Progetto Finale di Primo Modulo

***Obiettivi***

L’obiettivo di questo progetto è quello di simulare, in un ambiente di laboratorio virtuale, un’architettura client server: Windows con indirizzo 192.168.32.101 richiede tramite web browser una risorsa allʼhostname epicode.internal che risponde all’indirizzo 192.168.32.100 Kali.

Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS. Ripetere l’esercizio, sostituendo il server HTTPS, con un server HTTP.

Si intercetti nuovamente il traffico, evidenziando le eventuali differenze tra il traffico appena catturato in HTTP ed il traffico precedente in HTTPS. Spiegare, motivandole, le principali differenze se presenti.

***FASE 1***

***Configurazione IP statico***

Il lavoro richiede una configurazione delle VM differente da quelle già preimpostate; perciò, come prima azione si configura: Kali con indirizzo IP 192.168.32.100 e Windows 192.168.32.101. Per avere la conferma delle modifiche si andrà a digitare nel terminale di Kali ip a, che, come si nota dall’immagine 1, è avvenuto con successo. Mentre con Windows si esegue il ping su sé stessa (Figura 2).

Immagine che contiene testo, software, schermata, computer

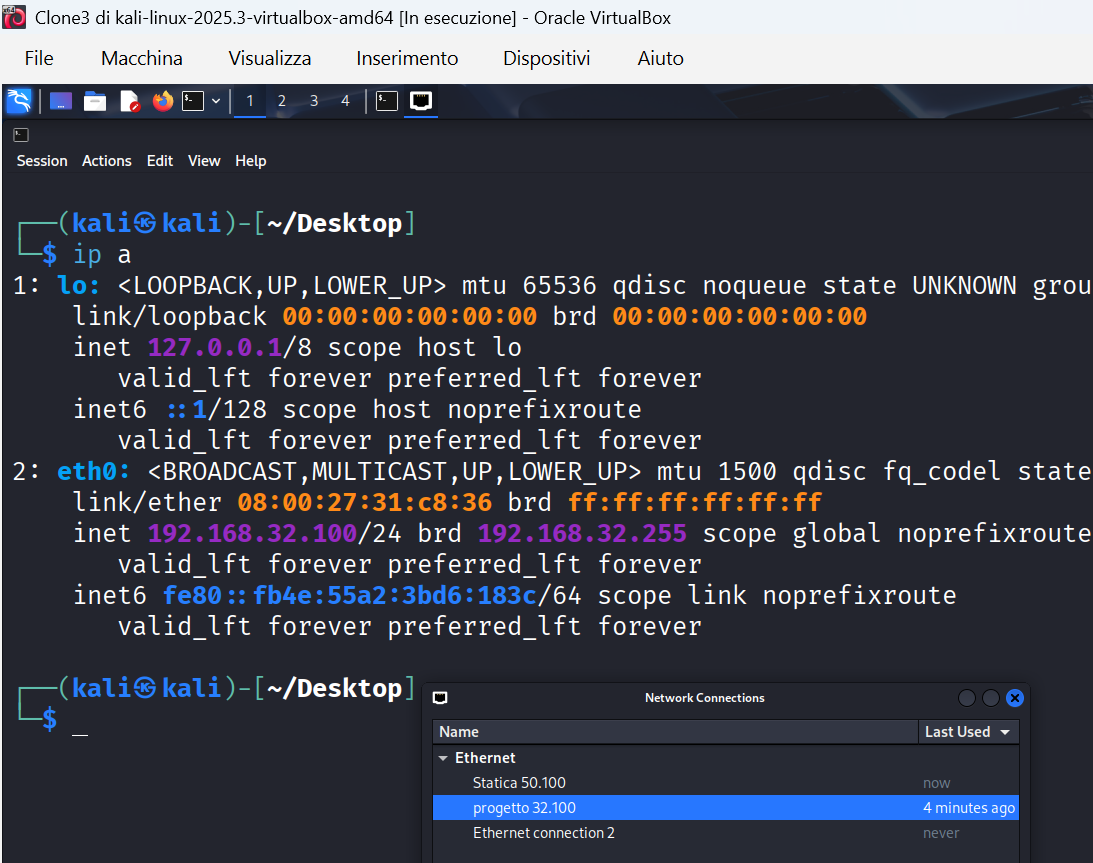
Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 1

