به نام خدا

امنیت داده و شبکه

نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دانشکده‌ی مهندسی کامپیوتر

دانشگاه صنعتی شریف

موضوع **بهره‌برداری از آسیب‌پذیری برنامه‌ها**

موعد تحویل **ساعت ۲۳:۵۹ چهارشنبه ۲۳ اسفند ۱۴۰۲**

طراحی تمرین توسط **محمد حدادیان، امیر آقاپور**

با سپاس از **امیرمهدی کوششی**

مقدمه

هدف از این تمرین تجربه بیشتر در شناسایی و بهره‌برداری از آسیب‌پذیری‌های برنامه‌ها است. این تمرین از پنج بخش تشکیل شده‌است. برای چهار بخش اول شما باید ابتدا آسیب‌پذیری برنامه‌ی داده‌شده را پیدا کرده و سپس با نوشتن یک اسکریپت (به هر زبان دلخواه) از آن آسیب‌پذیری سوءاستفاده کرده، به shell دست یابید. در بخش‌های دوم و سوم و چهارم تمرین بعد از دست‌یابی به شِل، به پرچم موجود روی ماشین هدف دسترسی پیدا کنید. پرچم هر بخش به صورت یک رشته به فرمت `CE441{xxxx}` می‌باشد.

۱ بخش اول

۱.۱ راه‌اندازی محیط

برای بخش اول این تمرین، یک ماشین مجازی[۱] در اختیار شما قرار گرفته است. در بهره‌برداری از آسیب‌پذیری‌ها، همه چیز از نسخه‌ی کامپایلر تا مکانیزم‌های امنیتی سیستم‌عامل دخیل خواهند بود. با داشتن این ماشین مجازی، در اجرای اکسپلویت‌های خود یکپارچه خواهید بود.

این ماشین مجازی نسخه‌ی Ubuntu Linux 16.04 LTS با ASLR خاموش است. این ماشین یک کاربر با نام user و رمز ce441 دارد. شما می‌توانید به‌صورت موقتی با دستور sudo به کاربر root تبدیل شوید اما اکسپلویت‌های شما با دسترسی کاربر user اجرا می‌شوند و باید در آن به شل `/bin/sh` با دسترسی‌های root دست پیدا کنید.

پس از اجرای این ماشین، یک سرویس OpenSSH روی آن اجرا می‌شود که می‌توانید از سیستم خود به این ماشین ssh بزنید یا فایل منتقل کنید: `ssh user@192.168.56.144`

۲.۱ اهداف

در پوشه‌ی `targets/` از ماشین مجازی، کد منبع چند هدف آسیب‌پذیر همراه با `Makefile` آن‌ها برای کامپایل و اجرا قرار داده شده است که شما در بخش اول این تمرین فقط اهداف ۱ و ۲ را باید هدف قرار دهید. برای کامپایل این اهداف دستورات زیر را اجرا کنید:

```
cd targets
make
sudo make install
```

با این دستورات، فایل‌های اجرایی اهداف در آدرس `/tmp` قرار می‌گیرند. دقت کنید که اکسپلویت شما باید این اهداف را دقیقا در پوشه‌ی `/tmp/target1` اجرا و بهره‌برداری کند.

برای حل این بخش تمرین شما باید دنبال buffer overflow در آرایه‌های برنامه‌های هدف باشید؛ هرچند این سرریز بافر ممکن است به صورت کامل در اختیار شما نباشد

۳.۱ ساختار کد اکسپلویت

پوشه‌ی `sploits/` شامل ساختار موردنیاز برای نوشتن اکسپلویت شما است. همچنین هدرفایل `shellcode.h` شامل شل‌کد موردنیاز برای حل این بخش از تمرین است که شما باید اکسپلویت‌های خود برای این بخش تمرین را با استفاده از این ساختارها بنویسید.

[۱] http://partov.ce.sharif.edu/assets/40441-991/CE441_vm.ova.xz

۲ بخش دوم

۱.۲ راه‌اندازی محیط

در بخش‌های دوم و سوم و چهارم این تمرین به منظور فراهم کردن یک محیط یکسان برای exploit کردن آسیب پذیری‌ها، داکرفایلی در اختیار شما قرار خواهد گرفت تا بتوانید محیط مسئله را روی رایانه‌ی شخصی خود داشته و تست کنید. این داکرفایل فقط برای تمرین شماست و تنها درصورتی که روی سرورهای مقصد به پرچم دست یابید نمره‌ی بخش‌های مربوطه را کسب می‌کنید. همچنین برای اینکه داکرفایل به‌خوبی روی سیستم شما اجرا شود، مطمئن شوید که معماری سیستم شما x۸۶ باشد.

