به نام خدا

امنیت داده و شبکه

نیم‌سال دوم ۱۴۰۴-۱۴۰۵

دانشکده‌ی مهندسی کامپیوتر

دانشگاه صنعتی شریف

موضوع	**آسیب‌پذیری برنامه‌های کاربردی وب**
موعد تحویل	**ساعت ۲۳:۵۹ دوشنبه ۱۱ خرداد ۱۴۰۵**
طراحی تمرین توسط	**محمد حدادیان**
اقتباس بخشی از	`CS155, Spring 2020, Stanford University`

مقدمه

در این تمرین شما در چهار بخش، اقدام به شناسایی آسیب‌پذیری‌های برنامه‌های کاربردی تحت وب می‌کنید. بخش اول در سیستم خودتان پیاده‌سازی و اجرا می‌شود اما بخش‌های بعد را باید در یک محیط واقعی انجام دهید.

۱ بخش اول

در این بخش شما با یک برنامه وب به نام Bitbar روبرو هستید که در زبان Node.js پیاده‌سازی شده است و به کاربران این امکان را می‌دهد که Bitbar ها، یک ارز دیجیتال بسیار ایمن، را مدیریت کنند. هر کاربر هنگام ثبت‌نام در سایت، ۱۰۰ بیت‌بار دریافت می‌کند. آن‌ها می‌توانند بیت‌بارها را از طریق رابط وب به کاربران دیگر انتقال دهند و همچنین پروفایل کاربران دیگر را ایجاد و مشاهده کنند.

شما کد منبع برنامه Bitbar را در اختیار دارید. حمله‌کنندگان واقعی به طور عمومی چنین دسترسی‌ای ندارند، اما برای فهم بهتر آسیب‌پذیری‌ها در این بخش تمرین دسترسی به کد منبع برای شما فراهم شده است. Bitbar توسط یک مجموعه از بسته‌های Node اداره می‌شود که شامل چارچوب برنامه‌نویسی وب Express.js، پایگاه داده SQLite و EJS برای قالب‌بندی HTML است. لیست منابع در بخش بعدی شامل پیوندهایی برای اطلاعات بیشتر در مورد این بسته‌ها و همچنین اطلاعات دیگری است که می‌توانید به عنوان مرجع استفاده کنید.

شما برنامه Bitbar را در یک کانتینر Docker که در اختیارتان قرار گرفته‌است اجرا می‌کنید. هنگامی که سرور در حال اجراست، با رفتن به آدرس `http://localhost:3000` به سایت دسترسی دارید.

مرورگر

ما در این تمرین از نسخه ۹۳ فایرفاکس برای امتیازدهی استفاده خواهیم کرد و توصیه می‌شود که حملات خود را در فایرفاکس تست کنید. مرورگر کروم چون بعضی حفاظت‌های اضافه برای XSS دارد، ممکن است برخی از حملات را غیرممکن کند.

دستورالعمل‌های نصب دقیق:

سرور وب شما در یک کانتینر Docker اجرا خواهد شد. دستورالعمل‌های زیر شما را در نصب Docker و کانتینر کمک خواهند کرد.

۱. اگر از ویندوز یا مک استفاده می‌کنید، نرم‌افزار Desktop Docker را از سایت رسمی داکر دانلود و نصب کنید. اگر از لینوکس استفاده می‌کنید، Engine Docker را از طریق مدیر بسته‌ی توزیع خود نصب کنید (برای اوبونتو: sudo apt install docker.io docker-compose-v2).

۲. بعد از دانلود فایل‌های تمرین به پوشه‌ی `hw2-1` رفته و دستور `bash build_image.sh` را بزنید. این دستور داکر ایمیج مربوط به این سوال را می‌سازد.

۳. برای شروع به کار برنامه، دستور `bash start_server.sh` را بزنید. پس از این کار، با رفتن به آدرس http://localhost:3000 به وب‌اپلیکیشن دسترسی خواهید داشت.

می‌توانید سرور را با فشار دادن Ctrl+C در ترمینال ببندید. **هر بار که سرور را خاموش می‌کنید، سرور به طور کامل بازنشانی خواهد شد.**

برای راه‌اندازی مجدد سرور با یک پایگاه داده پاک، فقط دستور `bash start_server.sh` را دوباره اجرا کنید.

