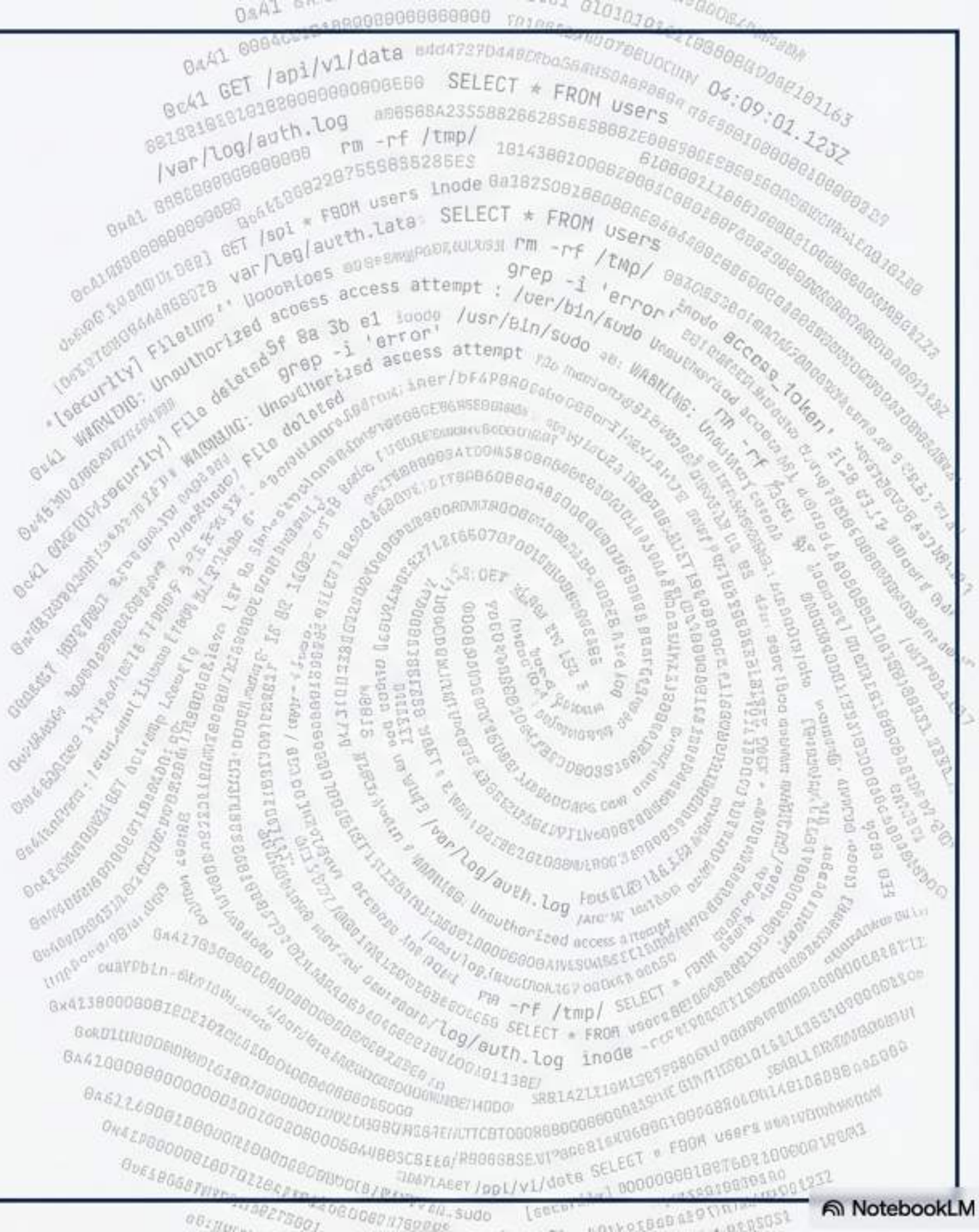
Cancellato. Ma non abbastanza.

Un post-mortem su DFIR, file orfani e l’errore di cancellare un intruso.

```
CASE: testuser | STATUS: exit 0 | #iam #dfir #observability
```



L'Avviso delle 09:23

09:23

@barno

Ho visto una cosa strana stanotte. C'era un utente `testuser` che non ricordo di aver creato. L'ho cancellato, ma magari vale la pena guardare.

