

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

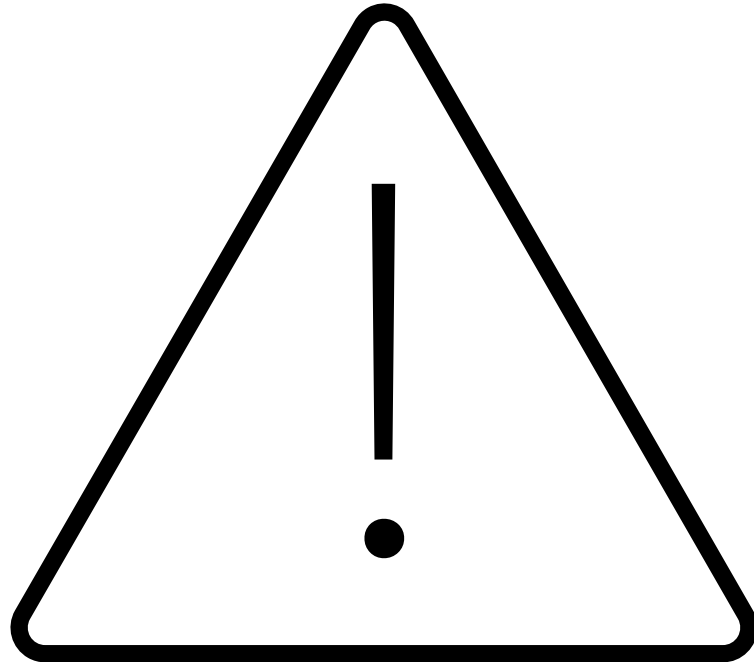
Candidato:

Fabio Zanichelli

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 possibile 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

POF

- 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

```
Length: 508
Version: TLS 1.2 (0x0303)
Random: 046dc3bb9ba055417618f53e7ffcd2ec4d02c3fd88b3dc362834c0b2a522fde1
Session ID Length: 32
Session ID: 392ec5a99b647849e2d03680230a8501f2a81e1da5f6f26bda57f43d9624560e
Cipher Suites Length: 34
  v Cipher Suites (17 suites)
    Cipher Suite: Reserved (GREASE) (0x0a0a)
    Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
    Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
    Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
    Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)

0000 00 0c 29 80 b5 e6 00 50 56 c0 00 08 08 00 45 00  --)---P V---E-
0010 02 2d b5 db 40 00 80 06 43 1d c0 a8 3f 01 c0 a8  ---@---C---?---
0020 3f 80 08 d2 01 bb bc bb a8 f0 55 94 ee ce 50 18  ?-----U---P-
0030 02 01 5d 8f 00 00 16 03 01 02 00 01 00 01 fc 03  --]-----
0040 03 04 6d c3 bb 9b a0 55 41 76 18 f5 3e 7f fc d2  --m---U Av-->---
0050 ec 4d 02 c3 fd 88 b3 dc 36 28 34 c0 b2 a5 22 fd  -M-----6(4---"
0060 e1 20 39 2e c5 a9 9b 64 78 49 e2 d0 36 80 23 0a  -9---d xI--6-#-
0070 85 01 f2 a8 1e 1d a5 f6 f2 6b da 57 f4 3d 96 24  -----k-W==-$
0080 56 0e 00 22 0a 0a 13 01 13 02 13 02 13 03 c0 2b  V--"-----+
0090 c0 2f c0 2c c0 30 cc a9 cc a8 c0 13 c0 14 00 9c  -/.,-0- -----
00a0 00 9d 00 2f 00 35 01 00 01 91 8a 8a 00 00 00 17  ---/5- -----
00b0 00 00 ff 01 00 01 00 00 0a 00 0a 00 08 aa aa 00  -----
00c0 1d 00 17 00 18 00 0b 00 02 01 00 00 23 00 00 00  -----#---
00d0 10 00 0e 00 0c 02 68 32 08 68 74 74 70 2f 31 2e  -----h2 -http/1.
00e0 31 00 05 00 05 01 00 00 00 00 00 0d 00 12 00 10  1-----
00f0 04 03 08 04 04 01 05 03 08 05 05 01 08 06 06 01  -----3+ -)-----
0100 00 12 00 00 00 33 00 2b 00 29 aa aa 00 01 00 00  -----
0110 1d 00 20 26 1a 79 44 2f 95 c9 ba 34 c3 c5 74 54  --&-yD/ ---4- tT
```

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