دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

پایان نامه کارشناسی
کامپیوتر

تولید ترافیک هجوم ناگهانی کاربران و ارزیابی سرورها

نگارش:
علیرضا حسن پور

استاد راهنما:
دکتر امیرحسین جهانگیر

تابستان ۱۴۰۲

به نام خدا

چکیده

حملات منع سرویس[1]توزیع شده، نوعی از تهدیدهای امنیتی مرتبط با شبکه‌های کامپیوتری میباشند که دسترس پذیری منابع شبکه را هدف قرار میدهند. یکی از ویژگیهای این نوع حملات حجم ترافیک بالا و یا درخواست سرویس توسط تعداد زیادی مهاجم غیرمجاز است که با هم شبکه‌ای از رباتها را تشکیل میدهند و باعث کاهش کارایی شبکه میشوند. مسئله تشخیص تعداد زیاد کاربران مجاز هنگام هجوم ناگهانی از درخواستهای غیرمجاز حمله، امروزه یکی از بزرگترین چالشهای پیش روی متخصصان امنیت شبکه میباشد. روشهایی که تاکنون ارائه شده‌اند، عمدتاً یا کارایی لازم را نداشته‌اند و یا با بالا رفتن دانش مهاجمین در تقلید رفتار کاربران مجاز، عمدتاً در کوتاه مدت پاسخگو بوده‌اند. روشهایی که عملکرد بهتری نسبت به سایر روشها داشته‌اند اکثراً بر اساس استخراج ویژگیهای آماری عمل میکنند.

هدف از این پروژه تولید یک جریان ترافیک است که سالم باشد (از نوع حمله منع سرویس نباشد) و طبیعی به نظر برسد به طوری که بتواند رفتار یک گروه از کاربران را شبیه سازی کند تا امکان ارزیابی سامانه‌ها و سرورهای موجود، پیش از قرار گرفتن زیر بار، فراهم گردد.

این پژوهش ابتدا به بررسی ویژگی‌های ترافیک های حمله و هجوم ناگهانی کاربران می پردازد. سپس یک ترافیک از نوع هجوم ناگهانی کاربران تولید می‌کنیم و ویژگی‌های آماری سرور را مورد ارزیابی قرار می‌دهیم.

نتایج به دست‌آمده شامل ترافیک‌های تولید شده و ارزیابی سرورها تحت این ترافیک است.

واژه‌های کلیدی:

حملات منع سرویس، هجوم ناگهانی کاربران، ارزیابی کارایی، ترافیک شبکه.

[1] Denial of Service Attacks

فهرست مطالب

فهرست اشکال

فهرست جدول ها

فصل ۱: مقدمه

هجوم ناگهانی [۲]وقتی اتفاق میافتد که تعداد زیادی از کاربران در یک بازه کوتاه زمانی اقدام به دسترسی به منابع یک شبکه کنند. در این حالت سرویس دهنده ممکن است توان پاسخگویی به حجم عظیم درخواست دهندگان را نداشته باشد و با مشکل مواجه شود و حتی در برخی موارد به‌طور کامل از کار بیفتد. برای مثال، ورود تعداد زیادی کاربر به یک سایت خبری در هنگام وقوع رویدادی خاص می‌تواند نمونه‌ای از هجوم ناگهانی باشد. ازآنجایی که افزایش ترافیک رسیده به سرویس دهنده در حملات منع سرویس تا حد زیادی مشابه هجوم ناگهانی کاربران میباشد، تشخیص حملاتی که این هجوم را شبیه سازی و تقلید می کنند و یا همزمان با وقوع این رویداد اعمال می شوند بسیار دشوار است و نیازمند تحلیل و بررسی دقیق ترافیک شبکه و شناخت ویژگیهای مختلف هر یک از موارد فوق برای تفکیک آن‌ها از یکدیگر می‌باشد. تا آنجایی که در سالهای اخیر نوع جدیدی از حملات منع سرویس معرفی شده‌اند که به آنها حمله هجوم ناگهانی کاربران گفته میشود. این حملات به صورت توزیع شده انجام میشود و حجم ترافیک به صورت ناگهانی افزایش می‌یابد به طوریکه سیستم‌های تشخیص حمله موجود نمیتوانند تشخیص دهند که آیا تعداد زیادی از کاربران که با آدرس های مختلف در حال درخواست سرویس هستند، کاربران مجاز سیستم هستند و یا با هدف خرابکارانه، بخشی از یک حمله منع سرویس می‌باشند. همانطور که گفته شد ممکن است که این حملات همزمان با هجوم ناگهانی کاربران نیز اتفاق بیفتد. برای مثال، یک رقیب اقتصادی که از هجوم خریداران به سایت رقیب خود در ساعات مشخصی آگاه است، میتواند اقدام به یک حمله منع سرویس در آن زمان نموده که ترافیک خاص از ترکیب این حمله با هجوم کاربران میتواند بسیار بزرگ و غیر قابل روانه‌سازی باشد و پیش بینی‌های در نظر گرفته شده در مورد میزان سرویسدهی و پهنای باند را با خطا و اختلال مواجه کند تشخیص حملات با گستردگی بالا و تعداد حمله کننده زیاد و یا حملاتی که هنگام هجوم ناگهانی کاربران اتفاق میافتند، یکی از چالشهای پیش روی متخصصان و محققین امنیت شبکه میباشد.

ما در فصل ۲، به طرح مفاهیم اولیه و مرور کارهای پیشین می‌پردازیم. سپس در فصل ۳، به معرفی ابزارهای استفاده شده می‌پردازیم. در فصل ۴، روش پیشنهادی را ارائه می‌کنیم و در نهایت در فصل ۵، به نتیجه‌گیری، جمع‌بندی کارهای انجام شده و ارائه چشم‌انداز کارهای آتی می‌پردازیم.

[۲] Flash Crowd

فصل ۲: مفاهیم اولیه و کارهای پیشین

۱.۲ ویژگیهای تشخیص حملات منع سرویس

یک روش و سیستم تشخیص حملات منع سرویس باید ویژگیهای زیر را دارا باشد:

۱.۱.۲ تشخیص سریع

ازآنجایی که حملات منع سرویس بسیار سریع عمل می‌کنند و در زمان بسیار کمی حجم ترافیک بسیار بالا میرود، اگر عمل تشخیص به موقع انجام نشود، ممکن است بعد از تشخیص، توانایی مقابله با حمله وجود نداشته باشد. به عبارتی، سیستم وقتی متوجه حمله منع سرویس شود که حمله تأثیر مورد نظر خود را گذاشته و برای مقابله با آن دیر شده است. بنابراین باید دقت کرد که لایه‌های دفاعی را قبل از سرویس دهنده مورد نظر قرار داد. برای مثال، اگر بتوان حمله را در یک مسیریاب تشخیص داد، میتوان عمل مقابله با آن را در آن مسیریاب و یا حتی قبل تر انجام داد و سرو یس دهنده با استفاده از مسیریاب‌های دیگر تعبیه شده به کار خود ادامه دهد. این مسئله اهمیت تشخیص حمله منع سرویس در لایه‌های پایینتر شبکه، به خصوص لایه شبکه [۳]و لایه پیوند [۴]داده، بیش از پیش روشن می‌کند. تشخیص حمله در لایه‌های بالاتر مانند لایه کاربرد نیازمند بررسی محتویات بسته‌ها در یک سرویس گیرنده است که نیازمند صرف منابع مختلف و زمان است.

۲.۱.۲ قابلیت اطمینان

روش پیاده سازی شده باید به درستی میان ترافیک رسیده از کاربران مجاز هنگام هجوم ناگهانی کاربران و ترافیک حمله تمایز قائل شود. هرچند ممکن است در عمل این مساله امکانپذیر نباشد، اما در روش مورد استفاده، باید تعداد تشخیصهای منفی نادرست (کاذب[۵]) و مثبت نادرست[۶]، صفر باشد. در عمل میتوان به این شرط بسنده کرد که این موارد حداقل باشند.

---

[۳] Network Layer

[۴] Data Link Layer

[۵] False Negative

[۶] False Positive

۳.۱.۲ امکانپذیری (عملی بودن)

روش مورد نظر باید قابلیت پیاده سازی در دنیای واقعی را، با توجه به زیرساخت‌ها و همبندی[۷]های موجود در شبکه‌ها، داشته باشد. برای مثال، روشهایی که نیازمند نگهداری حجم عظیم اطلاعات هستند و یا نیاز به انجام پردازش‌های سنگین روی داده دارند، ممکن است کارایی لازم را نداشته باشند و پیاده سازی آنها از لحاظ عملی امکانپذیر نباشد.

۴.۱.۲ تشخیص بی‌درنگ

روش مورد نظر باید به گونه‌ای باشد که به صورت بی‌درنگ و به محض دریافت ترافیکهای غیرطبیعی، عملیات تشخیص خود را آغاز کند. ازآنجایی که ویژگی حمله منع سرویس، بالا بردن حجم ترافیک است، در صورت گذر زمان و عدم تشخیص بیدرنگ، حجم ترافیک بالا رفته و حمله به میزان زیادی اثرگذار خواهد بود. بنابراین هرگونه تأمل یا تاخیر اضافی در تشخیص ترافیک حمله ممکن است حتی در صورت تشخیص، مقابله با حمله را غیرممکن سازد.

۵.۱.۲ انعطاف پذیری

روش مورد نظر باید توانایی تشخیص حملات مختلف را داشته باشد و مختص به یک حمله خاص نباشد. مهاجم ممکن است پروتکل مورد استفاده یا روش خود را تغییر دهد. این تغییر روش نباید بر روی سیستم تشخیص تأثیر خاصی داشته باشد و یا بتوان با حداقل تغییرات در سیستم آن را با تغییرات انجام شده توسط مهاجم، سازگار کرد.

اساس کارهایی که تا کنون در این مورد انجام گرفته‌اند، چه روشهایی که در سطح جریان و چه روشهایی که مقایسه‌ها در سطح کاربران انجام می دهند، بر پایه ویژگیهای آماری جریان قرار دارد. طبق تحقیقی که انجام گرفته است[۱]، مهاجم در بهترین حالت تنها میتواند از ۳۰ درصد سیستم‌هایی که به ربات حمله آلوده شده‌اند و تحت کنترل درآمده‌اند، برای تولید ترافیک حمله استفاده کند. دلیل این امر نیز این است که تمامی این سیستم‌ها درآن واحد روشن و در دسترس نیستند و به‌علاوه هماهنگی و بکار گرفتن همه این رباتها با توجه به اینکه روی شبکه‌های گوناگون با پهنای باندهای مختلف و زیرساختها ی مختلف قرار دارند دشوار میباشد. لذا

[۷] Topology

ویژگیهای آماری ترافیکی که توسط یک کاربر مجاز در هجوم ناگهانی کاربران تولید میشود، با ترافیکی که توسط یک برنامه خودکار یا نیمه خودکار که با دریافت دستور حمله، چه به صورت زمان بندی شده و چه به صورت دریافت دستور حمله، اقدام به تولید ترافیک میکند تفاوت‌هایی دارد.

۲.۲ هجوم ناگهانی کاربران

منظور از هجوم ناگهانی کاربران درخواست تعداد زیادی کاربر به یک سرویس دهنده وب در یک بازه زمانی کوچک است به طوریکه تعداد درخواستهای کاربران به سرویس دهنده، به صورت نمایی افزایش می‌یابد. این در حالی است که سرویس دهنده موردنظر در حالت عادی، این تعداد سرویس گیرنده نداشته و ممکن است برای آن برنامه ریزی نشده باشد که این باعث اختلال در عملکرد سخت افزاری و نرم‌افزاری آن خواهد شد. به همین دلیل است که این پدیده را جزو ناهنجاری‌ها[۸] ترافیک شبکه دسته بندی میکنند. رویدادهایی همچون رقابتهای ورزشی جهانی و بازیهای المپیک، منتشر شدن نسخه جدید از یک نرم‌افزار محبوب مانند نسخه‌های جدید یک سیستم عامل یا وقوع حوادث سیاسی و اجتماعی مانند حملات تروریستی و ...نمونه‌هایی از وقایعی هستند که می توانند هجوم ناگهانی کاربران را به سایتهای مرتبط با آنها در پی داشته باشند. اگر تعداد درخواستهای رسیده از کاربران در دقیقه، به‌صورت نمایی رشد کند، هجوم ناگهانی اتفاق افتاده است. این مسئله در رابطه ۱ نشان داده شده است[۲].

$$r_{ti} > 2^{i} r_{t0} \quad (۱)$$

که در این رابطه r_{ti} بیانگر میانگین نرخ درخواستها در بازه زمانی ti میباشد. طبق این رابطه اگر تعداد درخواستها از مقداری مشخص، که با توجه به درخواستهای قبلی هر کارگزار منحصربه فرد متفاوت میباشد، بیشتر شود، هجوم ناگهانی کاربران اتفاق افتاده است. البته به طورکلی، رشد تعداد درخواست‌های کاربران در یک بازه ی زمانی، تا حدی که سرویس دهنده مجبور باشد برای ادامه سرویس دهی خود عملیات خود را مدیریت و تنظیم کند، هجوم ناگهانی نامیده میشود.

[۸] Anomaly

۳.۲ تاریخچه‌ی هجوم ناگهانی کاربران

اصطلاح رویداد هجوم ناگهانی کاربران در سال ۱۹۷۰، سالها قبل از ابداع شبکه اینترنت، توسط نویسنده‌ای به نام لاری نیون[۹]، در یک رمان داستانی علمی تخیلی به نام پرواز اسب[۱۰]، مورد استفاده قرار گرفت. در داستان این کتاب، دانشمندی ماشین زمانی اختراع میکند که افراد به وسیله آن میتوانند به گذشته، زمانی که رویداد خاصی اتفاق افتاده است، سفر کنند. اوج داستان زمانی اتفاق می افتد که جمعیت انبوهی برای سفر به زمان رویداد خاصی هجوم می آورند و این هجوم ناگهانی باعث میشود که تغییراتی در آن رویداد رخ داده و گذشته تغییر کند که این موضوع باعث بروز ناهنجاری و آشوب‌هایی میشود.

سالها بعد و با ظهور شبکه اینترنت این اصطلاح، به دلیل مشابهت فراوان این داستان با هجوم کاربران به یک کارگزار وب که باعث افزایش درخواستهای رسیده به آن به صورت نمایی میشود، وارد دنیای فناوری اطلاعات شد. [۳]

۴.۲ اثر SlashDot

اثر اسلش‌دات به افزایش موقت ترافیک به یک وب‌سایت اشاره دارد، که می‌تواند زمانی رخ دهد که یک وب‌سایت پربازدید، پیوندی به سایت یا وبلاگ کوچک‌تر ارسال می‌کند، بنابراین افزایش بی‌سابقه‌ای در ترافیک ایجاد می‌کند. اگر افزایش ترافیک بسیار زیاد باشد، باعث کاهش سرعت سایت یا غیرقابل دسترس شدن آن می شود.

در این حالت پدیده هجوم ناگهانی کاربران برای سایت مذکور اتفاق میافتد. سایت SlashDot یک سایت خبری در مورد آخرین اخبار تکنولوژی و نرم افزار است که بسیاری از لینک‌های خبری آن، به سایت کوچک دیگر ارجاع داده میشود. به همین دلیل این پدیده که برای اولین بار در این سایت اتفاق افتاده است، به اثر SlashDot معروف است. [۳]

دسته بندی‌های مختلفی در مورد انواع هجوم ناگهانی انجام شده است. برای مثال،تقسیم بندی هجوم ناگهانی به دو نوع قابل پیشبینی و غیر قابل پیشبینی، عادی و بحرانی، دو نوع از این دسته بندی ها هستند [۳]. برای مثال، هجوم کاربران در مواقعی مانند وقتی که زمان معینی برای انتشار محصولی جدید، مانند نسخه جدید یک سیستم عامل، گوشی هوشمند و... تعیین شده است، از پیش محتمل است. اما در مواقعی که یک حادثه غیرمنتظره مانند وقوع یک حمله تروریستی و یا حوادثی مانند طوفان و ... اتفاق میافتد، هجوم ناگهانی کاربران قابل پیش بینی نیست.

در مواقع قابل پیش بینی می‌توان جلوی از کار افتادن سرویس دهنده را گرفت و یا تا حد زیادی از کاهش کارایی آن جلوگیری کرد [۴,۵]. اما در برخی مواقع، به خصوص وقتی که این پدیده غیر قابل پیش بینی است، هجوم

---

[۹] Larry Niven

[۱۰] The Flight of the Horse

ناگهانی کاربران باعث از کار افتادن سرویس دهنده می‌شود و هزینه‌های زیادی را در بر دارد. این نوع از هجوم ناگهانی کاربران بحرانی نامیده میشود.

۵.۲ نمونه‌هایی از هجوم ناگهانی کاربران در سالهای اخیر

در ۲۹ جولای ۲۰۱۵ که تاریخ عرضه سیستم عامل ویندوز ۱۰ از سوی شرکت مایکروسافت اعلام شده بود، شماری زیادی از کاربران برای به روزرسانی سیستم عامل‌های خود به این نسخه، به کارگزارهای این شرکت هجوم آوردند. باوجود اینکه این شرکت این هجوم را پیش بینی کرده بود و شبکه‌های توزیع محتوایی با پهنای باند۴۰ ترابایت در ثانیه برای آن در نظر گرفته بود، اما باز هم تعداد زیادی از کاربران، در ساعات اولیه موفق به به روزرسانی سیستم عامل خود نشدند. البته این هجوم از قبل پیش بینی شده بود و بسیاری از سایتهای خبری دنیا با تیتر "امروز پرترافیک ترین روز تاریخ اینترنت خواهد بود" به استقبال آن رفته بودند.

