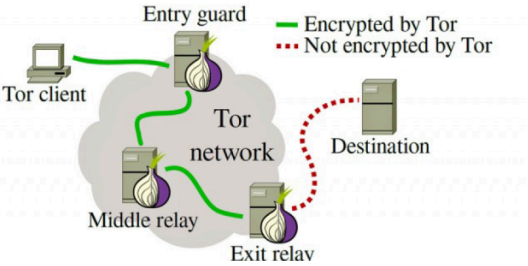
HOW WAS THE TOR ATTACKER CAUGHT?

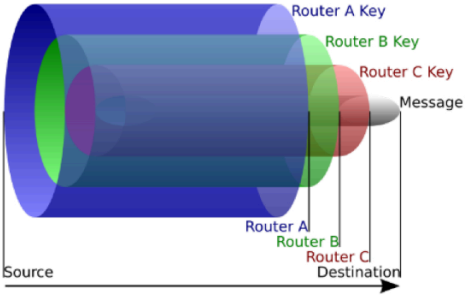
INTRODUCTION

Most people know that **The Onion Router (Tor)** network is the basic building block of the Dark Web. Originally developed by the US Navy to hide their online communications, Tor is now available to everyone as an open source platform through the Tor Project.

Most users tend to think that their online activities are completely anonymous and secure when using the Tor network. However, many people who assume that they are anonymous on this network can make fatal mistakes by ignoring basic security measures and leave traces that can reveal their identities.

How did I catch the attacker from these traces as a white hat?

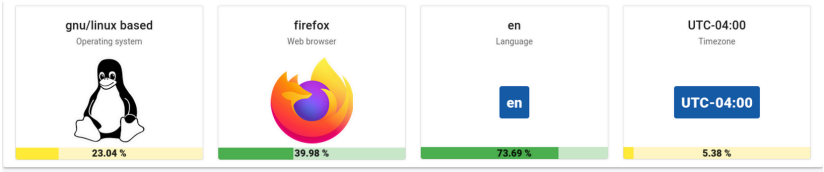




Websites and intelligence agencies have been using cookies to identify users for years. Even if users who seek anonymity disable cookies, they can still be tracked using more advanced techniques like browser fingerprinting. **This method analyzes more than 50 variables, such as browser and device characteristics, to uniquely identify users**.

The combination of these variables is enough to eliminate anonymity, offering a greater number of unique possibilities than the world's population. Law enforcement can use traffic fingerprinting to reveal users' identities. **I analyzed these methods and went after the attacker.**

For example, here are some of **the fingerprint elements of my default browser in Kali**.



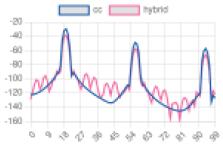
HTTP HEADERS ATTRIBUTES

Search for an attribute

Attribute	Similarity ratio	Value
1 - User agent	0.39 %	Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
2 - Accept	28.58 %	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
3 - Content encoding	87.40 %	gzip, deflate, br
4 - Content language	31.77 %	en-US,en;q=0.5
5 - Upgrade Insecure Requests	90.68 %	1

Search for an attribute

Attribute	Similarity ratio	Value
1 - User agent	0.90 %	Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
2 - Platform	25.67 %	Linux x86_64
3 - Cookies enabled	90.46 %	✓
4 - Timezone	5.38 %	UTC-04:00
5 - Content language	41.38 %	en-US,en
6 - Canvas	0.11 %	Cwm fjordbank glyphs vext quiz,
7 - List of fonts (JS)	0.01 %	Arial, Arial Narrow, Bitstream Charter, Bitstream Vera Sans, Bookman Old Style, And 36 others
8 - Use of Adblock	63.62 %	✗
9 - Do Not Track	65.05 %	✗
10 - Navigator properties	2.31 %	42 properties detected
11 - BuildID	34.67 %	20181001000000
12 - Product	91.18 %	Gecko
13 - Product sub	34.95 %	20100101
14 - Vendor	35.30 %	No value
15 - Vendor sub	91.37 %	No value

