بنام خدا

گزارش تمرین ایجاد کانال پنهان زمانی

میلاد تیموری ۹۵۷۲۵۱۲۷
دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران
پست الکترونیکی: milad72t@gmail.com
خرداد ۹۶

۱- مقدمه

کانال پنهان، کانالی است که به دو پردازش مختلف این اجازه را می‌دهد که بتوانند با یکدیگر به تبادل داده‌هایی که می‌تواند منجر به نقض سیاست امنیتی سامانه شود، بپردازند. تمرین ایجاد کانال پنهان، در سامانه عامل لینوکس و با زبان پایتون انجام‌شده است. سامانه عامل لینوکس به تمام کاربران موجود در سامانه این اجازه را می‌دهد که بتوانند لیست تمام پردازش‌های در حال اجرا در سامانه را مشاهده کنند؛ اما اجازه بستن یا تغییر در پردازش را نمی‌دهد. با استفاده از همین آسیب‌پذیری و اجرای یک پردازش مشخص میان فرستنده و گیرنده در زمان‌های مشخص، فرستنده و گیرنده می‌توانند به تبادل داده‌های پنهان بپردازند.

۲- شیوه تبادل اطلاعات

شیوه تبادل اطلاعات در این کانال پنهان به این‌گونه است که فرستنده داده‌های ارسالی را به قطعات ۲ بیتی تقسیم می‌کند و بسته به مقدار ۲ بیتی که شامل داده‌ی پنهان است، در زمان مشخصی اقدام به اجرای یک پردازش به مدت نیم ثانیه می‌کند.

Timeout 0.5 python CovertChannelLopp.py

فایل پایتون اجراشده چیزی جز یک حلقه همواره صحیح نیست، که بعد از گذشت نیم ثانیه توسط دستور timeout از بین برده می‌شود. با توجه به نویز کانال و تأخیر بین زمان اجرای دستور و متوجه شدن گیرنده، هر ۲ ثانیه زمان بیانگر مقدار ۲ بیت خواهد بود. شیوه ارسال اطلاعات این‌گونه خواهد بود که فرستنده قطعه ۲ بیتی که هم‌اکنون باید ارسال شود را در نظر گرفته و با توجه به مقدار دسیمال این داده و جدول مدولاسیون زیر، صبر می‌کند تا رقم آخر مقدار برچسب زمانی سامانه برابر با مقدار متناظر با ۲ بیت داده‌ی ارسالی شود و به‌محض برابر شدن رقم سمت راست برچسب زمانی با مدولاسیون داده اقدام به اجرای دستور فوق که باعث ایجاد یک پردازش در سامانه می‌شود، می‌کند.

جدول ۱: سطوح مدولاسیون

مقدار باینری داده پنهان	رقم سمت راست متناظر با داده موردنظر در مقدار برچسب زمانی
۰۰	۰ و ۱
۰۱	۲ و ۳
۱۰	۴ و ۵
۱۱	۶ و ۷
انتهای فایل	۸ و ۹

پس از ارسال داده موردنظر و ارسال سیگنال انتهای فایل، فرستنده برای اینکه گیرنده با توجه به تأخیرهای موجود در سیستم بتواند داده پنهان را دریافت کند، به مدت نیم ثانیه به حالت خواب می‌رود.

به‌عنوان‌مثال درصورتی‌که داده پنهان جهت ارسال مقدار "۱۰" باشد، فرستنده تا زمانی که رقم سمت راست برچسب زمانی به مقدار ۴ نرسیده است صبر می‌کند و به‌محض رسیدن به این مقدار اقدام به ایجاد پردازش موردنظر می‌کند.

حال در سمت دیگر، گیرنده نیز مدام در حال بررسی پردازش‌های موجود در سامانه است و به دنبال وجود پردازش موردنظر در سامانه با استفاده از دستور زیر است:

```
Ps –ef | grep "python CovertChannelLopp.py"
```

به‌محض اینکه گیرنده به وجود پردازش موردنظر پی ببرد مقدار رقم سمت راست برچسب زمانی را می‌خواند و با استفاده از جدول ۱ به داده پنهان موردنظر دست پیدا می‌کند؛ و بعدازآن به‌منظور هم‌زمانی، به‌مانند گیرنده به مدت نیم ثانیه به حالت خواب می‌رود.

```
milad72t@milad72t ~/Public/course/SafeSystem $ whoami
milad72t
milad72t@milad72t ~/Public/course/SafeSystem $ python sender.py
data to send :  10110100011010
--------------------
10 bit sent
--------------------
11 bit sent
--------------------
01 bit sent
--------------------
00 bit sent
--------------------
01 bit sent
--------------------
10 bit sent
--------------------
10 bit sent
--------------------
EOF sent
--------------------
Done!
```

شکل ۱: اجرای برنامه ارسال‌کننده کانال پنهان

```
$ whoami
guest
$ python receiver.py
10 bit received
--------------------
11 bit received
--------------------
01 bit received
--------------------
00 bit received
--------------------
01 bit received
--------------------
10 bit received
--------------------
10 bit received
--------------------
EOF received
--------------------
received data =   10110100011010
$
```

شکل ۲: اجرای برنامه دریافت‌کننده کانال پنهان

۳- کارهای آینده

ازآنجایی‌که ایجاد پردازش توسط فرستنده کانال پنهان و دریافت اطلاعات توسط گیرنده معمولاً با تأخیری بین ۱۰۰ تا ۲۰۰ میلی‌ثانیه همراه است از نیم ثانیه حالت خواب جهت جلوگیری از ایجاد تداخل و از بین رفتن داده‌های پنهان استفاده می‌شود و هر ۲ ثانیه می‌تواند باعث ارسال ۲ بیت داده پنهان شود. می‌توان با قرار دادن روش‌های تصحیح خطا یا ارسال مجدد در ارتباط بین فرستنده و گیرنده، فاصله زمانی ارسال بسته‌های مختلف و مقدار حالت خواب هرکدام از پردازش‌های فرستنده و گیرنده را کاهش داد و به پهنای باند ۳ بیت بر ثانیه دست پیدا کرد .