FINAL REPORT

Penetration testing
Festival Finance

Table of Contents

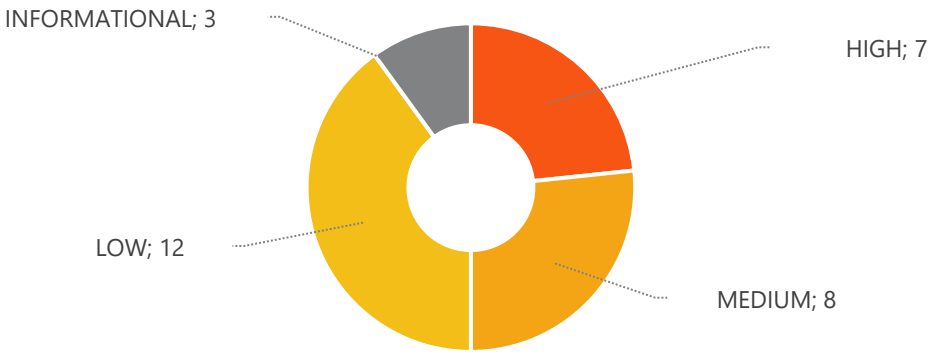
1. Executive summary

This document describes the penetration test results for Festival Finance, conducted by the Active Audit Agency team led by Eugene Ermolaev. The report provides a detailed assessment of identified cybersecurity risks and recommendations for mitigating these risks.

To evaluate the actual security of the systems, the team simulated the actions and capabilities of the following threat profiles: External Attacker, External Attacker+, and Insider with Remote Access.

The team conducted a comprehensive information security assessment using various tools and methods to identify cybersecurity deficiencies and manually check for potential vulnerabilities. The following areas of the company were tested: the external network perimeter, public-facing web applications, the back-office administration application (internal-facing web surface), social engineering against employee mailboxes, and the Active Directory environment together with the corporate Microsoft 365 tenant.

As a result of this assessment, the team identified and confirmed the presence of 25 cybersecurity risks, particularly:



We evaluated the overall state of cybersecurity as follows:

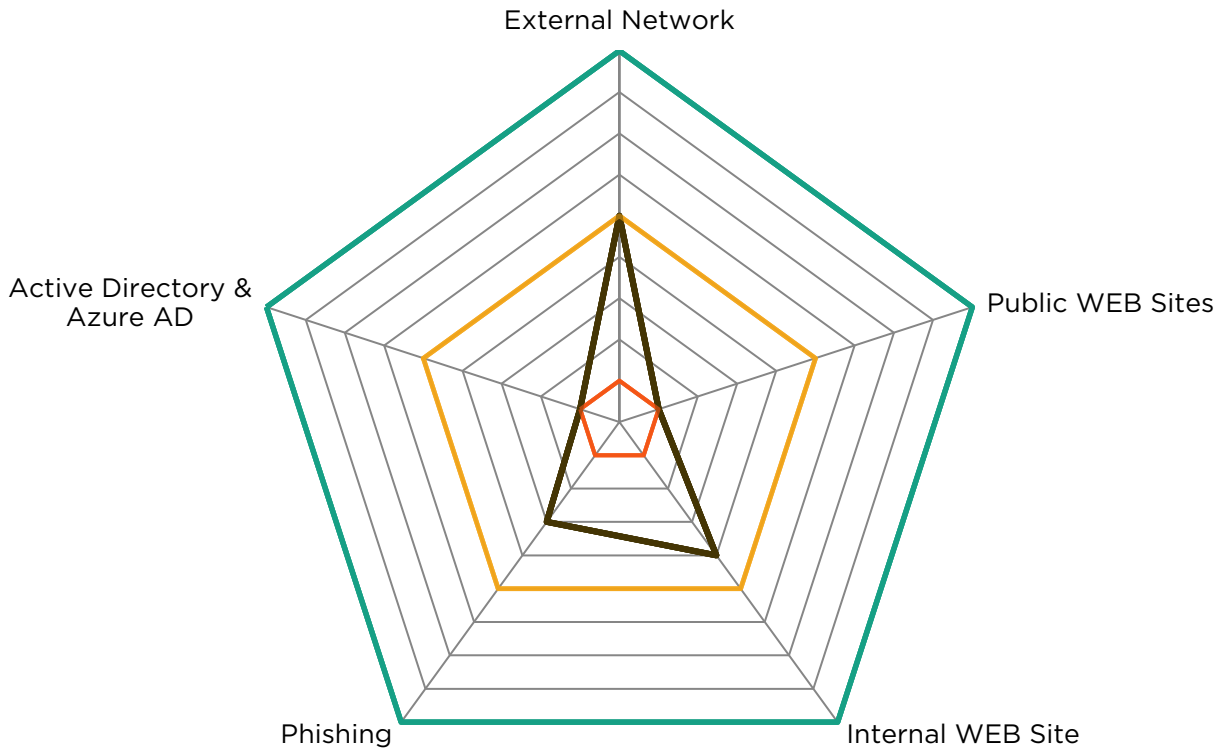
Four HIGH-severity risks, in combination, demonstrate full compromise paths from the bulk identity surface to enterprise-management platforms and customer data. The realized impact - administrative access to the platform that manages every corporate laptop, control of a regional production site's surveillance-and-gate-control console, cross-customer leakage in the corporate portal, and mass exposure of confidential personal-data documents in the partner portal - places the overall posture in the **UNSATISFACTORY** category. These outcomes are not theoretical; each chain was demonstrated end-to-end during the engagement. Granular threat-by-threat capability descriptions follow in the «Assessment of Simulated Threat Capabilities» section; per-risk details and recommendations provided below.

On the positive side, the defensive stack performed well on the social-engineering vector: the Security Operations Centre contained the fastest phishing-compromised account in 14 minutes, the joint Red Team / Blue Team retest confirmed remediation of the supporting email-infrastructure weakness, and engagement-rules discipline on both sides reflected a mature engagement culture. This defensive maturity has not yet been extended to the application and Active Directory surfaces.

The remediation path is focused and well-understood - immediate execution is required. The most impactful improvements are completing the multi-factor-authentication rollout across cloud services, the VPN, and remote desktop; banning predictable passwords; enforcing object-level authorization in customer-facing applications; and fixing the email-authentication record. Each item is a routine fix with established tooling; together they close the chains demonstrated during this engagement.

Level of information security (expert review)

Attack vector	Protection level	Risks			
		High	Medium	Low	Informational
External Network	**6**	0	1	1	1
Public WEB Sites	**2**	2	2	3	2
Internal WEB Sites	**5**	0	2	1	1
Phishing	**4**	0	1	0	0
Active Directory & Azure AD	**2**	2	2	3	1



Legend: protection level gradation scale from 0 to 10, where the level:

2 — Systems are not protected. A flaw exists that leads to a complete compromise of the system or its part, leading to a breach of the integrity of accessibility and confidentiality. IS processes are not performed consistently.

6 — Sufficient level of security. There is no possibility of easy systems compromise. Basic IS processes are performed, but not entirely. Systems can be compromised only partially, or vulnerabilities exist that are not publicly known, or vulnerabilities exist only exploitable from complex contexts (for example, locally)

10 — IS processes are performed on an ongoing basis, the systems are reliably protected, and best security practices are applied.

— Current protection level

Assessment of Simulated Threat Capabilities

As a result of the testing, several cybersecurity risks were identified, including 4 HIGH, 8 MEDIUM, 8 LOW and 5 INFORMATIONAL

The capabilities of simulated threats in the context of the identified risks are described below:

External Attacker - HIGH RISK

13 risks are relevant for this threat, of them: **2 HIGH, 5 MEDIUM, 3 LOW, 3 INFORMATIONAL**.

This profile applies to a pure outsider on the public Internet, optionally holding a self-registered customer account of the kind any prospective Festival Finance customer can obtain through the open registration flow.

The found risks of the **HIGH** level of criticality allow the External Attacker profile to do the following:

- **Browse and download records and personal data of other corporate customers** of the Corporate Portal. Three different parts of the application return another customer's data when the audit team changes a single number in the URL. Any registered B2B customer of Festival Finance can perform this with no further privileges (WEB-R1).
- **Download 4,612 confidential documents - national IDs, driver's licences, financial statements, vehicle titles - from the partner portal's file storage with nothing more than a self-registered customer account (WEB-R5).**

The found risks of the **MEDIUM** level of criticality allow the External Attacker profile to do the following:

- **Read an internal monitoring dashboard** that is reachable from the public Internet without any login. The dashboard contains failed login attempts, internal system names, and **live login tokens that can be replayed to impersonate users** (WEB-R2).
- **Use the mobile app's "remind me my PIN" feature to receive another customer's card PIN** in the attacker's own email inbox, by changing the card identifier the app sends in the background. Any registered B2C customer can do this (WEB-R3).
- **Send email that appears to come from any internal employee**. A small typo in the company's email-authentication record bypasses the protection that would normally block this kind of impersonation (NET-EXT-R1).
- **Run a phishing campaign that defeats multi-factor authentication, using a modern technique that captures the user's login session in real time. The campaign against 187 employees captured four live Microsoft 365 sessions; the Security Operations Centre contained the fastest case in 14 minutes (SOCIAL-R1).**

Other **LOW**-level and **INFORMATIONAL** risks do not directly enable attacks on specific systems but may allow the acquisition of additional valuable data or represent deviations from best practices.

External Attacker+ - HIGH RISK

13 risks are relevant for this threat, of them**: 2 HIGH, 5 MEDIUM, 3 LOW, 3 INFORMATIONAL (same as External Attacker above).**

External Attacker+ is the audit-phase variant of External Attacker, conducted with Audit3a IPs whitelisted at WAF / IPS / AV for operational efficiency. After each finding is discovered, its applicability is assessed against both profiles: reachable under production protections, or whitelist-only.

Result: all 13 findings apply equally to both profiles. None is a signature-based attack pattern that WAF / IPS / AV would block in production — they are application-logic flaws (IDOR, broken authorization, business logic), configuration weaknesses, or hygiene issues. The protection layer slows automated discovery but does not protect against the underlying bugs; a recheck under active production protections confirmed every finding remains exploitable.

Insider with Remote Access - HIGH RISK

12 risks are relevant for this threat, of them**: 2 HIGH, 3 MEDIUM, 5 LOW, 2 INFORMATIONAL.**

This profile simulates an attacker who **already holds legitimate internal-network access** - a remote employee, an authorized contractor, or any actor that has previously obtained credentials by means outside the scope of this stage. For this engagement, the audit team's internal-network position was provisioned by the Client via VPN; the findings below describe what an attacker can do **from** that position, not how they would reach it.

The found risks of the **HIGH** level of criticality allow the Insider with Remote Access profile to do the following:

- **Sign in to 41 employee accounts** in the corporate directory using a short list of 8 predictable passwords built around the company name and the year. From one of those accounts, the team reached over 15 internal servers via remote desktop, found further passwords saved in a web browser on a regional administrator's workstation, and from there opened two business-critical systems: the platform used to manage every employee laptop, and the surveillance-and-gate-control system of a regional production site (21 live cameras and remote "open/close" gate buttons). The security monitoring team did not detect the password-guessing attack, and 14 days later, most of the captured passwords were still working (AD-R1).
- **Sign in to Microsoft 365 (general-staff scope)**, the corporate VPN, and remote desktop with nothing but a stolen password - no second factor is required. Any captured credential immediately yields mailbox access, internal-network access via the VPN, and remote-server access (AD-R2).

The found risks of the **MEDIUM** level of criticality allow the Insider with Remote Access profile to do the following:

- **Extract password hashes for high-privilege service accounts and crack them offline** by exploiting service principals that still encrypt their Kerberos tickets with RC4. A captured ticket can be brute-forced on commodity GPU hardware within hours rather than years - any successful crack yields the cleartext password and the privileges associated with that service account.
- **Bypass the integrity protections on directory traffic** because the domain controllers accept unsigned LDAP requests and unbound LDAP-over-TLS sessions. An attacker positioned on the internal network can man-in-the-middle authentication exchanges and steal or alter directory queries in transit.
- **Plant hidden code in the back-office that runs in the browser of any other administrator** who opens an affected page and silently captures their login session, allowing the attacker to act as that administrator. Requires a compromised back-office administrator account (WEB-R4).

The found risks of the **LOW** level of criticality allow the Insider with Remote Access profile to do the following:

- For employees who have been moved into the multi-factor-authentication programme but have not yet completed their device setup, **register the attacker's own phone as the second factor instead of the legitimate employee's.** From that point on, the attacker controls the multi-factor channel for that account (WEB-R6).
- **Guess passwords without any restriction** on the back-office administration sign-in page. There is no account lockout, no rate limit, no CAPTCHA, and no second factor. The back-office is reachable only from corporate IP ranges, so this finding is exploitable specifically from the Insider with Remote Access position (WEB-R8).
- **Keep using a compromised account indefinitely** because hundreds of accounts in the corporate directory have the "password never expires" flag set - once captured, they remain valid until a manual reset or until the account is decommissioned.
- **Plant arbitrary internal DNS records** because the internal DNS zone accepts unauthenticated dynamic updates - useful for redirecting internal services to attacker-controlled hosts or for forging server identities during a lateral-movement attack.
- **Reuse stale enabled accounts** that should have been disabled or removed during normal employee turnover but were not - these are easy targets for credential-guessing and rarely have an active owner to notice misuse.

The found **INFORMATIONAL** risks affecting this profile allow the Insider with Remote Access profile to do the following:

- **Self-elevate from a partial back-office role to full administrator** by editing the role definition or creating a privileged account, if the attacker already holds the "Users" or "Roles" permission (WEB-R7). The technical vulnerability is significant, but the surrounding controls (limited network access, account-approval workflow) keep the practical risk at an informational level in the current architecture.
- **Bypass certain endpoint protections** on the subset of corporate clients where Group Policy enforcement is inconsistent - useful for an attacker preparing further lateral movement or persistence.

2. Scope, limitations & threat model

A simulation of actions by a group of hackers with various commercial or personal motives was conducted during the testing. The primary objective of simulating these threats was to enhance the understanding of the system and gain logical access to the company's internal resources at the highest possible level.

Simulated threats:

№	Threats	Scope (attack vector)	Test mode (knowledge and access)

1	External Attacker	• External perimeter and sites:
		  ○ 192.0.2.0/24, 198.51.100.0/24
		  ○ https://app.festival-finance.example
		  ○ https://support.festival-finance.example
		  ○ https://partners.festival-finance.example
		  ○ https://voucher.festival-finance.example
			• Grey Box — operates from the public Internet using OSINT-derived intelligence
			• Where the application registration flow is open, a self-registered customer account of the kind that any prospective Festival Finance customer can obtain
			• Security systems (IPS/WAF/AV) fully enabled

2	External Attacker+	• Phishing • Mail infrastructure of festival-finance.example • Public-Internet-facing Microsoft 365 authentication endpoints	• All conditions from № 1 apply, with the addition that Audit3a IPs are **whitelisted** at security systems (IPS / WAF / AV) during this phase
3	Insider with Remote Access	• Internal-network resources are reachable via the corporate VPN (10.10.0.0/16) • Active Directory: festival-finance.example domain (domain controllers + member servers) • Microsoft 365 tenant (general-staff scope of festival-finance.example) • Back-office administration application: https://admin.festival-finance.example	• Has basic knowledge about the test objects - "Grey Box" mode. • Has a domain account of a regular user • Has one unprivileged account and two privileged in the WEB application BackOffice

Testing conditions were established before the testing:

Network Access

- For testing internal systems, the Client provides remote VPN access to all members of the Contractor’s team.
- The Client arranges network access for the Contractor’s team, identical to that available to a typical user within the office’s internal network.

Credentials

- For the back-office administration application, two test administrator accounts with deliberately limited permissions were provided to support privilege-escalation testing.
- For the social-engineering stage, the Contractor operates entirely from external attacker infrastructure — no credentials were provided in advance.

Whitelisting

- During the external perimeter testing phase (External Attacker+), the Contractor's IP addresses are added to the whitelist of protection systems (IPS / WAF / AV, etc.) to focus testing on application-layer findings rather than the protection layer. After completing the main testing phase, a recheck is performed with active protection systems to simulate a real attack scenario (External Attacker).
- The IP addresses and email addresses of auditors will not be added to the email system whitelist. However, auditors will attempt to bypass anti-phishing filters, and a test email account will be provided to facilitate phishing campaign debugging.

Physical Access

- Physical access and presence of the Contractor’s team are not envisaged.

Authorization and Client Approval

- All risky actions (DoS, active exploitation, social engineering) must be authorized by the Client and carried out within agreed-upon time windows to minimize impact.
- In the case of confirmation of high-level risks, the Client will be informed immediately.

Additional Information on Testing Targets

- No additional data is provided; testing is carried out based on empirical data gathered during the process.

Other

- In the event of a significant number of objects or potential issues being identified, the Contractor will apply a selective approach, focusing on the most critical targets and risks. In cases of limited time, priority will be given to high and medium-risk vulnerabilities. This approach allows effective use of resources within the agreed scope of work. Internal network testing will be conducted using a sample of homogeneous hosts – a total of 128 hosts in the sample.

Out of Scope:

- Any actions that could negatively affect the continuity of the Client’s business and/or cause service, system, network, or infrastructure outages.
- Any actions that could result in a leak of the Client’s confidential data.
- Any other systems and objects not specified within the scope of work.

Additional Client Requirements:

- Protection of employees' personal data.

3. Security assessment methods and brief results

The table below provides information about the stages of work, the tools employed, and concise results. Detailed reports on key activities can be found through the links provided.

№	Stages	Brief Report
1	External pentest (NET-EXT)	
1.A	Attack surface discovery	
1.1	Open ports scanning (TCP, UDP), identification of active services and software versions	*Detailed findings...*
1.2	Web services identification	*Detailed findings...*
1.B	Vulnerabilities discovery	
1.3	Network perimeter vulnerability scan	*Detailed findings...*
1.4	Web application vulnerability scan	All web pages identified in the agreed external scope were scanned by Burp Suite Professional and Acunetix. Initial findings were validated manually to discard false positives. Among the confirmed exposures, two HIGH-severity findings received detailed remediation guidance and are documented under WEB-R1 (Insecure Direct Object Reference in the Corporate Portal) and WEB-R5 (MinIO bucket listing exposing 4,612 documents). Several MEDIUM-severity findings (monitoring dashboard exposure, customer-PIN-reminder authorization gap, stored-XSS in back-office) were also confirmed at this stage.
1.5	Email infrastructure analysis (DMARC, SPF, DKIM, SMTP relay)	The authoritative DMARC, SPF, and DKIM records of the company's email domains were retrieved and validated against RFC 7489 using both automated validators (MXToolbox, EasyDMARC) and manual SMTP probing of the inbound mail-protection front-end. A duplicated v=DMARC1 tag in the DMARC TXT record was found to invalidate the published policy on receivers that strict-parse the record, allowing external delivery of From:-spoofed messages impersonating internal employees - documented under NET-EXT-R1.
1.6	Analysis of outdated versions of software services	*Detailed findings...*
1.7	Brute-force and dictionary attacks on external authentication interfaces	*Detailed findings...*
2	Web Application Testing - authenticated, per OWASP WSTG	
2.1	Information Gathering (authenticated vulnerability discovery)	*Detailed findings...*
2.2	Configuration and Deployment Management Testing	*Detailed findings...*
2.3	Identity Management Testing	*Detailed findings...*
2.4	Authentication Testing	*Detailed findings...*
2.5	Authorization Testing	Coverage of object-level authorization, role enforcement, privilege-escalation controls, and tenant isolation. Includes the headline WEB-R1 IDOR in the Corporate Portal and the WEB-R3 customer-PIN-reminder authorization gap.
2.6	Session Management Testing	*Detailed findings...*
2.7	Input Validation Testing	*Detailed findings...*
2.8	Testing for Error Handling	*Detailed findings...*
2.9	Testing for Weak Cryptography	*Detailed findings...*
2.10	Business Logic Testing	Coverage of workflow integrity, time-dependency issues, rate-limit bypasses, and abuse of legitimate-feature combinations. Includes the headline WEB-R5 MinIO bucket-listing finding on the partner portal.
2.11	Client-Side Testing	*Detailed findings...*

№	Stages	Brief Report
3	Social Engineering	
3.1	Reconnaissance of company employees (OSINT)	*Detailed findings...*
3.2	Identification of mail-system vulnerabilities; bypass of email protections	*Detailed findings...*
3.3	Verification of harvested employee emails	*Detailed findings...*
3.4	Preparation of phishing infrastructure	*Detailed findings...*
3.5	Creation of a phishing-campaign scenario	*Detailed findings...*
3.6	Execution of the phishing campaign	The campaign was executed in two stages: an initial focused mailing to 14 senior recipients, followed by a mass mailing to 187 additional employees. During the campaign 70 recipients clicked the link, 7 entered credentials, and 4 approved a multi-factor-authentication prompt, allowing the audit team to capture live Microsoft 365 session tokens. Festival Finance's Security Operations Centre contained the fastest compromised account in 14 minutes (account lockout and password reset). Detailed results are documented under SOCIAL-R1.
4	Active Directory Testing - assumed-breach internal scope	
4.1	Active Directory enumeration	*Detailed findings...*
4.2	Kerberos password spraying	A wordlist of 8 candidate passwords derived from the company name and the year was sprayed against the 2,847 enumerated accounts using kerbrute (TCP/88, internal Domain Controller). 44 accounts (1.4 %) authenticated successfully across the 8 rounds; the campaign stayed below the lockout threshold (6 attempts in 10 minutes) and was not detected by the SOC over a 14-day persistence check. This is the headline finding of the Active Directory assessment - documented as AD-R1.
4.3	Lateral movement and post-compromise validation	Using the captured credentials, the audit team connected via remote desktop to over 15 internal servers. On a regional administrator's workstation, browser-saved credentials yielded access to two business-critical systems: the enterprise endpoint-management platform (covering every corporate laptop) and the physical-security console of a regional production site (21 live cameras and remote gate controls). The post-compromise chain is the main concern under the Insider with Remote Access threat profile.
4.4	MFA-enforcement audit	Coverage of every external and internal authentication path against the documented MFA policy. Confirmed that MFA is not enforced on Microsoft 365 (general-staff scope), FortiClient VPN, or RDP via domain credentials - documented as AD-R2.
4.5	SPN review (Kerberoasting risk)	*Detailed findings...*
4.6	LDAP signing and channel-binding audit	*Detailed findings...*
4.7	DNS dynamic-update authorisation	*Detailed findings...*
4.8	Account-management hygiene review	*Detailed findings...*

4. Risk analysis results and recommendations

The table below contains an excerpt from the risk assessment. Detailed risk reports are provided as attachments, which can be found by the links in the first column.

