کالبدشکافی عملیات IP Spoofing

مبحث IP Spoofing یکی از هیجان انگیزترین موضوعات در بین افرادی است که شروع به مبحث Hacking کرده اند و در عین حال موضوعی است که افرادی زیادی از آن اطلاع دقیق ندارند. قبل از شروع باید این نکته نیز ذکر شود که درک مفهوم IP Spoofing مقداری دشوار خواهد بود و بسیاری از افراد در درک این مکانیزم مشکل دارند. مورد دیگری که وجود دارد آن است که تقریبا این روش را نمی توان در سیستم های Windows پیاده سازی کرد که یک مدیر سیستم به راحتی می تواند سیستم خود را از IP Spoofing در امان نگه دارد.

IP Spoofing در حقیقت حیله ای است که معمولا بر روی سرورها اعمال می شود و معمولا به این معنی استفاده می شود که سیستم سرور مقابل را طوری فریب دهید تا فرض کند مقصدی که از آن اطلاعات دریافت می کند شما نیستید (و می تواند هر کسی دیگر باشد)! در نهایت کامپیوتر مقصد بسته های اطلاعاتی را از شما دریافت خواهد کرد اما پیش خود بر این فرض خواهد بود که این بسته ها از ماشین دیگری (که ماشین شما نیست) دریافت می شوند. با مثالی این مطلب را روشن می سازیم:

در این مثال:

-IP Address خود را به صورت فرضی 203.45.98.1 (واقعی) در نظر می گیریم.

-IP Address سیستم قربانی نیز 202.14.12.1 (قربانی) در نظر گرفته می شود.

-آن IP Address که می خواهیم وانمود کنیم برای ما است، 202.14.12.1 (تقلبی) می باشد.

و به طور سمبولیک به صورت زیر می باشند:

Real IP Address (yours): 203.45.98.1
Victim IP Address: 202.14.12.1
Fake IP Address (yours): 173.23.56.89

در حالت عادی، هنگامی دیتاگرام ها از کامپیوتری با IP Address واقعی به سمت قربانی خارج می شود، مسلما اطلاعات دریافتی توسط قربانی نیز مشخصه IP Address شما را در بر دارند و این به آن معناست که کاملا طرف مقابل کامپیوتر قربانی (که شما هستید)، به راحتی قابل شناسایی خواهد بود. اکنون درباره حالتی فکر کنیدکه شما می خواهید بسته هایی را برای قربانی بفرستید، اما او گمان کند که این بسته ها از سیستم تقلبی دیگر (Fake) با IP Address برابر 173.23.45.89 می آیند. در اینجا باید از مکانیزم IP Spoofing بهره جست!

بیائید فرض کنیم که سه فـــرد با نام های A, B و C وجود دارند. ما در این بین B هستیم و می خواهیم A را فریب دهیم. می توانیم از پشت تلفن به شخص A زنگ بزنیم و طوری صدا را تغییر دهیم که شخص A تصور کند شخص مقابل فرد C می باشد و بنابراین شخص گمان خواهد کرد که واقعا با شخص C در حال صحبت است. اکنون اگر شما، سه شخص مذکور در بالا فوق را با سه کامپیوتر تغییر دهید و نیز شرط و موضوع صدا (مربوط به تغییر صدا دادن) را با IP Address (مربوط به IP Spoofing) تغییر دهید، به راحتی می توانید مفهوم IP Spoofing را درک کنید.

مشکلاتی که در هنگامی اجرای IP Spoofing با آن روبرو هستید

یکی از دلایلی که چرا عملیات IP Spoofing برای اجرا اینقدر دشوار شده است، آن است که در حقیقت یک حمله کـــــــورکورانه می باشد. منظوری که از حـــــــمله کورکورانه مورد نظر است، آن است که، در حقیقت ما هیچ گونه پیام یا feedback regard را به پروسه های خودمان دریافت نخواهیم کرد. هنگامی که یک نفوذگر تلاش بر انجام عملیات IP Spoofing دارد، هیچ گونه مکانیزمی وجود ندارد تا به او بگوید در کارش موفق شده یا خیر! اگر بلی، تا چه حد و اگر خیر، چه مشکلی وجود دارد. بنابراین می توان به صورت تحت اللفظی گفت که در حقیقت نفوذگر دست به یک حمله کورکورانه زده است!

در زیر علت این موضوع را ذکر کرده ایم:

مشکل اصلی در رابطه با IP Spoofing این است که حتی اگر شما (Real)، قادر باشید که یک datagram به صورت Spoof شده را به سمت سیستم مقابل (Victim) بفرستید و طوری عمل کنید که سیستم مقابل تصور کند که این بسته ها از سیستم دیگری می آیند (Fake)، در این هنگام سیستم قربانی به IP Address که به صورت Spoof شده می باشد (Fake) جواب خواهد داد نه به سیستم شما (که با آدرس واقعی می باشد)! در نتیجه، به هیچ وجه سیستم شما (Real)، هیچ گونه feedback را در خصوص پروسه های خود، دریافت نخواهد کرد.

در زیر توضیحاتی درباره طبیعت کورکورانه ی IP Spoofing می خوانید، که از مفهوم دست تکانی ۳ مرحله ای استفاده شده است که مجبور است هر هنگام که یک ارتباط TCP/IP برقرار می شود، اتفاق بیفتد.

توجه: در ادامه متن برای سهولت در نوشتار و نیز افزایش درک در خواننده از کلمات REAL به منظور سیستم واقعی شما (که آدرس آن مسلما Spoof شده نیست)، از VICTIM برای معرفی قربـــــــــانی و از FAKE برای معرفی سیستم شـــــــما هنگامی که از IP Spoofing استفاده می کند، استفاده کرده ایم! در زیر سمبولیک این توضیح را می بینید:

Your System – Without Spoofing ===> REAL
Victim's System ===> VICTIM
Your System –With IP Spoofing ===> FAKE

اکنون به توضیح این فرآیند می پردازیم:

اگرREAL بخواهد، یک ارتـــــباط TCP/IP با VICTIM، بـــــدون spoof کردن هیچ گونه IP Address (Without Spoofing) برقرار کند، در این هنگام در حالت عادی فرآیند دست تکانی مرحله ای به دست زیر اتفاق خواهد افتاد:

۱. REAL یک بسته SYN را به VICTIM خواهد فرستاد.
۲. VICTIM یک بسته SYN/ACK به REAL پس خواهد فرستاد.
۳. REAL آن را با پاسخ یک بسته SYN، تصدیق می کند.

اما، اگرREAL بخواند IP Address خود را Spoof کند (With IP Spoofing) و آنرا به صورت تقلبی (FAKE) نمایش دهد، در این هنگام مراحل زیر اتفاق خواهد افتاد:

۱. REAL یک بسته SYN را به VICTIM خواهد فرستاد؛ اما اینبار با آدرس مقصدی (Source Address) برابر با FAKE.

۲. **VICTIM یک بسته SYN/ACK را به FAKE، پس می فرستد. هیچ راهی برای REAL وجود ندارد تا معلوم کند آیا VICTIM واقعا درخواست را با SYN/ACK جواب داده است یا خیر. چرا که همان طور که گفته شد بسته SYN/ACK به FAKE خواهد رفت نه به REAL. این قسمت از فرآیند در واقع منشا کورکورانه بودن این حمله می باشد و REAL تنها مجبور است مقداری صبر کند (هنگامی که او یک بسته SYN را به قربانی فرستاد) و بعد از مدت زمانی (کمتر از ۳۰ ثانیه)، حدس بزند که VICTIM دیگر باید پاسخ را با یک SYN/ACK به FAKE داده باشد.**

۳. **بعد از گذشت مدتی (کمتر از ۳۰ ثانیه)، REAL مجبور است یک بسته SYN را به قربانی بفرستد. این بسته SYN در واقع تصدیقی می باشد مبنی بر اینکه FAKE بسته ی SYN/ACK را دریافت کرده است.**

اگر تمامی سه مرحله ی برنامه ریزی شده در فوق اجرا شوند، یک ارتباط TCP/IP می تواند بین VICITM و REAL به وسیله FAKE برقرار شود!!