L'intenzione era proteggere il sistema.
Il risultato è stato inquinare la scena.

L'Assenza di un Account Non È l'Assenza di una Storia

```
▶ $ getent passwd testuser
[Nessun output]
```

L'utente non esiste più. Ma nel Digital Forensics, il nome è solo un'etichetta. La vera caccia inizia dai numeri rimasti nell'ombra.

La Timeline dell'Intrusione

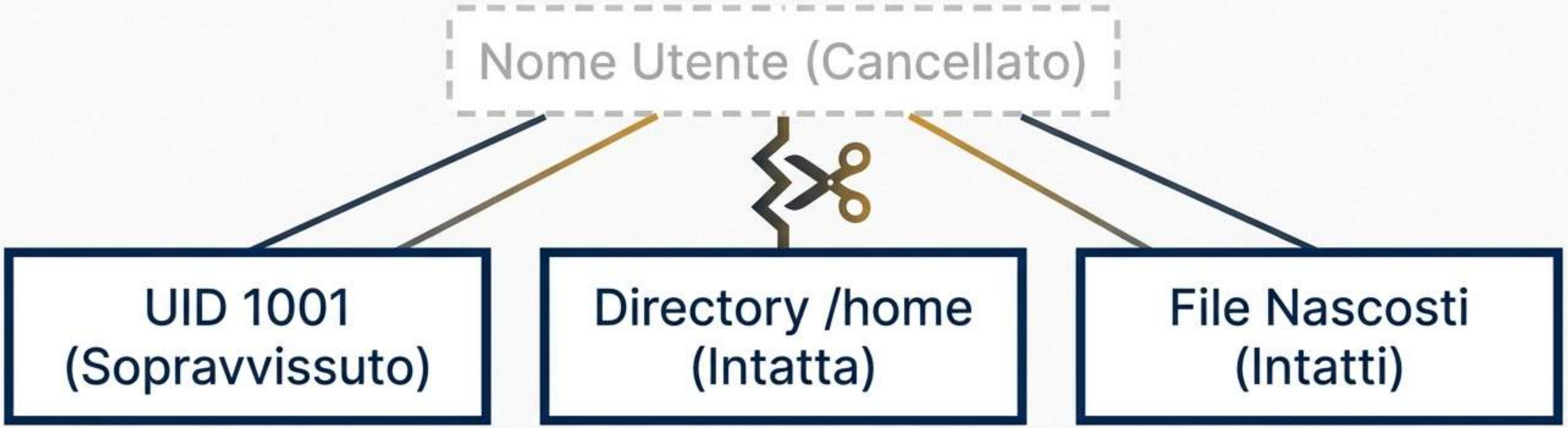
useradd
Creazione account testuser. Inizio attività.

Scrittura .bash_history
Ultime tracce di attività manuale nella shell.

userdel
Cancellazione dell'account da parte dell'amministratore.