ID	Vector	Vulnerable service	Risk Level	Risk description	Recommendation
WEB-R1	Public WEB Sites	support.festival-finance.example (Corporate Portal)	HIGH	The Corporate Portal exposes Insecure Direct Object References (IDOR) / Broken Object Level Authorization (BOLA). By iterating predictable identifiers in several endpoints, an authenticated client (or an attacker with a stolen session) can access other tenants' objects and personally identifiable information (PII) belonging to other corporate clients.	Enforce object-level authorization on every read/update/delete operation - verify that the ownerCompanyId / tenantId of the exact object matches the current user. Prefer unpredictable identifiers (UUIDv4 or server-issued opaque references) over sequential numeric IDs. Apply automated authorization tests in CI to detect IDOR regressions.
WEB-R5	Public WEB Sites	partners.festival-finance.example/storage/officecp-bucket/ (MinIO)	HIGH	The web application's file storage is implemented on a MinIO S3-compatible service with the bucket-listing feature enabled. Any authenticated user of the system can retrieve the complete list of every uploaded file and download arbitrary objects with no per-object authorization. The bucket holds 4,612 confidential documents, including national IDs, driver's licenses, military-registration documents, financial statements, and counterparty personal data.	Disable bucket-listing at the MinIO configuration level (no s3:ListBucket for end users). Enforce per-object authorization - each request to read or download a file must verify that the object belongs to the authenticated user's tenant / counterparty. Replace direct URLs with pre-signed URLs that have a short lifetime and are scoped to a single object. Add server-side logging and alerting on bulk-listing requests against the bucket.
AD-R1	Active Directory & Azure AD	Domain password policy for festival-finance.example; downstream: enterprise MDM, physical-security console	HIGH	The domain password policy does not block predictable passwords like the company name + current year. 41 of 2,847 active domain accounts (1.4 %) were compromised in 73 seconds via a wordlist of 8 candidates, including a regional system administrator. Browser-saved credentials on that account further opened the corporate MDM platform (Domain-Admin-equivalent privilege) and a regional facility's physical-security console (18 cameras + remote gate controls). The SOC did not detect the spray; 14 days later most captured passwords still worked.	Deploy a banned-password list (company name, year, common patterns) via Azure AD Password Protection or equivalent. Enforce phishing-resistant MFA on every authentication path (VPN, RDP, Microsoft 365). Add Kerberos audit logging and honeytoken accounts to detect spray attempts. Rotate browser-saved credentials on shared and admin systems; require enterprise password managers.
AD-R2	Active Directory & Azure AD	Microsoft 365 tenant (general-staff scope, accounts outside the pilot-group Conditional Access policy); FortiClient SSL VPN gateway; RDP across servers reachable from the VPN	HIGH	Multi-factor authentication is not enforced on Microsoft 365 (general staff), the corporate VPN, or RDP via domain credentials. Any captured password — by any means — grants immediate access to mail, VPN, and remote desktop. This single defect converts otherwise self-contained findings into complete attack chains: password spray (AD-R1) and phishing (SOCIAL-R1) both yielded immediate access without further verification.	Enforce phishing-resistant MFA (FIDO2 / number-matching) on every authentication service that accepts domain credentials — Microsoft 365, VPN, and RDP. Block legacy auth protocols (POP3, IMAP, SMTP AUTH) that bypass MFA. Set an MFA-enrollment deadline; lock accounts that miss it. Audit and remove any auth path that bypasses MFA.
WEB-R2	Internal WEB Sites	admin.festival-finance.example/telescope	MEDIUM	The back-office application's Laravel Telescope monitoring dashboard is exposed to the public Internet with no authentication, authorization, or network restriction. Anyone can browse failed requests, exceptions, request bodies, response payloads, internal hostnames, and unredacted session tokens issued by the application.	Telescope must not be publicly reachable. Enforce a strict in-app authorization gate (viewTelescope) limited to designated administrator and developer accounts. Apply a compensating infrastructure control by restricting /telescope to VPN / corporate IP ranges at the reverse proxy or WAF. Configure Telescope to redact secrets, session tokens, and sensitive request parameters in both requests and responses.
WEB-R3	Public WEB Sites	app.festival-finance.example	MEDIUM	An authenticated customer can abuse the "PIN reminder" function to have the card PIN of another customer's payment card	Implement strict server-side ownership validation: when processing PIN-reminder requests, the backend must verify that the card identified

ID	Vector	Vulnerable service	Risk Level	Risk description	Recommendation
				delivered to the attacker's own email address. The backend does not verify that the card identified by the supplied cardHash belongs to the requesting user.	by the supplied cardHash is linked to the currently authenticated user, and reject the request otherwise. Add automated authorization tests covering this endpoint.
WEB-R4	Internal WEB Sites	admin.festival-finance.example	MEDIUM	Multiple stored XSS vectors were identified in the back-office admin application. Authenticated users can inject HTML and JavaScript via the *Site-Specific translations*, *Username*, and *Image upload* (filename, uploader) fields - the payload is stored server-side and executes in the browsers of other administrators who view the affected pages. Session theft, page-content tampering, and CSRF-token extraction were demonstrated.	Enforce server-side input sanitization on every text field (escape HTML/JS metacharacters; reject control bytes). Validate input on both client and server. Deploy a strict Content Security Policy (CSP) that blocks inline scripts and disallows untrusted script origins. Add hardening response headers (X-Content-Type-Options, Referrer-Policy). Audit every form field for output-context-aware escaping. Train developers on secure coding for output contexts.
NET-EXT-R1	External Network	DNS / SMTP infrastructure of festival-finance.example (and festival-finance.net)	MEDIUM	The DMARC record for festival-finance.example is published with a duplicated v=DMARC1 tag, making the record syntactically invalid per RFC 7489. Many receiving mail servers discard malformed DMARC and therefore do not enforce the published p=reject policy. Combined with permissive SMTP behavior on the mail-protection front-end, this allows an external attacker to deliver email to an internal recipient with the From: header impersonating any internal employee.	Publish a syntactically correct DMARC record - a single v=DMARC1 tag at the start, followed by the intended policy. Validate via external scanners (MXToolbox, EasyDMARC). Restrict the SMTP relay so unauthenticated external senders cannot inject mail addressed to internal recipients. Enable / tighten DKIM signing on every outbound stream. Configure the receiving gateway to reject (not quarantine) messages that fail DMARC alignment. Monitor DMARC aggregate reports to detect spoofing attempts.
SOCIAL-R1	Phishing	Festival Finance Microsoft 365 tenant; employee mailboxes	MEDIUM	A two-stage phishing campaign was run against 187 Festival Finance employees using a credential-harvesting MITM proxy (Evilginx2) and a look-alike Microsoft 365 login page. 64 employees clicked, 7 entered credentials, 4 approved MFA — defeating multi-factor authentication and yielding 4 live Microsoft 365 session tokens. The audit team accessed Outlook, Teams, SharePoint, and Microsoft Entra as the compromised users and extracted 7,400+ employee email addresses from the tenant. The SOC contained the fastest case in 14 minutes.	Enforce phishing-resistant MFA (FIDO2 / number-matching) for Microsoft 365 — app-notification approval alone is bypassable by MITM. Apply Conditional Access policies (location, device compliance, sign-in risk). Tighten email-gateway filtering on look-alike and freshly registered domains. Run continuous user-awareness training with realistic phishing simulations. Fix the underlying email-spoofing weakness (NET-EXT-R1).
WEB-R9	Public WEB Sites	Customer-facing and partner-facing applications	MEDIUM	*Risk description truncated…*	*Recommendation truncated…*
AD-R3	Active Directory & Azure AD	Service accounts with Service Principal Names	MEDIUM	*Risk description truncated…*	*Recommendation truncated…*
AD-R4	Active Directory & Azure AD	Domain controllers	MEDIUM	*Risk description truncated…*	*Recommendation truncated…*
WEB-R6	Public WEB Sites	Microsoft 365 / Entra ID account-enrollment workflow for accounts in the MFA-required Conditional Access policy (pilot group); vpn.festival-finance.example/saml (downstream lateral-movement target)	LOW	Microsoft 365 accounts in the MFA pilot group that have not yet enrolled their second-factor device accept enrollment of an arbitrary Authenticator app by anyone who can present the username and password. Using a credential captured during the phishing campaign (SOCIAL-R1), the audit team registered their own phone as the second factor on a pilot-group account — taking persistent control of that account's MFA channel. Follow-on lateral movement was blocked by the SOC and account permissions, capping the impact at LOW.	Require MFA enrollment within 24 hours of account provisioning, from a trusted device or location. Alert the account owner on every new MFA device via a separate channel (SMS / corporate phone). Use temporary access passes for first-time enrollment and require administrator confirmation for sensitive accounts.
WEB-R8	Internal WEB Sites	admin.festival-finance.example	LOW	The back-office sign-in endpoint does not implement an account-lockout mechanism: there is no limit on the number of consecutive failed authentication attempts, no delay between	Configure account lockout: after 5 failed sign-in attempts, lock the account for 10 minutes (mirror the existing policy on the customer-facing application). Add per-account and per-IP rate limiting on the

ID	Vector	Vulnerable service	Risk Level	Risk description	Recommendation
				attempts, no CAPTCHA, and no second factor. Successful authentication after an unlimited series of failures is possible. The customer-facing application (app.festival-finance.example) already enforces a 5-failures-in-10-minutes lockout - the same control is missing here.	sign-in API. Introduce CAPTCHA after 3 failed attempts. Require multi-factor authentication for back-office administrators given the privilege level of those accounts. Log and alert on bursts of failed sign-ins.
WEB-R10	Public WEB Sites	Customer-facing application	LOW	*Risk description truncated…*	*Recommendation truncated…*
WEB-R11	Public WEB Sites	Customer / partner-facing web applications	LOW	*Risk description truncated…*	*Recommendation truncated…*
NET-EXT-R2	External Network	External perimeter	LOW	*Risk description truncated…*	*Recommendation truncated…*
AD-R5	Active Directory & Azure AD	Active Directory user accounts	LOW	*Risk description truncated…*	*Recommendation truncated…*
AD-R6	Active Directory & Azure AD	Active Directory account-management process	LOW	*Risk description truncated…*	*Recommendation truncated…*
AD-R7	Active Directory & Azure AD	Internal DNS infrastructure	LOW	*Risk description truncated…*	*Recommendation truncated…*
WEB-R7	Internal WEB Sites	admin.festival-finance.example	INFO	Users granted the Users or Roles permission in the back-office can create arbitrary new accounts with any privileges, delete existing administrators, and modify any role - including assigning themselves the full administrator role. The application does not enforce a tiered approval model on user / role administration.	Restrict the Users and Roles permissions to a small set of strictly governed accounts. Implement a tiered model: a user with role-management rights must not be able to grant themselves higher privileges than they currently hold. Apply four-eyes / dual-approval for sensitive permission grants. Audit log every change to user permissions and roles, and alert on self-elevation events.
WEB-R12	Public WEB Sites	Customer-facing applications and back-office	INFO	*Risk description truncated…*	*Recommendation truncated…*
WEB-R13	Public WEB Sites	Customer-facing application client bundle	INFO	*Risk description truncated…*	*Recommendation truncated…*
NET-EXT-R3	External Network	External perimeter	INFO	*Risk description truncated…*	*Recommendation truncated…*
AD-R8	Active Directory & Azure AD	Group Policy infrastructure	INFO	*Risk description truncated…*	*Recommendation truncated…*

5. Annex A – Risk Analysis

The purpose of our work is not only to identify and describe vulnerabilities but also to assess the actual level of risk they can create for your business. We consider real threats in the context of the technological environment and evaluate potential impacts that could arise from exploiting vulnerabilities.

Risk assessment is performed using the 'expert assessment' method, and the risk levels are determined by the project team. If necessary, the approach to risk analysis can be adapted according to the risk analysis methodology used in your company.

We use the following criteria to determine the levels of risk:

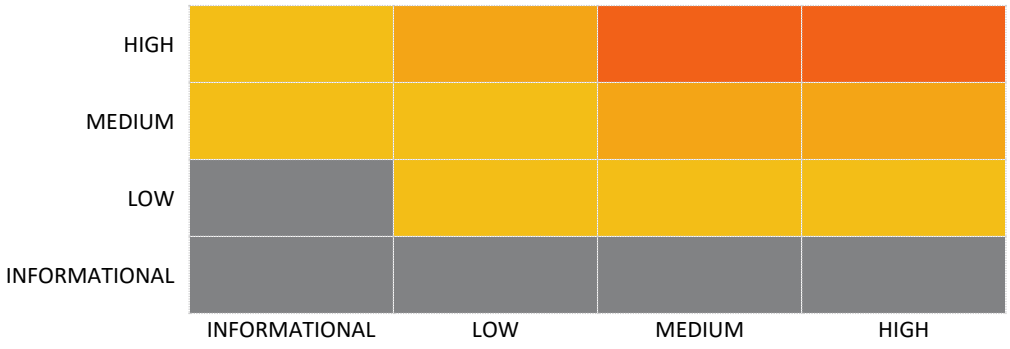
Risk level	Description
HIGH	This level signifies that potential adversaries with a high likelihood of success could attain significant control over the system or its critical components. They might gain access to highly confidential information, manipulate critical data, initiate a sustained denial of service, or severely damage the Company's reputation. The associated risk has the potential to lead to substantial losses.
MEDIUM	At this level, attackers with a substantial chance of success could obtain partial access to confidential information, compromise the integrity of specific system components, induce short-term denial of service incidents, or impact the reputation of a subset of the Company's customers. This category also includes recurring low-level incidents and critical vulnerabilities with a low likelihood of exploitation.
LOW	The LOW-risk level indicates that attackers might acquire limited information about the system, initiate a minor denial of service events targeting specific elements, or cause other relatively insignificant impacts.
INFORMATIONAL	This level is designated for situations where no discernible risk is posed to the affected systems. The identified issue aligns with established best practices but does not present a direct threat.
INSIGNIFICANT	This classification is reserved for cases where the risk of impact is minimal, and its presence could be disregarded or deemed inapplicable under the current circumstances.

Also, there were vulnerability, threats, and impact levels defined: **HIGH**, **MEDIUM**, **LOW**, **INFORMATIONAL**, **NOT APPLICABLE**

Risk assessment methodology defines the risk level by formula:

$$Risk = Fx(Fx(Vulnerability, Threat), Impact)$$

Where Fx can be calculated from the matrix:



6. Annex B – Detailed Risk Descriptions

The findings below combine into two end-to-end attack chains that account for the bulk of the residual risk in this engagement.

WEB-R1 – IDOR / Broken Object Level Authorization in Corporate Portal

RISK BRIEF

ID	WEB-R1
RISK LEVEL	**HIGH**
SUMMARY	The Corporate Portal exposes Insecure Direct Object References (IDOR) / Broken Object Level Authorization (BOLA). By iterating predictable identifiers in several endpoints, an authenticated client (or an attacker with a stolen session) can access other tenants' objects and personally identifiable information (PII) belonging to other corporate clients.
VULNERABLE SERVICES	support.festival-finance.example (Corporate Portal)
RECOMMENDATIONS	Enforce object-level authorization on every read/update/delete operation - verify that the ownerCompanyId / tenantId of the exact object matches the current user. Prefer unpredictable identifiers (UUIDv4 or server-issued opaque references) over sequential numeric IDs. Apply automated authorization tests in CI to detect IDOR regressions.

RISK ASSESSMENT

	SEVERITY	DESCRIPTION	
VULNERABILITY	**HIGH**	Classic IDOR / BOLA: changing the corpClientId / fileId query parameters returns records owned by other corporate tenants.	
THREAT	**HIGH**	Threat Actor Profiles	External Attacker
		Attack scenario	The attacker enumerates sequential identifiers via a simple script and harvests every record the API returns.
		Conditions	The vulnerable functionality is reachable to any registered corporate user; no special privileges are required.
IMPACT	**MEDIUM**	Actual Impact	Unauthorized disclosure of PII (email addresses, names), card identifiers / tokens, and operational metadata of unrelated corporate clients.
		Potential Impact	Enables targeted phishing, account mapping, and potential fraud; may trigger regulatory exposure depending on jurisdiction (GDPR, PCI DSS) and data classification.

TECHNICAL DETAILS & PROOF OF CONCEPT

A test corporate account was created with "corpClientId=10042". By iterating predictable numeric identifiers on three endpoints, the audit team confirmed unauthorized access to records belonging to other tenants:

1. "/PortalServices/v1/user/corpClientInfo?corpClientId=<N>" - returns valid IDs, names, and client types of arbitrary corporate clients (Figure 1).

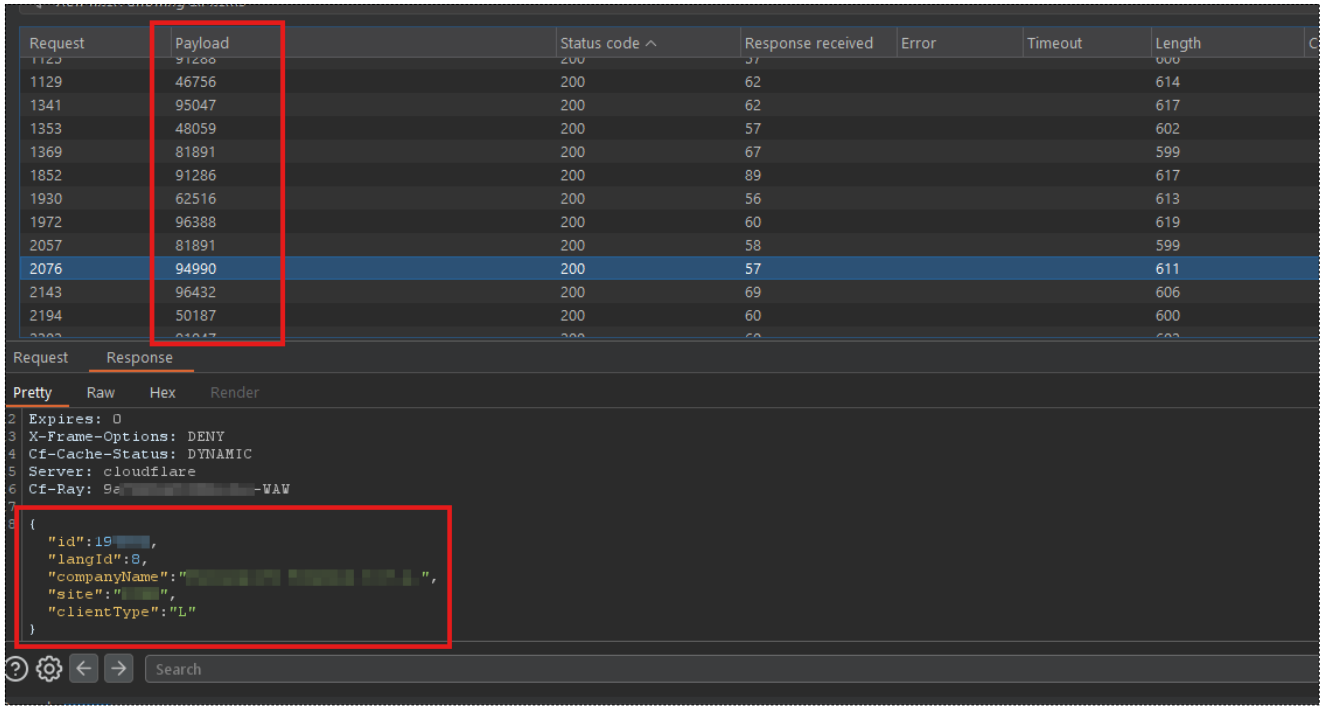


Figure 1 – Enumeration of "corpClientId" reveals valid IDs, names, and client types of other tenants.

2. "/PortalServices/v1/client/file?fileId=<N>" - returns file metadata (owning client ID, upload time, program ID) for any file in the system (Figure 2-3).

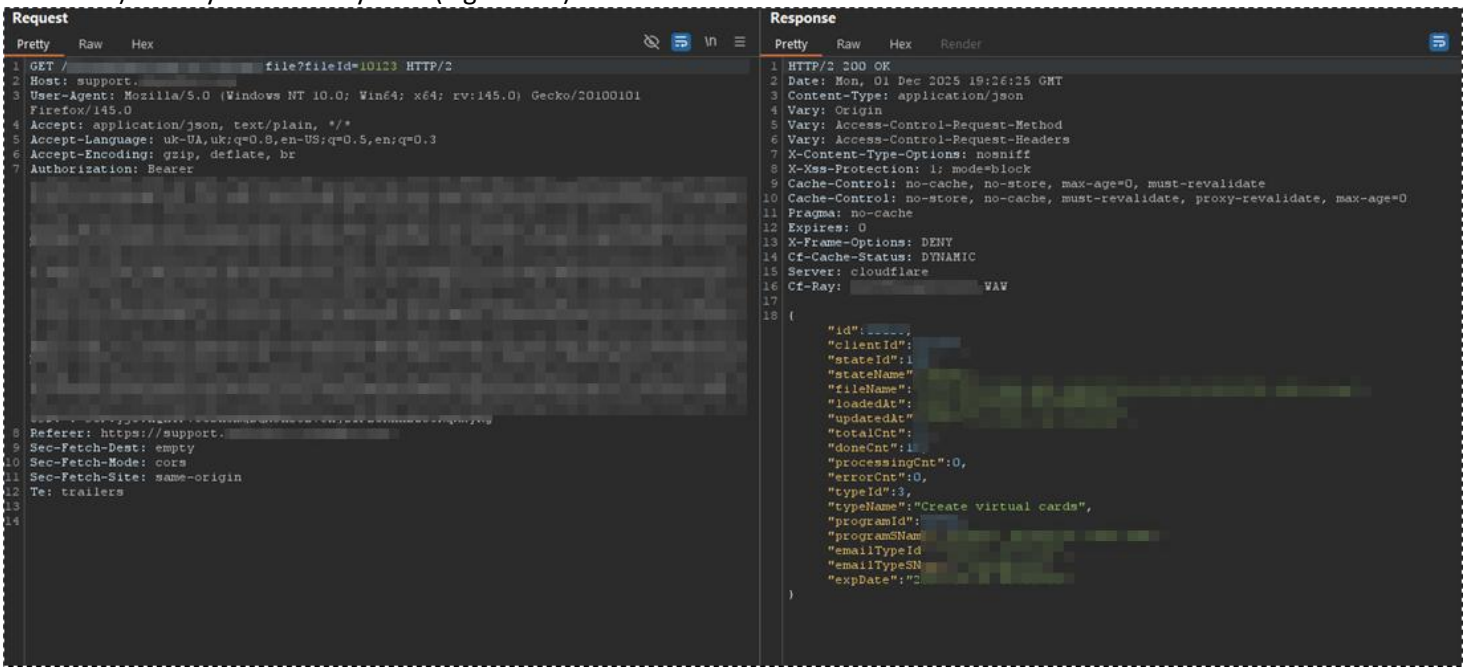


Figure 2 – File metadata leakage via "fileId" - "loadedAt" timestamp confirms data is unrelated to the testing account.

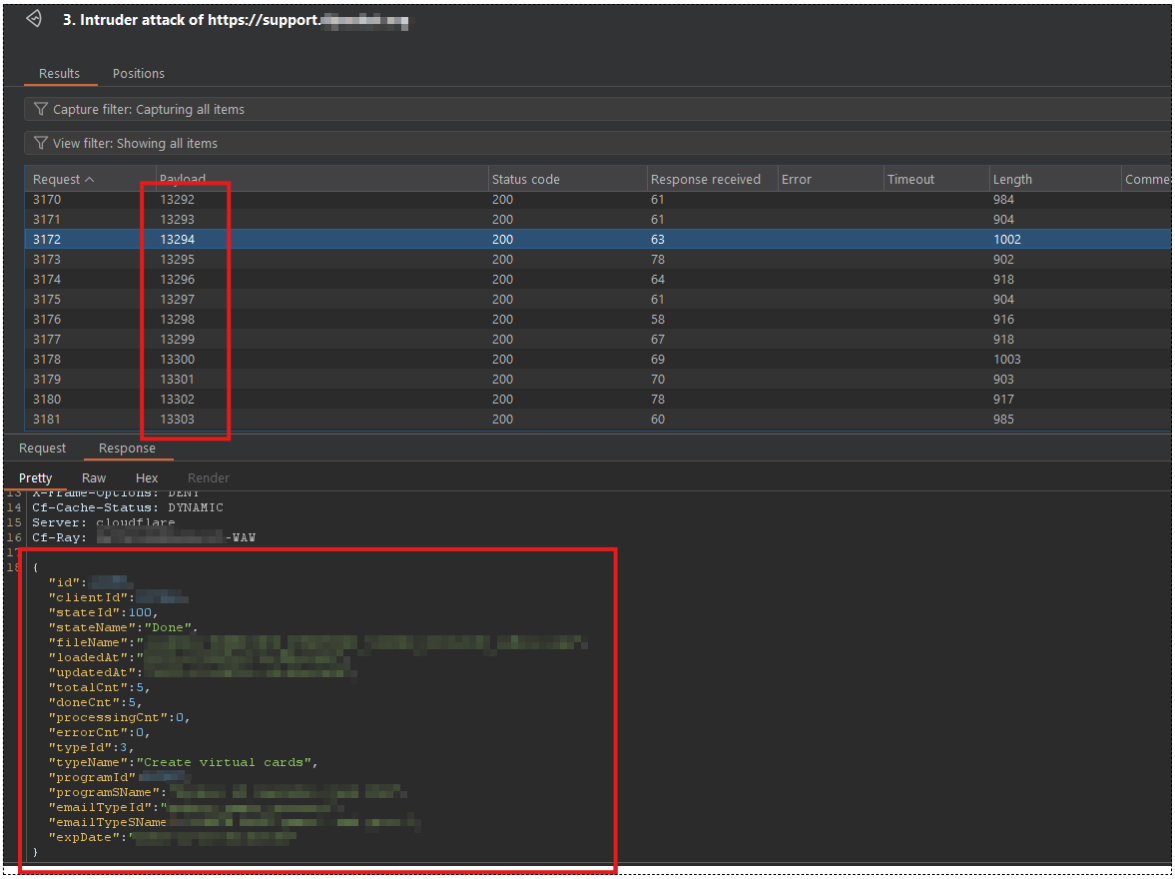


Figure 3 – Same endpoint returns full record set for any chosen "fileId".

3. "/PortalServices/v1/client/fileEntry?fileId=<N>" - returns the content of arbitrary uploaded files, including PII (email, first/last name, city, phone number) (Figure 4-5).

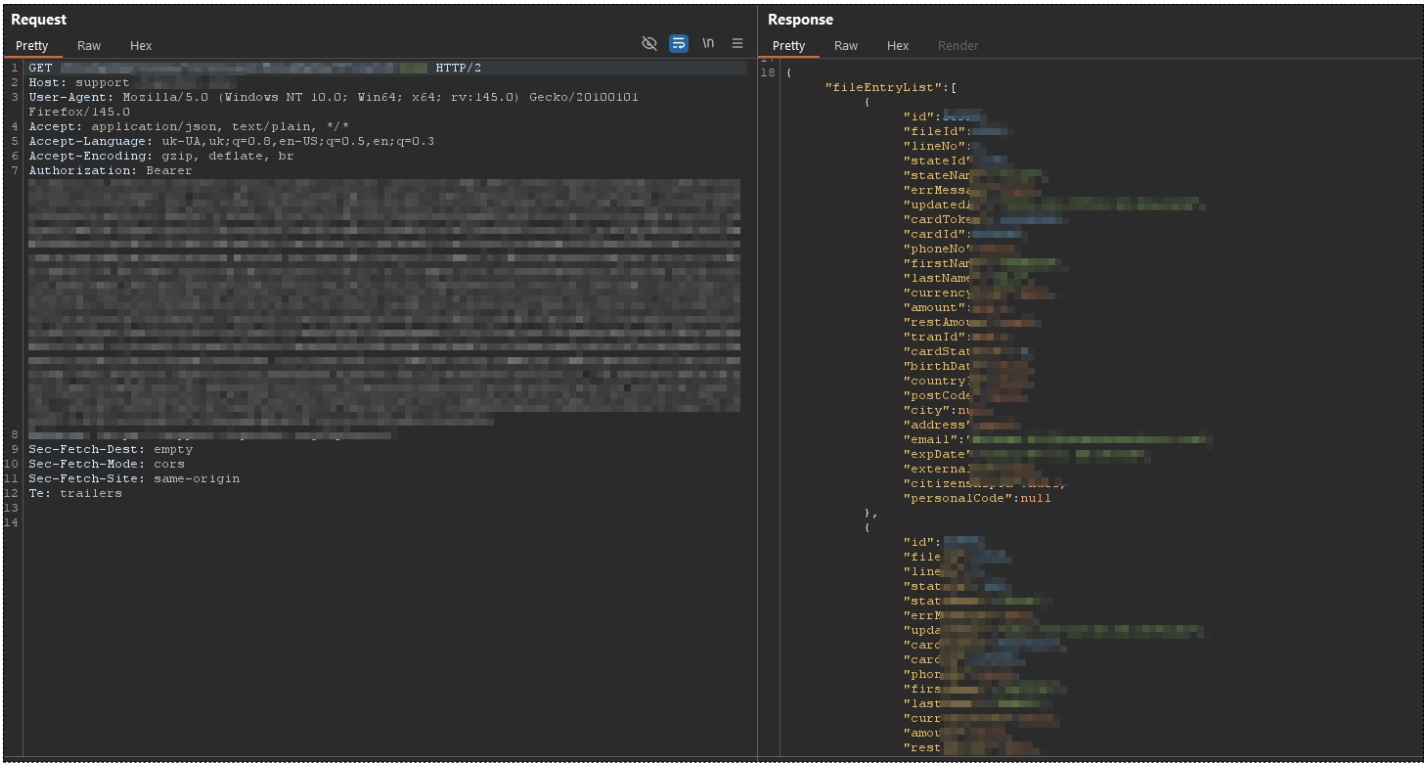


Figure 4 – Full file content disclosure including PII (email, name, city, phone).