در سال ۲۰۱۲ طوفان "سندی" بخش وسیعی از سواح شرقی ایالات متحده را درنوردید. در پی این حادثه طبیعی، در عرض تنها چند ساعت، استفاده از اینترنت در جهان ۱۱۳ درصد افزایش یافت. سایت نتفلیکس با افزایش ناگهانی ۱۵۰ درصدی ترافیک مواجه شد. همچنین استفاده از سرویسهای اسکایپ ۱۱۲ درصد افزایش یافت. این مورد که در مواجهه با یک رویداد طبیعی رخ داد، یکی از بزرگترین موارد هجوم ناگهانی کاربران در سالهای اخیر محسوب میشود. [۳]

در سال ۲۰۱۰ و در طی برگزاری جام جهانی در آفریقای جنوبی، حجم عظیمی از کاربران به سایت توییتر هجوم آوردند، به طوریکه حجم توییتهای کاربران در طول یک بازی به ۷۵۰ توییت در ثانیه رسید. البته ماجرا به همینجا ختم نشد و با به ثمر رسیدن گل در یک مسابقه، این سایت با ۲۹۴۰ توییت در ثانیه مواجه شد که باعث از سرویس خارج شدن کل سایت به مدت چندین دقیقه شد که خسارت‌های فراوانی به آن وارد کرد. البته این اتفاق در طول بازیهای المپیک و بازیهای زمستانی نیز، بسیار اتفاق می‌افتد. یکی از بزرگترین و اولین موارد هجوم ناگهانی کاربران، در سال۱۹۹۸ و در طول بازی‌های نیمه نهایی جام جهانی فرانسه اتفاق افتاد که باعث شد سایت اصلی رقابت‌ها برای مدتی شاهد ترافیک بسیار سنگینی باشد.

در سال ۲۰۱۴، سایت فروش مایکروسافت آفیس، به دلیل هجوم کاربران برای خرید، از سرویس خارج شد. این اتفاق برای سایت اپل که نسخه جدید سیستم عامل iOS خود را برای دانلود در اختیار کاربرانش قرار داده بود، در سال ۲۰۱۳ نیز رخ داد.[۳]

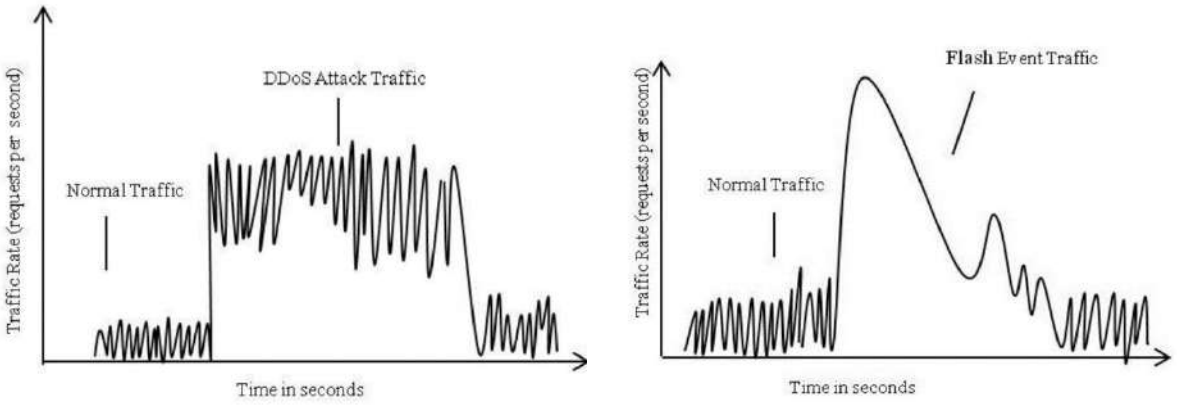
سایتهای خبری بزرگ دنیا در حوادثی مانند یازده سپتامبر و همچنین حوادث بمبگذاری در لندن در سال ۲۰۰۸ با حجم عظیمی از ترافیک از سوی بازدیدکنندگان خود مواجه شدند. موارد دیگری ازجمله رویدادهایی در سایت‌های خبری که خبر از درگذشت یک شخصیت معروف، یک خبر اقتصادی سیاسی مهم و ... میدهند، نمونه‌هایی از هجوم ناگهانی کاربران در سال‌های اخیر بوده‌اند.[۳]

از سرویس خارج شدن سرویس دهنده‌ی شرکت‌های بزرگ اینترنتی جهان در هنگام هجوم ناگهانی کاربران در سالهای اخیر نشان میدهد که باوجود قابل پیشبینی بودن در برخی موارد و اتخاذ تمهیدات مختلف برای مقابله با این پدیده، هنوز هم این مسئله نیازمند تحقیقات بیشتر و ارائه راهکارهای کارآمدتری برای فائق آمدن بر آن است.

۶.۲ شباهت‌ها و تفاوت‌های هجوم ناگهانی کاربران و حملات منع سرویس

مواردی چون بالا رفتن تعداد درخواستها و افزایش حجم ترافیک به صورت ناگهانی، بالا رفتن زمان پاسخ دهی و ناپایداری سیستم و درنهایت از دسترس خارج شدن سیستم، ویژگی‌های مشترک بین حمله منع سرویس و هجوم ناگهانی کاربران هستند. اما باید در نظر داشت که حمله منع سرویس توسط منبع مهاجم و با اهداف خرابکارانه صورت می گیرد. همچنین با وجود افزایش توانایی‌ها و دانش مهاجمین برای تقلید عملکرد کاربران مجاز، هنوز ازنظر پراکندگی آدرسهای اینترنتی، تعداد کاربران منحصربه فرد، الگوی درخواست‌ها و بسیاری از موارد دیگر، تفاوت‌هایی بین این دو پدیده وجود دارد که از آنها می توان برای تشخیص حملات و پیشگیری از آنها استفاده کرد.

اولین تفاوتی که ازنظر ترافیکی بین این دو پدیده وجود دارد نحوه افزایش و ثبات حجم ترافیک است. در هجوم ناگهانی کاربران، نرخ ترافیک در واحد زمان به صورت نوسانی می باشد که نمودار آن به صورت امواج زیگزاگی مشاهده می‌شوند، درحالیکه ترافیک حمله منع سرویس، پس از رسیدن به میزان مشخصی، معمولاً نرخ ثابتی را در پیش می‌گیرد. دو نمودار شکل ۱-۲ بیانگر این تفاوت هستند[۳, ۶]

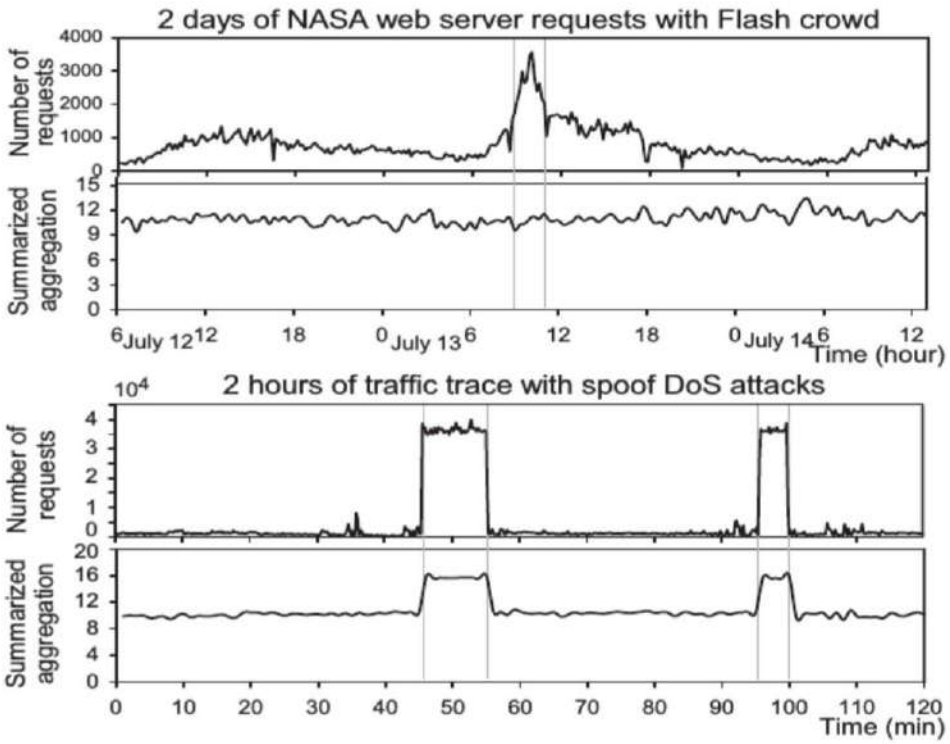


شکل ۲-۱ نرخ ترافیک در واحد ثانیه برای هجوم ناگهانی کاربران (راست) و حمله منع سرویس (چپ)[۳]

دلیل این تفاوت، ماهیت دو ترافیک می باشد. در هجوم ناگهانی کاربران، پس از اینکه کاربران متوجه افت کیفیت سرویس دهی شوند و یا با از کار افتادن‌های متوالی سرویس دهنده مواجه شوند، به دو دسته تقسیم میشوند، دسته‌ای بر ارسال درخواست‌های خود اصرار می‌ورزند و دسته‌ای نیز دلسرد شده و سعی میکنند زمان دیگری را برای دسترسی مورد نظر خود انتخاب کنند. با خروج این دسته از کاربران، کاربران دیگر وضع بهتری در سرویس دهی پیدا می کنند و به خواسته‌های خود میرسند. البته همانطور که گفته شد دلیل هجوم ناگهانی کاربران، رویداد خاصی است که طبیعتاً با گذشت زمان و اطلاع کاربران در مورد آن، حجم درخواست‌ها به تدریج کاهش می یابد. اما همانطور که گفته شد، حمله منع سرویس توسط رباتها حمله و به صورت برنامه‌ریزی شده و زمانبندی‌شده انجام میشود. بنابراین حمله در لحظه‌های خاص آغاز شده و با توجه به هدف آن و مکانیزم های دفاعی در مقابل آن، مدت زمان مشخصی به طول می انجامد. البته در حملات جدیدتر مهاجمان سعی می کنند افزایش نرخ ترافیک به صورت ناگهانی نباشد، اما کنترل افت ترافیک در پایان حمله با توجه به مکانیزم‌های دفاعی و همچنین نیاز به هماهنگی و در دسترس‌بودن رباتها، هنوز مساله‌ای دشوار برای مهاجمان است. شکل ۲ نمودار مربوط به تعداد درخواستها و همچنین تعداد آدرسهای اینترنتی خلاصه شده منحصربه فرد را برای دو مورد، هجوم ناگهانی کاربران و حمله منع سرویس نشان میدهد.[۷]

همانطور که مشاهده می شود در هر دو مورد تعداد درخواست‌ها به طور ناگهانی افزایش یافته است، مهمترین تفاوت این دو پدیده از منظر سرویس دهنده، نحوه مواجهه با آنهاست. در هنگام وقوع هجوم ناگهانی، باید تمهیداتی اندیشیده شود تا باوجود افزایش حجم ترافیک و مشکلات ناشی از آن، رضایت کاربران تأمین و درخواستهای آنان پاسخ داده شود. این کار را می‌توان با افزایش قابلیت‌های سخت‌افزاری، استفاده از شبکه‌های محتوا توزیع شده و ...انجام داد. اما در مورد حمله منع سرویس، آنچه اهمیت دارد جلوگیری از رسیدن درخواست‌های غیرمجاز و عدم پاسخ به آنها از جانب سرویس دهنده است. این کار میتواند با استفاده از پازل‌های گرافیکی[۸]، مسدودکردن آدرسهای مهاجمین و رباتها در مسیریاب‌های بالادستی، استفاده از تله عسل[۹] [۱۱]و ... انجام داد.

[۱۱] Honey pot : یک منبع سیستم اطلاعاتی با اطلاعات کاذب است که برای مقابله با هکرها و کشف و جمع‌آوری فعالیت‌های غیرمجاز در شبکه‌های رایانه‌ای بر روی شبکه قرار می‌گیرد



شکل ۲-۲ تعداد درخواستها در هجوم ناگهانی کاربران (بالا) و حمله منع سرویس (پایین)[۷]

در جدول ۲-۱، مقایسه میان حمله منع سرویس و هجوم ناگهانی کاربران به صورت تجمیعی آورده شده است:

جدول۲-۱ مقایسه حمله منع سرویس و هجوم ناگهانی

هجوم ناگهانی کاربران	حمله منع سرویس
سرویس‌دهنده و شبکه با حجم زیادی از ترافیک دریافتی اشباع می‌شوند.	سرویس‌دهنده و شبکه با حجم زیادی از ترافیک دریافتی اشباع می‌شوند.
ترافیک به‌وسیله کاربران مجاز ارسال می‌شود و پاسخ‌دهی به آن ضروری است.	ترافیک به‌وسیله کاربران غیرمجاز ارسال می‌شود و نیازی نیست به آن پاسخ داده شود.
به دو دسته فابل پیش‌بینی و غیرقابل پیش‌بینی تقسیم میشود و در پی علاقه کاربران در مورد رویدادی خاص رخ می‌دهد.	غیرقابل پیش‌بینی است و به‌واسطه استفاده مهاجم از سیستم آلوده به ربات که به آن زامبی گفته می‌شود، حمله رخ می‌دهد.

۷.۲ انواع ترافیک

از نظر طبیعی بودن، ترافیک شبکه را می‌توان به دو دسته اصلی تقسیم کرد: ترافیک طبیعی و ترافیک غیرطبیعی یا مشکوک.

۱.۷.۲ ترافیک طبیعی:

ترافیک طبیعی شامل ترافیکی است که ناشی از فعالیت‌های طبیعی در شبکه‌ها و سیستم‌های ارتباطی است. به عنوان مثال، ترافیک معمولی ممکن است به دلیل استفاده معمول کاربران از اینترنت، تماس‌های تلفنی روزمره، ارسال و دریافت ایمیل‌ها و غیره به وجود آید.

- این نوع ترافیک ناشی از فعالیت‌های معمول کاربران در شبکه است که در حد معمول و پیش‌بینی شده قرار می‌گیرد.
- شامل استفاده روزمره از اینترنت، تماس‌های تلفنی، ارسال و دریافت ایمیل‌ها، استفاده از برنامه‌های معمول نرم‌افزاری و غیره می‌شود.
- ترافیک سرور: در این نوع ترافیک، سرورها و سرویس‌های شبکه به منظور ارائه خدمات خود، ترافیکی را تولید می‌کنند. مثال‌هایی از این نوع ترافیک عبارتند از: انتقال فایل‌ها، بروزرسانی‌های نرم‌افزاری، پشتیبان‌گیری، مدیریت دستگاه‌ها و...
- ترافیک مدیریتی: ترافیک مدیریتی شامل ترافیک مربوط به فرآیندهای مدیریت شبکه می‌شود. مثال‌هایی از این نوع ترافیک عبارتند از: ارسال پیام‌های خطا و هشدار، پروتکل‌های مدیریت شبکه (مانند SNMP) و ترافیک مدیریتی دیگر که برای تنظیم و نظارت بر شبکه استفاده می‌شود.

۲.۷.۲ ترافیک غیرطبیعی یا مشکوک:

ترافیک غیرطبیعی شامل ترافیکی است که به طور غیرمعمول و نامتعارف در شبکه‌ها وجود دارد. ممکن است این نوع ترافیک ناشی از حملات سایبری، نفوذهای ناخواسته به سیستم‌ها، بدافزارها و ویروس‌ها، تلاش‌های نفوذکنندگان برای دسترسی غیرمجاز به داده‌ها و سیستم‌ها و سایر فعالیت‌های خلاف قوانین و قواعد امنیتی باشد. این نوع ترافیک باعث نارسایی یا ناهنجاری در عملکرد شبکه می‌شود و ممکن است ناشی از عوامل خارجی و مهاجمان باشد.

- ممکن است ترافیک غیرطبیعی ناشی از حملات سایبری، نفوذهای ناخواسته به سیستم‌ها، بدافزارها و ویروس‌ها، تلاش‌های نفوذ کنندگان برای دسترسی غیرمجاز به داده‌ها و سیستم‌ها و سایر فعالیت‌های خلاف قوانین و قواعد امنیتی باشد.
- ترافیک ناشی از حملات: حملات شبکه‌ای، مانند حملات DDoS، حملات نفوذ، جاسوسی و سایر حملات مخرب، ترافیک غیر طبیعی را به شبکه اضافه می‌کنند. این نوع ترافیک معمولاً برای اخلال در عملکرد شبکه، سرقت اطلاعات، تخریب سرویس‌ها و همچنین سرقت هویت استفاده می‌شود.

- ترافیک ناشی از بدافزارها: بدافزارها (مانند ویروس‌ها، کرم‌ها، تروجان‌ها و رمزنگارها) می‌توانند ترافیک غیر طبیعی را در شبکه ایجاد کنند. این نوع ترافیک معمولاً به منظور آلوده کردن سیستم‌ها، جاسوسی بر روی فعالیت‌های کاربران و سرقت اطلاعات شخصی استفاده می‌شود.

