Detección interna de posibles virus

Por Angel Canudas

Ante la proliferación de virus informáticos, cualquier precaución es poca, y si nuestros programas serán usados por terceras personas, como mínimo deberían incorporar algún sistema de prevención contra posibles ataques víricos.

La mejor solución sería *rastrear* la memoria en busca de virus, y tomar el control de las interrupciones para impedir escrituras no deseadas en disco. Pero realizar esto en los lenguajes de alto nivel, no es tarea fácil.

Una solución modesta, aunque no menos válida, puede ser la aplicación de una sencilla rutina a nuestros programas, que impida la ejecución de estos, cuando detecte que su tamaño, fecha y hora de creación hayan sufrido algún cambio. Aunque la mayoría de virus respetan la fecha y hora, los troyanos quedarán al descubierto al *engordar* el ejecutable.

La función del fuente 1, que he desarrollado en *CA-Clipper 5.x* (se puede codificar en cualquier otro lenguaje) y que he llamado *Detector(),* se ha de insertar antes del núcleo principal del programa,y recibe los siguientes parámetros:

1) Nombre del ejecutable. En caso de no indicar extensión, añade *.EXE*

2) Tamaño final del ejecutable, que no sabremos hasta después de la primera compilación, por lo que tendremos que corregirlo.

3) Fecha de creación del programa.

4) Hora de creación del programa, que podemos modificar con alguna de las utilidades existentes en el mercado.

```
// --- Fuente 1 ------------------------------------------------------------
// ********************** SOFTWARE DE DOMINIO PUBLICO **********************
// Programa ..: DETECTOR.PRG
// Descripción: Ejemplo del uso de la función detectora de alteraciones en
//              los ejecutables.
// Compilación: clipper detector /n /w
//              rtlink file detector
// Autor .....: Angel Canudas Rey * Apdo. Correos, 372 * 08240 - Manresa
// Fecha .....: Enero-94
// ***************************************************************************

// ---------------------------------------------------------------------------
FUNCTION MAIN()

   Set Date to Italian
   CLS

   // Una vez sabemos el tamaño del ejecutable, y la fecha y hora de crea-
   // ción que le pondremos, indicamos dichos datos a la función Detector()
   Detector( "DETECTOR", 155648, "15-01-94", "01:00:00" )

   Alert( "No se ha detectado ninguna alteración," + ;
          ";y el programa seguiría ..." )

   QUIT

RETURN NIL
// ---------------------------------------------------------------------------
```

```
// ---------------------------------------------------------------------
// Función ...: Detector
// Descripción: Detecta posibles alteraciones del ejecutable, ya sea en
//              tamaño, fecha y hora. En caso de producirse, muestra un
//              mensaje e interrumpimos la ejecución.
// Parámetros : cFile    => Nombre del ejecutable (tipo carácter)
//              nTamanyo => Tamaño del ejecutable (tipo numérico)
//              cFecha   => Fecha de creación     (tipo carácter)
//              cHora    => Hora de creación      (tipo carácter)
// Devuelve ..: NIL
// Variable ..: acDir    => Array que almacenará los ficheros (.EXE) del
//                          directorio actual.
//              i        => Contador del bucle.
//              cMsg     => Almacena el mensaje a mostrar.
//              lVirus   => Variable lógica de control de posibles cambios.
// ---------------------------------------------------------------------
FUNCTION Detector( cFile, nTamanyo, cFecha, cHora )

   LOCAL acDir  := DIRECTORY( "*.EXE" )
   LOCAL i      := 0
   LOCAL cMsg   := "SISTEMA ANTIVIRUS ;;" + ;
                   "Este programa ha sido alterado:;"

   LOCAL lVirus := .f.

   // Si no hemos indicado la extensión al fichero se la ponemos ----------
   if upper( right( cFile, 3 ) ) != "EXE"
      cFile := cFile + ".EXE"
   endif
   // ---------------------------------------------------------------------

   // BUCLE DE COMPROBACION DE CAMBIOS -----------------------------------
   // Una vez localizado el nombre del fichero en el array, comprobamos que
   // su tamaño, fecha y hora de ceración no hayan sufrido cambios.

   for i := 1 to len( acDir )

       if allTrim( upper( acDir[ i, 1 ] ) ) == cFile

          if acDir[ i, 2 ] != nTamanyo         // Comprobación tamaño
             cMsg := cMsg + ;
                     ";;Tamaño inicial: "  + allTrim( str( nTamanyo ) ) + ;
                     " != Tamaño actual: " + allTrim( str( acDir[ i, 2 ] ) )
             lVirus := .t.
          endif

          if dToc( acDir[ i, 3 ] ) != cFecha   // Comprobación fecha
             cMsg := cMsg + ";;Fecha inicial: " + ;
                     cFecha + " != Fecha actual: " + dToc( acDir[ i, 3 ] )
             lVirus := .t.
          endif

          if acDir[ i, 4 ] != cHora            // Comprobación hora
             cMsg := cMsg + ";;Hora inicial: " + cHora + ;
                     " != Hora actual: " + acDir[ i, 4 ]
             lVirus := .t.
          endif

          // Si se detectan cambios salimos del programa ------------------
          if lVirus
             cMsg := cMsg + ";;Ejecución interrumpida. ¡Peligro de VIRUS!"
             Alert( cMsg, { " Salir " } )
             QUIT
          endif
          // -------------------------------------------------------------

        endif

   next
```

```
   // ---------------------------------------------------------------------

RETURN NIL
// ---------------------------------------------------------------------
```

Lo que hace la función *Detector(),* es cargar en un array todos los ficheros (*.EXE*) del directorio actual, y mediante un bucle busca el ejecutable que coincida con el indicado, y realiza las oportunas comprobaciones.

Si se encuentran cambios se muestra el correspondiente mensaje de advertencia, y terminamos la ejecución del programa. A continuación, y mediante la utilización de algún potente detector y/o antivirus externo, hemos de determinar si ha sido una falsa alarma o sopesar el alcance del ataque.