

# Root-Me Write-up: Bluetooth—Unknown File

The screenshot shows a challenge card for 'Bluetooth - Unknown file'. It has a dark background with white text. At the top left is the challenge title. Below it are '15 Points' and a small icon. A 'Google is your friend' link is present. On the left, there's an 'Author' section with 'Neptune' and the date '1 March 2019'. In the center, there's a 'Level' section with a progress bar showing four colored segments (green, yellow, orange, red). To the right, 'Validations' show '34403 Challengers' and '10%' completion. A 'Note' section includes a rating of 4 stars from 1625 votes, with 'I like' and 'I don't like' buttons. Below these sections is a 'Statement' block containing text about NSA recovery and a phone hash example.

## 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

.bin btsnoop fileheader

AI Mode All Shopping Videos Images News Short videos More Tools

◆ AI Overview

The **.bin btsnoop file header** is a fixed-length, 16-byte field at the beginning of a Bluetooth HCI snoop log file (which is a binary file often with a **.log** or **.bin** extension). It contains general metadata about the file and the format of the subsequent packet records.

The screenshot shows a Google search results page. The search query is ".bin btsnoop fileheader". The first result is a snippet from a document explaining the structure of a BT-Snoop file header. It includes a bulleted list about the Datalink Type field and a note about Wireshark being used to analyze such files.

Following this header, the file contains a sequence of variable-length packet records, each with its own 24-byte record header and associated packet data. Tools like [Wireshark](#) are typically used to read and analyze these binary log files.

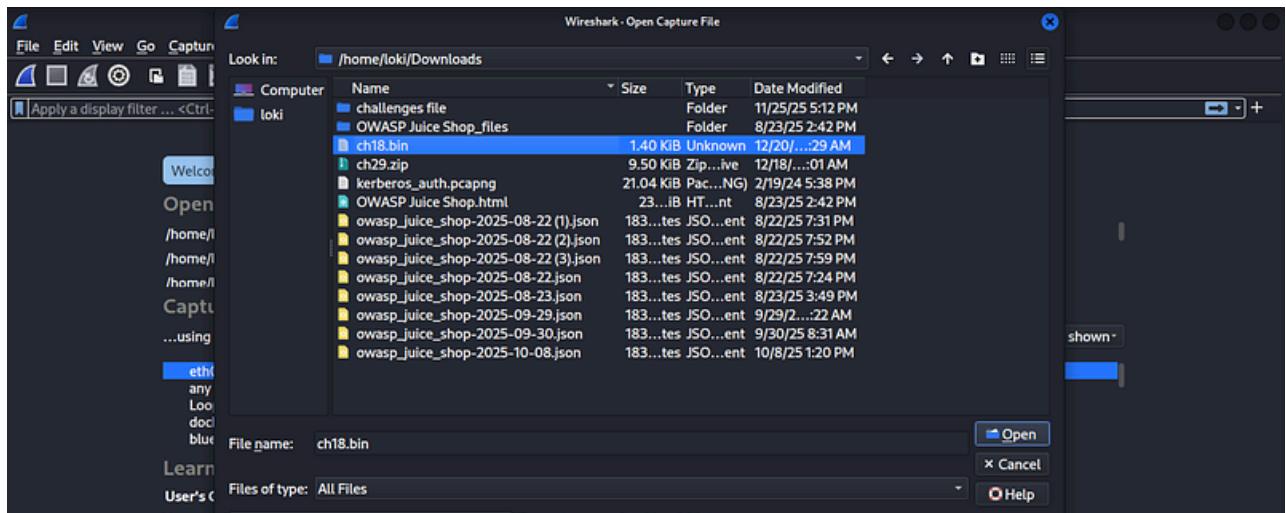
## What is a BT-Snoop File?

A **BT-Snoop 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**.

