



دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)



دستور کار آزمایشگاه شبکه‌های کامپیوتری

مسئول آزمایشگاه:
دکتر مسعود صبایی

بهار ۱۴۰۴

الحمد لله
الکریم
الحمد لله
الکریم
الحمد لله
الکریم

قوانین آزمایشگاه شبکه های کامپیوتری

برای افزایش کارایی درس آزمایشگاه شبکه های کامپیوتری، رعایت عدالت بین تمامی گروه های آزمایشگاهی و آموزش حداکثری مطالب درس به صورت عملی، مدرسین و دانشجویان ملزم به رعایت نکات و قوانین ذیل هستند:

۱. تعداد جلسات در طول نیمسال ۱۰ تا ۱۲ جلسه خواهد بود.
۲. مدرسین و دانشجویان موظفند رأس ساعت مقرر در کلاس حضور یابند.
۳. قبل از انجام هر آزمایش، مبحث تئوری مربوط به آن آزمایش باید به طور کامل مطالعه شود، زیرا در حین جلسه وقت کافی برای توضیح و یادگیری مطالب تئوری وجود ندارد.
۴. پس از گذشت پنج دقیقه از شروع کلاس، به ازای هر پنج دقیقه تأخیر ۱۰ درصد نمره آن جلسه کسر میشود.
۵. حداکثر میزان تأخیر ۳۰ دقیقه است.
۶. هر آزمایش شامل یک پیش گزارش است که باید پیش از شروع آزمایش ها به مدرس تحویل داده شود. پیش گزارش مطلوب هر آزمایش در دستور کار آمده است.
۷. به ازای هر آزمایش، یک گزارش کار تهیه می شود که شامل تمامی مواردی است که در حین آزمایش با آن ها برخورد شده است. در این گزارش باید تمامی مشکلات پیش آمده و نحوه برطرف کردن آن ها ذکر گردد. همچنین، چگونگی انجام آزمایش مشتمل بر تحلیل آزمایش، به همراه اسکرین شات از مراحل انجام آزمایش ها تهیه شود.
۸. جهت کسب نمره قبولی در آزمایشگاه، کسب حداقل نمره قبولی در درس الزامی است.
۹. به منظور حفظ حرمت کلاس و نظافت آزمایشگاه، از خوردن و آشامیدن در طول کلاس خودداری نمایید.
۱۰. وارد آوردن هرگونه خسارت به تجهیزات آزمایشگاه مستلزم جبران خسارت است.

فهرست آزمایش ها

شماره آزمایش	عنوان آزمایش	صفحه
۱		
۲		
۳		
۴	تحلیل http با استفاده از نرم افزار Wireshark	
۵		
۶		
۷		
۸		
۹		
۱۰		
۱۱		
۱۲		

قالب آزمایش‌های آزمایشگاه شبکه‌های کامپیوتری

1- عنوان آزمایش:

تحلیل ترافیک HTTP با Wireshark

2- هدف آزمایش:

در این آزمایش چگونگی استفاده از Wireshark برای ضبط و تحلیل ترافیک HTTP آموزش داده خواهد شد. تحلیل ترافیک HTTP برای درک ارتباطات وب، شناسایی مشکلات امنیتی احتمالی و بررسی ناهنجاری‌های ترافیک شبکه ضروری است.

3- آمادگی پیش از آزمایش:

- پروتکل HTTP چگونه کار می‌کند؟
- عملیات رمزنگاری در HTTPS و TLS چگونه انجام می‌شود؟
- ارتباط بین کلاینت و سرور در HTTP چگونه انجام می‌شود؟
- چه تفاوتی بین HTTP/1.1 و HTTP/2 وجود دارد؟

4- تجهیزات/ابزار مورد نیاز:

- نصب Wireshark روی سیستم شما
- یک مرورگر وب برای ایجاد ترافیک HTTP

شرح آزمایش:

1. Wireshark: Wireshark را از <https://www.wireshark.org/download.html> دانلود و نصب کنید.
- مرورگر وب: از هر مرورگر (مانند Chrome، Firefox) برای ایجاد ترافیک HTTP استفاده کنید.

