Redacted Corp.

Demo Internal Penetration Test

Date: February 6, 2023

Report Version: 1.0

Contact:
Petar Anastasov
phone: +359 885554477
email: panastasov@secragon.com

Table of Contents

1 Confidentiality

1.1 Confidentiality Statement

This document is the sole and exclusive property of Redacted Corp. and Secragon LLC. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both Redacted Corp. and Secragon LLC. Redacted Corp. may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

1.2 Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period. Time-limited engagements do not allow for a full evaluation of all security controls. Secragon LLC prioritized the assessment to identify the weakest security controls an attacker would exploit. Secragon LLC recommends conducting similar assessments by internal or third-party assessors at least annually and after any significant infrastructure or policy change occurs to ensure the continued success of the controls.

1.3 Contact Information

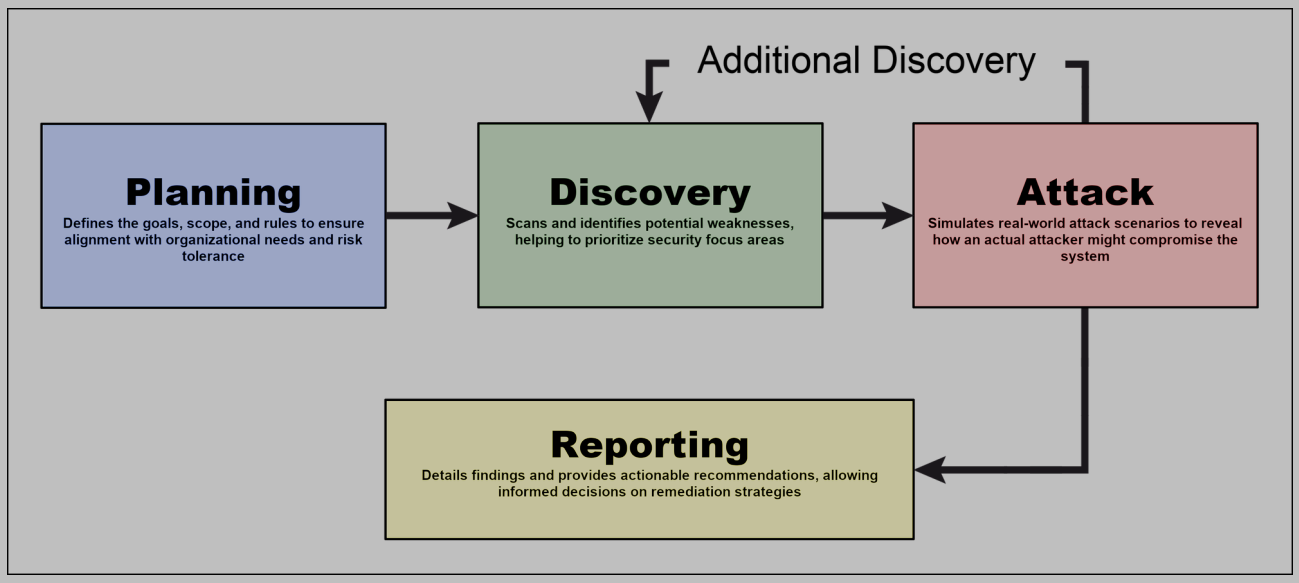
Name	Title	Contact
Redacted Corp.		
Redacted Name	Chief Information Security Officer	redacted@redacted.com
Secrgon LLC		
Petar Anastasov	Lead Penetration Tester	panastasov@secragon.com

2 Assessment Overview

2.1 Overview

From 2023-01-17 to 2023-01-23, Redacted Corp. engaged Secragon LLC to evaluate the security posture of its infrastructure compared to current industry best practices that included an internal network penetration test. All testing performed is based on the NIST SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks. Phases of penetration testing activities include the following:

- *Planning* – Gathering customer goals and obtaining rules of engagement.
- *Discovery* – Performing scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- *Attack* – Confirming potential vulnerabilities through exploitation and performing additional discovery upon new access.
- *Reporting* – Documenting all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



2.2 Components

Internal Penetration Test
An internal penetration test emulates the role of an attacker from inside the network. An engineer will scan the network to identify potential host vulnerabilities that could lead to unauthorized access and control over internal systems. Subsequent steps will seek to gain access to hosts through lateral movement, compromise domain user and admin accounts, and exfiltrate sensitive data. The aim is to evaluate the security of the organization's internal network and understand how an internal attacker could leverage vulnerabilities to obtain unauthorized access, perform malicious actions, or acquire confidential information.

3 Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3.2 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are typically non-exploitable, but addressing them would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

CVSS Explanation

The Common Vulnerability Scoring System (CVSS) is a widely recognized standard for assessing the severity of computer system security vulnerabilities. It provides a way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity. The CVSS score can then be used to determine urgency and priority of response. CVSS is beneficial for:

- **Comparing Vulnerabilities:** It allows vulnerabilities to be compared across different systems and technologies.
- **Prioritizing Response:** By aligning the severity score with the organization’s risk appetite, appropriate resources can be allocated to respond to the vulnerability.
- **Consistent Scoring:** It provides a standardized scoring system that is consistent across different organizations and industries.
- **Aligning with Industry Best Practices:** CVSS is aligned with industry best practices and is used by many vulnerability databases and security vendors.

In this report, CVSS V3.1 has been utilized to rate the vulnerabilities, ensuring an objective, repeatable, and widely understood method of assessing the severity of findings.

4 Scope

Host/URL/IP Address	Notes
*	No details provided

Scope Exclusions

Per client request, Secragon LLC did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing/Social Engineering

All other attacks not specified above were permitted by Redacted Corp.

Client-Provided Information

Black-Box testing.

No information about the company network infrastructure was provided. Client granted access to a virtual machine on a separate network with a route to the company network.

5 Executive Summary

5.1 Overview

Secragon LLC was engaged by Redacted Corp. to conduct an internal black-box penetration test. Without any prior knowledge of the company's network infrastructure or direct access to the main network, our team started the engagement from a virtual machine on a separate network, with only a route to the company's primary infrastructure.

The objective of this assessment was to simulate a real-world attack scenario where an attacker has minimal initial information but aims to discover and exploit vulnerabilities to gain deeper access into Redacted Corp.'s systems.

Our assessment identified several security vulnerabilities. By exploiting a combination of these, our team successfully acquired valid domain credentials using different techniques, navigated across the company's networks, and ultimately secured full control over the domain. This emphasizes a clear risk: with the right tools and expertise, a skilled adversary could potentially replicate these actions.

Our findings during the assessment highlighted concerns related to weak password practices, misconfigurations, and exposed services. These vulnerabilities enabled unauthorized access to key areas within the company's network.

In the detailed report that follows, Secragon LLC provides comprehensive recommendations tailored to address and rectify these vulnerabilities. Implementing these solutions is crucial for enhancing Redacted Corp.'s security measures, thereby reducing risks and improving the organization's overall security position.

We appreciate the trust and openness shown by Redacted Corp. throughout this assessment. Their active cooperation ensured a thorough and effective evaluation. We remain committed to assisting Redacted Corp. further, ensuring that they maintain a strong defense against potential cyber threats. Given the serious nature of our findings, we strongly recommend immediate action to protect Redacted Corp.'s internal systems from potential security breaches.

