Análisis de Software Malicioso
(MiniProyecto 1)

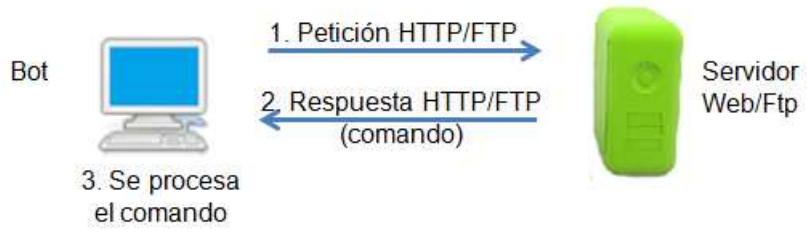
Revisión intermedia: miércoles 14 de septiembre de 2016.
Fecha de entrega: lunes 19 de septiembre de 2016.

Archivos a entregar:

- ZIP (con contraseña si es que se trata de código malicioso) con el código fuente (debidamente comentado e indentado) y el ejecutable (con sus debidas dependencias) correspondiente al malware o a la herramienta de seguridad.
- Documento (con capturas de pantalla) de la implementación paso a paso, conclusiones (de nivel licenciatura) y referencias de la fuentes consultadas.
- Manual de usuario (con capturas de pantalla).

Nota: El alumno elegirá una de dos opciones planteadas para la entrega del primer proyecto.

Propuesta 1: Software malicioso tipo Bot



Desarrollar en cualquier lenguaje de programación una aplicación maliciosa con las siguientes característica:

1.- Ruta particular de ejecución.- El malware siempre se debe ejecutar en la ruta a la que apunta la variable de entorno %temp% (en nuestro laboratorio sería "C:\Users\malware\AppData\Local\Temp", lo pueden comprobar poniendo la variable de entorno como ruta en el explorador de Windows) por lo que primero debe validar el estar posicionado en dicha ruta, es decir, si su muestra se ejecuta en el Escritorio "C:\Users\malware\Desktop" obtendrá dicha cadena, la comparará con la ruta %temp% (se sugiere usar la variable de entorno ya que el usuario cambia) y al no coincidir se procede a copiarse a sí mismo en %temp%. Posteriormente la muestra que se está ejecutando debe iniciar la réplica y terminar su ejecución. La réplica al iniciarse validará la ruta en la cual se está ejecutando y continuará con las acciones maliciosas.

```
if(chdir == %temp%){   // Este sólo es un ejemplo
      accionesMaliciosas();
}
else{
      copy chdir+nombreDelMalware %temp% // El nombre de la muestra no se debe
      winexec %temp%+nombreDelMalware    // especificar estático en su código ¬¬
      exit(0);                           // se puede obtener con "argv[0]"
}
```

2.- Persistencia.- Dentro de las acciones maliciosas a realizar por su programa está la creación del valor del registro que lo hace persistente, por lo que el malware debe obtener su ruta de ejecución (aunque siempre será %temp% se debe manejar como una variable en su programa) y concatenarle su nombre (que es el valor de "argv[0]" ya que su muestra puede ser renombrada y no debe perder funcionalidad).

```
REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v "Proyecto1" /t
REG_SZ /d "%temp%/nombreDelMAlware" /f
```

Nota: antes de agregar el valor de registro anterior se recomienda verificar su existencia para evitar actividad de sobrescritura.

```
if (valorDeRegistroExiste){
	continúa();
}
else{
	creaValorDelRegistro();
}
```

3.- Conexión de red.- Finalmente, lo que debe hacer su programa es conectarse al dominio proyecto1.asm.mx (máquina Linux ya sea por protocolo HTTP (80) o FTP (21)) para descargar el archivo comando.txt, este último tendrá cadenas específicas como "apagaEquipo", "borraArchivosEnElEscritorio", etc. Ustedes propondrán 10 acciones maliciosas que quieran que se realicen en el equipo que está ejecutando su muestra de malware.

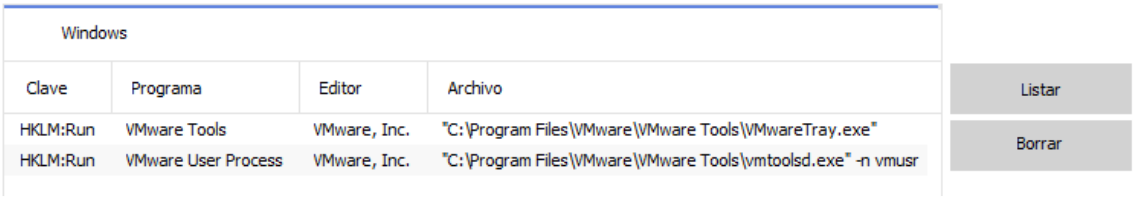
Esta es una forma de hacerlo con llamadas al sistema:

- Crear una conexión HTTP y una sesión:
  InternetOpen() e InternetConnect()
- Construir la petición HTTP con cabecera personalizada:
  HttpOpenRequest() y HttpAddRequestHeaders()
- Enviar la petición HTTP (se inicia la conexión TCP/IP):
  HttpSendRequest()
- Leer la respuesta del servidor web:
  InternetReadFile()

Una vez que el malware consulte (no es necesario descargar el archivo, hay forma de únicamente leer el contenido del recurso) el archivo comando.txt, comparará la cadena con las 10 que fueron definidas para realizar una acción y se llevará a cabo. La actividad del malware deberá suspenderse por 30 segundos y volver a realizar la consulta del archivo por alguna actividad nueva, este proceso de preguntar cada cierto tiempo será un ciclo infinito [while(1)].

```
switch(comando){
	case "apagaEquipo":
		shutdown /s /f /t 0
		break;
		.
		.
		.
	defalut:
		printf("Ha ingresado un comando no valido\n");
		break;
}
```

Propuesta 2: Herramienta para desactivar persistencia



Programar en cualquier lenguaje (de preferencia con interfaz gráfica) una herramienta que permita listar las entradas de todo lo que es persistente en el equipo Windows con la finalidad de identificar rápidamente qué tipo de software se inicia. La herramienta permitirá seleccionar alguna entrada y borrarla para así rápidamente quitar la persistencia de las aplicaciones identificadas como maliciosas sin tener que ir a las rutas (en el registro o en el sistema de archivos) de forma manual.

Para probar su herramienta se recomienda primero agregar o modificar valores del registro para probar la persistencia de algunas aplicaciones como se mostró en clase. Ya verificadas todas las entradas listadas a continuación se procede a desarrollar y probar su herramienta.

--- INICIO ---

1.- Valores de registro (Entradas creadas del lado derecho. Otra cosa importante es que después de llegar a la carpeta “Policies” se debe crear la carpeta “Explorer” y dentro de esta “Run” para finalmente agregar un valor)

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce

HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

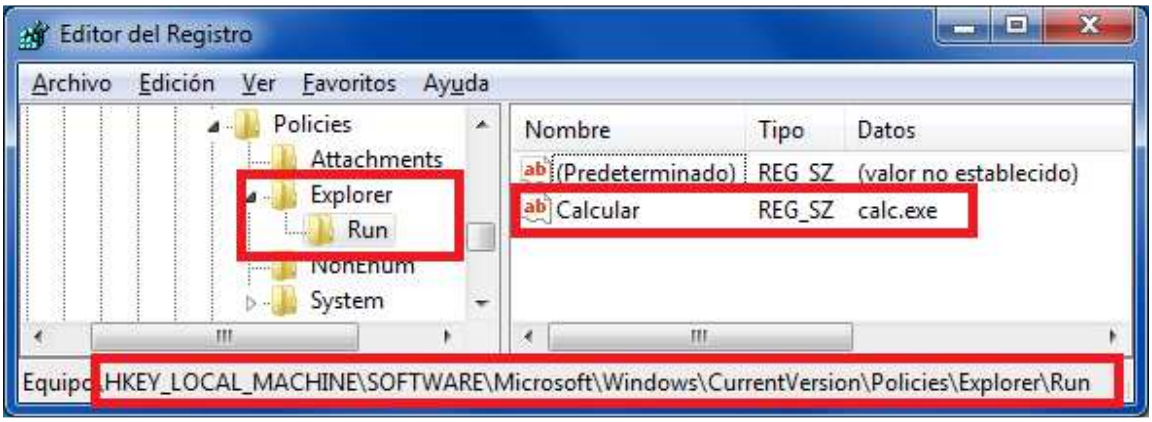
2.- Dato modificado en el Valor del registro (En clase modifiqué la cadena “explorer.exe” por “explorer.exe, mspaint.exe” en el valor Shell)