- ترافیک ناشی از خطاها و اشتباهات: خطاها و اشتباهات در تنظیمات شبکه، نرم‌افزارها یا دستگاه‌ها ممکن است منجر به ترافیک غیر طبیعی شوند. این نوع ترافیک ممکن است عملکرد شبکه را به طور قابل توجهی تحت تأثیر قرار دهد و مشکلات امنیتی و عدم قابلیت اطمینان را ایجاد کند.
- ترافیک ناشی از استفاده نادرست: استفاده نادرست از شبکه توسط کاربران، مانند دریافت غیرقانونی فیلم‌ها و موسیقی‌ها، استفاده از نرم‌افزارهای غیرمجاز یا استفاده از پروتکل‌های تجاری بدون مجوز، ترافیک غیر طبیعی را به شبکه اضافه می‌کند.

تشخیص و مدیریت ترافیک ناطبیعی اهمیت بسیاری در حفظ امنیت و ادامه‌ی عملکرد صحیح شبکه دارد.

در هر شبکه، ترافیک طبیعی و انسانی به صورت معمول و پیش‌بینی شده وجود دارد، اما ترافیک غیرطبیعی یا مشکوک نشانه‌ای از نقض امنیت یا فعالیت ناهنجار در شبکه است و نیازمند بررسی و اقدامات امنیتی می‌باشد.

فصل ۳: ابزارهای تولید و ارزیابی ترافیک شبکه

در این فصل می‌خواهیم به ابزارهای تولید ترافیک و ارزیابی ترافیک شبکه بپردازیم. در زیر تعدادی از این ابزارها را معرفی و مقایسه می‌کنیم.

۱.۳ سلنیوم

ابزار سلنیوم [10, 11] یک ابزار تست خودکار و اتوماسیون وب است که برای تست و اجرای عملیات بر روی مرورگرها استفاده می‌شود. سلنیوم ابتدا توسط جیسون هاوارد در سال ۲۰۰۴ توسعه داده شد و اکنون توسط یک جامعه گسترده از توسعه‌دهندگان پشتیبانی می‌شود.
سلنیوم یک ابزار تست خودکار برای بررسی صحت عملکرد وبسایت است. ما می‌توانیم هر برنامه، نرم‌افزار و اپلیکیشن تلفن همراه را با استفاده از Selenium تست کنیم.

۱.۱.۳ معرفی تست خودکار

خودکارسازی تست نرم‌افزار با استفاده از ابزارهای تخصصی برای کنترل اجرای تست‌ها و مقایسه نتایج واقعی با نتایج مورد انتظار می‌باشد. معمولاً تست‌های رگرسیون، که اقدامات تکراری هستند، به صورت خودکار انجام می‌شود. ابزار تست نه تنها به ما در انجام تست‌های رگرسیون کمک می‌کند، بلکه به ما کمک می‌کند تا تولید داده‌ها، نصب محصول، تعامل GUI ، لاگ کردن نقص‌ها و غیره را به طور خودکار انجام دهیم.

تست خودکار یک تکنیک می‌باشد که یک برنامه کاربردی (اپلیکیشن) یا نرم‌افزار را برای پیاده‌سازی کل چرخه عمر آن در زمان کم مورد استفاده یا اجرا قرار می‌دهد و بهره وری و اثربخشی را برای نرم‌افزار تست فراهم می‌کند. تست خودکار یک تکنیک اتوماتیک است که تست‌کننده‌ها آن را مستقیماً می‌نویسند و از نرم‌افزار مناسب برای تست نرم‌افزار استفاده می‌کنند. معمولاً، تسترها اسکریپت‌های تست و موارد تست را با استفاده از ابزار خودکارسازی فراهم می‌کنند. هدف اصلی تست خودکار، افزایش کارایی تست و توسعه ارزش نرم‌افزاری است.

۲.۱.۳ ویژگی‌های سلنیوم (Selenium)

بعضی از ویژگی‌های اصلی ابزار سلنیوم عبارتند از:

- پشتیبانی از زبان‌های برنامه‌نویسی: سلنیوم از زبان‌های برنامه‌نویسی متنوعی مانند Java، C#، Python، Ruby و JavaScript پشتیبانی می‌کند. این ویژگی به توسعه دهندگان امکان می‌دهد ابزار را با استفاده از زبانی که بهترین تجربه را برای آن‌ها فراهم می‌کند، استفاده کنند.

- تعامل با مرورگرها: سلنیوم قادر است با مرورگرهای مختلفی مانند Chrome، Firefox، Safari و Edge تعامل کند. این ویژگی به توسعه دهندگان اجازه می‌دهد تست‌های خود را بر روی مرورگرهای متنوعی اجرا کنند و اطمینان حاصل کنند که برنامه‌ها به درستی در همه مرورگرها عمل می‌کنند.

- قابلیت تست تعاملی: با استفاده از سلنیوم، می‌توانید عملیات تعاملی را در صفحات وب شبیه‌سازی کنید. مثلاً می‌توانید فرم‌ها را پر کنید، دکمه‌ها را کلیک کنید، صفحات را پیمایش کنید و ورود به حساب کاربری را شبیه‌سازی کنید.

- امکان انتخاب المان‌ها: سلنیوم به شما امکان می‌دهد المان‌های صفحه وب را بر اساس مشخصه‌های مختلفی انتخاب کنید. می‌توانید المان‌ها را بر اساس نام، کلاس، شناسه، مسیر XPath و CSS انتخاب کنید.

- پشتیبانی از تست‌های چند مرورگره: سلنیوم به شما امکان می‌دهد تست‌های خود را بر روی چند مرورگر به صورت همزمان اجرا کنید. این ویژگی به شما اجازه می‌دهد تا تست‌های خود را در مرورگرهای مختلف بررسی کرده و سازگاری صفحه‌های وب را بررسی کنید.

- امکان توسعه سفارشی: سلنیوم قابلیت توسعه سفارشی را به شما می‌دهد. می‌توانید ابزار را با استفاده از پلاگین‌ها و افزونه‌ها به نیازهای خود تنظیم کنید و قابلیت‌های جدیدی به آن اضافه کنید.

۳.۱.۳ معایب استفاده از سلنیوم

با وجود مزایا و قابلیت‌های بسیاری که سلنیوم ارائه می‌دهد، برخی معایب نیز وجود دارد که در زیر آورده شده است:

- پیکربندی پیچیده: راه‌اندازی و پیکربندی سلنیوم ممکن است پیچیده باشد، به خصوص برای توسعه دهندگانی که تازه با این ابزار آشنا می‌شوند. نیاز به تنظیمات و پیکربندی مرورگرها و محیط تست می‌تواند زمان‌بر و دشوار باشد.

- پشتیبانی ناکامل از برخی مرورگرها: ممکن است سلنیوم پشتیبانی ناکاملی از برخی مرورگرها داشته باشد. برخی مرورگرها ممکن است قابلیت اجرا و تعامل کامل با سلنیوم را نداشته باشند و این می‌تواند محدودیتی در تست و عملکرد صفحه‌های وب ایجاد کند.

- مستلزم دانش برنامه‌نویسی: برای استفاده کامل و بهره‌وری از سلنیوم، نیاز به دانش و تسلط بر زبان‌های برنامه‌نویسی مانند Java، Python یا C# و مفاهیم برنامه‌نویسی وب دارید. این به معنای نیاز به توسعه‌دهندگان مجرب یا تیم‌هایی با توانایی‌های برنامه‌نویسی است.

- زمان اجرا طولانی: در برخی موارد، زمان اجرای تست‌ها با استفاده از سلنیوم ممکن است طولانی باشد، به خصوص زمانی که تست‌ها پیچیده و حاوی عملیات تعاملی زیادی باشند. این ممکن است باعث افزایش زمان اجرای کل فرآیند تست و کاهش بهره‌وری شود.

- وابستگی به تغییرات صفحه‌های وب: هنگامی که صفحه‌های وب تغییر می‌کنند، ممکن است نیاز به تغییر و به‌روزرسانی تست‌ها و صحت‌یابی داشته باشید. این نیاز به روزرسانی می‌تواند منجر به زمان‌بری و پیچیدگی بیشتر در فرآیند توسعه و نگهداری شود.

۴.۱.۳ حوزه‌های کاری ابزار سلنیوم:

ابزار سلنیوم حوزه‌های مختلف کاربرد دارد. برخی از حوزه‌های کاری سلنیوم عبارتند از:

- تست عملکردی: سلنیوم می‌تواند برای تست و اعتبارسنجی عملکرد صفحات وب استفاده شود. با استفاده از سلنیوم، می‌توانید تست‌های خود را برای بررسی عملکرد صفحه‌های وب در شرایط مختلف بارگذاری و تعامل با کاربران انجام دهید.

- تست عملکرد صفحات پویا: صفحات وبی که از تعامل کاربر با استفاده از اسکریپت‌ها و فناوری‌های جاوا اسکریپت استفاده می‌کنند، می‌توانند با استفاده از سلنیوم تست شوند. سلنیوم امکان تعامل با المان‌ها و اجرای اسکریپت‌های جاوا اسکریپت را فراهم می‌کند.

- تست رابط کاربری (UI)[۱۲]: سلنیوم به توسعه دهندگان امکان می‌دهد تست‌های رابط کاربری را برای صفحات وب انجام دهند. با استفاده از سلنیوم، می‌توانید تعامل با المان‌ها، کلیک کردن روی دکمه‌ها، ورود اطلاعات به فرم‌ها و بررسی نتایج را تست کنید.

- خودکارسازی فرآیندهای تکرارشونده: سلنیوم می‌تواند برای اتوماسیون فرآیندهای تکرارشونده در صفحات وب استفاده شود. برای مثال، می‌توان سناریوهای خاصی را برای ثبت نام کاربران، خرید محصولات یا ارسال فرم‌ها ایجاد کرد و این فرآیندها را به صورت اتوماتیک با استفاده از سلنیوم اجرا کرد.

موارد فوق تنها برخی از حوزه‌های کاری سلنیوم هستند و این ابزار در زمینه‌های مختلف دیگر نیز استفاده می‌شود.

۵.۱.۳ رقبای سلنیوم

سلنیوم یکی از ابزارهای محبوب تست و اتوماسیون وب است، اما در مقایسه با برخی رقبا، مزایا و معایب مختلفی وجود دارد. در زیر، به برخی از رقبای سلنیوم و مزایا و معایب آن‌ها نسبت به سلنیوم می‌پردازیم:

- ابزار Appium [12]:
- مزیت‌ها:
- Appium برای تست و اتوماسیون برنامه‌های موبایل و تلفن همراه استفاده می‌شود.
- قابلیت پشتیبانی از چندین پلتفرم از جمله iOS و Android را دارد.
- امکان استفاده از زبان‌های برنامه‌نویسی مختلف مانند Java، Python و JavaScript را فراهم می‌کند.

- معایب:
- Appium تنها برای تست برنامه‌های موبایل استفاده می‌شود و قابلیت تست صفحات وب را ندارد.
- پیکربندی و راه‌اندازی Appium ممکن است پیچیده باشد و نیاز به تنظیمات و پیکربندی مختلف داشته باشد.

[۱۲] User interface

- ابزار Cypress[13]:

مزیت‌ها:

- سرعت اجرای بسیار بالا و زمان اجرای کوتاه تست‌ها را دارد.
- قابلیت دیباگ و نمایش واقعی زمان تغییرات در صفحه را فراهم می‌کند.
- پشتیبانی از تست‌های تک واحدی و تست‌های سیستمی را دارد.

معایب:

- Cypress محدود به مرورگر Chrome است و برای تست در مرورگرهای دیگر پشتیبانی نمی‌کند.
- عدم پشتیبانی کامل از تست بر روی دستگاه‌های موبایل را دارد.

- ابزار Puppeteer[14]:

مزیت‌ها:

- Puppeteer ابزاری مبتنی بر Chrome DevTools است و قابلیت کنترل و تست مرورگر Chrome را فراهم می‌کند.
- امکان اجرای سریع و نمایش دقیق عملیات در صفحه را دارد.
- قابلیت نمایش صفحه وب و ذخیره تصاویر از صفحه را فراهم می‌کند.

معایب:

- Puppeteer محدود به مرورگر Chrome است و پشتیبانی کامل از مرورگرهای دیگر را ندارد.
- تنها برای تست صفحات وب و تعامل با آن‌ها استفاده می‌شود و قابلیت تست برنامه‌های موبایل را ندارد.

- ابزار TestCafe[15]:

مزیت‌ها:

- TestCafe یک ابزار تست وب بر مبنای JavaScript است که بر روی مرورگرهای مختلف قابل اجرا است.
- نیازی به نصب و پیکربندی مرورگرها یا پلاگین‌ها ندارد.
- اجرای موازی و سریع تست‌ها را فراهم می‌کند.

- معایب:

- قابلیت ارائه عملکرد تعاملی با صفحه وب به اندازه سلنیوم ندارد.
- برخی از ویژگی‌های پیشرفته‌تر مانند تست بصری (Visual Testing) به صورت پیشفرض در TestCafe فراهم نیست.

- ابزار WebdriverIO[16]:

مزیت‌ها:

- WebdriverIO ابزاری مبتنی بر سلنیوم است که با استفاده از JavaScript و Node.js قابل استفاده است.
- قابلیت اجرا بر روی مرورگرهای مختلف را دارد.
- قابلیت ایجاد تست‌های قابل خواندن و نوشتن به سادگی را فراهم می‌کند.

- معایب:
- ممکن است نیاز به تنظیمات پیکربندی مرورگرها و محیط تست داشته باشد.
- نیاز به دانش برنامه‌نویسی JavaScript دارد.

هر یک از این رقبا مزایا و معایب خود را دارند و انتخاب بین آنها وابسته به نیازها و محدودیت‌های خاص پروژه است.

در جدول ۳-۱ به مقایسه این ابزار ها می‌پردازیم :

جدول ۳-۱ مقایسه ابزار های خودکارسازی[۲۴]

ابزار	زبان پشیبانی‌شده	مزایا	معایب
Selenium	- Java - C# - Python - Ruby - JavaScript	- تعامل با مرورگرها - قابلیت تست تعاملی - پشتیبانی از تست‌های چند مرورگره	- پیکربندی پیچیده - پشتیبانی ناکامل از برخی مرورگرها - مستلزم دانش برنامه‌نویسی - زمان اجرا طولانی
Puppeteer	- Node.js	- امکان اجرای سریع و نمایش دقیق عملیات	- محدود به مرورگر Chrome - تنها برای تست صفحات وب
Appium	- Java - Python - JavaScipt	- تست و اتوماسیون برنامه‌های موبایل و تلفن همراه - قابلیت پشتیبانی از چندین پلتفرم	- تنها برای تست برنامه‌های موبایل - نیاز به تنظیمات و پیکربندی پیچیده
Cypress	- JavaScript	- سرعت اجرای بسیار بالا و زمان اجرای کوتاه تست‌ها - قابلیت دیباگ و نمایش واقعی زمان تغییرات در صفحه	- محدود به مرورگر Chrome است - عدم پشتیبانی کامل از تست بر روی دستگاه‌های موبایل
TestCafe	- JavaScript	- اجرا بر روی مرورگرهای مختلف - اجرای موازی و سریع تست‌ها - بدون نیاز به نصب و پیکربندی	- عدم وجود رابط گرافیکی مناسب
WebdriverIO	- JavaScript - Node.js	- اجرا بر روی مرورگرهای مختلف	- نیاز به دانش برنامه‌نویسی JavaScript - نیاز به تنظیمات پیکربندی مرورگرها

۲.۳ برنامه تحلیل شبکه Wireshark

[17] Wireshark یک تحلیل‌گر بسته‌های شبکه به صورت رایگان و منبع باز است. این ابزار برای رفع اشکالات شبکه، تحلیل، توسعه نرم‌افزار و پروتکل‌های ارتباطی و آموزش استفاده می‌شود. در ابتدا به نام

Ethereal شناخته می‌شد، اما در ماه مه سال ۲۰۰۶ به دلیل مسائل مربوط به علامت تجاری به Wireshark تغییر نام داده شد.

Wireshark یک برنامه چندسکویی است که در نسخه‌های کنونی از این ابزار از کتابخانه Qt برای پیاده‌سازی رابط کاربری استفاده می‌کند و از pcap برای ضبط بسته‌ها استفاده می‌کند. این برنامه بر روی سیستم‌عامل‌های لینوکس، macOS، BSD، Solaris و برخی سیستم‌عامل‌های مشابه یونیکسی و ویندوز مایکروسافت اجرا می‌شود. همچنین نسخه‌ای بدون رابط گرافیکی به نام TShark نیز وجود دارد. Wireshark و دیگر برنامه‌های همراه با آن مانند TShark، نرم‌افزارهای رایگانی هستند که تحت شرایط مجوز عمومی GNU نسخه ۲ یا نسخه‌های بعدتر منتشر می‌شوند.

۱.۲.۳ قابلیت‌ها

Wireshark بسیار شبیه به tcpdump [18] است، اما دارای رابط کاربری گرافیکی و گزینه‌های مرتب‌سازی و فیلتر کردن یکپارچه است.

در لینوکس، BSD و macOS، با استفاده از libpcap نسخه ۱.۰.۰ یا بالاتر، Wireshark 1.4 و نسخه‌های بعدی می‌توانند کنترل‌کننده‌های رابط شبکه بی‌سیم را در حالت نظارتی قرار دهند.