Tre ore e mezza di permanenza sul server. Fonte: `/var/log/auth.log`

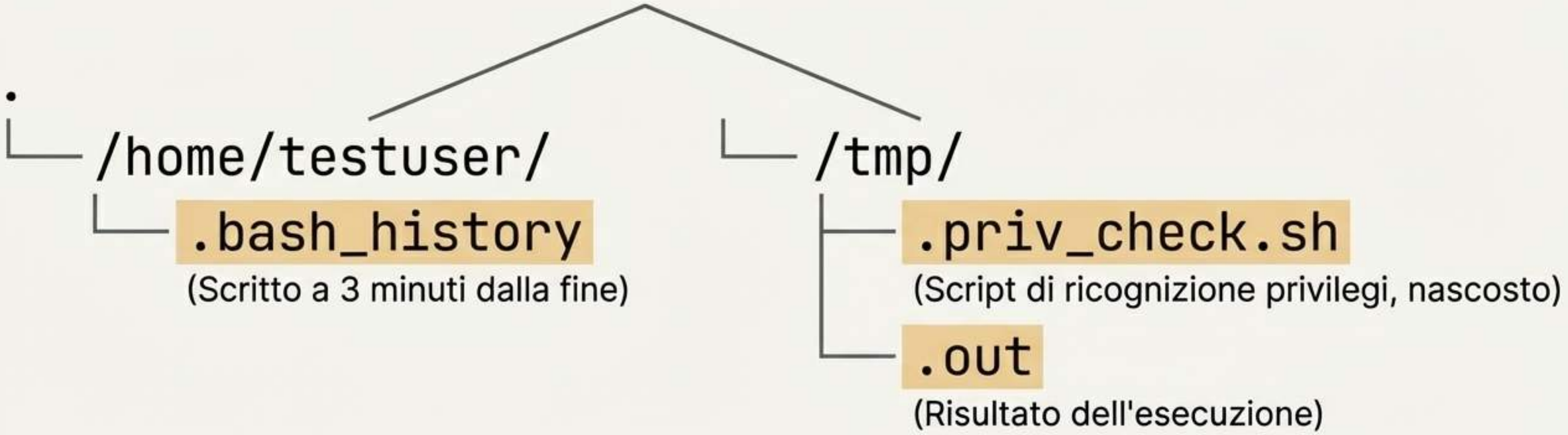
L’Anatomia di un Errore: I File Orfani



Eseguire `userdel` senza il flag `-r` rimuove l’identificatore dal sistema, ma lascia sul disco file associati a un numero (`UID 1001`) senza più un nome. Questi sono gli Orphaned Files.

La Caccia ai Residui

```
$ find / -uid 1001 2>/dev/null
```



Il punto iniziale nei file in **/tmp** indica un chiaro intento di eludere un normale comando ls.

Il Tentativo di Escalation

```
python3 -c "import os; os.setuid(0); os.system('/bin/bash')"
```

Richiama la libreria di sistema operativo.

Tenta di forzare l'assegnazione dell'identificatore Root (UID 0).

Tenta di generare una nuova shell interattiva con i privilegi appena acquisiti.

FALLITO

Sudo era bloccato, ma dimostra l'alta competenza dell'attaccante nell'usare un interprete per bypassare i controlli.

La Kill Chain Rivelata dal .bash_history

Tutto in ordine cronologico. Tutto tracciato. Il file **.out** in **/tmp** conferma che l'attaccante ha cercato, non ha trovato vulnerabilità, e si è fermato.

L'Origine della Minaccia

192.168.64.47

Analizzando le sessioni da wtmp, emerge che il login proviene da un IP IP della rete interna.

Non è un attaccante esterno casuale. È qualcuno già all'interno del perimetro, o una macchina interna gravemente compromessa.

La Fonte della Verità: auth.log vs journald

 **auth.log (L'Indizio)**	 **journalctl (La Prova)**
Formato: Testo in chiaro.	**Formato: Binario.**
Vulnerabilità: Alterabile o svuotabile da Root (es. `> /var/log/auth.log`).	**Sicurezza:** Usa **Forward Secure Sealing (FSS)**. Ogni blocco è firmato crittograficamente. **Modifiche rilevabili.**
Ruolo: Registra la cancellazione (`userdel[3912]`), ma non chi l'ha avviata.	**Ruolo:** Rivela l'identità dell'utente di turno (`barno`) che ha eseguito il comando.

Threat Intelligence Canvas (IoC)

IDENTITY

testuser | UID 1001 (T1136.001)

NETWORK

Origine SSH
192.168.64.47 (T1021.004)

FILES

/tmp/.priv_check.sh | /tmp/.out
(T1087.001)

TACTICS

Tentativo Escalation via Interprete (T1548.003)

Cancellazione Incompleta senza -r (T1070.001)

Checklist Post-Mortem

- [!] Identificare la macchina `192.168.64.47` e verificarne lo stato.
- [!] Cercare accessi dallo stesso IP nelle settimane precedenti.
- [!] Verificare accesso a `/var/backups/passwd.bak` (visto nel `bash_history`).
- [!] Abilitare log forwarding esterno (SIEM alert su `useradd`/`userdel`).

Senza log forwarding, questa analisi è dipesa dalla fortuna e dagli errori dell'attaccante.

La Regola d’Oro

“Cancellare un account sospetto non è incident response. È contaminare la scena.”

La prossima volta che trovi un utente sospetto: isolalo, non cancellarlo.

▶ $ exit 0

Il nome di un utente si recupera in tre secondi.
I log in memoria, la connessione TCP attiva, i file aperti...

...quelli no.