Esercizi_S10_L1

Consegna

ESERCIZIO 1: Preparazione alle esercitazioni dei prossimi giorni

Scaricare e installare Per VirtualBox:

- https://drive.google.com/file/d/1w9DG0erQ763XsJVOu7zH8RbZM2IkSob3/view?usp=drive_link
- https://drive.google.com/file/d/1FSneSlqbyCD_dTo8F2NJMkE650y1UXZo/view?usp=sharing

ESERCIZIO 2: Esplorazione di Processi, Thread, Handle e Registro di Windows.

Obiettivi In questo laboratorio, esplorerai i processi, i thread e gli handle utilizzando Process Explorer della Suite SysInternals. Utilizzerai anche il Registro di Windows per modificare un'impostazione.

Parte 1 Esplorazione dei Processi
Parte 2 Esplorazione di Thread e Handle
Parte 3 Esplorazione del Registro di Windows Risorse Richieste

Risorse richieste:

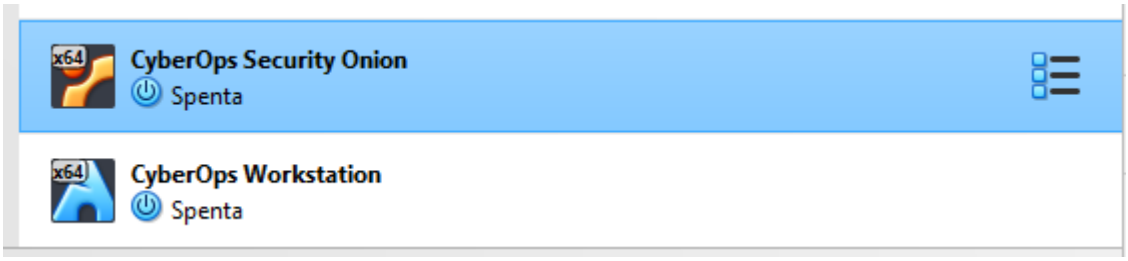
1 PC Windows con accesso a internet

Svolgimento

Esercizio 1

Il primo esercizio consisteva semplicemente nel scaricare i due file .ova di Security Onion e di CyberOps Workstation.

Una volta scaricate le due VMs è stato sufficiente caricarle all'interno di VBox ed avviarle per renderle già operative.

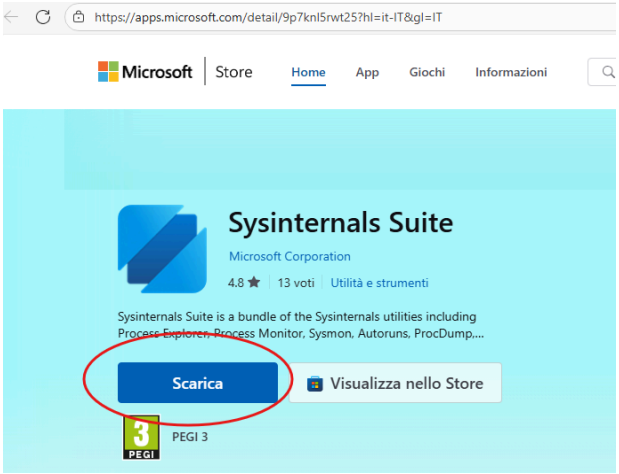


Una volta avviato CyberOps Workstation ci verrà presentata una classica schermata di login e a seguito dell'inserimento delle credenziali Analyst:cyberops avremo accesso al desktop utente della piattaforma:

Esercizio 2

Fase 1

Per lo svolgimento del secondo esercizio si è resa necessario il download di Sysinternals tramite lo store di Microsoft:



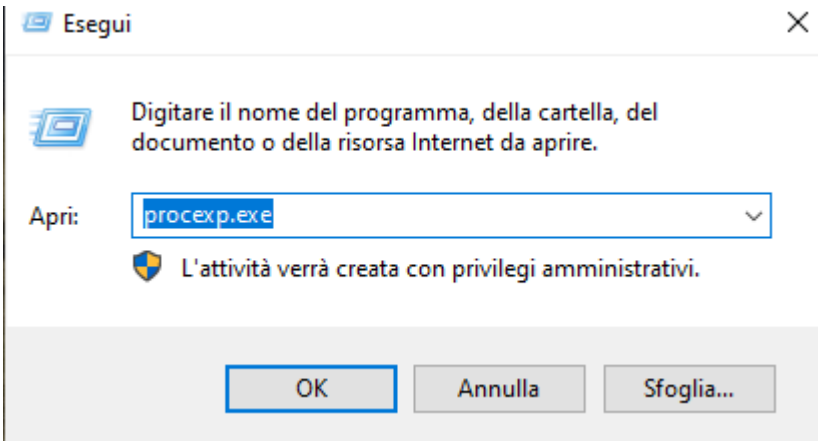
Dopo aver scaricato ed avviato l'installer, partirà il download effettivo della Suite (c.a 85MB) ed al termine del download avremo a disposizione tutti i tools del pacchetto.

DI seguito possiamo trovare una lista presa dal sito di Microsoft che ci comunica come è possibile richiamare tramite shell ogni programma appena installato:

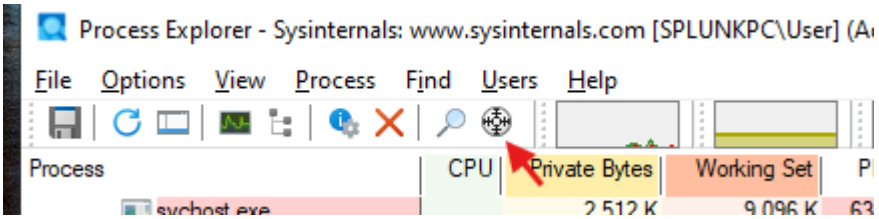
```
crosoft.SysinternalsSuite_8wekyb3d8bbwe

 accesschk.exe      AccessEnum.exe     ADExplorer.exe     ADInsight.exe
 adrestore.exe      Autologon.exe      Autoruns.exe       autorunsc.exe
 Bginfo.exe         Cacheset.exe       Clockres.exe       Contig.exe
 Coreinfo.exe       CPUSTRES.EXE       ctrl2cap.exe       Dbgview.exe
 Desktops.exe       disk2vhd.exe       diskext.exe        Diskmon.exe
 DiskView.exe       du.exe             efsdump.exe        FindLinks.exe
 handle.exe         hex2dec.exe        junction.exe       Listdlls.exe
 livekd.exe         LoadOrd.exe        LoadOrdC.exe       logonsessions.e
 movefile.exe       notmyfault.exe     notmyfaultc.exe    ntfsinfo.exe
 pendmoves.exe      pipelist.exe       procdump.exe       procexp.exe
 Procmon.exe        PsExec.exe         psfile.exe         PsGetsid.exe
 PsInfo.exe         pskill.exe         pslist.exe         PsLoggedon.exe
 psloglist.exe      pspasswd.exe       psping.exe         PsService.exe
 psshutdown.exe     pssuspend.exe      RAMMap.exe         RDCMan.exe
 RegDelNull.exe     regjump.exe        ru.exe             sdelete.exe
 ShareEnum.exe      ShellRunas.exe     sigcheck.exe       streams.exe
 strings.exe        sync.exe           Sysmon.exe         tcpvcon.exe
 tcpview.exe        Testlimit.exe      vmmap.exe          Volumeid.exe
 whois.exe          Winobj.exe         ZoomIt.exe
```

