به نام خداوند بخشنده‌ی مهربان

دانشگاه صنعتی شریف | **تمرین سری اول*** | دانشکده‌ی مهندسی کامپیوتر

مدرس: دکتر خرازی | **امنیت شبکه و داده** | بهار ۱۳۹۶

هدف‌ها

- آشنایی با Stack،
- آشنایی با حمله‌ی Buffer Overflow
- آشنایی با حمله‌ی Printf format vulnerability

مقدمه

هدف از این تمرین، به‌دست آوردن تجربه در شناسایی نقاط آسیب‌پذیر برنامه و استفاده از آن نقاط برای حمله است. بنابراین در این تمرین به شما دو فایل binary داده‌شده است که در هر کدام آسیب‌پذیری وجود دارد. شما باید این آسیب‌پذیری‌ها را شناسایی کرده و برای هر کدام یک برنامه(با C/C++ یا python) یا script نوشته و فایل اجرایی آن‌ها به‌همراه source code را به‌عنوان خروجی این تمرین تحویل دهید.

برنامه‌ها

در پوشه‌ی questions دو فایل باینری به‌همراه source code آن‌ها با نام‌های prog_vuln1 و prog_vuln2 وجود دارد(برای استفاده از objdump نیاز است که پوشه‌ی questions را در مسیر /tmp/hw1/questions قرار دهید). در هر یک از این فایل‌ها، آسیب‌پذیری وجود دارد که از آن برای اجرای shellcode استفاده خواهید کرد. این فایل‌های باینری برای سیستم x86(32 bit) و بر روی سیستم‌عامل Linux ساخته‌شده‌اند، بنابراین برای اجرای این باینری‌ها در

*

سیستم x86-64 شاید نیاز به نصب پکیج‌های ۳۲ بیتی باشد که در این لینک‌ها می‌توانید آن‌ها را مشاهده کنید:

http://www.archlinuxuser.com/2013/01/how-to-run-32bit-application-on.html

http://askubuntu.com/questions/454253/how-to-run-32-bit-app-in-ubuntu-64-bit

هدف از این تمرین اجرای Aleph one shellcode است که به‌صورت زیر آن را می‌توانید تعریف کنید(یک نمونه از نحوه‌ی استفاده دز فایل ta_first_class آمده است).

```
static char shellcode[] =
"\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"
"\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
"\x80\xe8\xdc\xff\xff\xff/bin/sh";
```

گرفتن remote shell در این تمرین مجاز نمی‌باشد و در صورت استفاده از هر shellcodeای که منجر به یک عملیات خراب‌کارانه در هنگام تست بشود، نمره‌ی شما صفر خواهد شد.

شما برای استفاده از آسیب‌پذیری‌های درون برنامه فقط حق استفاده از ابزارهای gdb و objdump را دارید و استفاده از ابزارهایی مانند Metasploit مجاز نیست و هیچ نمره‌ای به آن تعلق نمی‌گیرد.

در هر دو سوال این تمرین، ASLR[1] خاموش است. برای خاموش کردن موقت ASLR می‌توانید از دستور

```
$ echo 0 | sudo tee /proc/sys/kernel/randomize_va_space
```

و برای روشن کردن دوباره‌ی آن می‌توانید از دستور

```
$ echo 2 | sudo tee /proc/sys/kernel/randomize_va_space
```

استفاده نمایید. همچنین مکانیزم‌های دفاعی در هر فایل باینری را می‌توانید در شکل ۱ مشاهده کنید.

```
ce40442@ubuntu:~/Desktop/hw1/Questions$ ~/Desktop/proj1/checksec.sh --file prog_vuln1
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      FORTIFY Fortified Fortifiable  FILE
Partial RELRO   No canary found   NX disabled   No PIE          No RPATH   No RUNPATH   No      0               0       prog_vuln1

ce40442@ubuntu:~/Desktop/hw1/Questions$ ~/Desktop/proj1/checksec.sh --file prog_vuln2
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      FORTIFY Fortified Fortifiable  FILE
Partial RELRO   Canary found      NX disabled   No PIE          No RPATH   No RUNPATH   Yes     0               4       prog_vuln2
```

شکل ۱ : مکانیزم‌های دفاعی

تحویل‌دادنی‌ها

برای هر برنامه شما موظفید که یک گزارش بنویسید و در آن آسیب‌پذیری موجود در برنامه را شرح دهید(شما باید تابعی را که آسیب‌پذیری دارد نام ببرید مانند strcpy یا printf) همچنین باید بگویید که آسیب‌پذیری برنامه از چه نوع آسیب‌پذیری است(اگر برنامه چند آسیب‌پذیری دارد باید هر کدام را نام ببرید و بگویید که دامنه‌ی خراب‌کاری هر کدام از آسیب‌پذیری‌ها چیست).

[1] randomization layout space Address

هم‌چنین شما باید دو فایل Script(به‌همراه برنامه‌ی اجرایی و source code برنامه) با نام‌های exploit1.sh و exploit2.sh در پوشه‌ای با نام exploits تحویل دهید که بتوان مسیر فایل *prog_vuln را به عنوان آرگومان به آن‌ها داد. درحقیقت برای تصحیح این تمرین، دستور

```
$ ./exploit1.sh /path/to/prog_vuln1
```

مورد استفاده قرار می‌گیرد. بنابراین برای گرفتن نمره‌ی این تمرین رعایت این نکات الزامی است.
شما می‌توانید از پوشه‌ی ta_first_class که کدهایی است که در کلاس حل تمرین توضیح داده‌شده‌است، استفاده نمایید.

نکات ضروری

- توجه کنید که شما باید روی فایل‌های باینری داده‌شده در این تمرین shell بگیرید و دوباره کامپایل کردن از روی source code ممکن است باعث تغییر آدرس‌ها در فایل باینری و درنتیجه عدم موفقیت در ماشین تست شود.
- به علت اینکه بخشی از نمره‌ی تمرین به صورت خودکار داده می‌شود، ساختار پوشه‌ی تحویل داده‌شده باید دقیقاً به صورت گفته‌شده باشد.
- در صورتی‌که هر مشکل یا پرسشی داشتید که فکر می‌کنید پاسخ آن برای همه مفید خواهد بود، آن را در صفحه‌ی پرسش و پاسخ درس در Quera یا در میل لیست بپرسید. .
- از فرستادن جواب تمرین به گروه اینترنتی درس خودداری کنید.
- تمام برنامه‌ی شما باید توسط خود شما نوشته شده باشد. فرستادن کل یا قسمتی از برنامه‌تان برای افراد دیگر، یا استفاده از کل یا قسمتی از برنامه‌ی فرد دیگری، حتی با ذکر منبع، تقلب محسوب می‌شود.
- پس از اتمام کارتان، لازم است که گزارش و پوشه‌ی exploits را با فرمت tar.gz. فشرده کرده و با اسم NS_PA1_#STDID.tar.gz در Quera بفرستید. گزارش نیز باید با فرمت pdf. باشد و الگوی نام‌گذاری آن مانند فایل فشرده است.
- سایر نکات مربوط به تاخیر و زمان تحویل را می‌توانید از صفحه‌ی درس مشاهده فرمایید.