اگر یک دستگاه از راه دور بسته‌ها را ضبط کند و بسته‌های ضبط شده را با استفاده از پروتکل TZSP یا پروتکل استفاده شده توسط OmniPeek به یک دستگاهی که Wireshark در آن اجرا می‌شود، ارسال کند، Wireshark این بسته‌ها را تجزیه و تحلیل می‌کند، بنابراین می‌تواند بسته‌های ضبط شده در یک دستگاه از راه دور را در زمان ضبط تحلیل کند.

۳.۳ ابزار تولید بار کاری Locust

Locust [19] یک ابزار تست کاربری ساده است که برای اجرای آزمایشی وب‌سایت -یا سیستم‌های دیگر- ساخته شده تا مشخص شود که آن وب‌سایت چه تعداد کاربر را می‌تواند در لحظه مدیریت‌کند. ایده پیاده‌سازی این ابزار به این شکل است که در طول یک فرآیند تست وب‌سایت، گروهی از اصطلاحاً «ملخ‌ها» می‌توانند به وب‌سایت شما حمله کنند. رفتار هر ملخ (یا همان کاربر فرضی)، توسط شما تعریف و فرآیند هجوم آوردن از طریق یک رابط کاربری نظارت می‌شود. این جنگ آزمایشی به شما کمک خواهد کرد تا تنگناهای کد خود را قبل از ورود کاربران واقعی شناسایی و رفع کنید. با استفاده از Locust،

می‌توان تعداد بالایی از کاربران همزمان و تراکم بالای تراکنش‌ها را شبیه‌سازی کرده و عملکرد سیستم را در شرایط مختلف مورد ارزیابی قرار داد.[20]

در واقع، هر حمله ملخ‌ها به سایت شما داخل سندباکس خودش اجرا می‌شود. اجرا در سندباکس [۱۳]به معنای اجرای یک برنامه یا کد در یک محیط ایزوله و امن است که جدا از محیط عملکرد واقعی سیستم عامل اجرا می‌شود. سندباکس به عنوان یک محیط مجازی و ایمن عمل می‌کند و کاربر اجازه می‌دهد برنامه‌ها و کدها را بدون ایجاد تأثیرات غیرمطلوب یا خطر بر روی سیستم عملی کند.

استفاده از سندباکس در اجرای برنامه‌ها و کدها دارای مزایا و کاربردهای گوناگون است. تعدادی از این مزایا عبارتند از:

- جداسازی: سندباکس اجازه می‌دهد برنامه‌ها و کدها را در یک محیط جداگانه و ایزوله اجرا کنید، بدون اینکه تأثیری بر روی سیستم عملی شما داشته باشد. این به شما امکان می‌دهد که برنامه‌ها را در یک محیط کنترل شده تست و از احتمال بروز مشکلات و آسیب به سیستم عملی جلوگیری کنید.

- امنیت: با اجرای برنامه‌ها در سندباکس، خطرات امنیتی که ممکن است در برنامه وجود داشته باشد محدود می‌شود. سندباکس اجازه می‌دهد تا هرگونه آسیب رساندن به فایل‌ها و داده‌های حساس سیستم عملی را کنترل کنید.

- تست و ارزیابی: با استفاده از سندباکس، می‌توانید برنامه‌ها و کدها را به صورت ایزوله و مستقل تست کنید. این به شما امکان می‌دهد که نحوه عملکرد و صحت برنامه‌ها را در یک محیط کنترل شده بررسی کنید و اشکالات (Bug) و مشکلات را شناسایی کنید.

در کل، اجرا در سندباکس یک روش امن و موثر است که در برخی موارد برای تست و اجرای برنامه‌ها و کدها به صورت جداسازی‌شده و کنترل شده استفاده می‌شود.

۱.۳.۳ ویژگی‌های Locust

برخی از مهم‌ترین ویژگی‌های این ابزار عبارتند از:

- ارسال سناریوهای تست کاربران در پایتون: نیازی به رابط کاربری پیچیدهXML ندارید؛ مثل هر پروژه پایتون دیگری، به سادگی می‌توانید کد بزنید.

[۱۳] Execution in Sandbox

- توزیع‌پذیری/مقیاس پذیری (پشتیبانی از صدها هزار کاربر) Locust: از آزمایش‌های بار شدن بر روی چندین دستگاه پشتیبانی می‌کند. با توجه به این که لوکاست مبتنی بر رویداد است، می‌توانید هزاران نفر از کاربران را در یک اجرا مدیریت‌کنید. بخشی از دلیل چنین سیاستی این است که حتی اگر شما اقدام به شبیه‌سازی تعداد بالایی از کاربران نمایید، همه آن‌ها به طور جدی به سیستم شما هجوم نمی‌آورند؛ اکثر مواقع، کاربران نمی‌داند که چه کاری می‌خواهند انجام دهند. به عبارت دیگر، درخواست‌های در لحظه هرگز به معنای تعداد کاربران آنلاین وب‌سایت نیست.

- رابط کاربری مبتنی بر وب: پیشخوان[14] Locust با HTML/JS کدنویسی شده‌است و تمام جزئیات مربوطه را به صورت بی‌درنگ (Real-time) نشان می‌دهد و در صورت نیاز هم قادر خواهید بود که آن را اختصاصی‌سازی [۱۵]کنید.

- قابلیت تست هر سیستمی را دارد: با این که لوکاست مبتنی بر وب است، اما می‌توان آن را برای تست تقریباً هر نوع اپلیکیشنی به کار برد. فقط یک کلاینت برای آزمایش آن چیزی را که می‌خواهید تست کنید می‌نویسیم و با ملخ‌ها به آن هجوم می‌بریم.

- منبع‌باز[۱۶] است: لذا به سادگی می‌توان آن را هک کرده و بسته به نیازهای خود این ابزار را دستخوش تغییر ساخت و جالب است بدانیم که پیچیدگی‌های ابزارهای تست فعلی، دلیل اصلی ایجاد Locust بوده است.

- رابط کاربری مناسب: این ابزار با دارا بودن رابط کاربری کاربرپسند و قابلیت مدیریت آسان، برای کاربرانی که با مفاهیم تست عملکرد آشنایی کمی دارند نیز مناسب است. همچنین، Locust از پروتکل HTTP و WebSocket پشتیبانی می‌کند و می‌تواند با سیستم‌هایی که از این پروتکل‌ها استفاده می‌کنند، ارتباط برقرار کند.

- بهبود کیفیت و کارایی: استفاده از Locust در فرآیند توسعه نرم‌افزار، باعث بهبود کیفیت و پایداری سیستم می‌شود و به تیم‌های توسعه و عملیات کمک می‌کند تا با تمرکز بر عملکرد سیستم، مشکلات و ضعف‌ها را بهبود دهند و اطمینان حاصل کنند که سیستم به درستی عمل خواهد کرد.

---

[۱۴] Frontend
[۱۵] Customize
[۱۶] Open source

با اینکه Locust ابزاری رایگان است، اما دارای جامعه فعالی از توسعه‌دهندگان است و به‌روزرسانی‌ها و پشتیبانی منظمی دارد. این ابزار با مجوز MIT منتشر شده است و قابلیت انتشار، استفاده و تغییرات آزاد را به کاربران می‌دهد.

استفاده از Locust مزایا و معایب خاص خود را دارد، که در زیر به آنها اشاره می‌کنم:

۲.۳.۳ مزایای استفاده از Locust :

- سادگی و آسانی استفاده Locust :یک ابزار ساده و قابل استفاده است که از طریق نوشتن کدهای Python قابل توسعه و قابل پیکربندی است. این ابزار برای توسعه دهندگان و تسترها به راحتی قابل فهم و قابل استفاده است.

- مقیاس پذیری Locust : این ابزار می‌تواند بار و تعداد کاربران همزمان را به راحتی تنظیم کند. با استفاده از Locust، شما می‌توانید با تعداد زیادی کاربر مصنوعی تست خود را اجرا کنید و مقیاس پذیری برنامه یا سرویس خود را در برابر بارهای بالا بسنجید.

- امکان ثبت و تجزیه و تحلیل دقیق Locust : ابزاری قدرتمند برای ثبت و تجزیه و تحلیل جزئیات بار و عملکرد است. شما می‌توانید نتایج جزئی و آمارهای دقیق را از طریق پنل مدیریتی Locust مشاهده کنید و به راحتی گزارش‌ها و نمودارهای مربوط به تست خود را تولید کنید.

۳.۳.۳ معایب استفاده ازLocust :

- محدودیت در پروتکل‌های پشتیبانی شده Locust : از پروتکل‌های HTTP و WebSocket پشتیبانی می‌کند، اما نمی‌تواند برای پروتکل‌های دیگر مانندFTP، SMTP و غیره استفاده شود. این محدودیت ممکن است برخی از تست‌ها را محدود کند.

- پیکربندی پیچیده: در صورتی که نیاز به تنظیمات پیشرفته و پیکربندی‌های پیچیده داشته باشید، Locustممکن است به طور محدود برای شما مناسب باشد. به عنوان مثال، برای تنظیمات پیچیده بار مانند استفاده از پروکسی، اتصال به پایگاه داده و غیره، نیازمند تغییرات و کدهای اضافی خواهید بود.

- نیاز به تجربه در برنامه‌نویسی Locust: برای تنظیمات پیشرفته نیاز به تسلط بر زبان برنامه‌نویسی Pythonدارد. این به این معنی است که برای استفاده بهینه از Locust و تغییرات پیکربندی پیشرفته، نیازمند تجربه و آشنایی با برنامه‌نویسی خواهید بود.

در زیر به حوزه‌های استفاده از Locust می‌پردازیم :

۴.۳.۳ حوزه‌های استفاده از Locust :

Locust به عنوان یک ابزار تست بار و عملکرد، در حوزه‌های مختلفی قابل استفاده است. در زیر، تعدادی از حوزه‌های استفاده از Locust را ذکر می‌کنم:

- تست عملکرد و بار میزبان‌های وب: با استفاده از Locust می‌توانید تست عملکرد و بار سایت‌ها، وبسرویس‌ها و سیستم‌های میزبان وب را انجام دهید. شما می‌توانید بار تراکنش‌های HTTP را به صورت همزمان و توزیع شده ایجاد کنید و عملکرد سیستم را در شرایط بار زیاد بررسی کنید.

- تست عملکرد :API با استفاده ازLocust، می‌توانید عملکرد و پاسخ‌دهی سریع و قابل اعتماد API ها را تست کنید. شما می‌توانید تعداد زیادی درخواست به صورت همزمان به API ارسال کرده و عملکرد و زمان پاسخ‌دهی آن را بررسی کنید.

- تست عملکرد سیستم‌های توزیع شده: اگر سیستم شما از طراحی توزیع شده برخوردار است، Locust می‌تواند برای تست عملکرد و پایداری این سیستم‌ها مفید باشد. شما می‌توانید بار و ترافیک بین قسمت‌های مختلف سیستم را تنظیم و بررسی کنید.

- تست عملکرد برنامه‌های موبایل Locust : قادر است تست عملکرد برنامه‌های موبایل را نیز انجام دهد. شما می‌توانید ترافیک مصنوعی مرتبط با برنامه‌های موبایل را شبیه‌سازی کنید و عملکرد و پاسخ‌دهی برنامه را بررسی کنید.

- تست عملکرد سیستم‌های:IoT اگر سیستم شما شامل دستگاه‌های اینترنت اشیاء (IoT) است، Locust می‌تواند در تست عملکرد و پایداری این سیستم‌ها مفید باشد. شما می‌توانید بار و ترافیک مرتبط با دستگاه‌های IoT را شبیه‌سازی کنید و عملکرد سیستم را بررسی کنید.

- تست عملکرد پروتکل‌های دیگر: علاوه بر HTTP وWebSocket ، Locust ممکن است برای تست عملکرد پروتکل‌های دیگر مانندMQTT، AMQP و غیره نیز قابل استفاده باشد. با توجه به توسعه پلاگین‌های Locust، قابلیت پشتیبانی از پروتکل‌های جدیدتر نیز ممکن است اضافه شود.

- تست عملکرد سیستم‌های تلفن همراه Locust: می‌تواند برای تست عملکرد و پاسخگویی سیستم‌های تلفن همراه مورد استفاده قرار بگیرد. می‌توانید بار و ترافیک مصنوعی را برای برنامه‌ها و سرویس‌های موبایل ایجاد کرده و عملکرد آنها را در شرایط بار زیاد بررسی کنید.

- تست عملکرد سیستم‌های بلاکچین: با رشد فناوری بلاکچین، استفاده از Locust برای تست عملکرد سیستم‌های بلاکچین نیز رایج شده است. می‌توانید بار تراکنش‌های بلاکچین را شبیه‌سازی کنید و عملکرد و زمان پاسخ‌دهی شبکه بلاکچین را بررسی کنید.

- تست عملکرد سیستم‌های ابری Locust: قابلیت تست عملکرد و پایداری سیستم‌های ابری را نیز دارد. می‌توانید بار و ترافیک بین ابرها، خدمات میانی و سرویس‌های ابری را تنظیم و عملکرد سیستم‌های ابری را بررسی کنید.

این تنها چند نمونه از حوزه‌های استفاده از Locust هستند. با توجه به قابلیت گسترش و توسعه Locust، می‌توان این ابزار را در صنایع و حوزه‌های دیگر نیز مورد استفاده قرار داد.

۵.۳.۳ محدودیتهای Locust

استفاده از Locust همچنین با محدودیت‌هایی همراه است. در زیر، تعدادی از محدودیت‌های استفاده از Locust را بررسی می‌کنیم:

- مصرف منابع سیستم Locust: برای تولید بار و ترافیک بالا نیاز به منابع سیستمی قابل توجهی دارد. زمانی که تعداد کاربران همزمان و درخواست‌ها بسیار زیاد می‌شوند، نیاز به پردازش و حافظه بیشتری برای اجرای Locust وجود دارد. در نتیجه، در سیستم‌های با منابع محدود، ممکن است با محدودیت‌های عملکردی مواجه شوید.

- تأخیر شبکه: استفاده از Locust به منزله ایجاد ترافیک شبکه بالا است. این ممکن است باعث افزایش تأخیر شبکه شود، به خصوص در صورتی که شبکه شما محدودیت‌های پهنای باند داشته باشد. در صورت استفاده از Locust در یک شبکه محدود، ممکن است برخی درخواست‌ها با تأخیر بیشتری پاسخ داده شوند و دقت نتایج تست کاهش یابد.

- محدودیت پروتکل‌ها Locust: برای استفاده از پروتکل‌های مختلف به پشتیبانی از آنها نیاز دارد. در حال حاضر، Locust به طور پیش فرض برای پروتکل‌های HTTP و WebSocket طراحی شده است. اگر نیاز به استفاده از پروتکل‌های دیگری مانندMQTT ، AMQP و غیره دارید، ممکن است نیازمند توسعه پلاگین‌ها یا اعمال تغییرات دستی در Locust باشید.

- عدم پشتیبانی از تمامی ابزارها Locust: ابزاری محبوب و گسترده است، اما همه ابزارها و فریمورک‌ها را پشتیبانی نمی‌کند. ممکن است برخی از ابزارها و فناوری‌های خاص در محیط Locust قابل استفاده نباشند یا نیازمند تنظیمات و اعمال تغییرات خاصی باشند.

۶.۳.۳ رقبای Locust :

Locust یکی از ابزارهای شناخته شده برای تست عملکرد و ترافیک شبکه است. اما همچنین رقبا و جایگزین‌هایی نیز برای Locust وجود دارند. در زیر به برخی از رقبا معروف Locust و مزایا و معایب Locust نسبت به آنها اشاره می‌کنم:

۱. ابزار Jmeter [21] :

- مزایا JMeter : یکی از ابزارهای قدرتمند تست عملکرد است که قدرت و انعطاف بیشتری در تنظیمات پیشرفته و نوشتن اسکریپت‌های سفارشی برای تست دارد. همچنین از پروتکل‌های مختلفی مانند HTTP،FTP، JDBC، SOAP ، REST و غیره پشتیبانی می‌کند.
- معایب JMeter : معمولاً برای تست‌های بزرگ و پیچیده مناسب است و برای تست‌های سبک و ساده ممکن است پیچیدگی اضافی داشته باشد. همچنین نیاز به آموزش و تجربه برنامه‌نویسی جاوا برای استفاده از قابلیت‌های پیشرفته JMeter است.

۲. ابزار Gatling [22] :

- مزایا Gatling :یک چهارچوب[۱۷]بسیار قدرتمند و سریع برای تست عملکرد است که از زبان برنامه‌نویسی Scala پشتیبانی می‌کند. این ابزار به عنوان یک سیستم توزیع شده عمل می‌کند و قابلیت ضبط و پخش ترافیک را داراست.

- معایب Gatling :برای برنامه‌نویسانی با تجربه قابل استفاده است و برای کسانی که با Scala آشنایی ندارند، ممکن است دشوار باشد. همچنین اگر پروژه شما به زبان برنامه‌نویسی دیگری مانند Pythonتوسعه یافته است، نیازمند ترجمه اسکریپت‌ها به Scala خواهید بود.

۳.Apache Bench (ab) [23]:

- مزایا Apache Bench :یک ابزار ساده و سبک برای تست عملکرد است که به صورت پیش‌فرض در بسته نرم‌افزاری Apache HTTP Server وجود دارد. این ابزار قابلیت ارسال درخواست‌ها با تعداد بالا را دارد و برای تست سریع و ساده عملکرد وب سرویس‌ها مفید است.

