第四章 使用 Fiddler 测试接口安全

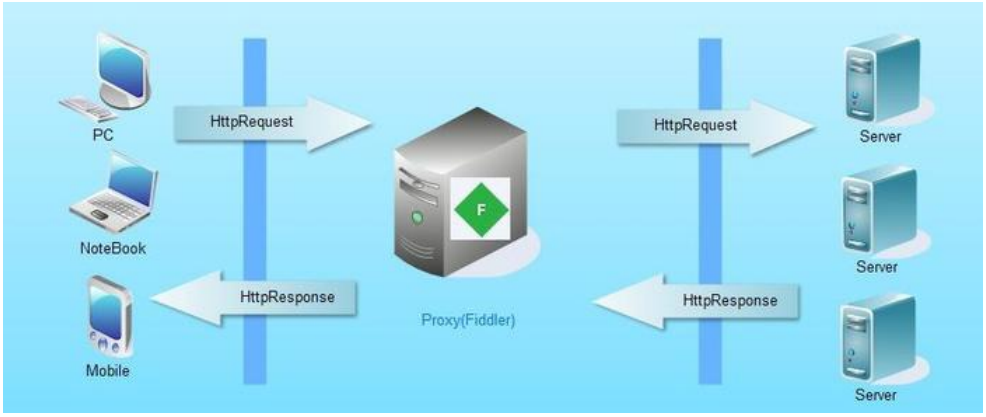
一、 Fiddler 基础

1 什么是 Fiddler

Fiddler 是强大且好用的 Web 调试工具之一，它能记录客户端和服务器的 http 和 https 请求，允许你监视，设置断点，甚至修改输入输出数据。

2 Fiddler 的运行机制

- 本机上监听 8888 端口的 HTTP 代理。
- 对于 PC 端 Fiddler 启动的时候默认 IE 的代理设为了 127.0.0.1:8888，而其他浏览器是需要手动设置的，所以如果需要监听 PC 端 Chrome 网络请求，将其代理改为 127.0.0.1:8888 就可以监听数据了。

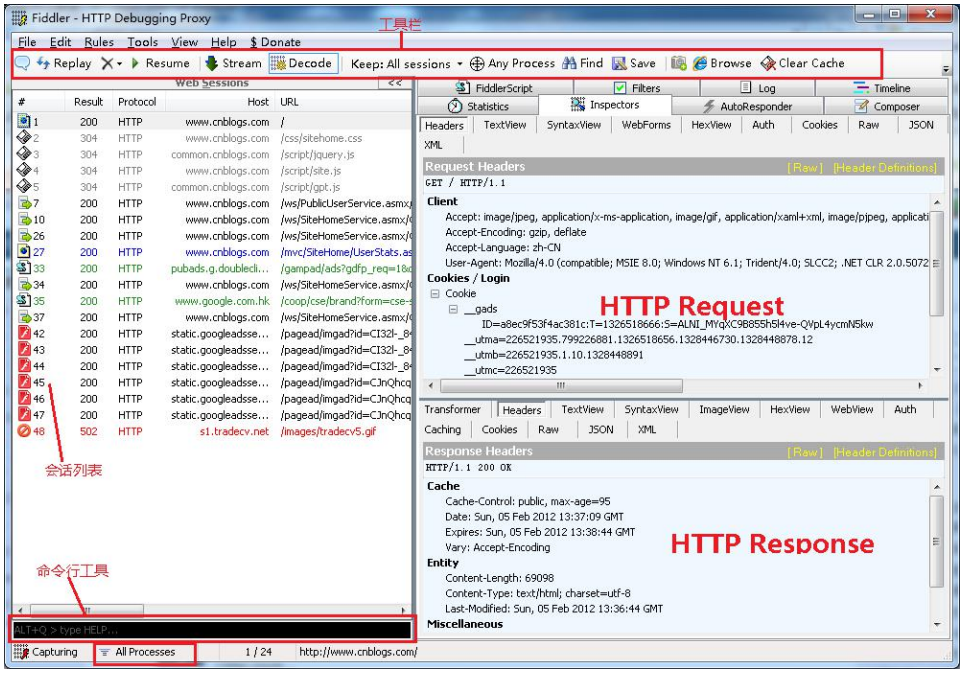


二、 安装 Fiddler

- 安装 NDP451
  - ✓ Microsoft .NET Framework 4.5.1
  - ✓ 是微软的一个基础接口程序
- 安装 fiddler4
- 安装 fiddlerSyntax

