

初识 WEB - Lab 00

在这个 Lab 中，我们将了解 Web 的一些常用信息及获取信息的工具。

Lab所在地址为:<https://github.com/fudansswebfundamental/fdu-19ss-web-lab>

今后的lab也均会在此仓库下发布，请各位同学熟悉GitHub的使用

在此推荐一份文档

(<https://github.com/fudansswebfundamental/Docs/blob/master/%e5%86%99%e7%bb%99%e6%96%b0%e6%89%8b%e7%9a%84%20Github%20%e6%8c%87%e5%8d%97.md>)请大家务必仔细阅读&动手尝试，如有疑问可以及时提出

这次的作业为多个截图，请做成一个 PDF 文件。

你的 IP 地址 (IP address)

1. 不同的操作系统使用不同工具：

Windows 7: 点击 Windows 徽标，在开始菜单下的搜索栏中输入“cmd”，点开命令提示符。

Windows 10: 右击 Windows 徽标，选择“Windows PowerShell(I)”。

macOS / OS X: 通过 Spotlight Search 输入“terminal”打开 Terminal。或者打开 Launchpad，在 Other 文件夹下点击 Terminal。

Linux: 打开 Terminal。(用 Linux 的人找不到 Terminal 的话请换以上其他操作系统。)

2. 在新开的窗口中输入命令以获得你的 IP 地址。因为大多数电脑都有着多个适配器 (adapter)，所以对于每个安装和配置的设备都会有相应的 IP 信息。这些设备可能是有线连接的，也可能是无线连接的，但任何时候只有一个正在被使用，我们所要寻找的就是这个适配器。输出的内容虽多，但会提供许多有用的信息。

Windows: 在命令提示符或者 PowerShell 中输入 `ipconfig`。结果样例：

无线局域网适配器 WLAN：

```
连接特定的 DNS 后缀 . . . . . :  
IPv4 地址 . . . . . : 192.168.1.106  
子网掩码 . . . . . : 255.255.255.0  
默认网关 . . . . . : 192.168.1.1
```

macOS / OS X / Linux: 在 Terminal 中输入 `ifconfig`，若失败则尝试 `/sbin/ifconfig`。结果样例：

```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500  
ether ac:bc:32:ac:c6:ed
```

```
inet6 fe80::10e8:b017:2689:c45e%en0 prefixlen 64 secured
scopeid 0x5
inet 192.168.1.101 netmask 0xffffffff00 broadcast 192.168.1.255
nd6 options=201<PERFORMNUD,DAD>
media: autoselect
status: active
```

3. 这时的你很可能会发现你拥有一个内部 IP 地址，直接连接到因特网 (internet) 则另当别论。

这些 IP 地址都是**内部地址**： 10.0.0.0 - 10.255.255.255 172.16.0.0 - 172.31.255.255 192.168.0.0 - 192.168.255.255

若是如此，我们需要确定我们究竟在和别人共用着哪个 IP 地址，因为这个 IP 地址才是被这个内网 (internal network) 外的人所“看见”的地址。查询的方法很简单，可以通过教材提供的地址 <http://examples.funwebdev.com/ipaddress.php> 或者直接在百度中搜索“ip地址”即可查询到。

注：在你的本科学习过程中，更建议你使用 Google 作为更常用的搜索工具，但这门课的资料丰富，大多数内容也可以被百度得到。在此处的查询中，如果使用了虚拟网络代理 (VPN)，则会获得该代理的 IP 地址。

每次你访问一个 web 页面的时候，你都会将你的外部 IP 地址传输出去来获得返回的数据。同样的，每一个访问你的页面的访客都会给你发送他们的 IP 地址。

作业要求

Windows 用户要求仅提交有 IP 地址的那个部分截图，请不要保留含“媒体已断开”或“媒体已断开链接”的部分。macOS / OS X / Linux 用户提交“status”为“active”的部分的截图，其他部分请不要保留。

分析一个网页的组成部分

此处推荐使用 Chrome 浏览器，但依然会提供 Safari 和 Microsoft Edge 两款浏览器的操作流程。只需完成其中之一即可（**建议大家尽量使用Chrome**）：

Chrome:

1. 打开网页 <http://examples.funwebdev.com/ipaddress.php>
2. 右击页面，点击“检查(N)” (Inspect)。
3. 你会看见一个开发者工具的面板在网页右侧或者是一个独立窗口。Style 下展示的是 Element 下选中的 HTML 标签的层叠样式表 (CSS)，自定义的 CSS 在这里可以做临时的编辑。
4. 展开左边所有的 HTML 标签，将整个网页内容连同开发者工具页面一同截图。
5. 选中 Network 标签，然后刷新页面。这里会看到一些这个页面使用的不同的资源。
6. 打开百度页面，你会看到一个真实的网页会用到的不同的资源，这个面板也会显示加载时长等一些有用的信息。
7. 关闭面板。

Safari:

1. 打开网页 <http://examples.funwebdev.com/ipaddress.php>

2. 右击页面，点击“检查元素” (Inspect Element)，或者选取“开发” (Develop)，然后选取“显示网页检查器” (Show Web Inspector)，选取 Element 标签。
3. 你会看见一个网页检查器 (Web Inspector) 的面板在网页右侧或者是一个独立窗口。Properties 下展示的是 Elements 下选中的 HTML 标签的层叠样式表 (CSS)，自定义的 CSS 在这里可以做临时的编辑。
4. 展开左边所有的 HTML 标签，将整个网页连同网页检查器页面一同截图。
5. 选中 Networks 标签，然后刷新页面。这里会看到一些这个页面使用的不同的资源。
6. 打开百度页面，你会看到一个真实的网页会用到的不同的资源，这个面板也会显示加载时长等一些有用的信息。
7. 关闭面板。

Microsoft Edge:

1. 打开网页 <http://examples.funwebdev.com/ipaddress.php>
2. 右击页面，点击“检查元素” (Inspect Element)，或者选取“设置及更多”，然后选取“F12 开发人员工具”，选取元素 (Element) 标签。
3. 你会看见一个 F12 开发人员工具的面板在网页下侧或者是一个独立窗口。样式 (Style) 下展示的是元素下选中的 HTML 标签的层叠样式表 (CSS)，自定义的 CSS 在这里可以做临时的编辑。
4. 展开左边所有的 HTML 标签，将整个网页连同 F12 开发人员工具页面一同截图。
5. 选中网络 (Network) 标签，然后刷新页面。这里会看到一些这个页面使用的不同的资源。
6. 打开百度页面，你会看到一个真实的网页会用到的不同的资源，这个面板也会显示加载时长等一些有用的信息。
7. 关闭面板。

作业要求

以上三种浏览器，请选择一种完成作业，截下相应的图片。

域名服务器 (Name server)

1. 操作系统提供了一些工具让我们可以对网站做一些网络查询，来获得他们的域名服务器和名字解析。
2. 比较常用的工具有 nslookup (所有操作系统)，dig (Linux 和 macOS / OS X)，在这一部分我们将尝试运行 nslookup。在这一小节中，我们将查询 A 地址记录 (IPv4)：

Windows: `nslookup google.com` 或者 `nslookup -qt=A google.com`

输出样例：

```
服务器:  promote.cache-dns.local
Address:  221.131.143.69

非权威应答:
名称:      google.com
Address:   172.217.2.206
```

macOS / OS X: `nslookup google.com` 或者 `nslookup -type=A google.com`

输出样例：

```
Server:      221.131.143.69
Address:     221.131.143.69#53
```

```
Non-authoritative answer:
Name:   google.com
Address: 172.217.2.206
```

Linux: 安装命令: CentOS: `yum install bind-utils` Debian: `apt-get install dnsutils`

注: 安装之前先更新软件包管理器, 非 root 用户需要 sudo 权限

查询命令: `nslookup google.com` 或者 `nslookup -type=A google.com`

输出样例:

```
Server:      108.61.10.10
Address:     108.61.10.10#53
```

```
Non-authoritative answer:
Name:   google.com
Address: 172.217.2.78
```

3. 有时你会得到多个地址来响应对该网站的查询:

输出样例:

```
Server:      108.61.10.10
Address:     108.61.10.10#53
```

```
Non-authoritative answer:
Name:   google.com
Address: 173.194.33.37
Name:   google.com
Address: 173.194.33.46
Name:   google.com
Address: 173.194.33.41
Name:   google.com
Address: 173.194.33.40
Name:   google.com
Address: 173.194.33.33
Name:   google.com
Address: 173.194.33.39
Name:   google.com
Address: 173.194.33.36
Name:   google.com
Address: 173.194.33.32
```

4. 你服务器的输出内容的第一部分往往是你的本地DNS解析器 (DNS resolver)。输出的答案极可能是正确的，但我们可以深入研究看看 Google 认为它的 IP 地址应该是什么，而不必依赖我们路由器给的回答。要确定 google.com 的域名服务器则输入：

Windows: `nslookup -qt=ns google.com` **macOS / OS X / Linux:** `nslookup -type=ns google.com`

返回的信息为：

```
Non-authoritative answer:
google.com      nameserver = ns3.google.com.
google.com      nameserver = ns2.google.com.
google.com      nameserver = ns4.google.com.
google.com      nameserver = ns1.google.com.
```

5. 现在我们可以直接查询 Google 的官方域名服务器认为 google.com 的 IP 地址应该是什么，这时我们利用上一小节的查询结果。`nslookup google.com ns2.google.com` 这使得直接从 Google 的授权服务器(Authoritative NameServer)中查询，同时也有可能更新我们的缓存(cache)。当我们运行这命令的时候，可能会得到不同的结果，但我们的缓存会以更新来反映这些更改。

输出样例：

```
Name:   google.com
Address: 216.58.200.46
Name:   google.com
Address: 172.217.24.14
Name:   google.com
Address: 172.217.27.142
Name:   google.com
Address: 172.217.24.14
Name:   google.com
Address: 216.58.200.238
```

6. 其他类型的 DNS 记录的查询方式也是类似的，如邮件记录。例如，要查看托管在 google.com 的邮件地址，可以查询 `nslookup -type=mx google.com ns1.google.com` (Windows 下为 `nslookup -qt=mx google.com ns1.google.com`)。不同的参数可以获得不同的返回结果。使用指南：

Windows: [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb490721\(v%3dtechnet.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb490721(v%3dtechnet.10)) **macOS / OS X / Linux:** 使用命令 `man nslookup`

注：Windows 4.0 的组件 DnsClient 也提供了类似的功能。

作业要求

使用 nslookup：

1. 查询 baidu.com 的 A 地址记录截图；
2. 查询 baidu.com 的 域名服务器截图；
3. 使用授权服务器查询 baidu.com 的 IP 地址的截图；

4. 查询 114.114.114.114 匹配的主机名的截图。

观察 HTTP 标头 (HTTP headers)

1. 使用 Chrome (/ Safari / Microsoft Edge) 浏览器打开开发者工具面板 (Safari 为网页检查器, Microsoft Edge 则是 F12 开发人员工具), 点击 Network (网络)。
2. 打开 <http://www.baidu.com>。
3. 点击任意一条信息, 观察右侧的“Headers” (Microsoft Edge 显示“标头”) 标签 (Safari 则是 Resource) 。
4. 截下 user-agent 的完整信息的图。

作业要求

提交截图。

追踪数据包 (Tracing a packet)