Fase 2

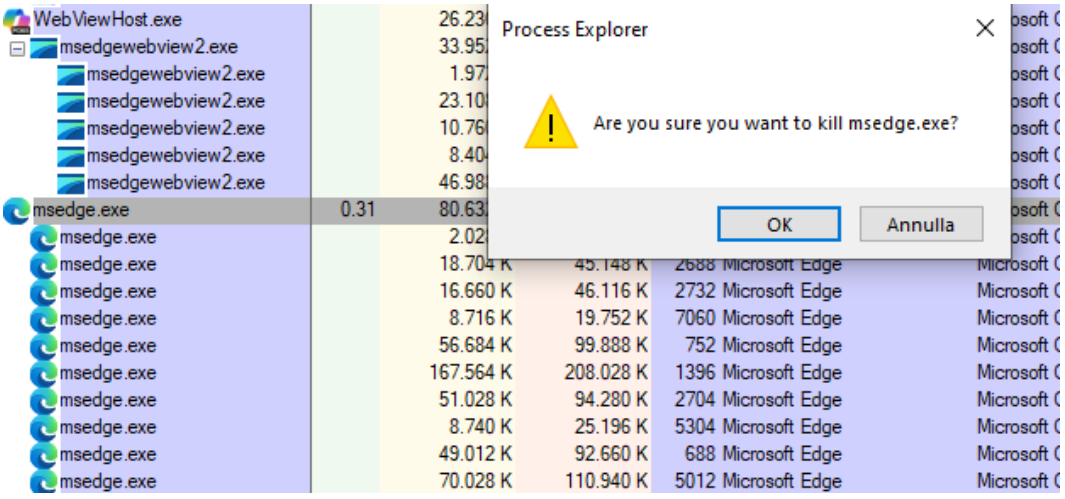
La seconda fase dell'esercizio ci chiedeva di avviare il Process Explorer:



Utilizzando l'utility Find windows's process ci è possibile trascinare il simbolo del target sopra al processo/applicazione di nostro interesse ed in automatico ci verrà evidenziato in lista il processo relativo all'interno di procexp:



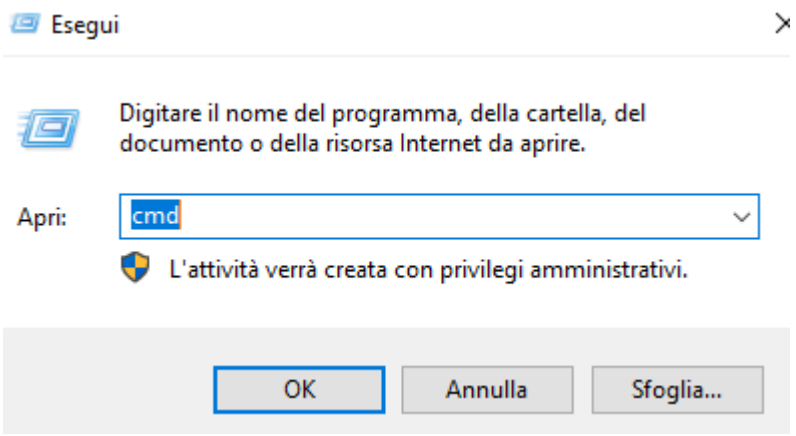
Una volta trovato il processo interessato è possibile terminarlo tramite un click destro → Kill Process ed immediatamente il processo in oggetto e tutti i processi figli verranno rapidamente evidenziati in rosso, verranno chiusi ed infine spariranno dalla lista dei processi attivi:



Ovviamente anche la GUI relativa al processo in questione verrà forzatamente terminata.

Fase 3

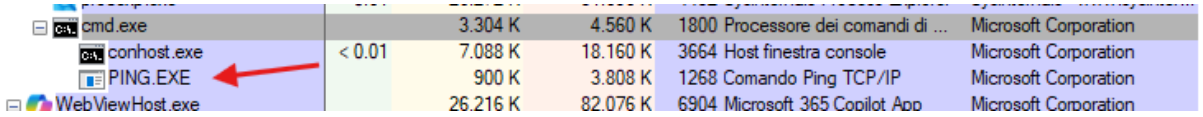
La terza fase dell'esercizio prevedeva di avviare un ulteriore processo, in questo caso il **prompt dei comandi**.



