

به نام خدا



امنیت داده و شبکه

نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دانشکده‌ی مهندسی کامپیوتر

دانشگاه صنعتی شریف

موضوع آسیب‌پذیری برنامه‌های کاربردی وب

موعده تحویل ساعت ۲۳:۵۹ شنبه ۱ اردیبهشت ۱۴۰۳

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 گرفته است 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، دستورالعمل‌های موجود در [لینک ۱](https://docs.docker.com/engine) و [لینک ۲](https://docs.docker.com/engine) را دنبال کنید تا Desktop Docker را نصب کنید.^۱ اگر از لینوکس استفاده می‌کنید، توزیع لینوکس خود را در <https://docs.docker.com/engine> انتخاب کنید و دستورالعمل‌ها برای نصب Docker Engine را دنبال کنید.

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

^۱ برای راهنمای گام به گام برای ویندوز، به <https://www.youtube.com/watch?v=IwLQ92XRiGg> مراجعه 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: <https://docs.docker.com/>

در ادامه‌ی این بخش شما باید سه سری حمله علیه برنامه Bitbar توسعه دهید. برای هر حمله، در ادامه شرح می‌دهیم که دقیقاً باید چه کار 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 دسترسی است. شما نمی‌توانید از کتابخانه‌های خارجی استفاده کنید و همچنین نمی‌توانید به برنامه وب خود دست بزنید. به خصوص نمی‌توانید از Query استفاده کنید. شما می‌توانید از منابع آنلاین استفاده کنید، اما لطفاً آن‌ها را در `README.txt` ارسالی تان 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" به نظر 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 بیت بار را از کاربر لاگین شده به حساب کاربر attacker منتقل می کند. به محض اتمام انتقال، حمله شما باید بلافاصله کاربر را به

<http://sharif.edu/~kharrazi/courses/40441-011/>

هدایت کند. این کار باید به گونه ای 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 بیت بار از حساب او به حساب مهاجم منتقل شود، (۲) سایت حمله به سایت CE441 به سرعت منتقل شود و (۳) هیچ گاه 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 می کند، سعی در استخراج داده می کند. 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 حمله کننده را از مشاهده مستقیم پاسخ HTML به یک درخواست ورود جلوگیری کند، مقدار زمانی که سرور برای پاسخ دادن به درخواست زمان می برد 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 بهره گیری 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 گذرواژه های نادرست پاسخ می دهد، گذرواژه قربانی را پیدا خواهید کرد.

شما باید یک نام کاربری مخرب بسازید که شامل یک اسکریپت باشد که گذرواژه userx را با تست گذرواژه ها در یک لغتنامه ارائه شده و اندازه زمان پاسخ سرور برای هر گذرواژه ای که ارائه می شود اندازه گیری کند. اسکریپت شما باید زمان پاسخ سرور را برای تمام گذرواژه های 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.

می توانید از کد proj2/code/gamma_starter.html به عنوان نقطه شروع برای حمله خود استفاده کنید. این قطعه کد شامل لغتنامه ای از گذرواژه ها 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 می رویم، اسکریپت نام کاربری مخرب را که در حلقه اجرا مشخص کرده اید را در فیلد username وارد می کنیم و ۱۰ بیت بار به آن منتقل می کنیم.

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 کلیک می کند 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 می کنید. کانال های 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 یادداشت های خاصی را به مصحح بدهید و آن را هم در دایرکتوری hw2-1 قرار دهید.
- برای هر بخش، راه حل 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 می کنیم، انتظار داریم که پنج فایل زیر داخل 'hw2-1' باشد:

README.txt -

report.pdf -

a.txt -

b.html -

g.txt -

۲ بخش دوم

در بخش دوم، با یک آسیب‌پذیری تزریق پایگاه داده‌ی ساده روبرو هستیم. با رفتن به آدرس <http://91.58:8008.157.107> یک برنامه به زبان php که به یک دیتابیس متصل است را می‌بینید. فایل‌های موردنیاز برای استقرار این محیط روی سیستم شما هم در اختیارتان قرار گرفته است. شما باید پرچم این سوال که در دیتابیس ذخیره شده است را پیدا کنید. **راهنمایی:** شما باید یوزرنیم و پسورد مخرب را به عنوان پارامترهای یک درخواست GET به سمت هدف بفرستید.

۳ بخش سوم

در این بخش، با یک حمله‌ی ترکیبی XSS و CSRF به سمت یک سایت که بر روی 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 hw2-3 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://91.58:8080.157.107> 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 کتاب‌هدایی 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.

۴ بخش چهارم

در آخرین بخش این تمرین، کد مربوط به پیاده‌سازی منطق login از یک وب‌سایت، در زبان nodeJS پیاده‌سازی شده است. در این پیاده‌سازی، یک (دو!) آسیب‌پذیری ساده اما رایج در این زبان وجود دارد و شما باید با کشف و (سو)استفاده از این آسیب‌پذیری، به پرچم ۱ دست یابید.

هدف اول شما در آدرس `/login` قرار دارد که پرچم ۱ را به شما می‌دهد. اما در ادامه‌ی کد برنامه، بخش دومی که تا حدی سعی کرده تا آسیب‌پذیری بخش اول را از بین ببرد هم قرار دارد. چنانچه بتوانید آسیب‌پذیری هوشمندانه‌ی این بخش را پیدا کنید، همزمان به دو پرچم دست پیدا می‌کنید!

دقت کنید که تنها دسترسی به پرچم اول برای گرفتن نمره‌ی این بخش کافیست. همچنین در این بخش باید حتما پرچم موجود در سمت سرور را به دست آورید و در صورت گرفتن پرچم صرفاً در سیستم خود، امتیاز کامل این بخش را نخواهید گرفت. هدف این بخش در آدرس <http://91.58:8000.157.107> قرار دارد.

تحويل بخش‌های دو تا چهار

برای این بخش شما باید کد اکسپلویت خود را به همراه یک ویدیو ۵ دقیقه‌ای که قدم به قدم تا رسیدن به پرچم راه حل را توضیح می‌دهد در پوشه‌ی hw2-2, hw2-3, hw2-4 از فایل‌های تحويلی خود قرار دهید. ویدیو خود را در سرویس‌های میزبانی فایل بارگذاری کرده و آدرس آن به همراه هش فیلم را در فایل url.txt قرار دهید. در نهایت تحويل‌دانی‌های شما باید یک فایل zip با نام‌گذاری <STUDENT_ID>-hw2-ce441 باشد که به ترتیب شامل پوشه‌ها و فایل‌های زیر است:

```
1 ce441-hw2-99101234:
2   hw2-1:
3     README.txt
4     a.txt
5     b.html
6     g.txt
7     report.pdf
8   hw2-2:
9     exploit.py
10    url.txt
11  hw2-3:
12    exploit.py
13    url.txt
14  hw2-4:
15    exploit.py
16    url.txt
```