26 - Screen available width	11.10 %	1280
27 - Screen left	34.82 %	0
28 - Screen top	35.82 %	0
29 - Permissions	9.29 %	**accelerometer** : Not supported **accessibility** : Not supported **ambient-light-sensor** : Not supported **camera** : Not supported **clipboard-read** : Not supported **clipboard-write** : Not supported **geolocation** : prompt **background-sync** : Not supported **magnetometer** : Not supported **microphone** : Not supported **midi** : prompt **notifications** : prompt **payment-handler** : Not supported **persistent-storage** : prompt **push** : prompt
30 - WebGL Vendor	0.25 %	Mesa
31 - WebGL Renderer	8.28 %	llvmpipe
32 - WebGL Data	1.06 %	
33 - WebGL Parameters	0.22 %	28 different extensions 25 different general parameters analyzed 36 different shaders precisions analyzed
34 - Use of local storage	90.12 %	✓
35 - Use of session storage	90.21 %	✓
36 - Use of IndexedDB	91.02 %	✓
37 - Audio formats	26.87 %	**audio/aac** : maybe **audio/flac** : maybe **audio/mpeg** : maybe **audio/ogg; codecs="flac"** : probably **audio/ogg; codecs="vorbis"** : probably **audio/ogg; codecs="opus"** : probably **audio/wav; codecs="1"** : probably **audio/webm; codecs="vorbis"** : probably **audio/webm; codecs="opus"** : probably **audio/mp4; codecs="mp4a.40.2"** : probably
38 - Audio context	13.26 %	**sampleRate** : 48000 **state** : suspended
39 - Frequency analyser	76.23 %	**channelCount** : 2 **channelCountMode** : max **channelInterpretation** : speakers **fftSize** : 2048 **frequencyBinCount** : 1024 **maxDecibels** : -30 **minDecibels** : -100 **numberOfInputs** : 1 **numberOfOutputs** : 1 **smoothingTimeConstant** : 0.8
40 - Audio data	0.62 %	
41 - Video formats	67.60 %	**video/mp4; codecs="flac"** : probably **video/ogg; codecs="theora"** : probably **video/ogg; codecs="opus"** : probably **video/webm; codecs="vp9, opus"** : probably **video/webm; codecs="vp8, vorbis"** : probably

WHAT IS BROWSE FINGERPRINTING?

Browser fingerprinting is the collection of parameters that your browser sends to a website that can be used to uniquely identify you. This technique uses **browser and device features to track the user without the need for cookies and consists of more than 50 variables**.

What is Canvas Fingerprinting?

How Does It Work?

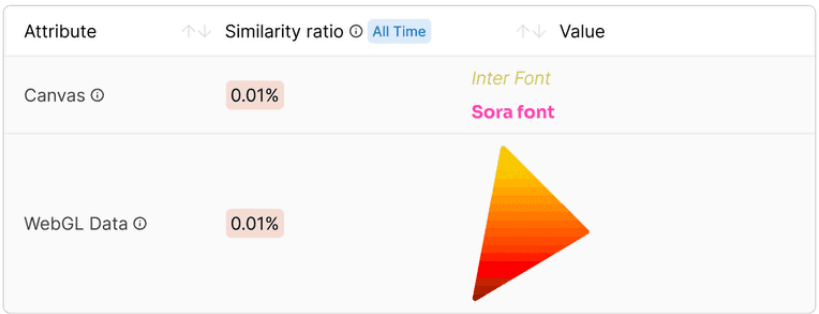
A Canvas fingerprint is a unique identifier created using the HTML5 Canvas element. A web page draws text or graphics on the canvas, specifying various attributes such as font size and color. The drawn content is then converted into a Base64 encoded string, and this data is combined into a unique fingerprint using a hashing algorithm.

What is WebGL Fingerprinting?

How Does It Work?

WebGL fingerprinting is a technique that uses the graphics hardware features of users' devices to create unique identifiers. This method takes advantage of the GPU's processing capacity, leaving a unique trace that can persist across different browser sessions and websites.

1. Creating a WebGL Context: When a site that uses WebGL is accessed, the browser creates a WebGL context to communicate directly with the GPU and begins managing graphics rendering tasks.



2. Rendering Graphics: The website asks the browser to render graphic elements such as complex 3D shapes or patterns. This rendering process varies depending on factors such as the device's graphics card, driver version, and browser.

3. Capturing and Hashing Data: The rendered graphics are analyzed and the results are processed with a hashing algorithm to create a unique fingerprint that reflects the device's graphics processing capacity.

4. Storage and Tracking: The resulting WebGL fingerprint is stored on the server. Thus, the website can compare newly created fingerprints with previously stored ones to identify returning users.

Factors That Determine Uniqueness

- Graphics Card Types: Different GPUs render images slightly differently due to hardware differences.

- Driver Versions: Updates to graphics drivers can change rendering methods.

- Browser Differences: Each browser implements WebGL differently, which can affect rendering results.

- OS Effects: The operating system itself can also affect how graphics are rendered.

In summary, WebGL fingerprinting provides a powerful way to **track users by taking advantage of different aspects of the graphics hardware**.

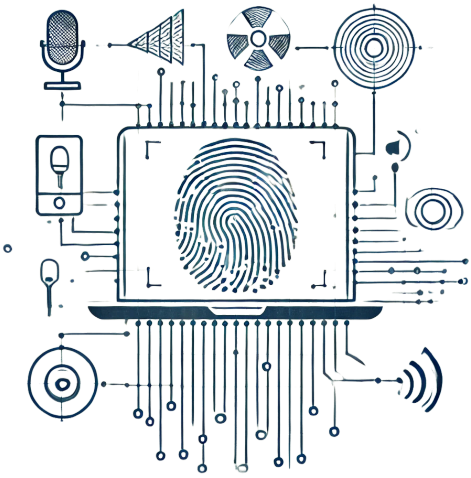
What is Media Device Fingerprinting?