Utilizzando nuovamente l'utility Finder dei processi possiamo facilmente individuare i processi responsabili dell'esecuzione di cmd:

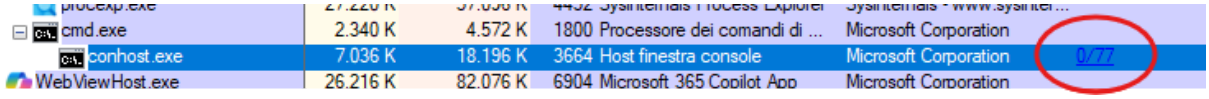
Dall'immagine possiamo notare che cmd.exe, **parent process**, da origine ad un processo figlio, **child process**, chiamato conhost.exe.

Eseguendo un comando come PING all'interno del prompt dei comandi noteremo che **per svolgere l'azione cmd.exe avvierà un altro child process chiamato PING.EXE**:



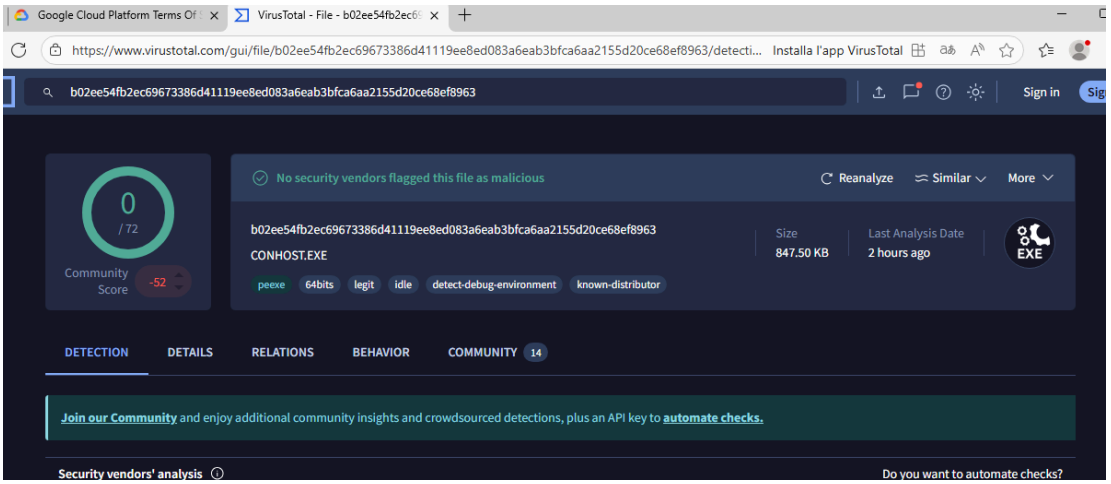
Qualora un processo ci sembri sospetto è possibile analizzarlo tramite VirusTotal; per fare ciò è sufficiente cliccarci sopra con tasto destro e procedere con Check VirusTotal.com.

A quel punto procexp.exe invierà a VirusTotal l'hash del processo e ci restituirà immediatamente il risultato della scansione avvenuta nei sui Database:



Il risultato verrà mostrato alla destra del processo in oggetto e, nella fattispecie possiamo vedere che 0 dei 77 motori di ricerca ha flaggato il processo come dannoso.

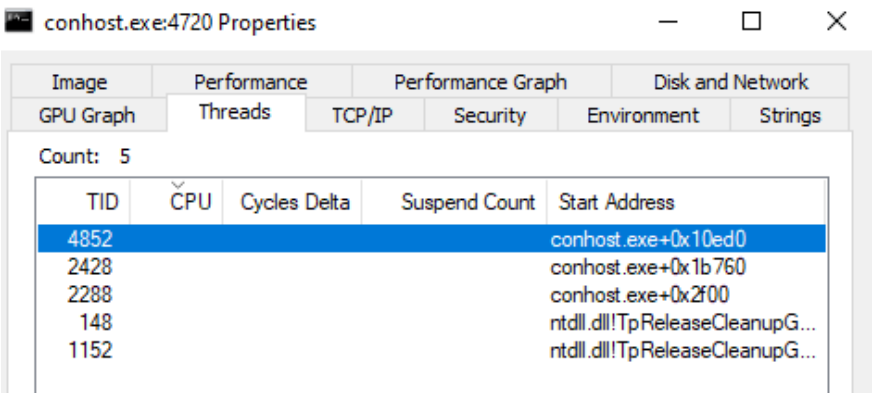
Cliccando su quel risultato verremo reindirizzati tramite browser alla pagina di virustotal dove potremo analizzare la scansione nel dettaglio:



