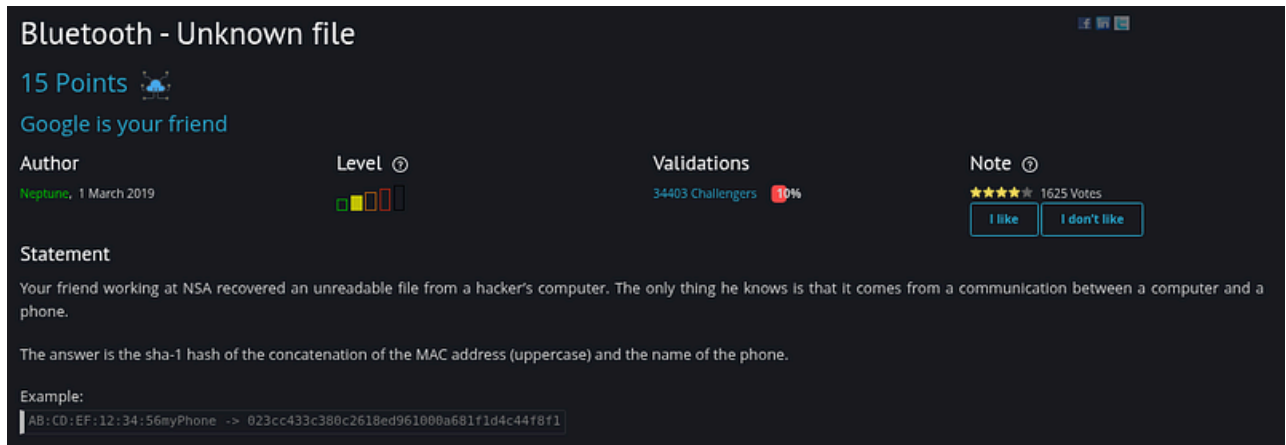


Root-Me Write-up: Bluetooth—Unknown File



Challenge Statement

A friend working at the NSA recovered an **unreadable file** from a hacker's computer. The only information available is that the file originates from a **communication between a computer and a phone**.

The objective is to find:

The SHA-1 hash of the concatenation of the MAC address (uppercase) and the phone name.

Hint: Bluetooth—Unknown file

Initial Analysis

The first step was to identify the file type.

```
(loki@SolarisFortress)-[~/Downloads]
$ hexedit ch18.bin

zsh: suspended  hexedit ch18.bin

(loki@SolarisFortress)-[~/Downloads]
$ file ch18.bin
ch18.bin: BTSnoop version 1, HCI UART (H4)

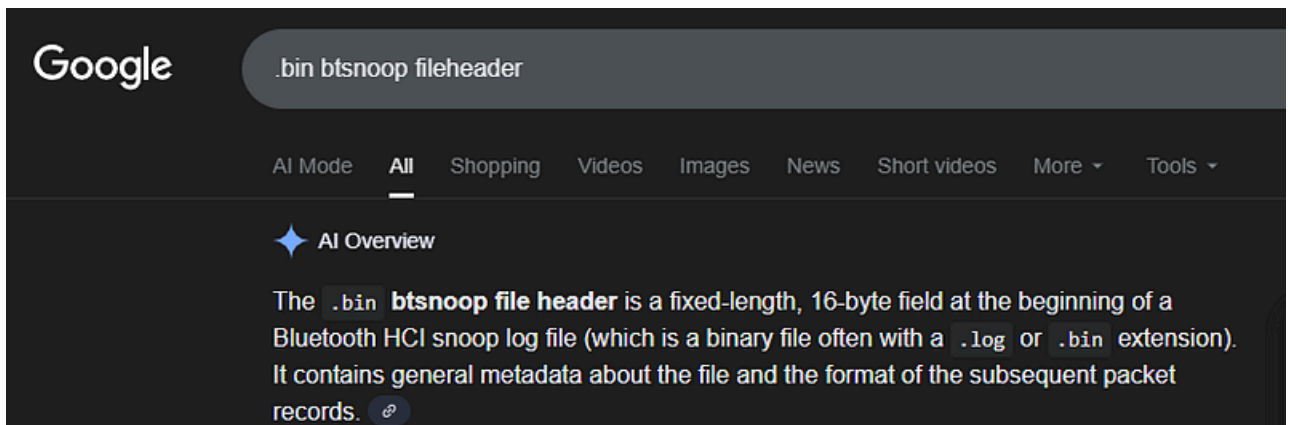
(loki@SolarisFortress)-[~/Downloads]
$
```