نکات Docker

شما نیازی به آشنایی با Docker برای تکمیل این تکلیف ندارید. با این حال، چند نکته ممکن است مفید باشد:

- `docker ps -a` تمام کانتینرهای شما را لیست می‌کند.
- `docker images` ایمیج‌های شما را لیست می‌کند.
- `docker system prune -a` ایمیج‌ها و کانتینرهای بی‌استفاده از ماشین شما را حذف می‌کند.
- اسکریپت‌های `build_image` و `start_server` به طور ساده دستورات Docker یک خطی برای ساخت یک ایمیج Docker و ایجاد یک کانتینر موقت از آن ایمیج هستند.
- تنها فایلی که از ماشین محلی شما به کانتینر Docker در حال اجرا منتقل می‌شود `code/router.js` است. پس اگر شروع به اصلاح فایل‌های دیگر کنید و تغییرات نمایان نشود، نگران نباشید. شاید برای اعمال تغییرات لازم باشد کانتینر خود را پس از اصلاح `code/router.js` دوباره راه‌اندازی کنید. اگر تصمیم به اصلاح فایل‌های دیگر کنید، باید ایمیج Docker را بازسازی کنید تا تغییرات شما در ایمیج جای گیرند.
- مستندات رسمی Docker در `https://docs.docker.com/` موجود است.

در ادامه‌ی این بخش شما باید سه سری حمله علیه برنامه Bitbar توسعه دهید. برای هر حمله، در ادامه شرح می‌دهیم که دقیقاً باید چه‌کار کنید. برای دریافت امتیاز، باید نتیجه‌ای که در هر حمله توصیف شده را به دست آورید. تمامی حملات شما باید فرض کنند که سایت به آدرس `http://localhost:3000` قابل دسترس است.
شما نمی‌توانید از کتابخانه‌های خارجی استفاده کنید و همچنین نمی‌توانید به برنامه وب خود دست بزنید. به خصوص نمی‌توانید از jQuery استفاده کنید. شما می‌توانید از منابع آنلاین استفاده کنید، اما لطفاً آن‌ها را در `README.txt` ارسالی‌تان ذکر کنید.

۱.۱ اکسپلویت آلفا: دزدیدن کوکی

در حمله اول، هدف شما دزدیدن کوکی جلسه (session) کاربر وارد شده به سیستم و ارسال آن به یک URL تحت کنترل مهاجم است. شما باید یک URL ایجاد کنید که با http://localhost:3000/profile?username= شروع شود و زمان بازدید از این آدرس، کوکی دزدیده شده را به
http://localhost:3000/steal_cookie?cookie=stolen_cookie_here
ارسال کند. زمانی که حمله موفق باشد، سرور کوکی دزدیده شده را در خروجی ترمینال ثبت خواهد کرد.
توجه! حمله باید به شکل نامحسوس برای کاربر باشد. این به این معناست که نباید تغییراتی در ظاهر سایت ایجاد شود و متن اضافی دیگری قابل مشاهده نباشد. به جز نوار مکان مرورگر (که ممکن است متفاوت باشد)، کاربر باید صفحه‌ای را ببیند که هنگام بازدید از پروفایل خود به نظر معمولی برسد. اجتناب از متن آبی هشدار دهنده که یک کاربر یافت نشد راه مهمی از حمله است. اگر تعداد بیت‌بارها یا محتوای پروفایل صحیح نباشد (تا جایی که "معمولی" به‌نظر بیاید)، مشکل ندارد.
تحویل و امتیازدهی. شما باید یک فایل به نام `a.txt` را تحویل دهید که فقط آدرس URL مخرب شما را دربردارد. مصحح به عنوان `user1` وارد Bitbar شده و در تب پروفایل قرار دارد. از این مکان، مصحح آدرس URL شما را از نوار آدرس کپی می‌کند و به آنجا می‌رود. مصحح باید با انتخاب "Containers/Apps" در Docker، سپس "bitbar-container" و کلیک بر روی "LOGS"، کوکی دزدیده شده را ببیند.

۲.۱ اکسپلویت براوو: جعل درخواست راه دور (CSRF)

در حمله دوم، شما باید یک حمله جعل درخواست از راه دور (CSRF) بسازید که از یک کاربر دیگر ۱۰ بیت‌بار به حساب مهاجم انتقال دهد.

حمله ارسالی شما یک صفحه HTML است که ۱۰ بیت‌بار را از کاربر لاگین‌شده به حساب کاربر attacker منتقل می‌کند. به محض اتمام انتقال، حمله شما باید بلافاصله کاربر را به
http://sharif.edu/~kharrazi/courses/40441-011/
هدایت کند. این کار باید به گونه‌ای سریع انجام شود که کاربران عادی آن را متوجه نشوند.
تحویل و امتیازدهی. شما باید یک فایل HTML مستقل به نام b.html را تحویل دهید که حاوی حمله شما باشد. با باز کردن این صفحه توسط کاربری که از قبل در سامانه لاگین است، باید ۱) ده بیت‌بار از حساب او به حساب مهاجم منتقل شود، ۲) سایت حمله به سایت CE441 به‌سرعت منتقل شود و ۳) هیچ‌گاه آدرس localhost:3000 مشخص نشود.

۳.۱ اکسپلویت گاما: حمله زمانی

حمله زمانی یک نوع حمله جانبی است که مهاجم با تجزیه و تحلیل زمانی که یک سیستم برای انجام یک عمل صرف می‌کند، سعی در استخراج داده می‌کند. به عنوان مثال، یک سرور وب ممکن است زمان بیشتری برای پاسخ به یک درخواست ورود که حاوی یک گذرواژه صحیح است، نسبت به یک درخواست حاوی گذرواژه نامعتبر صرف کند. حتی اگر قوانین سایت یک حمله‌کننده را از مشاهده مستقیم پاسخ HTML به یک درخواست ورود جلوگیری کند، مقدار زمانی که سرور برای پاسخ دادن به درخواست زمان می‌برد ممکن است اطلاعاتی در مورد اینکه گذرواژه ارائه شده صحیح یا نادرست بوده باشد را لو دهد.
در این قسمت شما باید یک حمله انجام دهید که گذرواژه کاربر دیگر را با بهره‌گیری از چنین کانال جانبی زمانی استخراج کنید. شما به ویژه با تحلیل مقدار زمانی که صفحه ورود به Bitbar به یک گذرواژه صحیح در مقابل گذرواژه‌های نادرست پاسخ می‌دهد، گذرواژه قربانی را پیدا خواهید کرد.
شما باید یک نام کاربری مخرب بسازید که شامل یک اسکریپت باشد که گذرواژه userx را با تست گذرواژه‌ها در یک لغتنامه ارائه شده و اندازه زمان پاسخ سرور برای هر گذرواژه‌ای که ارائه می‌شود اندازه‌گیری کند. اسکریپت شما باید زمان پاسخ سرور را برای تمام گذرواژه‌های موجود در لیست ارائه شده تحلیل کرده، گذرواژه صحیح را تشخیص دهد و آن را به

```
http://localhost:3000/steal_password?password=[password]&timeElapsed=[time_elapsed]
```

ارسال کند.
می‌توانید از کد proj2/code/gamma_starter.html به عنوان نقطه شروع برای حمله خود استفاده کنید. این قطعه کد شامل لغتنامه‌ای از گذرواژه‌ها برای تست است.
تحویل و امتیازدهی. شما باید یک فایل به نام g.txt حاوی اسکریپت نام کاربری مخرب را تحویل دهید. برای امتیازدهی حمله، ما به عنوان attacker وارد شده و به صفحه انتقال می‌رویم، اسکریپت نام کاربری مخرب را که در حلقه اجرا مشخص کرده‌اید را در فیلد username وارد می‌کنیم و ۱۰ بیت‌بار به آن منتقل می‌کنیم.
اگر مصحح بعد از انجام حمله به عنوان userx وارد شده باشد، مشکلی نیست، و حمله ممکن است چند ثانیه به طول بیانجامد. مصحح در حین اجرای حمله کلیکی انجام نخواهد داد و یا از صفحه انتقال خارج نخواهد شد. تغییرات قابل مشاهده در وب‌سایت نباید وجود داشته باشد و پیام خطای آبی در صفحه انتقال باید "کاربر یافت نشد" باشد.
راهنمایی: مطمئن شوید که برای این حمله از علامت backtick به جای quote استفاده می‌کنید. کانال‌های جانبی زمانی ممکن است حساس باشند.

۴.۱ تحویل بخش اول

- فایل تحویلی برای هر حمله را داخل یک دایرکتوری به نام hw2-1 قرار دهید.
- یک README.txt بسازید تا منابع آنلاین مورد استفاده خود را نقل قول کرده و یادداشت‌های خاصی را به مصحح بدهید و آن را هم در دایرکتوری hw2-1 قرار دهید.
- برای هر بخش، راه‌حل خود را گام به گام توضیح دهید و تصاویر مرتبط را در گزارش خود قرار دهید و آن را به عنوان یک فایل به نام report.pdf ارسال کنید.
- زمانی که فایل تحویلی شما را دانلود می‌کنیم، انتظار داریم که پنج فایل زیر داخل 'hw2-1' باشد:

- `README.txt`
- `report.pdf`
- `a.txt`
- `b.html`
- `g.txt`

۲ بخش دوم

در این بخش، با یک حمله‌ی ترکیبی XSS و CSRF به سمت یک سایت که برروی سیستم خود میزبانی نمی‌شود روبرو هستیم. مشابه تمرین گذشته، در این بخش هم به شما تنظیمات داکر مربوط به سوال داده شده است که با رفتن به پوشه‌ی `hw2-2` و زدن دستور `docker compose up` می‌توانید سایت مدنظر را به صورت لوکال بالا بیاورید. هرچند گرفتن نمره‌ی این تمرین منوط به گرفتن پرچم موجود در سایت اصلی به آدرس http://185.205.203.123:1202/ است. قالب پرچم این سوال هم به شکل `CE441{xxx}` است.

در فایل‌های این سوال، می‌توانید کدهای مربوط به سایت را ببینید هرچند در واقعیت شما به کد منبع اهداف خود دسترسی ندارید.

کار خود در این سوال را با بررسی کد منبع سایت و دیدن اسکریپت‌های JS موجود در آن شروع کنید. صفحه‌ی اصلی این سایت هدف به شما اجازه‌ی ارسال یک آدرس به عنوان کتاب اهدایی به سایت را می‌دهد. شما در این بخش می‌توانید URL موردنظر خود را ارسال کنید.

همچنین آمار مربوط به تعداد کتاب‌های اهدایی به کتاب‌خانه هم از طریق ایجاد یک ارتباط وب‌سوکت به‌روز می‌شود.

بعد از ارسال این URL ، ربات ادمین سایت به آن آدرس درخواست می‌زند. پس برای حل این سوال باید به‌گونه‌ای آدرس موردنظر خود را بسازید که بعد از ارجاع ربات ادمین به این آدرس، به‌نوعی به پرچم دست پیدا کنید.

راهنمایی: تفاوت این بخش با بخش اول، این است که چون ربات ادمین روی سیستم شما در حال اجرا نیست، باید محتویات اکسپلویت خود را در یک آدرس دسترس‌پذیر از سمت ربات ادمین قرار دهید. از این رو می‌توانید از ابزارهایی که کدها و صفحات شما را به صورت موقت میزبانی کرده و لاگ درخواست‌های ارسالی به سمت آن‌ها را ذخیره می‌کنند استفاده کنید.

راهنمایی: برای راهنمایی بیشتر، حملات جعل درخواست از راه دور در وب‌سوکت‌ها را بررسی کنید.

راهنمایی: ممکن است در ارسال دستورات وب‌سوکت به خطای http/s بخورید. برای رفع عیب از خطاها همواره کنسول مرورگر را بررسی کنید.

۳ بخش سوم

در سومین بخش این تمرین، با یک سایت انجمن (forum) روبرو هستیم که در آن کاربران می‌توانند پست بسازند و پست‌های دیگران را گزارش کنند. ربات ادمین سایت به صورت خودکار پست‌های گزارش‌شده را بررسی می‌کند.

برای دسترسی به این وبسایت به آدرس http://185.205.203.123:1203 بروید و همچنین می‌توانید با استفاده از فایل‌های داکر موجود در پوشه `hw2-3`، سایت را به صورت محلی بالا بیاورید.

پرچم در متغیرهای محیطی (`environment variables`) برنامه نگهداری می‌شود. هدف شما استخراج پرچم از طریق بهره‌برداری از ربات ادمین است.

نکته: سایت دارای `Content Security Policy` است که از اجرای اسکریپت‌های `inline` و منابع خارجی جلوگیری می‌کند.