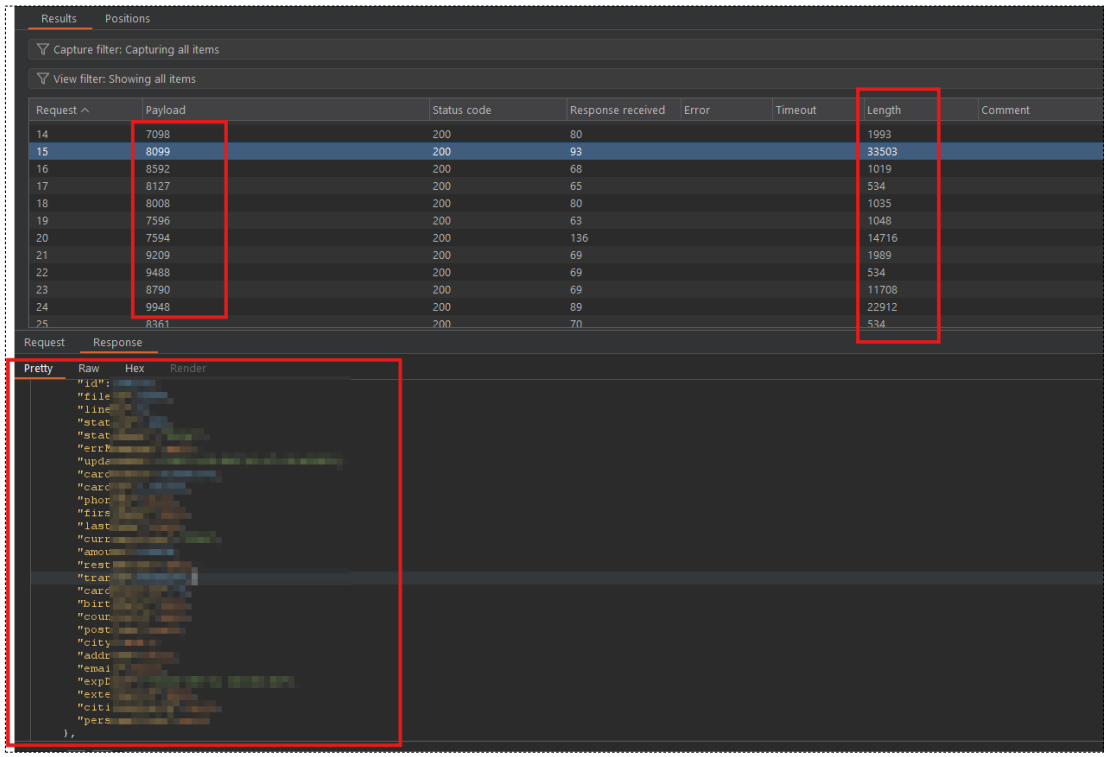


Figure 5 – Variable response length per "fileId" confirms different real customer datasets are exposed.

Identifiers follow a simple, monotonically increasing sequence, which makes mass enumeration trivial. The session belonging to the test account was used for all requests - no privilege escalation was required.

WEB-R5 – Public S3-Compatible Object Storage Lists 4,612 Confidential Documents

RISK BRIEF

ID	WEB-R5
RISK LEVEL	**HIGH**
SUMMARY	The web application's file storage is implemented on a MinIO S3-compatible service with the bucket-listing feature enabled. Any authenticated user of the system can retrieve the complete list of every uploaded file and download arbitrary objects with no per-object authorization. The bucket holds 4,612 confidential documents including national IDs, driver's licenses, military-registration documents, financial statements, and counterparty personal data.
VULNERABLE SERVICES	partners.festival-finance.example/storage/officecp-bucket/ (MinIO)
RECOMMENDATIONS	Disable bucket-listing at the MinIO configuration level (no s3:ListBucket for end users). Enforce per-object authorization - each request to read or download a file must verify that the object belongs to the authenticated user's tenant / counterparty. Replace direct URLs with pre-signed URLs that have a short lifetime and are scoped to a single object. Add server-side logging and alerting on bulk-listing requests against the bucket.

RISK ASSESSMENT

	SEVERITY	DESCRIPTION	
VULNERABILITY	**HIGH**	Bucket-listing is enabled on the MinIO endpoint; the standard S3 ListBucket API returns the full object inventory. No per-object authorization is performed - any authenticated user can download any file.	
THREAT	**HIGH**	Threat Actor Profiles	External Attacker
		Attack scenario	The attacker self-registers a regular customer account on the partner portal (the registration flow is open to anyone, completes in under five minutes, and requires no out-of-band verification), locates the storage path from any API response that returns a document URL, issues the standard ListBucket request, and downloads every object listed.
		Conditions	All conditions for exploitation are already present. Only a valid account in the system (obtainable via the public self-registration flow) and the storage path (/storage/officecp-bucket/, disclosed by the API in every document-related response) are required.
IMPACT	**HIGH**	Actual Impact	The audit team retrieved the complete index of 4,612 documents belonging to all users of the system. The collection includes personal documents (national IDs, driver's licenses, taxpayer numbers, military-registration documents), corporate financial statements (annual and quarterly reports, VAT and tax filings), corporate registry documents (charters, licences, registry extracts, auditor

[redacted]		reports), and vehicle titles. The full filename inventory was captured for evidence.
	Potential Impact	A breach of this size would constitute a major personal-data incident: violations of national data-protection law and GDPR; reputational and regulatory exposure; financial claims from affected individuals; downstream fraud and identity-theft risk using the harvested government-issued IDs.

TECHNICAL DETAILS & PROOF OF CONCEPT

The application exposes a MinIO-backed object store at the path "/storage/officecp-bucket/" reachable from the same hostname as the application API. Every API response that references a document includes the full storage URL, for example:

```
    "files": [
  {
    "fileID": 4729,
    "intFileName": "driver_license_scan",
    "url": "partners.festival-finance.example/storage/officecp-bucket/1768261643984-
drivers_136_driverLicense.png"
  }
]
```

The audit team observed that the path is reachable directly. Issuing the standard S3 "GET /storage/officecp-bucket/?list-type=2" returned the complete bucket inventory in an XML response - 4,612 keys spanning every customer's uploaded files. Each entry includes the object key, size, and last-modified timestamp.

Sampling 30 objects at random confirmed that no per-object authorization is applied: the audit-team account (a regular customer) was able to download national-ID scans, financial statements, vehicle titles, and registry extracts belonging to other counterparties.

Figure 1 – Sample national-ID document (passport) downloaded from the public bucket without authorization.

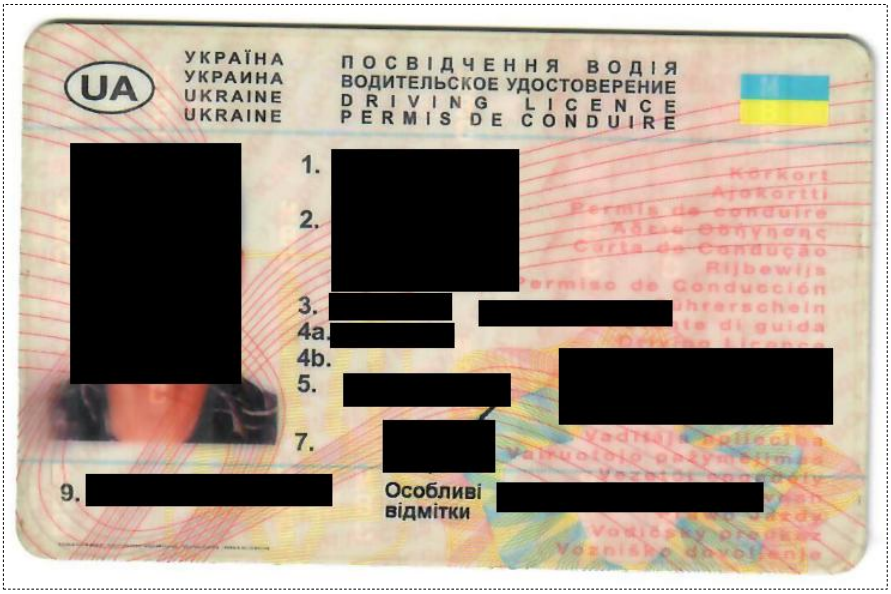


Figure 2 – Sample driver's license downloaded from the public bucket without authorization.

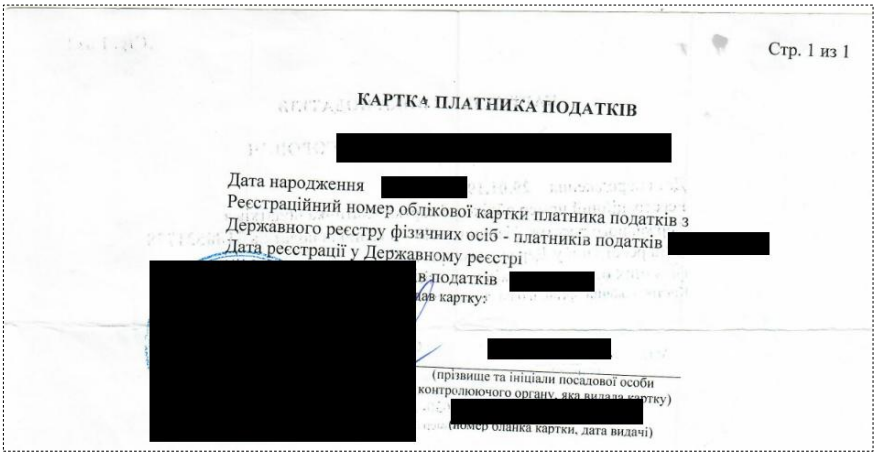


Figure 3 – Sample taxpayer-ID card downloaded from the public bucket without authorization.

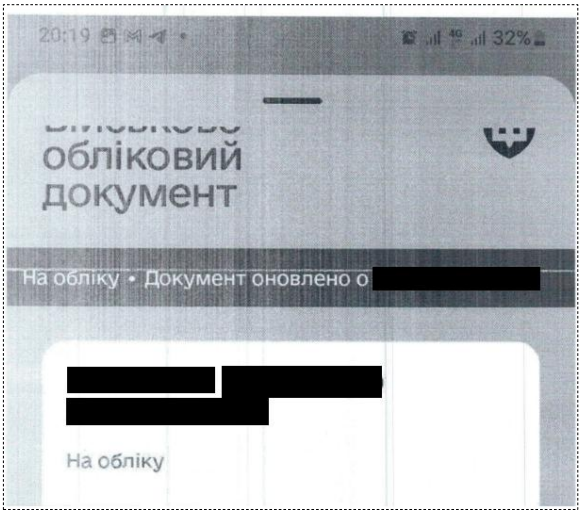


Figure 4 – Sample military-registration document downloaded from the public bucket without authorization.

Figure 5 – Sample vehicle-registration card downloaded from the public bucket without authorization.

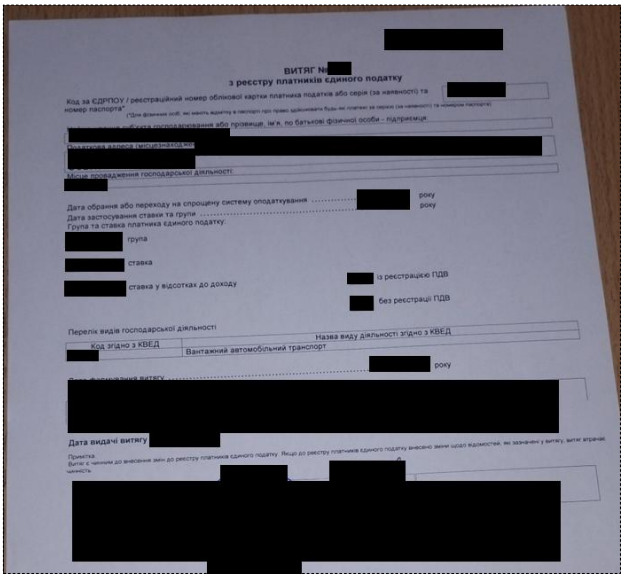
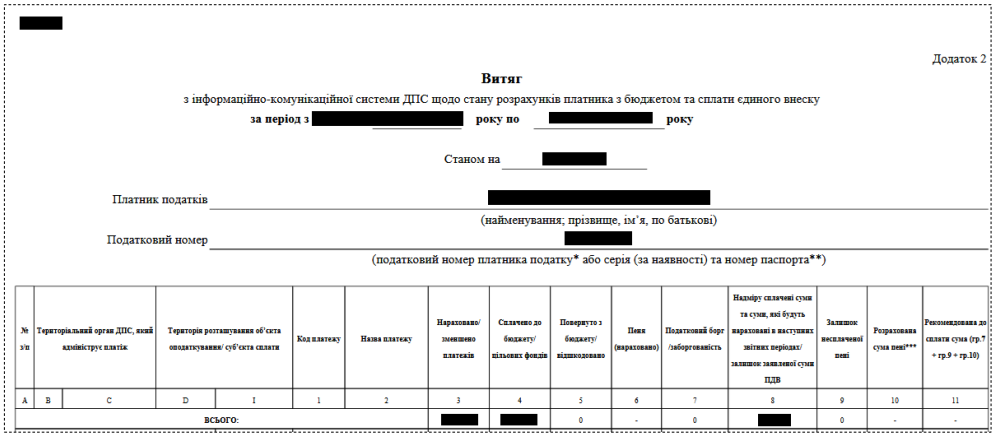


Figure 6 – Sample single-tax registry extract downloaded from the public bucket without authorization.



Додаток 2

Витяг

з інформаційно-комунікаційної системи ДПС щодо стану розрахунків платника з бюджетом та сплати єдиного внеску

за період з [redacted] **року по** [redacted] **року**

Станом на [redacted]

Платник податків [redacted]

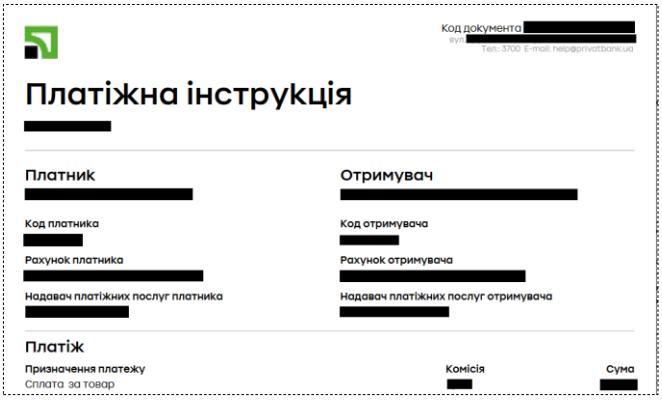
(найменування; прізвище, ім'я, по батькові)

Податковий номер [redacted]

(податковий номер платника податку* або серія (за наявності) та номер паспорта**)

№ з/п	Територіальний орган ДПС, який адмініструє платіж	Територія розташування об'єкта оподаткування/ суб'єкта сплати		Код платежу	Назва платежу	Нараховано/ зменшено платежів	Сплачено до бюджету/ цільових фондів	Повернуто з бюджету/ відшкодовано	Пеня (нараховано)	Податковий борг /заборгованість	Надміру сплачені суми та суми, які будуть нараховані в наступних звітних періодах/ залишок заявленої суми ПДВ	Залишок несплаченої пені	Розрахована сума пені***	Рекомендована до сплати сума (гр.7 + гр.9 + гр.10)
A	B	C	D	I	1	2	3	4	5	6	7	8	9	10	11
ВСЬОГО:							[redacted]	[redacted]	0	-	0	[redacted]	0	-	-

Figure 7 – Sample tax-service statement of account downloaded from the public bucket without authorization.



Код документа [redacted]

Платіжна інструкція

[redacted]

Платник [redacted]

Код платника [redacted]

Рахунок платника [redacted]

Надавач платіжних послуг платника [redacted]

Отримувач [redacted]

Код отримувача [redacted]

Рахунок отримувача [redacted]

Надавач платіжних послуг отримувача [redacted]

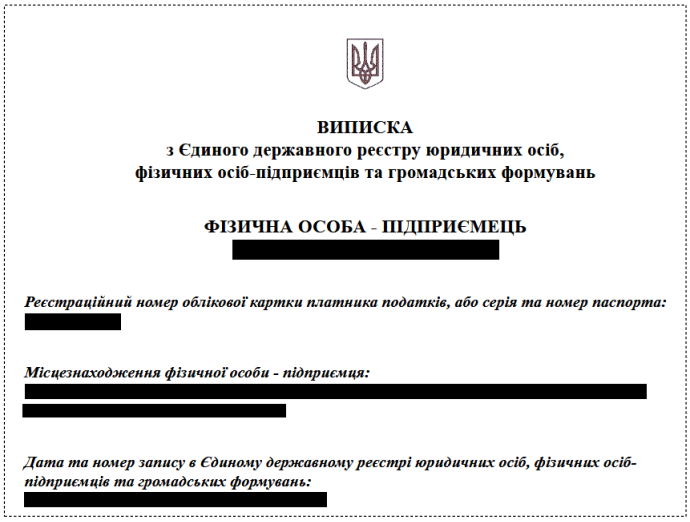
Платіж

Призначення платежу
Сплата за товар

Комісія [redacted]

Сума [redacted]

Figure 8 – Sample payment instruction downloaded from the public bucket without authorization.



ВИПИСКА
з Єдиного державного реєстру юридичних осіб,
фізичних осіб-підприємців та громадських формувань

ФІЗИЧНА ОСОБА - ПІДПРИЄМЕЦЬ [redacted]

Реєстраційний номер облікової картки платника податків, або серія та номер паспорта: [redacted]

Місцезнаходження фізичної особи - підприємця: [redacted]

Дата та номер запису в Єдиному державному реєстрі юридичних осіб, фізичних осіб-підприємців та громадських формувань: [redacted]

Figure 9 – Sample legal-entity registry extract downloaded from the public bucket without authorization.

AD-R1 – OSINT-Driven Password Spray - 41 Domain Accounts Compromised, Chained to MDM + Physical Security

RISK BRIEF

ID	AD-R1
RISK LEVEL	**HIGH**
SUMMARY	The domain password policy does not block predictable passwords like the company name + current year. 41 of 2,847 active domain accounts (1.4 %) were compromised in 73 seconds via a wordlist of 8 candidates, including a regional system administrator. Browser-saved credentials on that account further opened the corporate MDM platform (Domain-Admin-equivalent privilege) and a regional facility's physical-security console (18 cameras + remote gate controls). The SOC did not detect the spray; 14 days later most captured passwords still worked.
VULNERABLE SERVICES	Domain password policy for festival-finance.example; downstream: enterprise MDM, physical-security console
RECOMMENDATIONS	Deploy a banned-password list (company name, year, common patterns) via Azure AD Password Protection or equivalent. Enforce phishing-resistant MFA on every authentication path (VPN, RDP, Microsoft 365). Add Kerberos audit logging and honeytoken accounts to detect spray attempts. Rotate browser-saved credentials on shared and admin systems; require enterprise password managers.

RISK ASSESSMENT

	SEVERITY	DESCRIPTION	
VULNERABILITY	**HIGH**	Domain password policy (min 10 chars + complexity) does not prohibit predictable patterns based on the company name. All eight tested candidates (FestivalFinance2026!, FestivalFinance123, FestivalFinance@2026, etc.) technically satisfy complexity rules but are trivially guessable from a wordlist of 10-20 candidates. No banned-password list.	
THREAT	**HIGH**	Threat Actor Profiles	Insider with Remote Access
		Attack scenario	The attacker harvests potential usernames via OSINT (LinkedIn, corporate site, public email format) and a candidate-password list derived from public information about the company. The spray runs via Kerberos (port 88) from inside the network. The attacker tries each password against every account, staying under the lockout threshold of 6 attempts in 10 minutes.
		Conditions	(1) Users select predictable, company-themed passwords. (2) The domain policy allows them. (3) The attacker has access to an authentication service (Kerberos, LDAP, Microsoft 365, VPN). (4) No MFA. All four conditions are present in this environment.
IMPACT	**HIGH**	Actual Impact	44 compromised accounts granted RDP access to 15+ servers (terminal, file, SQL). Through the regional sysadmin account, the audit team reached the SERVREGIONAL server. In the sysadmin's Chrome saved-passwords store the team

		recovered credentials to: (a) the enterprise MDM platform - the enterprise MDM that manages every endpoint in the company; (b) the physical-security console of a regional facility (18 CCTV cameras + remote control of perimeter gates). The absence of MFA meant every recovered password was immediately usable - no additional verification stood between the attacker and full access.
	Potential Impact	Authenticated access to the enterprise MDM platform is equivalent to Domain Admin - the MDM can execute arbitrary commands as SYSTEM on every managed endpoint. Access to the physical-security console enables remote gate opening and live-camera surveillance, creating a physical-intrusion vector against industrial sites. An APT-class adversary can use the compromised accounts as a persistent presence and incrementally escalate via lateral movement; the absence of detection (14 days, no SOC alert) suggests months of dwell time would be feasible.

TECHNICAL DETAILS & PROOF OF CONCEPT

Step 1 - Wordlist construction. Eight candidate passwords were derived purely from the company name and the year. All satisfy the domain complexity policy:

```
FestivalFinance2026
FestivalFinance2026!
FestivalFinance2025
FestivalFinance2025!
FestivalFinance123
FestivalFinance1234
FestivalFinance@2026
Spring2026!
```

Step 2 - Account enumeration. A list of 2,847 enabled domain accounts was harvested via "ldapdomaindump". In a real-world scenario the list would be reduced via OSINT (LinkedIn, corporate site, email-format harvesting) - smaller in absolute count but accurate enough for a successful spray.

Step 3 - Kerberos password spray (8 rounds). Each round tries one password against all 2,847 accounts, with intervals exceeding the 10-minute lockout duration:

```
kerbrute passwordspray --dc 10.3.1.45 -d festival-finance.example users.txt 'FestivalFinance123'
```

Outcome: 41 accounts captured (1.4%) across the 8 rounds. The largest single round - "FestivalFinance123" - returned 16 valid accounts (Figure 1).

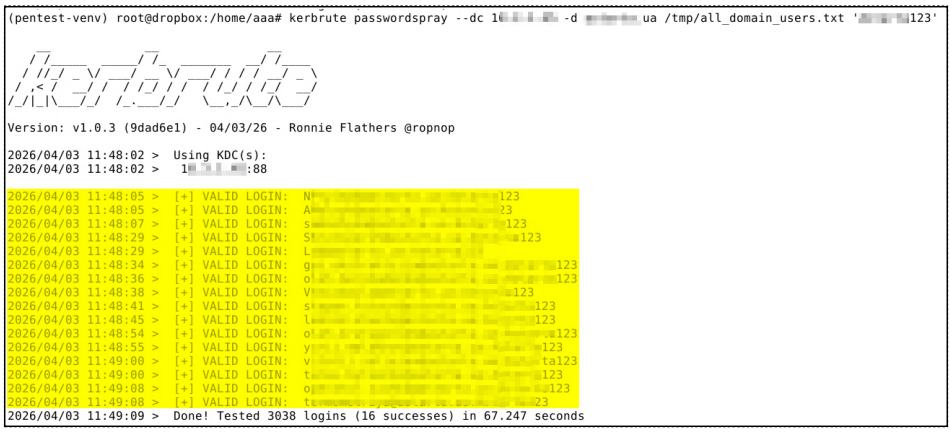


Figure 1 – Kerberos password-spray output for the largest round ("FestivalFinance123", 16 hits).

Step 4 - Lateral movement via RDP. The captured account "<regional_sysadmin>" (password "FestivalFinance123") granted RDP access to "SERVREGIONAL" (192.0.2.120). The audit team logged in to the server (Figure 2).

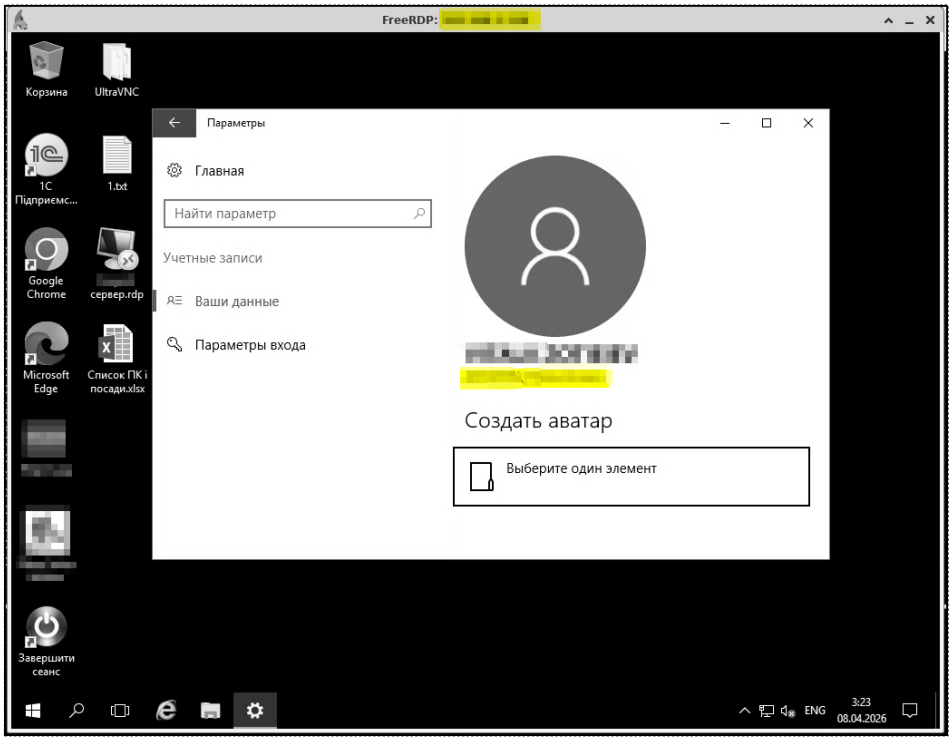


Figure 2 – RDP session on "SERVREGIONAL" opened with the captured sysadmin credentials.

Step 5 - Browser-saved credential harvest. Chrome's saved-password store on "SERVREGIONAL" contained working credentials for:

- the enterprise MDM platform (the enterprise MDM that manages every endpoint).
- Physical-security console for the regional facility (18 CCTV cameras + remote control of perimeter gates).

(Figure 3).