اما، این سه مرحله، به یک مشکل دیگری منجر خواهند شد. در مرحله دوم در دست تکانی IP Spoofing، ما می بینیم که VICTIM یک SYN/ACK را به FAKE می فرستد و این مقوله دو راه و مسیر مختلف را به وجود می آورد:

۱. **در مورد اول فرض می کنیم که سیستمی با IP Address برابر با FAKE در حال حاضر فعال و Active می باشد. آن وقت، FAKE یک بسته ی SYN/ACK دریافت خواهد کرد که از سمت VICTIM برای او فرستاده شده است. مسلما سیستم FAKE هم انتظار دریافت چنین بسته ای را نداشته است و این بسته در واقع هیچ معنی را برای او نخواهد داشت و بنابراین آنرا با یک پیام بی-اعتبار بودن یا NACK (Non-Acknowledgement) به VICTIM پاسخ خواهد داد.**

تذکر: در داخل پرانتز باید ذکر شود که یک پیام NACK یا Non-Acknowledgement به معنای خاتمه و پایان یک ارتباط می باشد. در نتیجه، ارتباط بین دو سیستم قطع خواهد شد و دیگر وجود نخواهد داشت. درخواست مجدد ارتباط برای هر طرف بعد از ۳۰ ثانیه باید انجام گیرد!!

بنابراین، VICTIM با دریافت این پیام NACK از FAKE ارتباط را قطع خواهد کرد و این در حالی است که REAL در حال تلاش برای برقراری ارتباط بین VICTIM و FAKE می باشد (بدون اطلاع خودشان). بدین صورت، تلاش REAL برای Spoof کردن، IP Address خود بی معنی بوده و خنثی می شود. IP Spoofing تنها زمانی موفقیت آمیز خواهد بود که کامپیوتری که با آدرس FAKE در نظر گرفته می شود، به سیستم VICTIM جواب ندهد و در نتیجه ارتباط Spoof شده را قطع نکند. باز هم به مثال تلفن فوکوس می کنیم: شما (B) می توانید به شخص A زنگ زده و وانمود کنید که C هستید و این تا زمانی است که C محاوره بین شما را قطع نکند و در نتیجه به بازی خاتمه داده نشود ☺ !!

۲. **در مورد دوم فرض می کنیم که سیستمی که با IP Address برابر با FAKE در نظر گرفته شده است در حال حاضر وجود نداشته باشد و به اصطلاح exist یا active نباشد. در این صورت، VICITM هیچ پیامACK از FAKE در برابر بسته ی فرستاده شده ی SYN/ACK دریافت نخواهد کرد. بنابراین، یک روند Time Out اتفاق خواهد افتاد و در نتیجه VICTIM ارتباط را با FAKE پایان می دهد و در نتیجه، تلاش REAL برای Spoof کردن IP Address خود بی نتیجه خواهد بود.**

بدین گونه، توضیحات فوق نتایج زیر را در برداشت، برای ما خواهد داشت:

۱. **IP Spoofing یک حمله کورکورانه می باشد (که Blind نامیده می شود) و ما هیچ گونهfeedback را مربوط به پروسه های خود**

دریافت نخواهیم کرد و بنابراین هیچ نظری درباره اینکه آیا موفق شده ایم یا خیر نخواهیم داشت. بسته های تقلبی (fake) که به سیستم هدف (victim) می فرستیم، وانمود می کنند که از سیستم FAKE می آیند.

۲. اگر REAL می خواهد IP Address خود را spoof کند و طوری تظاهر کند که FAKE می باشد، در این صورت، شروط زیر باید درست باشند:

الف)-FAKE باید به اینترنت وصل بوده و در نهایت وجود داشته باشد.

ب)-FAKE باید هیچ گونه جواب (respond) در برابر بسته SYN/ACK که برای او می آید، نداشته باشد. اینجاست که یک حمله DoS یا SYN Flooding آماده به کار می باشد ☺!

ج)-اگر شما یک ارتباط trust را اکسپلویت می کنید، آنوقت FAKE باید طوری انتخاب شود که، FAKE و VICTIM یک ارتباط trust با هم داشته باشند.

قبل از شروع به یک راهنمایی مرحله به مرحله در مورد استفاده IP Spoofing برای اکسپلویت کردن ارتباط های trust کنیم، احتیاج دارید که مفهوم های ویژه ای که درگیر با IP Spoofing هستند را درک کنید.

موضوعات پایه ای درگیر با مبحث IP Spoofing

Sequence Number ها، دلیل این است که چرا سیستم مقصد (**destination**) قادر است که مقدار کمتری از قطعه اطلاعاتی که از سیستم منابع می گیرد را در یک قطعه بزرگ تر قرار دهد. تمامی اطلاعات که روی اینترنت در حال انتقال هستند، در منبع (**Source**) شکسته و قطعه قطعه می شوند و سپس در مقصد (**Destination**) بازسازی می گردند. اطلاعات به بسته هایی با یک توالی یا **Sequence** ویژه در منبع شکسته می شوند. این به آن معناست که: برای مثال، اولین بایت، اولین **Sequence Number** را دارد، دومین بایت، دومین **Sequence Number** و الی آخر!! بسته ها روی اینترنت، می توانند به صورت مستقل از هم به سمت مقصد سیر کنند و برسند. بنابراین، بعضی مواقع، هنگامی که بسته های اطلاعاتی به مقصد می رسند، با نظم **Sequence Number** این کار شاید انجام نشده باشد. این به آن معناست که مثلا، بعضی مواقع شاید پیش بیاید بسته ای که دارای **Sequence Number** برابر ۴ است، زودتر از بسته ای با **Sequence Number** برابر ۳ ، می رسد.

Sequence Number در بسته ها، سیستم مقصد را کمک می کنند تا در بازسازی و **Merge** کردن بسته ها به بسته ی اولیه (که مسلما به دلیل حجم بزرگ قطعه قطعه شده بوده است)، دچار اشتباه نشود و بسته های خورد شده با همان نظمی و توالی که در ابتدا بودند، بازسازی شوند.

لایه **Application** یا لایه ای که در حال اجرا در سیستم مقصد است، به صورت اتوماتیک بسته های بزرگ تر را به وسیله سوار کردن مجدد قطعه های رسیده از منبع، با کمک **Sequence Number** ها، ایجاد می کند.

Sequence Number ها تنها سیستم مقصد را در بازسازی بسته ها به بسته اولیه کمک نمی کنند، بلکه همچنین مطمـــئن می سازند که **TCP** یک پروتکل معتبر باقی مانده است، که می تواند با بسته های از دست رفته، دو نسخه ای و بسته های غیرقابل انتظار سر و کار داشته باشد (**Lost Packets, Duplicated Packets & UnOrdered Packets**)!

در زیر، شکلی عادی از یک **TCP Header** از یک بسته اطلاعاتی که به وسیله **host** به **client** فرستاده می شوند، می بینید:

| Source Port | Destination Port |

```
-----------------------------------------------------------------
| Sequence Number |
-----------------------------------------------------------------
| Acknowledgment Number |
-----------------------------------------------------------------
| Data | |U|A|P|R|S|F| |
| Offset | Reserved |R |C|S|S|Y|I | Window |
| | |G|K|H|T|N|N| |
-----------------------------------------------------------------
| Checksum | Urgent Pointer |
-----------------------------------------------------------------
| The Actual Data follows |
| |
```

توجه: در پاراگراف های زیر، ما فرض می کنیم که فیلدها و مقادیری (Values)که درباره آنها صحبت می کنیم و متعلق به یک بسته هستند، به وسیله یک Host به یک Client فرستاده می شوند.