三、 Fiddler 的界面

1 Fiddler 主界面

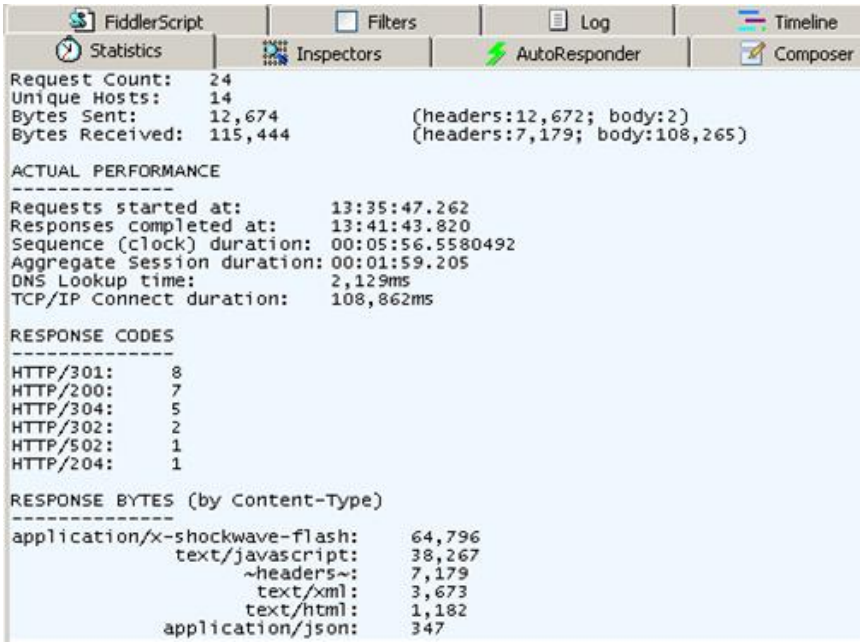


2 Fiddler 会话区

#	Result	Protocol	Host	URL	Body	Caching	Content-Type	Process	Comments	Custom
1	302	HTTP	192.168.0.238:1080	/webtours	0		text/html	iexplor…		
2	304	HTTP	192.168.0.238:1080	/webtours/	0		text/html	iexplor…		
4	200	HTTP	192.168.0.238:1080	/webtours/welcome.pl?sig…	630	no-cac…	text/html; c…	iexplor…		
6	304	HTTP	192.168.0.238:1080	/webtours/images/webtou…	0		text/html	iexplor…		
7	200	HTTP	192.168.0.238:1080	/webtours/nav.pl?in=home	1,393	no-cac…	text/html; c…	iexplor…		
9	304	HTTP	192.168.0.238:1080	/WebTours/images/mer_lo…	0		text/html	iexplor…		
10	200	HTTP	192.168.0.238:1080	/webtours/login.pl	405	no-cac…	text/html; c…	iexplor…		

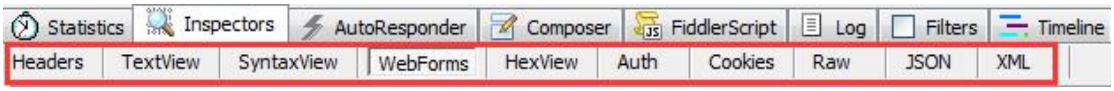
- #：顺序号，按照抓包的顺序从 1 递增
- Result：HTTP 状态码
- Protocol：请求使用的协议，如 HTTP/HTTPS/FTP 等
- HOST：请求地址的主机名或域名
- URL：请求资源的位置（路径）
- Body：请求大小
- Caching：请求的缓存过期时间或者缓存控制值
- Content-Type：请求响应的类型
- Process：发送此请求的进程
- Comments：备注
- Custom：自定义值

3 Fiddler 的 HTTP 统计视图



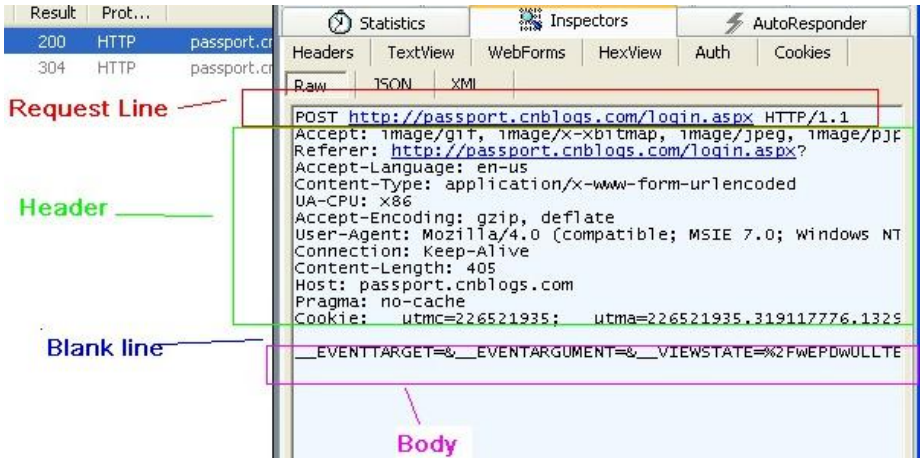
- 请求总数、请求包大小、响应包大小。
- 请求起始时间、响应结束时间、握手时间、等待时间、路由时间、TCP/IP、传输时间。
- HTTP 状态码统计。
- 返回的各种类型数据的大小统计以及饼图展现。

4 Fiddler 的监控面板（Inspector）



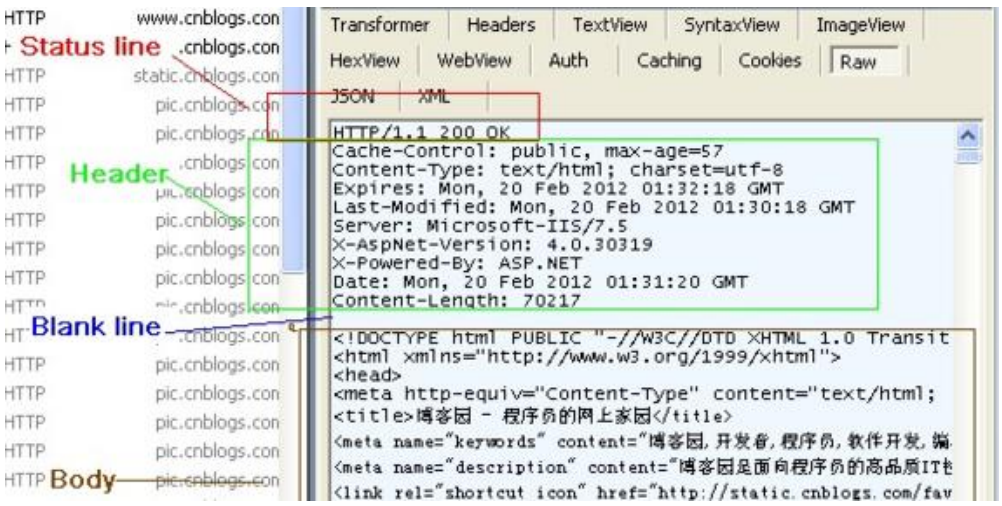
- Headers：信息头，若用手机和电脑打开的页面不一样，与此设置有关。
- TextView：以文本形式显示请求或响应的数据。
- SyntaxView：同 TextView，但有语法着色。
- WebForms：请求部分以表单形式显示所有的请求参数和参数值；响应部分与 TextView 内容是一样的。
- HEX：十六进制形式的数据。
- Auth：显示认证信息，如 Authorization。
- Cookies：显示所有 cookies。
- Raw：显示 Headers 和 Body 数据。
- JSON：请求或响应数据是 json 格式时，json 形式显示请求或响应内容。
- XML：请求或响应数据是 xml 格式，xml 形式显示请求或响应内容。

5 Fiddler 的 Request 消息结构



- POST：请求方式，HTTP/1.1 表示协议与版本
- Accept：浏览器端可接受的媒体类型
- Referer：告诉服务器是从哪个页面链接过来的
- Accept-Language：语言类型
- Accept-Encoding：压缩方法
- User-Agent：客户端使用的操作系统和浏览器的名称和版本
- Connection：网页打开后，客户端和服务器之间用于传输数据的 TCP 连接是否关闭，keep-alive 表示不会关闭，客户端再次访问这个服务器上的网页，会继续使用这一条已经建立的连接
- COOKIE：将 cookie 值发送给服务器

6 Fiddler 的 Response 的消息结构

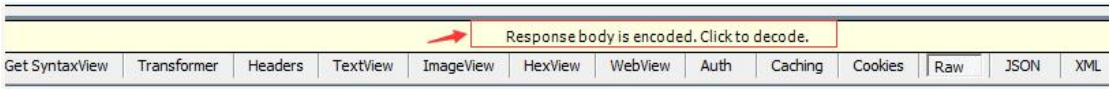


- HTTP/1.1：协议，200：状态码，OK 响应消息文本
- Cache-Control：private 的消息不能被共享缓存处理，对于其他用户的请求无效
- Content-Type：charset：告知客户端服务器本身响应的对象的类型和字符集
- Expires：浏览器会在指定过期时间内使用本地缓存
- Last-Modified：客户端请求的资源文件在服务器端最后被修改的时间
- Date：生成消息的具体时间和日期

- Content-Length：正文长度
- Set-Cookie：把 cookie 发送到客户端

四、 Fiddler 捕获请求

- 使用真实的 IP 或域名
- Fiddler 的 Response 乱码



  - ✓ 这是因为 HTML 被压缩了， 通过两种方法去解压缩。
    - ✧ 方法一：点击红框内容“Response body is encouded.Click to decode.”
    - ✧ 方法二：选中工具栏中的"Decode"，这样会自动解压缩。
- Fiddler 不仅能监听 HTTP 请求而且默认情况下也能捕获到 HTTPS 请求
  - ✓ Tool -> Fiddler Option -> HTTPS 下面进行设置，勾选上“Decrypt HTTPS traffic”
  - ✓ 如果不必监听服务器端得证书错误可以勾上“Ignore server certification errors”

五、 使用 Fiddler 实现接口测试

1 测试登录接口

- 抓包，完成后退出
- 选中需要测试的 url，找到并复制请求部分 raw 中的 cookie 整行
- 重新启动 Fiddler
- Composer→Parsed
  - ✓ Composer
    - ✧ 创作
  - ✓ 输入越权访问的 url
  - ✓ 粘贴 Cookie
  - ✓ 执行越权 url
    - ✧ Execute

2 传递非法参数

- 修改参数为非法数据，使用此方式伪造或篡改数据。
- 【例】测试 Webtours 处理信用卡号的接口。
  - ✓ 方法 1
    - ✧ 在 Fiddler 中设置抓包
    - ✧ 选择 url，F2，修改参数
    - ✧ 回放 url
    - ✧ 查看响应数据或后台数据
  - ✓ 方法 2
    - ✧ Composer
      - ● 选择 get 或 post 方法

- 输入 url，url 后应携带/
- post 方法应添加
  - Content-Type: application/x-www-form-urlencoded
- 参数
  - get 方法直接在 url/后写
    - ?参数名 1=值 1&参数名 2=值 2
  - post 方法在 RequestBody 中写
    - 参数名 1=值 1&参数名 2=值 2
- Execute

✧ 查看响应数据或后台数据

➢ 响应中文乱码问题

✓ 进入注册表

✧ HKEY_CURRENT_USER\Software\Microsoft\Fiddler2

✧ 新建字符串值：HeaderEncoding，值设置为 GBK

✧ 重启 Fiddler

3 伪造数据

➢ 设置断点修改 Request，使用此方式伪造或篡改数据。

➢ 【例】攻击 Webtours 付款接口。

✓ 无需截获数据流（抓包），在浏览器中打开将数据（如价格）发给服务器的页面，停止操作

✓ 在 Fiddler 中设置抓包和拦截请求

✓ 返回软件操作，给服务器发送数据

✓ 拦截完成，篡改将要发送的数据，点击 Break on Response，点击 Run to Completion

✓ 在 Fiddler 中取消抓包和拦截请求

✓ 继续后续操作，到完成

✓ 查看后台数据

➢ 支付漏洞与解决方案

✓ 漏洞 1：在支付过程中直接发送含有支付金额的数据包

✧ 让开发不要在数据包中加入价格和数量等敏感值。

✓ 漏洞 2：没有对购买数量进行限制

✧ 严格控制购买数量的大小，不允许数量为负数，控制总支付金额是一个正常的数。

✓ 漏洞 3：程序的异常处理

✧ 指支付的数据包异常的程序的错误处理。

✧ 这种异常可以是数据与 KEY 不符，支付的金额有错误，购买的数量不正确等。

✧ 程序的异常处理出现的原因主要是开发人员对出现异常后的处理不当造成的。