۱. ضبط ترافیک HTTP

مراحل

1. Wireshark را باز کنید.
2. کارت شبکه‌ای را که به اینترنت متصل است انتخاب کنید.
3. روی "Start Capture" کلیک کنید.
4. مرورگر وب خود را باز کنید و به یک وبسایت که از HTTP استفاده می‌کند بروید (http://example.com.).
5. اجازه دهید صفحه کاملاً بارگذاری شود و سپس ضبط را در Wireshark با کلیک بر روی آیکون مربع قرمز متوقف کنید.

خروجی مورد انتظار

- یک فایل ضبط شده حاوی ترافیک شبکه، شامل درخواست‌ها و پاسخ‌های HTTP

۲: فیلتر کردن ترافیک HTTP

مراحل

۱. در Wireshark ، به نوار فیلتر در بالای صفحه بروید.

۲. فیلتر http را وارد کرده و Enter بزنید.

انواع فیلترهای دیگر به صورت زیر می باشد که در صورت نیاز استفاده خواهد شد:

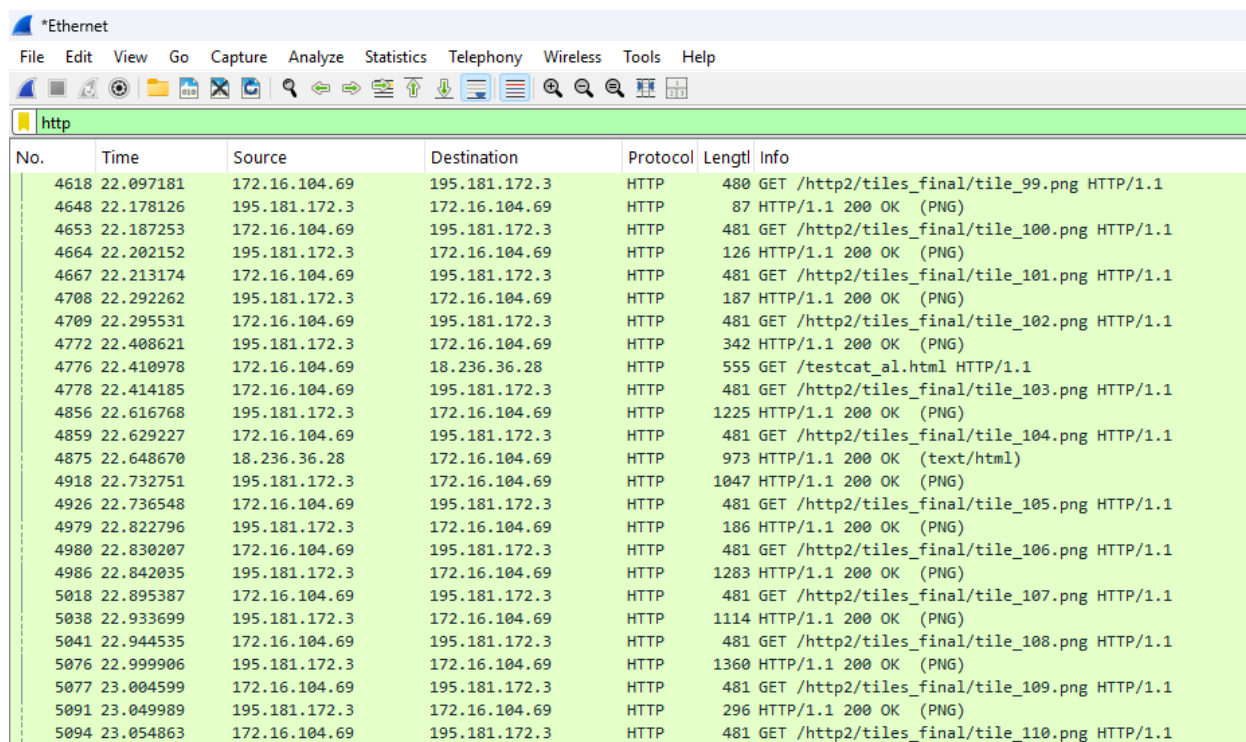
- `http.request.method == "GET"`
- `http.request.method == "POST"`
- `http.response.code == 200`
- `http.host contains "example.com"`

سوال:چند نمونه از فیلترهای دیگر را مشخص کرده و توضیح دهید که برای چه کاری استفاده می شوند.