Figura 2

***Servizio DNS***

Durante la configurazione del servizio DNS utilizzando Inetsim, è emersa una problematica nella quale il sistema ha restituito il seguente messaggio:

Immagine che contiene testo, software, Software multimediale, Pagina Web

Il contenuto generato dall'IA potrebbe non essere corretto.

Questo errore, scritto nel linguaggio di scripting Perl, indica che il programma sta tentando di richiamare un metodo denominato main\_loop sull'oggetto Net::DNS::Nameserver, che fa sì che l'esecuzione del servizio DNS su Inetsim venga interrotta, impedendo il corretto funzionamento.

Per affrontare il problema riscontrato durante la configurazione del servizio DNS tramite Inetsim, è stato necessario modificare l'impostazione di rete della VM Kali. In particolare, la scheda di rete è stata impostata in “Scheda con Bridge” per consentire la connessione a Internet. Successivamente è stato attivato il DHCP Automatico dal Networking Manager, così da ottenere automaticamente un indirizzo IP valido, dopo queste modifiche si è riavviata la macchina.

Una volta ristabilita la connettività su Kali, si è proceduto a scaricare la versione 1.37 di DNS, fondamentale per poter configurare adeguatamente il servizio DNS su InetSim. L'installazione è stata effettuata seguendo la sequenza dei seguenti comandi:

* perl Makefile.PL
* make
* make test
* sudo make install

Completata con successo l'installazione, è stato quindi possibile proseguire con la configurazione di Inetsim, si rimuovere il commento “#” da ogni servizio che si vuole abilitare, quindi da HTTP, HTTPS e DNS e si aggiunge l’indirizzo IP corretto su cui il servizio deve ascoltare

“service\_bind\_address 192.168.32.100”

Proseguendo nell’impostazioni del servizio DNS si imposta l’indirizzo IP di Kali come da consegna (198.162.32.100) e si aggiunge l’ultima sintassi evidenziata nella figura 3.

Immagine che contiene testo, schermata, software, computer

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 3

Si salva e si esce dalla directory e si avvia il servizio con sudo inetsim, e si provano i servizi cercando per prima su Mozilla Firefox. Come si legge dalla figura 4 , tutti i servizi necessari son attivi e funzionanti.

Immagine che contiene testo, schermata, software, computer

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 4. Servizio https KALI

Immagine che contiene testo, schermata, computer, software

Il contenuto generato dall'IA potrebbe non essere corretto.Immagine che contiene testo, schermata, computer, software

Il contenuto generato dall'IA potrebbe non essere corretto.Si configura il DNS anche su Windows attraverso il pannello di controllo e si inserisce l’indirizzo IP 198.168.32.100 come da consegna. Si avvia la ricerca di “Epicode.internal” su Google Chrome.

Figura 5. Servizio http WINDOWS

Figura 6. Servizio https WINDOWS

***FASE 2***

***Analisi del traffico su Wireshark***

Durante questa fase, si evidenzia l’identificazione dei MAC address di sorgente e destinazione. Per procedere con l’analisi, si avvia innanzitutto il servizio inetsim abilitando esclusivamente il protocollo HTTP. Una volta che il servizio è attivo, si dà inizio alla ricerca del traffico HTTP su Windows.