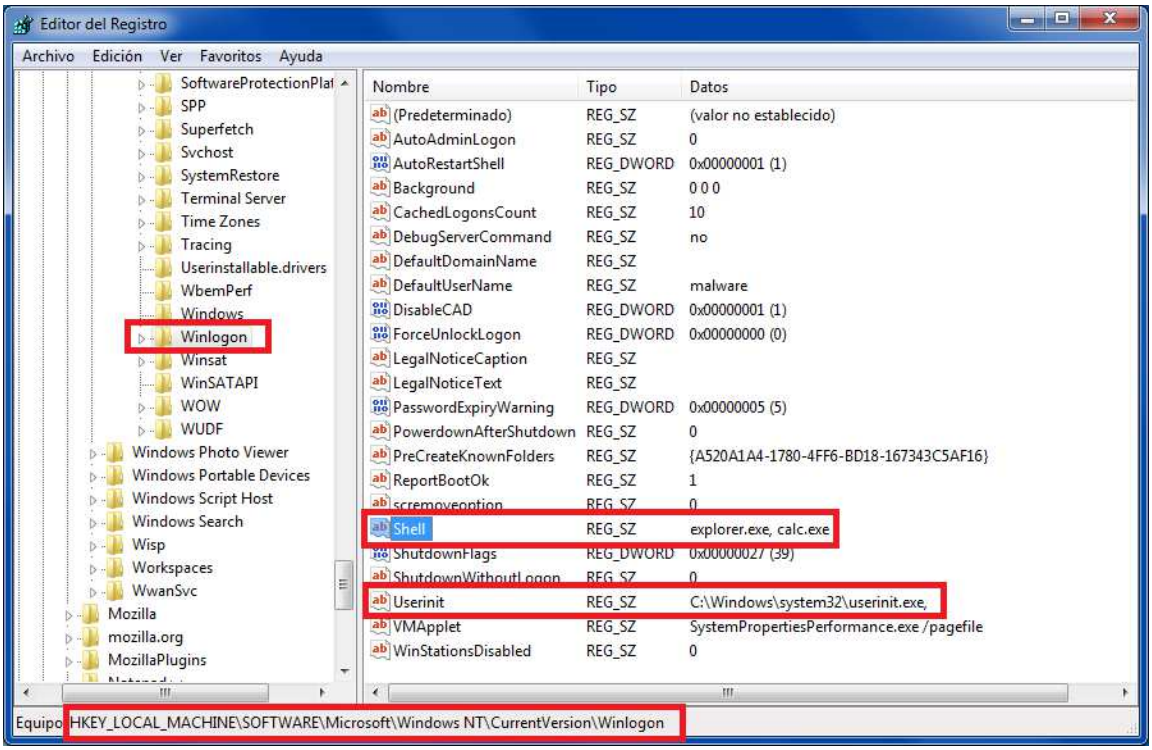
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon: Shell
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon: Userinit

3.- Carpetas inicio (Recuerden que pueden ser accesos directos o ejecutables en la carpeta)

%AppData%\Microsoft\Windows\Start Menu\Programs\Startup
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup

--- FIN ---


Editor del Registro
Archivo Edición Ver Favoritos Ayuda
Policies
Attachments
Explorer
Run
System
Nombre Tipo Datos
(Predeterminado) REG_SZ (valor no establecido)
Calcular REG_SZ calc.exe
Equipo\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run


Editor del Registro
Archivo Edición Ver Favoritos Ayuda
SPP
Superfetch
Svchost
SystemRestore
Terminal Server
Time Zones
Tracing
Userinstallable.drivers
WbemPerf
Windows
Winlogon
Winsat
WinSATAPI
WOW
WUDF
Windows Photo Viewer
Windows Portable Devices
Windows Script Host
Windows Search
Wisp
Workspaces
WwanSvc
Mozilla
mozilla.org
MozillaPlugins
Nombre Tipo Datos
(Predeterminado) REG_SZ (valor no establecido)
AutoAdminLogon REG_SZ 0
AutoRestartShell REG_DWORD 0x00000001 (1)
Background REG_SZ 0 0 0
CachedLogonsCount REG_SZ 10
DebugServerCommand REG_SZ no
DefaultDomainName REG_SZ
DefaultUserName REG_SZ malware
DisableCAD REG_DWORD 0x00000001 (1)
ForceUnlockLogon REG_DWORD 0x00000000 (0)
LegalNoticeCaption REG_SZ
LegalNoticeText REG_SZ
PasswordExpiryWarning REG_DWORD 0x00000005 (5)
PowerdownAfterShutdown REG_SZ 0
PreCreateKnownFolders REG_SZ {A520A1A4-1780-4FF6-BD18-167343C5AF16}
ReportBootOk REG_SZ 1
scremoveoption REG_SZ 0
Shell REG_SZ explorer.exe, calc.exe
ShutdownFlags REG_DWORD 0x00000027 (39)
ShutdownWithoutLogon REG_SZ 0
Userinit REG_SZ C:\Windows\system32\userinit.exe,
VMApplet REG_SZ SystemPropertiesPerformance.exe /pagefile
WinStationsDisabled REG_SZ 0
Equipo\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon