



محدودیت در تعداد کپی (Copy Limited)

در این حالت برنامه نصب کننده نرم افزار، فضای مشخصی در دیسک را با روش خاصی فرمت کرده و تعداد مجاز نسخه برداری را در آن درج می‌کند. بدین طریق با هر بار کپی کردن برنامه، یک واحد از این عدد کم می‌شود و هنگامی که تعداد مجاز آن به صفر رسید، دیگر نمی‌توان برنامه را بر روی سیستم نصب نمود. حال ممکن است این سوال مطرح شود که مگر نمی‌توان پس از نصب برنامه، از آن پشتیبان (Back up) گرفته و سپس از نسخه پشتیبان نیز، بر روی سیستم دیگری استفاده نمود؟ پاسخ منفی است. زیرا هنگام نصب، اطلاعاتی راجع به سخت افزار سیستم که می‌تواند مثلاً شامل نوع قطعات و یا شماره سریال قطعات باشد، در جایی، در محدوده قفل ذخیره می‌شود و از این پس هر بار در هنگام اجرای برنامه، این اطلاعات به دقت چک می‌شود و در صورت هر گونه

تغییر، برنامه اجرا نمی‌شود.
استفاده از دیسکت، در هنگام برنامه (Required Disk)

در این حالت، دیسکت مورد نظر، یا به روش خاصی فرمت می‌شود و سپس در هنگام اجرا، اطلاعات روی آن بررسی می‌شود و یا اینکه قسمتی از دیسکت را بصورت فیزیکی و عمدی خراب می‌کنند و در اینجا، در واقع همان صدمه‌ای که به عمد، بر سطح دیسکت وارد شده است، به عنوان قفل و محافظ نرم افزار عمل می‌کند. از این پس برای انتقال برنامه از یک سیستم به سیستم دیگر، این فلاپی مانند قفل سخت افزاری عمل می‌کند و می‌بایست مختصات آن توسط برنامه تایید شود و چنانچه این فلاپی در درایو نباشد، برنامه اجرا نخواهد شد.

آشنایی با نحوه قفل گذاری بر روی یک برنامه

الف: طراح به سورس برنامه دسترسی دارد. در این حالت طراح پس از انتخاب روش قفل گذاری، کفایت آن را به زبان مورد نظر خود پیاده سازی نموده و در برنامه خود بگنجانند.

ب: طراح (مجری پروژه) به سورس برنامه دسترسی ندارد. گاهی اوقات به یکسری برنامه‌های ارزشمندی برخورد می‌کنیم که فاقد قفل هستند، بنابراین نیاز به قفل گذاری وجود دارد (البته این حالت بیشتر در کشور ما و چند کشور دیگر که در آن ها قانون Copyright معنی ندارد، کاربرد دارد). جهت تزریق قفل به این گونه برنامه‌ها، نیاز به آشنایی کامل به ساختار فایل‌های اجرایی (EXEY, COM, SYS) وجود دارد چرا که باید برنامه‌ای را طراحی کنیم تا همانند یک ویروس کامپیوتری به فایل اجرایی مشخصی بچسبد. البته جهت اینکار بهترین زبان برنامه نویسی، اسمبلی می‌باشد (بدلیل توانایی دخالت در روند اجرای برنامه). ضمناً برای بالا بردن سطح امنیت برنامه لازم است تا یکسری کدهای ضد دیباگ در برنامه گنجانده شوند.

کدهای ضد دیباگ، دستوراتی به زبان اسمبلی هستند که در حالت اجرای عادی برنامه، هیچ تغییری در روند اجرائی نمی‌گذارند بلکه در صورتی که برنامه توسط دیباگرها اجرا گردد (مورد ارزیابی قرار گیرد) نتواند از اجراء آن، جلوگیری نماید. با اضافه کردن کدهای ضد دیباگ به ابتدای برنامه (یا قبل از کنترل قفل) می‌توان احتمال دستکاری در برنامه را پایین آورد.

1- قفل گذاری با استفاده از شماره سریال اصلی دیسکت

همانطور که می‌دانید، سیستم عامل جهت هر دیسکت یک شماره سریال واحد (UNIQUE) اختصاص می‌دهد، بطوریکه شماره سریال هر دو دیسکت با هم یکی نیستند. بنابراین همین خود یک راه تشخیص دیسکت کلید (قفل) می‌باشد. جهت استفاده از این قفل می‌بایست شماره سریال دیسکت را خوانده و سپس در داخل برنامه آنرا کنترل نمائیم. یک راه ساده جهت خواندن شماره سریال، اجرای دستور VOL بصورت زیر است: `C:\DOS\LCK.TMP >> VOL` بعد با باز کردن فایل LCK.TMP، می‌توانیم به محتویات آن دسترسی پیدا کنیم.

راه دیگر مراجعه به Boot Sector جهت کنترل قفل می‌باشد.

