MALWARE

server-client topology

NETWORK PROGRAMMING PROJECT

Prepared For :

Network course at Isfahan university of technology

Dr . Ali Fanian

Ali Dakik - Zahra Sarami

Deadline:

sunday 19th Farvadin

مقدمه :

در این پروژه قرار است با استفاده از socket programming و در توپولوژی کلاینت سرور طریقه ی عملکرد یک ویروس را شبیه سازی کنیم .
در پیاده سازی این پروژه از دستورات Linux bash استفاده شده است لذا در بستر این سیستم عامل پروژه را انجام دهید .
زبان های مجاز :

python - c - cpp

عملکرد ویروس :

فایل ویروس در سمت کلاینت ها (victim) اجرا میشود و به سرور (attacker) متصل میگردد .
هدف از پیاده سازی ویروس در این پروژه این است که بتوانیم از راه دور (از طریق سرور) دستورات مد نظر خود را روی سیستم کلاینت اجرا کنیم .
این دستورات در واقع bash command هایی مانند pwd , ls , ... هستند که انتظار داریم پس از اجرا در سیستم قربانی خروجی شان برای سرور ارسال شود .
جهت آشنایی بیشتر با مفهوم bash و دستورات آن میتوانید این مقاله را بخوانید.
در واقع ویروس با وصل کردن ورودی و خروجی shell سیستم خود به سرور، به آن دسترسی غیر مجاز میدهد .
جهت آشنایی بیشتر با این سناریو میتوانید درباره Reverse Shell بخوانید.

شرح پروژه :

شبکه :

در این پروژه اساس کار انتقال داده بر پایه ی برنامه نویسی سوکت و پروتکل tcp است .
همچنین یک کد سرور(attacker) و به تعداد دلخواه کد کلاینت(victim) خواهیم داشت که به سرور به طور همزمان متصل شده و توسط آن مدریت میشوند .
اندازه بافر tcp مقدار محدودی بوده و طول داده های ارسالی ما نامعین است لذا مدریت ترتیب و نحوه ی ارسال به گونه ای که داده ها صحیح منتقل شوند به عهده شماست .
مقدار ip و socket port اصلی سرور مقدار ثابتی بوده و به صورت متغیر static در کد malware، مقداردهی میشوند .
کد سرور و هر یک از کلاینت ها در ماشین های مجازی مجزا اجرا میشوند که به صورت Nat یا bridge با هم شبکه شده اند .

کلاینت :

در سمت کلاینت کد malware اجرا میشود .
این کد باید علاوه بر ایجاد socket connection با سرور، بسته های ارسالی از آن را دریافت کرده و دستور bash موجود در آن را استخراج و اجرا کند .
همچنین نتیجه ی اجرای دستور باید برای سرور ارسال و نمایش داده شود .
کد راهنمای پیاده سازی این بخش در زبان سی را میتوانید در پیوست مشاهده کنید .

بخش امتیازی :

۱- برنامه ی کلاینت پس از اجرا شدن، یک interface گرافیکی داشته باشد که پنهان کننده کد ویروسی باشد که در پشت کد اجرا میشود .
در واقع کاربر سیستم با اجرا کردن کد ویروس یک برنامه ی غیرمخرب و کاربردی (مثلا پنجره ای که یک عکس را نمایش میدهد) میبیند ولی در پشت آن کد ویروس شما در حال فرستادن داده است .

۲- اجرا کردن دستور cd علاوه بر دستورات امتیاز اضافه دارد.

سرور :

سرور به طور پیش فرض در حال Listen کردن شبکه روی یک پورت مشخص است .
با اجرا شدن کد malware در سمت کلاینت ، درخواست اتصال به سرور آن باید توسط سرور پذیرفته و مدریت شود .
برای این کار نیاز است که کانکشن های ایجاد شده را در یک آرایه ذخیره کنید تا در صورت نیاز به آنها دسترسی داشته باشید .
پس از برقراری ارتباط باید دستورات وارد شده، به سمت کلاینت ارسال شوند و نتیجه ی آن در interface مربوطه نمایش داده شود .

نمایش عملکرد :

ابتدا با اجرای کد لیستی از کانکشن های برقرار شده را در interface اول میبینیم که انتظار میرود با ایجاد شدن کانشن جدید این صفحه به روزرسانی شده و اتصالات سرور را مانیتور کند .
سپس کاربر میتواند از بین کلاینت ها یکی را انتخاب کند تا برای آن دستور ارسال کند .
با انتخاب یک کلاینت صفحه ی دوم باز شده و در آن دستور تایپ شده و نتیجه ی اجرای دستور برای ما نمایش داده میشود .