接下来介绍一个强大的工具, 可以跟踪一个包从计算机到 Web 主机的路由过程。Windows 下使用 **tracert**, 而 macOS / OS X / Linux 使用 **traceroute**。现在尝试从你的计算机到 taobao.com 的路线:
Windows: **tracert taobao.com** **macOS / OS X / Linux:** **traceroute taobao.com** 输出的结果与很多因素有关, 但输出结果大致是这样的:

```
1  promote (192.168.1.1)  5.272 ms  2.971 ms  2.608 ms
2  100.65.152.1 (100.65.152.1)  9.150 ms  2.456 ms  6.567 ms
3  221.181.243.113 (221.181.243.113)  4.101 ms  3.312 ms  3.706 ms
4  221.181.146.66 (221.181.146.66)  3.924 ms  17.730 ms  5.130 ms
5  221.181.146.61 (221.181.146.61)  11.801 ms  5.372 ms  5.445 ms
6  183.207.204.81 (183.207.204.81)  9.372 ms * *
7  41.22.207.183.static.js.chinamobile.com (183.207.22.41)  17.522 ms
   45.22.207.183.static.js.chinamobile.com (183.207.22.45)  10.152 ms
   41.22.207.183.static.js.chinamobile.com (183.207.22.41)  10.994 ms
8  221.183.59.53 (221.183.59.53)  10.109 ms  11.868 ms  9.514 ms
9  221.183.40.137 (221.183.40.137)  11.562 ms  16.361 ms  35.975 ms
10 221.183.39.138 (221.183.39.138)  13.798 ms  15.047 ms  12.831 ms
11 (211.136.189.6)  14.833 ms  16.812 ms  17.546 ms
12 * * *
13 * 140.205.50.246 (140.205.50.246)  62.621 ms
   42.120.241.34 (42.120.241.34)  15.117 ms
14 42.120.247.149 (42.120.247.149)  12.725 ms
   140.205.27.18 (140.205.27.18)  26.068 ms
   123.56.34.241 (123.56.34.241)  24.858 ms
15 * china.alibaba.com (119.38.215.1)  33.939 ms
   119.38.214.249 (119.38.214.249)  23.534 ms
```

在输出中可以看到整个请求到达 taobao.com 所经历的路径。在演示中经历了 15 次跳跃, 其中第 12 条对于我们来说是隐藏的。输出的每行都是到目的地路径上的一个跃点, 并包含了一些关于请求路径的信息。不同的电脑的输出一定是不同的。

```
9  221.183.40.137 (221.183.40.137)  11.562 ms  16.361 ms  35.975 ms
```

在这一条中可以看到这个请求是通过 221.183.40.137 这个服务器，并且这个路由器话费的时间在 11.562 ms 到 35.975 ms 之间。如果列出了域名，说明这个域名可用。如果没有，则显示 IP 地址。一些路由器将仅显示 * * * 这表示访问出现问起，或者服务器配置了防火墙设置以防止路由器回复信息。在繁忙的路由器上，优先级较低的 ICMP 请求 (这是 traceroute 使用的) 将被丢弃。

在 macOS / OS X / Linux 上可以使用 `whois` 命令查询域名，如 `whois github.com`，可以获得相关域名的信息。如果没有办法通过命令行查询，可以访问 <https://www.whois.com> 或者 <https://whois.aliyun.com> (中国万网) 查询。

追踪路线使得我们可以确定谁可以在理论上改变一个数据包，或找出延迟 (delay) 来源。如果您意识到许多提供商在将您的数据包发送给 Google 并返回时发挥作用，那么整个网络中立性问题 (Network Neutrality) 就会变得更有意义。

注：My traceroute 将 ping 命令和 traceroute 命令合并，在 macOS / OS X / Linux 上使用更加方便，运行需要 sudo 或 root 权限。

作业要求

1. 跟踪一个从你的计算机发往 microsoft.com 的数据包，并截图。
2. 利用 whois 查询 tencent.com 的相关信息，并截图 (网页或命令行都需要所有相关信息)。

作业提交要求

你的 PDF 文件必须包含至少 5 张截图，分别对应以上五个部分 (每个部分计 20% 的分数)，完成后作为一个 PDF 文件上传至 github。文件名称为 `[学号]_[姓名].pdf` (如：16302010043_龙京奇.pdf)。提交截止时间为 **2020 年 3 月 1 日 23:59:59**，逾期每日扣除该 LAB 总分的 20% 分数直至 0 分为止。

作业提交网址

`//github.com/(Your ID)/SOFT130002_lab`

e.g. `//github.com/mondaylord/SOFT130002_lab`

即建立一个名为 `SOFT130002_lab` 的仓库上传

文件结构

`root` 即直接打开仓库所见的层，或写作 `/`

```
/lab0
  /16302010043_龙京奇.pdf
  /images
    /images/nslookup.jpg
    /images/whois.jpg
    ...
```

参考资料

1. Randy Connolly, Ricardo Hoar, "LAB 1 - HOW THE WEB WORKS" in *Fundamentals of Web Development* 1st ed., Addison-Wesley, 2014.

LAB 相关

有任何疑问欢迎通过超星讨论区或微信群聊联系助教。