برنامه‌ی آسیب‌پذیر در داکر ایمیج‌هایی که در اختیار شما قرار داده شده با پورت مشخص شده اجرا می‌شوند و شما باید با بهره‌برداری از آن‌ها به این ماشین‌ها دسترسی پیدا کرده و پرچم را بدست آورید.

به جهت راه‌اندازی محیط بر روی رایانه شخصی کافیست پس از نصب ابزارهای docker به پوشه‌ی تمرین رفته و آن را build و نهایتا run کنید. با این دستور محیط تمرین روی سیستم شما بالا آمده و با دستور `nc localhost [port]` می‌توانید به آن‌ها متصل شوید.

همچنین در صورت نیاز می‌توانید با کمک دستور `docker exec` از محیط داکر برای بررسی سوالات و بهره‌برداری از آسیب‌پذیری‌ها استفاده کنید.

۲.۲ ابزارها

Pwntools یک کتابخانه‌ی پایتون است که exploit نویسی را بسیار ساده می‌کند. در این تمرین از این ابزار برای یافتن gadget ها به صورت خودکار، ساختن ROP chain و موارد مشابه می‌توانید استفاده کنید. همچنین برای پیدا کردن return address ها می‌توانید از ابزارهایی مانند gdb و objdump بهره ببرید.

برای آشنایی بیشتر با pwntools می‌توانید به این سایت مراجعه کنید. همچنین برای خواندن داکیومنت‌های این کتابخانه می‌توانید به این سایت مراجعه کنید. شما نیز می‌توانید فیلم‌های متعددی در یوتوب در رابطه با حل سوال با pwntools پیدا کنید. برای آشنایی با gdb می‌توانید به این سایت مراجعه کنید. یک اکستنشن خوب و قوی نیز برای کار حرفه‌ای با gdb به نام gef موجود است که در صورت علاقه نیز می‌توانید با آن به حل سوالات بپردازید یا راجع‌به آن مطالعه کنید. شما راجع‌به این اکستنشن نیز می‌توانید در این سایت مطالعه کنید.

۳.۲ هدف

در هدف این بخش یک فایل باینری به شما داده شده‌است. به ویژگی‌های امنیتی این فایل توجه کنید. یک راه‌کار این موضوع استفاده از دستور `checksec` است.

برای اتصال به ماشین میزبان این بخش تمرین، از دستور زیر استفاده کنید:

```
nc 65.109.185.193 5000
```

پس از اتصال به سرور پیامی برای شما چاپ می‌شود و شما امکان تعامل با برنامه را خواهید داشت. شما باید با بهره‌برداری از آسیب‌پذیری برنامه‌ی داده شده، به شل دسترسی پیدا کنید و پرچم موجود در ماشین را چاپ کنید.

برای شروع اکسپلویت، چون کد برنامه در اختیار شما نیست بهتر است آن را در ابزارهای دیباگ یا دیکامپایل بررسی کنید. دقت کنید که در این بخش تمرین شل‌کد در اختیار شما نیست و باید با استفاده از توابع برنامه به هدف برسید.

۳ بخش سوم

۱.۳ هدف

در هدف مربوط به این بخش هم مانند بخش قبل، یک فایل باینری به شما داده شده‌است. برخی ویژگی‌های امنیتی این فایل ممکن است متفاوت باشد. با بررسی فایل به حل تمرین بپردازید.

برای اتصال به ماشین میزبان این بخش تمرین، از دستور زیر استفاده کنید:

```
nc 65.109.185.193 5001
```

در این بخش تمرین تمام مکانیزم‌های امنیتی روی برنامه‌ی هدف فعال است و حل تمرین را مجددا با بررسی هدف در ابزارهای دیباگ و دیکامپایل شروع کنید. این بار برخلاف اهداف قبلی شما نیاز به به‌دست آوردن قناری خواهید داشت. همچنین دقت کنید که ASLR نیز روشن می‌باشد و آدرس‌های برنامه هر سری که اجرا شود، عوض می‌شوند.

۴ بخش چهارم

۱.۴ هدف

در هدف مربوط به این بخش هم مانند بخش قبل، یک فایل باینری به شما داده شده‌است. برخی ویژگی‌های امنیتی این فایل ممکن است متفاوت باشد. با بررسی فایل به حل تمرین بپردازید.

