

Submission 1

by Gabriel Aragão

Submission date: 15-Jan-2026 02:49PM (UTC+0000)

Submission ID: 273407359

File name: 67166_Gabriel_Aragao_Submission_1_712353_673866881.docx (1.41M)

Word count: 5486

Character count: 36081

Abstract

Recently there have been a significant increase in the complexity, scale, and impact of cyberattacks targeting healthcare and pharmaceutical organizations, especially with the growth in IT fields. These entities represent high-value targets because of their reliance on sensitive patient data, intellectual property, and time-critical operational systems. Modern adversaries are increasingly operating as advanced persistent threats (APTs), which combine stealth, persistence, and strategic intent to bypass traditional perimeter-based defenses and avoid detection for extended periods.

This report presents a strategic and technical analysis of a recent major cyber incident affecting the healthcare sector, examining how real-world attackers leveraged specific tactics, techniques, and procedures (TTPs) to achieve operational objectives. The incident is systematically mapped into the MITRE ATT&CK® Enterprise framework to clarify the attack lifecycle and identify critical defensive failures that enabled the breach.

Based on these insights, the report develops and executes a simulated red team engagement, "Operation IP Shield", against a hypothetical pharmaceutical organization operating a hybrid infrastructure. This engagement simulates APT-style behavior to assess detection capabilities, architectural vulnerabilities, and organizational readiness. The findings are translated into both a technical report and an executive-level briefing for security teams, demonstrating how offensive security insights can be transformed into actionable, business-focused risk mitigation strategies.

Introduction

Both healthcare and pharmaceutical sectors have increasingly been targeted for cyber adversaries due to the high value data, research outputs, and operational dependencies. Unlike traditional attacks where they're financially motivated, many recent incidents demonstrate characteristics associated with Advanced Persistent Threats (APTs), including long term reconnaissance, careful lateral movement, and deliberate efforts to evade detection mechanisms. These attacks frequently aim not only to disrupt services but also to exfiltrate sensitive intellectual property or strategic research data.

Offensive security has evolved in response to this changing threat landscape. Conventional penetration testing, which often focuses on identifying isolated vulnerabilities, is no longer sufficient to assess an organisation's true resilience against modern adversaries. Instead, red team operations and adversary emulation exercises have emerged as more effective approaches, as they replicate the behaviour, intent, and tradecraft of real-world threat actors. Frameworks such as MITRE ATT&CK® provide a common analytical language that enables defenders and attackers alike to reason about adversary behaviour in a structured and evidence-based manner[1].

This report is structured around two interconnected objectives. The first is to conduct a strategic analysis of a recent, high-impact healthcare cyber attack, examining how the adversary progressed through the attack lifecycle and identifying the root causes that enabled

success. The second objective is to apply these insights within a controlled, ethical red team simulation targeting a hybrid pharmaceutical environment that includes on-premises infrastructure, cloud services, and modern CI/CD pipelines such as GitHub Actions.

By combining critical analysis of real-world incidents with hands-on threat emulation, this work moves beyond vulnerability discovery to explore detection gaps, architectural weaknesses, and organisational blind spots. The ultimate aim is not only to demonstrate technical competence, but also to translate offensive security findings into clear, defensible recommendations that support informed decision-making at both technical and executive levels[2]

Overview of NPM Maintainer Supply-Chain Compromise

In recent years, cyber adversaries have been moving away from attacking well-defended enterprise environments and have instead focused on exploiting trusted upstream dependencies. A clear example of this shift can be seen in the supply-chain attack against the JavaScript package ecosystem, npm, where legitimate maintainers within the npm repository were compromised. In this case, attackers successfully phished the credentials of recognised maintainers responsible for widely used packages and used this access to inject malicious code into trusted updates, which were then distributed to thousands of downstream users. Unlike traditional malware campaigns, this attack didn't depend on exploiting a technical vulnerability or deploying obviously malicious binaries. Instead, it relied on abusing the implicit trust placed in open-source maintainers and automated update mechanisms. By taking control of legitimate maintainer accounts, the attackers were able to publish altered package versions that appeared authentic, passed integrity checks, and were automatically integrated into dependent applications as part of routine build and deployment processes[3], [4].

Subsequent technical analysis showed that the malicious packages were designed to harvest sensitive data, including environment variables, authentication tokens, and cryptocurrency credentials. In several instances, the injected code executed only under specific runtime conditions, allowing it to remain dormant in most environments. This selective execution reduced the likelihood of detection and reflects tradecraft typically associated with advanced threats, where persistence and stealth are prioritised over immediate disruption[3].

Although the initial compromise occurred within a software development ecosystem rather than directly targeting a healthcare organisation, the relevance to pharmaceutical and healthcare sectors is significant. Modern healthcare platforms, clinical research systems, and pharmaceutical data portals rely heavily on web technologies and CI/CD pipelines that routinely consume third-party JavaScript libraries. As a result, a single compromised dependency within npm can propagate into systems responsible for processing sensitive patient data, clinical trial information, and proprietary research assets[4].

From a strategic standpoint, this incident illustrates how attackers can effectively bypass traditional security controls by operating upstream of the target environment. Controls such as network firewalls, endpoint detection and response solutions, and intrusion detection systems offer little protection when malicious code is introduced through trusted development workflows and executed as part of legitimate application logic. Existing research increasingly identifies software supply-chain compromise as a major blind spot in

organisational security models, particularly in highly regulated sectors where vendor and dependency trust is often assumed rather than verified[4].

For pharmaceutical organisations, the implications extend beyond immediate credential theft or financial loss. A compromised dependency integrated into research platforms or data processing pipelines could enable long-term intellectual property exfiltration, manipulation of research data, or covert access to cloud-hosted clinical systems. Such outcomes may remain undetected for extended periods, undermining research integrity and competitive advantage while creating significant regulatory and reputational risk[5].

Threat Actor Behaviour and MITRE ATT&CK Mapping

To properly analyse the npm supply-chain attack, this section focus on mapping the adversary's behaviour to the MITRE ATT&CK® framework. MITRE ATT&CK provides a structure diagram for describing how attackers achieve their objectives across the cyber chain, enabling consistent analysis of adversary work beyond isolated indicators or tools[6]

Credential Access and Persistence – Valid Accounts (T1078)

After the successful phishing attack, the attackers leveraged compromised maintainer credentials to authenticate to npm infrastructure. The use of valid accounts allowed the adversary to bypass security mechanisms designed to detect anomalous or unauthorised access attempts. Because the activity originated from trusted identities, actions such as publishing package updates appeared indistinguishable from normal maintainer behaviour. As long as credential access remained undiscovered, the attackers retained ongoing control over affected repositories, enabling repeated or staged malicious updates over time.

Figure 1 illustrates the end-to-end attack chain noticed in the npm maintainer supply-chain compromise. The diagram demonstrates how an initial identity-based compromise is used to propagate malicious code through trusted development workflows, enabling credential exploitation, repository access, and data exfiltration without triggering traditional security controls.

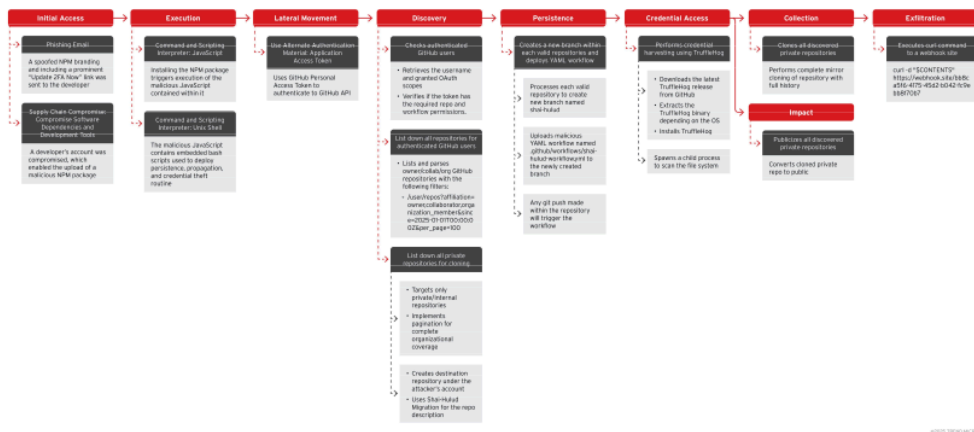


Figure 1: End-to-end attack chain for an identity-driven npm supply-chain compromise (adapted from incident analysis).

Defense Evasion – Trusted Developer Workflow Abuse

Rather than deploying a standalone malware, the attackers embedded malicious logic directly into legitimate package source code. This approach allowed the payload to be distributed, executed, and updated through normal development workflows. By operating entirely within expected software supply-chain processes, the attack evaded endpoint detection, antivirus tooling, and perimeter monitoring controls

This technique reflects a broader shift in adversary behaviour, where evasion is achieved through conformity rather than concealment. The malicious code behaved like normal application logic, significantly reducing the likelihood of detection.

3.4 Execution – User Execution via Trusted Packages (T1204)

Execution of the malicious payload occurred when downstream applications installed or updated the compromised npm packages as part of routine build or deployment processes. In many cases, this occurred automatically through CI/CD pipelines, requiring no direct user interaction beyond normal development activity[7], [6], [2].

Because execution took place within trusted application contexts, the malicious code inherited the same privileges as the host application, including access to environment variables, API keys, and cloud credentials.

Once authenticated using compromised maintainer credentials, persistence was established by programmatically modifying repositories via the GitHub API. Figure 2 illustrates an example of an automated script used to upload malicious workflow or placeholder files to attacker-controlled branches, ensuring continued execution through standard CI/CD triggers.

```
# Upload file to the new branch
echo " → Uploading $FILE_NAME to branch..."
FILE_DATA=$(jq -n \\\
  --arg message "Add $FILE_NAME placeholder file" \\\
  --arg content "$FILE_CONTENT_BASE64" \\\
  --arg branch "$BRANCH_NAME" \\\
  '{message: $message, content: $content, branch: $branch}')

FILE_RESPONSE=$(github_api PUT "/repos/$REPO_FULL_NAME/contents/$FILE_NAME" "$FILE_DATA")
FILE_ERROR=$(echo "$FILE_RESPONSE" | jq -r '.message // empty')

if [ -n "$FILE_ERROR" ] && [ "$FILE_ERROR" != "null" ]; then
  if [ "$FILE_ERROR" == "already exists" ]; then
    echo -e " ${YELLOW}⚠ File already exists on branch${NC}"
  else
    echo -e " ${RED}✗ Failed to upload file: $FILE_ERROR${NC}"
  fi
else
  echo -e " ${GREEN}✓ File uploaded successfully${NC}"
fi

echo ""
done

echo -e "${GREEN}🎉 Script execution completed!${NC}"
```

Figure 2: Code snippet used in the attack of a modification using GitHub API to establish persistence via trusted CI/CD workflows

3.5 Collection – Credential and Data Harvesting (T1552)

Once executed, the injected code focused on harvesting sensitive information available within the runtime environment. This sensitive information included credentials stored in environment variables, authentication tokens, and configuration data commonly used by applications to interact with external services. Such data is particularly valuable in cloud and healthcare research environments, where service-to-service authentication is pervasive[6].

This collection activity was typically lightweight and conditional, further reducing the likelihood of performance degradation or detection.

3.6 Exfiltration – Exfiltration Over Web Services (T1567)

Collected data was exfiltrated over standard web protocols, often using HTTPS connections to attacker-controlled endpoints. By blending exfiltration traffic with legitimate outbound application communications, the attackers avoided triggering network-based alerts or anomaly detection systems[8], [6].

This technique is particularly effective in cloud-native and healthcare environments, where outbound HTTPS traffic is both high-volume and business-critical.

3.7 Strategic Implications of the Attack Chain

Taken together, these TTPs illustrate a deliberate strategy focused on abusing trust relationships and automation rather than exploiting technical weaknesses. The attack chain demonstrates how identity compromise at the ecosystem level can cascade into widespread downstream impact, bypassing many of the controls traditionally relied upon by healthcare and pharmaceutical organisations. This behaviour aligns closely with modern APT-style operations, even when conducted by financially motivated actors.

The structured mapping of these behaviours to MITRE ATT&CK provides a foundation for the subsequent red team engagement. By emulating similar TTPs within a controlled hybrid environment, it becomes possible to evaluate whether existing defensive controls [9] can detect and disrupt comparable supply-chain driven threats.

4. Root Cause Analysis: Why the Attack Succeeded

The npm supply-chain compromise didn't result from a single technical vulnerability or isolated misconfiguration. Instead, it resulted from a combination of structural weaknesses, trust assumptions, and visibility limitations embedded within modern software development ecosystems. These weaknesses allowed attackers to progress from initial access to large-scale downstream compromise while remaining largely undetected. This section examines the root cause that enabled the attack, with particular focus on identity abuse, trusted automation, and defensive blind spots relevant to healthcare and pharmaceutical organisations.

4.1 Identity as a Primary Control Weakness

The most significant enabling factor in the attack was the compromise of legitimate maintainer identities through phishing. Once valid credentials were obtained, attackers were able to authenticate normally to npm and associated source control platforms, operating

entirely within expected access patterns. Because no technical exploitation occurred, security controls designed to detect unauthorised access were effectively bypassed[10].

This reliance on identity as a primary trust mechanism mirrors conditions commonly found in healthcare and pharmaceutical environments. Developers, researchers, and DevOps engineers often hold privileged access to repositories, build pipelines, and cloud services. When such identities are compromised, attackers inherit the same authority and legitimacy, creating a disproportionate security impact from a single failure.

4.2 Implicit Trust in Developer Workflows and Automation

A second critical root cause was the implicit trust placed in developer workflows and automation. Rather than introducing malicious artefacts through external delivery mechanisms, the attackers embedded malicious logic directly into legitimate packages and repository updates. Automated dependency resolution and CI/CD processes ensured that these changes were propagated widely without manual inspection[11], [7].

The use of authenticated GitHub API interactions to create branches, upload files, and trigger workflows illustrates how automation can be abused once trust boundaries are crossed. In healthcare research environments, where automation supports rapid development and deployment, such trust assumptions significantly expand the effective attack surface.

4.3 Wrong Authentication Hardening for High-Privilege Accounts

The success of the initial phishing campaign highlights insufficient authentication hardening for high-risk accounts. Password-based authentication, even when combined with basic multi-factor mechanisms, remains vulnerable to social engineering and credential theft. Phishing-resistant authentication methods were not enforced for maintainer accounts, enabling attackers to retain persistent access once credentials were stolen[12].

This weakness has direct relevance for pharmaceutical and healthcare organisations, where privileged accounts frequently control source repositories, CI/CD pipelines, and cloud infrastructure. Compromise of these identities allows attackers to pivot across environments without triggering perimeter-based or network-centric detection mechanisms.

4.4 Limited Visibility Into Supply-Chain Execution Contexts

Another major contributing factor was the lack of visibility into third-party code execution within trusted application contexts. Because the malicious logic executed as part of legitimate packages and workflows, endpoint detection and response solutions and network monitoring tools were largely ineffective. Activity generated by the malicious code closely resembled normal application behaviour, reducing the likelihood of alert generation[4].

This limitation is particularly concerning for healthcare environments, where applications routinely handle sensitive data and maintain persistent access to cloud services. Once malicious code executes within these trusted contexts, attackers can operate with the same privileges as legitimate workloads for extended periods.

4.5 Weak Dependency Governance and Update Validation

The attack also exposed deficiencies in dependency governance practices. Automated updates and transitive dependencies are often introduced without rigorous validation, code review, or provenance verification. In the absence of controls such as version pinning, code signing enforcement, or independent integrity verification, malicious updates can propagate rapidly across multiple systems[5], [3].

For regulated sectors such as healthcare and pharmaceuticals, this represents a systemic risk. Compromised dependencies may be deeply embedded within clinical applications or research tooling, making detection, remediation, and regulatory response significantly more complex.

4.6 Strategic Implications for Healthcare and Pharmaceutical Organisations

Taken together, these root causes demonstrate a fundamental misalignment between contemporary attacker strategies and prevailing defensive models. The npm supply-chain attack succeeded because security controls were optimised to detect external intrusion rather than abuse of trusted internal processes. Identity compromise, trusted automation, and limited behavioural visibility created conditions in which attackers could persist with limited resistance from existing controls.

For healthcare and pharmaceutical organisations, this misalignment poses a serious threat to the confidentiality, integrity, and availability of research data and clinical systems. Addressing these root causes requires a shift towards identity-centric security controls, improved visibility into software supply-chain behaviour, and governance mechanisms designed for modern, automated development environments.

The next section builds on this analysis by presenting strategic recommendations for prevention and detection that are both technically realistic and operationally appropriate for healthcare and pharmaceutical contexts.

Conclusion: Strategic Threat Analysis

This task examined an identity-driven software supply-chain attack targeting the npm ecosystem, using it as a case study to analyse how modern adversaries exploit trust relationships within development environments. The analysis demonstrated that the compromise of legitimate maintainer accounts enabled attackers to bypass traditional security controls and distribute malicious code through trusted workflows, resulting in widespread downstream exposure.

By mapping the attack to the MITRE ATT&CK framework, the report highlighted how techniques such as phishing, abuse of valid accounts, trusted workflow manipulation, and covert data exfiltration form a coherent and repeatable attack pattern. The root cause analysis showed that the success of the attack was not due to a lack of security controls, but rather to misplaced trust in identities, automation, and third-party dependencies.

For healthcare and pharmaceutical organisations, the relevance of this case study is clear. Modern research platforms, clinical systems, and development pipelines depend heavily on

third-party software and automated processes. When these dependencies are compromised upstream, sensitive data and intellectual property may be exposed without any direct breach of the target organisation's perimeter.

This task establishes that defending against contemporary threats requires a shift away from perimeter-focused models towards improved identity protection, supply-chain visibility, and governance. These findings provide the analytical foundation for the applied red team engagement presented in Task 2.

TASK 2: RED TEAM ENGAGEMENT

Operation IP Shield

6. Red Team Engagement Overview and Assumptions

This task presents a simulated red team engagement, code-named Operation IP Shield, conducted against a fictional pharmaceutical organisation, Cureton Pharmaceutical Company (CPC). The engagement is designed to apply the findings from Task 1 within a controlled, ethical testing environment in order to evaluate CPC's defensive posture against a realistic, identity-driven supply-chain threat.

CPC is assumed to operate a hybrid infrastructure that reflects common industry practice. This includes an on-premises Active Directory environment supporting research and development operations, an AWS-hosted clinical data portal, and modern CI/CD pipelines used to develop and deploy internal and external applications. These components were selected to reflect the types of systems that would be most attractive to adversaries seeking to steal proprietary research data or gain long-term access to pharmaceutical intellectual property.

The primary objective of the red team engagement is to emulate the behaviour of a sophisticated adversary rather than to identify isolated vulnerabilities. In particular, the engagement focuses on techniques associated with identity compromise, trusted workflow abuse, and covert data access, as identified in the npm supply-chain case study analysed in Task 1. The goal of the operation is to successfully access and exfiltrate a predefined sample of dummy research data without triggering detection by existing security controls.

Several assumptions are made to ensure the engagement remains realistic while respecting ethical and operational constraints. It is assumed that CPC employs baseline security controls typical of a mid-sized pharmaceutical organisation, including endpoint protection, centralised logging, and standard authentication mechanisms. The engagement explicitly excludes denial-of-service attacks, destructive actions, or access to real patient data. All activities are conducted within an agreed scope and are intended to support defensive improvement rather than operational disruption.

This section establishes the context for the red team operation. The following sections define the rules of engagement, threat emulation plan, and execution approach used to assess CPC's resilience against modern supply-chain driven attacks.

7. Rules of Engagement (RoE)

The red team engagement was conducted under a defined set of Rules of Engagement to ensure that all activities remained ethical, controlled, and aligned with the objectives of the assessment. The RoE establishes the scope, boundaries, and constraints of the operation, reflecting professional standards commonly applied in real-world red team assessments.

7.1 Scope of Engagement

The scope of the engagement includes the following simulated assets belonging to Cureton Pharmaceutical Company (CPC):

- An on-premises Active Directory environment supporting research and development users
- An AWS Virtual Private Cloud hosting a clinical data portal and supporting services
- Github Actions CI/CD pipelines used for application build and deployment, including source code repositories and automated workflows

The engagement focuses exclusively on systems and identities associated with these components. No third-party organisations, external production services, or real patient data are included within scope.

7.2 Objectives

The primary objective of the engagement is to emulate a sophisticated adversary seeking to gain unauthorised access to proprietary research data. Success is defined as the ability to access and exfiltrate a predefined sample of dummy research data without triggering detection by existing security controls.

Secondary objectives include:

- Evaluating CPC's ability to detect identity-based compromise
- Assessing visibility into CI/CD and software supply-chain activity
- Identifying gaps in monitoring, logging, and alerting coverage

7.3 Authorised Techniques and Constraints

The red team is authorised to employ techniques consistent with realistic adversary behaviour, including phishing-based identity compromise, abuse of valid credentials, and manipulation of trusted development workflows. Any type of activities that might impact the system availability and services will not be tolerated. Living-off-the-land techniques and legitimate system utilities may be used where appropriate to minimise operational noise.

The following activities are explicitly excluded:

- Denial-of-service attacks

- Destructive actions or data modification
- Exploitation of zero-day vulnerabilities
- Access to real patient or clinical data

These constraints ensure that the engagement remains non-disruptive and focused on confidentiality and detection rather than availability.

7.4 Communication and Safety Controls

A clear communication protocol is assumed between the red team and CPC stakeholders. A notional point of contact is designated for escalation in the event of unexpected system instability or unintended impact. A kill-switch condition is defined, under which all red team activity would immediately cease if critical business operations were affected.

All findings generated during the engagement are documented for post-operation reporting and are intended solely to support defensive improvement.

7.5 Assumptions and Limitations

The engagement assumes baseline security controls typical of a mid-sized pharmaceutical organisation, including endpoint protection, centralised logging, and standard authentication mechanisms. As the engagement is conducted as a controlled simulation, findings are based on observed behaviour within the defined environment and may not capture all real-world operational complexities.

8. Threat Emulation Plan and Adversary Profile

8.1 Adversary Profile

The threat emulated in Operation IP Shield reflects a financially motivated but highly capable adversary whose tactics and operational discipline resemble those commonly associated with Advanced Persistent Threats. Rather than relying on technical exploits or disruptive malware, the adversary prioritises identity compromise, abuse of trusted workflows, and covert access to sensitive data. This profile is consistent with the real-world npm maintainer supply-chain attack analysed in Task 1, where compromise of legitimate developer identities enabled large-scale downstream impact.

The adversary's assumed objectives are long-term and strategic. The primary goal is the theft of proprietary pharmaceutical research data, while secondary goals include establishing persistent access to development and cloud environments to enable future exploitation. Stealth and evasion are prioritised over speed, reflecting an intent to remain undetected for extended periods.

8.2 Threat Emulation Objectives

The threat emulation plan is designed to test CPC's ability to detect and respond to identity-driven attacks that operate within trusted operational contexts. The emulation focuses on the following objectives:

- Compromise of a developer or DevOps identity associated with CI/CD workflows

- Abuse of trusted automation to access cloud-hosted research systems
- Lateral movement across environments using valid credentials
- Access and exfiltration of a defined sample of dummy research data without detection

These objectives align directly with the attack patterns identified in the npm supply-chain case study and reflect realistic adversary decision-making.

8.3 Planned Attack Phases and MITRE ATT&CK Alignment

The emulated campaign follows a phased approach that mirrors observed adversary behaviour. Each phase is mapped to relevant MITRE ATT&CK tactics and techniques to ensure consistency and analytical clarity.

Reconnaissance and Targeting

The adversary identifies publicly exposed developer information and repository metadata to select high-value identities associated with CI/CD pipelines and cloud access.

Initial Access

Phishing techniques are used to obtain valid credentials associated with a developer or pipeline-maintainer account, reflecting identity-based compromise rather than vulnerability exploitation.

Persistence and Privilege Use

Once access is obtained, persistence is achieved through continued use of valid credentials and trusted access paths, including repository access and automated workflow triggers.

Lateral Movement and Access Expansion

The adversary leverages inherited permissions and service-to-service trust to access cloud-hosted resources connected to the compromised pipeline.

Collection and Exfiltration

Sensitive research data is accessed through legitimate application interfaces and exfiltrated using standard encrypted web protocols to avoid detection.

This phased structure ensures that the emulation tests detection and visibility across identity, CI/CD, and cloud layers rather than focusing on a single control failure.

8.4 Success Criteria

The threat emulation is considered successful if the red team can access and exfiltrate the predefined dummy research data without generating high-confidence alerts or triggering an incident response. Partial success is defined as detection occurring only after data access has been achieved.

These criteria are intentionally aligned with adversary objectives rather than defensive metrics, ensuring that the engagement evaluates real-world risk rather than tool performance alone.

9. Simulated Environment Architecture and Design

The simulated environment used for Operation IP Shield was designed to replicate a realistic hybrid infrastructure commonly found within companies that use hybrid environments such as CPC. The architecture intentionally reflects environments where research, development, and clinical data systems are tightly integrated with cloud services and automated deployment pipelines. The design prioritises realism over complexity to focus the assessment on adversary behaviour and defensive visibility rather than infrastructure scale. In a real organisation, additional legacy constraints would likely complicate this design.

9.1 On-Premises Environment

The on-premises component of the environment represents CPC's research and development network. It consists of a Windows Active Directory domain used for identity management, authentication, and access control for developers and research staff. Domain-joined workstations are assumed to be used for software development and administrative tasks, with authentication federated to cloud services where required.

This environment is intentionally modelled with standard enterprise configurations, including shared identity services and centralised authentication. This reflects common practice within pharmaceutical organisations and creates realistic conditions under which identity compromise can enable lateral movement and privilege reuse across environments.

9.2 Cloud Environment (AWS)

The cloud environment of the simulated environment is hosted within an AWS Virtual Private Cloud. This environment represents CPC's clinical data portal and supporting services used to store and process research-related information. Application workloads are assumed to run on managed compute services and access data stores using identity-based permissions rather than static credentials.

Connectivity between the on-premises environment and the cloud is assumed to be established through secure integration mechanisms, enabling authenticated users and services to access cloud-hosted resources. This hybrid trust relationship reflects a common architectural pattern and provides a realistic pathway for adversaries to pivot from identity compromise into cloud workloads.

9.3 CI/CD Pipeline Architecture

A central element of the simulated environment is the CI/CD pipeline used for application development and deployment. Source code repositories are assumed to be hosted on a third-party platform, with automated workflows responsible for building, testing, and deploying applications to the cloud environment.

These pipelines are configured to consume third-party dependencies automatically and to execute build steps using inherited credentials and environment variables. This design mirrors the conditions exploited in the npm supply-chain attack analysed in Task 1 and provides a credible mechanism through which trusted automation can be abused following identity compromise.

9.4 Trust Relationships and Attack Surface

The architecture deliberately includes implicit trust relationships between identities, pipelines, and cloud resources. Developers authenticated to source repositories are assumed to have permissions that allow workflows to access cloud services. Build systems are trusted to retrieve dependencies and execute code without manual validation at runtime.

These trust relationships form the primary attack surface evaluated during the red team engagement. Rather than exposing unnecessary services or vulnerabilities, the environment reflects a realistic scenario in which security controls exist but are misaligned with identity-driven and supply-chain based threats.

9.5 Design Rationale

The simulated architecture was designed to support the objectives of the threat emulation rather than to replicate a full production environment. Each component was selected to enable evaluation of specific detection and visibility challenges, including identity misuse, trusted workflow abuse, and covert access to sensitive data.

By focusing on a simplified but representative hybrid design, the environment allows for clear attribution of attack paths and defensive gaps. This approach aligns with professional red team methodology, where clarity and analytical value are prioritised over unnecessary architectural complexity.

The simulated hybrid architecture, including the AWS VPC, identity boundaries, CI/CD trust relationships, and data flows evaluated during Operation IP Shield, is illustrated in Appendix A.

The workflow shown in Appendix B is executed within a trusted GitHub Actions pipeline and deploys JavaScript artefacts to AWS services using inherited permissions with AWS IAM Role assuming. No credential theft or vulnerability exploits are required; access is granted through official federated authentication between GitHub OIDC and AWS OIDC. From a defence perspective, this activity is indistinguishable from a standard operational behavior, showing how trusted CI/CD pipelines can be abused following identity compromise.

10. Detection Gap Analysis

The red team engagement for Operation IP Shield uncovered several systemic blind spots. These gaps allowed the campaign to reach its objective of data exfiltration without triggering a high-priority incident response. These failures are not isolated tool errors. They represent a deeper misalignment between CPC's defensive assumptions and modern identity-driven attack patterns

10.1 Identity-Based Detection Gaps

The initial breach succeeded because it used valid credentials stolen through phishing. Since the adversary authenticated as a legitimate maintainer, the system recorded a "successful login" rather than an "unauthorized access attempt". Standard perimeter controls and malware signatures were ineffective here because the attacker didn't use any malicious files to gain entry. The detection failed because the security team lacked a behavioural baseline for developer identities. This allowed the misuse of valid accounts to go unnoticed until the attack moved into the cloud environment.

10.2 Limited Visibility Into CI/CD Activity

The GitHub Actions workflows executed exactly as the files intended. The build and deployment processes were treated as trusted zones inside the CPC environment. There was almost no monitoring of how these workflows were changing over time or what new dependencies were being pulled in. Because the organization viewed the pipeline as "trusted automation," they missed the moment a compromised account inserted a malicious step into the deployment chain.

10.3 Insufficient Monitoring of Cloud Workload Behaviour

Once the operation moved into the AWS VPC, the actions took place through inherited permissions and official OIDC roles. Standard cloud access logs recorded the data requests but could not distinguish them from routine research syncing. The detection gap here was a lack of behavioural modelling for service roles. Without knowing what "normal" data access looks like for a specific research workload, the monitoring tools simply treated the exfiltration as standard operational traffic.

10.4 Data Access and Exfiltration Blind Spots

Access to the "made up" dummy research data didn't require privilege escalation or unusual queries. Data was accessed through normal application logic and exfiltrated using standard encrypted web protocols. Because outbound traffic patterns aligned with expected application behaviour, network-based detection mechanisms were not triggered.

10.5 Strategic Implications of Detection Failure

Collectively, these gaps demonstrate that CPC's detection capabilities were optimised for detecting external threats rather than abuse of trusted internal processes. Identity misuse, workflow abuse, and low-volume data exfiltration blended into normal operational activity.

This misalignment significantly increases the likelihood of long-term, undetected compromise in environments handling sensitive pharmaceutical research data.

11. Executive Summary and Business Risk Implications

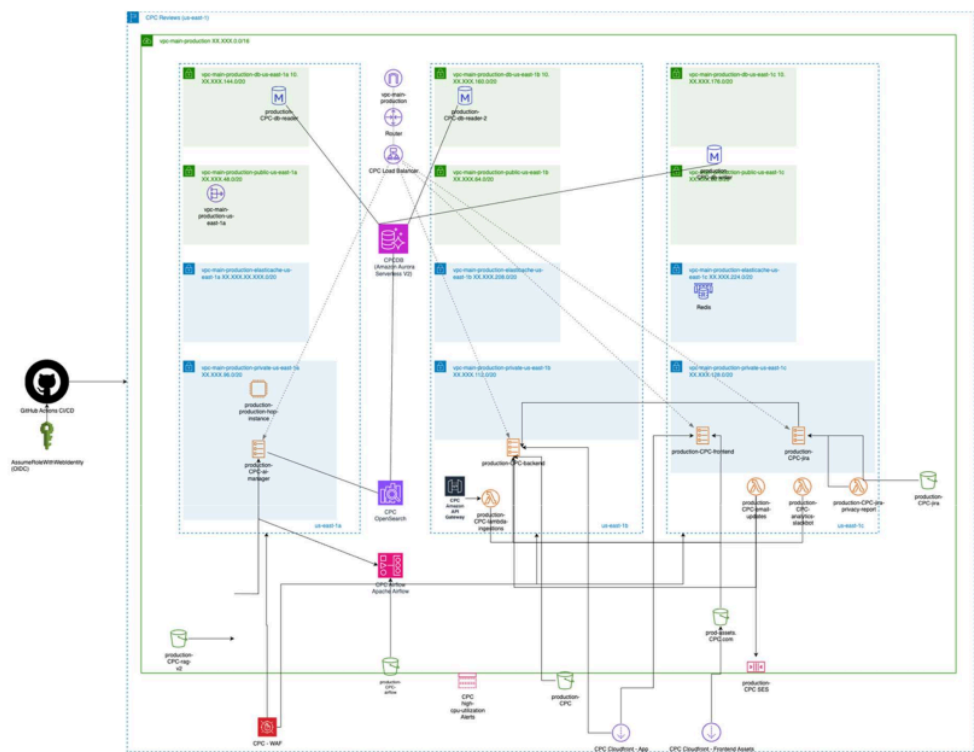
The red team engagement conducted as part of the operation demonstrates how an expert adversary could exploit trusted identities and automated workflows to access sensitive research data without triggering timely detection. The simulated attack didn't rely on advanced exploits or destructive malware, that would cause alarm. Instead, it leveraged normal operational processes, highlighting a category of risk that is often underestimated by organisations with mature perimeter and endpoint controls.

From a business perspective, the primary risk identified is the silent compromise of intellectual property. Unlike disruptive ransomware attacks, supply-chain and identity-driven intrusions may remain undetected for extended periods, allowing adversaries to extract high-value research data while business operations continue uninterrupted. For a pharmaceutical organisation, such an outcome could undermine competitive advantage, invalidate research investment, and expose the organisation to regulatory and reputational harm.

The engagement also highlights a misalignment between existing security investments and contemporary threat behaviour. While controls may be effective at detecting external intrusion or commodity malware, they offer limited protection against abuse of trusted developer identities, CI/CD pipelines, and cloud service roles. This creates a false sense of assurance, where systems appear secure while critical trust relationships remain insufficiently monitored.

In conclusion, this engagement shows that modern cyber risk in the pharmaceutical sector is increasingly driven by trust abuse rather than technical exploitation. Red team exercises grounded in real-world attack patterns provide valuable insight into these risks and support informed decision-making at an executive level.

Appendix A – Simulated Hybrid Environment Architecture



Appendix B – JavaScript CI/CD Pipeline Evidence (GitHub Actions)

```
1 name: CPC Backend - CD Not Committed Yet
2 concurrency:
3 group: ${{ github.ref }}
4 cancel-in-progress: false
5
6 on:
7   push:
8     branches:
9       - develop
10      - beta
11      - main
12      - preprod
13
14 permissions:
15   id-token: write
16   contents: read
17
18 env:
19   ECS_SERVICE: ${{ fromJson('{"develop":"staging-CPC", "beta":"staging-CPC-beta", "main":"production-CPC", "preprod":"preprod-CPC"}') }}[github.ref_name]
20   ECS_CLUSTER: CPC-cluster
21   AWS_REGION: ${{ secrets.AWS_REGION }}
22   AWS_DEFAULT_REGION: ${{ secrets.AWS_REGION }}
23   ENV_FILE: ${{ fromJson('{"develop":"deploy/staging.env", "beta":"deploy/beta.env", "main":"deploy/production.env", "preprod":"deploy/preprod.env"}') }}[github.ref_name]
24   VERSION: app-{{ github.sha }}
25
26
27 jobs:
28   build-app-image:
29     runs-on: ubuntu-24.04
30
31     steps:
32       - name: Checkout
33         uses: actions/checkout@v4
34
35       - uses: actions/setup-node@v3
36         with:
37           node-version: 18
38
39       - run: set -u
40       - run: npm install -g yarn
41       - run: yarn install --skip-integrity-check
42       - env:
43           MUSKY_SKIP_INSTALL: true
44       - run: cp ${{ env.ENV_FILE }} .env
45       - run: NODE_ENV=production yarn build
46       - run: yarn install --frozen-lockfile --production
47       - env:
48           MUSKY_SKIP_INSTALL: true
49
50       - name: Build and send images to AWS
51         id: build-docker-images
52         uses: ./github/actions/build-docker-images
53         with:
54           aws_iam_role: ${{ secrets.AWS_ROLE_AWS }}
55           aws_region: ${{ env.AWS_REGION }}
56           ecr_url: ${{ secrets.AWS_ACCOUNT }}.dkr.ecr.${{ secrets.AWS_REGION }}.amazonaws.com/${{ env.ECS_SERVICE }}
57
58
59   deploy-ecs-task:
60     runs-on: ubuntu-24.04
61     needs:
62       - build-app-image
63
64     steps:
65       - name: Checkout
66         uses: actions/checkout@v4
67
68       - name: Update ECS Service
69         id: update-ecs-service
70         uses: ./github/actions/update-ecs-service
71         with:
72           aws_iam_role: ${{ secrets.AWS_ROLE_AWS }}
73           aws_region: ${{ env.AWS_REGION }}
74           ecr_url: ${{ secrets.AWS_ACCOUNT }}.dkr.ecr.${{ secrets.AWS_REGION }}.amazonaws.com/${{ env.ECS_SERVICE }}
75           ecs_service: ${{ env.ECS_SERVICE }}
76           ecs_cluster: ${{ env.ECS_CLUSTER }}
77           env_file: ${{ env.ENV_FILE }}
78           version: ${{ env.VERSION }}
```

Figure 3: GitHub Actions CI/CD workflow demonstrating trusted execution with inherited AWS permissions via federated identity (OIDC), used during Operation IP Shield.

References

- [1] M. ATT&CK, "Mitre att&ck framework," vol. 24, pp.2024, 2024,
- [2] D. Selmanaj, *Adversary Emulation with MITRE ATT&CK*, " O'Reilly Media, Inc.", 2024.
- [3] T. A. Pham, "Supply Chain Attacks Through Open Source Software: A Comprehensive Analysis of NPM, PyPI, and Docker Hub Vulnerabilities," 2025,
- [4] A. Kuzmickas, S. Chan, R. Spousta III and S. Sala, "Blind Spots and Counterfeits in the Supply Chain," pp.43, 2015,
- [5] F. D. G. Solfa, "Impacts of cyber security and supply chain risk on digital operations: evidence from the pharmaceutical industry," vol. 2, no. 2, pp.18–32, 2022,

- [6] G. Skjøtskift, M. Eian and S. Bromander, "Automated attack technique chaining," vol. 6, no. 1, pp.1–11, 2025,
- [7] U. Lee and C. Park, "SoFTEE: Software-based trusted execution environment for user applications," vol. 8, pp.121874–121888, 2020,
- [8] C. Lim, B. Jin and H. Kim, "Stealth Extension Exfiltration (SEE) Attacks: Stealing User Data without Permissions via Browser Extensions", *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, pp. 1820–1828.
- [9] B. Venkata, "Red Team vs. Blue Team: Assessing Cybersecurity Resilience Through Simulated Attacks," 2021,
- [10] S. Eggers, "A novel approach for analyzing the nuclear supply chain cyber-attack surface," vol. 53, no. 3, pp.879–887, 2021,
- [11] M. Wessel, T. Mens, A. Decan and P. R. Mazrae, "The github development workflow automation ecosystems," in *Software Ecosystems: Tooling and Analytics*, Springer, 2023, pp. 183–214.
- [12] M. Bock, " Measuring Adoption of Phishing-Resistant Authentication Methods on the Web," 2023,

Submission 1

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

85/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

GRADING FORM: L7 - (ETHICAL HACKING)

GABRIEL ARAGÃO**84****ENGAGEMENT WITH LIT (15)***Engagement with Literature Skills*

Use of high-quality academic, industry and threat-intelligence sources. Referencing is accurate, consistent and at a scholarly standard

13**KNOWLEDGE & UNDERSTANDING (10)***Knowledge and Understanding Skills*

Demonstrates exceptional understanding of modern adversary behaviour, supply-chain compromise and identity-driven attacks.

9**COGNITIVE & INTELLECTUAL (40)***Cognitive and Intellectual Skills*

Outstanding critical analysis, synthesis and strategic reasoning. Task 1 and 2 done at an expectational level.

37

PRACTICAL APPLICATION (30)

Practical Application Skills



Very strong threat emulation and red-team design, but could provide more hands-on execution evidence (screenshots, tool output).

22

TRANSFERABLE SKILLS (5)

Transferable Skills for Life and Professional Practice



No Cover page, good academic writing and structure. However, paragraphs could be fully justified and the overall report could do with some formatting such as starting main headings on a new page to make it more professional.

3