- معایب Apache Bench :از قابلیت‌های محدودی برخوردار است و معمولاً برای تست‌های ساده و ابتدایی مناسب است. برای تست‌های پیچیده‌تر و نیازمندی‌های پیشرفته‌تر، ابزارهای دیگری مانند Locustممکن است مناسب‌تر باشند.

[۱۷] Framework

جدول ۳-۲ به مقایسه این ابزار ها می پردازیم:

جدول ۳-۲ مقایسه ابزارهای ایجاد ترافیک[۲۵]

ابزار	زبان پشتیبانی‌شده	معایب	مزایا
Locust	Python	محدودیت پروتکل‌ها پیکربندی پیچیده	سادگی و آسانی استفاده مقیاس‌پذیری
Jmeter	Java	دارای پیکربندی پیچیده نیازمند تجربه برنامه‌نویسی با Java	پشتیبانی از پروتوکل‌های مختلف
Gatling	Scala, Python	نیاز به آشنایی با زبان Scala	استفاده به‌عنوان سیستم توزیع‌شده قابلیت ضبط و پخش ترافیک
Apache Bench	Scala	برای تست‌های پیچیده مناسب نیست. دارای قابلیت‌های محدود	مفید برای تست‌های ساده و ابتدایی ابزار ساده و سبک

فصل ۴: روش پیشنهادی

ما در این فصل به شرح آزمایش‌های انجام شده و نتایج به دست‌آمده می‌پردازیم. برای انجام آزمایش از ابزار Selenium، Locust، Wireshark استفاده کردیم و در هر آزمایش مشخصات کامپیوترها و سیستم‌عامل آنها نوشته شدهاست:

۱.۴ آزمایش اول

در این آزمایش می‌خواهیم به کمک ابزار اتوماسیون رفتار کاربران عادی را شبیه‌سازی و به کمک این شبیه‌سازی یک ترافیک از نوع هجوم ناگهانی کاربران تولید و عملکرد سرور را به کمک این ترافیک ارزیابی کنیم.

۱.۱.۴ شرایط آزمایش:

یک لپتاپ با مشخصات زیر:

هارد: 512 GB

RAM: 16 GB

پردازنده: 12th Gen intel Core i7-12650H

سیستم عامل: Windows 11 home

۲.۱.۴ نرم‌افزارهای مورد استفاده :

Python: نسخه ۳.۱۰[۱۸]

Selenium: نسخه ۴.۱.۰

Wireshark: نسخه ۴.۰.۵

[۱۸] Version

۳.۱.۴ شرح آزمایش:

ابتدا کار یک سایت که شامل پرکردن یک فرم ساده در آن سایت است را در نظر گرفتیم و با استفاده از وایرشارک، ترافیکی که یک کاربر در حین تعامل با سایت تولید کرده را ضبط می‌کنیم. این ترافیک، ترافیکی از نوع طبیعی است. سپس با استفاده از سلنیوم، یک اسکریپت می‌نویسیم که رفتاری مشابه یک کاربر داشته باشد و این ترافیک را نیز ضبط می‌کنیم. ترافیک به دست آمده، ترافیکی از نوع غیرطبیعی و آزمایشگاهی است که با استفاده از اسکریپت، رفتار یک کاربر را شبیه‌سازی و تقلید می‌کند. سپس به کمک پایتون و این ترافیک به دست آمده رفتار چندین کاربر را شبیه سازی می کنیم و کارایی سروری که سایت بر رو آن سرور قرار داده شده است را به دست می‌آوریم.

استفاده از Selenium به منظور اتوماسیون تست وب، نیازمند تعامل با مرورگر وب و تعریف فعالیت‌های موردنظر در وب سایت است. در زیر، نحوه استفاده از Selenium را به صورت کامل شرح می‌دهیم:

نصب Selenium :

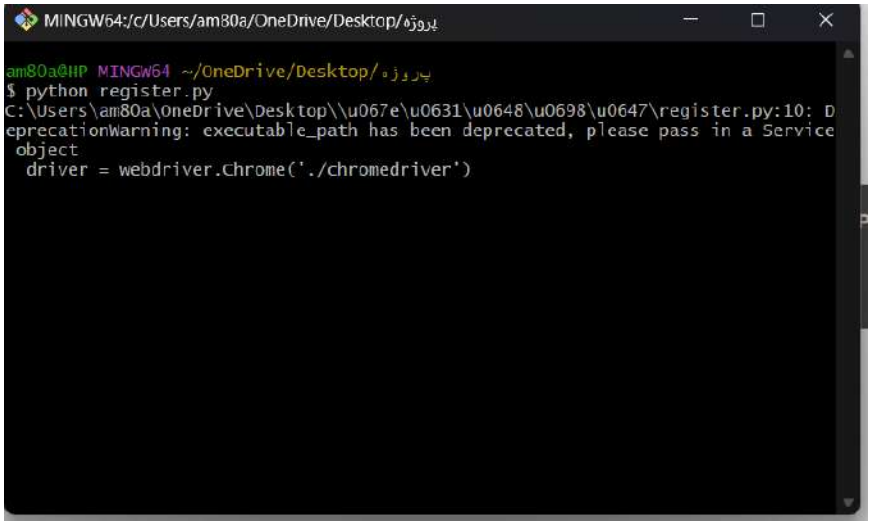
- ابتدا باید Selenium را در محیط خود نصب کنیم.

راه‌اندازی WebDriver:

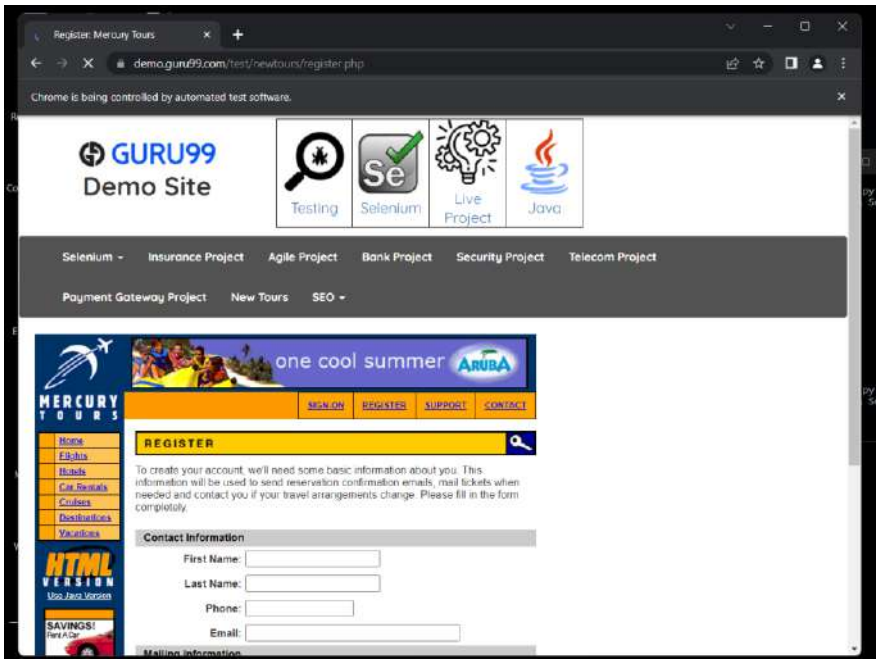
WebDriver یک رابط بین Selenium و مرورگرها است که امکان کنترل مرورگر را فراهم می‌کند. برای هر مرورگر ممکن است نسخه خاصی از WebDriver لازم باشد. برای مثال، برای مرورگر Chrome نیاز به ChromeDriverداریم.

پس نصب این دو ابزار یک کد به زبان پایتون مینویسم و عملکرد یک کاربر را شبیه‌سازی می‌کنیم و این کد را چندین بار با فاصله زمانی تعیین شده اجرا می‌کنیم. شکل ۴-۱ و ۴-۲ نمایی از عملکرد سلنیوم است و شکل ۴-۳ نیز نتایج تست برای ۵ کاربر را نشان می‌دهد.

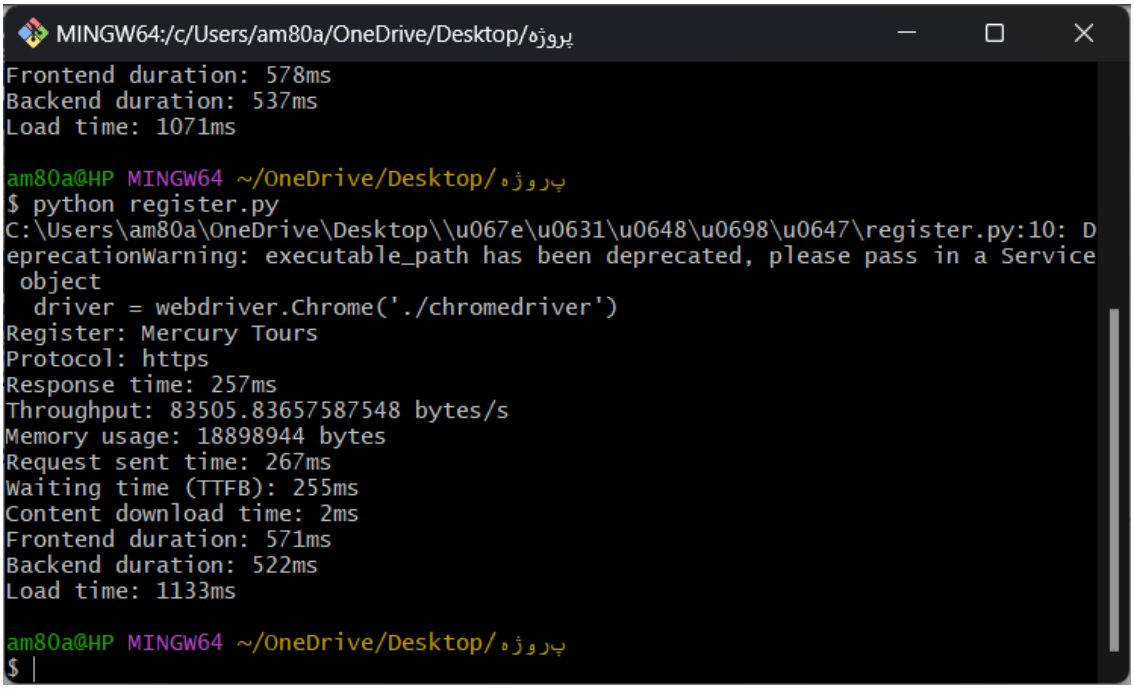
در فصل بعد نتایج آزمایش اول را مورد ارزیابی قرار می‌دهیم.



شکل۴-۱ نمایی از اجرای سلنیوم



شکل۴-۲ نمایی از انجام تست خودکار توسط درایور



شکل۴-۳ نتیجه ارزیابی سایت

۲.۴ آزمایش دوم:

در این آزمایش می‌خواهیم به کمک ابزار تولید بار، رفتار تعداد کاربران زیادی در بازه زمانی کوتاه را شبیه‌سازی کنیم و در نهایت یک ترافیک از این هجوم کاربران تولید و همچنین عملکرد سرور را به کمک این ابزار ارزیابی کنیم.

۱.۲.۴ شرایط آزمایش:

یک کامپیوتر (به عنوان مشتری [۱۹]) و یک سرور (به عنوان میزبان[۲۰]) با مشخصات زیر

هارد: 2.5 TB

RAM: 32.0 GB

پردازنده: 12th Gen intel Core i7-12700K

[۱۹] Client

[۲۰] Host

سیستم‌عامل: Ubuntu 22.04.2 LTS

۲.۲.۴ نرم‌افزارهای مورد استفاده :

Python: نسخه ۳.۱۰

Locust: نسخه ۲.۱۵.۱

Wireshark: نسخه ۴.۰.۵

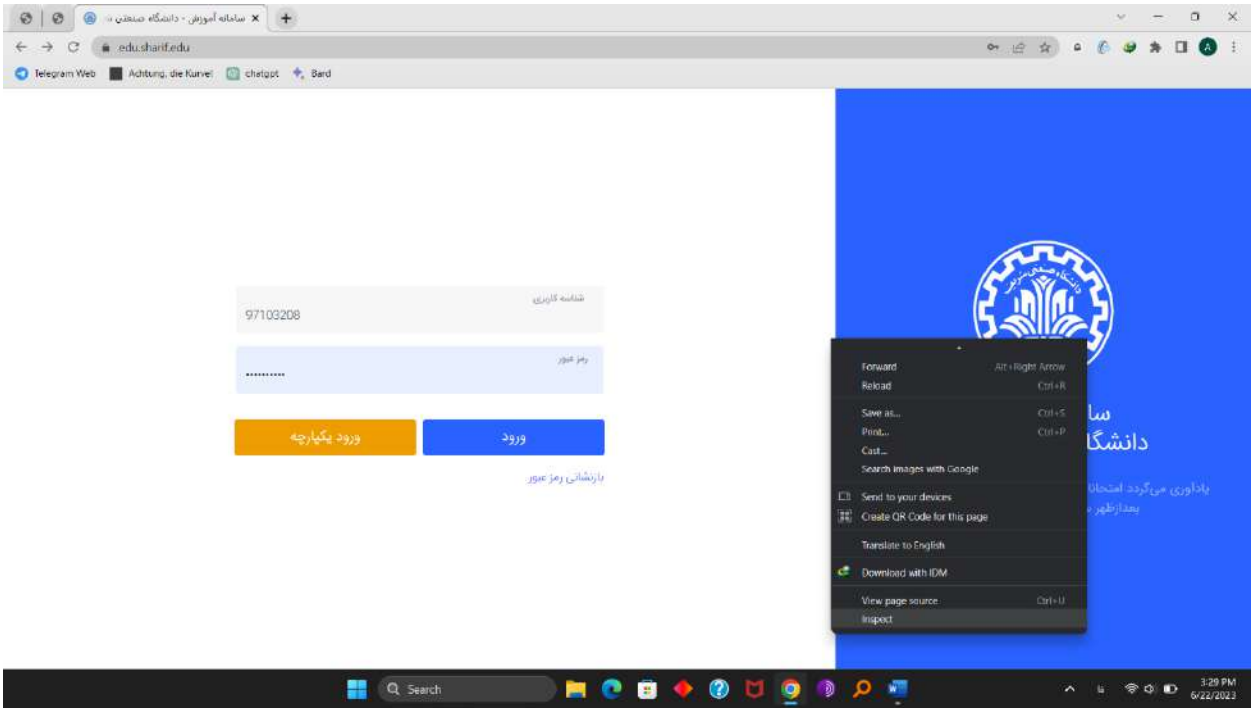
۳.۲.۴ شرح آزمایش:

ابتدا بر روی سرور موجود در آزمایش شبکه دانشگاه یک وب‌سرور ساده که فقط دارای یک تیتر ساده بود ایجاد شد. حال با نرم‌افزار پایتون یک اسکریپت ساده می‌نویسیم که با Locust یک هجوم را شبیه‌سازی کند و در نهایت با Wiresharkترافیک را ضبط می‌کنیم.

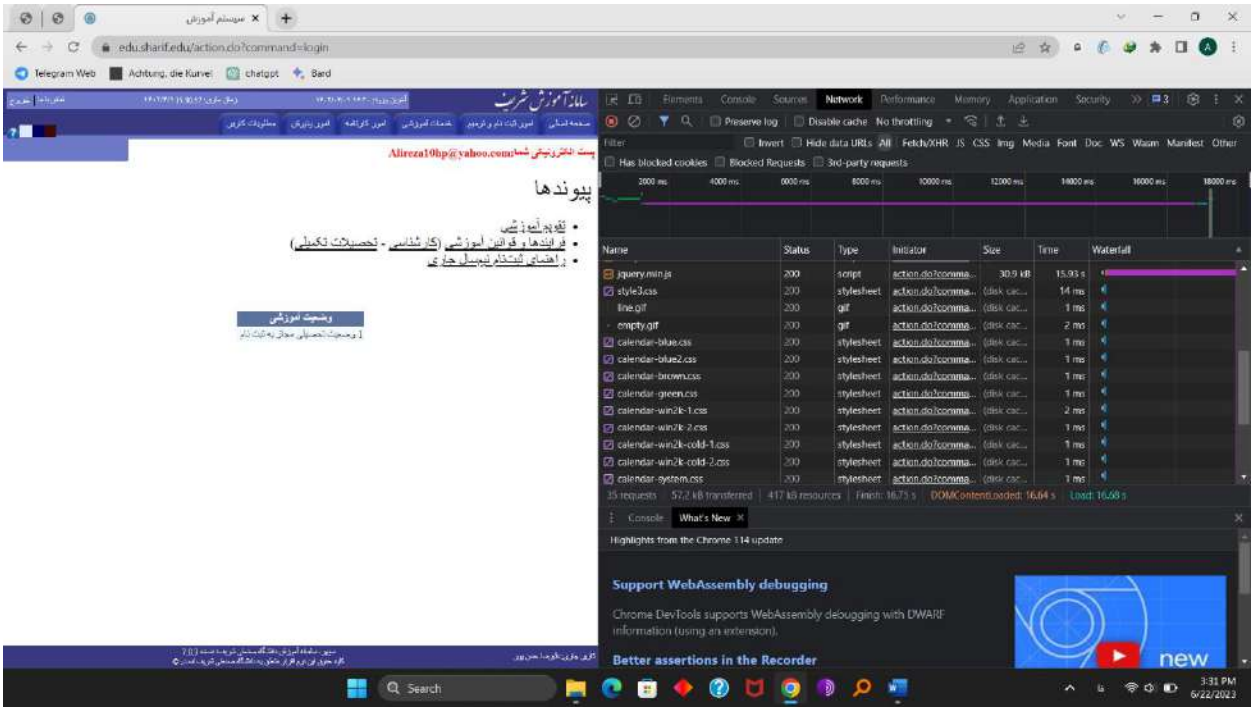
برای سایت‌های پیچیده‌تر و دارای المان های بیشتر نیز می‌توانیم مانند روش قبل عمل کنیم و با نوشتن اسکریپت پایتون سایت را مورد هجوم قرار دهیم؛ همچنین، می‌توانیم به روش دیگری نیز این اسکریپت را بنویسیم:

ابتدا وارد سایت موردنظر می‌شویم (شکل ۴-۴ و ۴-۵) و سپس با استفاده از ابزار موجود در قسمت "Inspect Element"، رفتار کاربر یا به عبارت دیگر، لاگ [۲۱]سایت در هنگام حضور کاربر در سایت را به فرمت Har. به عنوان خروجی استخراج می‌کنیم (شکل۴-۶ و ۴-۷).