۳. Wireshark تنها ترافیک HTTP ضبطشده را نمایش خواهد داد.

خروجی مورد انتظار

- نمایش ترافیک HTTP فیلترشده از ضبط کلی مانند تصویر ۱



The screenshot shows the Wireshark interface with the filter 'http' applied. The packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
4618	22.097181	172.16.104.69	195.181.172.3	HTTP	480	GET /http2/tiles_final/tile_99.png HTTP/1.1
4648	22.178126	195.181.172.3	172.16.104.69	HTTP	87	HTTP/1.1 200 OK (PNG)
4653	22.187253	172.16.104.69	195.181.172.3	HTTP	481	GET /http2/tiles_final/tile_100.png HTTP/1.1
4664	22.202152	195.181.172.3	172.16.104.69	HTTP	126	HTTP/1.1 200 OK (PNG)
4667	22.213174	172.16.104.69	195.181.172.3	HTTP	481	GET /http2/tiles_final/tile_101.png HTTP/1.1
4708	22.292262	195.181.172.3	172.16.104.69	HTTP	187	HTTP/1.1 200 OK (PNG)
4709	22.295531	172.16.104.69	195.181.172.3	HTTP	481	GET /http2/tiles_final/tile_102.png HTTP/1.1
4772	22.408621	195.181.172.3	172.16.104.69	HTTP	342	HTTP/1.1 200 OK (PNG)
4776	22.410978	172.16.104.69	18.236.36.28	HTTP	555	GET /testcat_al.html HTTP/1.1
4778	22.414185	172.16.104.69	195.181.172.3	HTTP	481	GET /http2/tiles_final/tile_103.png HTTP/1.1
4856	22.616768	195.181.172.3	172.16.104.69	HTTP	1225	HTTP/1.1 200 OK (PNG)
4859	22.629227	172.16.104.69	195.181.172.3	HTTP	481	GET /http2/tiles_final/tile_104.png HTTP/1.1
4875	22.648670	18.236.36.28	172.16.104.69	HTTP	973	HTTP/1.1 200 OK (text/html)
4918	22.732751	195.181.172.3	172.16.104.69	HTTP	1047	HTTP/1.1 200 OK (PNG)
4926	22.736548	172.16.104.69	195.181.172.3	HTTP	481	GET /http2/tiles_final/tile_105.png HTTP/1.1
4979	22.822796	195.181.172.3	172.16.104.69	HTTP	186	HTTP/1.1 200 OK (PNG)
4980	22.830207	172.16.104.69	195.181.172.3	HTTP	481	GET /http2/tiles_final/tile_106.png HTTP/1.1
4986	22.842035	195.181.172.3	172.16.104.69	HTTP	1283	HTTP/1.1 200 OK (PNG)
5018	22.895387	172.16.104.69	195.181.172.3	HTTP	481	GET /http2/tiles_final/tile_107.png HTTP/1.1
5038	22.933699	195.181.172.3	172.16.104.69	HTTP	1114	HTTP/1.1 200 OK (PNG)
5041	22.944535	172.16.104.69	195.181.172.3	HTTP	481	GET /http2/tiles_final/tile_108.png HTTP/1.1
5076	22.999906	195.181.172.3	172.16.104.69	HTTP	1360	HTTP/1.1 200 OK (PNG)
5077	23.004599	172.16.104.69	195.181.172.3	HTTP	481	GET /http2/tiles_final/tile_109.png HTTP/1.1
5091	23.049989	195.181.172.3	172.16.104.69	HTTP	296	HTTP/1.1 200 OK (PNG)
5094	23.054863	172.16.104.69	195.181.172.3	HTTP	481	GET /http2/tiles_final/tile_110.png HTTP/1.1

۳: تحلیل درخواست های HTTP

درخواست های HTTP شامل بخش های زیر هستند:

- **Method:** مانند GET، POST، PUT، DELETE
- **Host:** آدرس دامنه‌ای که درخواست با آن ارسال شده است.
- **User-Agent:** اطلاعات درباره مرورگر یا کلاینتی که درخواست را ارسال کرده است.
- **Referer:** صفحه‌ای که این درخواست از آن ارسال شده است (در صورت وجود)
- **URL:** مسیر موردنظر
- **Headers:** اطلاعات اضافی مانند User-Agent، Cookie، Content-Type
- **Body:** داده‌های ارسال شده در متدهای POST و PUT

نمونه درخواست HTTP:

```
✓ Hypertext Transfer Protocol
> GET /http2/tiles_final/tile_138.png HTTP/1.1\r\n
Host: 1153288396.rsc.cdn77.org\r\n
Connection: keep-alive\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36\r\n
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8\r\n
Referer: http://1153288396.rsc.cdn77.org/http2/http1.html\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
\r\n
[Response in frame: 6170]
[Full request URI: http://1153288396.rsc.cdn77.org/http2/tiles_final/tile_138.png]
```

سوال: چند نمونه از هدرهای HTTP دیگر را بررسی کرده و به طور کامل توضیح دهید. به طور مثال E-tag

۲. تحلیل پاسخ HTTP

پاسخ‌های HTTP شامل موارد زیر هستند:

- **Status Code:** کد وضعیت 200: (موفق)، 404 (یافت نشده)، 500 (خطای سرور)
- **Headers:** مانند Content-Type، Set-Cookie، Cache-Control
 - **Content-Type:** نوع محتوای ارسال شده مانند (text/html, application/json)
 - **Content-Length:** طول محتوای پاسخ.
 - **Date:** تاریخ و زمان ارسال پاسخ.
 - **Server:** نام و نسخه سرور که درخواست را پردازش کرده است.
 - **Location:** در صورت وجود، نشانی جدید برای ریدایرکت
- **Body:** محتوای صفحه وب یا داده‌های JSON

نمونه پاسخ HTTP:

```
✓ Hypertext Transfer Protocol
> HTTP/1.1 200 OK\r\n
  Date: Sat, 22 Feb 2025 12:14:25 GMT\r\n
  Content-Type: image/png\r\n
> Content-Length: 3228\r\n
  Connection: keep-alive\r\n
  ETag: "5626d94c-c9c"\r\n
  Cache-Control: no-cache\r\n
  Access-Control-Allow-Origin: *\r\n
  X-77-NZT: EwwBw7WsAQH3uIcoAAwBuUwKEwH3sU0AAwBnJIhJwG3WgYAAA\r\n
  X-77-NZT-Ray: 47824138c6081853c6bfb96716ab810f\r\n
  X-77-Cache: HIT\r\n
  X-77-Age: 1626\r\n
  Server: CDN77-Turbo\r\n
  X-Cache: HIT\r\n
  X-Age: 2656184\r\n
  Accept-Ranges: bytes\r\n
\r\n
[Request in frame: 2165]
[Time since request: 0.105978000 seconds]
[Request URI: /http2/tiles_final/tile_12.png]
[Full request URI: http://1153288396.rsc.cdn77.org/http2/tiles_final/tile_12.png]
File Data: 3228 bytes
```

سوال: کدهای 3x, 4x, 5x را با جزییات بررسی کرده و علت وقوع هر کدام را توضیح دهید.

