URADICAL

The Legacy VB6 Threat

Why Classic Visual Basic applications are your network's weakest link

WHITE PAPER

May 2026

uradical.io

Contents

Executive Summary

Classic Visual Basic (VB6) applications are still everywhere. The TIOBE Index for May 2026 places Classic VB at position 21 globally — ahead of Kotlin, Dart, and Lua. VBScript sits at 40. These represent real, business-critical applications running on a technology stack that hasn't received meaningful security investment in over a decade.

Most organisations running these applications don't know what they're built on. They know the function — stock management, order processing, job scheduling — not the technology. This paper explains why that matters, documents the specific attacks that have exploited these technologies, and demonstrates why the standard response of patching and firewalling cannot resolve the underlying problem.

The argument is structural, not speculative. The evidence is public, specific, and current. And the financial exposure — regulatory fines, breach costs, downtime losses — falls hardest on the SMEs most likely to be running this software.

1. The Scale of the Problem

The TIOBE Programming Community Index measures programming language popularity monthly across more than twenty platforms including Google, Amazon, and Wikipedia.[1] The May 2026 data:

Position	Language	Rating
7	Visual Basic (VB.NET)	2.90%
21	Classic Visual Basic (VB6)	0.80%
40	VBScript	0.35%

Classic VB6 at 0.80% outranks Kotlin (0.65%), Dart (0.55%), and Lua (0.51%). This is not residual noise. It represents a substantial installed base of applications built on a stack that Microsoft stopped actively developing over fifteen years ago.

The VB.NET figure at position 7 compounds the problem. Many VB.NET applications were migrated from VB6 and still rely on COM interop layers that carry the same security exposures discussed in this paper. A .NET wrapper around a COM object does not make the COM object secure — a point Google's Project Zero made definitively in 2017.[2]

2. The Technology Stack

A typical VB6 application depends on four technology layers. Each has been directly exploited in documented attacks.

COM (Component Object Model) is Microsoft's binary interface standard for software components, introduced in 1993. VB6 is inseparable from COM — every form, control, and data access operation involves COM objects registered system-wide in the Windows Registry. A compromised COM component is a compromised machine.

DCOM (Distributed COM) extends COM across network boundaries via Remote Procedure Calls on TCP port 135 and dynamically allocated high ports. Any VB6 application that connects to remote databases or shared components likely uses DCOM. The protocol assumes trust boundaries that no longer exist.

ActiveX controls are COM components designed for visual containers. VB6 applications typically depend on multiple ActiveX controls (OCX files), many from third-party vendors who ceased trading years ago. These controls execute with full host process privileges. They cannot be audited without reverse engineering. They receive no patches.

VBScript is the scripting counterpart to Visual Basic, executed by Windows Script Host. Microsoft deprecated it in 2024. It remains functional on current Windows installations and continues to be used as a malware delivery mechanism.

These are not four separate concerns. They are the same concern at different scales. A VB6 application on a corporate network has all four attack surfaces active simultaneously.

3. What This Looks Like in Practice

Consider a scenario composited from common patterns but representative of hundreds of real deployments.

A manufacturing firm in the Midlands runs a stock management application built in 2008 by a contractor who has since retired. The application tracks inventory across two warehouses, generates purchase orders, and feeds data to the accounts system. It runs on a dedicated Windows machine in the server room. Nobody has the source code. IT have a standing instruction not to update the machine.

Under the surface, the application is a VB6 executable that registers twelve COM objects and four third-party ActiveX controls at install time. It connects to a SQL Server instance using an OLE DB provider with a connection string — including credentials — stored in the Windows Registry in plaintext. It uses DCOM to communicate with a component on a second machine that handles label printing. The DCOM authentication level is set to RPC_C_AUTHN_LEVEL_NONE — no authentication, no integrity checking, no encryption. Port 135 is open. The machine is on the same subnet as the office network.

Nobody configured it this way maliciously. This is simply how VB6 applications were deployed in 2008. The problem is that the machine is still deployed this way in 2026.

How to know if you have this problem. Most organisations that have it don't know. The application is known by its function, not its technology. These indicators identify VB6/COM dependencies: MSVBVM60.DLL (the VB6 runtime) in the application directory or System32; OCX files in System32 or SysWOW64; COM registrations under HKEY_CLASSES_ROOT\CLSID referencing VB6 components; DCOM entries visible in Component Services (dcomcnfg.exe); active listeners on TCP port 135; database connections using legacy OLE DB providers (SQLNCLI, Microsoft.Jet.OLEDB). And the operational indicators that everyone recognises: a machine that IT won't update, an application that only runs on a specific Windows version, an application whose developer is gone.

4. The Attack History

The COM/DCOM/ActiveX/VBScript stack has been a direct exploitation vector for over two decades. The ILOVEYOU worm (2000) was a VBScript file that infected 50 million systems and caused an estimated $10–15 billion in damage — enabled by the fact that the VBScript runtime is present by default on every Windows installation.[3] The Blaster worm (2003) exploited a buffer overflow in the DCOM RPC interface on TCP port 135, required no user interaction to propagate, and infected over 400,000 machines. The exploit code remains in Metasploit today.[4] These are not obscure historical incidents. They established that the technology stack underlying every VB6 application is inherently exploitable.