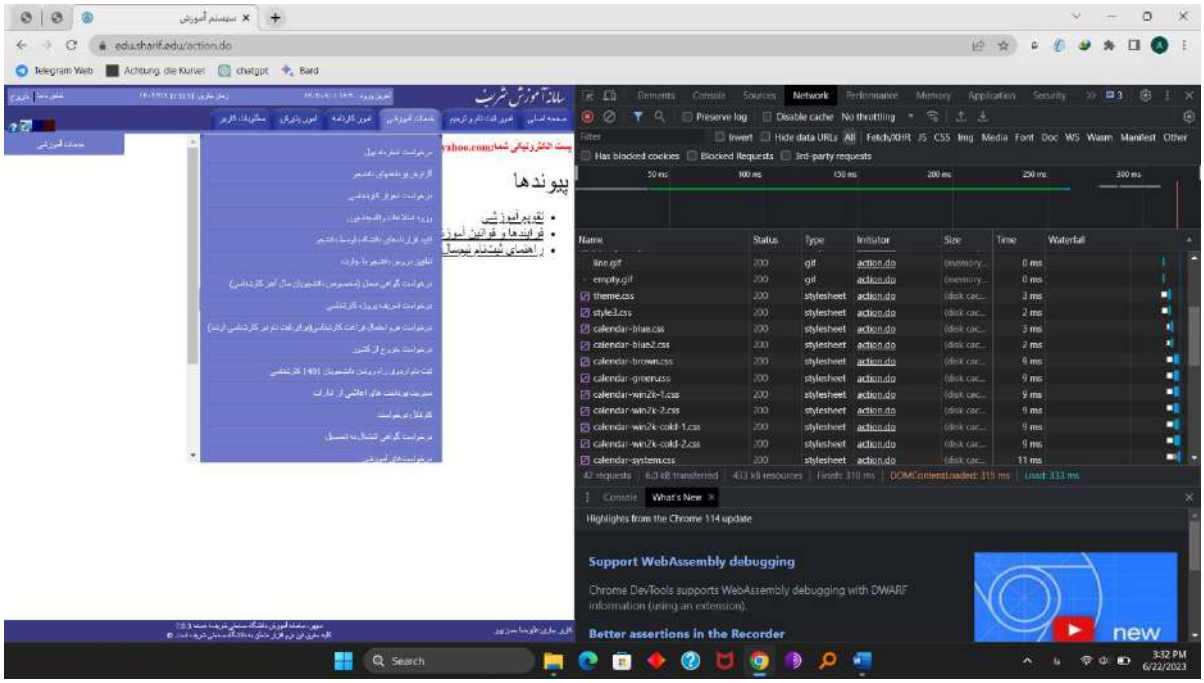
[۲۱] Log



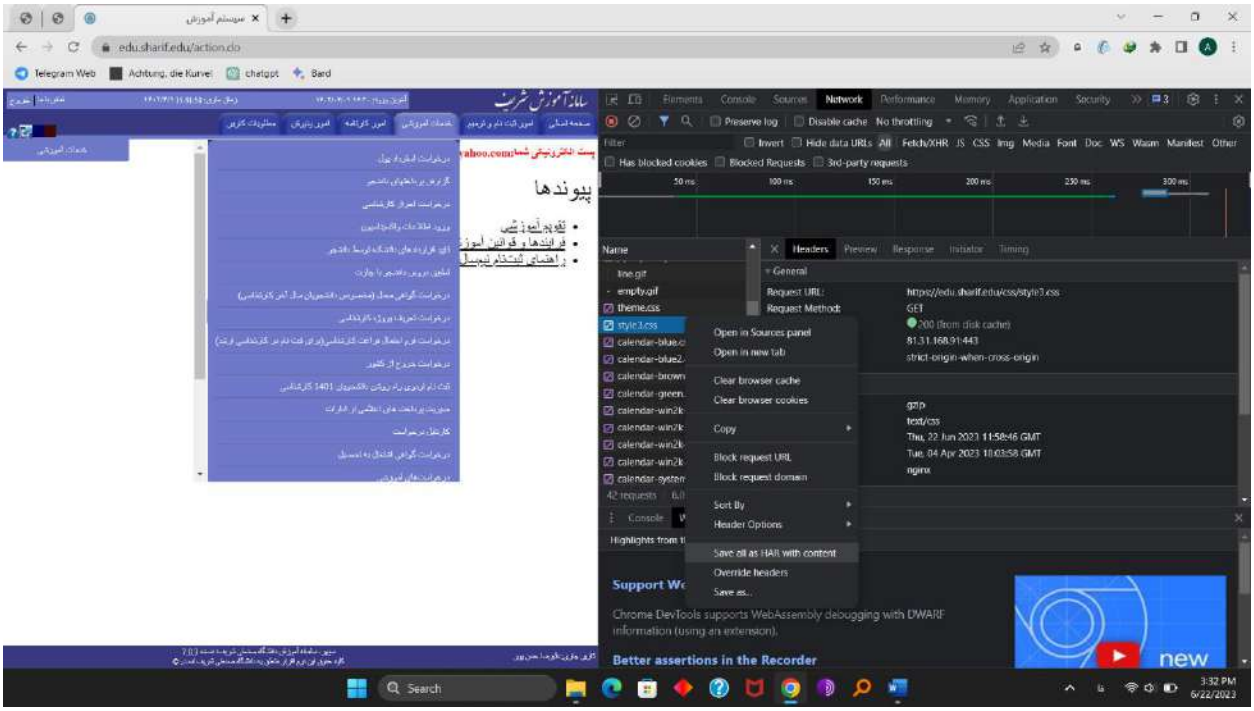
شکل ۴-۴ ورود به inspect element



شکل ۴-۵ اجرای فعالیت کاربر



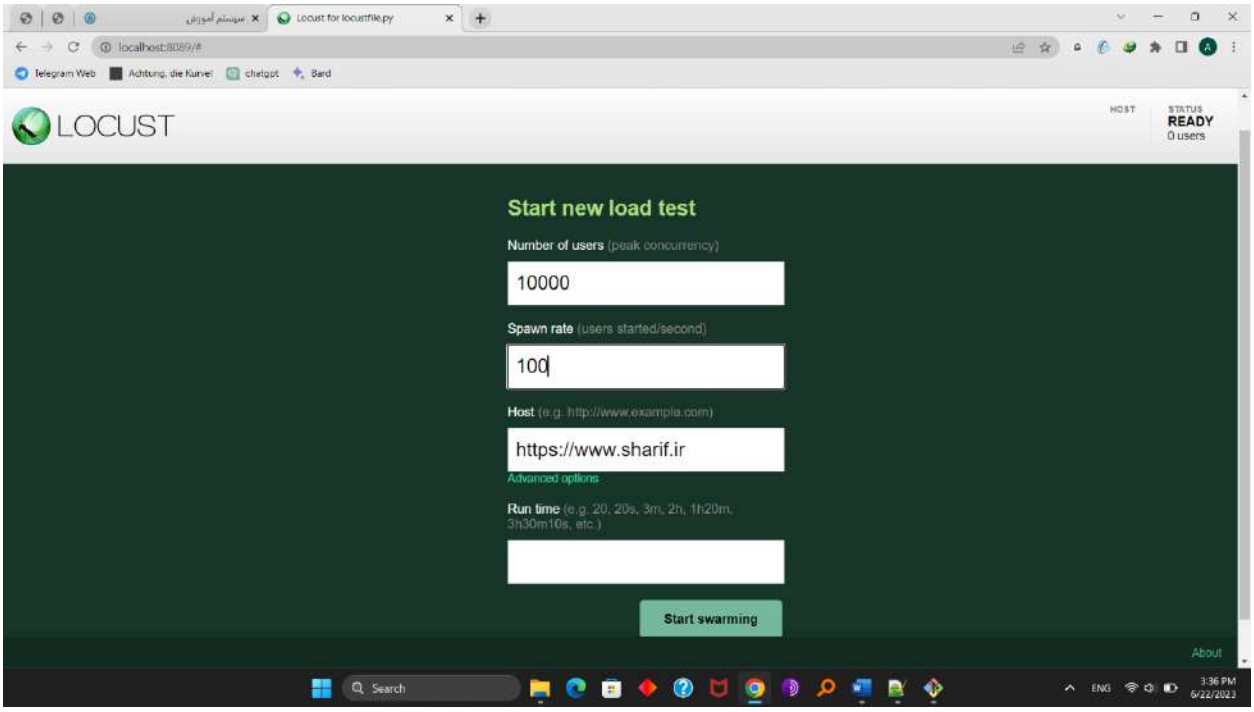
شکل۴-۶ گرفتن لاگ کاربر



شکل ۴-۷ ذخیره لاگ کاربر

در نهایت، لاگ به‌دست آمده را تبدیل به یک اسکریپت پایتون می‌کنیم. پس از اجرای آن و مشخص کردن تعداد کاربران شبیه‌سازی شده، تعداد کاربرانی که در لحظه وارد می شوند و زمان کل فرآیند شبیه سازی مانند شکل ۴-۸ میتوانیم خروجی این ترافیک را به صورت گرافیکی و همچنین ترافیک ضبط شده مشاهده کرده و نتایج به دست آمده را مقایسه کنیم.

در فصل بعد نتایج آزمایش را مورد ارزیابی قرار می‌دهیم.



شکل ۴-۸ اجرای تست با locust

Selenium و Locust دو ابزار مختلف هستند که برای تست و ارزیابی عملکرد وب سایت‌ها استفاده می‌شوند. این دو ابزار دارای مزایا و محدودیت‌های خود هستند و در زمینه‌های متفاوتی مورد استفاده قرار می‌گیرند. در ادامه، نتایج حاصل از Selenium و Locust را به صورت مقایسه‌ای بررسی می‌کنم:

Selenium:

- Selenium یک ابزار متن باز و قابل توسعه است که امکان اتوماسیون وب را فراهم می‌کند.

- با استفاده از Selenium می‌توانیم فعالیت‌های تست وب از جمله مرور وب سایت، ورود به حساب کاربری، وارد کردن فرم‌ها و صحت نتایج را تعریف و اجرا کنیم.

- Selenium از زبان‌های برنامه‌نویسی مختلفی مانند Python، Java، C# و Ruby پشتیبانی می‌کند.

- از طریق Selenium WebDriver، می‌توانید با مرورگرهای مختلفی مانند Chrome، Firefox، Safari و IE تعامل کنید.

- Selenium به عنوان یک ابزار برای ارزیابی عملکرد وبسایت‌ها، ایجاد و اجرای تست‌های پویا و تعاملی مناسب است.

Locust:

- Locust نیز یک ابزار متن باز است که برای تست عملکرد وب سایت‌ها و برنامه‌های تحت وب استفاده می‌شود.

- Locust بر پایه زبان برنامه‌نویسی Python ساخته شده است و بسیار ساده و قابل فهم است.

- Locust به شما امکان می‌دهد تعداد زیادی کاربر همزمان (به عنوان کلاینت‌ها) را شبیه‌سازی کنید و از طریق این کار عملکرد وب سایت را تحت بار قرار دهید.

- Locust قابلیت ضبط و تولید گزارش‌های جزئی و جامع در مورد عملکرد وب سایت را دارد.

- با استفاده از زبان Python، می‌توانید تنظیمات و سفارشی‌سازی‌های بیشتری را در Locust انجام دهید.

در مقایسه با یکدیگر، Selenium و Locust در زمینه‌های متفاوتی کاربرد دارند. Selenium بیشتر برای اتوماسیون تست عملکرد وبسایت‌ها استفاده می‌شود و امکان تعامل با مرورگرها را فراهم می‌کند. Locust به شما امکان می‌دهد تعداد زیادی کاربر همزمان را شبیه‌سازی کنید و برنامه یا وب سایت را تحت بار قرار دهید. هر کدام میتوانند یک سایت را از جنبه‌های مختلف مورد ارزیابی قرار دهند.

برای انتخاب بین این دو ابزار، باید نیازها و هدف اصلی تست خود را مورد بررسی قرار داده و با توجه به نیازهای خود، ابزار مناسب را انتخاب کنید.

به کمک ابزار Wireshark می‌توانیم ترافیک تولید شده هنگام آزمایش را ضبط کرد و می‌توان در سطح بسته مبادله شده در شبکه نیز سایت را مورد ارزیابی قرار داد.

فصل ۵: ارزیابی نتایج، نتیجه گیری

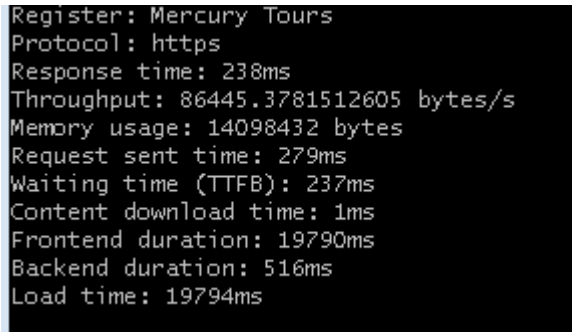
۱.۵ نتایج آزمایش اول

با توجه به اینکه مرورگر کروم مصرف بسیار بالای حافظه دارد، در نهایت میتوان ۱۰۰۰ کاربر را به صورت همزمان شبیه‌سازی کرد و خروجی‌های به دست‌آمده به صورت زیر شده‌است:

می‌توان متغیرهای زیر را تغییر داد و نتایج را مقایسه کرد:

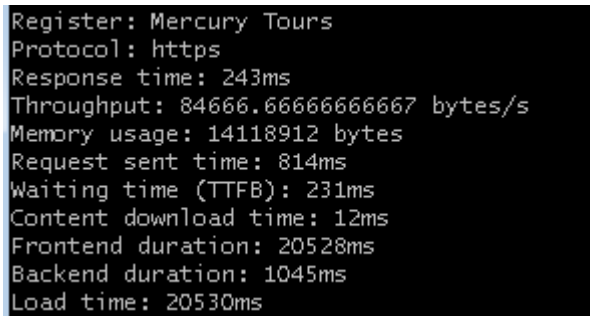
- تعداد کاربران
- فاصله زمانی بین پر کردن بخش‌های مختلف فرم
- فاصله زمانی میان ورود کاربران

