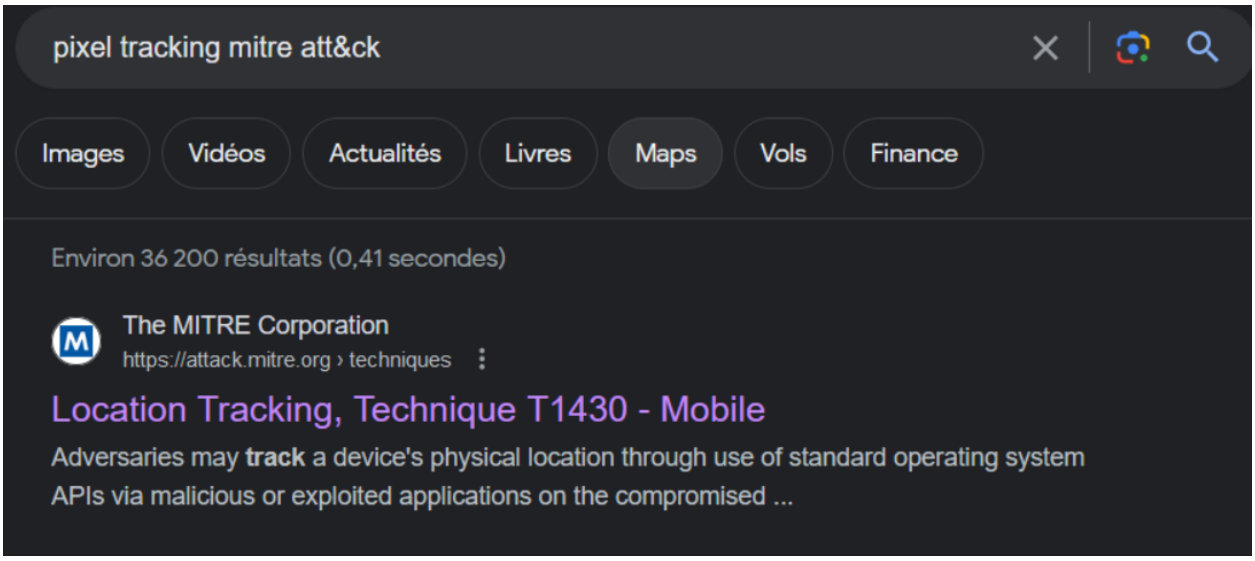
WriteUp

```
import email
from email import policy
from email.parser import BytesParser

def extract_body_from_eml(eml_file_path):
    with open(eml_file_path, 'rb') as eml_file:
        # Parse the .eml file
        msg = BytesParser(policy=policy.default).parse(eml_file)

        # Extract the body
        if msg.is_multipart():
            # If the message is multipart, iterate over the parts and find the text/html part
            for part in msg.iter_parts():
                if part.get_content_type() == 'text/html':
                    # Use a default encoding (e.g., 'utf-8') if get_content_charset() returns None
                    charset = part.get_content_charset() or 'utf-8'
                    return part.get_payload(decode=True).decode(charset, 'ignore')
        else:
            # If the message is not multipart, return the plain text body
            # Use a default encoding (e.g., 'utf-8') if get_content_charset() returns None
            charset = msg.get_content_charset() or 'utf-8'
            return msg.get_payload(decode=True).decode(charset, 'ignore')

# Example usage
eml_file_path = 'Forensic/Pixel Perfect/ressources/La prise de l EPUUBS.msg'
body = extract_body_from_eml(eml_file_path)
print(body)
```

```
                                                                                Qy:n e;"A!@QdKfU
                                                                                                {HYPERLINK OoPz`UL A?CEHFoGHg5

Je me balade dans l'cole et je bois mon caf.

Et la je croise un lve, y mdit : Eh o vous z'avez trouvez votre caf ?

Rien  foutre, Moi J chuis l directeur et j t'emmeeeerde



Oups !

Laboulette

 <https://imgur.com/a/qBwfK9k>
```

On voit un url qui renvoie vers un pixel host, technique utiliser par les commerciaux et les spammer pour tracker l’ouverture des emails



Home > Techniques > Mobile > Location Tracking

Location Tracking

Sub-techniques (2)

Adversaries may track a device's physical location through use of standard operating system APIs via malicious or exploited applications on the compromised device.

On Android, applications holding the `ACCESS_COAURSE_LOCATION` or `ACCESS_FINE_LOCATION` permissions provide access to the device's physical location. On Android 10 and up, declaration of the `ACCESS_BACKGROUND_LOCATION` permission in an application's manifest will allow applications to request location access even when the application is running in the background.[1] Some adversaries have utilized integration of Baidu map services to retrieve geographical location once the location access permissions had been obtained.[2][3]

On iOS, applications must include the `NSLocationWhenInUseUsageDescription`, `NSLocationAlwaysAndWhenInUseUsageDescription`, and/or `NSLocationAlwaysUsageDescription` keys in their `Info.plist` file depending on the extent of requested access to location information.[4]

ID: T1430

Sub-techniques: T1430.001, T1430.002

Tactic Type: Post-Adversary Device Access

Tactics: Collection, Discovery

Platforms: Android, iOS

MTC ID: APP-24

Version: 1.2

Created: 25 October 2017

Last Modified: 20 March 2023

Version Permalink