More critically, the exploitation is ongoing. In September 2021, Microsoft disclosed CVE-2021-40444, a critical remote code execution vulnerability (CVSS 8.8) in the MSHTML browser rendering engine. Attackers exploited it by embedding malicious ActiveX controls in Microsoft Office documents. Opening the document downloaded and executed Cobalt Strike beacons — a penetration testing tool widely adopted by criminal threat actors for command-and-control, credential theft, and lateral movement.[5] ActiveX controls have been part of Windows for over thirty years; because of the privileges they run with, a malicious control can access keystrokes and sensitive system data with no additional exploitation required.[6]

CVE-2021-40444 affected every supported version of Windows from Server 2008 through Server 2019 and was used in targeted campaigns across the US, UK, Canada, Australia, and Europe. This was not a historical relic being exploited nostalgically. This was an active zero-day using the same ActiveX technology that VB6 applications depend on daily.

5. The Patching Catch-22

This is the section that matters most, because it explains why conventional IT security responses cannot resolve the VB6 problem.

5.1 DCOM Hardening

In response to CVE-2021-26414, Microsoft rolled out DCOM hardening in three phases: optional (June 2021), enabled by default with opt-out (June 2022), and permanently enabled with no opt-out (March 2023).[7]

The hardening broke legacy applications across entire industries. Rockwell Automation — one of the largest industrial control system vendors in the world — reported widespread compatibility failures. AVEVA confirmed that with hardening enabled, remote OPC server browsing stopped working entirely. Fire alarm systems, climate control systems, and PLCs were affected.[8]

VB6 applications broke too. A documented case: a VB6 client-server application using ADO and COM+/DCOM began crashing in msado15.dll after a security rollup. The developer's only option was to uninstall the security update. They described continuing to operate "with the bad feeling to be excluded from important security fixes."[9]

That developer is not unusual. They are representative.

5.2 The Fork

Every organisation running a legacy VB6/COM application now faces the same two options:

Apply the patches. The VB6 application breaks. Database writes fail. Remote COM objects become inaccessible. ActiveX controls malfunction. Business stops.

Freeze the machine. The application keeps working. The machine stops receiving security updates. Every vulnerability patched since the freeze date — not just DCOM vulnerabilities, but all of them — remains open.

Most organisations choose the second option, because the application is more visible than the risk.

5.3 The Compounding Effect

A frozen machine is not exposed to one vulnerability. It is exposed to every vulnerability patched after it was frozen:

- **EternalBlue (CVE-2017-0144)** — the SMB exploit behind WannaCry and NotPetya, which together caused over $14 billion in documented damage[10]
- **PrintNightmare (CVE-2021-34527)** — remote code execution via the Windows Print Spooler
- **Zerologon (CVE-2020-1472)** — privilege escalation via Netlogon
- Every subsequent Windows security update

The VB6 application is not the vulnerability. It is the reason the machine cannot be protected against any other vulnerability. It is an anchor, and the chain gets longer every month.

6. The Business Cost of Inaction

The technical argument is clear: legacy VB6 applications create an expanding attack surface that cannot be closed by conventional means. But the financial and regulatory argument is what should move this from the IT risk register to the board agenda.

6.1 The Regulatory Environment Has Changed

GDPR enforcement is no longer sporadic. Cumulative fines now exceed €7.1 billion, with €1.2 billion issued in 2025 alone — and over 60% of all fine value has been imposed since January 2023.[11] European data protection authorities now receive 443 breach notifications per day, a 22% year-over-year increase.[12]

In the UK specifically, the ICO issued its largest-ever fines in 2025, all for inadequate technical and organisational security measures following cyberattacks. Capita was fined £14 million after a breach affecting 6.6 million individuals. The ICO found that inadequate penetration testing, insufficient security operations centre staffing, and poor access controls created a "foreseeable and avoidable risk which was exploited by the threat actor." The ICO was unequivocal: "being a victim of a sophisticated attack is not a defence where foundational controls, timely response and risk-based assurance were lacking."[13]

Advanced Computer Software Group was fined £3.07 million after a ransomware attack exploited an account lacking multi-factor authentication. The ICO identified deficiencies including insufficient vulnerability scanning and inadequate patch management — the exact security posture of a frozen VB6 machine.[14] Information Commissioner John Edwards was direct: "With cyber incidents increasing across all sectors, my decision today is a stark reminder that organisations risk becoming the next target without robust security measures in place. There is no excuse for leaving any part of your system vulnerable."[14]

Two points are critical for any organisation running legacy VB6 applications. First, the ICO did not accept that implementing security measures could be costly and time-consuming as an explanation for shortcomings. Second, the maximum fine under PECR rose from £500,000 to £17.5 million or 4% of global turnover in 2026. The idea that small businesses sit outside UK GDPR is, as the ICO has stated, a persistent myth.[15]

6.2 Breach Costs Fall Hardest on SMEs

The organisations most likely to be running legacy VB6 applications — mid-market manufacturers, regional service firms, independent healthcare providers — are the least equipped to absorb a breach.