Provando a terminare il processo figlio conhost.exe noteremo che anche il processo padre verrà interrotto; ciò non accade di regola ma nel caso in esempio il processo padre non può funzionare senza la presenza di conhost.exe.

Fase 4

Per la fase 4 ho avviato nuovamente il prompt dei comandi e, tramite procexp.exe, ho raggiunto le proprietà del servizio conhost.exe per osservare i threads utilizzati:



All'interno della scheda **Proprietà→Threads** è possibile visionare diverse informazioni:

TID (Thread Identifier)
Identificatore univoco del thread nel sistema. Ogni thread ha un ID distinto, usato per riferirsi ad esso in strumenti di debugging o diagnostica.

CPU
Indica la percentuale di utilizzo della CPU attribuita a quel singolo thread. È utile per capire quale thread sta consumando più risorse di calcolo.

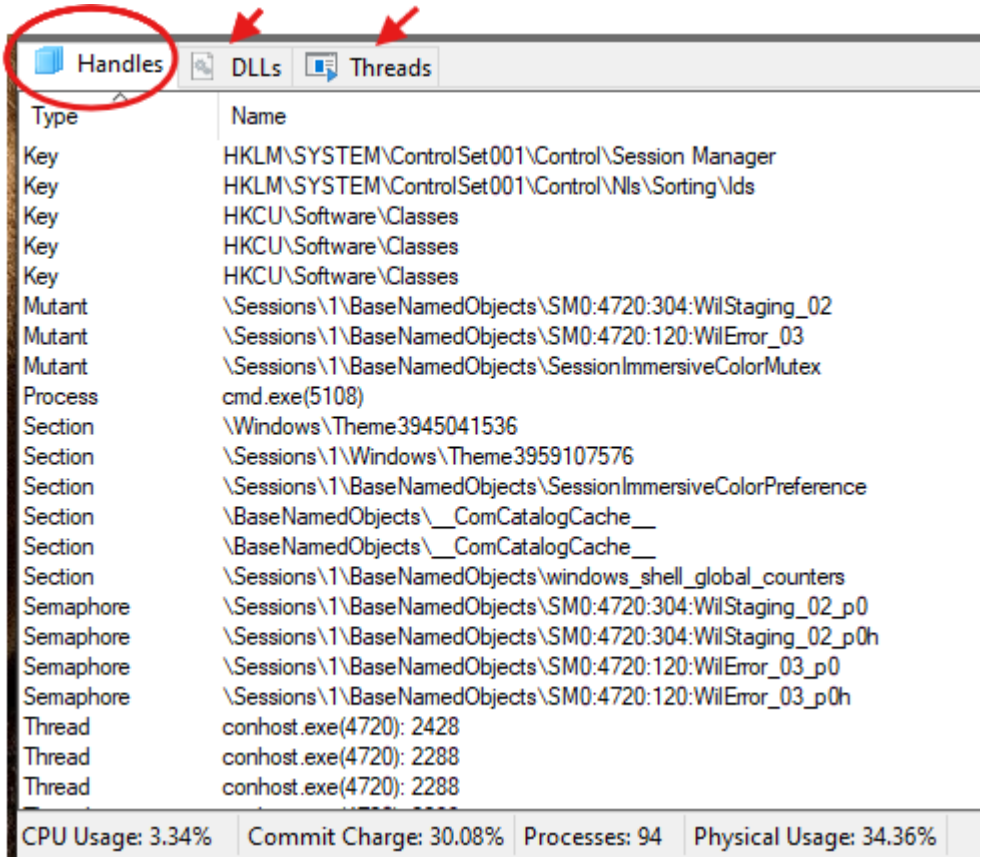
Cycles Delta
Misura il numero di cicli di CPU consumati dal thread in un intervallo di tempo. Fornisce un'indicazione più precisa dell'attività del thread rispetto alla sola percentuale CPU.