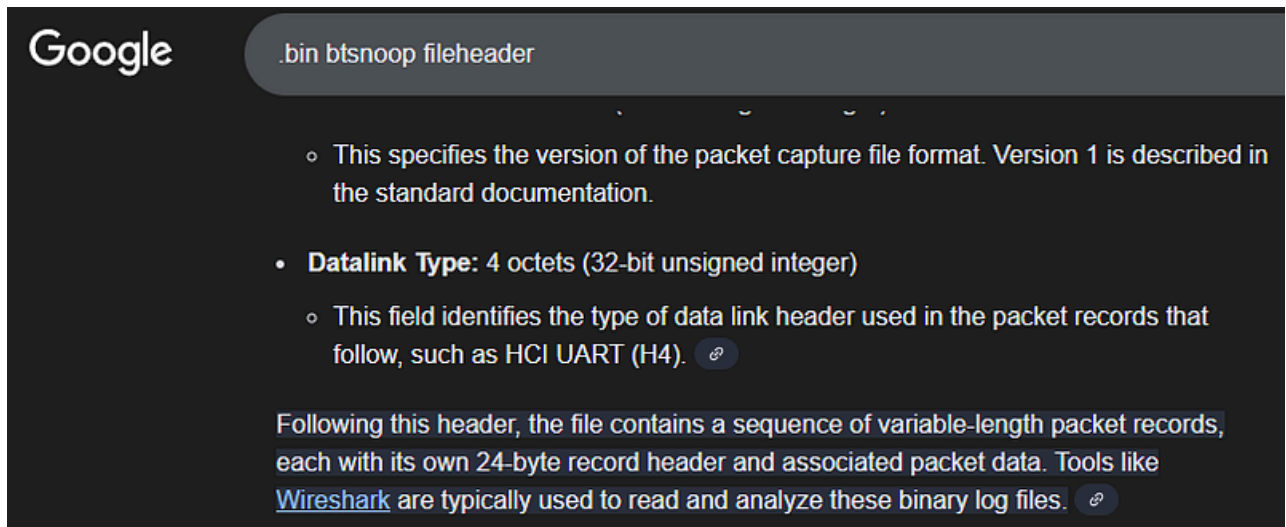
File Identification

I started by running the `file` command on the provided file to determine its format. Since the output was not immediately clear, I opened the file using a **hex editor** to manually inspect the **file header and footer**.

From the header, a clear signature appeared indicating:

BTSnoop





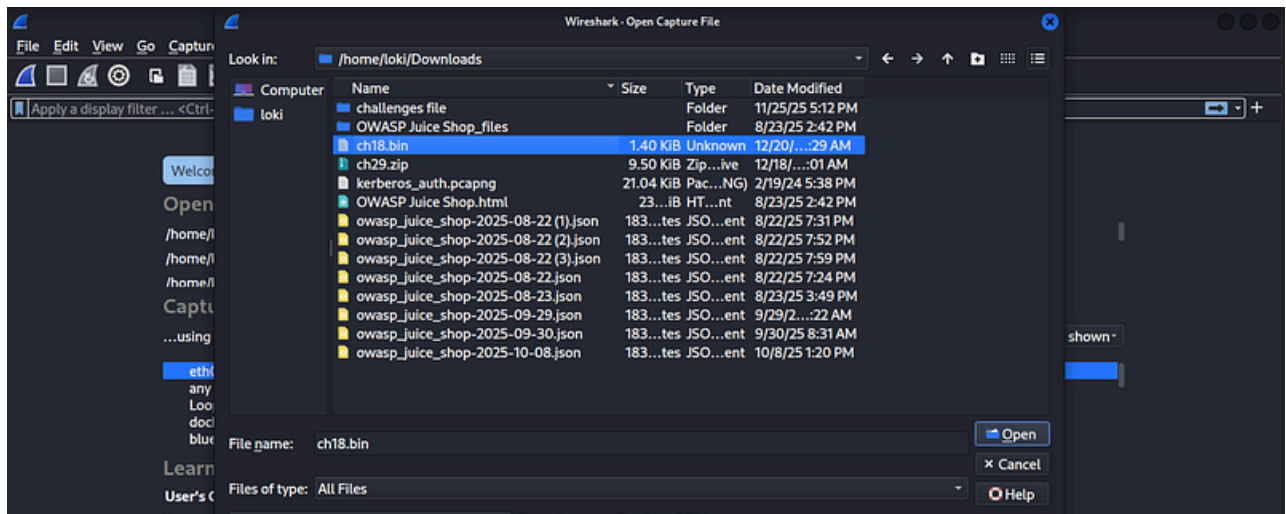
What is a BTSnoop File?