Immagine che contiene testo, elettronica, schermata, schermo

Il contenuto generato dall'IA potrebbe non essere corretto.Contestualmente all’avvio di inetsim, viene eseguito anche Wireshark per la cattura e l’ispezione dei pacchetti di rete. Grazie a Wireshark, è possibile monitorare in tempo reale le comunicazioni, evidenziando sia gli indirizzi MAC dei dispositivi che i protocolli coinvolti nella richiesta dell’hostname “epicode.internal”. Analizzando la figura 7, si osserva che i dati sono organizzati in colonne che indicano, nell’ordine: tempo, IP sorgente, IP destinatario, protocollo, lunghezza e ulteriori informazioni. Particolare attenzione va riservata al protocollo http, *HyperText Transfer Protocol*; selezionandolo, nella sezione inferiore sinistra dell’interfaccia vengono visualizzati tutti i dettagli specifici relativi allo stesso. L’area evidenziata in blu mostra il campo SRC, corrispondente all’indirizzo di sorgente e all’IP di destinazione. Contestualmente, sulla destra dello schermo viene visualizzato automaticamente il MAC address associato alla selezione corrente.

Figura 7

Prendendo sempre in considerazione la figura 7, si evince che http è in ascolto alla porta Dst Port:8, e continuando a leggere si trova la richiesta http: la prima parte è l’Header dove vi sono 1)il modulo GET, che ha lo scopo di recuperare una risorsa dal server; 2) l’host “epicode.internal” la risorsa richiesta; 3) keep-alive: questa direttiva indica che il client o il server desidera mantenere aperta la connessione per le richieste e le risposte successive; 4)Cache Control usate per gestire la memorizzazione delle risorse, ottimizzano le prestazioni web e limitando l'accesso ripetuto al server; 5) l’user-agent dà indicazione al server del programma lato client dal quale sta partendo la richiesta.

Immagine che contiene testo, elettronica, schermata, schermo

Il contenuto generato dall'IA potrebbe non essere corretto.***Si ripete l’esercizio attivando su inetsim unicamente i servizi HTTPS e DNS:***



Figura 8

È fondamentale precisare che il protocollo HTTPS non rappresenta una tecnologia distinta rispetto a http, piuttosto, è una variante HTTP che utilizza una crittografia specifica il Transport Layer Security (TLS) per proteggere le comunicazioni. TLS è un servizio ad uso generale implementato come un insieme di protocolli che si basano su TCP (Trasmission Control Protocol).

Nella figura 7 possiamo notare evidenziato nell’informazioni il messaggio Client\_Hello e la risposta Server\_hello: è Handshake Protocol che rappresenta la fase iniziale durante la quale il client e il server negoziano le modalità di crittografia. In particolare, consente alle parti di autenticarsi reciprocamente e concordare sia l'algoritmo di cifratura sia quello di MAC da utilizzare per una comunicazione sicura.



Perciò i dati inviati attraverso HTTPS avranno tre diversi livelli di protezione: Cifratura, Autenticazione e Integrità dei dati.

Il contenuto non è accessibile con strumenti come Wireshark, ma si possono comunque vedere i pacchetti TLS. La figura 7 mostra la presenza di TLSv1.3 nella colonna dei protocolli. Analizzando il dato in questione, si osserva che il MAC address sorgente è evidenziato in azzurro su entrambi i lati. Nella parte inferiore, viene riportato il protocollo TCP con porta sorgente 443 e porta di destinazione 49504.

Immagine che contiene testo, elettronica, schermata, schermo

Il contenuto generato dall'IA potrebbe non essere corretto.

Figura 9

Nella figura 9 possiamo notare più nello specifico il protocollo Handshake.

***FASE 3***

***Conclusioni***

Durante l’analisi è emersa una differenza rilevante nell’Header tra i protocolli HTTP e HTTPS. In particolare, il messaggio trasmesso tramite HTTPS risulta completamente cifrato, come si può osservare nell’area destra della figura 9. Al contrario, confrontando con la figura 7, appare evidente che, nel caso di HTTP, il messaggio rimane perfettamente leggibile e privo di cifratura.

Un’altra differenza sostanziale tra http e HTTPS molto evidente è come vi siano meno informazioni: come l’assenza di GET visto invece precedentemente in http, e l’assenza dell’hostname, della ricerca effettuata, quindi del contenuto vero e proprio.

In conclusione, tra HTTP e HTTPS vi sono differenze fondamentali in termini di sicurezza e affidabilità.