How Does It Work?

Media device fingerprinting is a method of creating unique identifiers based on the audio and video capabilities of devices. This technique allows devices to be **permanently identified even when traditional tracking methods such as cookies are ineffective**.

1. Data Collection: A website or application using media device fingerprinting collects data about the media features of the device. This may include supported **audio and video codecs, resolutions, hardware details such as microphone, camera, and operating system/browser used**.

2. Creating a Unique Identifier: The information collected is processed to create a fingerprint that is unique to the device. This fingerprint consists of unique combinations of the device's media capabilities and configurations.



3. Tracking Mechanism: The generated fingerprint is stored on the server and the user can be identified by comparing it with the new fingerprint when the same device is reconnected.

Media device fingerprinting offers a powerful solution to **track the user by using small differences in the device's hardware and software configurations**.

What is TLS Fingerprinting? How Does It Work?

TLS fingerprinting is a technique that allows clients to be identified based on details in the "Client Hello" message during the TLS (Transport Layer Security) handshake. This method uses features specific to the client's TLS implementation to identify users even when traditional tracking methods such as cookies are ineffective.

1. TLS Handshake: The process begins with the client requesting a secure connection to the server. During this handshake, cryptographic parameters are exchanged.

2. Client Hello Message: The client includes the following information in the **"Client Hello"** message:

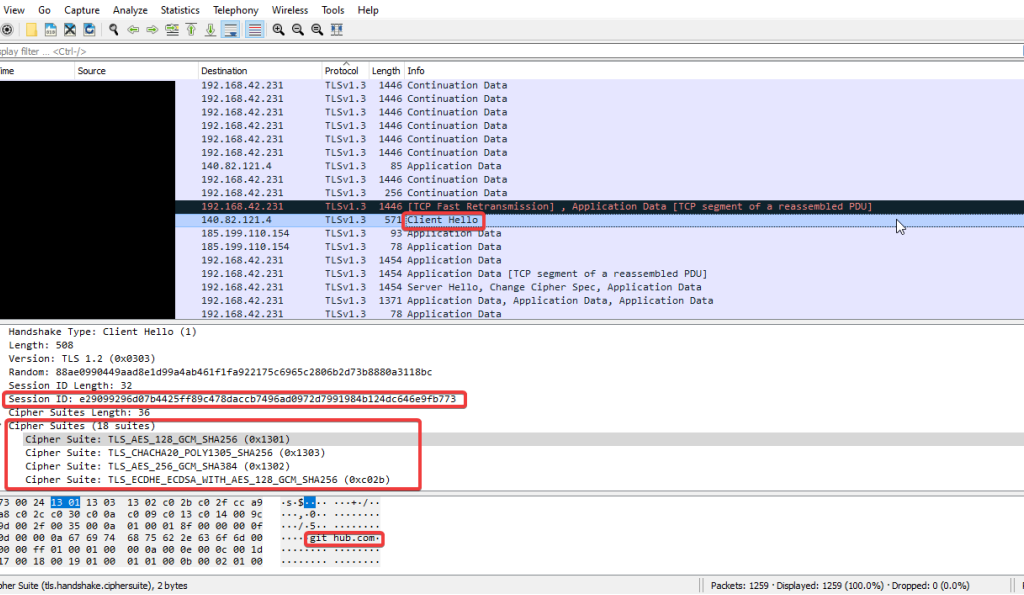
Supported Cipher Suites: List of encryption algorithms in order of preference.

TLS Version: The TLS protocol version to be used.
Extensions: Additional parameters that can enhance the connection, such as SNI.

3. Fingerprinting: By analyzing the information in this message, a fingerprint specific to the client's TLS library and cipher suites is created.

4. Hashing: Using formats such as JA3, the collected data is then combined into an MD5 hash, providing a compact representation of the fingerprint.

In summary, TLS fingerprinting provides a powerful method **for tracking users using client-specific TLS features**.



What is Font Fingerprinting? How Does It Work?

Font fingerprinting is a tracking method to identify users based on the fonts installed on a device. It allows tracking users by exploiting the uniqueness of font combinations across systems.

1. Detecting Installed Fonts: The website detects the fonts installed on the device through scripts running in the background. Several techniques can be used for this process.

- JavaScript and CSS: Rendering text in different fonts and measuring their size.
- Canvas API: Rendering text with fonts using HTML5 Canvas and measuring the results.
- Font Enumeration: Listing installed fonts using JavaScript.