نتایج آزمایش برای هجوم ۲۰ کاربر



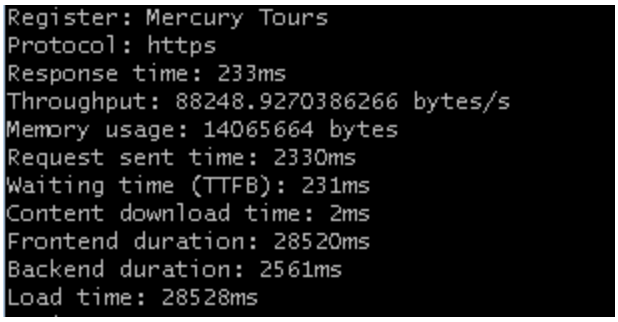
شکل ۵-۱ هجوم ۲۰ کاربر

نتایج آزمایش برای هجوم ۵۰ کاربر



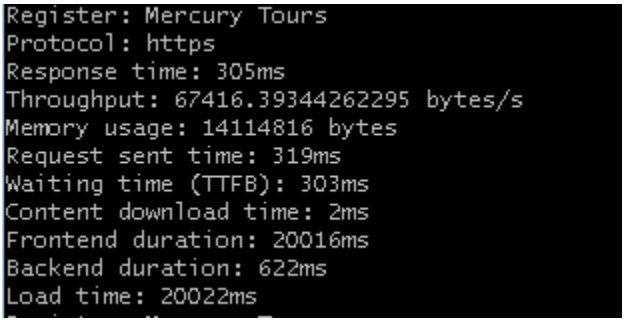
شکل ۵-۲ هجوم ۵۰ کاربر

نتایج آزمایش برای هجوم ۱۰۰ کاربر



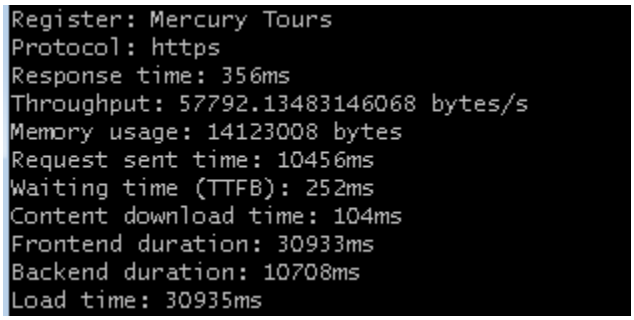
شکل ۵-۳ هجوم ۱۰۰ کاربر

نتایج آزمایش برای هجوم ۱۵۰ کاربر



شکل ۵-۴ هجوم ۱۵۰ کاربر

نتایج آزمایش برای هجوم ۲۰۰ کاربر

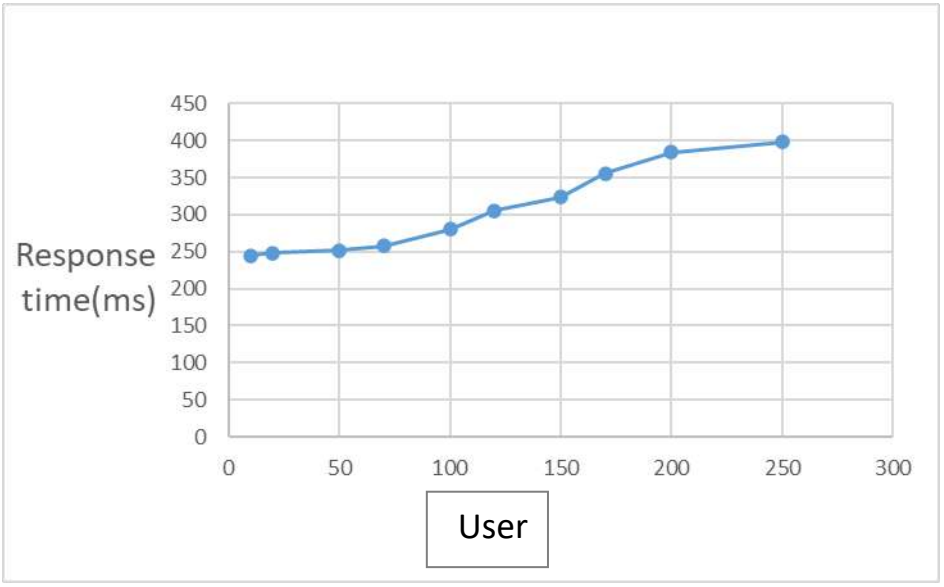


شکل ۵-۵ هجوم ۲۰۰ کاربر

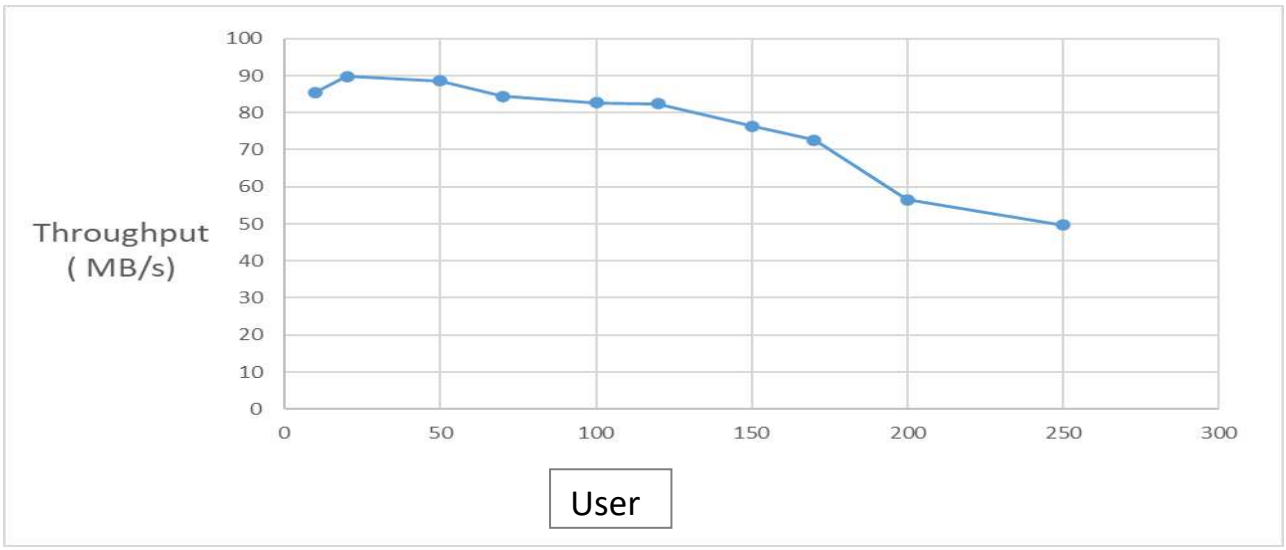
آزمایش برای ۲۷۰ کاربر به علت استفاده کامل از RAM نیمه‌کاره باقی ماند.

جدول ۵-۱ نتایج به دست آمده از آزمایش اول

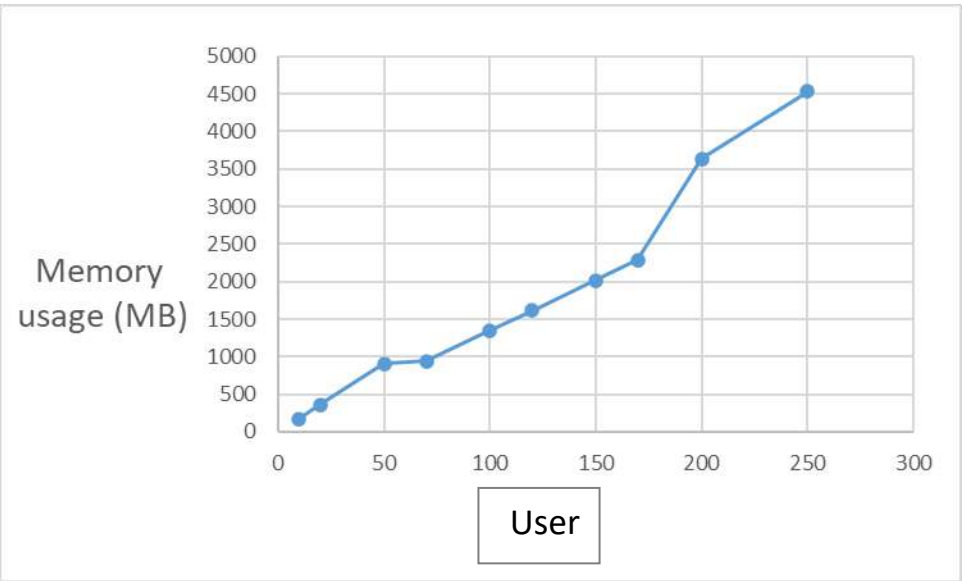
User	Response time(ms)	Throughput (MB/s)	Memory usage (MB)	Req send time(ms)	Waiting time (TTFB)	Frontend Duration(ms)	Backend Duration (ms)	Load Time (ms)
10	245	85.54199219	175.8984375	265	244	565	509	965
20	248	89.81542969	361.640625	399	383	817	782	7071
50	252	88.62792969	908.203125	1682	359	2117	821	16921
70	258	84.41933594	941.171875	279	237	1979	958	19794
100	280	82.68232422	1346.484375	814	231	2052	1045	20530
120	305	82.43789063	1615.3125	319	303	2001	1875	20022
150	324	76.41503906	2012.109375	2330	231	2852	2561	28528
170	356	72.578125	2289.6875	1045	252	3093	2768	30935
200	384	56.43759766	3635.9375	1508	249	5289	3562	31182
250	398	49.60058594	4529.296875	1860	397	5901	5842	56957



شکل ۵-۶ نمودار زمان پاسخدهی نسبت به تعداد متقاضیان



شکل ۵-۷ نمودار گذرهی نسبت به تعداد متقاضیان



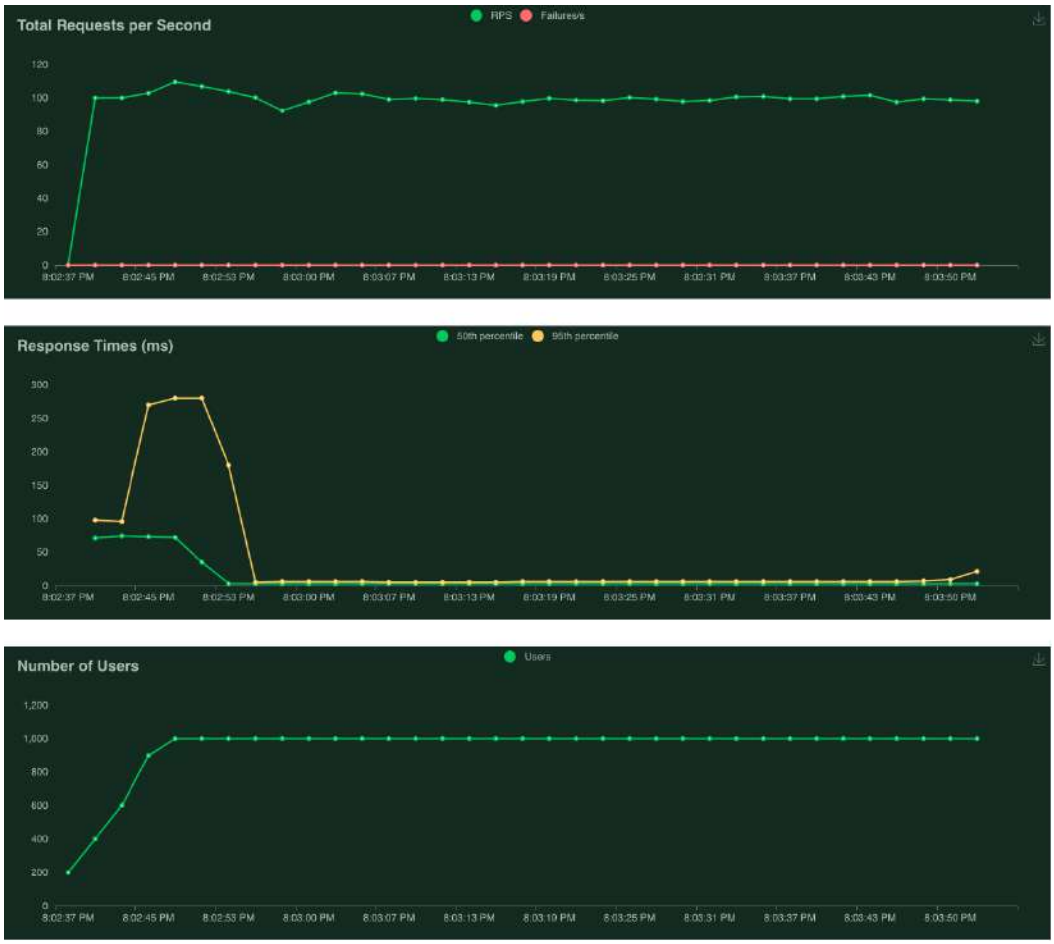
شکل ۵-۸ میزان حافظه مصرفی نسبت به تعداد متقاضیان

از این آزمایش می‌توان فهمید که با افزایش تعداد کاربران میزان مصرف حافظه به صورت تقریبا خطی افزایش پیدا می‌کند (نمودار ۵-۳)، میزان زمان پاسخدهی سایت نسبت به درخواست‌ها تا ۷۰ کاربر تقریبا به صورت ثابت است ولی پس از آن به صورت خطی افزایش می‌یابد (نمودار ۵-۱) و میزان گذردهی سایت تا ۱۲۰ کاربر تقریبا ثابت است ولی پس از آن به‌شدت افت می‌کند(نمودار ۵-۲).

۲.۵ نتایج آزمایش دوم

این آزمایش با توجه به اینکه در سندباکس اجرا میشود حافظه کمتری به ازای هر کابر اشغال می‌کند و میتوان کاربران بیشتری نسبت به Selenium را اختصاص داد؛ همچنین می‌توان تعداد کاربران، نرخ ورود کاربران و مدت زمان انجام آزمایش را تعیین کرد.

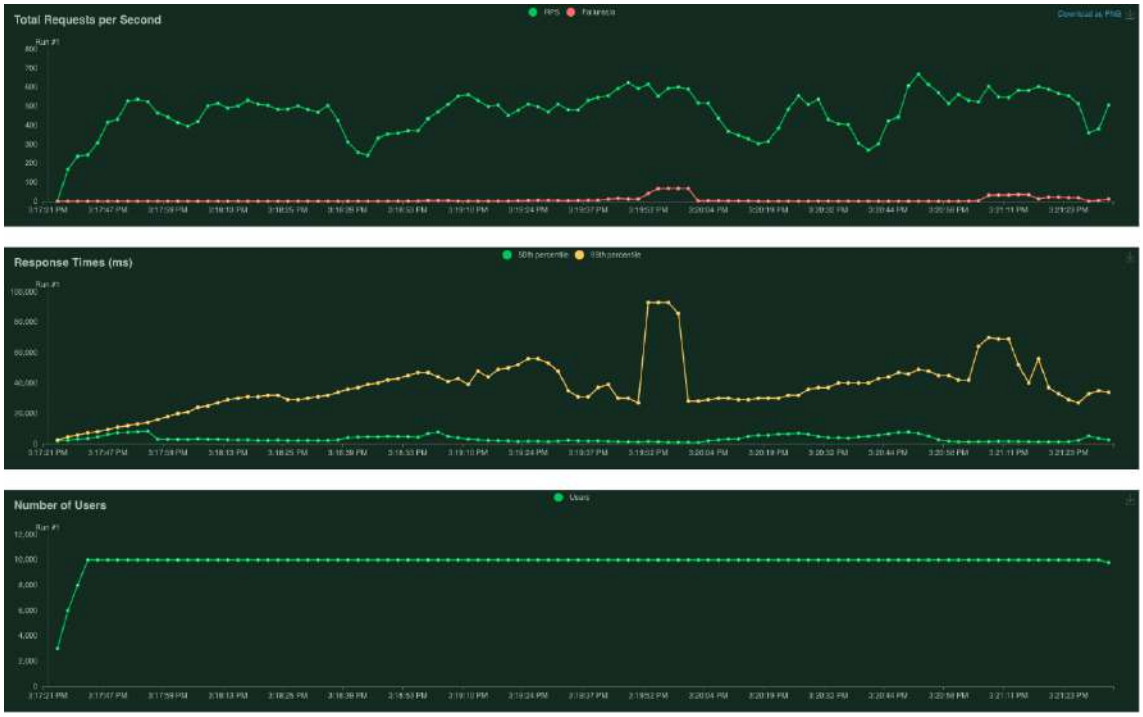
نتایج آزمایش برای ۱۰۰۰ کاربر با نرخ ورود ۱۰۰ کاربر بر ثانیه:



شکل ۵-۹ هجوم ۱۰۰۰ کاربر با نرخ ۱۰۰ کاربر بر ثانیه

در این آزمایش پایداری سرور ثابت است و همیطور نرخ شاهد زمان پاسخگویی پایینی هستیم.

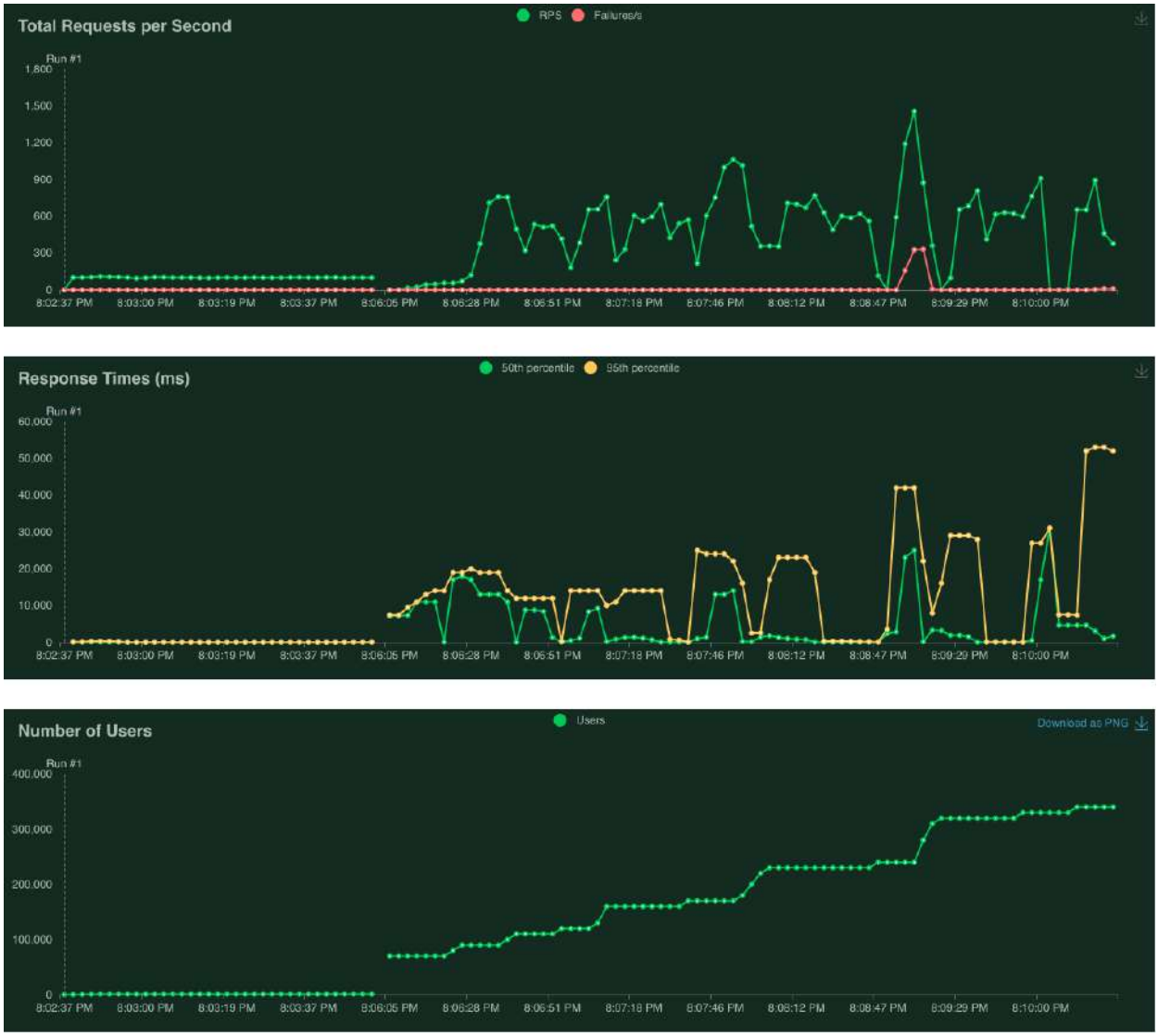
نتایج آزمایش برای ۱۰۰۰۰ کاربر با نرخ ورود ۱۰۰۰ کاربر بر ثانیه:



شکل ۵-۱۰ هجوم ۱۰۰۰ کاربر با نرخ ۱۰۰ کاربر بر ثانیه

پس از ورود ۱۰۰۰۰ کاربر و نرخ ۷۵۰ درخواست بر ثانیه عملکرد سرور از لحاظ پاسخگویی دچار نوسان می‌شود ولی به طور کامل مختل نمی‌شود و پس از آن دچار افت کیفیت می‌شود.