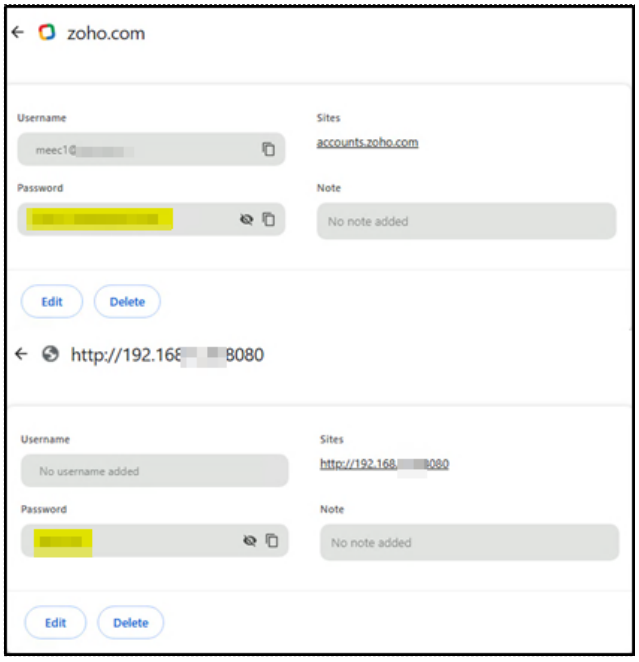


Figure 3 – Chrome saved-passwords dialog on "SERV1" exposing MDM and physical-security credentials.

Step 6 - Physical security access. With the recovered credentials, the audit team logged in to the physical-security console of the regional facility. The dashboard exposed 21 active CCTV streams of the production area (Figure 4) and remote Open / Close controls for the perimeter gates (Figure 5). The audit team did not actuate the gates - the access was demonstrated and documented only.

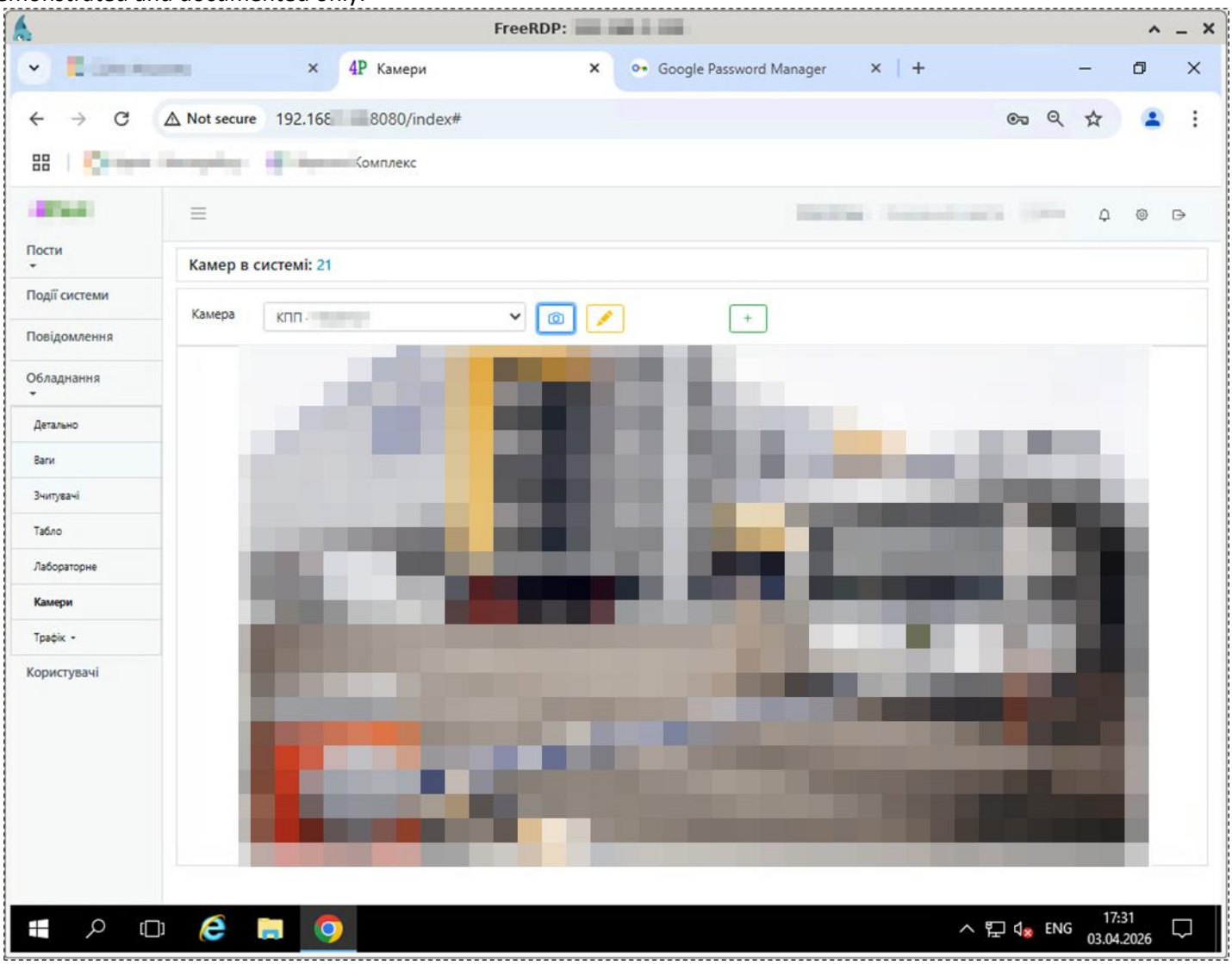


Figure 4 – Physical-security console: live CCTV grid (18 cameras of the regional facility).

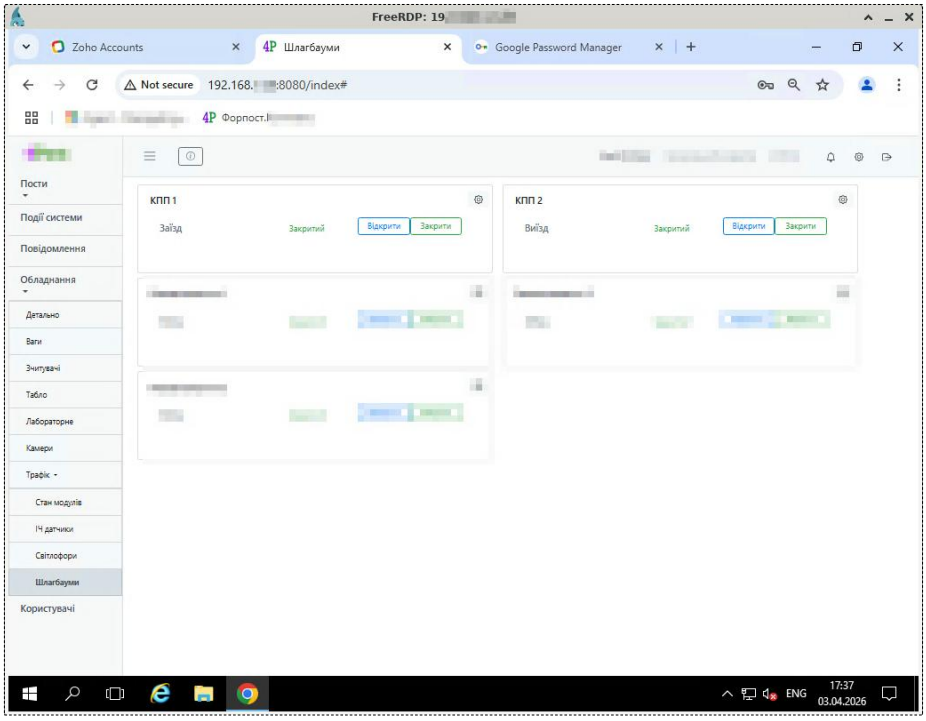


Figure 5 – Physical-security console: perimeter-gate control panel (Open / Close buttons).

Step 7 - 14-day persistence check. Fourteen days after the spray, the audit team re-tested the 41 captured passwords. 37 of the 41 accounts still authenticated with the same password. The SOC had not detected the Kerberos password-spray campaign and had not triggered a forced password reset on the affected accounts (Figure 6).

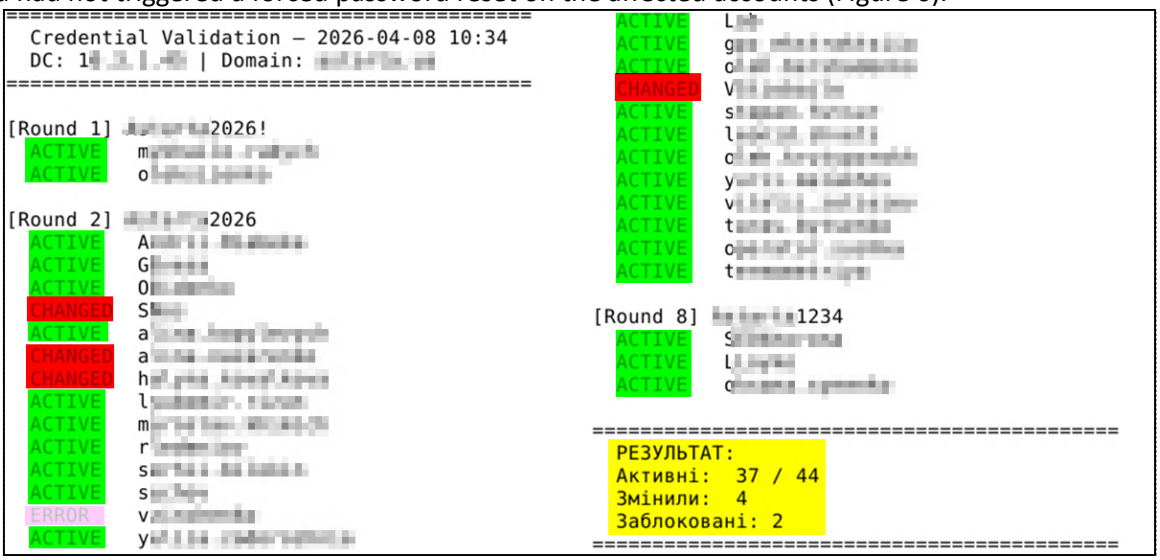


Figure 6 – 14-day persistence check: 34 of 41 sprayed passwords still valid; SOC did not detect.

AD-R2 – Multi-Factor Authentication Not Enforced on Microsoft 365, VPN, RDP - Common Root Cause of Two Independent Compromise Chains

RISK BRIEF

ID	AD-R2
RISK LEVEL	**HIGH**
SUMMARY	Multi-factor authentication is not enforced on Microsoft 365 (general staff), the corporate VPN, or RDP via domain credentials. Any captured password — by any means — grants immediate access to mail, VPN, and remote desktop. This single defect converts otherwise self-contained findings into complete attack chains: password spray (AD-R1) and phishing (SOCIAL-R1) both yielded immediate access without further verification.
VULNERABLE SERVICES	Microsoft 365 tenant (general-staff scope, accounts outside the pilot-group Conditional Access policy); FortiClient SSL VPN gateway; RDP across servers reachable from the VPN
RECOMMENDATIONS	Enforce phishing-resistant MFA (FIDO2 / number-matching) on every authentication service that accepts domain credentials — Microsoft 365, VPN, and RDP. Block legacy auth protocols (POP3, IMAP, SMTP AUTH) that bypass MFA. Set an MFA-enrollment deadline; lock accounts that miss it. Audit and remove any auth path that bypasses MFA.

RISK ASSESSMENT

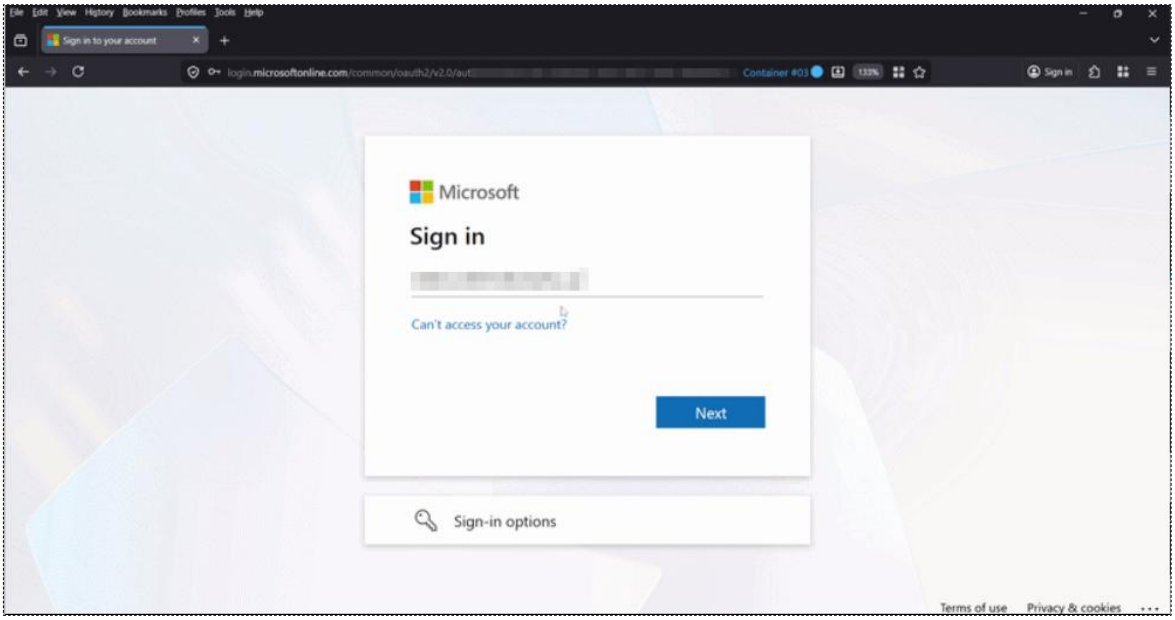
	SEVERITY	DESCRIPTION	
VULNERABILITY	**HIGH**	MFA is not enforced on the general-staff scope of Microsoft 365, on the FortiClient VPN, or on RDP. Compromised domain credentials in this population grant immediate, full access with no additional verification.	
THREAT	**HIGH**	Threat Actor Profiles	Insider with Remote Access
		Attack scenario	The attacker obtains domain credentials by any path. Without a second factor, the credentials grant immediate full access to Microsoft 365 (mail, Teams, SharePoint, OneDrive) and to the internal network via VPN. Follow-on activity includes searching mail and OneDrive for sensitive documents, impersonating the user in internal communications, and lateral movement on the internal network.
		Conditions	Only one valid set of domain credentials is required. Festival Finance has been compromised twice during this engagement (AD-R1 → 44 accounts; SOCIAL-R1 → 4 session tokens).
IMPACT	**HIGH**	Actual Impact	Demonstrated through two independent chains. AD-R1: 44 compromised domain accounts were used to log in to Microsoft 365 (mail, Teams) and to FortiClient VPN, granting internal-network access, browser-saved credential harvest on SERVREGIONAL, and downstream compromise of the corporate MDM and a regional facility's physical-security console. SOCIAL-R1: 4 Microsoft 365 session tokens captured

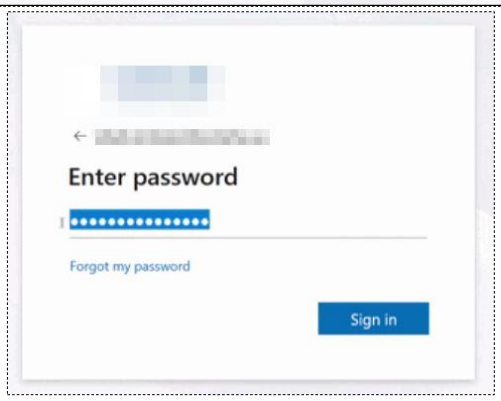
		via Evilginx2 phishing - immediate access to mail, Teams, and SharePoint with no further prompts; in several cases access persisted beyond the SOC's containment window.
	Potential Impact	The same vector applies to any future credential leak. Without MFA, none of the typical defensive controls (anomalous-sign-in detection, suspicious-session blocking, geolocation monitoring) stops the attack at the front door. MFA on Microsoft 365 and VPN would have halted both AD-R1 and SOCIAL-R1 at the authentication step, regardless of how the credentials were obtained.

TECHNICAL DETAILS & PROOF OF CONCEPT

Microsoft 365 without MFA. Credentials captured via password spray (AD-R1) were entered on the standard Microsoft 365 sign-in page. The tenant returned a successful authentication with no second-factor prompt. The audit team confirmed access to:

- **Outlook Web Access** - mailbox, contacts, calendar.
- **Microsoft Teams** - chats, channels, files, ability to post as the user.
- **SharePoint / OneDrive** - personal and shared corporate files.
- **Account settings** - password change, notification preferences (the latter was used in SOCIAL-R1 to pre-empt the legitimate owner).





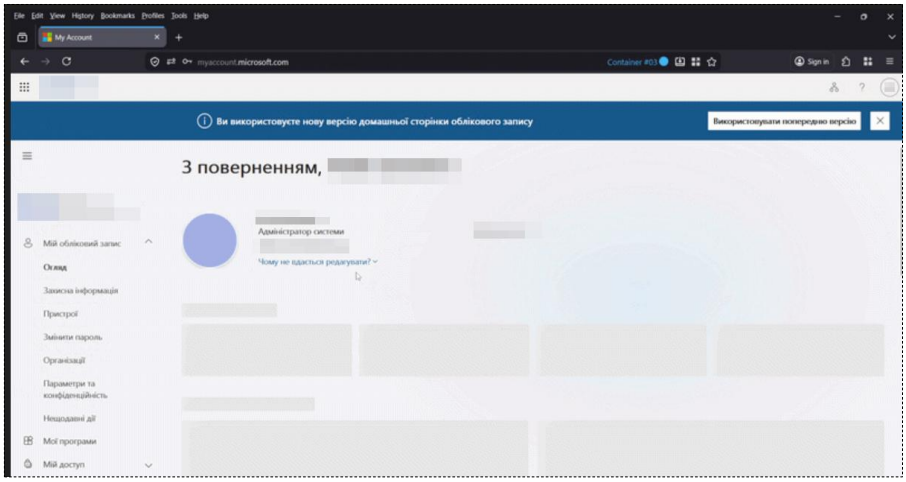


Figure 1 – Microsoft 365 portal accessed with compromised credentials - no MFA prompt. Access was indistinguishable from a legitimate sign-in.

FortiClient VPN without MFA. The same credentials were used to establish a FortiClient SSL-VPN session from an external virtual machine. The connection was authorised on the first attempt with no second-factor prompt; the audit team's VM received an IP address inside the corporate network.

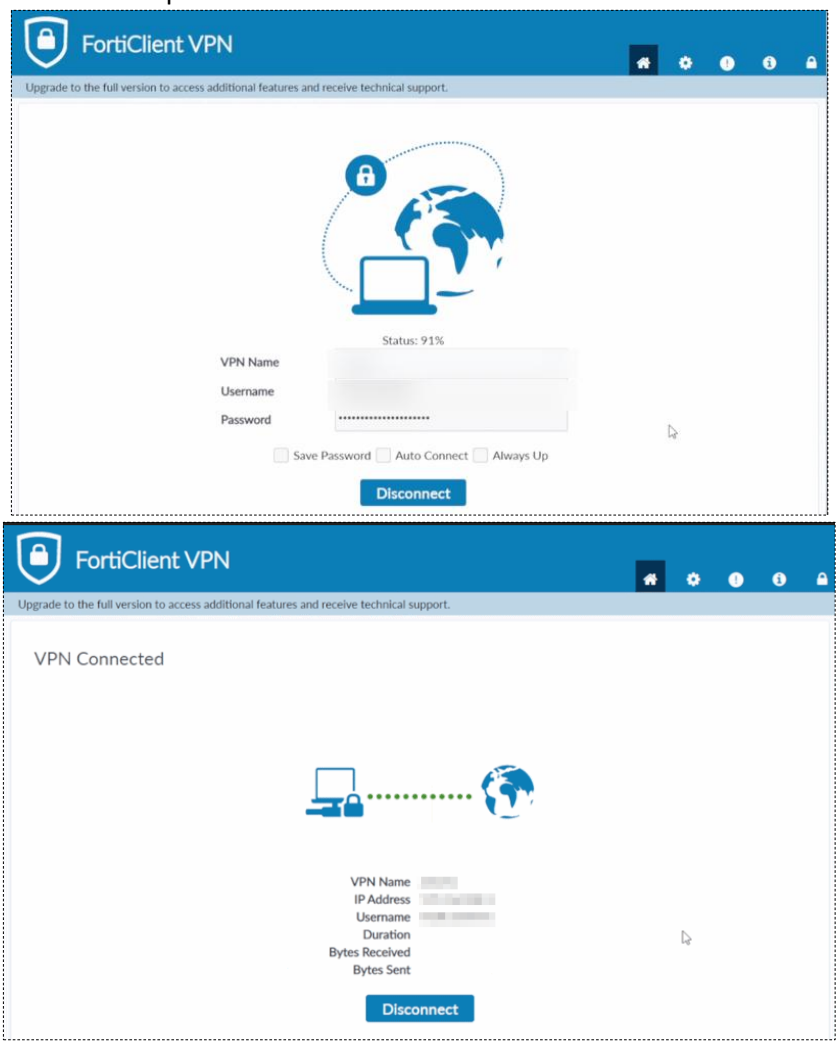


Figure 2 – FortiClient VPN connecting to the corporate network with compromised credentials - no MFA prompt.

WEB-R2 – Unrestricted Public Access to Laravel Telescope Monitoring Dashboard

RISK BRIEF

ID	WEB-R2
RISK LEVEL	**MEDIUM**
SUMMARY	The back-office application's Laravel Telescope monitoring dashboard is exposed to the public Internet with no authentication, authorization, or network restriction. Anyone can browse failed requests, exceptions, request bodies, response payloads, internal hostnames, and unredacted session tokens issued by the application.
VULNERABLE SERVICES	admin.festival-finance.example/telescope
RECOMMENDATIONS	Telescope must not be publicly reachable. Enforce a strict in-app authorization gate (viewTelescope) limited to designated administrator and developer accounts. Apply a compensating infrastructure control by restricting /telescope to VPN / corporate IP ranges at the reverse proxy or WAF. Configure Telescope to redact secrets, session tokens, and sensitive request parameters in both requests and responses.

RISK ASSESSMENT

	SEVERITY	DESCRIPTION	
VULNERABILITY	**HIGH**	Production monitoring dashboard is reachable from the public Internet without authentication.	
THREAT	**HIGH**	Threat Actor Profiles	External Attacker
		Attack scenario	The attacker browses Telescope directly, collects request and response payloads, harvests valid session tokens, and uses them to impersonate users.
		Conditions	All conditions for exploitation are already present - no chaining, no special context.
IMPACT	**MEDIUM**	Actual Impact	Confirmed unauthenticated access to the Requests and Exceptions sections, including failed HTTP 500 requests with timestamps, internal hostnames, internal proxy/application IP addresses, user identifiers, names, email addresses, full JSON request bodies, client IP addresses, and unredacted response data containing valid session tokens.
		Potential Impact	Long-term passive monitoring would yield additional credentials and tokens, enabling user impersonation. Exposed internal hostnames and IP space enables targeted follow-up against the internal network. The monitoring-rule creation feature could be abused to widen the data captured by Telescope.

TECHNICAL DETAILS & PROOF OF CONCEPT

Automated scanning of the back-office application surfaced a Laravel Telescope monitoring dashboard at "admin.festival-finance.example/telescope" reachable directly from the public Internet (Figure 1). The application implements no authentication, authorization, or network ACL on this path - the dashboard responds to any anonymous request.

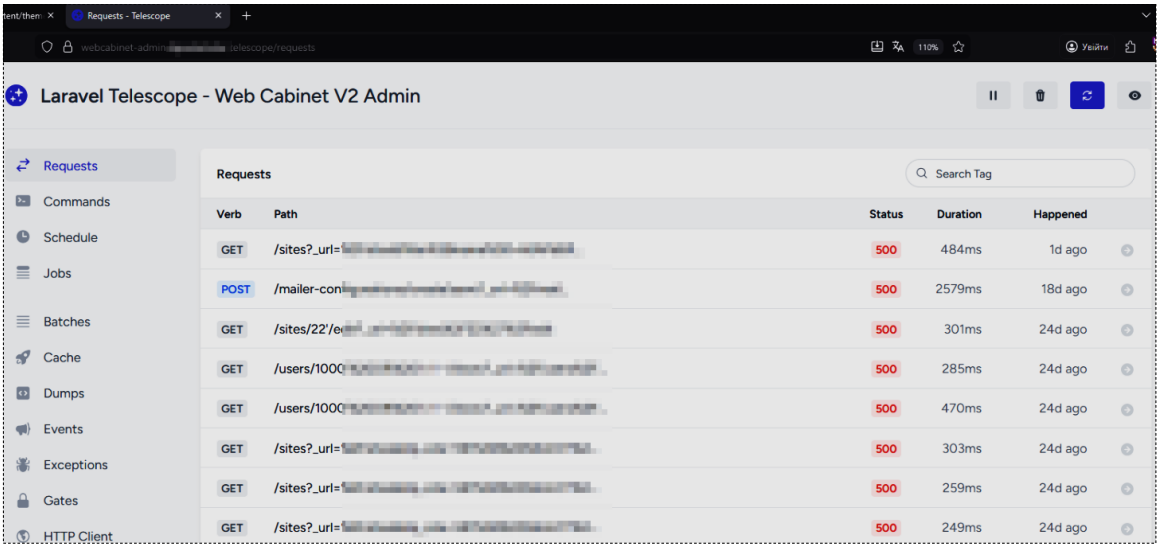


Figure 1 – Telescope dashboard reachable from the public Internet.