فیلد Sequence Number **در TCP Header شامل شماره تـــــوالی بسته اطلاعاتی است که در حال حاضر در حال انتقال می باشد. هر بسته که روی اینترنت در حال سیر به سمت مقصد است، ترتیب بندی شده است (Sequenced)! موارد گوناگونی درباره** Sequence Number **وجود دارد، که یکی از آنها باید به درستی و دقیق توضیح داده شود. تمامی آنها در پاراگراف های بعدی مطرح و توضیح داده شده اند.**

یک Sequence Number **در حقیقت یک شماره ۳۲ بیتی (32-Bit) می باشد و مـــحدود و** range **آن از** 1 **تا** 4,294,967,295 **می باشد. اکنون این سوال پیش می آید که چطور یک** host **تصمیم می گیرد که چگونه**Sequence Number **ها، باید به یک ارتباط محول و** assign **شوند؟ این سوال، به ما یک عبارت دیگر معرفی می کند و آن Initial Sequence Number یا ISN (اولـــین شماره توالی) می باشد.**

Initial Sequence Number **یا** ISN**، اولین شماره توالی یا** sequence number **می باشد که به یک** host**، در زمانی که سیستم در حال bootstrapped می باشد، داده می شود. در زمان** bootstrapping**،** ISN **مقداری برابربا ۱ خواهد داشت. هنگامیکه یک** host**، ارتباطی برقرار می کند، مقدار مربوط به** ISN **آن در آن نقطه مخصوص از زمان، به عنوان اولین شماره توالی از اولین تکه در حال ارسال به کلاینت می باشد، رفتار می کند، که با آن ارتباط برقرار خواهد شد! هنگامی که، یک سیستم دارای مقدار** ISN**، برابر ۱ در زمان** bootstrapping **بود، آن وقت، این مقدار به صورت اتوماتیک زیاد می شود و به صورت پیش گویانه ای با گذشت زمان، انتقال اطلاعات و برقراری ارتباط در بر خواهد داشت!**

این اولین شماره توالی یا ISN **در هر ثانیه مقدار** 128,000 **زیاد می شود و با هر ارتباطی که در حـــــال برقـــــراری است، آن به مقدار** 64,000 **زیاد خواهد شد. با تکیه بر این مطلب که: با گذشت هر ثانیه،** ISN **به مقدار** 128,000 **افزایش می یابد، می توان نتیجه گرفت که آن هر** 9.32 **ساعت،** Wrap **می شود و در نتیجه خاتمه خواهد یافت (در نتیجه تصور خواهد شد که هیچ ارتباطی برقرار نیست)!**

برای مثال:

اگر ISN **مربوط به یک** Host **برابر** 1897737287 **باشد، آنوقت بعد از ۳ ارتباط** (connection) **و دو ثانیه،** ISN **مربوط به آن باید برابر با مقدار زیر باشد:**

1898185287 = 1897737287+ (3*64 000) + (2*128 000)

برای بازگو کردن نکات مهم، می توانیم بگوئیم که مقدار مربوط به **Sequence Number** در **TCP Header**، باید **Sequence Number** مربوط به اولین بایت از اطلاعات در آن قطعه ی مخصوص باشد.

نکته: نکته مهمی در اینجا برای به خاطر سپردن نیز وجود دارد و آن این است که **Sequence Number** مربوط به اولین بایت از اطلاعات در حال ارسال به وسیله **host** به **client**، باید برابر با مقدار **ISN + 1** باشد (مقدار**ISN** بعلاوه ۱) !! دلیـــــل آن است که **SYN Flag**، مقدار ۱ را در **Sequence Number**، می گیرد و **Take Up** می کند! این مورد در ادامه بـــــــعد از بحث "***Sequence Number*** *ها و برقراری و پایان ارتباط*"، روشن تر و واضح تر خواهد شد.

در فیلد **Acknowledgement Number** نیز مثل **Sequence Number** باید یک شماره ۳۲ بیتی قرار داده شود، اما، این **Sequence Number** برای **Host** نیست بلکه برای **Client** می باشد (همان طور که در بررسی مثال و در عنوان یک *توجه* گفتیم: فرض کرده ایم که این بسته از سمت **Host** به **Client** فرستاده می شود). منظور این است که **Sequence Number** در فیلد **Acknowledgement Number**، در حقیقت مقدار مربوط به **Sequence Number** بعدی را نمایش می دهد (یعنی مقدار مربوط به sequence number بعدیِ اطلاعات) که Host از Client انتظار فرستادن آنرا دارد. این شماره همچنین تصدیق می کند که همه اطلاعات دارای این *شماره منهای یک* (**Acknowledgement Number – 1**) ، بسلامت دریافت شده اند!

Sequence Number ها و برقراری و پایان ارتباط

برای برقراری یک ارتباط **TCP/IP**، باید یک فرآیند دست تکانی سه مرحله ای بین کلاینت و **host**، رخ دهد. در زیر این سه مرحله را که برای یک ارتباط کامل و موفقیت آمیز، باید بین کلاینت و **host** اتفاق بیفتد را می بینید:

۱. کلاینت یک بسته **SYN** را به سرور برای درخواست برقراری یک ارتباط می فرستد. این بسته **SYN** باید شامل **ISN** مربوط به سیستم کلاینت باشد. فرض می کنیم که آن 4894305 می باشد. به دلیل اینکه کلاینت هنوز Sequence Number مربوط به host را نمی داند و هیچ گونه تصدیق برای دریافت اطلاعات نیز ندارد، بنابراین فیـــــــلد Acknowledgement Number را به صفر تغییر می دهد. این بسته همچنین شامل اطلاعات دیگری مانند آدرس مقصد و شماره پورت و ... می باشد.

۲. **host** در هنگام دریافت این بسته، آنرا با یک بسته**SYN/ACK** جواب خواهد داد. این بسته، باید شامل شماره **ISN** مربوط به سرور باشد. فرض می کنیم که آن 1896955367 می باشد. آن همچنیـــــــــن باید در Acknowledgement Number شامل، Sequence Number نیز باشد که در حقیقت، دلالت بر توالیِ اطلاعاتِ موردِ انتظارِ بعدی دارد و نیز تصدیق رسید اطلاعات بعدی را نیز می دهد. مقدار این Acknowledgement Number همیشه برابر با مقدار ISN مربوط به کلاینت به علاوه ۱ می باشد یعنی:

Client's ISN+1

بنابراین، در مثال ما، مقدار آن باید 4894305+1=4894306 باشد.

۳. سپس کلاینت، آنرا با یک بسته ACK جواب می دهد. فیلـــد ACK Number اکنون باید، ISN مربوط به سرور بعلاوه ۱ (**Server's ISN +1**) باشد که برابر با 1896955368خواهد بود.