A **BTSnoop file** is a Bluetooth packet capture format used to log **Bluetooth HCI (Host Controller Interface) traffic**. These files are commonly generated when debugging Bluetooth communications and can be analyzed using network forensic tools such as **Wireshark**.

This confirmed that the recovered file was a **Bluetooth traffic capture**.

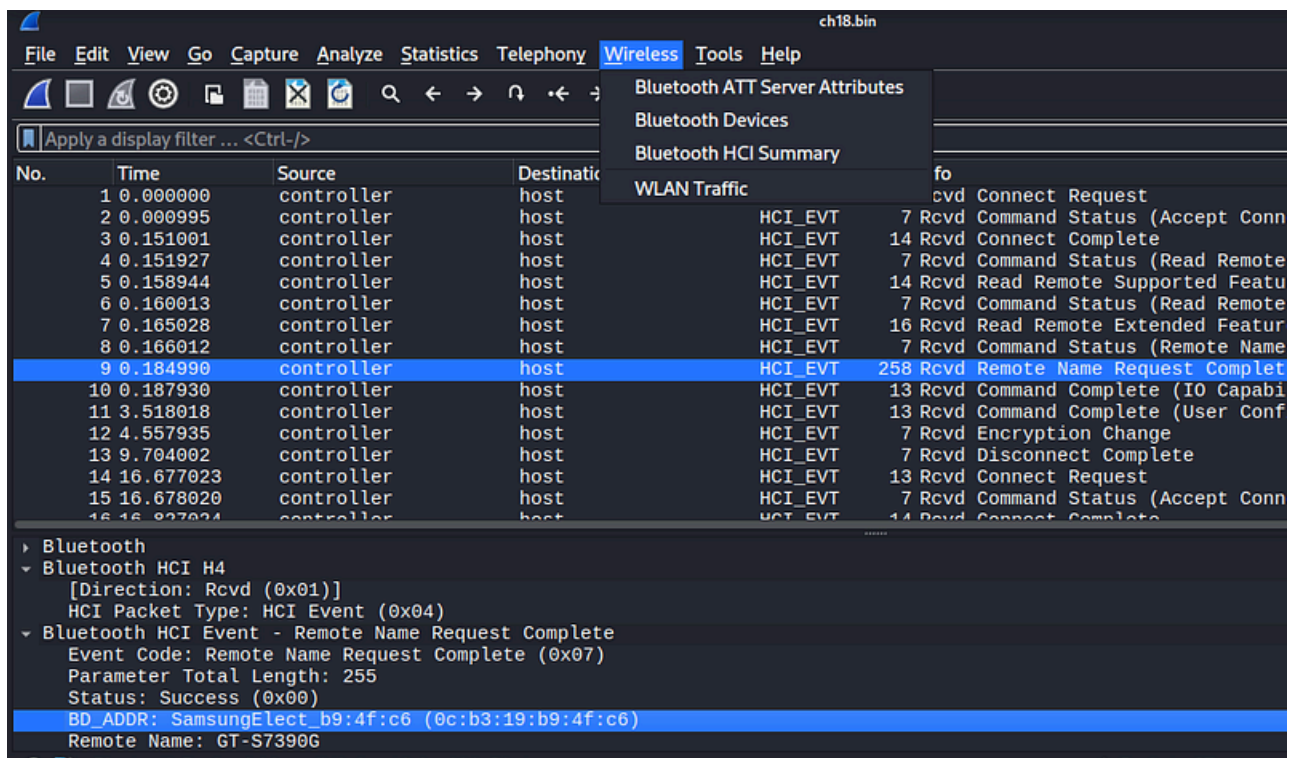
Traffic Analysis with Wireshark

With this information, I opened the file directly in **Wireshark**, which has built-in support for BTSnoop logs.