برای اتصال به ماشین میزبان این بخش تمرین، از دستور زیر استفاده کنید:

```
nc 65.109.185.193 5002
```

در این بخش تمرین تمام مکانیزم‌های امنیتی روی برنامه‌ی هدف فعال است و حل تمرین را مجددا با بررسی هدف در ابزارهای دیباگ و دیکامپایل شروع کنید. این بار همانند قسمت قبلی شما نیاز به به‌دست آوردن قناری خواهید داشت و شل‌کد مورد استفاده را باید با استفاده از return to libc بسازید. دقت کنید که نسخه‌ی مورد استفاده در اکسپلویت شما با نسخه‌ی ماشین هدف یکسان باشد (برای این کار می‌توانید از فایل libc در داکرایمیجی که در اختیارتان قرار گرفته استفاده کنید.) همچنین دقت کنید که ASLR نیز روشن می‌باشد و آدرس‌های برنامه هر سری که اجرا شود، عوض می‌شوند. این به این معنی است که آدرس load شدن libc نیز هر سری با اجرای جدید، عوض می‌شود.

۵ بخش پنجم

۱.۵ سوالات تئوری

۱.۱.۵ کنترل دسترسی

۱. می‌خواهیم یک سیستم کنترل دسترسی اجباری با دو سطح محرمانگی `secret` و `unclassified` را با استفاده از سیستم کنترل دسترسی اختیاری لینوکس شبیه‌سازی کنیم. برای این کار دو گروه با نام دو سطح محرمانگی ساخته و هر کاربر یا فایل را در یکی از این دو گروه قرار می‌دهیم. مجوزهای دو فایل زیر که با علامت سؤال مشخص شده را طوری تعیین کنید که اصول دسترسی BLP رعایت شوند. فرض کنید نیازی به کنترل دسترسی کاربر `root` وجود ندارد.

```
permission  owner  group         file name
rw-??-??-   root   secret        secret_file
rw-??-??-   root   unclassified  unclassified_file
```

۲. در سیستم عامل لینوکس، رمز عبور تمام کاربران به صورت رمزشده در فایل `/etc/shadow` نگهداری می‌شود. این فایل تنها توسط کاربر `root` قابل دسترسی و تغییر است. اگر کاربر بخواهد رمز عبور خود را تغییر دهد، باید از برنامه `/usr/bin/passwd` استفاده کنید. مجوز اجرای این برنامه دارای بیت `setuid` است. توضیح دهید این ویژگی چگونه امکان تغییر رمز کاربر را فراهم می‌کند و در صورتی که `passwd` دارای آسیب‌پذیری باشد، باعث چه خطر امنیتی می‌شود.

۳. ابزار SELinux برای تکمیل سیستم کنترل دسترسی لینوکس طراحی شده و امکان استفاده از مدل کنترل دسترسی اجباری و نقش-مبنا را فراهم می‌کند. توضیح دهید چگونه می‌توان از این ابزار از خطر مطرح شده در سؤال قبل پیشگیری کرد.

۶ تحویل‌دادنی‌ها

شما باید برای هر بخش، اسکریپت خود برای بهره‌برداری از آسیب‌پذیری سوال را به همراه یک ویدیو جامع برای هر بخش، که شامل توضیح اسکریپت و نحوه‌ی رسیدن به اطلاعات لازم برای حل و ساخت shell است ارسال کنید. ویدیوهای خود را در سایت‌های میزبانی فایل مانند گوگل‌درایو قرار داده و فقط لینک آن‌ها را همراه با hash ویدیو در cw ارسال کنید. ساختار فایل زیپ ارسالی شما با نام `ce441-hw1-SID` باید به شکل زیر باشد:

```
sploit1-1.c
sploit1-2.c
exploit2.py
exploit3.py
exploit3.py
urls.txt
theory.pdf
```

لازم است در گزارش به طور خلاصه مراحلی که طی کرده‌اید را گام به گام ذکر کنید. همچنین توضیحات مورد نیاز برای نحوه‌ی اجرای اسکریپت‌ها و پیش‌نیازهای آن را نیز به طور کامل در گزارش ذکر کنید. دقت کنید که اسکریپت‌های شما باید به صورت مستقل توسط ما اجرا شده و به پرچم برسد تا نمره‌ی آن بخش را کسب کنید.

در صورت داشتن هرگونه سوال در مورد این تمرین می‌توانید با ایمیل `m.hadadian76@sharif.edu` یا تالارهای گفتگوی درس در cw در ارتباط باشید.