به صورت اشکال هندسی می توان این مراحل را به صورت زیر نمایش داد:

Client----------SYN (4894305)----------→ Host
Host----------SYN (1896955367) and ACK (4894306)---------→ Client
Client----------ACK (1896955368)----------→ Host

در مطالب فوق، ما ارتباطِ برقراری ارتـــباط TCP/IP را با **Sequence Number** ها بیان کردیم. اکنون بعد از مراحل بالا، می خواهیم به طور ناگهانی ارتباط را قطع کنیم (ادامه کار با همان مقادیر ذکر شده برای SYN و ACK Sequence Number ها). بنابراین فرآیند زیر اتفاق خواهد افتاد:

Client----------FIN (4894306) and ACK (1896955368)----------→Host
Host----------ACK (4894307)----------→Client
Host----------FIN (1896955368) and ACK (4894307)-------→Client
Client----------ACK (1896955368)----------→Host

نگاه عمیق تری به Sequence Number ها

برای درک بهتر پدیده ی ارزش های **SYN** و **ACK**، اجازه دهید یک آزمایشی را هنـــــــــگامی که ما ابتدا به یک **remote system** به وسیله پورت ۲۳ تلنت (telnet) می کنیم و سپس فورا آنرا به وسیله دستور **Quit** قطع می کنیم، طراحی کنیم. **Header** های تمامی بسته های در حال انتقال به وسیله یک **Sniffer** می توانند ذخیره و **Capture** شوند (به خاطر داشته باشید که ما به **remote system** به عنوان **Host** و به سیستم خود به عنوان **Client** نگاه می کنیم):

#telnet targetsystem.com 23

با وارد کردن دستور فوق، انتقال بسته های زیر صورت می گیرد:

1. Client ----------SYN (856779)----------→ Host

فریم (**frame**)، ذخیره و **capture** شده از این انتقال اطلاعات در زیر برای مطالعات بعدی شما تعیین شده است (قسمتی که به صورت **Bold** می باشد، **Sequence Number** مربوط به بسته را نمایش می دهد):

20 53 52 43 00 00 44 45 53 54 00 00 08 00 45 00 00 2C C3 00 40 00 20 06 10 0C CB 5E FD BA CB 5E
F3 47 04 07 00 17 **00 0D 12 CB** 00 00 00 00 60 02 20 00 D9 70 00 00 02 04 05 B4 2D

در اینجا، می بینیم که Client در حال فرستادن یک بسته با فعال بودن گزینه SYN می فرستد که به این معنی است که خواستار ایجاد یک ارتباط TCP/IP می باشد. SYN مبنی بر synchronize کردن **Sequence Number** ها است. در این مثال، **Sequence Number** مربوط به بسته اطلاعاتی که به وسیله کلاینت فرستاده شده است، 856779 می باشد.

2. Host----------SYN (758684758) and ACK (856780)------→ Client

فریم ضبط شده از این انتقال اطلاعات در زیر برای مطالعات بعـــدی شما قرار داده شده است (قسمتی که به صورت **Bold** می باشد، **Sequence Number** و مقدار **ACK** مربوط به بسته را نمایش می دهد):

44 45 53 54 00 00 20 53 52 43 00 00 08 00 45 00 00 2C 8C 05 40 00 39 06 2E 07 CB 5E F3 47 CB 5E
FD BA 00 17 04 07 **2D 38 9C 56 00 0D 12 CC** 60 12 83 2C AC A4 00 00 02 04 05 B4

در اینجا، می بینیم که Host با دریافت درخواستی برای برقراری ارتباط TCP/IP، به آن به وسیله فرستادن یک بسته با روشن بودن هر دوی گزینه های SYN و ACK جواب می دهد. هنگامیکه Host هنوز نیاز به فرستادن،**Sequence Number** مربوط به خود برای **Client** دارد، گزینه SYN هنوز روشن می باشد. در مثال ما، **Sequence Number** مربـــوط به سیستم **Host**، برابر758684758 می باشد. Host همچنین گزینه ACK را فعال می کند و به آن مقداری می دهد که برابر با ISN مربوط به سیستم کلاینت + ۱ می باشد (Client's ISN+1). ACK Number نیز اطلاعات دریافـــــــــت شده تا الان را تصدیق می کند و **Sequence Number** مورد انتظار بعدی را که **Host**، انتظار دریافت آنرا دارد، نمایش می دهد!

در اینجا، **ACK Number** فرستاده شده به وسیله **Host** در واقع **ISN** مربوط به کلاینت + ۱ می باشد که برابر است با:
856779 + 1= 856780

3. Client----------SYN (856780) and ACK (758684759)----------→ Host

فریم ضبط شده از این انتقال اطلاعات برای مطالعات بعدی شما قرار داده است (قسمتی که به صورت **Bold** مشخص شده است، **Sequence Number** و مقدار **ACK** مربوط به بسته را نمایش می دهد):

20 53 52 43 00 00 44 45 53 54 00 00 08 00 45 00 00 28 C4 00 40 00 20 06 0F 10 CB 5E FD BA CB
5E F3 47 04 07 00 17 **00 0D 12 CC 2D 38 9C 57** 50 10 22 38 25 56 00 00

در فریم ضبط شده فوق، ما در می یابیم که کلاینت به بسته SYN/ACK فـــــرستاده شده توسط Host، به وسیله یک ACK message (پیام ACK) جواب می دهد. که در حقیقت این پیام، رسید اطلاعات تا الان را تصدیق می کند و همچنین توالی اطلاعات مورد انتظار بعدی را از Host به وسیله کلایـــــنت، ذکر می کند. در این مـــــثال، مقدار ACK برابر با ISN مربوط به Host + ۱ می باشد (Host's ISN + 1). بنابراین مقدار ACK برابر است با:

Host's ISN + 1= 758684758 + 1= 758684759

نکته: اما نکته ای که وجود دارد در مرحله ۳ می باشد.زمانی که client در حال فرستادن یک پیام ACK به Host می باشد (و نه یک SYN Message)، در این صورت، Sequence Number افزایش پیدا نمی کند. بسته سوم تنها حاوی فعال بودن گزینه ACK، هنگامی که گزینه SYN هنوز خاموش است، می باشد. در نتیجه، Sequence Number افزایش پیدا نمی کند. مفهوم این است که بسته بعدی فرستاده شده توسط کلاینت به سمت سرور، همچنان همان Sequence Number را خواهد داشت و افزایش پیدا نخواهد کرد.

هنگامی که سه مرحله فوق اتفاق افتادند، یک دست تکانی سه مرحله ای برای ایجاد یک ارتباط TCP/IP بین کلاینت و سرور اتفاق خواهد افتاد. به هر حال، در مثال ما، این فرآیند به سرعت اتفاق افتاد، ما از دستور Quit برای Disconnect شدن از Telnet (در حقیقت telnet daemon) استفاده می کنیم. با انجام این عمل، انتقال اطلاعات زیر رخ خواهد داد:

1. Client----------FIN (856780) and ACK (758684759)----------→ Host

فریم ضبط شده از این انتقال اطلعات در زیر برای مطالعات بعدی شما قرار داده شده است (قسمت **Bold** شده در زیر، در حقیقت **Sequence Number** و مقدار **ACK** مربوط به بسته را نمایش می دهد):

20 53 52 43 00 00 44 45 53 54 00 00 08 00 45 00 00 28 C5 00 40 00 20 06 0E 10 CB 5E FD BA CB
5E F3 47 04 07 00 17 **00 0D 12 CC 2D 38 9C 57** 50 11 22 38 25 55 00 00

خوانندگان تیز بین، باید متوجه شده باشند که مقادیر هگزا (مبنای ۱۶) یعنی همان مقادیری که در فریم ذخیره شده بالا به صورت Bold **می باشند، مشابه مرحله سوم در پروسه در برقراری ارتباط هستند. به هر حال، تفاوت این را بیان می کند که در این مثال، بسته،** option **ها و گزینه های** FIN **و** ACK **را به صورت فعال در خود دارد** (FIN = Finish) **در حالی که در مرحله سوم برقراری ارتباط گزینه های** SYN **و** ACK **فعال هستند. بنابراین، در این مـــرحله، کلاینت یک بسته** FIN/ACK **را به** host **می فرستد.** Option **و گزینه** FIN **به** host **می گوید که در حقیقت کلاینت قصد پایان دادن این ارتباط که بین آنها برقرار است را دارد.** **Sequence Number در این بسته از مرحله سوم تغییر نمی کند، در نتیجه، در مرحله سوم تنها یک ACK Message را حمل کرده است که هیچ Sequence Number را مصرف نمی کـــند. اگرچه این بسته ی به خصوص، به راستی یک FIN Message را حمل می کند و مسلما Sequence Number را مصرف می کند، اما یک Sequence Number مربوط به یک قطـعه، مقداری را نمایش می دهد که در ابتدای آن قطعه مخصوص قابـــــــــل اجرا و اطـــــــــــــــلاق (Applicable) می باشد نه در انتهای آن.**

