به نام خدا

آشنایی و مقابله با بعضی از روش های Crack نرم افزار

"اگر میخواهید نرم افزار شما شکسته نشود، هرگز برنامه ننویسید"

حتما تا حالا این جمله و مطالب نظیر این را بار ها و بارها شنیده اید. این مطالب تا حدود زیادی حقیقت دارد و می توان گفت همه نرم افزار ها به نوعی قابل شکستن میباشند. اما باید در نظر داشت برای شکستن هر برنامه مدت زمان خاصی نیاز است که طولانی تر شدن این زمان ممکن است Cracker را خسته کرده و از Crack نرم افزار منصرف کند. در واقع هدف از تهیه این مطالب ارائه روش هایی برای طولانی کردن این زمان و سخت تر کردن کار Cracker هاست.

رشته ها و پیغام های برنامه

به طور کلی Cracker ها برای شروع به کار پس از اجرای برنامه تمام پیغام ها را بررسی میکنند و از پیغام های موجود در برنامه برای پیدا کردن قسمت هایی از برنامه که شرط های اصلی شما قرار دارند استفاده میکنند. و اگر این شرط ها به سادگی استفاده شده باشند با یک عملیات جزئی می توانند شرط شما را برعکس کرده و به هدف خود برسند. و یا به Serial Number و یا کلمه عبور اصلی برنامه دسترسی پیدا کنند. به طور مثال

یک برنامه ساده ای را در نظر بگیرید که برای اجرا شدن به یک کلمه عبور 8 کاراکتری نیاز دارد و در صورتی که کلمه عبور به درستی وارد شود برنامه به درستی اجرا شده و در غیر این صورت برنامه پیغامی مبنی بر اینکه کلمه عبور اشتباه است را نمایش میدهد. Cracker با استفاده از Debugger هایی که در اختیار دارد برنامه شما را باز کرده و در ابتدا به دنبال رشته ی " کلمه عبور اشتباه است" می گردد. با پیدا کردن این رشته اولین شرطی که قبل از این پیغام وجود دارد را برعکس کرده و برنامه را ذخیره می کند. حال زمانی که کلمه عبور به اشتباه وارد شود برنامه اجرا خواهد شد.

برای جلوگیری از این مشکل راه حل های زیادی وجود دارد که ما در اینجا به بررسی بعضی از این روش ها می پردازیم :

1- هرگز رشته ها و پیغام های برنامه را به صورت مستقیم در کد قرار ندهید. و حتی المقدور از رشته های کد شده در برنامه استفاده کنید.

2- هرگز بلافاصله بعد از چک شرط ها عملیات مربوطه را قرار ندهید. به طور مثال شما میتوانید در صورتی که شرط نادرست بود متغییر دیگری را مقدار دهی کنید و در جایی دیگر مقدار این متغییر را چک کنید. در صورتی که متغییر حاوی مقدار مورد نظر بود عملیات نهایی انجام شود. و یا میتوان در صورت نادرست بودن شرط ، متغییر های اصلی برنامه را با مقادیر نا معتبر مقدار دهی نمود. به طوری که در روند اجرای برنامه اخلال ایجاد شود.

3- هرگز کلمه عبور را به صورت یک رشته صحیح در برنامه قرار ندهید. مثلا شما می توانید کلمه عبور کد شده را در برنامه قرار داده و رشته ورودی که کاربر وارد می کند را کد کنید و سپس با رشته داخل برنامه مقایسه کنید. برای مثال فرض کنید تصمیم دارید اگر کاربر کلمه عبور 1234 را وارد کرد برنامه شما فعال شود. برای این منظور در ابتدا یک الگوریتم Coding پیاده سازی کرده و 1234 را توسط این الگوریتم کد می کنید، مثلا در اینجا همه رقمها +2 می شود. حال در محیط برنامه نویسی ابتدا عدد ورودی را با الگوریتم مورد نظر کد کرده و سپس با مقدار ثابت 3456 مقایسه کنید. ***هرگز مقدار ورودی را با مقدار واقعی کلمه عبور(در اینجا 1234) مقایسه نکنید.***