Within Telescope, the audit team browsed the "Requests" and "Exceptions" sections and confirmed disclosure of:

- Request timestamps, internal hostnames, and internal proxy IP addresses (Figure 2)

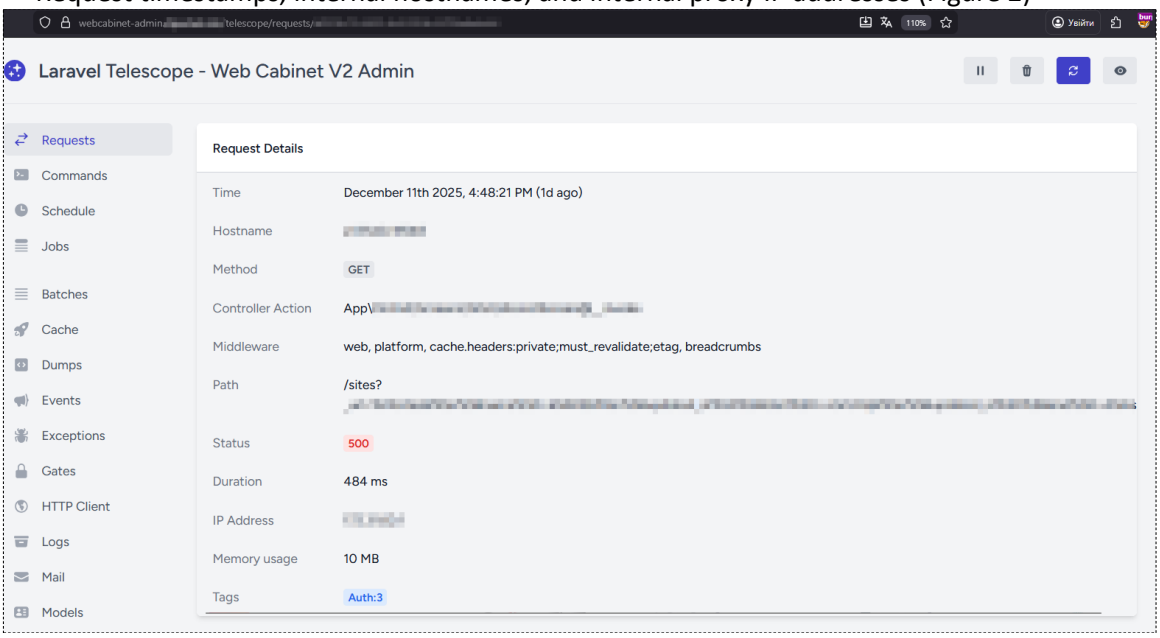


Figure 2 – Failed request record with internal hostname and proxy IP visible.

- Full request bodies (JSON payloads, possibly containing sensitive data) (Figure 3)

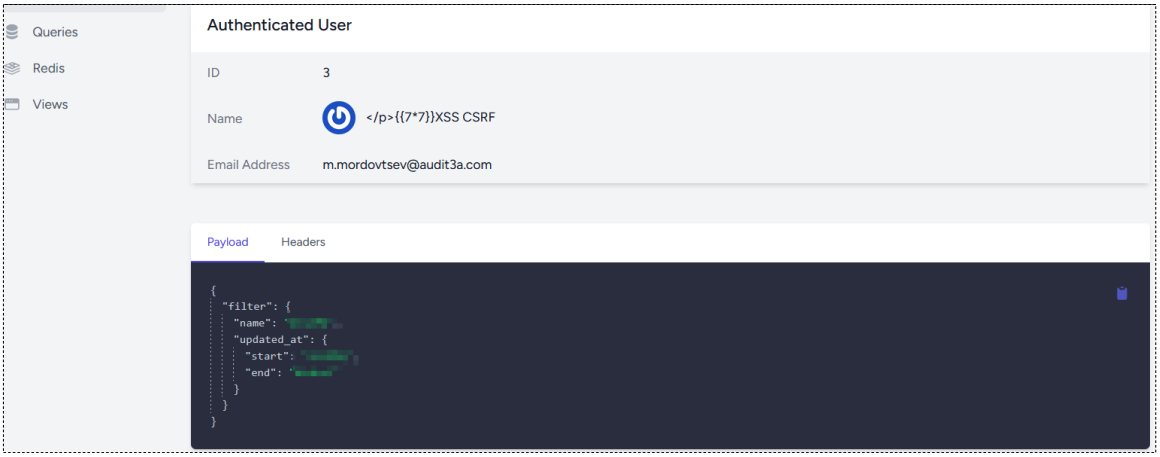


Figure 3 – Full JSON request body captured by Telescope.

- Request headers including authentication-related headers (Figure 4)

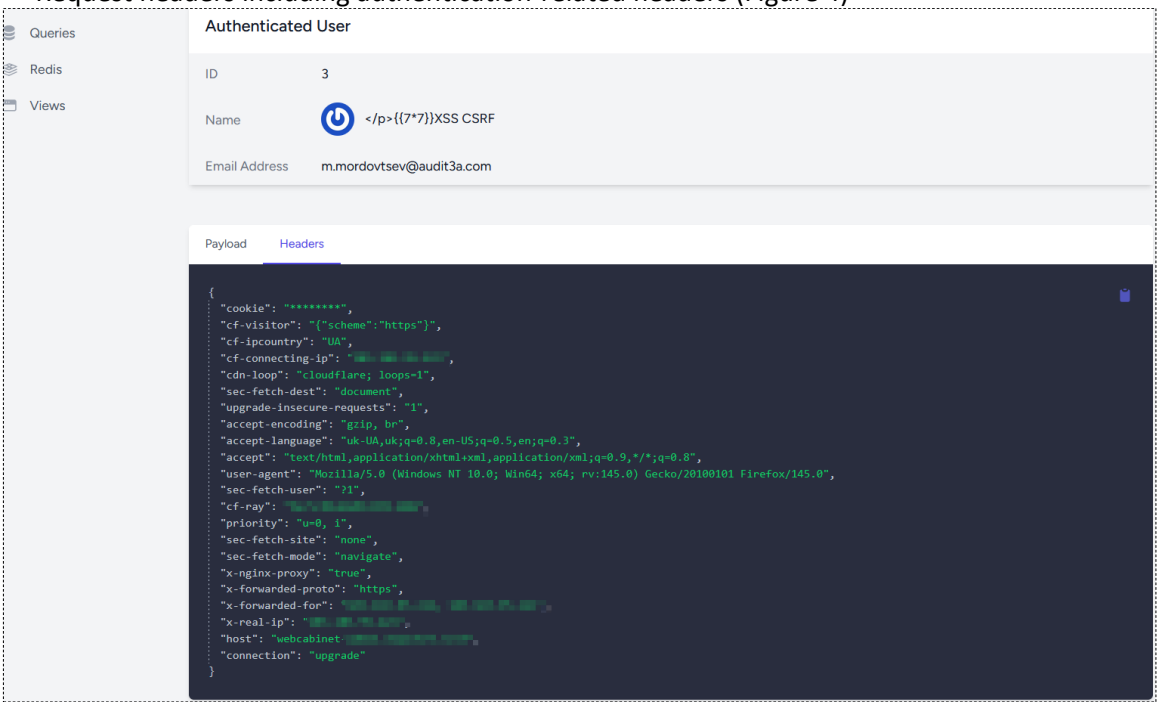


Figure 4 – Request headers captured (including auth-related metadata).

- Response data, including unredacted session tokens issued by the application (Figure 5)

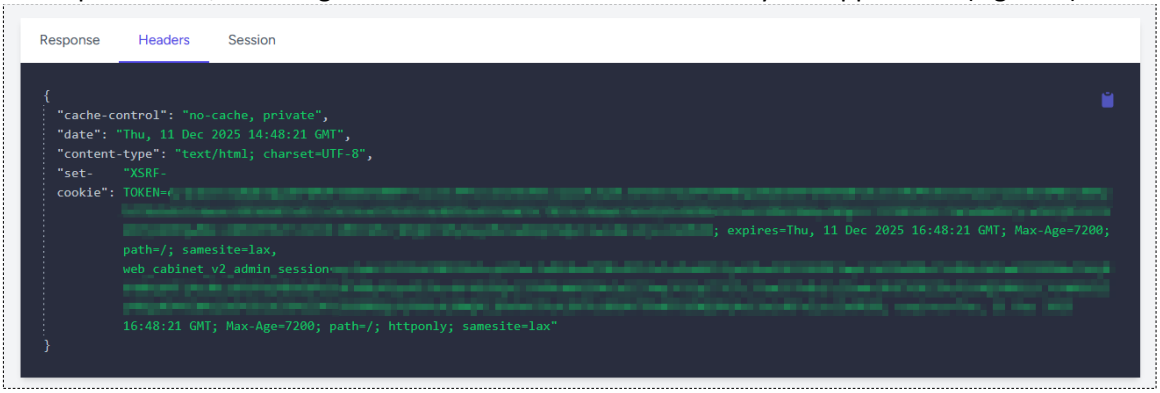


Figure 5 – Unredacted session token visible in response payload.

The audit team also confirmed the ability to create new monitoring rules via the dashboard UI - an attacker who finds insufficient data in existing logs can configure additional tags to capture future requests (Figure 6).

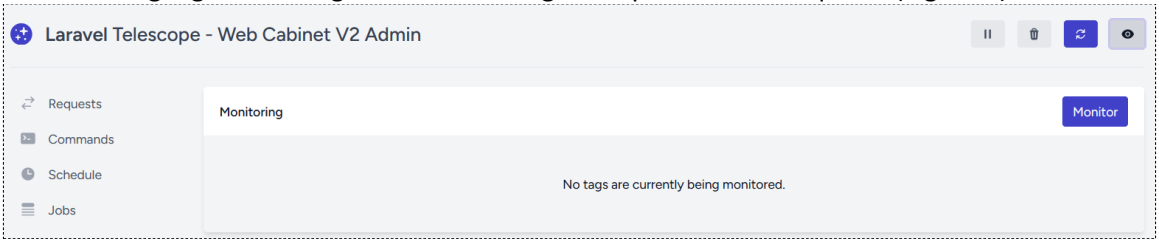


Figure 6 – New monitoring rule created via the dashboard UI.

WEB-R3 – Cross-Customer PIN Disclosure via PIN-Reminder Function

RISK BRIEF

ID	WEB-R3
RISK LEVEL	**MEDIUM**
SUMMARY	An authenticated customer can abuse the "PIN reminder" function to have the card PIN of another customer's payment card delivered to the attacker's own email address. The backend does not verify that the card identified by the supplied cardHash belongs to the requesting user.
VULNERABLE SERVICES	app.festival-finance.example
RECOMMENDATIONS	Implement strict server-side ownership validation: when processing PIN-reminder requests, the backend must verify that the card identified by the supplied cardHash is linked to the currently authenticated user, and reject the request otherwise. Add automated authorization tests covering this endpoint.

RISK ASSESSMENT

	SEVERITY	DESCRIPTION	
VULNERABILITY	**HIGH**	Missing server-side ownership check in the PIN-reminder endpoint.	
THREAT	**LOW**	Threat Actor Profiles	External Attacker
		Attack scenario	The attacker intercepts the PIN-reminder request via an HTTP proxy, replaces their own cardHash with the victim's cardHash, and receives the victim's PIN and the last four digits of the card number to their own registered email.
		Conditions	The attacker must hold an account in the application and know (or have obtained) the victim's cardHash.
IMPACT	**HIGH**	Actual Impact	Confirmed unauthorized disclosure of card PINs and partial card numbers belonging to other application customers (validated using audit-team test accounts).
		Potential Impact	If combined with another flaw that leaks cardHash values (e.g. an enumeration weakness or a related IDOR), the attacker could harvest PINs at scale - a PCI DSS reportable event.

TECHNICAL DETAILS & PROOF OF CONCEPT

The audit team registered two test accounts ("alice@festival-finance.example" and "bob@festival-finance.example"), each with its own payment card and "cardHash".

When Alice triggered the PIN-reminder action in the application, the client sent a JSON-RPC request to the API:

```
POST /api/v1/card/ HTTP/2
Host: app.festival-finance.example
…
{
  "jsonrpc": "2.0",
  "method": "card@resendEpin",
  "params": { "cardHash": "<alice_cardHash>" }
}
```

The server validated the session belonged to Alice and emailed Alice her PIN (Figure 1).

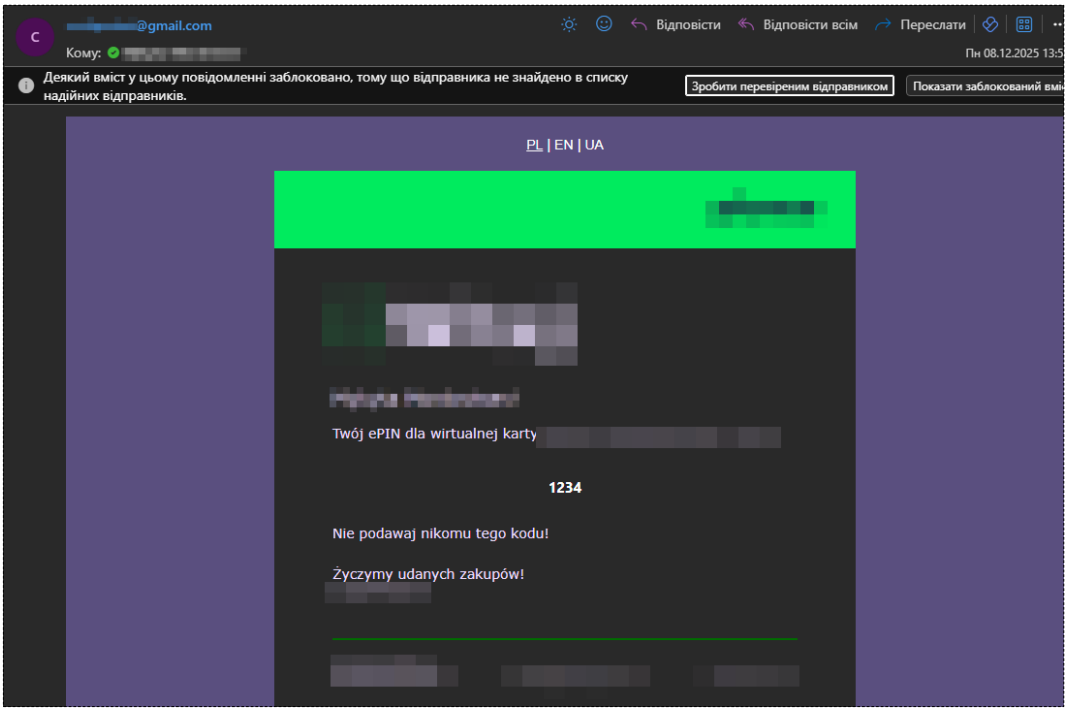


Figure 1 – Legitimate PIN-reminder request returns the requester's own PIN.

Repeating the request from Alice's session, but with Bob's "cardHash" substituted in the JSON body, the backend processed it successfully and delivered Bob's PIN and the last four digits of Bob's card number to Alice's email (Figures 2-3). The backend never verified that the "cardHash" actually belonged to the authenticated user.

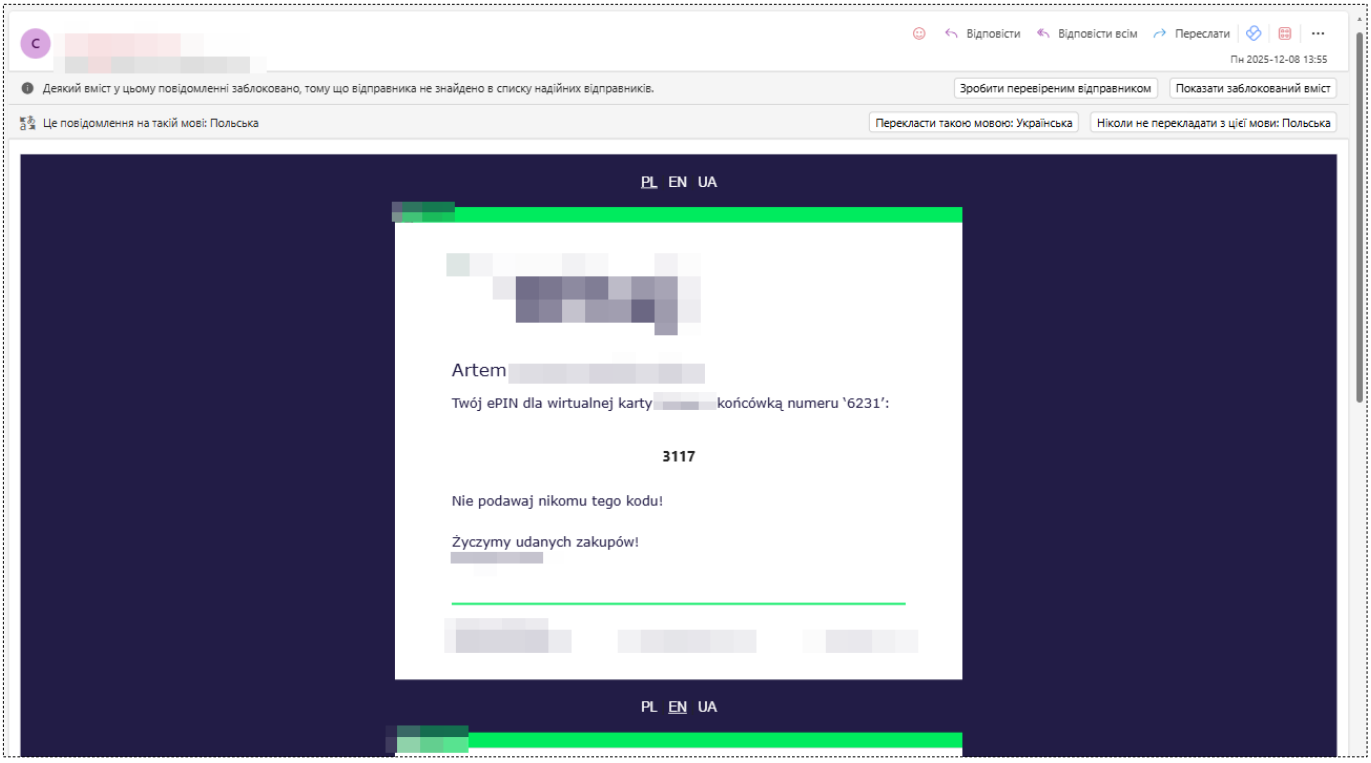


Figure 2 – Request body with the victim's "cardHash" substituted in place of the attacker's.

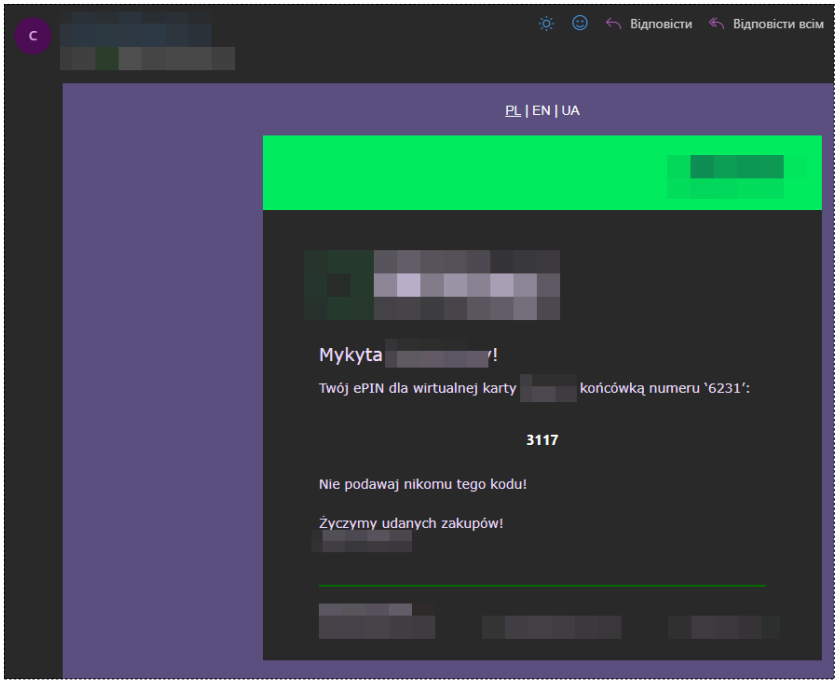


Figure 3 – Inbox of the attacker's email account, containing the victim's PIN and partial card number.

WEB-R4 – Multi-Vector Stored Cross-Site Scripting (XSS) in Back-Office

RISK BRIEF

ID	WEB-R4
RISK LEVEL	**MEDIUM**
SUMMARY	Multiple stored XSS vectors were identified in the back-office admin application. Authenticated users can inject HTML and JavaScript via the Site-Specific translations, Username, and Image upload (filename, uploader) fields - the payload is stored server-side and executes in the browsers of other administrators who view the affected pages. Session theft, page-content tampering, and CSRF-token extraction were demonstrated.
VULNERABLE SERVICES	admin.festival-finance.example
RECOMMENDATIONS	Enforce server-side input sanitization on every text field (escape HTML/JS metacharacters; reject control bytes). Validate input on both client and server. Deploy a strict Content Security Policy (CSP) that blocks inline scripts and disallows untrusted script origins. Add hardening response headers (X-Content-Type-Options, Referrer-Policy). Audit every form field for output-context-aware escaping. Train developers on secure coding for output contexts.

RISK ASSESSMENT

	SEVERITY	DESCRIPTION	
VULNERABILITY	**HIGH**	Stored cross-site scripting across multiple input fields - JS payloads persist in the database and execute in the browsers of other users.	
THREAT	**LOW**	Threat Actor Profiles	Insider with Remote Access
		Attack scenario	The attacker stores a JavaScript payload in a translation, username, or image-filename field; another administrator opens the affected page and the payload executes, stealing the session cookie or CSRF token.
		Conditions	Operation requires an authenticated back-office account. The realistic acquisition path is the phishing chain documented in SOCIAL-R1 - a captured employee credential whose owner happens to hold the back-office admin role. A pure outsider with only a self-registered customer account cannot reach the affected pages.
IMPACT	**HIGH**	Actual Impact	The audit team successfully exploited Stored XSS and exfiltrated session data, including session cookies and CSRF tokens.
		Potential Impact	Full administrative session takeover; arbitrary actions on behalf of any administrator who views a poisoned page; long-term persistence as long as the payload remains in the database.

TECHNICAL DETAILS & PROOF OF CONCEPT

Three independent stored-XSS vectors were confirmed in the back-office application:

Vector 1 - Site-Specific translations. Arbitrary HTML/JS injected into the "Translation" field on "/site-translations/<id>/show" is stored without sanitization and rendered verbatim on "/sites/<id>/show". The audit team submitted the payload:

```
'"></div><img src=1 onerror="alert('Domain: ' + document.domain
  + '\nCSRF Token: ' + document.querySelector('meta[name=csrf_token]').content);">
```

The payload triggered on page load, leaking the application domain and the CSRF token in an alert (Figures 1-2).

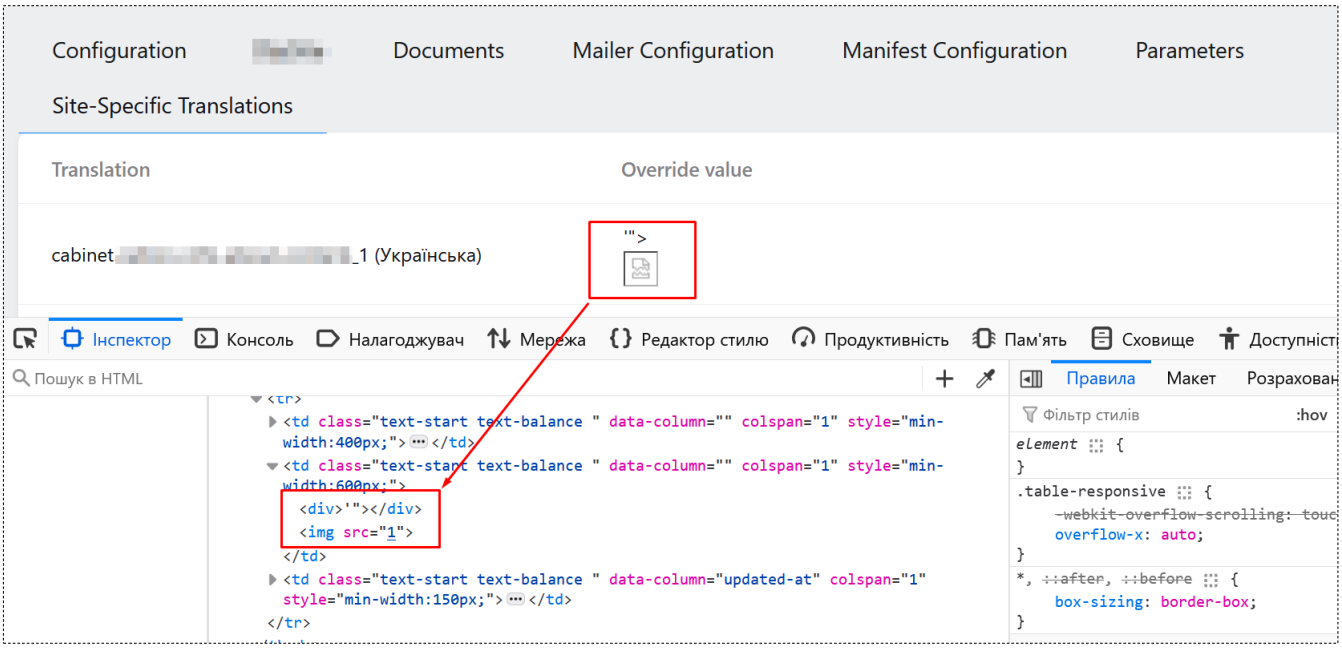


Figure 1 – Injected "<img>" tag rendered into the page DOM.

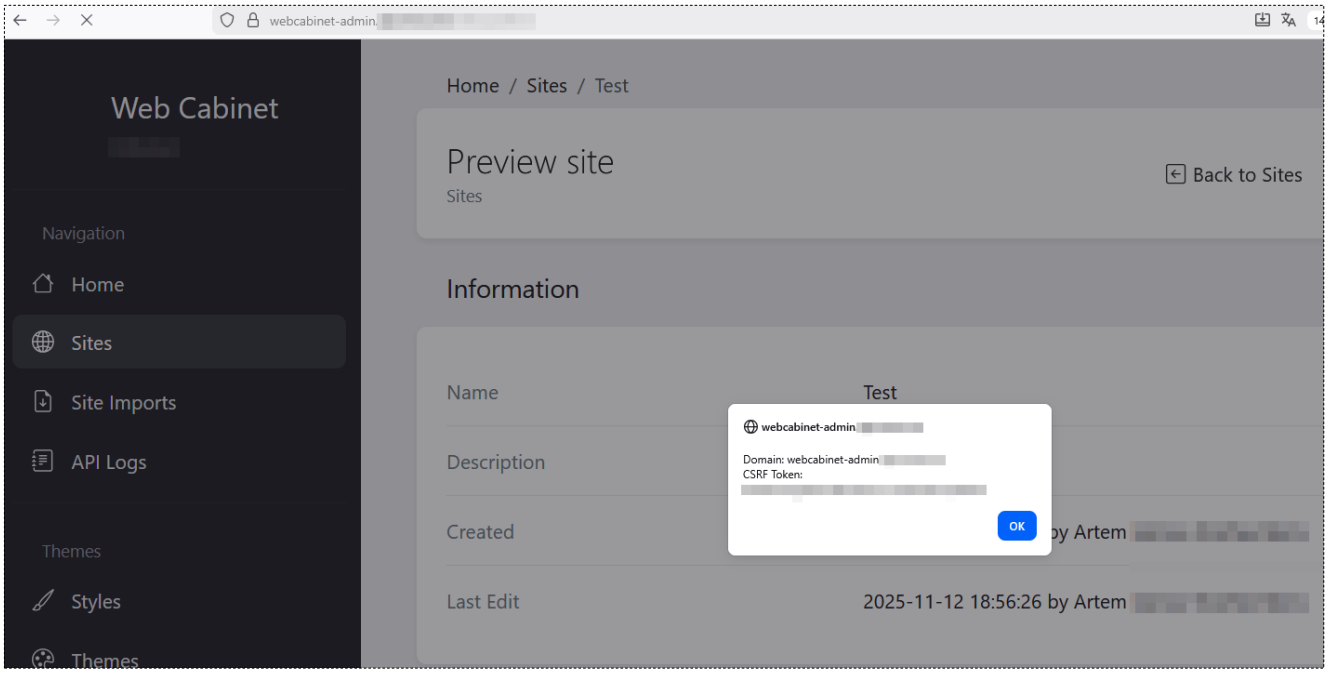


Figure 2 – Browser alert displaying the application domain and the user's CSRF token.

