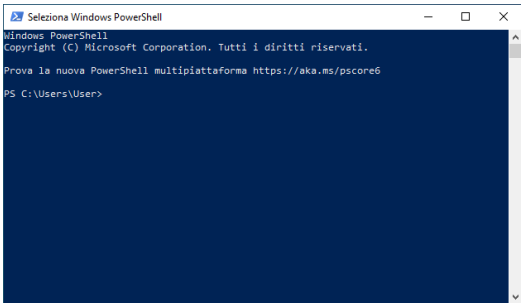
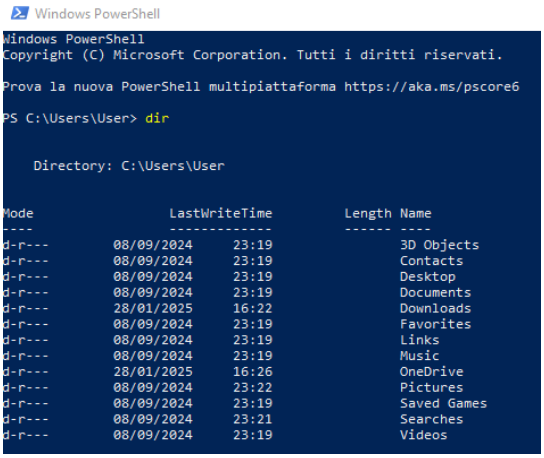
Progetto S11-L5

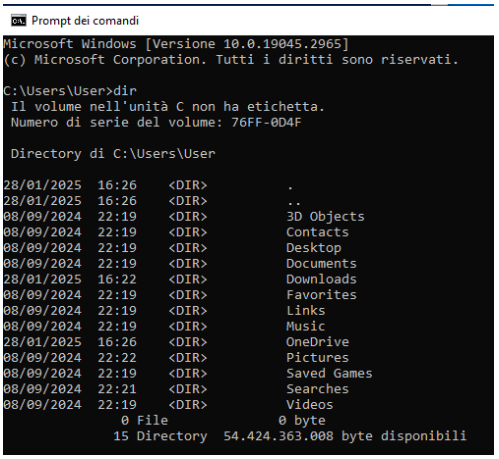
- ## Parte 1 - Windows Powershell

Avvio macchina virtuale Windows 10 pro ed avvio di **Windows Powershell** e **Prompt dei comandi**

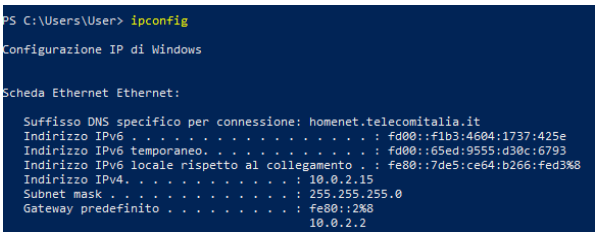


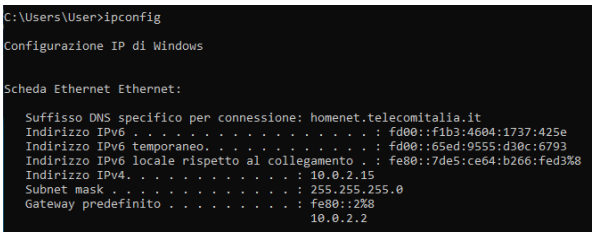
Inserimento comando **dir** in entrambe le finestre per individuare eventuali differenze





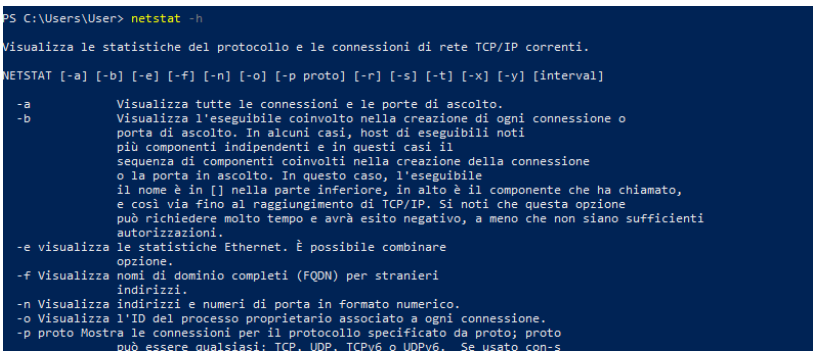
Come possiamo notare dalle immagini soprastanti, l'unica vera differenza degna di nota è la categoria "**Mode**" presente in Windows Powershell e non presente invece nel prompt dei comandi. Essa fornisce informazioni sugli attributi dei file: ad esempio la **d** indica che si tratta di una directory, mentre la **r** che si tratta di un file di sola lettura.
Facendo un tentativo con altri comandi, come ad esempio **ipconfig**, i risultati sono invece simili.