۳. استخراج و بررسی داده‌های Payload

payload معمولاً به داده‌هایی اطلاق می‌شود که در قسمت محتوای یک بسته (packet) وجود دارد و شامل اطلاعات مفیدی مانند درخواست‌های HTTP، داده‌های فایل، محتوای ایمیل یا هر نوع داده دیگر است که در پروتکل‌های مختلف ارسال می‌شود. به طور مثال در Command Injection مهاجم تلاش می‌کند تا دستوراتی به سرور ارسال کند که در سیستم عامل اجرا شوند.

- در جزییات پاسخ HTTP، به دنبال داده‌های Payload مانند محتوای HTML بگردید.
- برای نمایش payload، می‌توانید روی بسته‌ها کلیک کنید و در بخش "Packet Details" یا "Data" محتوای بسته را مشاهده کنید.
- روی بسته پاسخ کلیک راست کرده و "TCP Stream" > "Follow" را انتخاب کنید تا کل مکالمه HTTP را مشاهده کنید.

خروجی مورد انتظار

- استخراج و بررسی داده‌های Payload از پاسخ HTTP.
نمونه بررسی Payload یک Command Injection:

```
POST /execCommand HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Content-Length: 54
command=; ls -la / && echo "Injected Command Executed"
```

- در اینجا، payload شامل یک Command است که به سرور دستور می‌دهد که ابتدا دستور ls -la / (برای نمایش لیست فایل‌ها و دایرکتوری‌ها) را اجرا کند و سپس پیامی را چاپ کند. Injected Command Executed.

این نمونه‌ها فقط مثال‌هایی از حملات مخربی هستند که می‌توانند از طریق **payload**های مختلف انجام شوند. این نوع حملات می‌توانند به سرقت اطلاعات، تغییر داده‌ها یا ایجاد آسیب در سیستم‌های هدف منجر شوند.

سوال: یک نمونه از حملات که از طریق **Payload** انجام می‌شود بررسی کرده و توضیح دهید. مثلاً **Sql injection**

۳. استخراج داده‌های حساس از HTTP

- داده‌های حساس شامل اطلاعاتی است که به راحتی می‌توانند به حملات امنیتی منجر شوند و ممکن است شامل مواردی مانند اطلاعات حساب کاربری، رمزهای عبور، داده‌های کارت‌های اعتباری، کوکی‌ها و داده‌های پزشکی یا شخصی باشند.

۱. مشاهده نام کاربری و رمز عبور ارسال شده در فرم‌های POST

- در صورت استفاده از HTTP ناامن، اطلاعات ورودی کاربران ممکن است ارسال شود. می‌توان با فیلتر زیر اطلاعات ارسال شده را مشاهده کرد:

- `http.request.method == "POST" && http.request.uri contains "login"`

۲. بررسی کوکی‌های احراز هویت

- کوکی‌ها برای نگهداری نشست کاربران استفاده می‌شوند. برای مشاهده کوکی‌های ارسال شده در درخواست‌های HTTP از فیلتر زیر استفاده کنید.

`http.cookie`

۳. استخراج فایل‌های دانلود شده از وبسایت‌ها

- می‌توان با استفاده از منوی **Follow HTTP Stream** محتویات فایل‌های منتقل شده را بررسی کرد.

مراحل

۱. در ترافیک HTTP فیلتر شده، یک درخواست POST پیدا کنید.
۲. روی درخواست POST کلیک کنید تا جزئیات آن در قسمت جزئیات بسته نمایش داده شود.
۳. به تب "Data" در بخش "Hypertext Transfer Protocol" بروید و اطلاعات بدنه درخواست را مشاهده کنید. معمولاً این داده‌ها به صورت پارامترهای فرم (form parameters) ارسال می‌شوند.

خروجی مورد انتظار

- نمایش اطلاعات دقیق درباره یک درخواست HTTP POST
- مثال خروجی اطلاعات محرمانه

```
Hypertext Transfer Protocol
POST /login HTTP/1.1\r\n
Host: example.com\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Content-Length: 29\r\n
\r\n
username=testuser&password=testpass
```