Vector 2 - Username field. HTML embedded in the username persists into the "Last edit" column on translation pages, where it is rendered into the DOM unescaped (Figure 3).

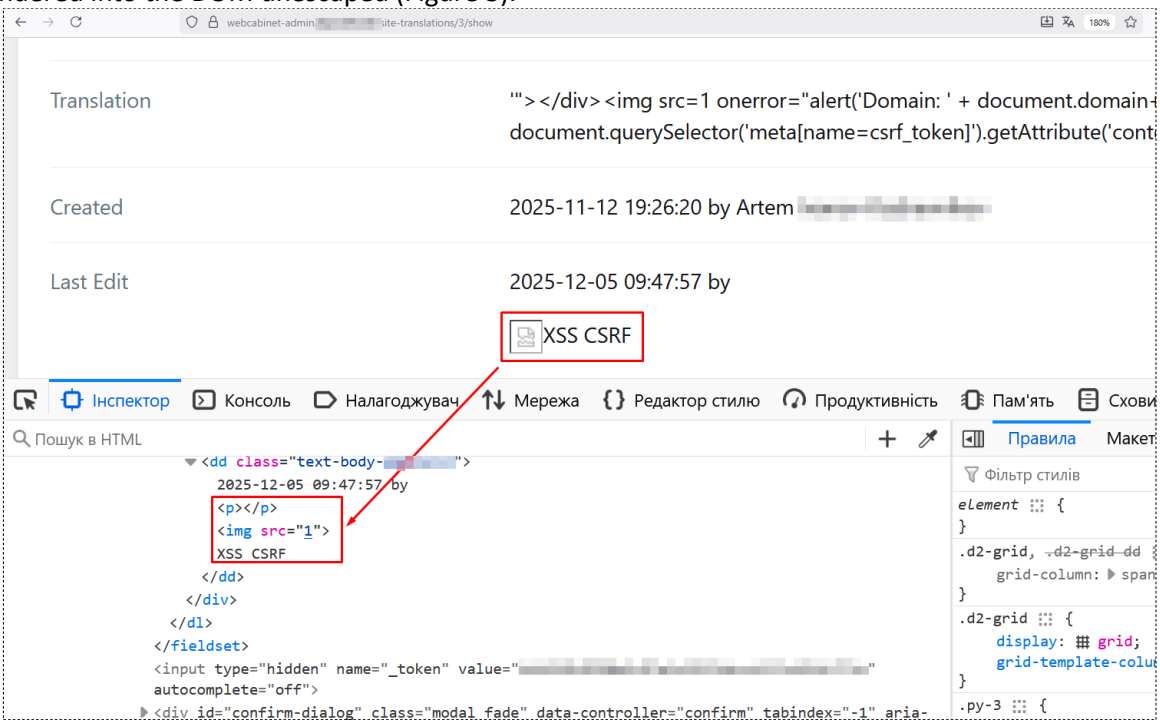


Figure 3 – XSS payload persisted via the username field, rendered on translation page.

Vector 3 - Image upload metadata. The "Name" field (image filename) and the "Created by" field (uploader name) accept HTML payloads that execute when the gallery page is rendered (Figures 4-5).

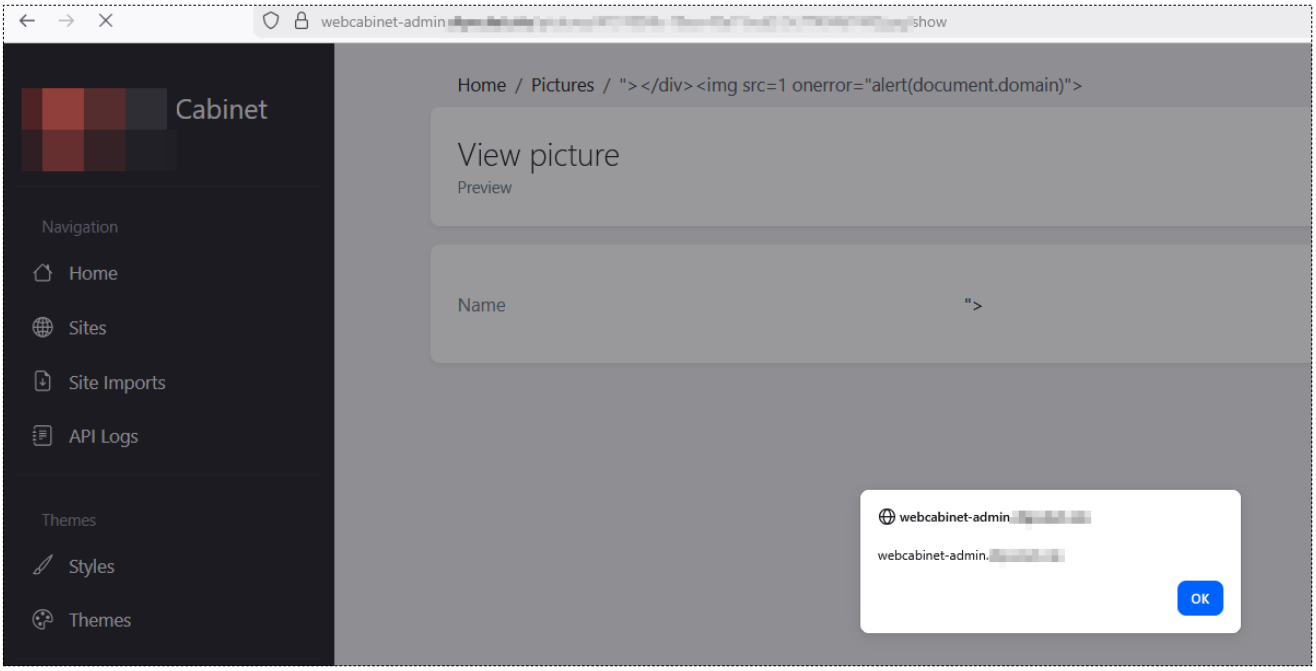


Figure 4 – Alert triggered by malicious image-filename metadata.

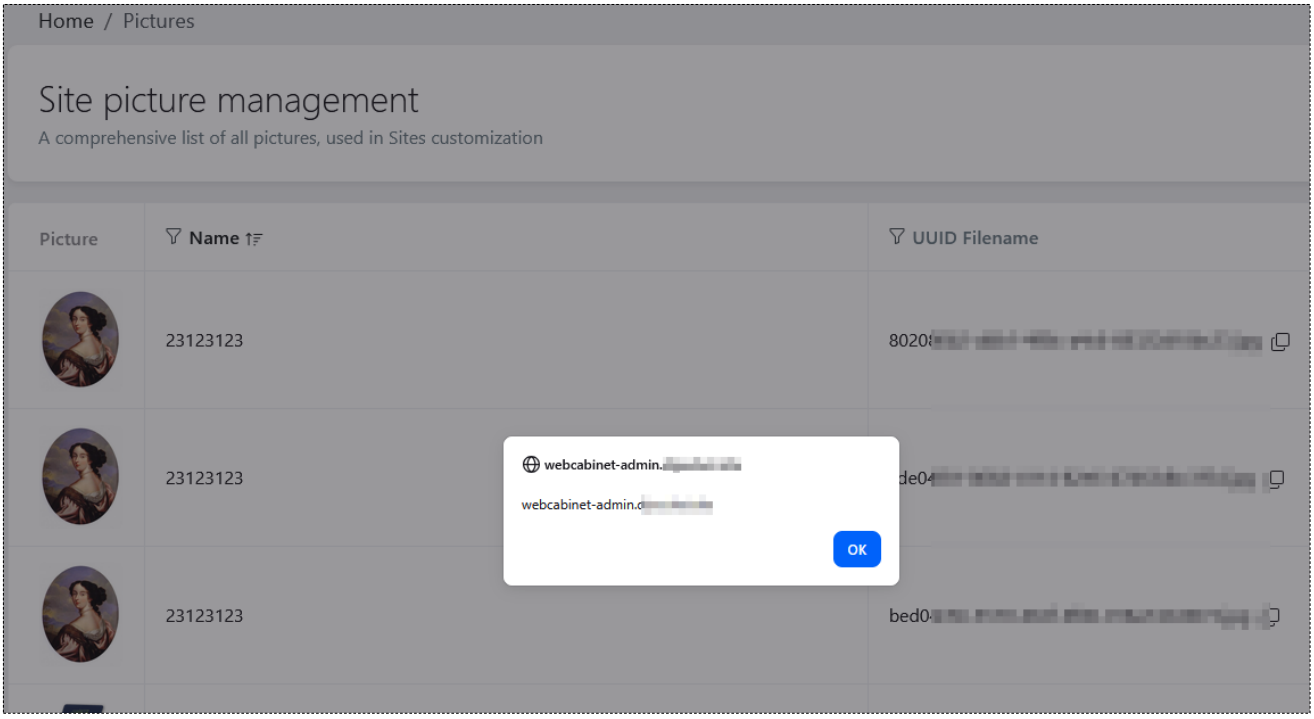


Figure 5 – Alert triggered when the gallery enumerates uploaded images.

In every case the application returned the injected markup with content type "text/html" and no "Content-Security-Policy" header, allowing inline scripts to execute (Figure 6).

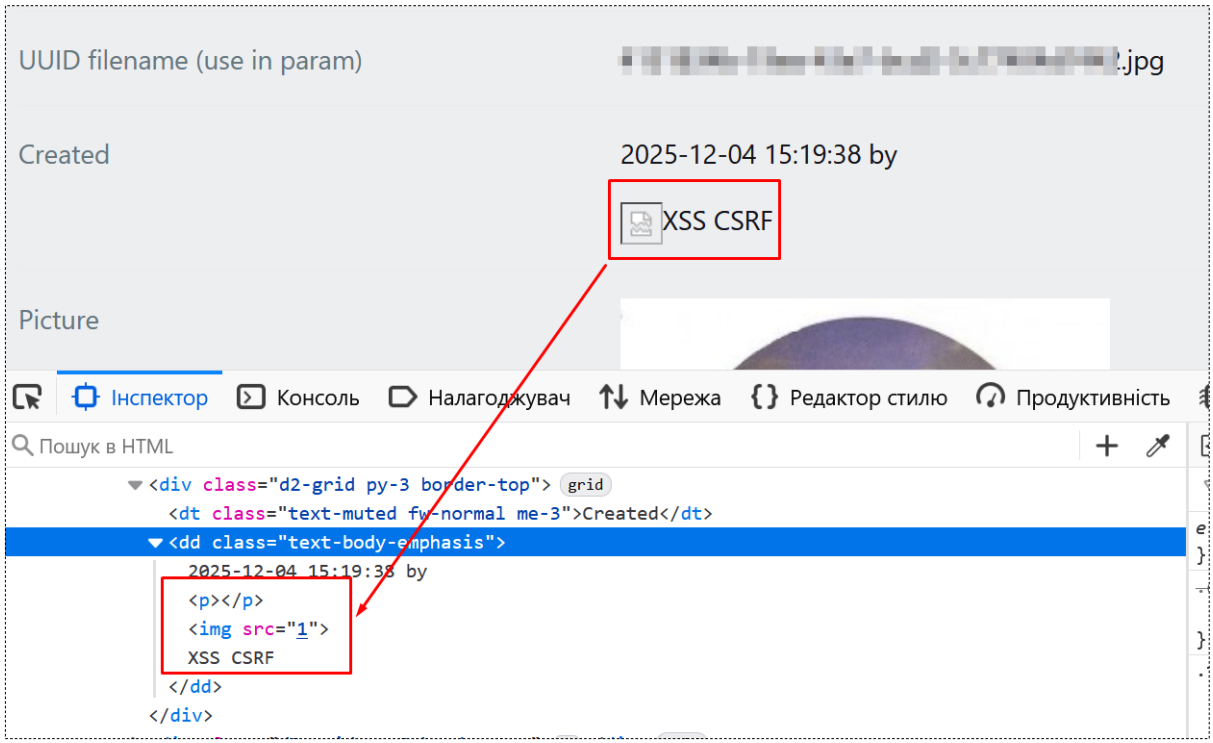


Figure 6 – Browser dev-tools view confirming HTML injection in server response.

NET-EXT-R1 – Malformed DMARC Record + Permissive SMTP Enable Internal From:-Field Spoofing

RISK BRIEF

ID	NET-EXT-R1
RISK LEVEL	**MEDIUM**
SUMMARY	The DMARC record for festival-finance.example is published with a duplicated v=DMARC1 tag, making the record syntactically invalid per RFC 7489. Many receiving mail servers discard malformed DMARC and therefore do not enforce the published p=reject policy. Combined with permissive SMTP behavior on the mail-protection front-end, this allows an external attacker to deliver email to an internal recipient with the From: header impersonating any internal employee.
VULNERABLE SERVICES	DNS / SMTP infrastructure of festival-finance.example (and festival-finance.net)
RECOMMENDATIONS	Publish a syntactically correct DMARC record - a single v=DMARC1 tag at the start, followed by the intended policy. Validate via external scanners (MXToolbox, EasyDMARC). Restrict the SMTP relay so unauthenticated external senders cannot inject mail addressed to internal recipients. Enable / tighten DKIM signing on every outbound stream. Configure the receiving gateway to reject (not quarantine) messages that fail DMARC alignment. Monitor DMARC aggregate reports to detect spoofing attempts.

RISK ASSESSMENT

	SEVERITY	DESCRIPTION	
VULNERABILITY	**MEDIUM**	Malformed DMARC TXT record combined with permissive SMTP settings allows external spoofing of From: as an internal address.	
THREAT	**MEDIUM**	Threat Actor Profiles	External Attacker
		Attack scenario	The attacker delivers email with the From: header set to an internal address. SPF or DKIM may pass independently, but receiving servers that treat the malformed DMARC record as invalid do not enforce p=reject. The spoofed email is delivered with no signal in DMARC reports.
		Conditions	Exploitation relies solely on external email delivery and public DNS lookup behaviour. Effective wherever the receiving server performs DMARC validation but discards malformed records (common in 2025 deployments).
IMPACT	**MEDIUM**	Actual Impact	The audit team successfully delivered a spoofed message to an internal Festival Finance user, with the From: header impersonating another internal employee.
		Potential Impact	Enables successful phishing campaigns (see SOCIAL-R1), business-email-compromise scams against third parties who receive mail "from" Festival Finance, reputational damage, and customer-trust erosion.

TECHNICAL DETAILS & PROOF OF CONCEPT

DMARC record audit. The published DMARC record contains two consecutive "v=DMARC1" tags:

```
v=DMARC1; v=DMARC1; p=reject; rua=mailto:postmaster@festival-finance.example; ...
```

RFC 7489 requires a single "v=" at the start of the record. Common DMARC parsers reject the record as malformed; affected receivers silently ignore the entire policy and never enforce "p=reject". External validators (MXToolbox, EasyDMARC) flag the record as non-compliant (Figures 1-2).

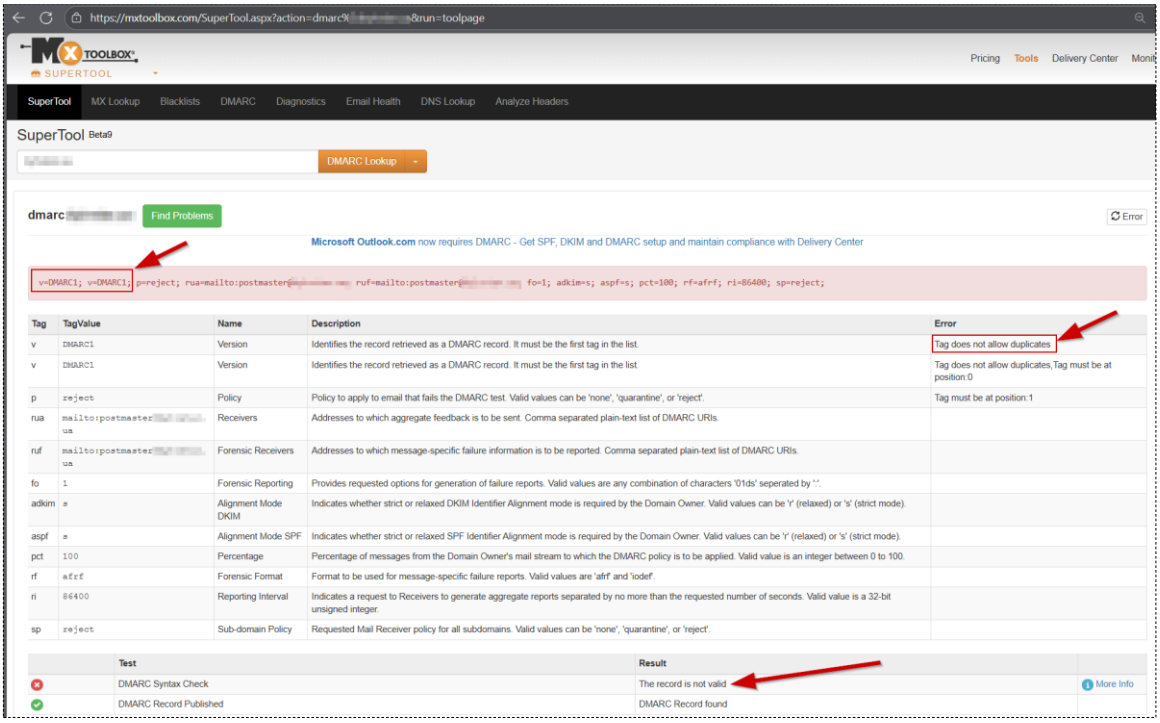


Figure 1 – MXToolbox flags the DMARC record as syntactically invalid.

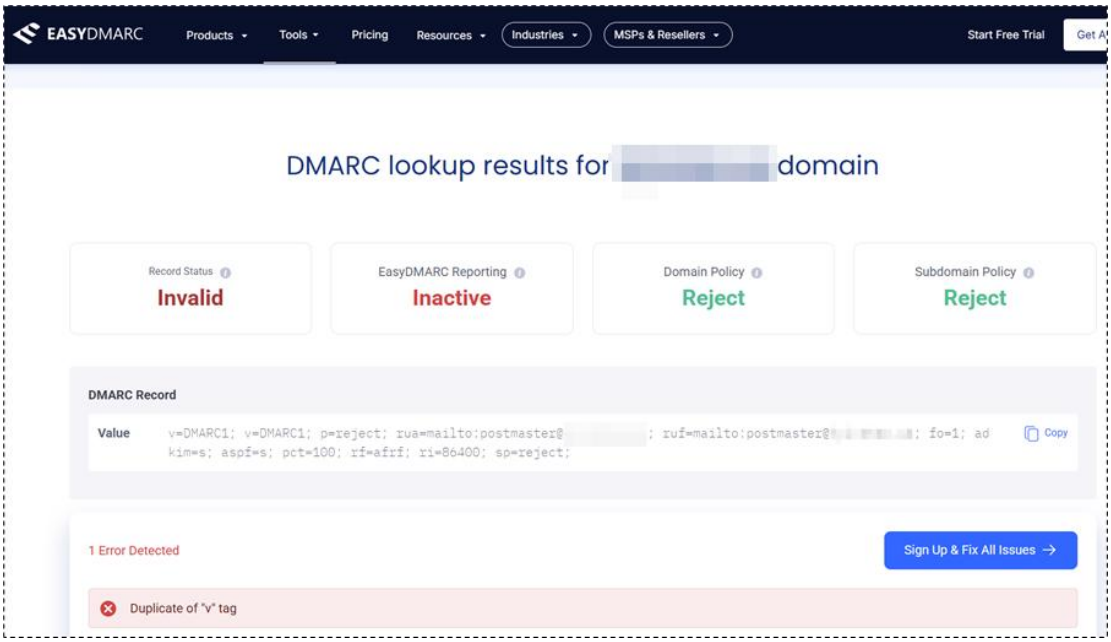


Figure 2 – EasyDMARC marks the record as non-compliant for the same reason.

From-field spoofing PoC. A direct SMTP session to the mail-protection front-end accepted a message with an arbitrary internal "From:" header and delivered it to an internal recipient:

```
$ telnet *.mail.protection.outlook.com 25
220 ...mail.protection.outlook.com Microsoft ESMTP MAIL Service ready
EHLO admin
MAIL FROM:<sender@external.example>
RCPT TO:<recipient@festival-finance.example>
DATA
From: pxxx.x@festival-finance.example
Subject: ...
...
.
250 2.0.0 OK Queued mail for delivery.
```

The recipient's inbox displays the message as if it had been sent by the impersonated internal user, with no warning banner, no spam mark, and no DMARC failure indicator (Figure 3-4).

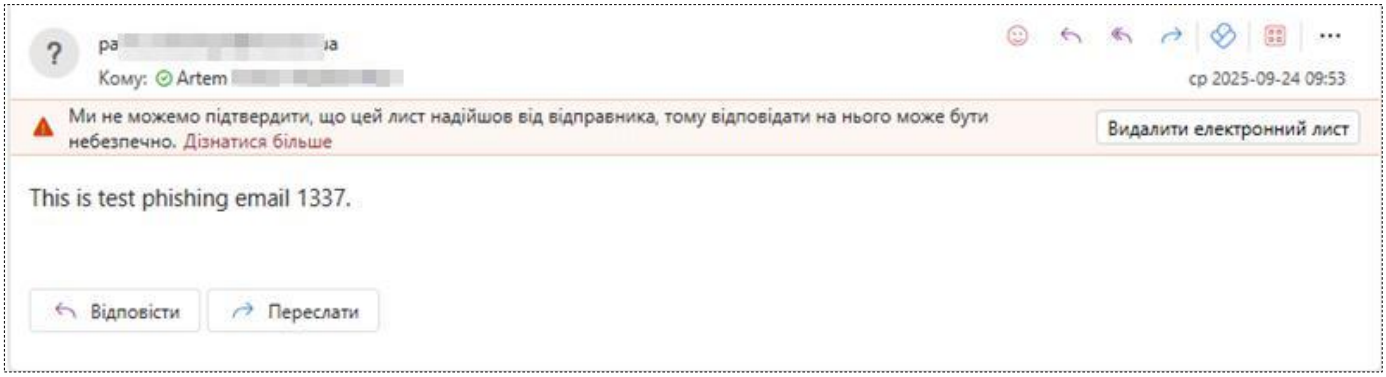


Figure 3 – Telnet SMTP session delivering a message with a spoofed internal "From:" header.

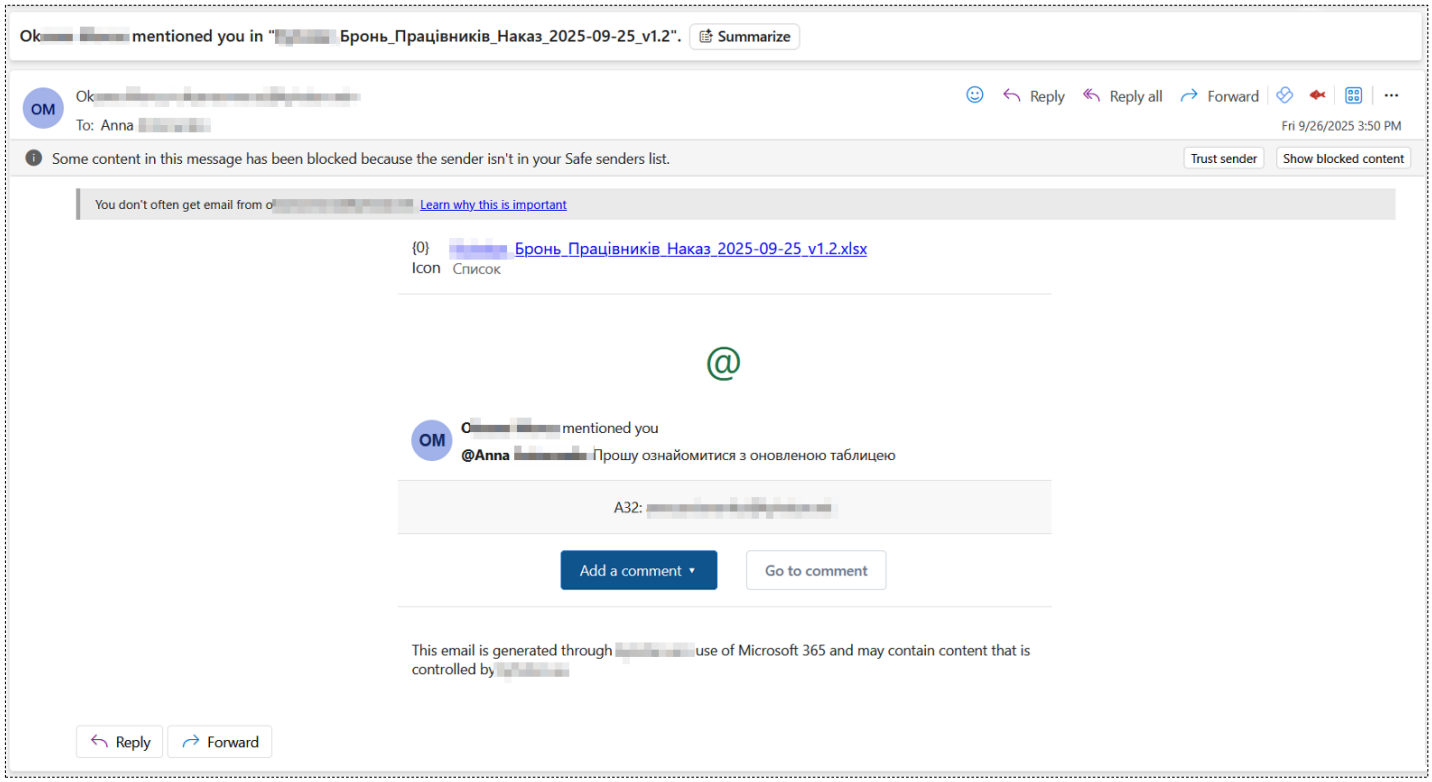


Figure 4 – Recipient's inbox view of the spoofed message - no warning, appears legitimate.

SOCIAL-R1 – Evilginx2 MITM Phishing Bypasses MFA - Microsoft 365 Account Takeover

RISK BRIEF

ID	SOCIAL-R1
RISK LEVEL	**MEDIUM**
SUMMARY	A two-stage phishing campaign was run against 187 Festival Finance employees using a credential-harvesting MITM proxy (Evilginx2) and a look-alike Microsoft 365 login page. 64 employees clicked, 7 entered credentials, 4 approved MFA — defeating multi-factor authentication and yielding 4 live Microsoft 365 session tokens. The audit team accessed Outlook, Teams, SharePoint, and Microsoft Entra as the compromised users and extracted 7,400+ employee email addresses from the tenant. The SOC contained the fastest case in 14 minutes.
VULNERABLE SERVICES	Festival Finance Microsoft 365 tenant; employee mailboxes
RECOMMENDATIONS	Enforce phishing-resistant MFA (FIDO2 / number-matching) for Microsoft 365 — app-notification approval alone is bypassable by MITM. Apply Conditional Access policies (location, device compliance, sign-in risk). Tighten email-gateway filtering on look-alike and freshly-registered domains. Run continuous user-awareness training with realistic phishing simulations. Fix the underlying email-spoofing weakness (NET-EXT-R1).

RISK ASSESSMENT

	SEVERITY	DESCRIPTION	
VULNERABILITY	**HIGH**	Phishing-driven credential harvesting plus session-token capture via Evilginx2 MITM proxy. Standard app-notification MFA is bypassed because the proxy transparently relays the MFA challenge and harvests the resulting session token.	
THREAT	**HIGH**	Threat Actor Profiles	External Attacker
		Attack scenario	The attacker delivers a phishing email impersonating Microsoft. The victim clicks, lands on the attacker's Evilginx proxy, enters credentials, completes MFA. The proxy captures the post-MFA session token and forwards the user to a benign 404 page. The attacker reuses the token from any IP to impersonate the user without re-authenticating.
		Conditions	No advanced email filtering or phishing-resistant MFA in place. Pre-existing DMARC weakness (NET-EXT-R1) increased message deliverability. Employee phishing-awareness gaps.
IMPACT	**LOW**	Actual Impact	Credentials and live session tokens captured for 4 employees (a fifth victim entered credentials but did not approve MFA, so no session was captured). Access to Outlook, Teams, SharePoint, and Microsoft Entra as the compromised users; 7,400+ employee email addresses extracted. Festival Finance's SOC contained the fastest case in 14 minutes (account lockout and password reset).

	Potential Impact	In the absence of SOC containment: full mailbox access, internal phishing from a trusted address, OAuth-consent abuse, mailbox-forwarding rules, lateral movement across cloud services, and large-scale internal mailings (the audit team identified internal distribution groups of 3,700+ and 7,400+ recipients, which were intentionally NOT used per engagement rules).

TECHNICAL DETAILS & PROOF OF CONCEPT

Campaign setup. A look-alike domain ("login.<attacker>.example") was registered and pointed at an Evilginx2 instance configured with the "o365" phishlet. The phishing email was crafted to mimic a Microsoft account notification (display name, layout, language). The mail-delivery weakness reported in NET-EXT-R1 increased the inbox-delivery rate (Figures 1-2).

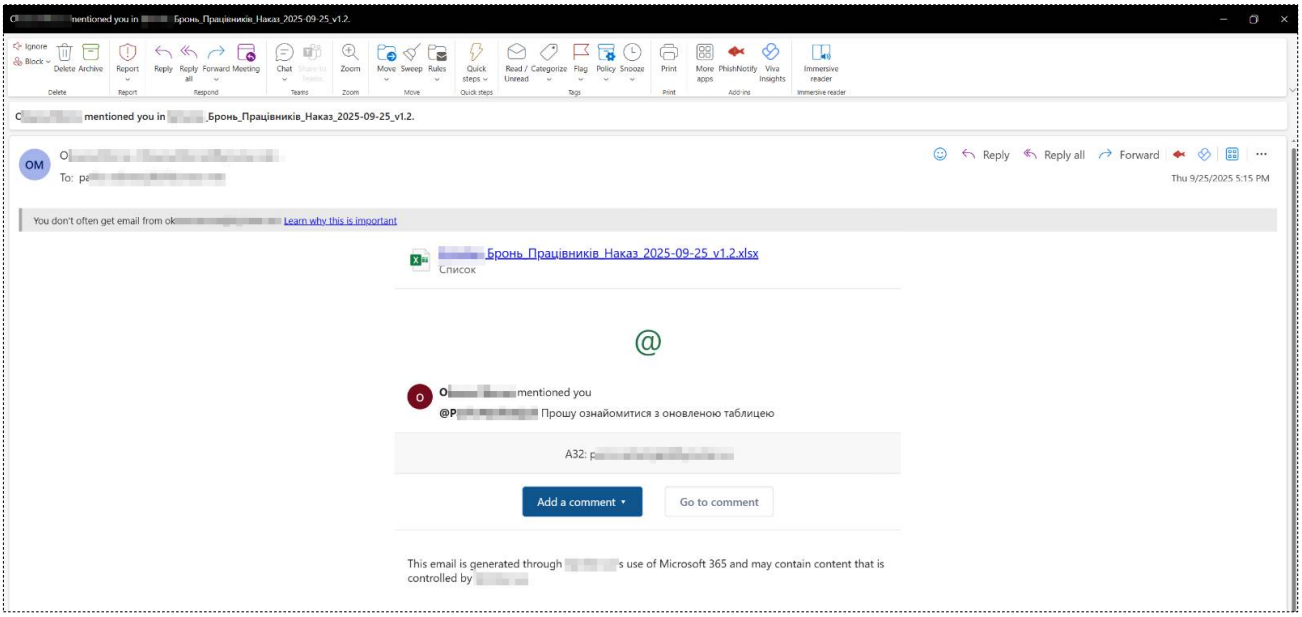


Figure 1 – Inbox view of the phishing email as received by the victim (Microsoft-branded pretext, malicious link).

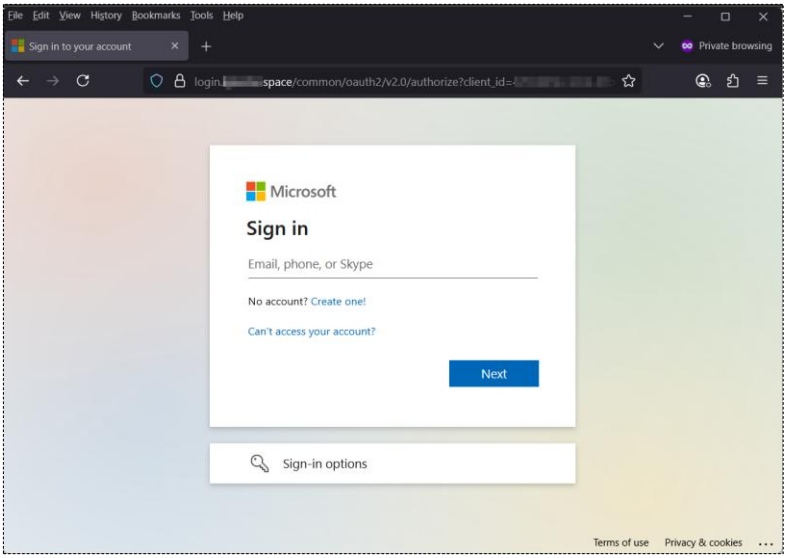


Figure 2 – Look-alike Microsoft 365 login page on the attacker-controlled domain (browser address bar shows the proxy URL).

Stage	Recipients	Total
Stage 1 - top 14 employees (executive sample)	14	14
Stage 2 - mass distribution	173	187

Attack mechanics. The Evilginx2 phishlet transparently proxies every request between the victim's browser and the genuine Microsoft 365 login flow: (Figure 3)

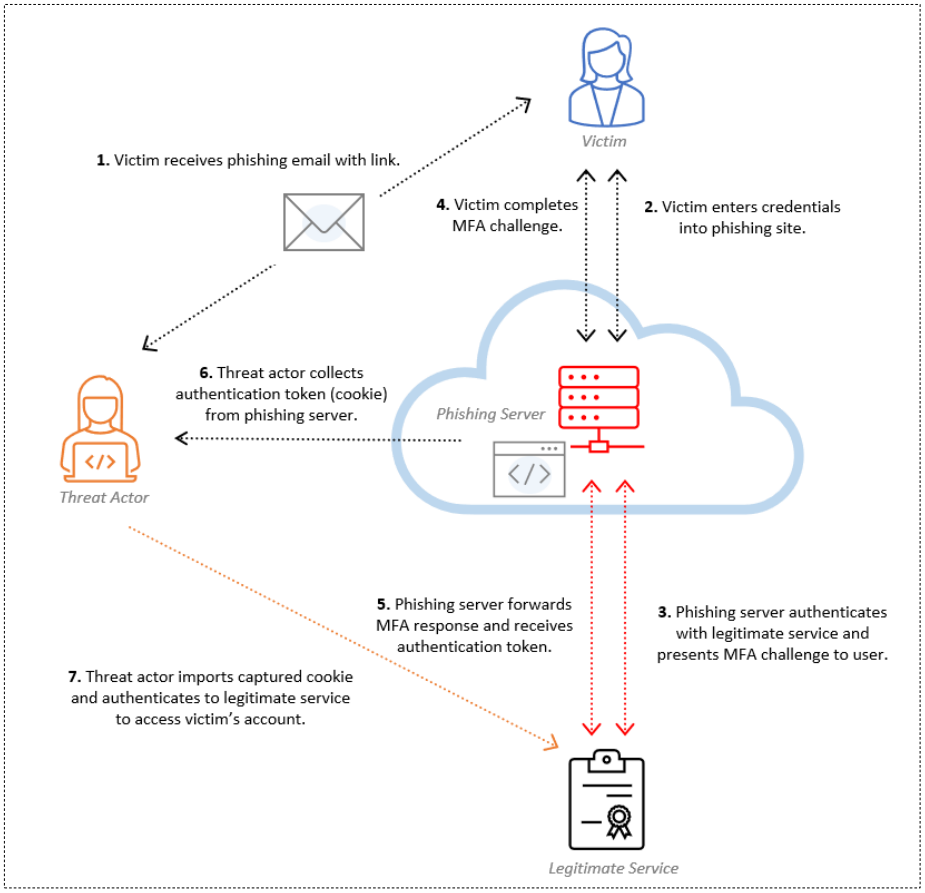


Figure 3 – Evilginx2 MITM phishing flow diagram (victim → attacker proxy → Microsoft Entra).

1. Victim enters credentials on the look-alike page ("login.<attacker>.example/<random>").
2. Evilginx relays them to "login.microsoftonline.com", which issues the MFA challenge.
3. The MFA challenge is relayed back to the victim, who approves it on their authenticator app.
4. Microsoft issues a session cookie ("ESTSAUTH", "MSPRequ", etc.) bound to the relay.
5. Evilginx captures the cookies, redirects the victim to "https://www.microsoft.com/404".
6. The attacker imports the captured cookies into a clean browser and acts as the user - **no password and no MFA prompt required**.

Campaign metrics.

Metric	Count	Rate
Phishing emails delivered	187	100.0 %
Recipients who clicked the link	64	34.2 %
Recipients who entered credentials	7	3.7 %
Recipients whose MFA approval yielded a captured session token	4	2.1 %

Post-compromise access. Using captured tokens the audit team confirmed access to: Microsoft Entra (account profile, sign-in history), Outlook (mailbox and recent emails), Teams (1:1 chats with confidential content), SharePoint (shared documents). The team also extracted >3000 employee email addresses from the global address list as material for hypothetical follow-on campaigns (Figures 4-8).

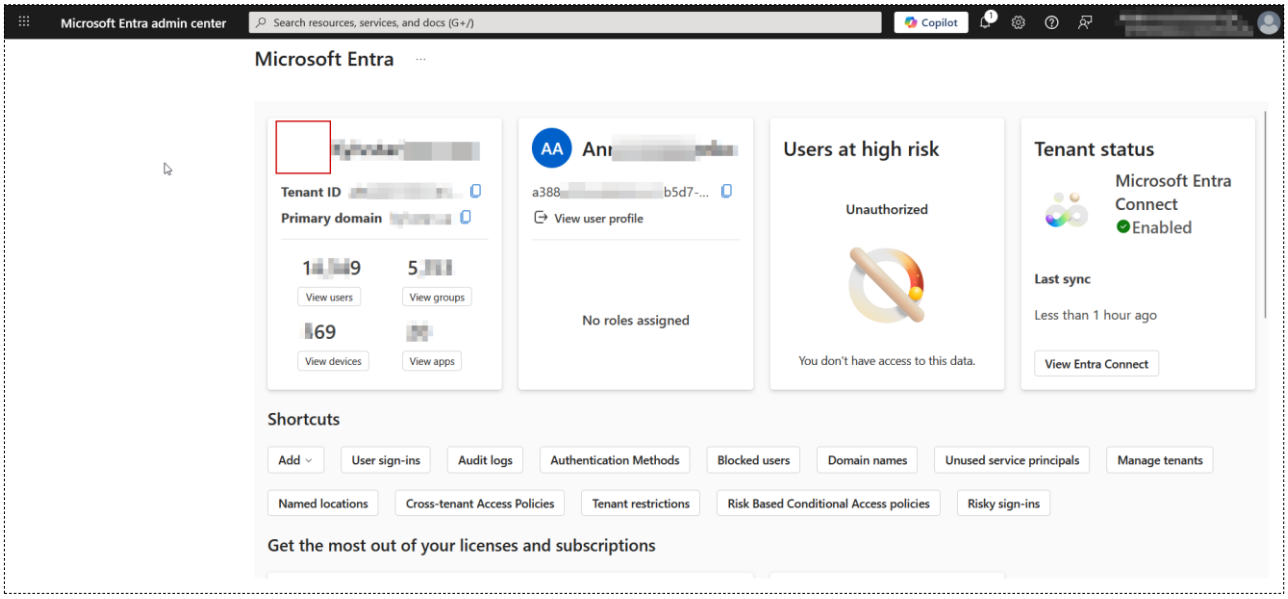


Figure 4 – Microsoft Entra portal authenticated as a compromised user.

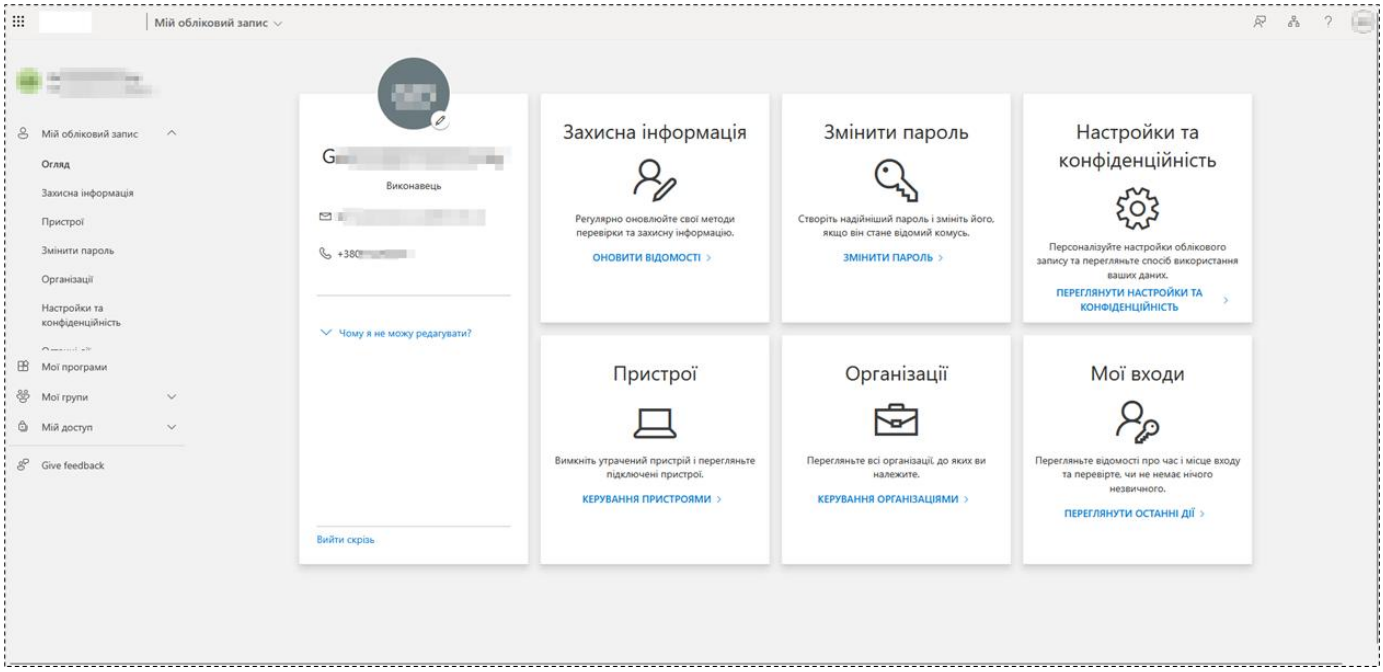


Figure 5 – Account settings page as the compromised user.

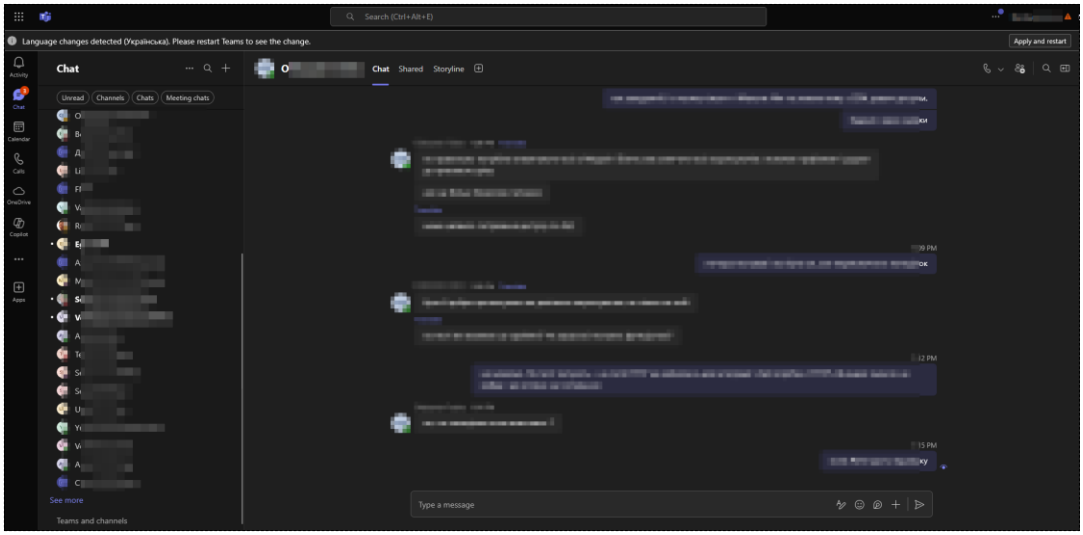


Figure 6 – Microsoft Teams 1:1 chat list visible to the audit team.

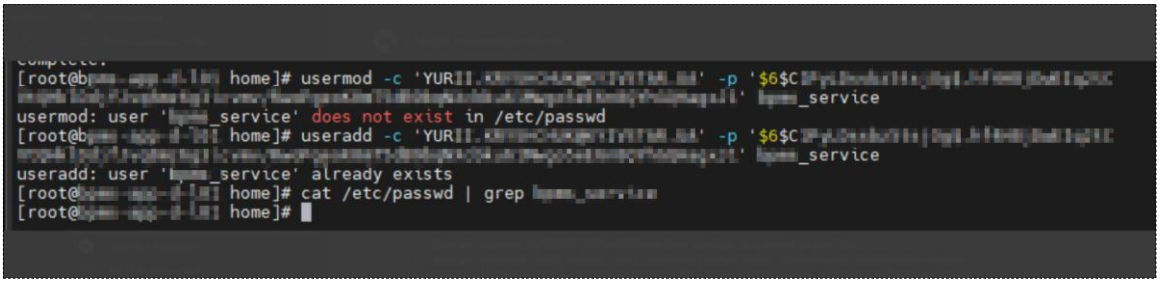


Figure 7 – Confidential conversation thread inside Microsoft Teams.

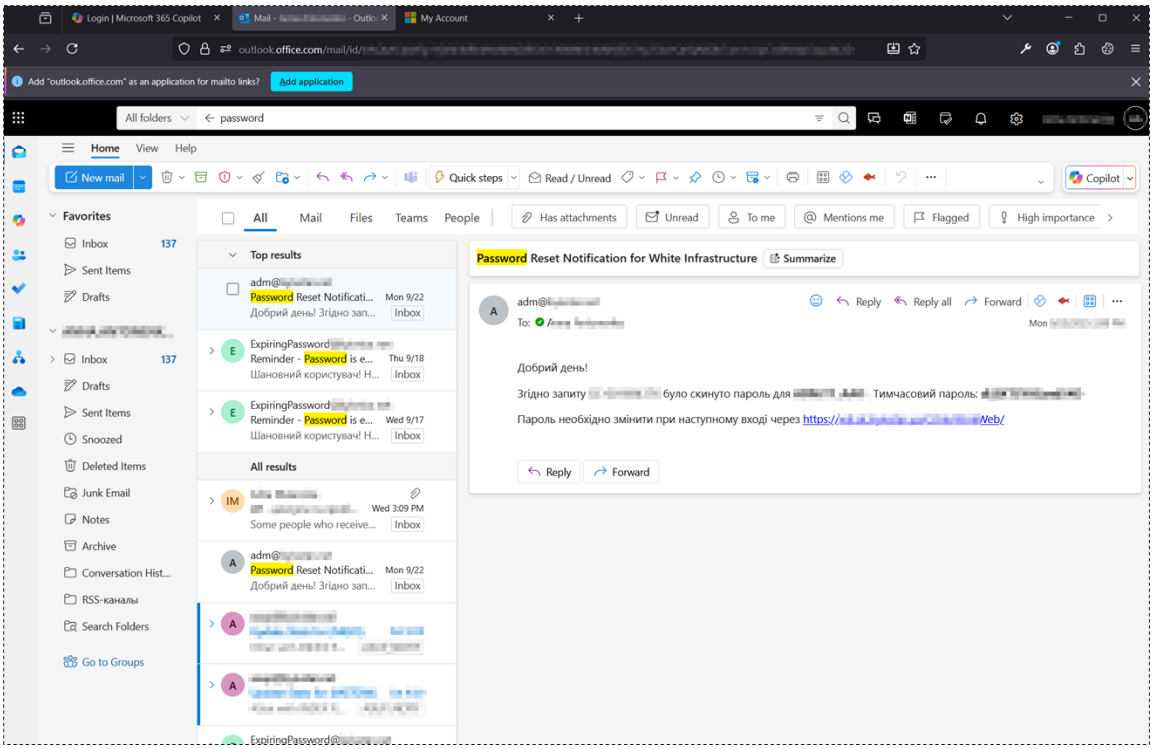


Figure 8 – Outlook mailbox view of the compromised user.

INCIDENT METRICS AND ENGAGEMENT MATURITY

SOC response and containment.

- Fastest containment: account lockout and password reset 14 minutes after compromise (compromised at 14:00 local; contained at 14:14).
- Slowest containment: still within the same business day.
- Overall: the SOC and the defensive stack demonstrated effective containment (credential revocation, URL blacklisting on the corporate proxy, mailbox-rule audit). Initial inbound filtering did allow part of the campaign to land, however - the remediation focus is on the inbound stack, not the response.

Engagement controls observed by the audit team (good practice, worth highlighting).

- Upon confirming successful account compromise, the audit team halted escalation and notified the client in accordance with the rules of engagement. No further use of harvested credentials was attempted.
- Two high-value targets were excluded from the campaign by prior agreement: the CEO and the Director of Human Resources. The audit team did not target either.
- A planned follow-on internal mailing using a compromised account - to internal distribution groups of 3,700+ and 7,400+ recipients - was cancelled by the audit team despite being technically feasible. Internal phishing-as-a-trusted-sender was intentionally NOT executed.

Remediation re-test (joint Red / Blue Team).

- On 13 September 2025, the audit team and Festival Finance's Blue Team jointly re-tested the vulnerability chain.
- Result: remediation confirmed. The phishing-as-an-internal-user scenario is no longer reproducible. Inbound filtering, DMARC enforcement, and SOC playbooks now cover the previously-exploited path.

WEB-R6 – Pre-Enrollment 2FA Hijack via Inactive Second Factor

RISK BRIEF

ID	WEB-R6
RISK LEVEL	**LOW**
SUMMARY	Microsoft 365 accounts in the MFA pilot group that have not yet enrolled their second-factor device accept enrollment of an arbitrary Authenticator app by anyone who can present the username and password. Using a credential captured during the phishing campaign (SOCIAL-R1), the audit team registered their own phone as the second factor on a pilot-group account — taking persistent control of that account's MFA channel. Follow-on lateral movement was blocked by the SOC and account permissions, capping the impact at LOW.
VULNERABLE SERVICES	Microsoft 365 / Entra ID account-enrollment workflow for accounts in the MFA-required Conditional Access policy (pilot group); vpn.festival-finance.example/saml (downstream lateral-movement target)
RECOMMENDATIONS	Require MFA enrollment within 24 hours of account provisioning, from a trusted device or location. Alert the account owner on every new MFA device via a separate channel (SMS / corporate phone). Use temporary access passes for first-time enrollment and require administrator confirmation for sensitive accounts.

RISK ASSESSMENT

	SEVERITY	DESCRIPTION	
VULNERABILITY	**HIGH**	Inactive-2FA exploitation - anyone with valid credentials and before the legitimate user enrolls can attach their own Microsoft Authenticator to the account and own the MFA channel from that point forward.	
THREAT	**LOW**	Threat Actor Profiles	Insider with Remote Access
		Attack scenario	The attacker signs in with the stolen credentials, encounters the "set up MFA" prompt, completes the enrollment with their own Authenticator app, and now possesses both the password and the MFA channel for the account.
		Conditions	The victim account belongs to the MFA pilot group (a Conditional Access policy slice that requires MFA) and has not yet enrolled their own second factor (typically a new hire or a recently moved account). The attacker must already have valid credentials.
IMPACT	**LOW**	Actual Impact	The audit team successfully enrolled their own Microsoft Authenticator to the compromised account. Subsequent attempts to use the hijacked account to authenticate to the corporate VPN (vpn.festival-finance.example/saml) were blocked - either by SOC intervention or by insufficient permissions on the victim's account. The impact was therefore contained.
		Potential Impact	If the victim's account had carried richer permissions or if SOC monitoring had been absent, the attacker could have authenticated into the corporate VPN, accessed internal infrastructure, and established persistent access.

TECHNICAL DETAILS & PROOF OF CONCEPT

The audit team had captured valid credentials for a Festival Finance user during the phishing exercise (SOCIAL-R1). When the team signed in with these credentials to the corporate identity provider, the account presented the **"set up Microsoft Authenticator"** screen - the legitimate user had not yet completed enrollment.

The team:

1. Proceeded through the standard enrollment flow on a controlled phone running Microsoft Authenticator.
2. Scanned the QR code and registered the device against the victim's identity (Figures 1-2).

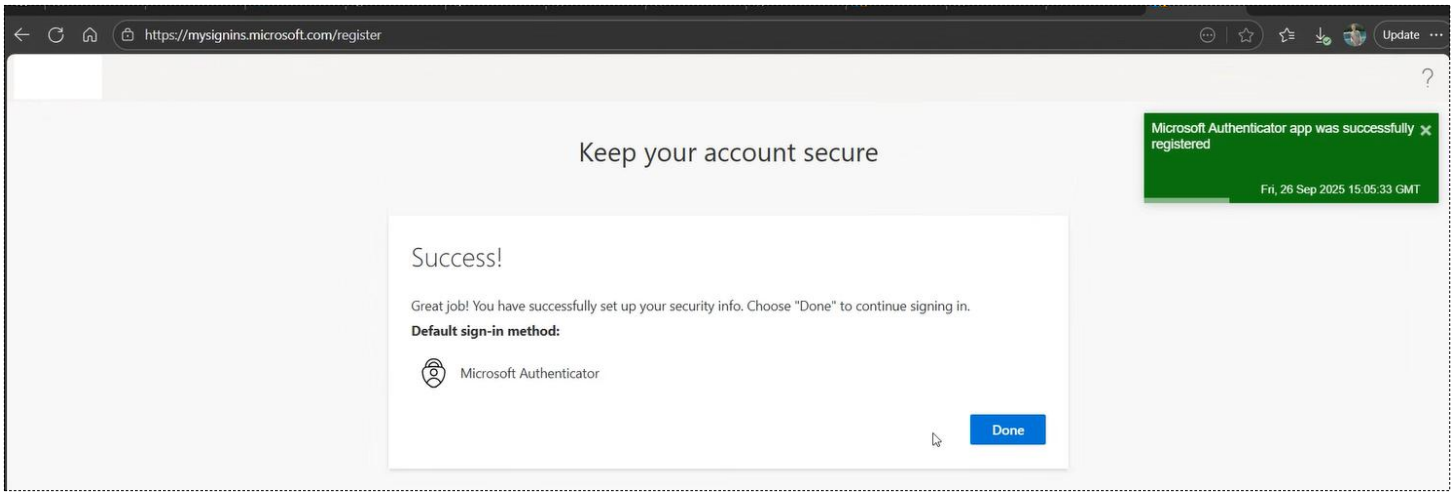


Figure 1 – Microsoft Authenticator app linked to the victim's account on the audit team's controlled device.

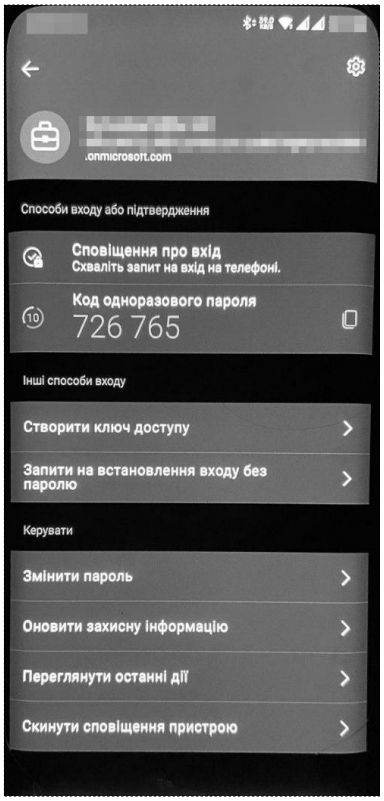


Figure 2 – Account-binding screen mid-enrollment (QR code displayed by the identity provider).

3. Completed the enrollment, receiving the "Your additional security verification is set up" confirmation (Figure 3).

Figure 3 – Confirmation page after successful 2FA setup.

From this point the audit team controlled the MFA channel: any sign-in for that user would prompt **the audit team's** Authenticator, not the legitimate user's.

The team then attempted to use the hijacked account to authenticate to the corporate VPN ("vpn.festival-finance.example/saml"). The sign-in failed at the authentication stage - the SOC had revoked the session and disabled the account in response to the phishing alarms from SOCIAL-R1, and the account's group memberships did not include VPN access. The hijacked MFA channel therefore had no useful effect on this engagement.

WEB-R8 – Missing Account-Lockout Protection on Back-Office Sign-In

RISK BRIEF

ID	WEB-R8
RISK LEVEL	**LOW**
SUMMARY	The back-office sign-in endpoint does not implement an account-lockout mechanism: there is no limit on the number of consecutive failed authentication attempts, no delay between attempts, no CAPTCHA, and no second factor. Successful authentication after an unlimited series of failures is possible. The customer-facing application (app.festival-finance.example) already enforces a 5-failures-in-10-minutes lockout - the same control is missing here.
VULNERABLE SERVICES	admin.festival-finance.example
RECOMMENDATIONS	Configure account lockout: after 5 failed sign-in attempts, lock the account for 10 minutes (mirror the existing policy on the customer-facing application). Add per-account and per-IP rate limiting on the sign-in API. Introduce CAPTCHA after 3 failed attempts. Require multi-factor authentication for back-office administrators given the privilege level of those accounts. Log and alert on bursts of failed sign-ins.

RISK ASSESSMENT

	SEVERITY	DESCRIPTION	
VULNERABILITY	**LOW**	The authentication endpoint has no rate limit, lockout, delay, CAPTCHA, or MFA - successful brute-force against weak passwords is feasible.	
THREAT	**LOW**	Threat Actor Profiles	Insider with Remote Access
		Attack scenario	The attacker scripts large-scale password guessing against the sign-in endpoint until a valid password is found.
		Conditions	The attacker must know a valid account identifier (email). Discovery is possible through OSINT or by guessing standard naming conventions.
IMPACT	**LOW**	Actual Impact	The audit team confirmed unlimited authentication attempts using only its own provided test account. No brute-force attempts were made against real employee accounts during the engagement.
		Potential Impact	If administrator passwords are predictable (a related concern reported under the Active Directory findings), this lack of lockout would let an attacker reach the back-office through brute-force in a manageable timeframe and gain administrative access to the application.

TECHNICAL DETAILS & PROOF OF CONCEPT

The audit team scripted 10 consecutive failed sign-in attempts against the back-office API followed by one successful attempt using the correct credentials. The server accepted the final attempt and issued a fresh session token without any rate-limiting, lockout, or CAPTCHA challenge between attempts (Figures 1-3).

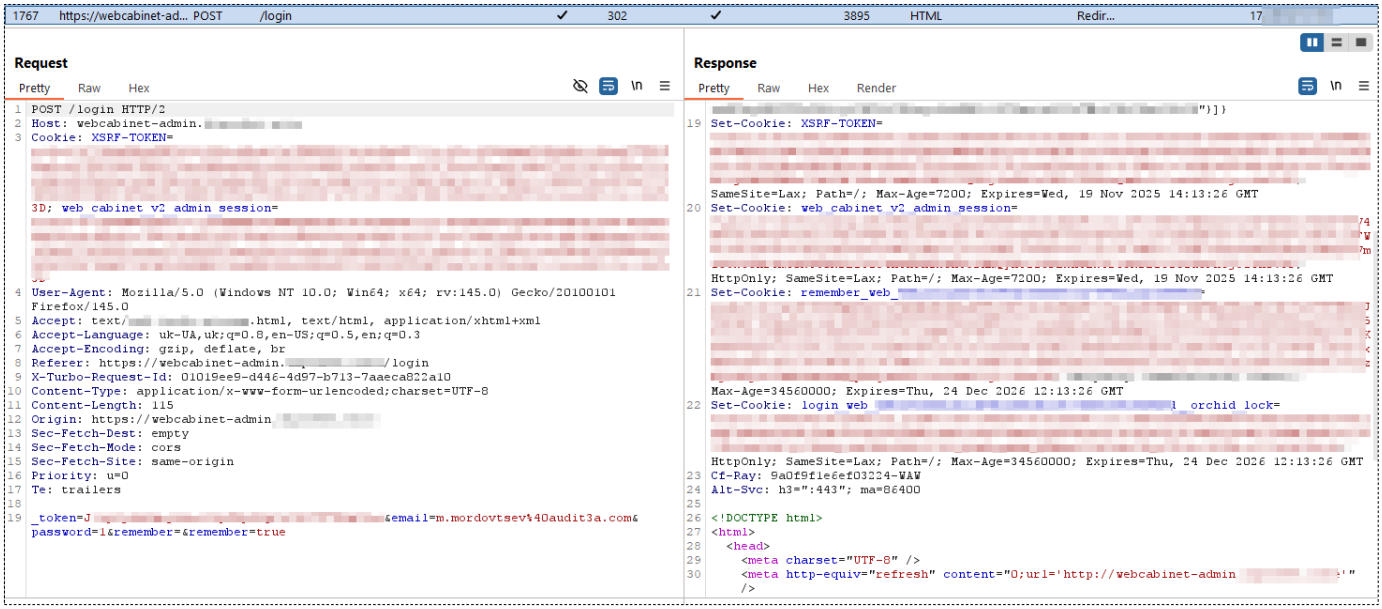


Figure 1 – Successful API authentication response, including session token issuance.

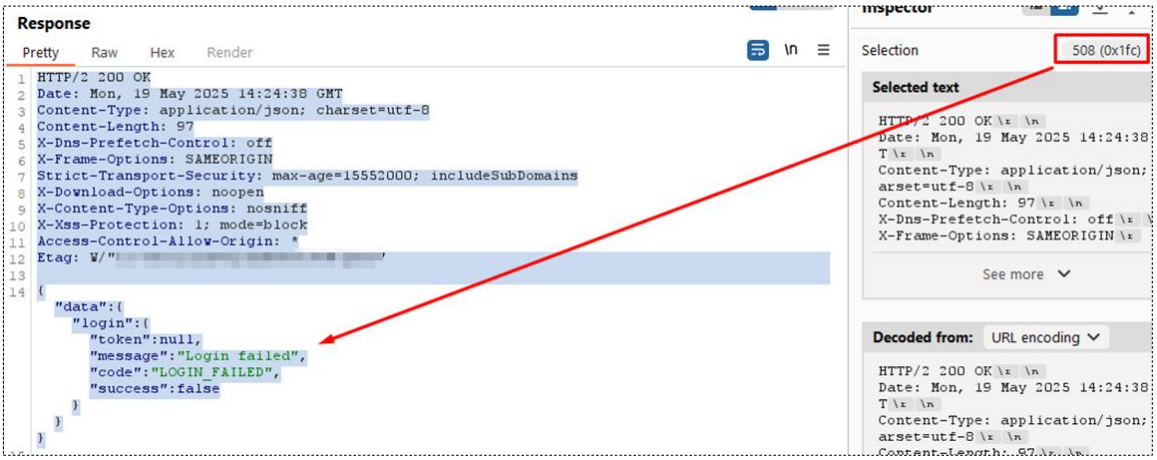


Figure 2 – Failed API authentication response - shorter body, no lockout signalled.

1817	https://webcabinet-ad...	POST	/login	✓	302	✓	2857
1818	https://webcabinet-ad...	GET	/login	✓	200		13252
1819	https://webcabinet-ad...	POST	/login	✓	302	✓	2849
1820	https://webcabinet-ad...	GET	/login	✓	200		13248
1821	https://webcabinet-ad...	POST	/login	✓	302	✓	2853
1822	https://webcabinet-ad...	GET	/login	✓	200		13250
1823	https://webcabinet-ad...	POST	/login	✓	302	✓	2853
1824	https://webcabinet-ad...	GET	/login	✓	200		13250
1825	https://webcabinet-ad...	POST	/login	✓	302	✓	2851
1826	https://webcabinet-ad...	GET	/login	✓	200		13258
1827	https://webcabinet-ad...	POST	/login	✓	302	✓	2859
1828	https://webcabinet-ad...	GET	/login	✓	200		13248
1829	https://webcabinet-ad...	POST	/login	✓	302	✓	2853
1830	https://webcabinet-ad...	GET	/login	✓	200		13252
1831	https://webcabinet-ad...	POST	/login	✓	302	✓	2849
1832	https://webcabinet-ad...	GET	/login	✓	200		13252
1833	https://webcabinet-ad...	POST	/login	✓	302	✓	2855
1834	https://webcabinet-ad...	GET	/login	✓	200		13244
1835	https://webcabinet-ad...	POST	/login	✓	302	✓	2859
1836	https://webcabinet-ad...	GET	/login	✓	200		13250
1837	https://webcabinet-ad...	POST	/login	✓	302	✓	3891

Figure 3 – Ten failed authentication attempts followed by one successful sign-in (no lockout triggered).

A request that succeeds and a request that fails differ only by response body length (3893-3895 bytes versus 2851-2859 bytes) - reliable signals for an automated brute-force loop.

WEB-R7 – Insecure User and Role Management Enables Privilege Escalation

RISK BRIEF

ID	WEB-R7
RISK LEVEL	**INFORMATIONAL**
SUMMARY	Users granted the Users or Roles permission in the back-office can create arbitrary new accounts with any privileges, delete existing administrators, and modify any role - including assigning themselves the full administrator role. The application does not enforce a tiered approval model on user / role administration.
VULNERABLE SERVICES	admin.festival-finance.example
RECOMMENDATIONS	Restrict the Users and Roles permissions to a small set of strictly governed accounts. Implement a tiered model: a user with role-management rights must not be able to grant themselves higher privileges than they currently hold. Apply four-eyes / dual-approval for sensitive permission grants. Audit log every change to user permissions and roles, and alert on self-elevation events.

RISK ASSESSMENT

	SEVERITY	DESCRIPTION	
VULNERABILITY	**HIGH**	Authorization model permits self-elevation - a user with the Roles permission can grant themselves any other permission, including full administrator.	
THREAT	**INFORMATIONAL**	Threat Actor Profiles	Insider with Remote Access
		Attack scenario	The attacker edits their own role and assigns elevated permissions, or creates a new high-privileged user under their own control.
		Conditions	The application is reachable only from whitelisted corporate IP ranges, accounts are provisioned out-of-band by an existing administrator, and the role-management permission is granted only to trusted personnel.
IMPACT	**HIGH**	Actual Impact	The audit team confirmed full privilege escalation: starting from a user limited to Users editing, they granted themselves the administrator role and gained access to every management page and dataset in the back-office.
		Potential Impact	A malicious or compromised insider could gain complete control of the back-office, including all customer data, configurations, and translations.

TECHNICAL DETAILS & PROOF OF CONCEPT

A test administrator account was provisioned with the "Users" permission only - intended to manage user accounts but not to modify role definitions.

The audit team then used that account to edit its own user entry, assigning additional permissions ("Roles", "Sites", "Translations", "Image Styles") directly through the UI. The application accepted the change without challenge or approval, and the menu immediately reflected the newly granted capabilities (Figures 1-3).

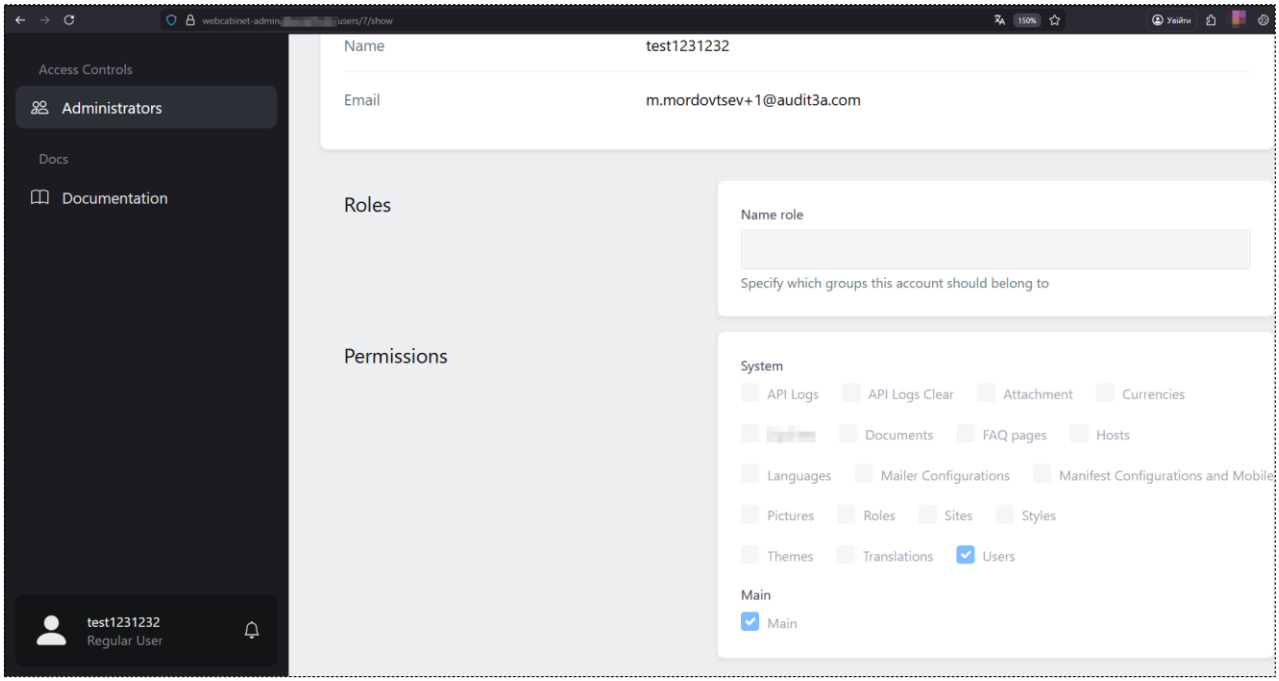


Figure 1 – Initial state: test user holds only the "Users" permission.

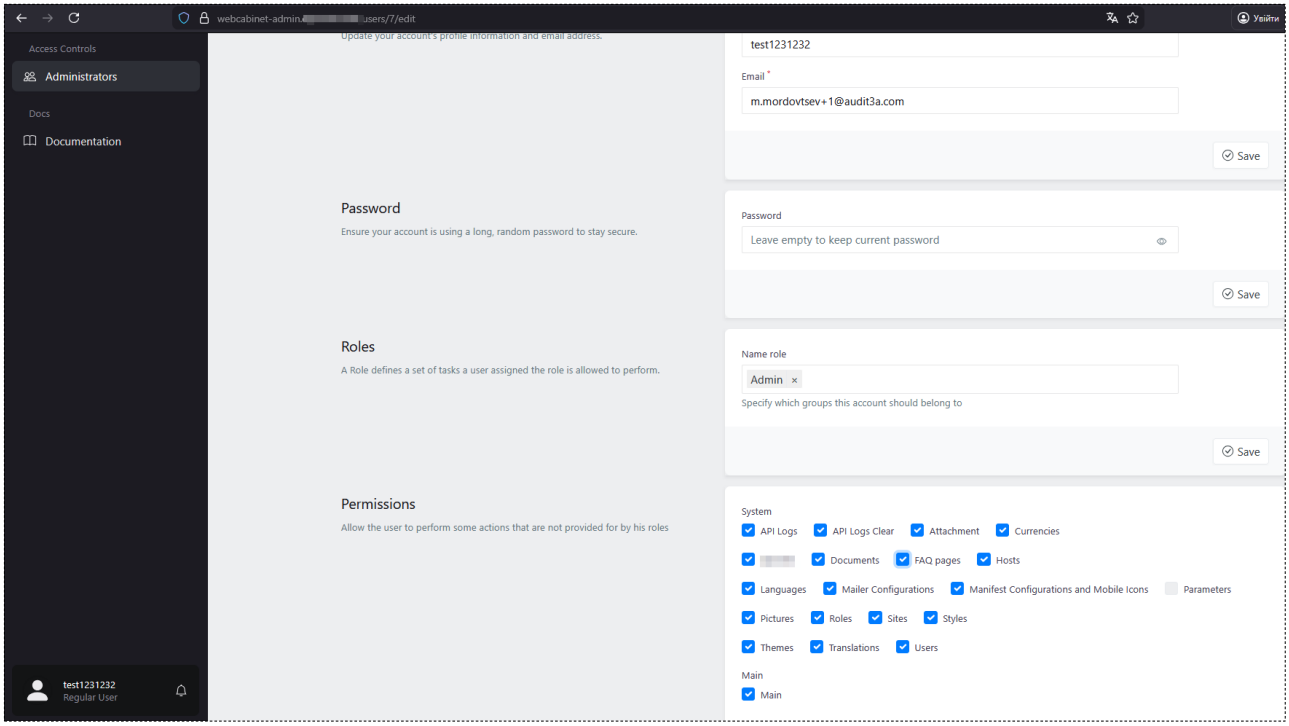


Figure 2 – Self-elevation in progress: granting additional permissions via the user-edit form.

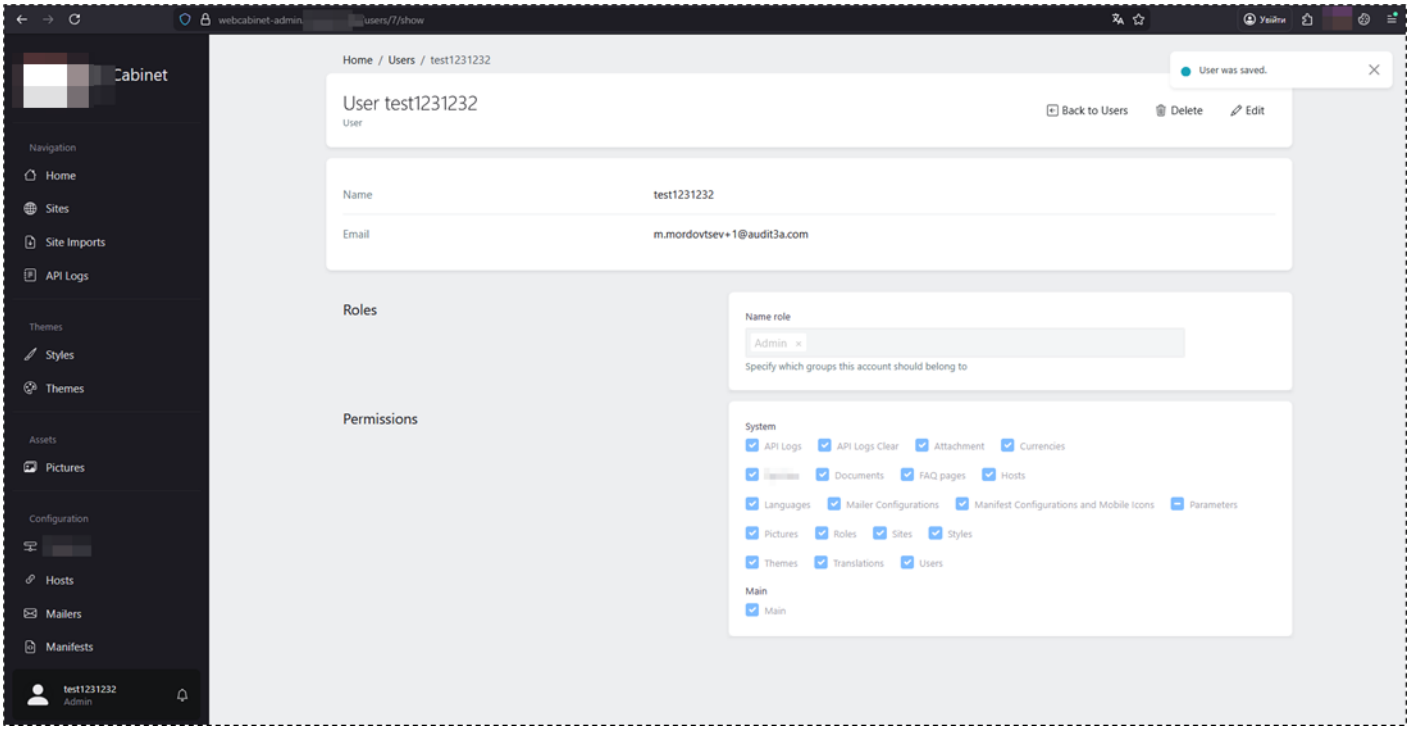


Figure 3 – Post-elevation state: full administrator menu visible; access to every management area.

The same outcome was reached by granting the "Roles" permission and then editing role definitions to extend the role's own scope - a self-elevation pattern that any user with role-management rights can perform.

WEB-R9 (MEDIUM)

Truncated...

AD-R3 (MEDIUM)

Truncated...

AD-R4 (MEDIUM)

Truncated...

WEB-R10 (LOW)

Truncated...

WEB-R11 (LOW)

Truncated...

NET-EXT-R2 (LOW)

Truncated...

AD-R5 (LOW)

Truncated...

AD-R6 (LOW)

Truncated...

AD-R7 (LOW)

Truncated...

WEB-R12 (INFO)

Truncated...

WEB-R13 (INFO)

Truncated...

NET-EXT-R3 (INFO)

Truncated...

AD-R8 (INFO)

Truncated...