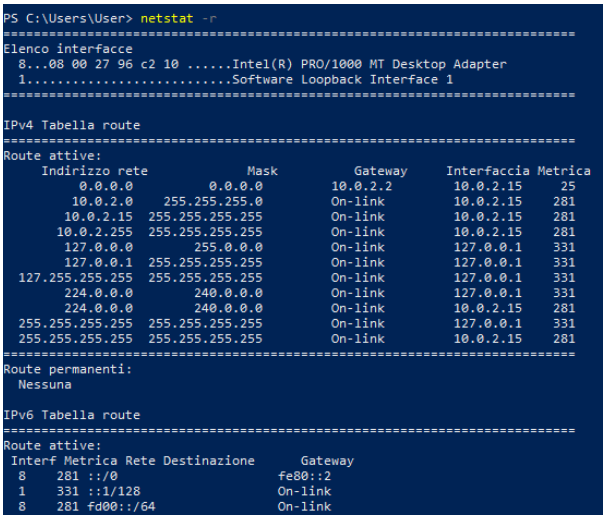


Esistono una grande quantità di comandi in Powershell chiamati **cmdlets**.
Ad esempio, inserendo il comando **Get-Alias** in powershell, è possibile visualizzare gli alias disponibili per quel comando e i comandi a cui esso è associato.

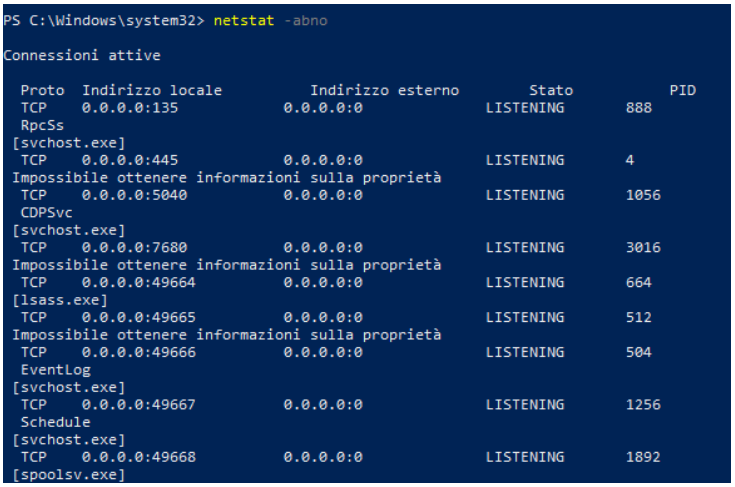
Facendo un tentativo con il comando **dir**, possiamo vedere che esso è un alias del comando **Get-ChildItem**.
Un altro comando molto importante su Powershell è **netstat**. E' possibile inserire il comando **netstat -h** per visualizzare tutte le opzioni disponibili per tale comando.



Inserendo il comando **netstat -r** è possibile visualizzare la tabella di route con i "percorsi" attivi.



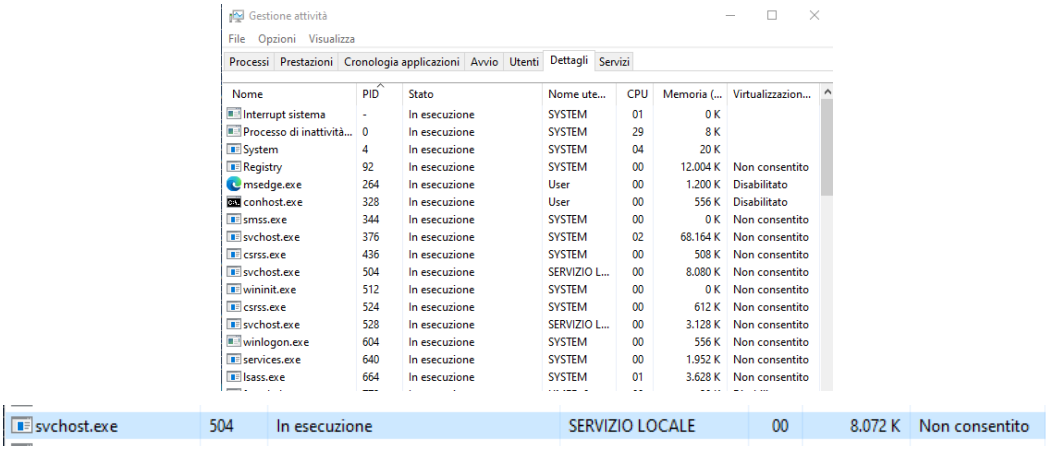
Grazie a questo comando, tra le altre cose, è possibile risalire all'indirizzo IP del gateway impostato, che in questo caso è 10.0.2.2.
Apriamo ora una nuova finestra di Windows Powershell con permessi di Amministratore ed inseriamo il comando **netstat -abno**.



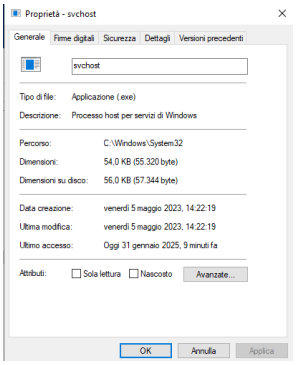
Questo comando ci permette di visualizzare le connessioni TCP attive.
Ogni processo è identificato da un valore **PID**. Prendiamo come esempio il processo con PID 504.

```
  TCP    [::]:49666             [::]:0                 LISTENING       504
  EventLog
[svchost.exe]
```

Ora apriamo il Task Manager (Gestione attività) ed individuiamo tale processo.



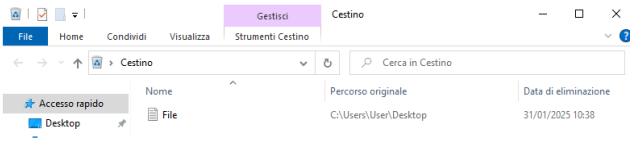
Da qui è possibile aprire la sezione Proprietà per tale processo e leggerne i dettagli.



Grazie a ciò, siamo in grado di risalire ad alcune informazione:

1. Nome del processo associato al PID: in questo caso “svchost.exe”
2. L’utente che sta eseguendo questo processo: in questo caso “SERVIZIO LOCALE”
3. La quantità di memoria utilizzata: in questo caso 8072 K.

Creiamo ora un file di testo chiamato file.txt ed eliminiamolo. In questo modo sarà possibile visualizzare tale file nel Cestino.

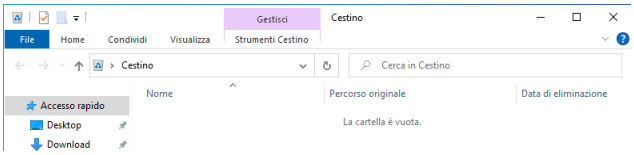


E’ possibile “svuotare” il cestino tramite Powershell grazie al comando **clear-recyclebin**.

```
PS C:\Windows\system32> clear-recyclebin

Conferma
Eseguire l'operazione?
Esecuzione dell'operazione "Clear-RecycleBin" sulla destinazione "Tutto il contenuto del Cestino".
[S] Sì  [T] Sì a tutti  [N] No  [U] No a tutti  [O] Sospendi  [?] Guida (il valore predefinito è "S"): si
[S] Sì  [T] Sì a tutti  [N] No  [U] No a tutti  [O] Sospendi  [?] Guida (il valore predefinito è "S"): t
PS C:\Windows\system32>
```

Tornando nel cestino, possiamo vedere che adesso risulterà vuoto.

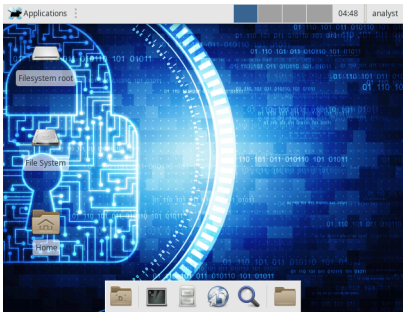


Ecco ora una rapida panoramica su alcuni comandi in Powershell utili per un security analyst:

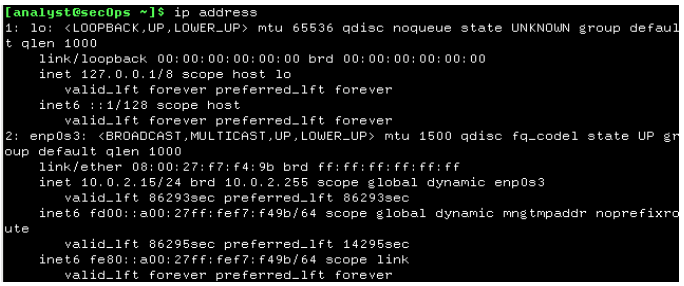
1. **Get-LocalUser**: per elencare tutti gli utenti locali.
2. **Get-LocalGroup**: per elencare tutti i gruppi locali.
3. **Get-Process**: per monitorare i processi in esecuzione sul sistema. E' possibile utilizzare anche il comando **Get-Process -Name "nome-processo"** per ottenere informazioni dettagliate su un processo specifico.
4. **Get-EventLog**: per controllare i log degli eventi ed identificare attività sospette. Ad esempio, il comando **Get-EventLog -LogName Security -Newest 10** mostrerà gli ultimi 10 eventi nel log di sicurezza di Windows.
5. **Get-MpPreference**: per monitorare la configurazione di Windows Defender.
6. **Get-Service**: per ottenere informazioni sui vari servizi. ad esempio, il comando **Get-Service -Name "wuauserv", "MpsSvc", "WinDefend"** mostrerà lo stato di Windows Update (wuauserv), Windows Firewall (MpsSvc), e Windows Defender (WinDefend).
7. **Get-LocalGroupPolicy**: per ottenere informazioni sulla configurazione delle policy di sicurezza di un sistema.
8. **Get-NetFirewallRule**: per monitorare e configurare il firewall di Windows.
9. **Clear-DnsClientCache**: per svuotare la cache DNS. Utile se si sospettano attacchi DNS spoofing.

Parte 2 - Analisi traffico HTTP e HTTPS

Scarichiamo, importiamo ed avviamo la CyberOps WS.



Apriamo il terminale ed inseriamo il comando **ip address** per ottenere informazioni sulla scheda di rete.

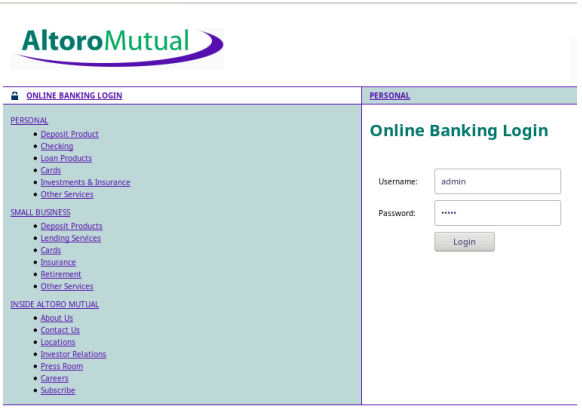


Dall'immagine soprastante possiamo vedere che alla scheda di rete **enp0s3** è associato l'indirizzo IP 10.0.2.15 con netmask 255.255.255.0 (come indica il **/24**), mentre l'indirizzo di loopback è 127.0.0.1/8.
Avviamo ora **tcpdump**, per la cattura dei pacchetti di rete, utilizzando il seguente comando:

```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap
```

1. **sudo**: permette di eseguire il comando con permessi da superuser.
2. **-i**: permette di selezionare l'interfaccia di rete.
3. **-s**: permette di definire la lunghezza di cattura di ogni pacchetto. Quando impostato su 0 si setta in automatico sul valore di default.
4. **-w**: permette di salvare la cattura in un file, in questo caso chiamato "httpdump.pcap".

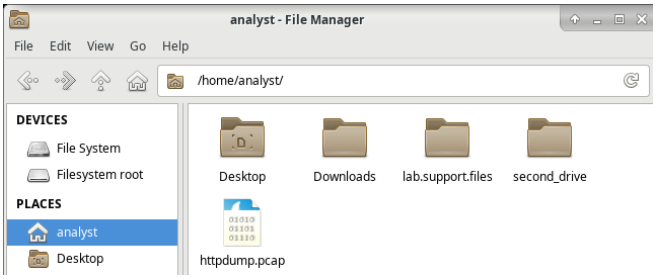
Una volta avviato tcpdump, apriamo il browser e colleghiamoci al sito "www.altoromutual.com/login.jsp".



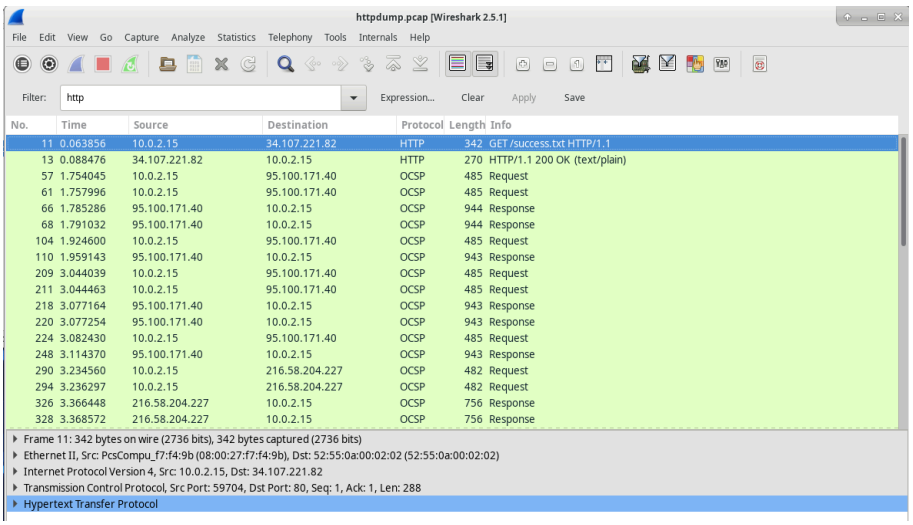
Tale sito utilizza il protocollo **HTTP**.
Una volta collegati, effettuiamo il login e successivamente terminiamo la cattura tramite tcpdump.

```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpdump.pcap
[sudo] password for analyst:
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 b
ytes
^C2686 packets captured
2686 packets received by filter
0 packets dropped by kernel
```

Nel percorso **/home/analyst** individuiamo il file appena creato.

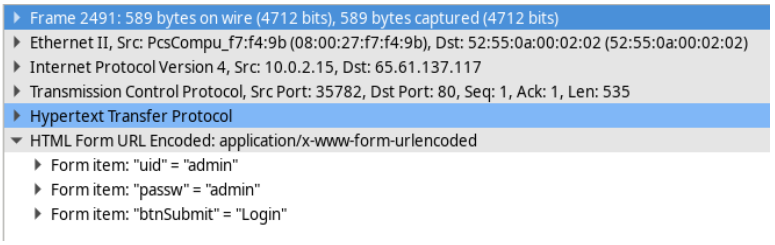


Apriamo il file con **Wireshark** ed applichiamo un filtro di ricerca per "http".

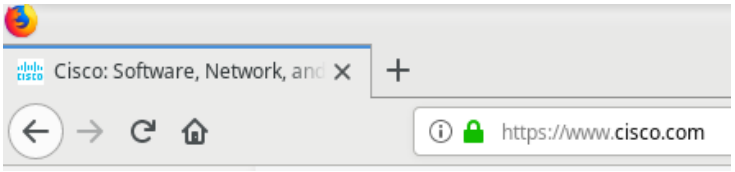


Scorriamo la lista di operazioni fino ad individuare una richiesta **POST**.

Cliccando su tale riga, nella tendina in basso ci vengono rilevate delle informazioni su tale operazione tra cui anche il nome utente e la password inseriti.



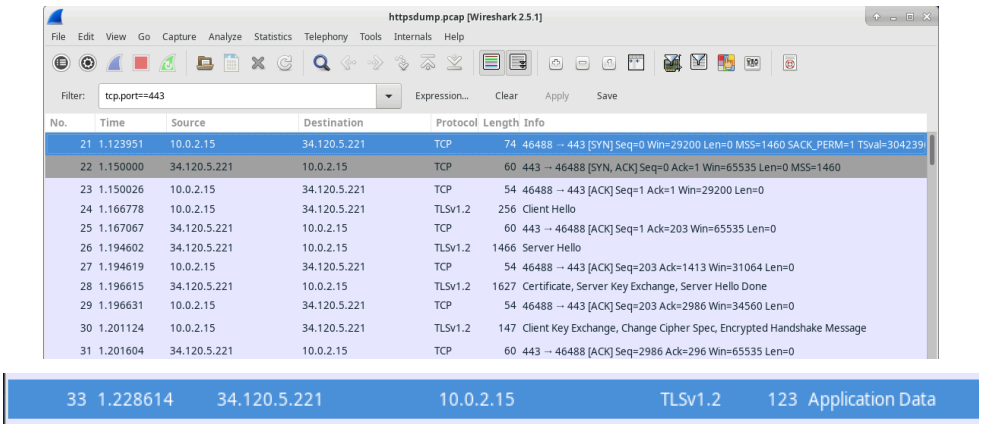
Passiamo ora ad analizzare il traffico relativo al protocollo **HTTPS**.
Per fare ciò, avviamo come prima una sessione di **tcpdump**, salvando la cattura nel file “httpsdump.pcap”, e ci colleghiamo a un sito web che utilizza il protocollo HTTPS.



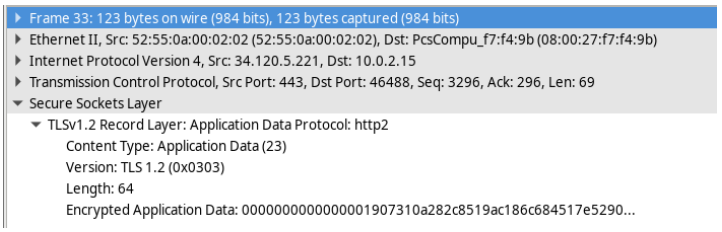
In questo caso il sito è “www.cisco.com” e la presenza del lucchetto accanto al nome indica l’utilizzo di un protocollo sicuro (HTTPS).
Come prima, effettuiamo il login sul sito e terminiamo la nostra cattura.

```
[analyst@secOps ~]$ sudo tcpdump -i enp0s3 -s 0 -w httpsdump.pcap
[sudo] password for analyst:
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 b
ytes
^C3018 packets captured
3018 packets received by filter
0 packets dropped by kernel
```

Apriamo ora la nostra nuova cattura con Wireshark e filtriamo i risultati per porta con il comando **tcp.port==443**. Successivamente individuiamo la riga relativa ad **Application Data**.



In questo caso possiamo notare che nella sezione protocollo troviamo la sigla **TLS1.2** invece di HTTP.
Inoltre, esaminando le informazioni relative a tale riga dal menù sottostante, possiamo notare come i dati non siano più visibili in quanto criptati.



Vantaggi di utilizzare HTTPS invece di HTTP:

1. **Crittografia**:
   HTTPS utilizza TLS (Transport Layer Security) per crittografare la comunicazione tra il browser e il server, impedendo a terzi di intercettare i dati trasmessi. Questo è fondamentale per proteggere informazioni sensibili come credenziali, dati bancari e dati personali.
2. **Autenticazione**:
   HTTPS garantisce che il sito web con cui stai comunicando è autentico e non un sito fasullo o di phishing. Infatti i certificati SSL/TLS verificano l'identità del sito, riducendo il rischio di attacchi "man-in-the-middle" (MITM).
3. **Integrità dei dati**:
   HTTPS assicura che i dati trasmessi non siano stati alterati o corrotti durante il trasferimento.

Tutti i siti web che utilizzano HTTPS sono considerati sicuri?
NO! Anche se HTTPS garantisce crittografia e autenticazione, non significa automaticamente che il sito sia affidabile o privo di pericoli.
Gli hacker possono ottenere certificati SSL validi e creare siti dannosi con HTTPS, sfruttando domini simili a quelli reali.
Inoltre, HTTPS non protegge da download dannosi, quindi anche siti con questo protocollo potrebbero contenere malware.