5.2 Identified Vulnerabilities

#	CVSS	Description	Page
C1	10.0	Password Reuse for Root Account	12
C2	9.6	PetitPotam (CVE-2021-36942)	13
C3	9.6	DFSCoerce Vulnerability	16
C4	9.0	Remote Code Execution (RCE) in Checkmk v2.1.0p10	17
H1	8.7	LLMNR/NBT-NS Response Spoofing	19
H2	8.6	Password Reuse for Active Directory	21
H3	8.5	Privilege Escalation via GPO Misconfiguration	24
H4	8.2	Symfony Development Mode Exposure	27
H5	7.5	Crackable Password for Cmkadmin Account	29
M1	6.8	Weak Active Directory Account Passwords	30
M2	5.4	Weak Admin Password for Laravel Application	31
M3	5.3	Unauthenticated Web Server Revealing Product Deployment Information	32
M4	4.8	Outdated iDRAC 9 Server	34

Vulnerability Overview

In the course of this penetration test **4 Critical** , **5 High** and **4 Medium** vulnerabilities were identified:

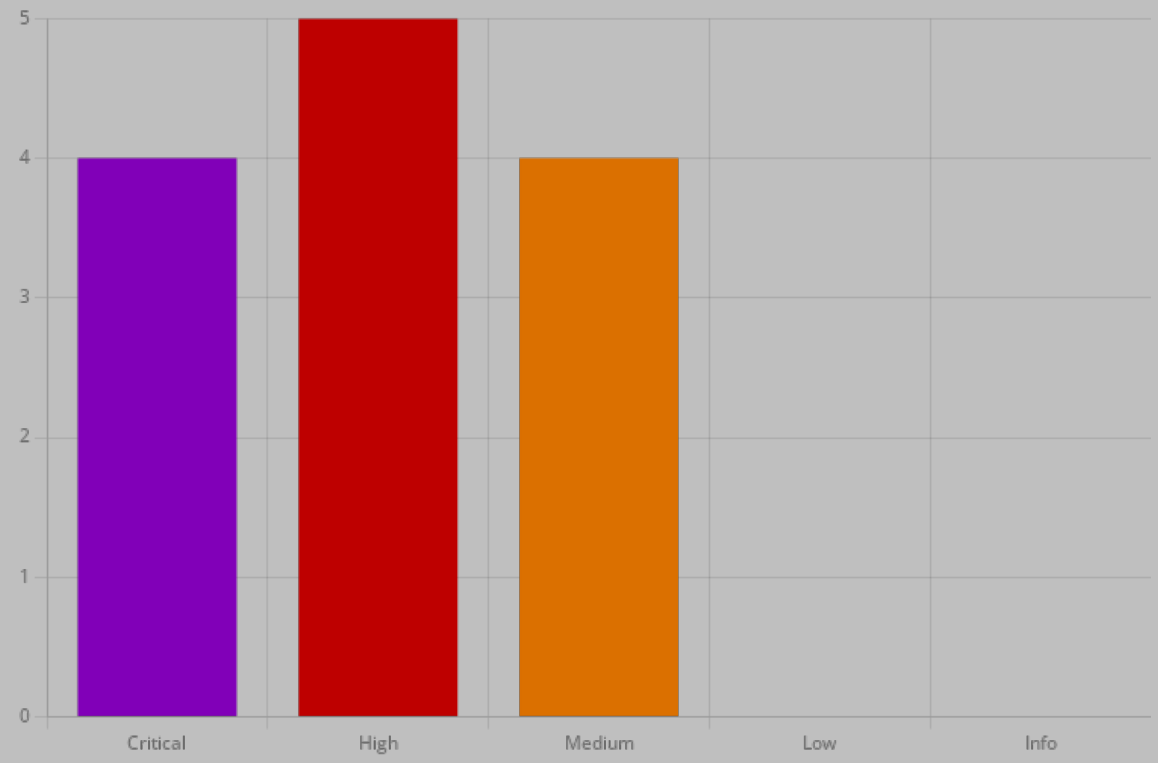


Figure 1 - Distribution of identified vulnerabilities

6 Internal Network Compromise Walkthrough

During this internal penetration testing engagement, Secragon LLC was presented with the task of simulating advanced threat actors targeting Redacted Corp.'s Active Directory infrastructure. Commencing from an isolated position without direct domain access, the objective was to authenticate, escalate privileges, and ultimately demonstrate domain-wide compromise potential. This report provides a structured sequence of actions that led to the said domain compromise. Not all vulnerabilities and misconfigurations identified during the test are detailed in this sequence. They are cataloged separately in the "Technical Findings Details" section, ranked by severity. The objective of this account is to illustrate to Redacted Corp. the real-world implications of combined vulnerabilities, emphasizing how they cumulatively pose a significant risk to the organization. The insights provided herein aim to guide prioritized remediation strategies. Even though other vulnerabilities in this report could lead to comparable results, the described path signifies the most straightforward approach taken by our testers to achieve complete domain compromise.

Initial Network Position and Objective:

Commencing the assessment from a virtual machine on a separate network, the initial challenges we faced were:

- Gaining a vantage position within a segment of the network housing Active Directory machines to execute potential relay attacks.
- Securing valid credentials to facilitate authentication against resources within Redacted Corp.'s domain.

Credential Acquisition Scenario 1:

1. **Remote Code Execution (RCE) in Checkmk v2.1.0p10**: Our team identified a vulnerable Checkmk instance. Through a purpose-crafted exploit, we secured low-privileged access on the associated machine.
2. **Crackable Password for Cmkadmin Account**: The 'cmkadmin' password was successfully cracked, granting us administrative privileges over the Checkmk instance.
3. **Password Reuse for Root Account**: It was discerned that the previously cracked cmkadmin password was also set for the root account, enabling further escalated access.
4. **LLMNR/NBT-NS Response Spoofing**: With our privileged stance, we executed an LLMNR/NBT-NS spoofing attack and intercepted NTLMv2 hashes.
5. **Weak Active Directory Account Passwords**: Several intercepted NTLMv2 hashes were decrypted, yielding additional valid credentials.

Credential Acquisition Scenario 2:

1. **Symfony Development Mode Exposure**: An instance of Symfony in its development mode was pinpointed, which allowed for the extraction of POST data.
2. **Password Reuse for Active Directory**: From the data sourced from Symfony, we derived multiple sets of credentials. On testing, we ascertained that four of these were valid for authentication within Redacted Corp.'s domain.

Achieving Elevated Domain Access:

- **PetitPotam (CVE-2021-36942)**: The domain controllers were identified as vulnerable to the PetitPotam exploit. Using this vulnerability, we initiated a relay against the Certificate Authority (CA), acquiring a certificate that mirrored the identity of the dc01 domain controller. This enabled us to gain a Ticket Granting Ticket (TGT) equivalent to dc01's access rights, subsequently allowing elevated control over the entire Active Directory.

- **DFSCoerce Vulnerability**: This vulnerability mimics the behavior of PetitPotam. Given the successful exploitation of PetitPotam, this vulnerability was not pursued.
- **Privilege Escalation via GPO Misconfiguration**: Potential GPO misconfigurations suitable for privilege escalation were identified. However, given the production environment, we opted not to exploit this to ensure Redacted Corp.'s operational stability.

The outlined steps unveil the potential of chaining specific vulnerabilities to gain escalated access and eventually compromise the entirety of the Redacted Corp.'s domain. It underscores the necessity for rigorous remediation, continuous monitoring, and the fortification of security controls. Such proactive measures are paramount in safeguarding the organization's critical assets and ensuring resilience against sophisticated adversaries.

7 Technical Findings

C1: Password Reuse for Root Account	
CVSS 3.1 Score	**10.0 (Critical)**
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N
Description	The password for the 'cmkadmin' account, which was previously identified as being vulnerable to cracking, was also found to be reused for the root account on the system.
Impact	Reusing the password for the root account amplifies the potential damage of a breach. Once an attacker gains root access, they have the highest level of privileges on the system. This allows for complete control over the machine, enabling actions such as data extraction, system manipulation, further lateral movement within the network, and potential persistence mechanisms.
Target	x.x.x.100
Remediation	Implement a password management policy that enforces unique passwords for different accounts, especially high-privilege accounts.
References	https://cwe.mitre.org/data/definitions/521.html

Finding Evidence

After successfully **cracking the 'cmkadmin' password**, our team tested its applicability on system accounts. The cracked password was found to be valid for the root account, granting full system-level access.