بنابراین، Sequence Number مصرف شده به وسیله گزینه FIN، مقدار Sequence Number مربوط به قطعه/بسته بعد را افزایش می دهد و در کل انگار هیچ تاثیری در بسته نگذارده است. مقدار ACK مربوط به بسته 758684759 **می باشد که** sequence **و توالی بسته که در کلاینت از** host **مورد انتظار است، نشان می دهد.**

مقدار ACK **یا** ACK Value **مربوط به این پیام** (message) **از آن مرحله سوم، تغییر نمی کند، بنابراین، گویا هیچ اطلاعاتی به وسیله کلاینت از** host **دریافت نشده است و کلاینت هنوز منتظر و چشم انتظار** host **برای فرستادن** data sequence **(توالی اطلاعات) با مقدار** 758684759 **می باشد.**

2. Host----------ACK (856781)----------→ Client

فریم ضبط شده از این انتقال اطلاعات برای مطالعات بعدی شما قرار داده شده است (قسمت هایی که به صورت Bold نمایش داده شده اند، Sequence Number و ACK Value مربوط به بسته را نمایش می دهند):

44 45 53 54 00 00 20 53 52 43 00 00 08 00 45 00 00 28 8F BE 40 00 39 06 2A 52 CB 5E F3 47 CB 5E
FD BA 00 17 04 07 **2D 38 9C 57 00 0D 12 CD** 50 10 83 2C C4 60 00 00

در اینجا، host **یک بسته به صورتی که** ACK Option **آن فعال باشد، می فرستد، که رسید اطلاعاتی که تا الان توسط کلاینت فرستاده شده را تایید می کند. در اینجا** ACK Value **برابر با** 856781 **می باشد. هـــــــــــــمچنین به خاطر داشته باشید که Sequence Number مربوط به این particular به عنوان 758684759 رفتار می کند که مشابه پیام بعدی فرستاده شده توسط host به کلاینت می باشد، بنابراین، این پیام مخصوص (particular) تنها یک ACK Option را حمل می کند که هیچ گونه Sequence Number را مصرف نمی کند.**

3. Host----------FIN (758684759) and ACK (856781)----------→ Client

فریم ضبط شده از این انتقال اطلاعات برای مطالعات بعدی شما قرار داده شده است (قسمت های Bold شده، در حقیقت

Sequence Number و ACK Value مربوط به این بسته را نشان می دهند):

44 45 53 54 00 00 20 53 52 43 00 00 08 00 45 00 00 28 8F E0 40 00 39 06 2A 30 CB 5E F3 47 CB 5E
FD BA 00 17 04 07 **2D 38 9C 57 00 0D 12 CD** 50 11 83 2C C4 5F 00 00

در این فریم ضبط شده، host **یک بسته** FIN/ACK **را به کلایـــــــــــنت با Sequence Number برابر با** 758684759 **می فرستد، که مشابه همان مرحله ی اولیه می باشد. حتی** ACK Number **نیز مانند قبل می باشد. در نتیجه** host **هنوز هیچ اطلاعاتی که به وسیله کلاینت فرستاده شده را دریافت نکرده است.**

4. Client----------ACK (758684760)----------→ Host

فریم ضبط شده از این انتقال اطلاعات برای مطالعات بعدی شما قرار داده شده است (قسمت های Bold شده در حقیقت Sequence Number و ACK Value **مربوط به این بسته را نمایش می دهند):**

20 53 52 43 00 00 44 45 53 54 00 00 08 00 45 00 00 28 C6 00 40 00 20 06 0D 10 CB 5E FD BA CB
5E F3 47 04 07 00 17 **00 0D 12 CD 2D 38 9C 58** 50 10 22 38 25 54 00 00

در اینجا، کلاینت به بسته FIN/ACK **که توسط** Host **فرستاده شده بود، با یک بسته** ACK **جواب می دهد که تمام اطلاعات رسیده تا الان را تصدیق می کند و بدین گونه ارتباط پایان می پذیرد. به خاطر داشته باشید که Sequence Number مربوط به این بسته به صورت** 856781 **تلقی شده است. بنابراین، از توضیحات ارائه شده در فوق، می توانیم نتیجه بگیریم که Sequence Number در موارد و حالات زیر افزایش می یابند:**

Transfer of FIN Packet 1
Transfer of SYN Packet 1
Transfer of ACK Packet 0
Transfer of SYN/ACK Packet 1
Transfer of FIN/ACK Packet 1
Passage of 1 second 128,000 – **گذشت زمان از ۱ تا ۱۲۸,۰۰۰**
Establishment of 1 connection 64,000 – **برقراری ارتباط، ۱ تا ۶۴,۰۰۰**

اگر قادر باشید که پیشگوئی هایی درباره Sequence Number ها انجام دهید، در این صورت موارد زیر برای شما بسیار آسان خواهد بود:

۱. انجام عملیات TCP Hijacking **و منحرف کردن اطلاعات**

۲.اکسپلویت کردن ارتباطات Trust **که با عنوان** Trust Relationships **شناخته شده اند.**

اکنون که به درستی مفهوم افزایش Sequence Number ها و ACK Number ها را فرا گرفتیم، اجازه دهید به بررسی آن بپردازیم که چگونه IP Spoofing **می تواند برای اکسپلویت کردن ارتباطات** trust **استفاده شود!**

Trust Relationship **ها یا ارتباطات اعتباری (و تصدیق شده)**

شاید تا حال با فرمی برای انجام پروسه ی اعتـــــــــبارسازی (Authentication) و ... روبرو شده باشید. دوتایی UserName-PassWord قالبی پراستفاده و محبوب برای انجام عملیات Authentication می باشد که با این مورد آشنایی کافی حتما دارید! چیزی که در فرم اعتبار سازی در قسمت Username-Password اتفاق می افتد این است که آن Remote Host که کلاینت به آن وصل شده است، کلاینت را با وارد کردن Username و Password به مبارزه می طلبد!!! بنابراین، در این فرم اعتبارسازی، کاربر احتیاج به درگـــیر شدن با Remote Host دارد و از طـــرفی Remote Host نیز او را به وارد کردن Username & Password دعوت کرده است:D ، که در نتیجه مفهوم فرم اعتبارسازی معنا پیدا می کند.

در کنار فرم اعتبارسازی به روش User/Pass (User/Password Form of Auth)، فرمی دیگری در رابطه با اعتبارسازی وجود دارد که بسیاری از کاربران از آن اطلاع ندارند. این نوع از اعتـــــــــبارسازی، Trust Relationship نامیده می شود (Trust Relationship of Auth.)، که در حقیقت IP Address مربوط به سیستم کلاینت، نقش و دلیل اعتبار خواهد بود. در این حال از اعتبارسازی، چیزی که اتفاق می افتد آن است که Remote Host، در حقیقت IP Address مربوط به سیستم کلاینت را به نوعی می گیرد (یا پیدا می کند) و آنرا با لیست از پیش تعریف شده که اجازه دسترسی را دارند، مقایسه می کند. اگر IP Address مربوط به سیستم کلاینتی که در حال تلاش برای برقراری ارتباط با host می باشد، در لیست تعیین شده در host برای اجازه دسترسی، پیدا شد، آنگاه کلاینت می تواند به shell بدون هیچ گونه پسوردی دست پیدا کند چرا که هویت کلاینت قبلا تصدیق شده است (استفاده از مفهوم کلی Trust Relationship).

چنین گونه های trust relationship بیشتر در سیستم های یونیکس استفاده می شود که سرویس های R (R Services) مشخصی دارند مثل rsh, rlogin, rcp و ... و این سرویس ها مشکلات مشخصی دارند که باید از آنها اجتناب شود. با وجود چنین مشکلی، هنوز بسیاری از ISP ها، پورت های مربوط به R Service ها را باز می گذارند تا به راحتی توسط هکرها اکسپلویت شود!! با استفاده از دستور Unix Shell زیر، می توان یک ارتباط rlogin با یک remote host به سادگی برقرار کرد:

$>rlogin IP address

توجه: قطعا راه راحت تری برای برقراری یک Trust Relationship با یک Remote Host وجود دارد مثل: استفاده از telnet و ... . ضمن اینکه باید گفت: پورت پیش فرض (default) که R Service ها بر روی آن عمل می کنند معمولا 512, 513, 514 هستند.

اما مهم اینجاست که اگر یک Trust Relationship بین دو سیستم (مثلا یک کلاینت و یک سرور) وجود داشته باشد و شما قادر باشید که IP Address خود را تغییر دهید و وانمود کنید که کلاینت هستید، آن وقت شما در حقیقت می توانید خودتان را برای استفاده از این هویت جدید و spoof شده، اعتبار ببخشید و یک ارتباط مناسب با سرور برقرار کنید. یک چنین ارتباطی، شاید شما را به همه دستورات و همه قسمت ها در سرور برساند و در حقیقت به شما حق چنین کاری را بدهد. بنابراین، می توان با گفتن این مطلب نتیجه گرفت که اگر شخصی قادر باشد IP Address سیستمی دیگر را Spoof کند و از این هویت spoof شده برای اکسپلویت کردن رابطه Trust Relationship که بین دو سیستم در قبل وجود داشته، استفاده نماید، آن وقت نتایج مهمی در بر خواهد داشت.

Spoof کردن IP Address خود به منظور اکسپلویت کردن Trust Relationship ها

قبل از اینکه به مراحل گوناگونی که باید به منظور Spoof کردن IP Address خود انجام دهیم، بپردازیم و به exploit کردن Trust Relationship ها بپردازیم، مهم است که درک کنید انجام عملیات IP Spoofing نسبتا مشکل بوده و همچنین خیلی پیچیده می باشد. چیزی که آنرا سخت کرده اولا این است که در واقع یک حمله کورکورانه می باشد (قبلا مفهوم توضیح داده شده) و دلیل دیگر اینکه مبنی بر مفروضیات زیادی است که باید در طرف نفوذگر اعمال شوند. بنابراین، شاید فردی حتی نتواند با وجود چندین بار انجام

این عمل آنرا با موفقیت انجام دهد. به هر حال با تمرین کردن (و البته نا امید نشدن) می توان مقداری این فرآیند را نتیجه بخش تر کرد. هر چند، اجرای این عملیات غیر ممکن نیست و می توان گفت قبلا به نوعی این روش منسوخ شده است!!!

مراحل گوناگونی که در Spoof کردن یک IP Address و نیز در اکسپلویت کردن یک Trust Relationship دخالت دارند، عوامل زیر می باشند.

توجه: به خاطر داشته باشید که به خاطر انجام مراحل زیر، ما باید از ۳ سیستم استفاده کنیم:

الف)- سیستم قربانی یا Victim System – که در حقیقت به سیستم قربانی (VICTIM) یا سیستم هدف (TARGET)رجوع دارد.

ب)- سیستم اعتماد شده یا Trusted System – که توانایی برقراری یک Trust Relationship را با VICTIM دارد. این سیستم را با نماد TRUSTED می شناسیم.

ج)- سیستم نفوذگر یا Attacking System – که آنرا با نماد ATTACKER شناسائی می کنیم.

۱. پیدا کردن یک سیستم اعتماد شده یا Trusted System

اولین گام در اکسپلویت کردن Trust Relationship ها، در حقیقت این است که بفهمیم آیا سیستم مقابل در یک Trust Relationship حضور دارد یا خیر. در این مرحله، کار به خصوصی انجام نمی شود فقط در این مرحله باید فهمید که کدام کامپیوتر سیستم هدف (TARGET) را trust می کند و معمولا یک Trust Relationship بر اساس اعتباری سازی های انجام شده توسط IP Address ها، ایجاد می کند. در نتیجه، ما می خواهیم trusted سیستم را کشف کنیم، سیستمی که به وسیله target system اجازه برقراری ارتباط remote trust را با خودش یافته است یعنی سیستمی که با عنوان TRUSTED مشخص شده است.

Trusted system یک ارتباط trust را با target system به وسیله راه های شناخته شده ای مانند R Service ها برقرار می کند. به وسیله هر یک از روش های ذیل می توانیم بفهمیم که target system با کدام سیستم، یک trust relationship برقرار می کند:

الف)-به وسیله دستورات گوناگونی مثل (این متد بیشتر استفاده می شود):

```
rpcinfo -p
showmount -e
```

ب)-مهندسی اجتماعی: می توان به وسیله این روش اطلاعات زیادی را در مورد **target system** و حتی **network** آن به دست آورد.

ج)-روش های **Brute Force**: که در آن، تمامی سیستم های موجود در همان شبکه چک می شوند که آیا توانایی این را دارند که با **target system** یک **Trust Relationship** برقرار کنند یا خیر. اما این روش، بسیار خسته کننده و کند انجام می شود.

هنگامی که **trusted system** را کشف کردید که **target system** با آن یک ارتباط **Trust** برقرار کرده است، کاری که باید انجام شود این است که: **Trusted System** را **DoS** کنید، در نتیجه، این سیستم بی مصرف و بلااستفاده خواهد بود. سپس، مطمئن شوید که ارتباطی که به صورت **spoof** شده نصیب ما شده است را با **target system** قطع نکند. مراحل را می توان به صورت زیر، سمبولیک نشان داد:

Target System <==== <u>DoS Attack</u>
Spoofed Connection (gathered) <==== <u>Must not Interrupt</u>

۲. مسدود (block) کردن TRUSTED یا Trusted System

این گام از مراحل، بسیار مهم می باشد. به محض اینکه Trusted System را کشف کردید، مرحله بعدی شما این است که به سرعت آنرا Block یا Disable کنید. اگر به خاطر می آورید، در قبل حول آن مشکل که فردی سعی در انجام عملیات IP Spoofing بود، بحث کردیم. بعد از یک توضیح مفصل، به برخی از ضروریات برای اینکه عملیات IP Spoofing با موفقیت انجام شود، رسیدیم. در بین آنها، یکی از آنها مورد زیر بود (جمله زیر را می توانید مشابها از توضیحات گفته شده در صفحات قبل پیدا کنید):

"....FAKE نباید هیچ گونه جوابی به بسته SAN/ACK که VICTIM برای او می فرستد، بدهد..."

در نتیجه، برای محقق شدن کلام فوق یعنی برای اطمینان حاصل کردن از این که trusted system به target system جوابی ندهد، باید به نوعی آنرا block کرد یعنی باید مطمئن شویم که تمامی حافظه (memory) مربوط به trusted system مورد استفاده قرار گرفته است و در نتیجه او قادر نخواهد بود که به بسته SYN/ACK که برای او فرستاده شده است (از طرف target system = VICTIM) جواب بدهد. یک راه بسیار راحت برای محقق شدن فرآیند فوق (یعنی blocking)، انجام عملیات SYN Floodig DoS Attack می باشد. هنگامی که ما بسته های SYN Flood را به طرف TRUSTED فرستادیم و تمامی حافظه قابل دسترس روی آنرا به نوعی اشغال کردیم، آنوقت احتمالا می توانیم مطمئن باشیم که به هیچ یک از بسته های SYN/ACK که از طرف TARGET ارسال می شود، جواب نخواهد داد. بنابراین، هنگامی که احساس کردیم سیستم تحت حمله SYN Flood قرار دارد (TARGET=Blocked)، آنوقت، می توانیم راجع به بسیاری از مشکلاتی که در حین IP Spoofing برای ما پیش می آید، خاطر جمع باشیم (منظور همان مشکلِ به میان آمدن Trusted System و پایان دادن ارتباط و Connection که به صورت Spoof شده به دست آوردیم).

۳.به دست آوردن آخرین Sequence Number و پیش گویی در مورد آنها

هنگامی که trusted system به نوعی Disable شد، آنوقت نفوذگر باید Sequence Number مربوط به Target System را به دست آورد. این مرحله، مرحله ای است که نفوذگر مجبور است محاسبات و مفروضیات زیادی را پیش خود به عمل آورد. به منظور به دست آوردن Sequence Number مربوط به Target System، نفوذگر فقط باید فورا (قبل از اقدام به حمله) به پورت 23 یا 25 وصل شده و Sequence Number مربوط به آخرین بسته ای که توسط target system فرستاده شده است را ضبط (record) کند. مصحلت آن است که این مرحله را چندین بار تکرار کنیم و در هر بار Sequence Number را ضبط کنیم. نفوذگر همچنین باید Round Trip Time یا RTT را به وسیله ابزاری مثل icmptime، دریافت کند (در صورت نیاز می توانید به مقاله های مربوط به ICMP مراجعه کنید). برای گرفتن یک مقدار دقیق و صحیح از RTT، باید چندین بار یک بسته را بفرستید و زمان را record کنید (در آخر هم اگر مقدارهای گوناگونی به دست آمد، صرف نظر از سرعت و پهنای باند و مسائل حاشیه ای، می توانید میانگین آنها را به عنوان RTT در نظر بگیرید).

توجه: RTT یا Round Trip Time در حقیقت زمانی است که یک بسته برای حرکت و سیر از مبدا به مقصد و برگشت از آن مسیر، لازم دارد. بنابراین، فاصله زمانی که یک بسته برای رسیدن از مبدا به مقصد می خواهد، می تواند به وسیله تقسیم کردن مقدار RTT بر ۲ به دست آید (تقسیم بر دو به خاطر آن است که RTT زمان برگشت را هم حساب می کند). یعنی: RTT/2

هنگامی که نفوذگر آخرین Sequence Number مربوط به target system را به دست آورد، آنگاه بر اساس توضیح بالا، مقدار RTT، مقدار گذشت زمان (بینِ ضبط کردنِ آخرین Sequence Number و اجرای واقعی حمله) و دیگر موارد را محاسبه می کند. هنگامی که یک افزایش از Sequence Number می تواند وجود داشته باشد، در حقیقت نفوذگر نیز می تواند

Sequence Number بعدی را پیش گویی کند!!!

برای اجرای موثرِ این مرحله، کارهای زیر باید انجام شوند:

۱. ضبط کردن RTT یا Route Trip Time و گردش مدت زمانی که یک بسته باید از target به سیستم شما قبل از مراجعه به خود، بیاید. بنابراین، آن زمان در حین اجرای واقعی حمله ذخیره می شود.

۲.بعد از اینکه شما آخرین Sequence Number مربوط به سرور را log برداری کردید، به سرعت (به هیچ وجه زمان را هدر ندهید)، Sequence Number بعدی را با سرعت هر چه تمام تر محاسبه کنید و به سرعت به اجرای واقعی مربوط به حمله بروید، چرا که در صورتی که وقت زیادی را هدر دهید، شاید سیستم دیگری در همین حـــــین به target system وصل شود و مسلما مقدار Sequence Number آن بیشتر خواهد شد و در نتیجه Sequence Number که شما در این مرحله می خواهید حدس بزنید، 64,000 تا بیشتر از مقدار واقعی خواهد بود!!!

۳. قبل از اینکه شروع به IP Spoofing **کنید، جدول** Case-Increment **را یاد بگیرید.**

۴. البته در انجام این گونه حرکات باید به شدت سریع بود و از حضور ذهن زیادی برخوردار بود.

هنگامی که شما در حال پیش گویی کردن Sequence Number بعدی هستید، تصورا، مقدار پیش گویی شده ی شما، باید مشابه Sequence Number مربوط به Target System که در حال حـــــــــــــاضر می باشد، باشد. هرچند، حتی اگر Sequence Number که شما پیش گویی کردید، به مقدار زیادی از Sequence Number بعدی زیاد نباشد (به صورت تقریبی به مقدار واقعی نزدیک باشد)، آن وقت، Target System آنرا در صف قرار خواهد داد (queue up) و آنرا در استفاده های بعد مورد استفاده قرار خواهد داد. بیائید فرض کنیم که نفوذگر واقعا Sequence Number بعدی را به درستی حدس زده است و در نتیجه باید برای انجام مرحله بعد آماده باشد.

۴. آغاز حمله واقعی:

هنگامی که شما قادر به پیش گویی Sequence Number بعدی شدید، سپس، وقت اجرای حمله واقعی بر طبق مراحل زیر فرا رسیده است:

۱. ATTACKER یک بسته SYN را که IP Address به صورت Spoof شده است (که در حقیقت IP Address مربوط به سیستم TRUSTED یا همان Trusted System می باشد) را به سیستم VICTIM می فرستد. این بسته ی SYN برای پورت rlogin که شماره ی آن ۵۱۲ است، آدرس گذاری شده است و ضمنا خواستار (درخواست) برقراری یک ارتباط trust بین VICTIM و Trusted System می باشد.

۲. VICTIM در مقابل این بسته عکس العمل نشان می دهد و این بسته را با فرستادن یک بسته ی SYN/ACK که به سمت trusted system آدرس گذاری شده است، جواب می دهد. اگر تـــــمامی حافـــــــــــظه مربوط به trusted system به وسیله SYN Flooding اشغال نشده باشد، آنوقت می تواند در برابر این بسته SYN/ACK عکس العمل نشان داده و آنرا با یک بسته

NACK جواب داده و در نهایت تمامی فعالیت ها و تلاش های نفوذگر برای انجام عـــــــــــملیات IP Spoofing و Spoof کردن IP Address سیستم خود، بی نتیجه خواهد ماند.. به هر حال، در این مثال، TRUSTED به نوعی (DoS, SYN Flood, ...) مسدود و Disable شده است. بنابراین، قادر TRUSTED قادر نخواهد بود که به بسته ی SYN/ACK که به وســـیله VICTIM ارسال می شود، پاسخ دهد. در نتیجه، این بسته ی SYN/ACK که توسط VICTIM ارسال شده، حذف شده و دور انداخته شده یا به اصطلاح Discard می شود.

۳. بعد از گذشت زمان مشخصی (زمانی بسیار کوتاه) یعنی هنگامی که نفوذگر مطمئن شد که VICTIM دیگر باید بسته SYN/ACK را به TRUSTED فرستاده باشد، آن وقت او (نفوذگر) یک ACK را به VICTIM می فرستد. این پیام ACK یا ACK Message طوری طراحی می شود (از قبیل یک IP Address که به صورت Spoof شده در این بسته طراحی می شود. که IP Address مربوط به سیستم TRUSTED خواهد بود) که به نظر آید، در حقیقت از TRUSTED ارسال شده است.