ضرب اطمینان این قفل در مورد دیسکت ها، دو تا پنج درصد بوده و در رابطه با هارد دیسک پنجاه تا شصت درصد می‌باشد. دلیل این اختلاف این است که در حالت قفل دیسکتی با کپی Boot Sector، قفل بر روی دیسکت دیگر قرار خواهد گرفت اما در رابطه با هارد دیسک اینکار به سادگی انجام پذیر نیست.

2- قفل گذاری با استفاده از مشخصات سیستم

در این نوع قفل نرم افزاری، برنامه قبل از اجرا ابتدا مشخصات سیستم را خوانده (که اینکار از طریق مراجعه به بخش‌های خاصی از حافظه و یا مراجعه به اطلاعات BIOS انجام می‌شود). سپس آنرا با فایلی که قبلاً توسط نویسنده نرم افزار بر روی کامپیوتر کپی گردیده، مقایسه می‌کند و در صورت عدم برابری، اجرای برنامه پایان می‌پذیرد. این نوع قفل هنوز هم در بسیاری از برنامه‌ها استفاده می‌گردد، اما نکته قابل ذکر این است که جهت اطمینان بیشتر به قفل لازم است فایل حاوی مشخصات بصورت کد شده نوشته باشد تا امکان دستکاری آن توسط قفل شکنان به حداقل ممکن برسد. درصد اطمینان این نوع قفل 75-65% می‌باشد.

3- قفل با استفاده از موقعیت فایل روی هارد دیسک

این نوع قفل فقط بر روی هارد دیسک قابل استفاده بوده و به این صورت است که فایل اجرایی به موقعیت خود بر روی هارد حساس می‌باشد چرا که قبل از اجرا ابتدا موقعیت خود را از روی سکتورهای ROOT خوانده و سپس شماره کلاستر اشاره گر به خودش را بدست می‌آورد، سپس آنرا با شماره کلاستری که قبلاً توسط برنامه نویس بر روی یکی از فایل‌های برنامه (ممکن است بصورت کد شده باشد) قرار داده شده، مقایسه کرده و در صورت برابر بودن اجرا می‌شود. این نوع قفل نسبت به قفل قبلی (شماره 2) استفاده کمتری داشته چرا که در صورتیکه برنامه از روی بخشی از هارد به ناحیه دیگری انتقال یابد. اجرا نخواهد شد و این از نظر کاربر بسیار ناپسند می‌باشد (ضمناً امکان Scandisk، Defrag و... نیز وجود ندارد چرا که شماره کلاستر اشاره گر به فایل تغییر خواهد کرد). ضرب اطمینان این نوع قفل نیز 80-70% می‌باشد.

4- قفل با استفاده از فرمت غیراستاندارد

این شیوه یکی از رایج‌ترین قفل‌های نرم افزاری است که هنوز هم بصورت جدی مورد استفاده قرار می‌گیرد. برخی از دلایل اهمیت آن عبارتند از:

- امکان استفاده از روش‌های متفاوت در این روش راحتی و سرعت زیاد به هنگام استفاده آن عدم وجود نرم افزار خاصی جهت باز کردن این نوع از قفل‌ها همان طور که می‌دانیم سیستم عامل جهت دسترسی به اطلاعات یک دیسکت از فرمت خاصی (18 سکتور در هر تراک) استفاده می‌کند اما اگر یک تراک به صورت غیر استاندارد فرمت شود، (مثلاً 19 سکتور در تراک) سیستم عامل دیگر توانایی استفاده از سکتورهای غیرمجاز را نخواهد داشت و بنابراین تمام نرم افزارهای تحت سیستم عامل مزبور نیز از سکتورهای مخفی استفاده نکرده، در نتیجه امکان کپی برداری از آنها بسیار ضعیف است. ما نیز از همین روش جهت طراحی قفل مورد نظرمان استفاده می‌کنیم. بصورتیکه تراک آخر دیسک را بصورت یک سکتوری و با شماره 20 فرمت می‌کنیم. سپس جهت کنترل دیسکت به سکتور فوق مراجعه کرده و در صورت وجود، کنترل برنامه را پی می‌گیریم. البته غیر از تغییر شماره سکتور می‌توان از اندازه غیرمجاز نیز استفاده کرد یعنی بجای اینکه سکتورها را بصورت 512 بایتی فرمت کنیم، از اندازه 1024، 2048 و... استفاده می‌کنیم. این قفل فقط جهت فلایپی دیسک قابل استفاده می‌باشد و درصد اطمینان در این روش حدود 95%-85% می‌باشد.

در قسمت بعدی این مقاله (بخش پایانی) با سایر روش‌های قفل‌گذاری و نقاط قوت و ضعف هر یک از روش‌ها آشنا می‌شویم

منبع: ماهنامه رایانه