Once loaded:

- The Bluetooth packets and events were displayed correctly
- Individual packets could be inspected



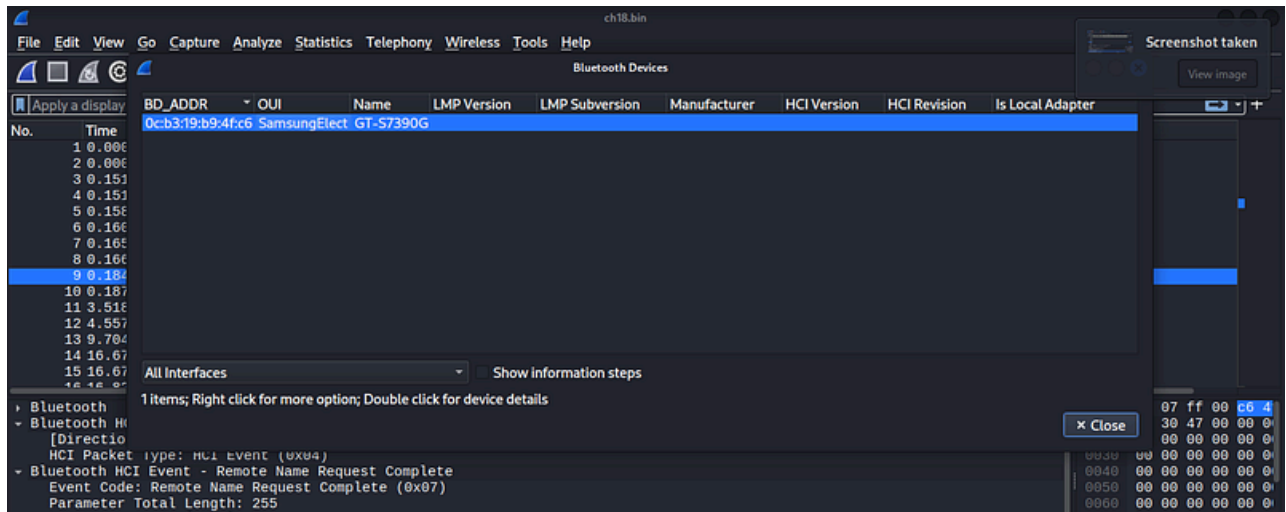
To extract the required information, I navigated to:

Wireless → Bluetooth Devices

This panel lists all discovered Bluetooth devices along with:

- MAC address (BD_ADDR)

- Device name



Extracted Information

From the Bluetooth Devices list, the following values were identified:

BD_ADDR: 0c:b3:19:b9:4f:c6

Device Name: GT-S7390G

Flag Construction

The challenge specifies that the flag is generated as:

SHA1(MAC_ADDRESS_UPPERCASE + DEVICE_NAME)