Suspend Count
Mostra quante volte il thread è stato sospeso. Un valore diverso da zero può indicare che il thread è stato temporaneamente messo in pausa, ad esempio da strumenti di debugging.

Start Address
Indica l'indirizzo di memoria o il modulo da cui ha avuto inizio il thread. Questo campo è cruciale perché rivela se il thread è stato avviato da un modulo legittimo del sistema (es. ntdll.dll) o da un codice sospetto.
Nell'immagine, ad esempio, vediamo thread avviati da:

- conhost.exe (normale, è il processo in analisi);
- ntdll.dll (legittimo, libreria di sistema di Windows).

Spostandoci invece in Process Explorer → View → Lower Pane View Vista → Handles ci è possibile visionare molte altre informazioni vitali per il controllo e la sicurezza di un processo:



Nell'immagine qui sopra possiamo infatti notare la possibilità di visionare i file **dll** utilizzati dal processo, lo **status dei threads** ed anche gli **handles.**

Un handle è un riferimento a una risorsa del sistema operativo. Quando un processo apre un file, una chiave di registro, un oggetto di sincronizzazione o un altro tipo di risorsa, il sistema gli assegna un handle. Questo handle è un identificatore che il processo usa per interagire con la risorsa senza dover conoscere i dettagli a basso livello.

Nell'immagine vediamo vari handle associati al processo conhost.exe. Gli handle possono puntare a diverse tipologie di oggetti:

Key
Riferimenti a chiavi del registro di sistema.
Ad esempio:

HKLM\SYSTEM\ControlSet001\Control\Session Manager
HKCU\Software\Classes

> *Questo mostra che conhost.exe sta leggendo configurazioni da aree del Registro relative a sessioni utente e impostazioni di sistema.*

Mutant (Mutex)
Oggetti di sincronizzazione che impediscono l'accesso contemporaneo a una risorsa.
Ad esempio:

\Sessions\1\BaseNamedObjects\SM0:4720:WinStaging_02

> *Serve a garantire che due processi non accedano allo stesso oggetto in conflitto.*

Process
Handle verso altri processi, in questo caso cmd.exe (5108).

> *Questo è normale in quanto conhost.exe viene spesso avviato come "Console Host" per gestire finestre della riga di comando (cmd.exe, PowerShell, ecc.).*

Section
Oggetti che rappresentano aree di memoria condivisa.
Ad esempio:

\Sessions\1\BaseNamedObjects__ComCatalogCache__

> *Queste sezioni permettono la condivisione di dati tra processi o il caricamento di moduli in memoria.*

Semaphore
Meccanismi di sincronizzazione usati per gestire l'accesso concorrente a risorse condivise.

> *Sono utilizzati internamente per coordinare i thread del processo.*

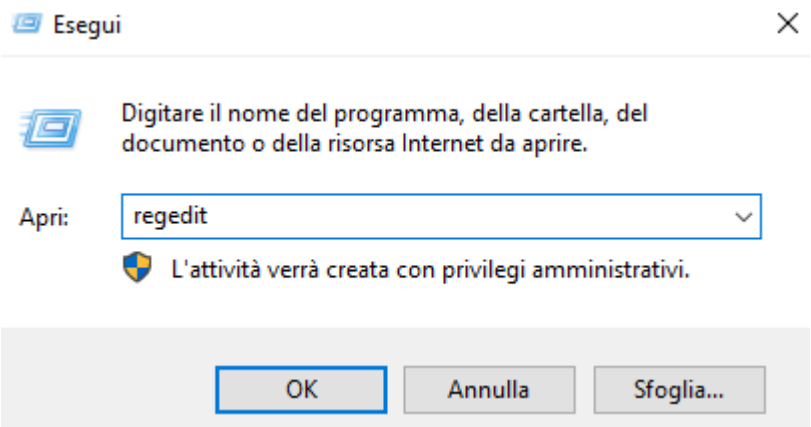
Thread
Handle ai thread appartenenti al processo stesso.
Ad esempio:

conhost.exe(4720): 2428
conhost.exe(4720): 2288

> *Questi riferimenti permettono di gestire, sospendere o terminare thread specifici.*