محاسبه CRC قسمتی از کد

معمولا در هر برنامه توابعی وجود دارد که عملیات مهمی در آنها انجام میگیرد. میتوان در قسمت های مختلف برنامه با استفاده از الگوریتم هایی مثل CRC32 صحت آن تابع را چک نمود. برای این منظور ابتدا آدرس شروع و پایان تابع را محاسبه نموده و مقدار عددی CRC آن قسمت از کد را محاسبه میکنیم. سپس مقدار بدست آمده را با مقدار واقعی مقایسه می کنیم. با این عمل در صورتی که Cracker کوچکترین تغییری در این قسمت کد ایجاد کند برنامه متوقف خواهد شد.

برنامه Demo

یکی دیگر از مواردی که به Cracker ها برای Crack نرم افزار های شما کمک می کند تولید نادرست برنامه های Demo است. بسیاری از تولید کنندگان نرم افزار با تغییراتی که بر روی نسخه نهایی اعمال میکنند مثل Disable کردن Form ها و Button ها و ...، اقدام به تهیه نسخه Demo میکنند.

در اینجا باید بدانید Cracker ها ابزار های بسیار قوی ای در دسترس دارند که توسط این ابزارها به راحتی میتوانند کنترل Resource برنامه شما را به دست گرفته و تمام کنترل های شما راEnable کنند. برای جلوگیری از رخ دادن چنین مشکلاتی بهتر است:

1- از Hide و یا Disable کردن کنترل ها بپرهیزید و سعی کنید اگر نیازی به آنها در برنامه Demo نیست آنها را حذف نمایید.

2- تمام توابع و متغییر هایی که در برنامه Demo قرار است غیر فعال باشند را از سورس پروژه کاملا حذف نمایید.

Thread و Timer

امکان دیگری که ابزار ها در اختیار Cracker ها قرار میدهند امکان از کار انداختن Timer ها و Thread ها ی برنامه است که کاری بسیار ساده می باشد پس به خاطر داشته باشید ، برای چک قفل برنامه هرگز **فقط** به Thread ها و Timer اکتفا نکنید و

در صورت استفاده از Thread ها و Timer ها سعی کنید بین آنها و exe اصلی به نوعی وابستگی ایجاد کنید.به نحوی که در صورت از کار انداختن Thread یا Timer ها ، برنامه شما به نوعی Crash کرده و روند اجرای exe مختل شود.

استفاده از قفل در برنامه

یکی از رایج ترین روش هایی که در حال حاضر برنامه نویسان برای بالا بردن امنیت برنامه هایشان استفاه میکنند بکار گیری انواع قفل ها است. در حال جاضر قفل هایی تحت عنوان قفل های نرم افزاری و سخت افزاری برای این منظور وجود دارد. اما استفاده از این قفل ها نیز**به تنهایی** نمی تواند امنیت نرم افزار شما را تامین کند. این قفل ها در واقع ابزاری هستند که نحوه به کار گیری این ابزار ها بسیار اهمیت دارد. اولین نکته ای که هنگام انتخاب این ابزار ها باید در نظر داشت قدرت مانوری است که این قفل ها به شما میدهند. هر چه این ویژگی بارز ترو دست برنامه نویس برای بکار بردن روش های متفاوت باز تر باشد آن قفل مفید ترخواهد بود. به عنوان مثال قفل های سخت افزاری فضایی را روی قفل به شما اختصاص میدهند که میتوانید مقادیر مورد نظر خود را روی آن نوشته و یا از آن بخوانید. حال استفاده از این فضا بستگی به خلاقیت برنامه نویس و نحوه مدیریت آن دارد. ساده ترین روش استفاده از این نوع قفل ها قرار دادن مقادیر ثابت روی قفل و چک این مقادیر در هنگام اجری برنامه است البته نرم افزار هایی که به این سادگی از این نوع قفل ها استفاده میکند با نرم افزار های بدون قفل از نظر Cracker ها فرق چندانی ندارند. اما مواردی وجود دارد که با بکار گیری آنها ارزش واقعی این نوع قفل ها مشخص شده وامنیت نرم افزار ها تا حد زیادی بیشتر میشود.