C2: PetitPotam (CVE-2021-36942)	
CVSS 3.1 Score	**9.6 (Critical)**
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N
Description	Both domain controllers (DCs) were found vulnerable to PetitPotam, an exploit targeting the Microsoft Encrypting File System Remote Protocol (MS-EFSRPC) API. This exploit leverages the protocol to force Windows servers, including domain controllers, to initiate the NTLM authentication process, forcing the server to share its NTLM credentials by exploiting legitimate functionalities of the MS-EFSRPC.
Impact	When subjected to a successful PetitPotam attack, an adversary could relay the extracted NTLM credentials to Active Directory Certificate Services (ADCS) to generate an authentication certificate. This action potentially allows the attacker to take complete control over a domain, especially when the domain controller is compromised.
Target	dc01.******.local dc03.******.local
Remediation	• Disabling NTLM Authentication on Windows domain controllers. • Disabling NTLM on any AD CS Servers in your domain using the group policy Network security: Restrict NTLM: Incoming NTLM traffic. • Disabling NTLM for Internet Information Services (IIS) on AD CS Servers in your domain running the "Certificate Authority Web Enrollment" or "Certificate Enrollment Web Service" services.
References	https://news.sophos.com/en-us/2021/08/25/how-petitpotam-hijacks-the-windows-api-and-what-you-can-do-about-it/ https://support.microsoft.com/en-gb/topic/kb5005413-mitigating-ntlm-relay-attacks-on-active-directory-certificate-services-ad-cs-3612b773-4043-4aa9-b23d-b87910cd3429

Finding Evidence

Building upon the valid AD credentials obtained from the **Weak Active Directory Account Passwords vulnerability** or the **Password Reuse for Active Directory vulnerability**, the following steps were executed to further demonstrate the compound risks:

1. Using the ***crackmapexec*** tool, the domain controllers were verified to be susceptible to the ***PetitPotam*** exploit. This analysis confirmed that they were vulnerable to the relay attack.

2. An NTLM relay session was set up using ***ntlmrelayx.py***, targeting the Certificate Authority (CA). Through the "DomainController" template, the PetitPotam exploit was launched to coerce the server into the NTLM authentication process against the relay.

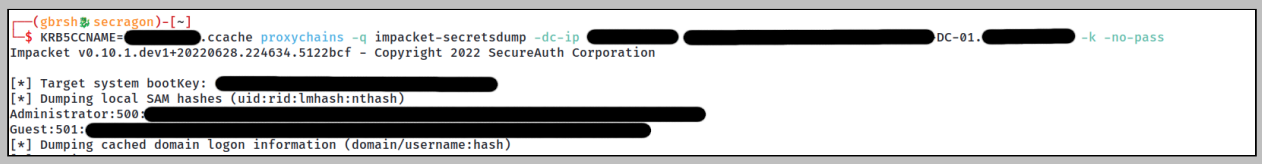
3. After the successful relay, a certificate linked to the dc01 domain controller was obtained. With this certificate, the team essentially had a copy of the identity of the dc01 domain controller. Using the ***gettgtpkinit.py*** tool, the team took advantage of PKINIT to get a Ticket Granting Ticket (TGT) for the domain. This means that the same access rights as the dc01 domain controller within the domain were acquired.

```
┌──(gbrsh㉿secragon)-[~]
└─$ cat cert.b64 | base64 -d > cert.pfx

┌──(gbrsh㉿secragon)-[~]
└─$ git clone https://github.com/dirkjanm/PKINITtools
Cloning into 'PKINITtools'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 32 (delta 13), reused 25 (delta 7), pack-reused 0
Receiving objects: 100% (32/32), 24.37 KiB | 6.09 MiB/s, done.
Resolving deltas: 100% (13/13), done.

┌──(gbrsh㉿secragon)-[~]
└─$ proxychains -q python3 PKINITtools/gettgtpkinit.py [redacted]DC-01\$ -cert-pfx cert.pfx out.ccache
2023-01-22 12:48:39,832 minikerberos INFO     Loading certificate and key from file
INFO:minikerberos:Loading certificate and key from file
2023-01-22 12:48:40,201 minikerberos INFO     Requesting TGT
INFO:minikerberos:Requesting TGT
2023-01-22 12:48:40,271 minikerberos INFO     AS-REP encryption key (you might need this later):
INFO:minikerberos:AS-REP encryption key (you might need this later):
2023-01-22 12:48:40,271 minikerberos INFO     28e84df2ff5d9e41d54709ce24e5d70c04bd6f6bfa8e8e8eb2b11fe62adafa56
INFO:minikerberos:28e84df2ff5d9e41d54709ce24e5d70c04bd6f6bfa8e8e8eb2b11fe62adafa56
2023-01-22 12:48:40,274 minikerberos INFO     Saved TGT to file
INFO:minikerberos:Saved TGT to file
```

4. Using the ***gets4uticket.py*** tool and the TGT, a silver ticket was crafted for a domain admin account. This ticket granted elevated privileges within the domain.

```
┌──(gbrsh㉿secragon)-[~]
└─$ KRB5CCNAME=out.ccache proxychains -q python3 PKINITtools/gets4uticket.py kerberos+ccache://[redacted]DC-01\$:out.ccache@[redacted]DC-01.[redacted] cifs/[redacted]DC-01 [redacted].ccache -v
2023-01-22 13:00:41,204 minikerberos INFO     Trying to get SPN with [redacted] for cifs/[redacted]DC-01.[redacted]
INFO:minikerberos:Trying to get SPN with [redacted] for cifs/[redacted]
2023-01-22 13:00:41,228 minikerberos INFO     Success!
INFO:minikerberos:Success!
2023-01-22 13:00:41,229 minikerberos INFO     Done!
INFO:minikerberos:Done!
```

5. Utilizing the privileges granted by the silver ticket, the team leveraged the ***secretsdump.py*** tool to extract hashes, password histories, Kerberos tickets, and other critical domain information, effectively taking control over the entire Active Directory.



C3: DFSCoerce Vulnerability	
CVSS 3.1 Score	**9.6 (Critical)**
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N
Description	Both domain controllers (DCs) were identified as being susceptible to the DFSCoerce vulnerability. This vulnerability could potentially allow attackers to coerce Windows servers, particularly domain controllers, into sharing NTLM credentials by leveraging the Distributed File System Replication (DFSR) service.
Impact	If successfully exploited, the DFSCoerce vulnerability allows an attacker to acquire NTLM authentication credentials from domain controllers or other vulnerable Windows servers. These credentials can then be relayed or misused to manipulate domain resources, potentially leading to escalated privileges within the domain. The attacker could potentially gain control over critical services and data, jeopardizing the integrity and confidentiality of the entire Active Directory environment.
Target	dc01.******.local dc03.******.local
Remediation	• Disabling NTLM Authentication on Windows domain controllers. • Disabling NTLM on any AD CS Servers in your domain using the group policy Network security: Restrict NTLM: Incoming NTLM traffic. • Disabling NTLM for Internet Information Services (IIS) on AD CS Servers in your domain running the "Certificate Authority Web Enrollment" or "Certificate Enrollment Web Service" services.
References	https://www.malwarebytes.com/blog/news/2022/06/dfscoerce-a-new-ntlm-relay-attack-can-take-control-over-a-windows-domain https://support.microsoft.com/en-gb/topic/kb5005413-mitigating-ntlm-relay-attacks-on-active-directory-certificate-services-ad-cs-3612b773-4043-4aa9-b23d-b87910cd3429

Finding Evidence

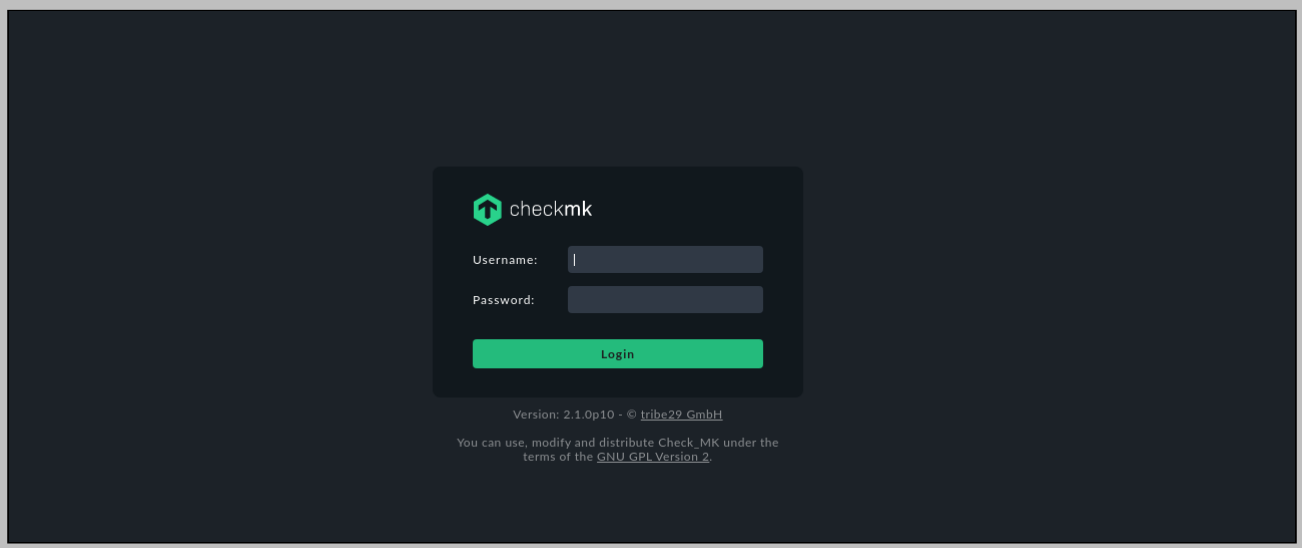
With the assistance of the ***crackmapexec*** tool, the domain controllers were verified to be susceptible to the DFSCoerce vulnerability, underlining the interconnected risk and importance of a comprehensive security posture.

(Note: Further exploitation was not pursued due to the similarities in the **exploitation pathway with PetitPotam**, demonstrating the compound risk in environments with multiple vulnerabilities.)

C4: Remote Code Execution (RCE) in Checkmk v2.1.0p10	
CVSS 3.1 Score	**9.0 (Critical)**
CVSS Vector	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H
Description	Checkmk v2.1.0p10 running on port 8000 was detected. This specific version is known to contain multiple vulnerabilities that, when chained together, could lead to remote code execution (RCE), allowing attackers to execute arbitrary code on the compromised system.
Impact	The successful exploitation of this vulnerability could allow an attacker to execute arbitrary commands on the server, potentially leading to unauthorized access to sensitive data, manipulation of system configurations, escalation of privileges, and other malicious activities. Additionally, it granted access to a system within the network where domain controllers are running, posing a significant risk for further attacks against the domain.
Target	x.x.x.100
Remediation	Upgrade Checkmk to the latest version to address the identified vulnerabilities.
References	https://www.sonarsource.com/blog/checkmk-rce-chain-1/ https://docs.checkmk.com/latest/en/update.html

Finding Evidence

During network scanning of IP address x.x.x.x, our team identified a running instance of Checkmk version 2.1.0p10.



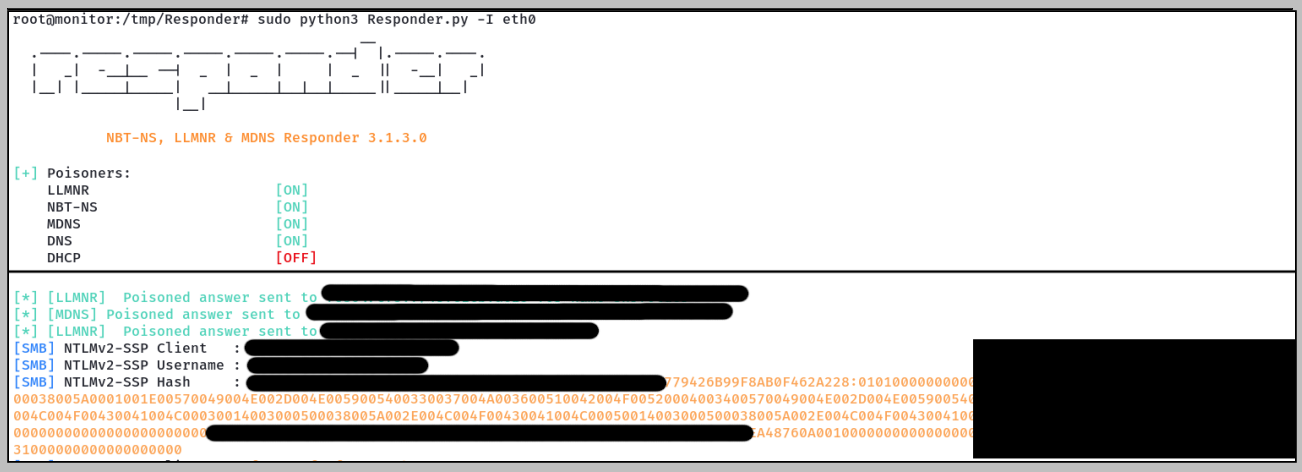
Subsequent analysis confirmed that the detected version was vulnerable to remote code execution (RCE), achievable by chaining multiple vulnerabilities together. The exploitation of these vulnerabilities was performed using an ***exploit*** developed by our researchers, which enabled successful unauthorized access to the underlying system.

Following successful exploitation, we obtained the password hash for the 'cmkadmin' account, as evidenced by the last screenshot.

H1: LLMNR/NBT-NS Response Spoofing	
CVSS 3.1 Score	**8.7 (High)**
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:N
Description	The network was observed to allow Link-Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS) broadcasts. By exploiting this configuration, LLMNR/NBT-NS traffic was intercepted, leading to poisoned name resolutions. Following the poisoned name resolutions, authentication hashes corresponding to multiple users were directed towards a simulated listening endpoint, where they were subsequently captured.
Impact	The captured NTLMv2 hashes pose a significant security risk. If cracked offline, these hashes could disclose plaintext credentials. Additionally, these hashes can be leveraged for NTLM relay attacks, where an attacker might relay a user's authentication session to another server or service, gaining unauthorized access. Unauthorized individuals using these methods could lead to further network exploitation, unauthorized access, potential data breaches, or deeper system compromise.
Target	******.local
Remediation	• Disable LLMNR and NBT-NS on all network devices if they are not explicitly needed for business functions. • Implement network segmentation to limit the broadcast domain of LLMNR and NBT-NS traffic. • Ensure strong, unique passwords are used across the network to reduce the risk of successful password cracking if hashes are obtained. • Regularly monitor network traffic for signs of malicious activity, such as unexpected LLMNR/NBT-NS responses.
References	https://attack.mitre.org/techniques/T1557/001/

Finding Evidence

Upon **gaining access** to the network containing Active Directory systems, our team employed the ***Responder*** tool to assess the network's vulnerability to LLMNR/NBT-NS spoofing. During this assessment, several authentication hashes were captured. The capture of an NTLMv2 hash serves as a notable example of the potential exposure.


root@monitor:/tmp/Responder# sudo python3 Responder.py -I eth0
NBT-NS, LLMNR & MDNS Responder 3.1.3.0
[+] Poisoners:
LLMNR [ON]
NBT-NS [ON]
MDNS [ON]
DNS [ON]
DHCP [OFF]
[*] [LLMNR] Poisoned answer sent to
[*] [MDNS] Poisoned answer sent to
[*] [LLMNR] Poisoned answer sent to
[SMB] NTLMv2-SSP Client :
[SMB] NTLMv2-SSP Username :
[SMB] NTLMv2-SSP Hash :

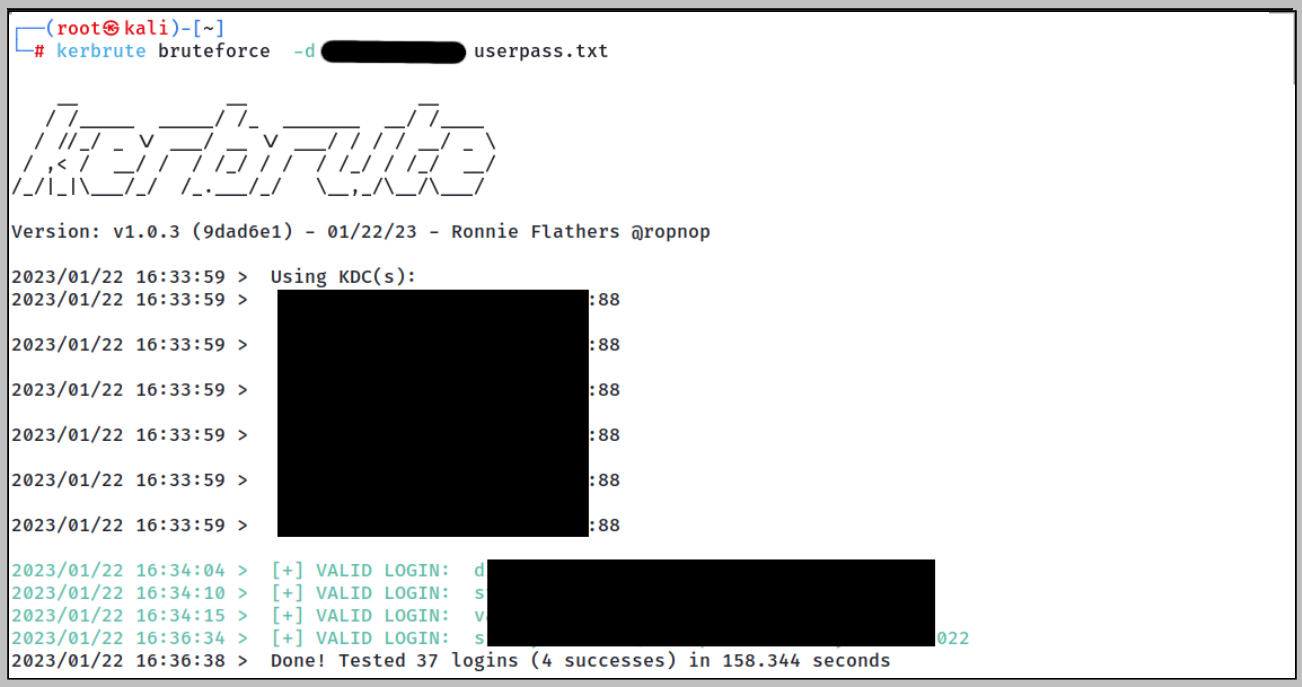
H2: Password Reuse for Active Directory	
CVSS 3.1 Score	**8.6 (High)**
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N
Description	Some of the credentials discovered within the misconfigured Symfony application were also in use within the company's Active Directory.
Impact	The reuse of these credentials between the web application and the Active Directory has potentially compromised the security of resources and services within the network tied to the Active Directory. An attacker who accessed the passwords from the Symfony application could leverage some of them for unauthorized network access, lateral movement, privilege escalation, and other malicious activities within the internal network.
Target	******.local
Remediation	• Reset all passwords that were reused between the Symfony application and the Active Directory. • Adopt a stringent password policy that mandates unique passwords for different platforms and systems. • Regularly audit accounts to eliminate instances of password reuse. • Train staff on the importance of unique password usage across different services, and recommend the adoption of password managers to help maintain strong, distinct credentials.
References	-

Finding Evidence

During our assessment, credentials **obtained from the Symfony application** were tested against the company's Active Directory. Utilizing ***Kerbrute***, the team enumerated the list of 45 user accounts acquired from the Symfony application against the company's Active Directory. This process confirmed 37 out of the 45 users as valid AD accounts.

```
┌──(root㉿kali)-[~]
└─# kerbrute userenum  -d [redacted] users.txt

   __               __               __
  / /_____  _____/ /_  _______  __/ /____
 / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
/ ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.3 (9dad6e1) - 01/22/23 - Ronnie Flathers @ropnop

2023/01/22 16:33:00 >  Using KDC(s):
2023/01/22 16:33:00 >   [redacted]:88

2023/01/22 16:33:05 >  [+] VALID USERNAME:       a[redacted]
2023/01/22 16:33:05 >  [+] VALID USERNAME:       a
2023/01/22 16:33:05 >  [+] VALID USERNAME:       b
2023/01/22 16:33:05 >  [+] VALID USERNAME:       a
2023/01/22 16:33:05 >  [+] VALID USERNAME:       e
2023/01/22 16:33:05 >  [+] VALID USERNAME:       d
2023/01/22 16:33:05 >  [+] VALID USERNAME:       a
2023/01/22 16:33:05 >  [+] VALID USERNAME:       j
2023/01/22 16:33:05 >  [+] VALID USERNAME:       h
2023/01/22 16:33:05 >  [+] VALID USERNAME:       i
2023/01/22 16:33:05 >  [+] VALID USERNAME:       k
2023/01/22 16:33:05 >  [+] VALID USERNAME:       k
2023/01/22 16:33:05 >  [+] VALID USERNAME:       d
2023/01/22 16:33:05 >  [+] VALID USERNAME:       k
2023/01/22 16:33:05 >  [+] VALID USERNAME:       k
2023/01/22 16:33:05 >  [+] VALID USERNAME:       k
2023/01/22 16:33:05 >  [+] VALID USERNAME:       m
2023/01/22 16:33:05 >  [+] VALID USERNAME:       k
2023/01/22 16:33:05 >  [+] VALID USERNAME:       m
2023/01/22 16:33:05 >  [+] VALID USERNAME:       m
2023/01/22 16:33:05 >  [+] VALID USERNAME:       m
2023/01/22 16:33:05 >  [+] VALID USERNAME:       n
2023/01/22 16:33:05 >  [+] VALID USERNAME:       n
2023/01/22 16:33:05 >  [+] VALID USERNAME:       n
2023/01/22 16:33:05 >  [+] VALID USERNAME:       p
2023/01/22 16:33:05 >  [+] VALID USERNAME:       s
2023/01/22 16:33:05 >  [+] VALID USERNAME:       s
2023/01/22 16:33:05 >  [+] VALID USERNAME:       p
2023/01/22 16:33:05 >  [+] VALID USERNAME:       n
2023/01/22 16:33:05 >  [+] VALID USERNAME:       m
2023/01/22 16:33:05 >  [+] VALID USERNAME:       s
2023/01/22 16:33:05 >  [+] VALID USERNAME:       s
2023/01/22 16:33:05 >  [+] VALID USERNAME:       v
2023/01/22 16:33:05 >  [+] VALID USERNAME:       v
2023/01/22 16:33:05 >  [+] VALID USERNAME:       v
2023/01/22 16:33:05 >  [+] VALID USERNAME:       t
2023/01/22 16:33:05 >  [+] VALID USERNAME:       y
2023/01/22 16:33:05 >  Done! Tested 45 usernames (37 valid) in 5.012 seconds
```

The subsequent phase involved testing the user:pass combinations of the 37 validated accounts, revealing that 4 of them had identical credentials in both Symfony and the AD.



H3: Privilege Escalation via GPO Misconfiguration	
CVSS 3.1 Score	**8.5 (High)**
CVSS Vector	CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:C/C:H/I:H/A:H
Description	Analysis of the Active Directory relationships and permissions revealed a concerning delegation chain within the environment. Starting with any authenticated user within the "Domain Users@redacted" group, there exists a series of permissions and relationships that could potentially lead to the unauthorized acquisition of elevated privileges. Specifically, this chain indicates that any authenticated user has the capability, through a series of steps, to potentially gain permissions equivalent to those of an Active Directory administrator.
Impact	Exploitation of this misconfiguration allows any authenticated user to potentially manipulate the "MAP COMPANY DATA" Group Policy Object (GPO). This could influence the behavior of users and even certain administrator accounts. The chain further suggests the possibility of abusing rights to achieve DCSync capabilities on the domain controller "DC03", effectively granting an attacker the ability to pull password hashes and other critical data from the AD database. In short, this vulnerability can lead to a complete compromise of the domain's security and integrity.
Target	******.local
Remediation	Restrict GPO Permissions: Immediately review the permissions on the "MAP COMPANY DATA" GPO. Remove the "GenericWrite" permission for "Authenticated Users" and any other broad groups. Ensure that only necessary accounts have write or modify access. Review Nested Group Memberships: Periodically inspect group and nested group memberships, particularly those with elevated privileges. Ensure that only intended users or groups are members. Principle of Least Privilege: Implement and enforce the principle of least privilege. Assign permissions only as needed, avoiding broad permissions that might encompass unintended users or groups.
References	https://bloodhound.readthedocs.io/en/latest/data-analysis/edges.html

Finding Evidence

During the assessment, domain enumeration was executed utilizing the renowned ***BloodHound*** tool to gain insights into potential privilege escalation paths within the domain's trust relationships and permissions.

From the collected data, BloodHound identified a significant attack vector that begins with standard domain user permissions and eventually leads to a domain compromise. The path discovered was as follows:

Domain Users@domain → Authenticated Users@domain (due to membership)

↓

Granted GenericWrite permissions over GPO MAP COMPANY DATA

↓

GPO MAP COMPANY DATA has a GPLink relationship with Users@domain

↓

Which contains redacted1@domain

↓

redacted1@domain is a member of AD Administrators@domain

↓

Then, through another relationship, to redacted2@domain which has AllExtendedRights over DC03

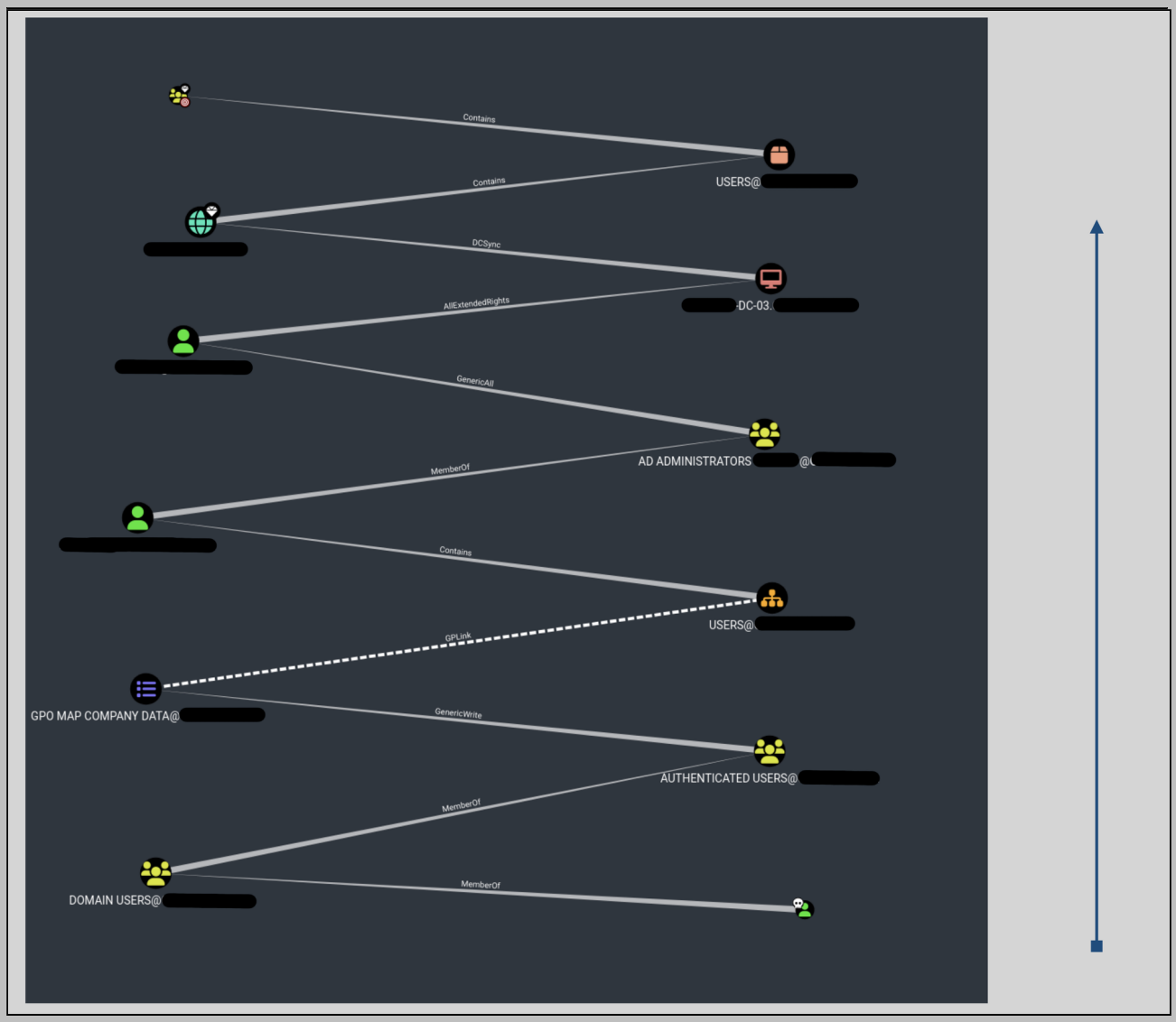
↓

This ultimately allows for a DCSync operation against the domain.

This vector signifies a potential pathway for malicious actors to escalate privileges from a basic domain user to acquiring rights equivalent to a domain administrator, all by exploiting inherent permissions and trust relationships.

However, it is essential to note that this attack pathway was not exploited during the assessment. Given the nature of the identified misconfiguration, actual exploitation in the production environment would risk modifying existing settings or relationships, which could have adverse effects on domain operations. Therefore, in adherence to safe testing principles and to ensure the integrity of the production environment, the attack was not conducted.

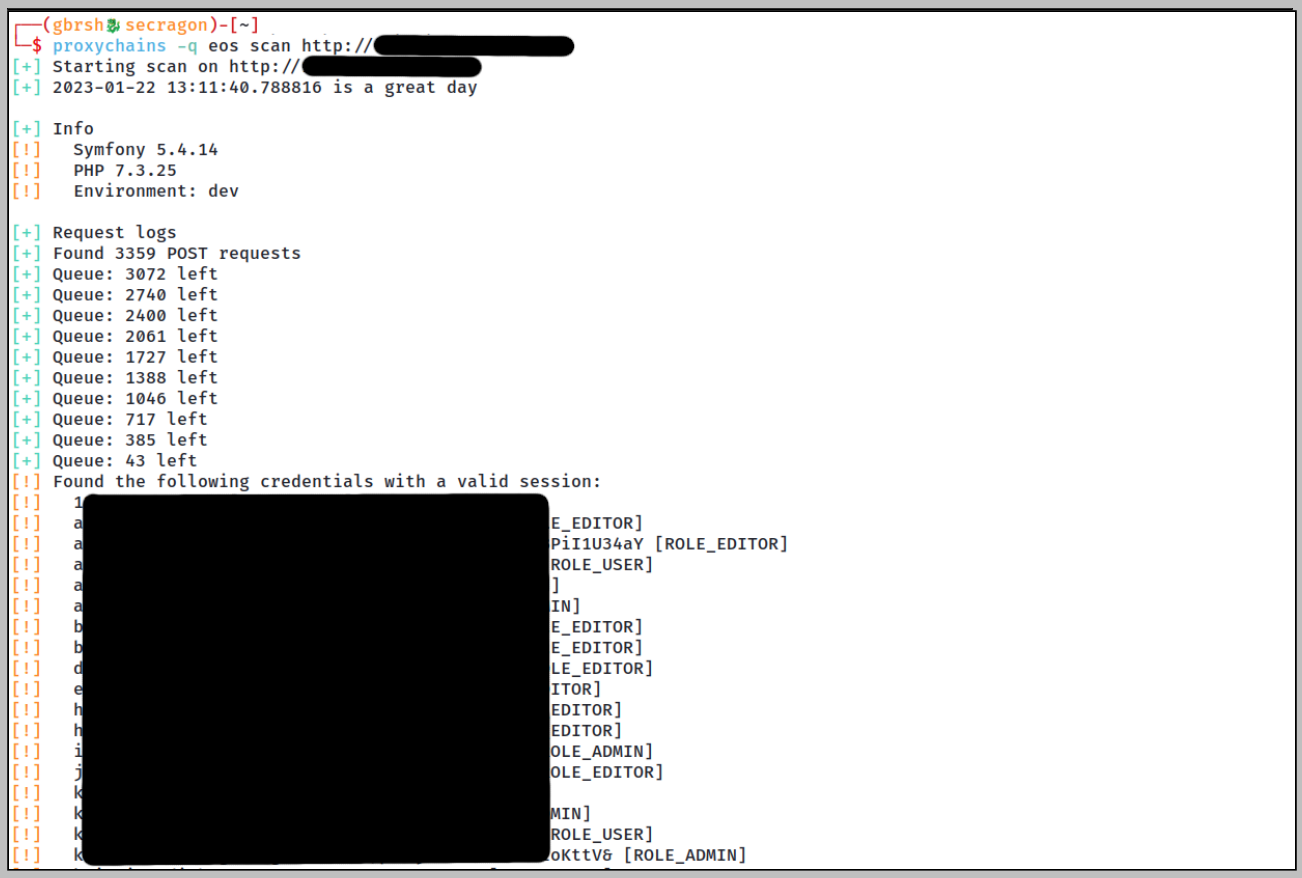
Attached screenshot from BloodHound for reference:



H4: Symfony Development Mode Exposure	
CVSS 3.1 Score	**8.2 (High)**
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N
Description	Symfony 5.4.14 runinng on port 8080 had enabled debugging, which gave an unauthenticated user access to sensitive information. Specifically, the vulnerability allowed unauthorized access to: • Plaintext request data, including potentially sensitive user inputs. • Configuration environment details, which might have included credentials or other secure configuration parameters. • Source code of the application, potentially exposing proprietary algorithms or business logic.
Impact	The exposure of this information could lead to unauthorized access to sensitive data or further exploitation of the system.
Target	x.x.x.240
Remediation	Disable development mode in Symfony to prevent unauthorized access to sensitive debugging information.
References	https://symfony.com/doc/current/configuration.html#selecting-the-active-environment

Finding Evidence

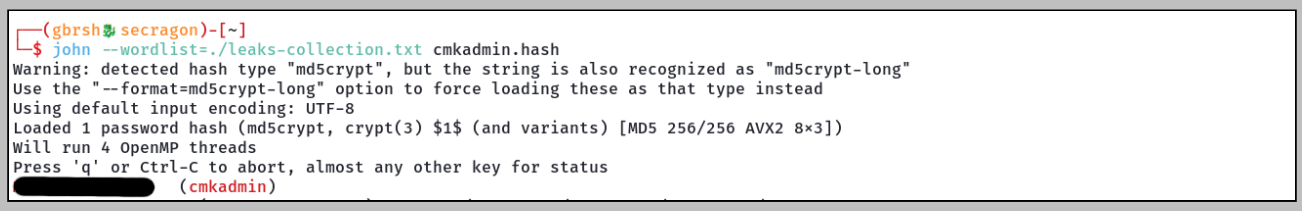
During network scanning, a web server running on port 8080 was discovered. The ***Wappalyzer*** plugin subsequently confirmed the presence of Symfony. Upon manual inspection of the `/_profiler/` URL, it was determined that the Symfony instance was in debug mode. Leveraging this misconfiguration with the ***eos tool*** tool, our team successfully extracted 45 credentials tied to valid sessions.



H5: Crackable Password for Cmkadmin Account	
CVSS 3.1 Score	**7.5 (High)**
CVSS Vector	CVSS:3.1/AV:L/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:H
Description	The password hash for the 'cmkadmin' account, obtained from the checkmk service exploitation, can potentially be cracked using lists of previously leaked passwords.
Impact	Successfully cracking the password could grant unauthorized access to systems. If the same or similar passwords are reused across different accounts or services, this could be leveraged by attackers through password spraying techniques, potentially accessing multiple assets in the network.
Target	x.x.x.100
Remediation	Implement a complex password for the 'cmkadmin' account, ensuring it contains a mix of uppercase, lowercase, numbers, and special characters to enhance security.
References	https://cwe.mitre.org/data/definitions/521.html

Finding Evidence

Through the exploitation of the **Checkmk RCE vulnerability**, a password hash for the 'cmkadmin' account was retrieved. Using the ***John the Ripper*** tool along with our company's collection of leaked passwords, the hash was successfully cracked, and the plaintext password was obtained.



M1: Weak Active Directory Account Passwords	
CVSS 3.1 Score	**6.8 (Medium)**
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:N/A:N
Description	Some Active Directory accounts within the environment utilized weak or previously leaked passwords, making them susceptible to password-cracking attempts. The usage of common or simplistic passwords increases the risk of unauthorized access.
Impact	The successful cracking of Active Directory account passwords could have led to unauthorized access. Given that these passwords were linked to Active Directory accounts, attackers could have potentially gained access to various resources and services within the network. Furthermore, these cracked credentials could have been leveraged for lateral movement, privilege escalation, or even launching further attacks like Pass-the-Ticket or Pass-the-Hash. Additionally, there was the risk of these credentials being used in potential password reuse attacks on other systems or platforms.
Target	******.local
Remediation	• Prompt users with identified weak passwords to change them immediately. • Enforce a robust password policy, which includes complexity requirements and regular rotations. • Educate users about the significance of unique and complex passwords. • Consider implementing Multi-Factor Authentication (MFA) for added security. • Regularly audit passwords for strength and ensure they are resilient against common password-cracking tools.
References	https://cwe.mitre.org/data/definitions/521.html

Finding Evidence

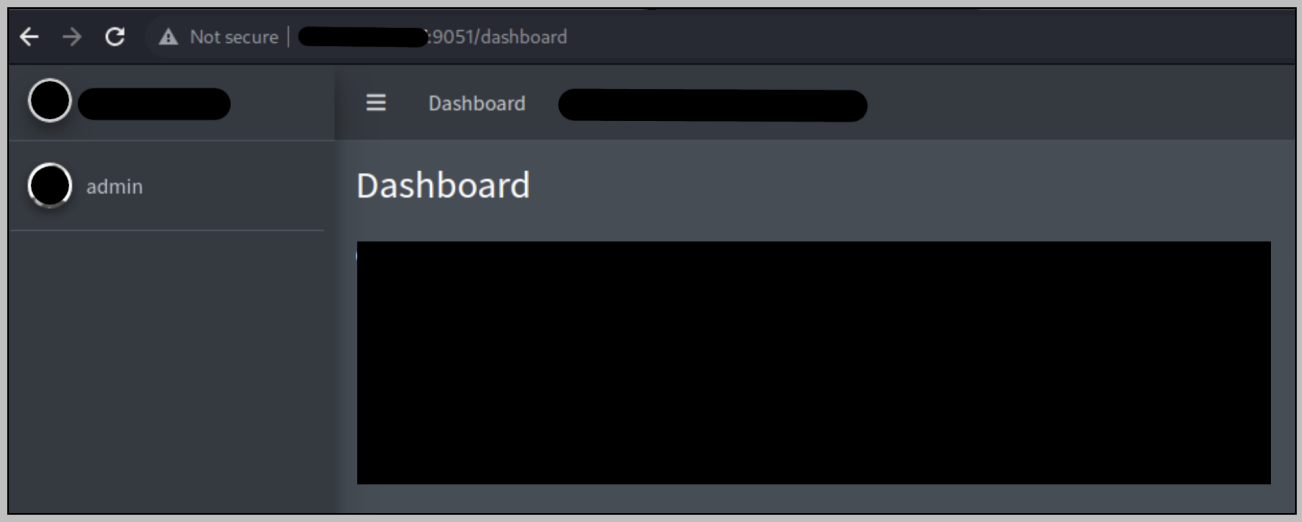
After intercepting LLMNR and NBT-NS queries on the network, ***several NTLMv2 hashes were captured***. Using '***John the Ripper***' and our company's collection of leaked passwords, two of these hashes were successfully cracked, revealing their plaintext passwords. The attached screenshot validates this successful decryption.

```
┌──(gbrsh㉿secragon)-[~]
└─$ john --wordlist=./leaks-collection.txt --format=netntlmv2 ntlmv2.hashes
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
[redacted]
[redacted]
```

M2: Weak Admin Password for Laravel Application	
CVSS 3.1 Score	**5.4 (Medium)**
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N
Description	A Laravel application running on port 9051 was found to utilize a weak administrative password. The password in question, "12345678", was manually guessed without the use of specialized tools, indicating a severe lack of complexity and security.
Impact	While the direct impact of this vulnerability might appear minimal due to no direct escalation vectors identified within the site, the implications are still significant. An attacker could misuse the administrative access to modify site contents, manipulate user data, or launch further attacks against site users. In addition, the fact that such a weak password was set raises concerns regarding other security practices that might be in place.
Target	x.x.x.207
Remediation	Update the password for the admin account to a complex, unique password that meets industry standards.
References	https://cwe.mitre.org/data/definitions/521.html

Finding Evidence

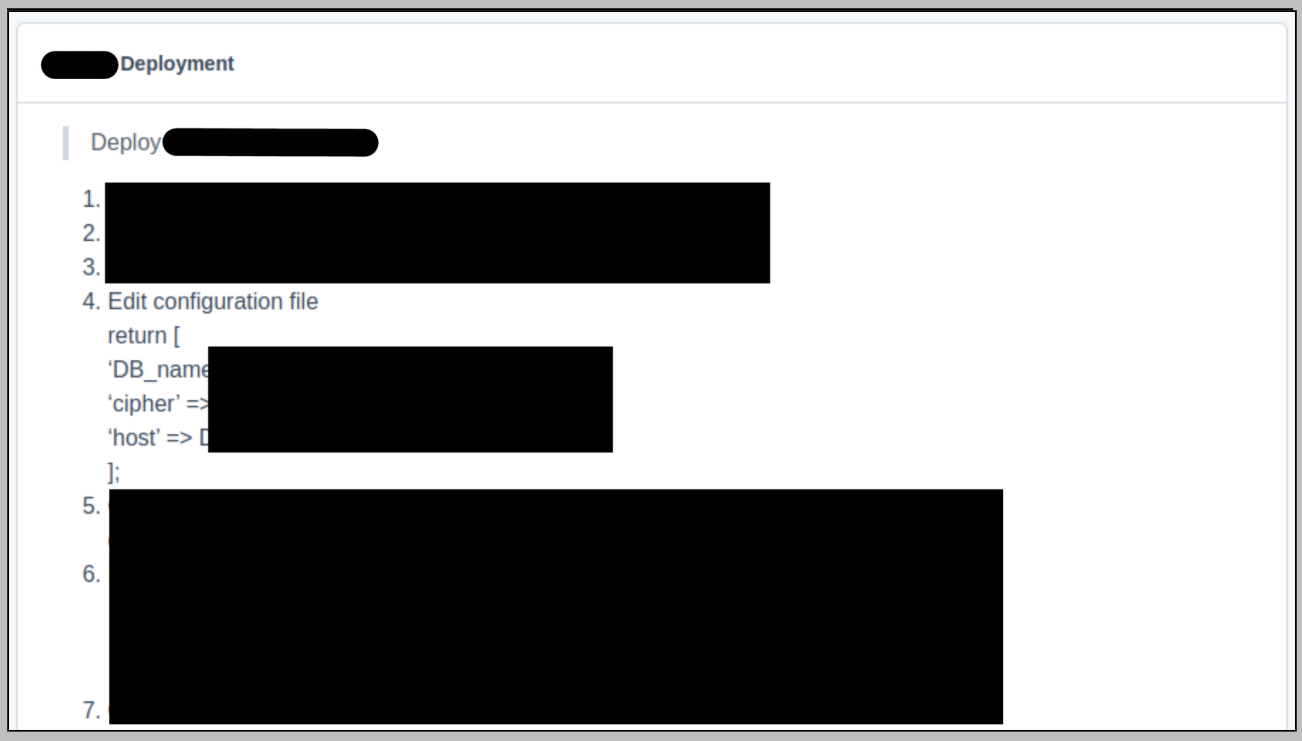
During manual examination of the Laravel site, the admin login was accessed and a set of common passwords were tried. The password "12345678" was identified as the valid password for the admin account. No further tools were used in this identification process.



M3: Unauthenticated Web Server Revealing Product Deployment Information	
CVSS 3.1 Score	**5.3 (Medium)**
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Description	An accessible web server running on port 9011 was identified that does not require authentication for access. This server contains detailed information regarding the company's product deployment processes.
Impact	The exposure of product deployment process details could aid adversaries in understanding the company's operational workflow. It might allow them to discover potential weaknesses or inefficiencies, gain insights for competitive advantage, or find potential points of attack for future malicious endeavors. Moreover, the disclosure of internal processes can damage the company's reputation and trustworthiness in the eyes of partners, customers, and investors.
Target	x.x.x.207
Remediation	Ensure that the web server requires proper authentication before granting access to any data or resources.
References	https://httpd.apache.org/docs/2.4/howto/auth.html

Finding Evidence

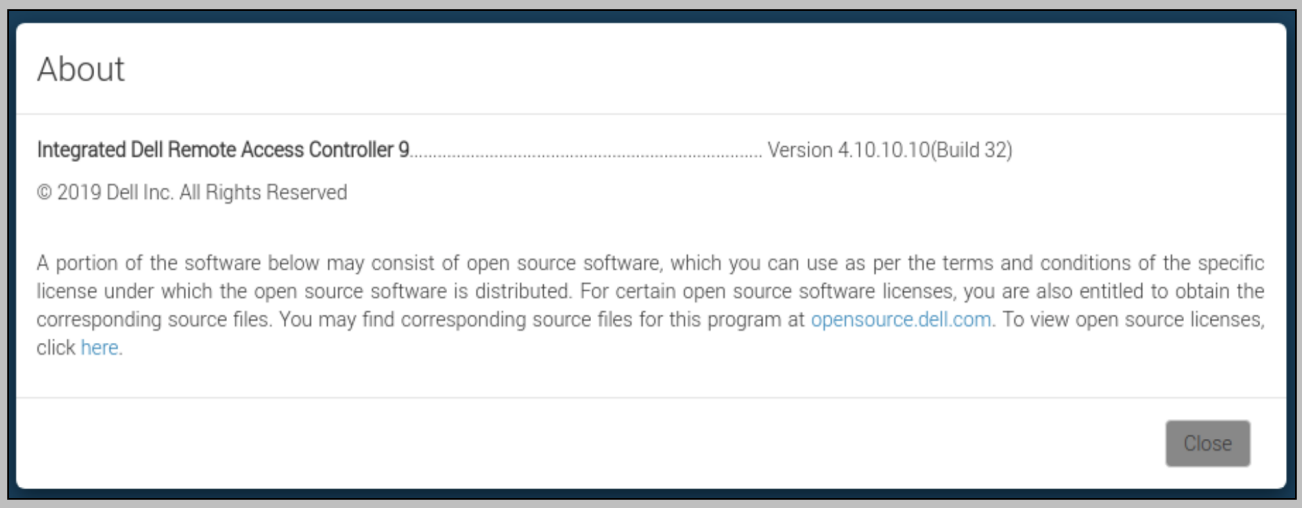
During the network enumeration phase, we discovered a web server hosted at x.x.x.207, port 9011. Upon accessing this server, it was evident that there was no authentication mechanism in place. The server revealed detailed information about the company's product deployment processes, which could be accessed freely without any credentials.



M4: Outdated iDRAC 9 Server	
CVSS 3.1 Score	**4.8 (Medium)**
CVSS Vector	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N
Description	During the assessment, an outdated iDRAC 9 server was identified on port 80. This version of iDRAC has numerous associated CVEs, indicating various vulnerabilities and potential risks.
Impact	While no unauthenticated remote code execution (RCE) vulnerabilities or public exploits have been identified for this version, the presence of known vulnerabilities poses potential security risks. It might leave the server exposed to attacks or exploitations if threat actors discover authenticated vulnerabilities or develop private exploits.
Target	x.x.x.54
Remediation	Update the iDRAC 9 server to the latest version to mitigate known vulnerabilities.
References	https://www.dell.com/support/kbdoc/en-us/000138130/how-to-update-the-idrac-integrated-dell-remote-access-controller

Finding Evidence

During network enumeration, an iDRAC 9 server was discovered open on port 80. The following screenshot showcases the outdated version identified on the front page of the server interface:



A Appendix - Exploited Systems

Host	Notes
x.x.x.100	Port: 8000, Remote Code Execution
x.x.x.240	Port: 8080, dumped credentials

B Appendix - Compromised Users

Username	Type	Notes
cmkadmin	Checkmk admin account	x.x.x.100:8000
root	system account	x.x.x.100
admin	web account	x.x.x.207:9051
redacted	domain account	
redacted	domain account	
redacted	domain account	
redacted	domain account	
redacted	domain account	
redacted	domain account	

C Appendix - Host Changes/Cleanup

Notes
All systems were restored to their initial state.