TECHNICAL SUMMARY

# Phishing Simulation & Session Replay

*Authorised external security engagement • Microsoft 365 identity hardening validation*

**ENGAGEMENT OUTCOME** The client had already hardened its Microsoft 365 environment. The assessment tested whether that hardening closed the remaining identity-compromise path. It did not: the exercise demonstrated that the MFA flow in scope was prompt-based rather than phishing-resistant, and controlled session replay validated the residual risk.

| 01 TESTED | 02 PROVED | 03 DELIVERED |
|---|---|---|
| **Hardened M365 identity controls** | **Residual MFA/session exposure** | **Phishing-resistant remediation plan** |

## What I actually delivered

| | |
|---|---|
| **ENGAGEMENT** | An authorised external assessment against a client environment that had already undergone Microsoft 365 hardening. The work tested whether the security posture held under a realistic identity-focused scenario, using contained infrastructure and disposable test accounts. |
| **FINDING** | Session capture and cookie-only replay demonstrated that the MFA control in scope was present but not phishing-resistant. A test user completed the login flow against the simulated domain; the resulting authenticated session could then be reconstructed without repeating the original MFA ceremony. |
| **REMEDIATION** | Delivered recommendations centred on FIDO2/WebAuthn, Conditional Access, browser/extension warning controls and stronger domain-verification training. The goal was to remove the root weakness, not merely make the simulation harder to recognise. |

## Primary security finding

> **"The assessment showed that infrastructure hardening and conventional MFA reduced risk, but did not eliminate this attack class because the tested authentication flow was not origin-bound and therefore not phishing-resistant."**

# What held, what failed, what changed

| | |
|---|---|
| **WHAT HELD** | The client's infrastructure had already been hardened, TLS was valid, the Microsoft 365 service remained the legitimate authentication endpoint, and the engagement operated inside a signed scope with disposable accounts. |
| **WHAT FAILED** | Traditional visual trust cues and prompt-based MFA did not provide phishing resistance. A valid certificate, familiar page appearance and successful live authentication were not enough to prove that the browser was communicating with the legitimate origin. |
| **WHY IT MATTERS** | This was an identity-control finding, not a "user trick" story. It demonstrated that domain-verification training alone cannot reliably defend against a live relay scenario when the authentication method itself is transferable. |
| **WHAT CHANGED** | The remediation outcome prioritised origin-bound authentication and conditional-access controls, backed by browser warnings and improved user guidance. The engagement moved from validation to concrete defensive action. |

## Remediation delivered

- FIDO2 / WebAuthn as the primary phishing-resistant control. WebAuthn binds the credential to the legitimate origin, so an authenticator will not complete the ceremony for an unrelated proxy domain.
- Conditional Access policy improvements to reduce the value of transferable sessions and strengthen contextual access decisions.
- Browser or endpoint warning controls capable of highlighting suspicious authentication domains where appropriate.
- User training reframed around verifying the actual authentication origin, rather than relying only on branding, HTTPS or the browser padlock.

## Engineering contribution

| | |
|---|---|
| **SECURITY INFRASTRUCTURE**<br>Deployed Apache as the public TLS/reverse-proxy layer in front of the simulation backend; managed virtual hosting, certificate workflows, Cloudflare DNS integration and controlled service operation. | **CUSTOM AUTOMATION**<br>Built a Python asyncio service that monitored captured-session data, maintained persistent state, normalised session artifacts, integrated a restricted Telegram control plane and drove Playwright/Chromium replay validation. |
| **DEBUGGING**<br>Diagnosed a browser-automation freeze under Python 3.13 through systematic isolation, traced it to version incompatibility and corrected the dependency/runtime path. | **OPERATIONS & VALIDATION**<br>Implemented logging, timeout handling, replay-state tracking, controlled screenshots and layered validation so infrastructure, capture and browser automation failures could be isolated independently. |

## Technology

**Linux • Apache • TLS / PKI • Evilginx • Python asyncio • Playwright • Chromium • Telegram Bot API • Cloudflare DNS • Microsoft 365 • FIDO2 / WebAuthn • Conditional Access**

**PUBLIC-SHARING NOTE** Signed scope. Client identity and test artifacts anonymised. Disposable test accounts used; captures disposed of and credentials rotated after the engagement.