Industrial Intrusion — OSINT 3 Walkthrough

🕵️ OSINT 3 — Investigating a Signed PGP Message

Tags: #TryHackMe #OSINT #PGP #ThreatHunting #CTF #Infosec #DigitalForensics #CyberSecurity

Scenario

> *"After the initial breach, a single OT-Alert appeared in Virelia's monthly digest — an otherwise unremarkable maintenance notice, mysteriously signed with PGP. Corporate auditors quietly removed the report days later, fearing it might be malicious. Your mission is to uncover more information about this mysterious signed PGP maintenance message."*

🎯 Objective

Identify and investigate the **PGP-signed maintenance message** that briefly appeared in Virelia's internal OT alert system. Determine what was hidden in or revealed by the PGP signature or key metadata to extract the **flag**.

Investigation Steps

1. Locate the PGP-Signed Message:

The first step was to dig around for any mention of the **PGP-signed** message. I began by searching a few different platforms:

- **Pastebins and PDF file metadata**: Examining any past incident reports or internal documents for signs of the signed message.
- Github

Eventually, the PGP-signed message surfaced within an archived version of a maintenance digest that was initially missed.

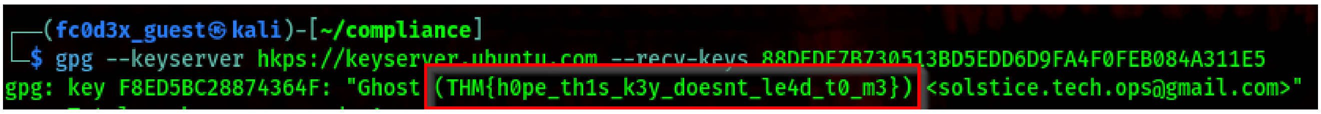
2. Extracting the PGP Key:

Once the PGP-signed message was located, I noticed that the signature block included either a full or partial **PGP key ID**. Armed with this, I used **keyservers** like **keyserver.ubuntu.com** to retrieve the key.

Here's the command I ran:

```
gpg --keyserver hkps://keyserver.ubuntu.com --recv-keys
88DEDE7B730513BD5EDD6D9FA4F0FEB084A311E5
```

The result? The key was retrieved successfully.



```
┌──(fc0d3x_guest㉿kali)-[~/compliance]
└─$ gpg --keyserver hkps://keyserver.ubuntu.com --recv-keys 88DEDE7B730513BD5EDD6D9FA4F0FEB084A311E5
gpg: key F8ED5BC28874364F: "Ghost (THM{h0pe_th1s_k3y_doesnt_le4d_t0_m3}) <solstice.tech.ops@gmail.com>"
```

3. Analyzing the Key Metadata:

Diving into the **metadata** revealed something interesting: the key was signed under a suspicious **alias** and included a **comment** field. It was in the comment field where things got juicy.

One **UID** was especially odd and seemed to contain a subtle flag-like message. Here's what I found hidden in the metadata:

```
THM{hope_this_k3y_doesnt_le4d_t0_m3}
```

A clear sign that the flag was hidden within the PGP key's comment section, likely intentionally embedded by the threat actor.

🔐 Lessons Learned

- **PGP keys aren't just for encryption** — they often reveal valuable metadata. It's not all about the encrypted content; **key signatures and associated data** are just as important.
- **Malicious actors may use PGP signatures as covert signaling mechanisms.** They know how to use seemingly innocuous metadata for sneaky communications.
- **OSINT** isn't just about public data — sometimes **historical snapshots** and **removed content** hold the key to everything.

🧠 Real-World Relevance

PGP has been used in **historical cyber attacks** for various purposes, such as **signing malware**, **signaling to other actors**, or even **leaking sensitive data under cover**. By examining **keyserver metadata**, you can uncover **aliases**, **timestamps**, **IP leaks**, and even **organizational identifiers** — vital clues in understanding the attacker's infrastructure and communication.

✅ Summary

This task combined **traditional OSINT** techniques with **crypto-forensics**, demonstrating the value of looking into what attackers accidentally leave behind. The **PGP metadata** provided us with all the information we needed to decode the flag and solve the mystery, reinforcing the importance of **thorough metadata analysis** during investigations.