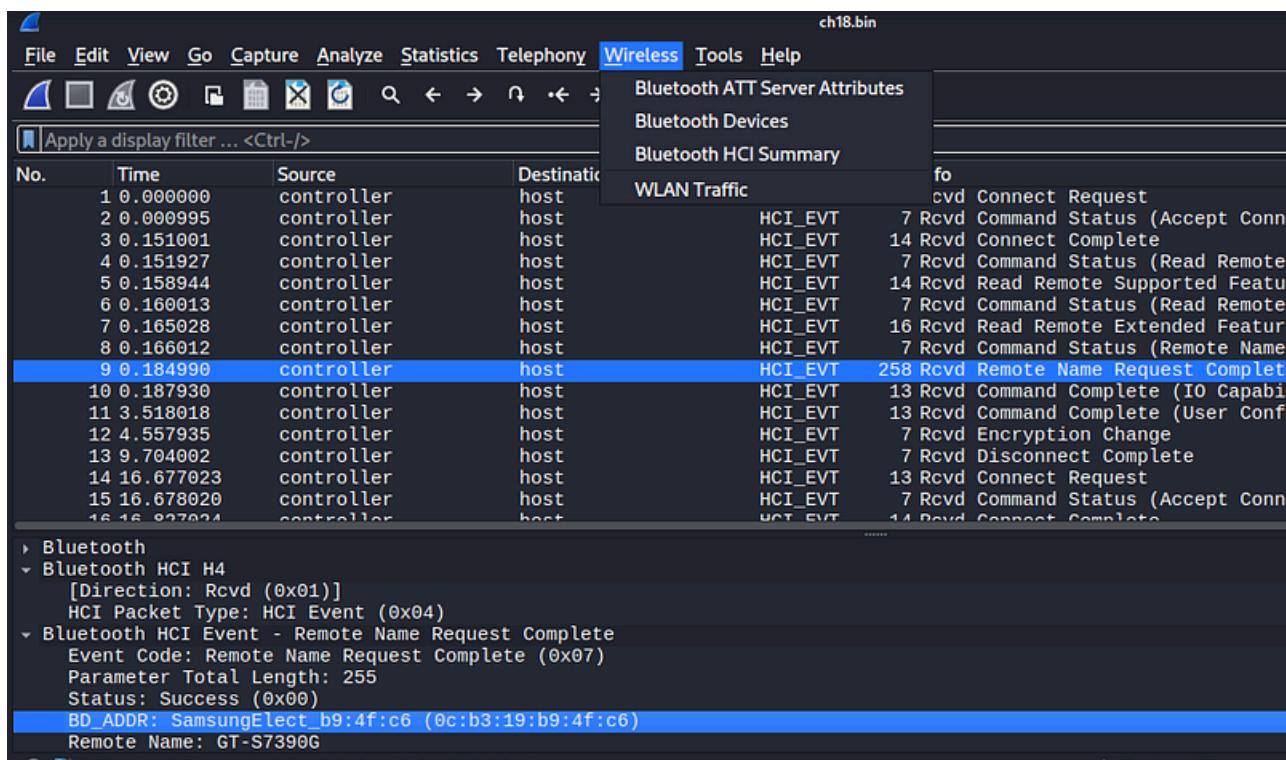
## Traffic Analysis with Wireshark

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



Once loaded:

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



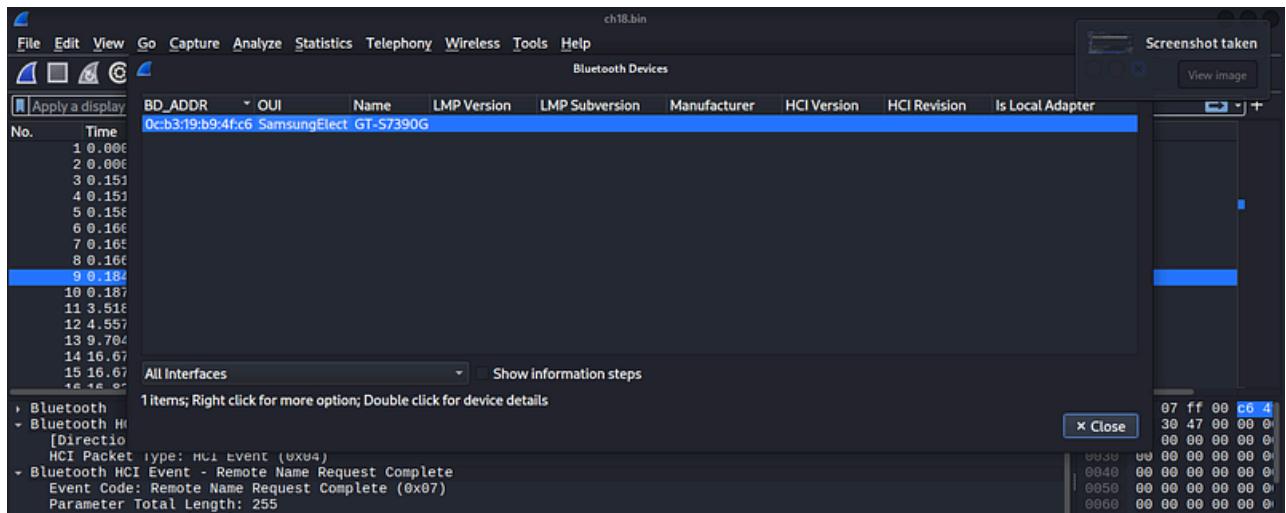
To extract the required information, I navigated to:

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/onlinedevtools/generate-hash/sha1/>

The screenshot shows a web application for generating SHA-1 hashes. At the top, the URL is https://www.bairesdev.com/tools/onlinedevtools/generate-hash/sha1/. The page has a dark header with various navigation links like SE, stuff, labs, social engineering, recon, malware testing, malware dev, OSINT tools, and OSINT resources. A search icon and a notification badge are also present.

**Generate SHA-1 hash from text**

**About SHA-1**  
Once widely used but now considered insecure due to collision attacks.

**Content**  
OC:B3:19:B9:4F:C6GT-S7390G

**Convert to SHA-1**

**Result**  
Resulting SHA-1 hash  
c1d0349c153ed96fe2fadf44e880aef9e69c122b

## Final Flag

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

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

[Canonical link](#)

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