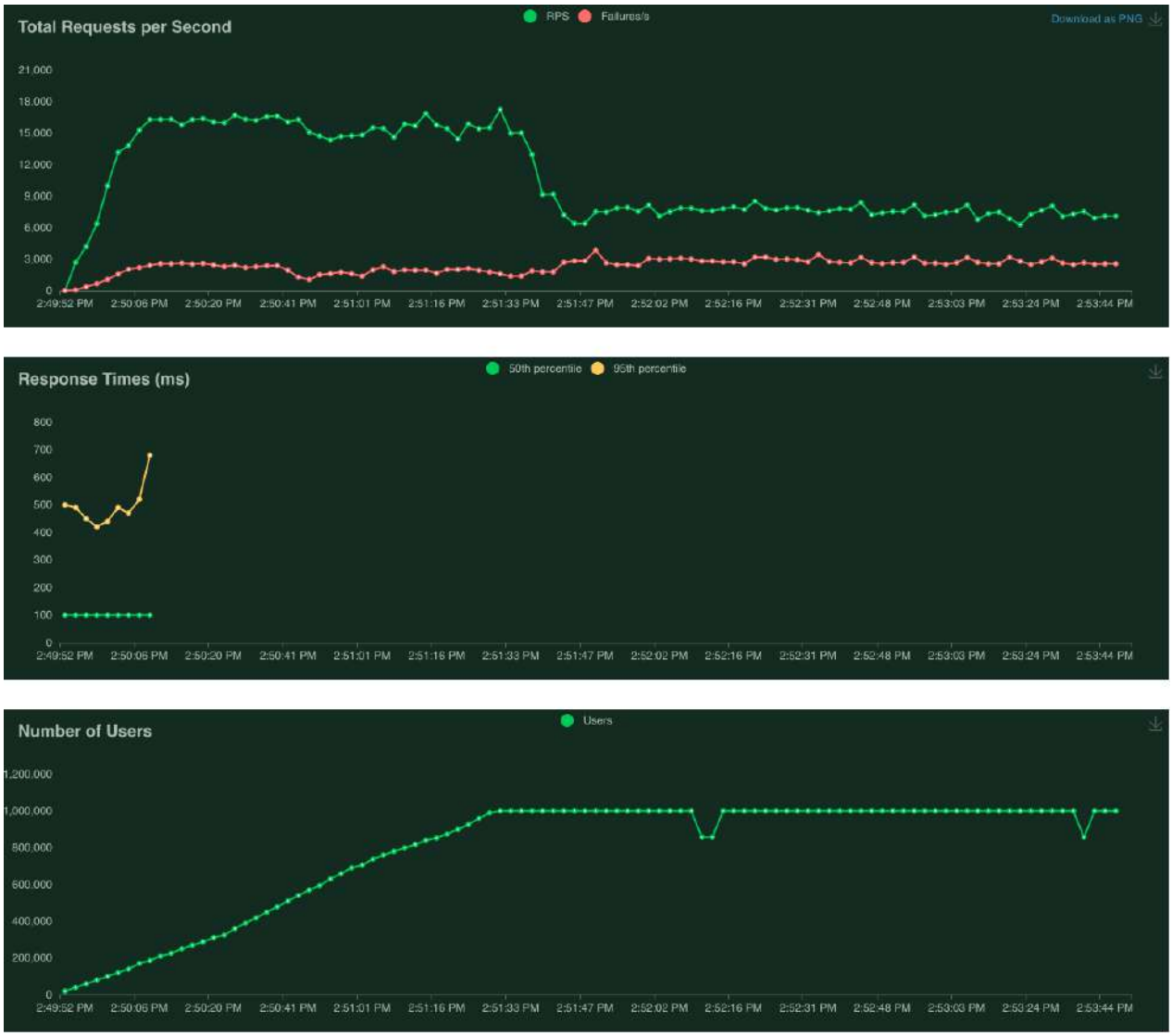
نتایج آزمایش برای ۱۰۰۰۰۰ کاربر با نرخ ورود ۱۰۰۰۰ کاربر بر ثانیه:



شکل ۵-۱۱ هجوم ۱۰۰۰۰۰ کاربر با نرخ ورود ۱۰۰۰۰ کاربر بر ثانیه

در این آزمایش پس از ورود ۲۰۰۰۰ کاربر ناگهان دچار نوسان شدید می‌شود به طوری که نرخ پاسخگویی تا ۲ ثانیه هم بالا می‌رود.

نتایج آزمایش برای ۱۰۰۰۰۰۰ کاربر با نرخ ورود ۱۰۰۰۰ کاربر بر ثانیه:



شکل ۵-۱۲ هجوم ۱۰۰۰۰۰۰ کاربر با نرخ ورود ۱۰۰۰۰ کاربر بر ثانیه

در این آزمایش پس از ورود ۲۰۰۰۰۰ کاربر سرور به طور کامل مختل می‌شود به طوری که دیگر پاسخگو نیست و میزان زمان پاسخگویی دیگر تعریف نمی‌شود.

پس از این آزمایش می‌توان این نتیجه را به دست آورد که ظرفیت کلی سرور در نهایت ۲۰۰۰۰۰ کاربر است ولی از ۱۰۰۰۰ کاربر و نرخ ۷۵۰ درخواست بر ثانیه به بالا کیفیت پاسخگویی سایت به‌شدت کاهش میابد.

هر دو آزمایش علاوه بر خروجی گرافیکی خروجی به صورت فایل pcap به ازای هر آزمایش نیز دارند.

فصل ۶: کارهای آینده

۱.۶ کشف گلوگاه:

کشف گلوگاه‌ها در سایت‌ها با استفاده از ابزار تولید بار می‌تواند کمک کند تا نقاط ضعف و نقاط قوت عملکردی سایت خود را شناسایی کنید. با تولید بار مصنوعی و افزایش ترافیک سایت، می‌توانید عملکرد سایت در شرایط مختلف را مورد بررسی قرار داده و گلوگاه‌ها را کشف کنید.

برای انجام این کار، می‌توان از ابزارهایی مانند Locust استفاده کنیم. این ابزار قابلیت تولید بار را دارد و با ارسال درخواست‌های بیشتر به سرور، میزان ترافیک و بار را افزایش می‌دهد.

برای کشف گلوگاه‌ها و نقاط ضعف سایت با استفاده از ابزار تولید بار، مراحل زیر را می‌توان دنبال کرد:

۱.۱.۶ تعیین هدف:

تعیین هدف از تست بار و کشف گلوگاه‌ها مهم است. ممکن است هدف ما کشف عملکرد سرور درشرایط بار بالا، تست پایداری سیستم، یا بررسی زمان پاسخ سایت در بار زیاد باشد

۲.۱.۶ طراحی سناریوهای بار:

بر اساس هدف تست، سناریوهای بار را طراحی می‌کنیم. این سناریوها شامل تعداد کاربران، نوع درخواست‌ها، توزیع بار، زمان و نحوه ارسال درخواست‌ها و سایر پارامترهای مربوطه است.

۳.۱.۶ پیکربندی ابزار تولید بار:

ابزار انتخابی را پیکربندی می‌کنیم. این شامل تنظیمات سرور مقصد، پروتکل‌های مورد استفاده، تعداد کاربران و سایر تنظیمات مربوط به بار است.

۴.۱.۶ اجرای تست بار:

سناریوهای بار را با استفاده از ابزار تولید بار اجرا می‌کنیم. این به ما امکان می‌دهد تا بار مصنوعی را روی سرورها و سایت خود ایجاد کنیم.

۵.۱.۶ پایش **۲۲**و تحلیل نتایج:

در حین اجرای تست بار، مانیتورینگ و ثبت نتایج از عملکرد سایت انجام شود. نتایج برای شناسایی گلوگاه‌ها، زمان پاسخ سایت، ظرفیت سرور و سایر معیارهای عملکرد مورد بررسی قرار می‌گیرند.

۷.۱.۶ تجزیه و تحلیل نتایج:

نتایج بدست آمده را تجزیه و تحلیل کنیم و گلوگاه‌ها و نقاط ضعف سایت را شناسایی کنیم. این شامل شناسایی زمان پاسخ بالا، افت کیفیت سرویس، مشکلات مرتبط با پهنای باند و سرور و سایر مشکلات ممکن است.

۲.۶ تولید ترافیک شبکه به کمک هوش مصنوعی

استفاده از هوش مصنوعی برای تولید ترافیک شبکه و ارزیابی سرورها به عنوان یک روش مؤثر برای شبیه‌سازی شرایط واقعی ترافیک و انجام آزمایش‌های تست می‌باشد. با استفاده از الگوریتم‌های هوش مصنوعی مانند شبکه‌های عصبی، الگوریتم‌های ژنتیک، یادگیری ماشین و سایر روش‌های هوش مصنوعی، می‌توانیم الگوهای ترافیک شبکه را شبیه‌سازی کنیم و عملکرد سرورها را ارزیابی کنیم.

در ادامه، روند عملکرد این روش را برای تولید ترافیک شبکه به کمک هوش مصنوعی توضیح می‌دهیم:

۱.۲.۶ جمع‌آوری داده‌ها:

در این مرحله، ابتدا باید داده‌های مربوط به الگوهای ترافیک شبکه را جمع‌آوری کنیم. این شامل الگوهای ترافیک در طول زمان، فرکانس درخواست‌ها، نوع درخواست‌ها، ویژگی‌های شبکه و سایر اطلاعات مرتبط است.

۲.۲.۶ پیش‌پردازش داده‌ها:

پس از جمع‌آوری داده‌ها، باید آن‌ها را پیش‌پردازش کنیم. این شامل تفکیک داده‌ها، مقیاس‌بندی، تبدیل فرمت‌ها و استخراج ویژگی‌های مهم می‌شود.

---

[۲۲] Monitoring

۳.۲.۶ طراحی و آموزش مدل:

در این مرحله، یک مدل هوش مصنوعی مانند شبکه‌های عصبی، الگوریتم ژنتیک یا سایر مدل‌های هوش مصنوعی طراحی و آموزش داده می‌شود. این مدل با استفاده از داده‌های جمع‌آوری شده، الگوهای ترافیک شبکه را یاد می‌گیرد و قادر به تولید ترافیک مشابه است.

۴.۲.۶ تولید ترافیک:

بعد از آموزش مدل، می‌توانیم از آن برای تولید ترافیک شبکه استفاده کنیم. با ورودی دادن به مدل، مدل ترافیک مشابه با الگوهای آموزش‌دیده را تولید می‌کند.

۵.۲.۶ ارزیابی سرور:

ترافیک تولید شده توسط مدل هوش مصنوعی را برای ارزیابی عملکرد سرور استفاده می‌کنیم. می‌توانیم معیارهایی مانند زمان پاسخ سرور، تاخیر بارگیری صفحات، ترافیک پذیرفته شده و سایر معیارهای عملکرد را بررسی کنیم.

۶.۲.۶ بهبود و بهینه‌سازی:

با تکرار مراحل بالا و تنظیم پارامترهای مدل هوش مصنوعی، می‌توانیم بهبودهای لازم را در عملکرد سرور و ترافیک تولید شده داشته باشیم.

مزایای استفاده از هوش مصنوعی برای تولید ترافیک شبکه عبارتند از:

- افزایش کارایی و بهینه‌سازی ترافیک شبکه
- کاهش زمان و هزینه‌های مربوط به مدیریت ترافیک شبکه
- پیش‌بینی دقیق‌تر و بهتر از الگوهای ترافیک
- قابلیت تطبیق با تغییر شرایط و نیازهای شبکه

معایب استفاده از هوش مصنوعی برای تولید ترافیک شبکه شامل موارد زیر است:

- نیاز به داده‌های بزرگ و جامع برای آموزش الگوریتم‌ها

- پیچیدگی برخی الگوریتم‌ها و نیاز به تجربه و دانش تخصصی
- احتمال اشتباه و خطا در تصمیمات هوشمندانه الگوریتم‌ها

استفاده از هوش مصنوعی برای تولید ترافیک شبکه و ارزیابی سرور به ما کمک می‌کند تا عملکرد و کارایی سرورها را در شرایط نزدیکتر به واقعیت بسنجیم و مشکلات و گلوگاه‌های سرور را شناسایی و بهبود ببخشیم.

مراجع

[1] R. Abu, Moheeb, Z. Jay, M. Fabian and T. Andreas, "A multifaceted approach to understanding the botnet phenomenon," in 6th ACM SIGCOMM conference on Internet measurement, Rio de Janeiro, Brazil, October 2007.

[2] P. Wendell and M. J. Freedman, "Going Viral: Flash Crowds in an open CDN," in IMC '11 Proceedings of the 2011 ACM SIGCOMM Internet measurement Conference, Berlin, Germany, November 2011

[3] A. Dhingra and M. Sachdeva, "Recent Flash Events: A Study," in International Conference on Communication, Computing & Systems, Firozpur, Punjab, India, August 2014

[4] I. Ari, B. Hong, E. Miller, S. Brandt and D. Long, "Managing flash crowds on the Internet," Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on, pp. 246 - 249, 2003.

[5] N. Yoshida, "Dynamic CDN Against Flash Crowds," in Content Delivery Networks, Springer Berlin Heidelberg, 2008, pp. 275-296.

[6] S. Bhatia, G. Mohay, A. Tickle and E. Ahmed, "Parametric Differences between a Real-world Distributed Denial-of-Service Attack and a Flash Event," in Availability, Reliability and Security (ARES), 2011 Sixth International Conference on, Vienna, Austria, August 2011

[7] J. Jaeyeon, B. Krishnamurthy and M. Rabinovich, "Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites," in 11th

International Conference on World Wide Web, Hawaii, USA, May 2002.

[8] Isaac and Elizabeth, "Defending DDoS Attack using Stochastic Model based Puzzle Controller," IJCSNS, vol. 13, no. 4, pp. 100-105, 2003.

[9] A. Sardana, K. Kumar and R. Joshi, "Detection and Honeypot Based Redirection to Counter DDoS Attacks in ISP Domain," in Information Assurance and Security, 2007. IAS 2007. Third International Symposium on, Manchester, England, August 2007

[10] Selenium, (Website: https://www.selenium.dev/) accessed at: March 10, 2023

[11] Selenium Frequently Asked Questions, (Website: http://wiki.openqa.org/display/SEL/FAQ) accessed at: March 12, 2023

[12] Appium (Website: http://appium.io/) accessed at: May 28, 2023

[13] Cypress (Website: https://www.cypress.io/) accessed at: June 25, 2023

[14] Puppeteer (Website: https://pptr.dev/) accessed at: June 25, 2023

[15] TestCafe (Website: https://testcafe.io/) accessed at: April 24, 2023

[16] WebDriverIO (Website: https://webdriver.io/) accessed at: April 24, 2023

[17] Wireshark (Website: https://www.wireshark.org/docs/wsug_html_chunked/) accessed at: March 10, 2023

[18] Tcpdump (Website: https://www.tcpdump.org/) accessed at: April 23, 2023

[19] Locust (Website: https://locust.io/) accessed at: March 2, 2023

[20] Locust Source Code (Website: https://github.com/locustio/locust) accessed at: May 2, 2023

[21] Apache JMeter (Website: https://jmeter.apache.org/) accessed at: May 7, 2023

[22] Gatling (Website: https://gatling.io/) accessed at: June 11, 2023

[23] Apache Bench (Website: https://httpd.apache.org/docs/2.4/programs/ab.html) accessed at: June 10, 2023

[24] Top 15 Automation Testing Tools (Website: https://katalon.com/resources-center/blog/automation-testing-tools) accessed at: June 15, 2023

[25] Open-source load testing tool review 2020 (Website: https://k6.io/blog/comparing-best-open-source-load-testing-tools/) accessed at: June 15, 2023