UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

DIPARTIMENTO DI SCIENZE FISICHE, INFORMATICHE E MATEMATICHE

CORSO DI LAUREA IN INFORMATICA

Fingerprinting tramite analisi di protocolli di rete e strategie di offuscamento

Relatore:
Luca Ferretti

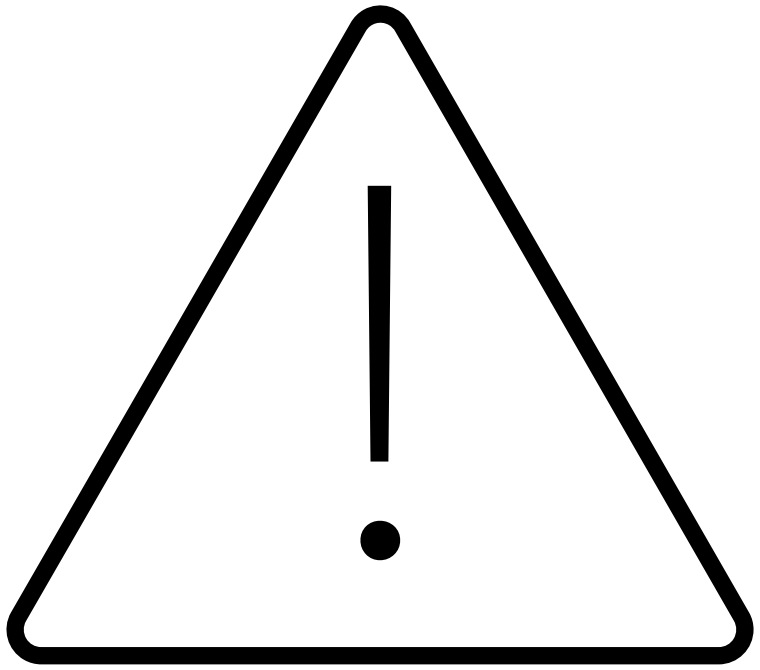
Candidato:
Fabio Zanichelli

ANNO ACCADEMICO 2021 - 2022

OS fingerprinting

«L'OS fingerprinting consiste nel rilevare da remoto il sistema operativo di un dispositivo analizzandone i pacchetti inviati.

Le differenze di implementazione dello stack TCP/IP, infatti, determinano comportamenti diversi che analizzati consentono di ottenere informazioni utili a questo scopo.»



Pericoli connessi al fingerprinting

- Privacy: è possible conoscere il sistema operativo in utilizzo
- Sicurezza: un attaccante che conosce il sistema operativo di un dispositivo ha un notevole vantaggio

Tool per il fingerprinting

NMAP

- Fingerprinting attivo
- Invia pacchetti specifici (*probe*) e analizza le risposte ricevute

P0F

- Fingerprinting passivo
- Analizza il traffico ordinario generato dal target

Obiettivi

- Analizzare le differenze di comportamento tra Windows 11 e Kali Linux
- Individuare meccanismi per ingannare i tool di fingerprinting
- Individuare le differenze tra i principali browser web rispetto all'header HTTP e l'handshake TLS

Principali differenze rilevate

	Windows 11	Kali Linux
Time To Live	128	64
Window Scaling	8	7
ICMP code in Echo Reply	0	Stesso valore della richiesta
ECN	0	1

Offuscamento: da Kali a Windows 10

- Modifica dei parametri del kernel per renderli simili a Windows
  → Modifiche al file /etc/sysctl.conf

- Manipolazione di pacchetti in uscita in base a determinati header
- Blocco pacchetti non diretti alla porta 80 (HTTP)
  → Utilizzo di nftables

Conclusioni OS fingerprinting

- L’offuscamento si può ottenere modificando le risposte ai pacchetti inviati dal tool che effettua il fingerprinting
    - Conoscere il tool utilizzato è un grosso vantaggio per l’ottenimento del punto precedente
- p0f mostra solo i fingerprinting esatti, per cui basta una piccola modifica per difendersi da esso

Analisi browser tramite protocollo HTTP

- Diverso ordine dei campi dell'header, dovuto al fatto che HTTP è un protocollo testuale
- Differenze campo *Accept,* che specifica i tipi di contenuti accettati
- Diversi *Quality Value,* un valore che indica la preferenza a determinate lingue

Protocollo TLS

- Utilizzato per garantire confidenzialità, integrità ed autenticità in una comunicazione
- Gli host si scambiano qualche messaggio di *handshake* prima di iniziare una comunicazione cifrata
- I messaggi dell'handshake, essendo in chiaro, sono facilmente analizzabili