نکته: دقت کنید که پست‌های ساخته‌شده توسط کاربران معمولی فقط برای همان کاربر قابل مشاهده هستند، اما ربات ادمین به تمام پست‌ها دسترسی دارد.

قالب پرچم این بخش به شکل `CE441{xxx}` است.

راهنمایی: برای دریافت پرچم استخراج‌شده، نیاز دارید محتوای آن را به یک سرویس گیرنده‌ی درخواست ارسال کنید. سرویس webhook روی سرور تمرین در دسترس است. با مراجعه به آدرس http://185.205.203.123:1205/ یک آدرس اختصاصی برای خود بسازید و پرچم را به آن ارسال کنید.

۴ بخش چهارم

در این بخش با یک سرویس رمزنگاری به زبان Node.js روبرو هستید که کاربران می‌توانند از آن برای مدیریت عملیات رمزنگاری خود استفاده کنند. این سرویس به کاربران امکان می‌دهد که دستورات (jobs) رمزنگاری ایجاد کنند و آن‌ها را ذخیره و بارگذاری کنند.

در این بخش هم به شما تنظیمات داکر و کد منبع مربوط به سوال داده شده است که با رفتن به پوشه‌ی `hw2-4` و زدن دستور `docker compose up` می‌توانید سرویس مدنظر را به صورت لوکال بالا بیاورید.

سرویس شامل قابلیت‌های زیر است:

- تعریف ورودی برای پردازش
- افزودن دستورات رمزنگاری (مانند Base64 Encode/Decode و Hex Encode/Decode)
- اجرای دستورات و دریافت خروجی
- ذخیره و بارگذاری دستورالعمل‌ها (recipes) با رمز عبور

یک دستورالعمل از پیش ساخته‌شده با شناسه `a94e9c000a02f03a` در سرور موجود است که حاوی پرچم است. این دستورالعمل با رمز عبور محافظت شده و شما نمی‌توانید آن را مستقیماً باز کنید. هدف شما استخراج پرچم از این دستورالعمل محافظت‌شده است.

برای دسترسی به این سرویس، از دستور زیر استفاده کنید:

nc 185.205.203.123 1204

این سرویس از طریق TCP به درخواست‌ها پاسخ می‌دهد. برای نوشتن اکسپلویت خود، می‌توانید از کتابخانه‌های pwntools یا socket در پایتون استفاده کنید.

کار حل خود را با بررسی کد منبع سرویس آغاز کنید. به ویژه توجه کنید که چگونه دستورات اضافه می‌شوند، چگونه اجرا می‌شوند و چه شرایطی برای دریافت خروجی وجود دارد.

قالب پرچم این بخش به شکل `CE441{xxx}` است.

نکته: این چالش نیازمند درک عمیق از مکانیزم‌های async/await و Promise در جاوااسکریپت است.

راهنمایی: بررسی کنید که نام‌گذاری دستورات چه تاثیری بر رفتار سیستم می‌گذارد.

تحویل بخش‌های دو تا چهار

برای این بخش شما باید کد اکسپلویت خود را به همراه یک ویدیو ۵ دقیقه‌ای (برای هر بخش) که قدم به قدم تا رسیدن به پرچم راه‌حل را توضیح می‌دهد در پوشه‌ی `hw2-2,hw2-3,hw2-4` از فایل‌های تحویلی خود قرار دهید. ویدیو خود را در هر سرویس میزبانی فایل بارگذاری کرده و آدرس آن به همراه SHA256 فایل ویدیو را در فایل `url.txt` قرار دهید.

در نهایت تحویل‌دادنی‌های شما باید یک فایل zip با نام‌گذاری `ce441-hw2-<STUDENT_ID>` باشد که به ترتیب شامل پوشه‌ها و فایل‌های زیر است (در اینجا به عنوان نمونه زبان پایتون قرار گرفته است اما شما می‌توانید با هر زبانی (مثلا HTML) اکسپلویت خود را بنویسید):

```
ce441-hw2-99101234:
   hw2-1:
      README.txt
      a.txt
      b.html
```

```
      g.txt
      report.pdf
    hw2-2:
      exploit.py
      url.txt
    hw2-3:
      exploit.py
      url.txt
    hw2-4:
      exploit.py
      url.txt
```