2. Generating a Unique Identifier: By analyzing the detected fonts, a fingerprint specific to the user's device is created. This fingerprint is unique enough to distinguish the user across different web sessions and sites.

3. Tracking Mechanism: The generated fingerprint is stored on servers so that the website can recognize returning users without the need for cookies.

- In summary, font fingerprinting provides an effective and persistent method to track users using the device's font configuration.

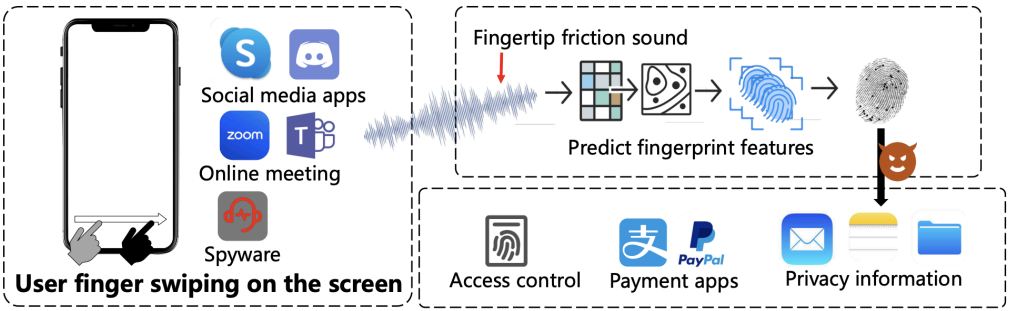
```
// Canvas font fingerprinting script.
Fonts = ["monospace" , ... , "sans-serif"];

CanvasElem = document.createElement("canvas");
CanvasElem.width = "100";
CanvasElem.height = "100";
context = CanvasElem.getContext('2d');
FPDict= {};
for (i = 0; i < Fonts.length; i++)
{
  CanvasElem.font = Fonts[i];
  FPDict[Fonts[i]] = context.measureText("example
      ").width;
}
```

What is Audio Fingerprinting?

How Does It Work?

Audio fingerprinting is a technique used to identify and track audio recordings. This method uses the acoustic properties of the audio signal (such as waveform, spectral envelope, pitch, harmonics) to create a unique digital signature.

1. Extraction of Perceptual Features: Acoustic details such as waveform, pitch, and spectral features are extracted from the audio signal.



2. Condensation into Digital Summary: These features are condensed to form a compact digital summary, called an "acoustic fingerprint".

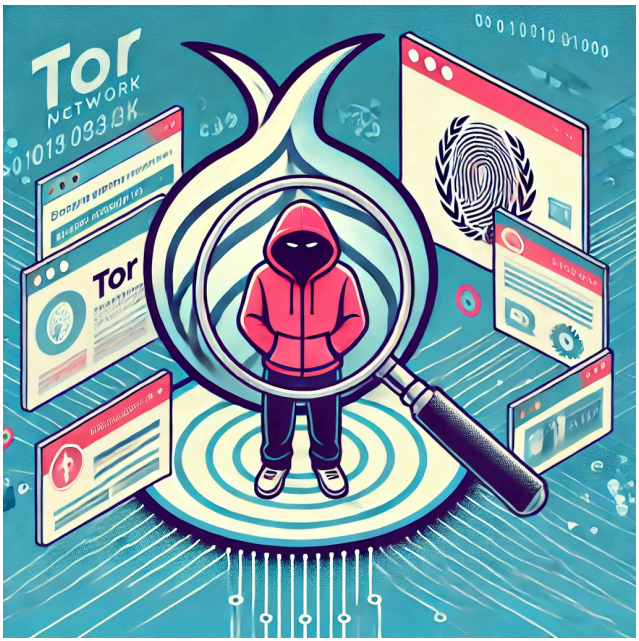
3. Database Comparison: The acoustic fingerprint created is compared to a database of previously calculated fingerprints to identify the sound.

Features of Audio Fingerprinting: Resistant to compression, noise, or transmission errors.

- Tolerant to small sound changes.
- Provides rapid identification in large databases**.**

CONCLUSION

Tools like Tor or VPN can help users remain anonymous by hiding their IP address. However, advanced techniques like browser fingerprinting provide an effective way to identify users even when cookies are disabled. Detailed data transmitted from the browser and device can be unique enough to reveal a user's identity. Therefore, it is possible to track down attackers using browser fingerprinting techniques.



In a **real-life incident**, while investigating a security breach experienced by our client, we realized that the attacker was trying to hide his identity through the Tor network. However, with our detailed analysis and browser fingerprinting techniques, we were able to identify the attacker. With this method, we were able to track the attacker down to his home address despite him using the Tor network.

In conclusion, even Tor, which is considered an anonymity tool, has some weaknesses. With advanced tracking techniques, even Tor users can be identified. **Anonymity may not be as strong as it seems**.