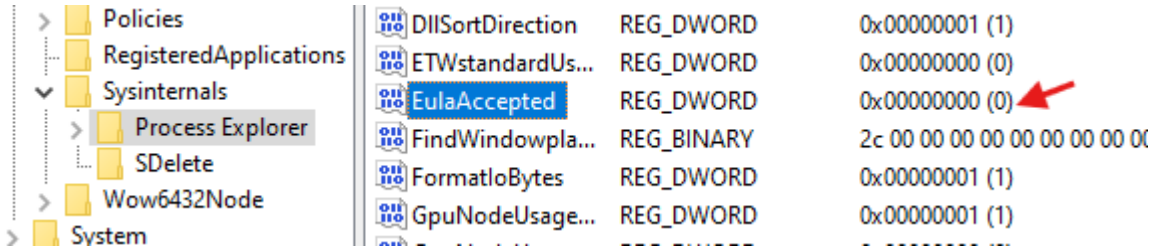
Fase 5

La fase 5 dell'esercizio richiedeva di avviare il registro delle chiavi di windows:



Una volta fatto ciò ci veniva richiesto di modificare il valore della chiave relativa al contratto utente siglato con Process Explorer:

HKEY_CURRENT_USER Software > Sysinternals > Process Explorer > EulaAccepted



Il valore era inizialmente settato a 1 e significava che il contratto era stato accettato; variando a 0 il valore, il contratto risulterà non accettato e al prossimo avvio di **procexp.exe** ci verrà chiesto nuovamente di accettare il contratto.

Conclusioni

Questo laboratorio ha permesso di acquisire familiarità con alcuni strumenti fondamentali per l'analisi e la gestione dei processi in ambiente Windows. In particolare, attraverso l'uso di VirtualBox e delle macchine virtuali fornite (CyberOps Workstation e Security Onion), si è predisposto un ambiente sicuro e isolato, adatto allo svolgimento di esercitazioni pratiche di cybersecurity.

Con l'impiego di Process Explorer della suite Sysinternals si è osservato in dettaglio il funzionamento dei processi, dei thread e degli handle. È stato possibile verificare la relazione padre–figlio tra processi (ad esempio cmd.exe e conhost.exe), identificare thread interni e analizzarne i parametri principali (CPU, TID, indirizzi di avvio). Inoltre, l'analisi degli handle ha mostrato come un singolo processo mantenga riferimenti verso risorse di sistema molto eterogenee (chiavi di registro, mutex, sezioni di memoria, semafori e altri processi).

Un ulteriore aspetto rilevante è stato l'intervento diretto sul Registro di Windows, modificando la chiave EulaAccepted per Process Explorer. Questa attività ha evidenziato come configurazioni e preferenze dei programmi vengano memorizzate a livello di registro, e come un semplice cambiamento di valore possa influire sul comportamento all'avvio dell'applicazione.

In sintesi, l'esercitazione ha evidenziato tre concetti fondamentali:

1. Ogni processo è un'entità complessa composta da più thread ed è strettamente legata a risorse esterne (handle).

2. Gli strumenti Sysinternals consentono un'analisi avanzata e granulare, utile sia per finalità amministrative sia di sicurezza.

3. Il Registro di Windows rappresenta un punto critico di configurazione, la cui manipolazione va effettuata con attenzione poiché può modificare sensibilmente il comportamento del sistema o dei programmi.

Queste competenze costituiscono le basi per attività più avanzate di monitoraggio, diagnostica e digital forensics su sistemi Windows, dove l'osservazione dei processi e delle risorse correlate è essenziale per distinguere un comportamento legittimo da uno potenzialmente malevolo.