Step 1: Uppercase the MAC Address

0C:B3:19:B9:4F:C6

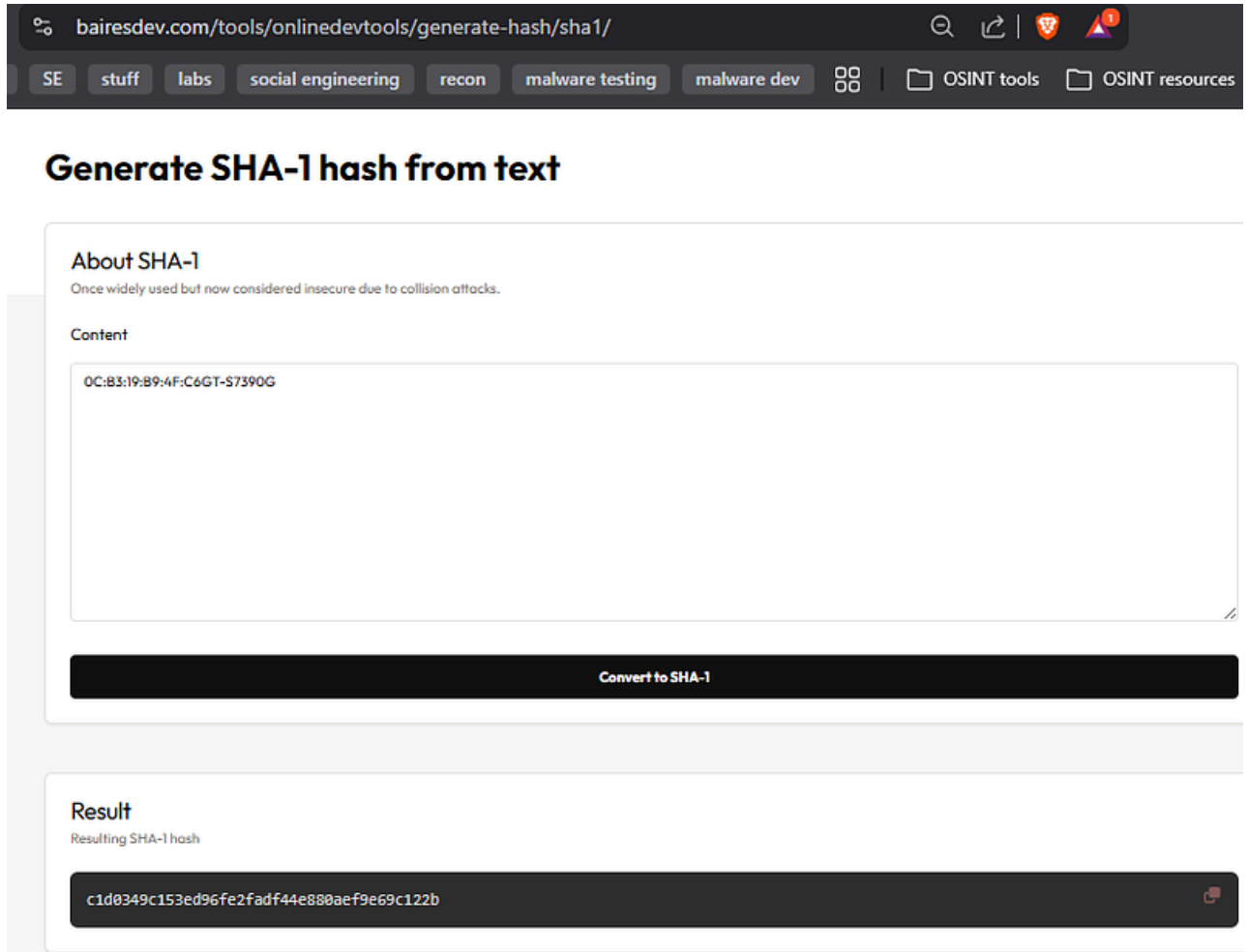
Step 2: Concatenate with the Device Name

0C:B3:19:B9:4F:C6GT-S7390G

Step 3: Generate SHA-1 Hash

The concatenated string was hashed using a SHA-1 generator.

<https://www.bairesdev.com/tools/online-devtools/generate-hash/sha1/>



The screenshot shows a web browser window with the URL `bairedv.com/tools/online-devtools/generate-hash/sha1/`. The browser's address bar and tabs are visible. The page has a dark header with navigation links: `SE`, `stuff`, `labs`, `social engineering`, `recon`, `malware testing`, `malware dev`, and icons for `OSINT tools` and `OSINT resources`. The main heading is **Generate SHA-1 hash from text**. Below this, there is a section titled **About SHA-1** with the text: "Once widely used but now considered insecure due to collision attacks." Underneath is a **Content** label and a large text input field containing the string `0C:B3:19:B9:4F:C6GT-S7390G`. A black button labeled **Convert to SHA-1** is positioned below the input field. The **Result** section shows the "Resulting SHA-1 hash" as `c1d0349c153ed96fe2fadf44e880aef9e69c122b` in a dark box.

Final Flag

SHA1(0C:B3:19:B9:4F:C6GT-S7390G)

By [Alexander Sapo](#) on [December 20, 2025](#).

[Canonical link](#)

Exported from [Medium](#) on February 7, 2026.