یکی از موارد مهمی که هنگام استفاده از قفل سخت افزاری باید مد نظر داشته باشید قرار دادن Data های مهم برنامه در داخل قفل است. ***شما می توانید اطلاعات مهم برنامه مثل پسورد Database و .. را به عنوان قسمتی از data قفل در نظر بگیرید*** و زمان Connect به Database مقدار پسورد را از قفل بخوانید و در برنامه قرار دهید. در این صورت Crack برنامه بدون وجود قفل تقریبا غیر ممکن میشود و Cracker برای کار روی برنامه نیاز دارد که حتما حداقل یک بار برنامه را با قفل اجرا کند.

مورد قابل توجه دیگر تعداد دفعات چک قفل در برنامه است. ***هر چه تعداد چک قفل در داخل برنامه بیشتر باشد امنیت برنامه نیز به مراتب بیشتر خواهد بود.*** بهتر است چک قفل در قسمت های مهم برنامه مثل زمان گزارشگیری و یا پرینت اطلاعات و ... حتما قرار بگیرد. البته چک قفل می تواند به صورت random باشد. به عنوان مثال شما می توانید یک عدد تصادفی Generate کنید و اگر آن عدد برابر عدد خاصی بود قفل چک شود. با این عملیات اگر Cracker بتواند با trace مکان هایی که قفل چک میشود را بردارد باز برنامه اجرا نخواهد شد زیرا امکان دارد در اجرای بعدی برنامه عدد Generate شده با عدد مورد نظر برابر شده و قفل چک شود.

خطر بزرگ دیگری که برنامه های شما را تهدید میکند استفاده از توابع برای چک قفل است. در صورتی که شما تمام روتین های چک قفل را در یک تابع قرار دهید و هر بار برای چک قفل فقط آن تابع را call کنید به Cracker ها کمک بزرگی کرده اید. زیرا در این صورت فقط کافیست این تابع پیدا شده و محتویات داخل تابع فقط برای یک بار تغییر کند. مثلا تمام توابع چک قفل از داخل تابع حذف شود. در این هنگام حتی اگر شما 1000 جای مختلف وجود قفل را چک کرده باشید هیچ فایده ای ندارد و با یک تغییر هیچ یک از توابع چک قفل شما کار نخواهد کرد. ***پس بهتر است برای چک قفل هرگز از تابع استفاده نشود وهر بار تمام روتین ها به صورت مستقیم فراخوانی شوند.***

استفاده از EXE Protector ها

همان طور که در بالا ذکر شد Cracker ها از ابزارهای فوق العاده قوی ای استفاده می کنند و در زمان خیلی اندک میتوانند exe شما را با دیباگر ها باز کنند و به Trace فایل exe شما بپردازند. و حتی مکان های چک قفل شما را پیدا کرده و به طور کامل آنها را حذف کنند. درهمین راستا برنامه نویسان اقدام به تولید ابزار هایی کردند که بتواند exe ها را در مقابل دیباگر ها و disassembler ها محافظت کند.

کار اصلی اینگونه نرم افزار ها اضافه کردن امکاناتی به exe اصلی است که باعث جلوگیری از باز شدن سورس exe شما در دیباگر ها و disassembler هاست. در واقع

این نرم افزار ها شامل کد های Anti Debug و Anti Trace میباشند و زمانی که برنامه شما با این ابزار ها pack می شوند بسته به قدرت این ابزار تا حد زیادی از باز شدن exe شما توسط ابزار های Crack جلوگیری می شود. همچنین بعضی از این ابزار ها شامل قابلیت هایی مثل Anti Attach و Anti Dump میباشند که از کپی exe شما از حافظه جلوگیری می کنند.

پس حتما بعد از قفل گذاری ، برنامه خود را با یک EXE Protector قوی پروتکت کنید.

با توجه به مطالب ذکر شده برنامه نویسان میتوانند مدت زمان Crack برنامه را طولانی تر کرده و کار را برای Cracker ها سخت و سخت تر کنند.

Diamond_sh
Mail: diamond.sh.vc@gmail.com
With special thanks from UNREAL team
2008/2/6