نکته: اما نکته مهم اینجاست که حتما باید مطمئن شوید که این بـــــــــسته ی ACK که می فرستید شـــــــــماره ی اعتبار یا Acknowledgement Number آن برابر Sequence Number ای هست که در مرحله قبل حدس زدید + ۱ !! (به هر حال توضیحات این مورد مفصل داده شده است) یعنی:

Sequence Number + 1

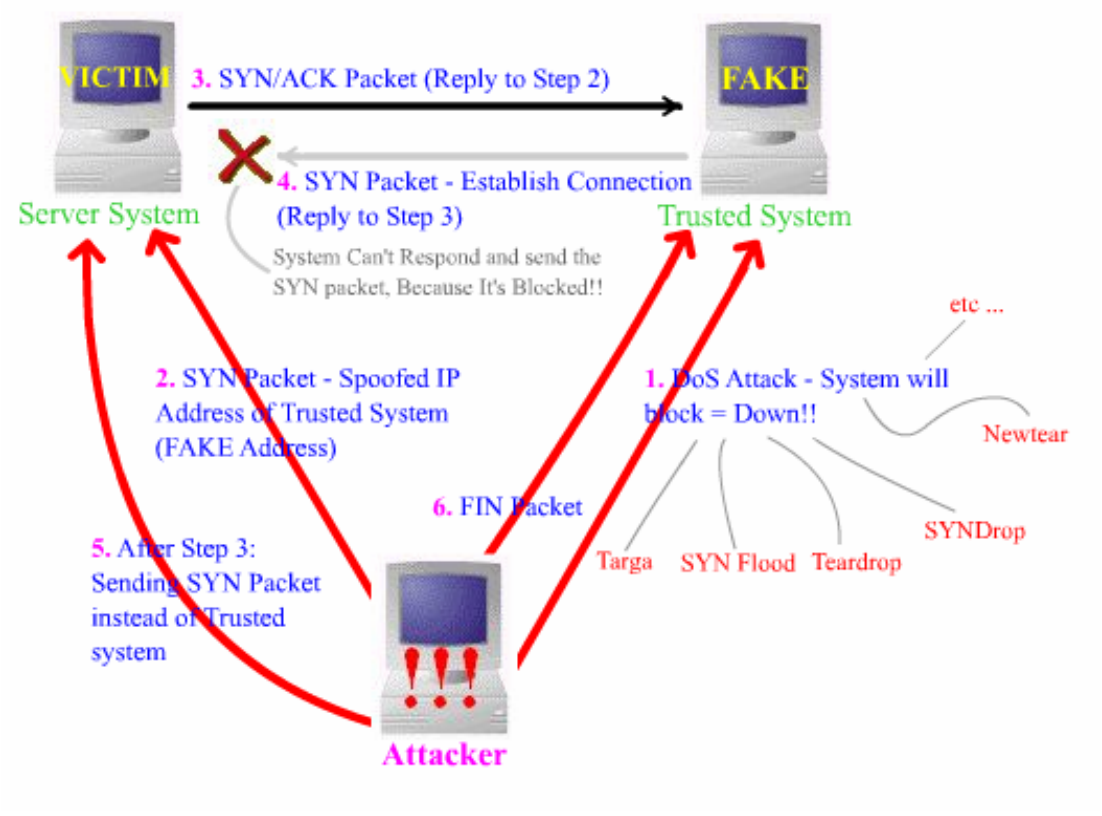
اگر همه چیز بر اساس نقشه ای باشد که طراحی شد و اگر نفوذگر Sequence Number را به درستی حدس زده باشد، آنوقت سیستم مقابل یعنی VICTIM این ارتباط را می پذیرد و یک Trust Relationship بین شما و VICTIM برقرار می شود !!!

۵. بیرون آوردن TRUSTED از DoS Attack

سرانجام، بعد از اینکه حمله با موفقیت انجام شد و از ارتباط با VICTIM استفاده لازم را بردید، وقـــت آن است که Trusted System یا TRUSTED را از حالت DoS Attack بیرون بیاوریم. البته این کار می تواند اختیاری انجام شود ولی اگر خواستید که کار بدون کوچکترین ردپا و شکی از سوی VICTIM و TRUSTED انجام شود بهتر است که TRUSTED را نیز از DoS Attacked در بیاورید. این کار به وسیله فـــــــرستادن بـــــــــسته هایی که گزینه ی FIN آنها به معنی Final (یعنی Final Connection) روشن است، انجام می شود. به صورت سمبولیک نشان می دهیم:

Putting Out TRUSTED (of DoS Attack) ======> Packet with FIN Option On ;)!!!

عکس زیر مراحل انجام حمله را نشان می دهد:



مراحل عملیات IP Spoofing

اقدام متقابل

در زیر به بررسی چند مورد برای جلوگیری از این نوع سو استفاده ها می پردازیم:

۱. باید از استفاده از trust relationship ها که در حقیقت بر اساس IP Address های مربوط به کلاینت ها استفاده می کنند، دوری کنید. بلکه می توان از دوتایی UserName-PassWord به عنوان یک ابزار اعتبار و تصدیق سازی (Authentication) استفاده کرد. بنابراین، تا آنجا که ممکن است از ارتباطات به نوع Trust یا به اصطلاح Trust Connection ها استفاده نکنید. از TCP Wrapper ها برای تعیین اجازه دستیابی برای سیستم های مشخصی که به صورت Good Countermeasure می باشند، استفاده کنید.

۲. یکی از راحت ترین روش ها برای جنگ با IP Spoofing استفاده از یک Firewall یا قواعد فیلترینگ (Filtering Rule ها) می باشد که تمامی بسته های خارج از شبکه را filter می کنند. اما به هر حال اجازه ورود به بسته هایی که با ساختار داخل شبکه می باشند، را می دهند. همچنین، باید قاعده ای باشد که بسته های خارجی (outgoing) که Source Address آنها متفاوت از شبکه داخلی است، فیلتر کند. این کار در واقع چشمه و منبع یک حمله IP Spoofing را از بین می برد و تمامی IP Spoofing ها که ممکن است در شبکه داخلی رخ دهد را کنترل و اجتناب می کند. مورد بعدی باید از بسته های ورودی (incoming) که از خارج از شبکه می آیند ولی یک Source IP دارند و متعلق به شبکه هستند، جلوگیری کند. به وسیله ACL زیر در مسیریاب می توانید این کار را انجام دهید. تنها کافی است ACL زیر را به router ACL اضافه کنید:

```
access-list 101 deny ip 201.94.0.0 0.0.255.255 0.0.0.0 255.255.255.255
```

این ACL به خصوص، در یک شبکه که یک Source Address درونی از 201.94.xx.xx دارد، قابل اجرا می باشد. اگر می خواهید شبکه خود را تنها از حملات Source IP Spoofing که ممکن است از داخل شبکه سرچشمه بگیرد، حفظ کنید، ACL زیر این کار را انجام می دهد:

```
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```

۳. استفاده از یک پروتکل رمزنگاری و امن مانند IP Security که به اختصار IPSEC نامیده می شود (که البته هنوز عمومی و فراگیر نشده است و علل خاص خود را دارد).

۴. با استفاده از Random Initial Sequence Number (RISN): که در حقیقت این کار Sequence Number ها را طوری (به صورت تصادفی) قرار می دهد که قابل پیش گویی نباشند. چنین راه حلی را می توانید با استفاده از pseudorandom-number generators ها یا PRNG ها انجام دهید. اگرچه، PRNG ها نیز، هنگامیکه sequence number ها به صورت تصادفی انتخاب شده باشند، بعضی مواقع برای علامت گذاری یا Marking (MARK) مشکل دارند.

Written by: Cephexin