بخش امتیازی :

پیاده سازی امکانات مدریتی برای سرور نمره ی امتیازی دارد .
این امکانات شامل نام گذاری ، مرتب کردن و بستن کانکشن یک کلاینت خاص است.
به این صورت که سرور ابتدا لیستی از کلاینت های فعال میبیند و میتواند از میان آنان یکی را برای ارتباط انتخاب کند و یا آن ارتباط را نام گذاری کند و در صورت نیاز آن کانکشن به خصوص را از بین ببرد .
همچنین attacker باید بتواند یک دستور خاص را برای همه ی victim های متصل ارسال کند و خروجی را به نحو مناسبی ببیند .

قابلیت دانلود و آپلود فایل (فاز دوم پروژه) :

در این قسمت میخواهیم علاوه بر دستورات bash قابلیت انتقال فایل بین سیستم attacker و victim ها را فراهم کنیم .

دانلود:
فرمت دستور ارسال شده از سرور به کلاینت به فرم زیر است :

```
>> DOWNLOAD  <file path>/filename.txt  <destonation path>
```

با اجرای این دستور فایل filename.txt از ماشین victim کپی شده و در <destonation path> سرور قرار میگیرد .

آپلود:
فرمت دستور ارسال شده از سرور به کلاینت به فرم زیر است :

```
>>UPLOAD  <file path>/filename.txt  <destonation path>
```

با اجرای این دستور فایل filename.txt از ماشین attacker کپی شده و در <destonation path> کلاینت قرار میگیرد .
قرارداد پروتکل و نوع تقسیم داده و ارسال آنها میان مبدا و مقصد به گونه ای که صحت آن حفظ شود به عهده شما میباشد .

با اتمام هر یک از فرآیند های بالا پیامی مبنی بر موفق بودن انتقال نمایش داده میشود و کانکشن بسته میشود .

پیوست :

راهنمای پیاده سازی کلاینت :

اجرای دستورات:

یکی از روشهای اجرای دستورات داخل زبان C استفاده از تابع ()popen است این تابع یک کانال ارتباطی یک طرفه (لوله) بین پروسس اصلی و یک فرآیند فرعی می سازد، که در حالت ما این فرآیند فرعی یک دستور لینوکس است. به پروسس اصلی اجازه می دهد تا دستوری را طوری اجرا کند که گویی در خط فرمان تایپ شده است و در ورودی یا خروجی استاندارد آن دستور بخواند یا بنویسد.

پروتوتایپ تابع:

```
FILE *popen(const char *command, const char *mode);
```

پارمترها:

- command : یک استرینگ که حاوی کامندی که میخواهیم آن را اجرا کنیم.
- mode : یک استرینگ که نوع کانال ارتباطی را مشخص میکند، دو مدل "r" برای اجرا و "w" برا خواندن داریم.

خروجی:
خروجی تابع از نوع فایل است که مانند هر فایلی که در زبان C هست میتوان از توابع , fread ,fprintf... برای خواندن آن استفاده کنیم. توجه کنید که اگر error رخ دهد تابع Null برمیگردوند.

ودرنهایت توصیه میشود پس از اتمام کار کانال ارتباطی را با استفاده از تابع زیر ببندیم.

int pclose(FILE *stream);

نمونه ای از تابع:

```c
#include <stdio.h>

#define BUFFER_SIZE 1024

int main() {

        char buffer[BUFFER_SIZE];

        // Execute command
        FILE *command_output = popen("ls -l", "r");
        if (command_output == NULL) {
            perror("Command execution failed");
            return -1;
        }

        // Read command output line by line
        while (fgets(buffer, BUFFER_SIZE, command_output) != NULL ) {
            printf("%s", buffer);
        }


        // Close command output stream
        pclose(command_output);

        return 0;
}
```

نکته :

تضمین میشود حجم خروجی هیچ دستوری بالاتر از یک کیلوبایت نیست .

نکات پایانی:

مهلت تحویل :
شما تا پایان مهلت ددلاین فرصت دارید فایل کد پروژه به همراه فایل های پیوست آن را به صورت یک فایل فشرده(zip.) در سامانه ی یکتا در ماژول مربوطه بارگزاری کنید .

ارائه :
ارائه ی پروژه به صورت فردی و حضوری بوده و باید بتوانید پروژه را در سیستم خود اجرا کنید همچنین باید بتوانید کد ، نحوه ی اجرای برنامه و سیاست های پیاده سازی شده در برنامه خود را توضیح دهید .
تاریخ ارائه متاقبا اعلام میشود .

نمره دهی :
مجموع نمره ی بخش های امتیازی از ۱۰۰ محاسبه شده و توزیع آن به این صورت است :

- پیاده سازی ui interface کلاینت : 20
- پیاده سازی دستور cd علاوه بر دیگر دستورات : 25
- امکانات مدریتی برای سرور : 30
- امکان ارسال دستور برای همه ی victim ها : 25

راه های ارتباطی :

Telegram : @X_AFDK_X
Telegram : @zhra_sarami