获的大型 HTML 文件的数据包流量,我们可以看看当您的浏览器下载带有嵌入对象的文件时会发生什么,即包含其他对象的文件（在下面的示例中,图像文件) 存储在另一台服务器上。

请执行下列操作：

- 启动您的网络浏览器,并确保清除浏览器的缓存,如上所述。

- 启动 Wireshark 数据包嗅探器
- 在浏览器中输入以下 URL
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html
  您的浏览器应该显示一个带有两个图像的简短 HTML 文件。这两个
  基本 HTML 文件中引用了图像。也就是说,图像本身不包含在 HTML 中;相反,图像的 URL 包含在下载的 HTML 文件中。正如教科书中所讨论的,您的浏览器必须从指定的网站检索这些徽标。我们的出版商徽标是从 gaia.cs.umass.edu 网站检索的。我们第五次的封面图片

  版本（我们最喜欢的封面之一)存储在 caite.cs.umass.edu 服务器上。
  （这是 cs.umass.edu 中的两个不同的 Web 服务器）。
- 停止Wireshark 抓包,并在display-filter-specification 窗口中输入“http”,这样只会显示捕获到的HTTP 消息。

- （注意:如果您无法在实时网络连接上运行 Wireshark,您可以
  使用 http-ethereal-trace-4 数据包跟踪来回答以下问题;请参见脚注 1。此跟踪文件是在
  作者的一台计算机上执行上述步骤时收集的。）

回答以下问题：

16. 您的浏览器发送了多少 HTTP GET 请求消息?这些 GET 请求发送到哪些 Internet 地址?

17. 你能分辨出你的浏览器是串行下载这两个图像的,还是从两个网站并行下载的?解释。

5 HTTP 认证

最后,让我们尝试访问一个受密码保护的网站,并检查为该网站交换的 HTTP 消息序列。 URL http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html 受密码保护。用户名是“wireshark-students”（不带引号）,密码是“network”（同样,不带引号）。因此,让我们访问这个“安全”的受密码保护的站点。请执行下列操作：

- 确保您的浏览器缓存已清除,如上所述,然后关闭
  你的浏览器。然后,启动浏览器
- 启动 Wireshark 数据包嗅探器
- 在浏览器中输入以下 URL
  http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark file5.html
  在弹出框中输入请求的用户名和密码。
- 停止Wireshark 数据包捕获,并在display-filter-specification 窗口中输入“http”,以便稍后在
  数据包列表窗口中仅显示捕获的HTTP 消息。 · （注意:如果您无法在实时网络连接上运行
  Wireshark,您可以使用 http-ethereal-trace-5 数据包跟踪来回答以下问题;请参阅脚注 2。此跟踪文件是在执行上述步骤时收集的在作者的一台计算机上。）

现在让我们检查一下 Wireshark 的输出。您可能想首先阅读http://frontier.userland.com/stories/storyReader$2159上有关“HTTP 访问身份验证框架”的易于阅读的材料来了解 HTTP 身份验证

回答以下问题：

18.服务器对初始响应的响应（状态码和短语)是什么
来自浏览器的 HTTP GET 消息?

19. 当您的浏览器第二次发送HTTP GET 消息时,HTTP GET 消息中包含哪些新字段?

您输入的用户名 (wireshark-students) 和密码 (network) 以字符串
(d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=) 进行编码

客户端 HTTP GET 消息中的“Authorization: Basic”标头。虽然您的用户名和密码看起来可能已加密,但它们只是以一种称为 Base64 格式的格式进行编码。用户名和密码未加密!要查看此内容,请访问http://www.motobit.com/util/base64-decoder-encoder.asp并输入 base64 编码的字符串 d2lyZXNoYXJrLXN0dWRlbnRz 并进行解码。瞧!您已从 Base64 编码转换为 ASCII 编码,因此应该会看到您的用户名!

要查看密码,请输入字符串 Om5ldHdvcms= 的其余部分,然后按解码。由于任何人都可以下载像 Wireshark 这样的工具并嗅探通过其网络适配器的数据包(不仅仅是他们自己的),并且任何人都可以将 Base64 转换为 ASCII(您刚刚做到了!),您应该清楚 WWW 上的简单密码除非采取额外措施,否则网站并不安全。

不要害怕!正如我们将在第 8 章中看到的,有一些方法可以使 WWW 访问更加安全。
但是,我们显然需要一些超出基本 HTTP 身份验证框架的东西!