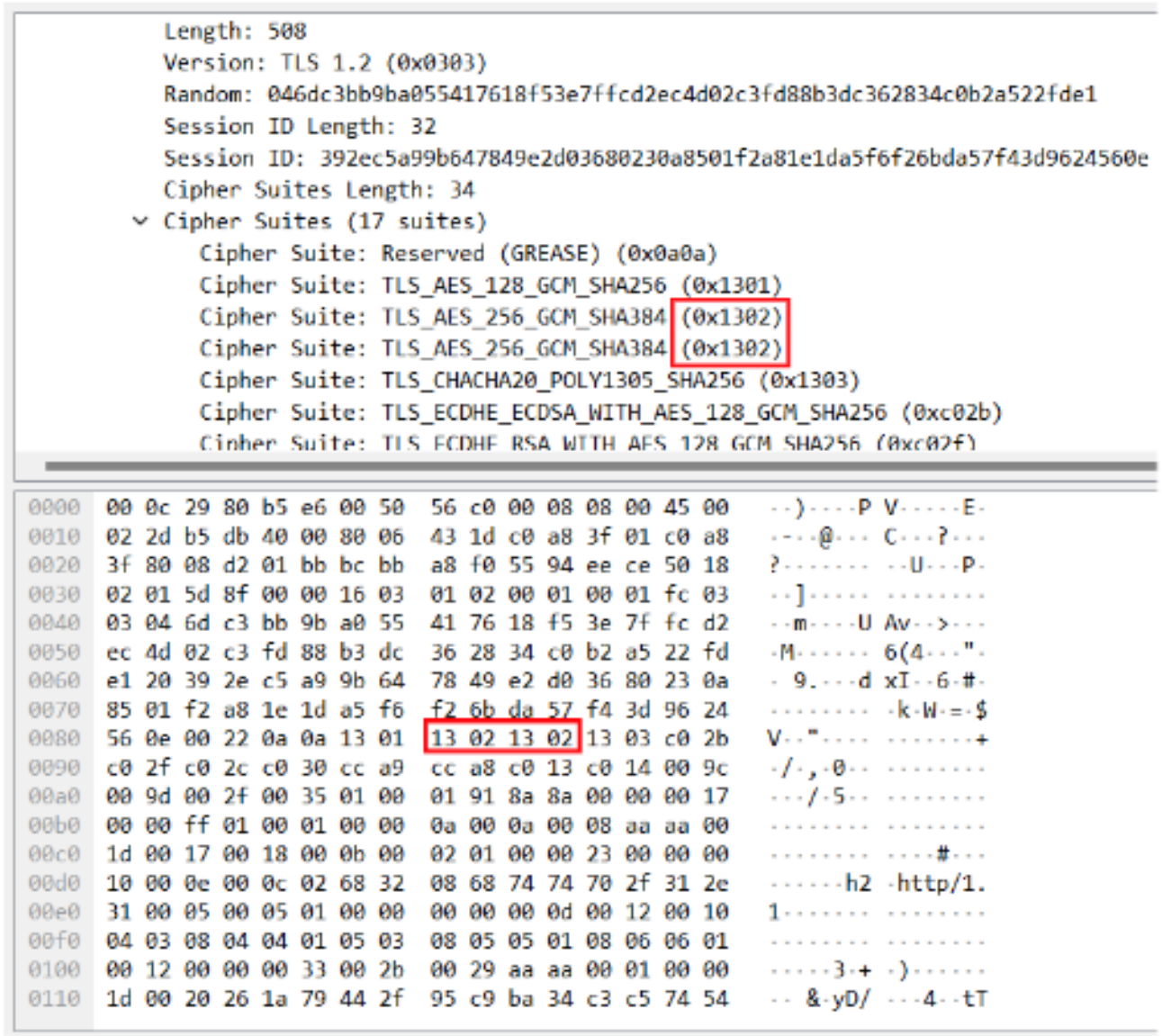
Differenze rilevate sull'handshake TLS

	Chrome	Firefox	Edge	Opera	Safari
Cifrari supportati	16	17	17	16	21
Algoritmi di hashing supportati	8	11	8	8	11
Utilizzo del GREASE	Sì	No	Sì	Sì	Sì

Inoltre, Firefox presenta i cifrari supportati in un ordine differente rispetto ai browser basati su Chromium

Cifrario ripetuto di Edge

- La ripetizione del cifrario non aggiunge dati utili all'handshake
- La medesima situazione avviene anche con gli algoritmi di hashing supportati da Safari



Conclusioni handshake TLS

- Modificare i cifrari supportati renderebbe meno evidente il fingerprinting del browser, ma potrebbe causare problemi di sicurezza o sulla compatibilità
- Chrome e Opera non presentano differenze
- L'eliminazione del cifrario ripetuto in Edge lo renderebbe indistinguibile da Chrome e Opera

Grazie per l'attenzione