The average cost of a data breach for UK SMEs reached a record £75,000 per incident in 2025, driven by regulatory fines, reputational damage, and customer loss.[16] The average cost of a cyberattack rose to £6,400 in 2025, up 52% from the previous year.[17] For context, the global average across all organisation sizes is $4.44 million (approximately £3.5 million).[18]

These figures mask the existential risk. 28% of UK SMEs say a single attack could put them out of business. 32% have no cybersecurity protections in place at all.[19] Among SMEs that have been attacked, 100% reported having to close temporarily, with average losses of nearly £31,000 for each day of closure.[20]

A single security breach is often a business-ending event for smaller firms. Large corporations have cash reserves to weather a month of downtime or a heavy fine. Most UK SMEs operate on much thinner margins.

6.3 The Scale is Enormous

The UK Government's Cyber Security Breaches Survey 2025/2026, published April 2026, found that 43% of UK businesses experienced a cyber breach or attack in the last 12 months — approximately 612,000 businesses. Micro firms (42%) and small firms (46%) were affected at rates close to the overall average.[21]

UK SMEs are incurring aggregate annual losses of £3.4 billion due to inadequate cybersecurity measures.[19] Cybercrime cost the UK economy an estimated £14.7 billion in 2025.[22] The problem is so pervasive that even the UK government cannot manage its own legacy estate: a National Audit Office report in January 2025 identified 228 legacy IT systems across government departments, 28% of which were "red rated" — meaning a high likelihood of operational and security risks occurring.[24]

6.4 What This Means for Legacy VB6

Return to the manufacturing firm from section 3. Their frozen VB6 machine has: no security patches since the DCOM hardening broke the application; DCOM authentication set to NONE; plaintext credentials in the registry; port 135 open on the corporate LAN; ActiveX controls from a vendor that no longer exists.

If that machine is compromised — and the attack surface makes it a question of when, not if — the organisation faces:

- An average breach cost of £75,000, potentially higher given the plaintext credentials and the data flows to the accounts system.
- An ICO investigation that will identify inadequate patch management, absent access controls, and plaintext credential storage — the exact deficiencies the ICO fined Capita and Advanced for.

- Regulatory fines of up to £17.5 million or 4% of global turnover.
- Average downtime losses of £31,000 per day of closure while systems are rebuilt.
- A 28% probability, per UK government data, that the business does not survive.

The modernisation project that replaces the VB6 application costs a fraction of one breach. The maths is not complicated.

7. The Remediation Path

Patching doesn't fix this. Firewalling mitigates it. Network segmentation buys time. The only thing that resolves it is replacing the application.

This is not just our assessment. The NCSC's CTO Ollie Whitehouse stated in May 2026: "Patching alone will not always suffice; some technical debt may be present in 'end of life' or legacy technology that is out of support, and so can't receive updates. In such instances, organisations will need to replace technologies, or bring them back within support."[23]

This is the direct consequence of the structural analysis in sections 5 and 6: as long as the application depends on COM, DCOM, and ActiveX, the host machine cannot be fully secured. The remediation must eliminate that dependency entirely.

7.1 Discovery and Audit

The first phase maps the complete dependency surface: every registered COM object, every ActiveX control, every DCOM configuration, every network connection, every stored credential, and every data flow. Most organisations have never done this. The audit itself is valuable — it tells you exactly what your application is doing at the system and network level, which is almost certainly more than you think.

This is forensic work, not box-ticking. It requires understanding COM registration, DCOM security descriptors, OLE DB connection semantics, and the VB6 runtime's interaction with the Windows security model. It produces a complete threat surface map and a data flow diagram that becomes the specification for the replacement.

Return to the manufacturing firm. The audit discovers: twelve registered COM objects, four of which reference OCX files from a vendor that was acquired in 2012 and whose website no longer exists. A DCOM configuration with authentication set to NONE. Plaintext SQL Server credentials in three registry locations. An undocumented nightly scheduled task that exports inventory data to a CSV on a shared network drive readable by the entire domain. A Windows Firewall rule created in 2009 that allows all inbound traffic on ports 135–139.

None of this was visible before the audit. All of it is exploitable. And all of it would be cited in an ICO investigation.

7.2 Architecture

The replacement is designed around principles that are the direct inverse of VB6/COM's failure modes.

Single-binary deployment. No runtime dependencies, no DLL registration, no COM objects, no ActiveX controls. One executable that runs where you put it. We build in Go because Go compiles to a statically-linked binary with no external dependencies — the direct opposite of the COM registration model that creates the problem. The binary is auditable, reproducible, and deployable without touching the Windows Registry.

Standard library first. The VB6 ecosystem's reliance on third-party OCX controls from defunct vendors is a core failure mode — you cannot patch a dependency from a company that no longer exists. Go's standard library handles HTTP, TLS, cryptography, SQL database access, JSON, and file I/O without third-party packages. Every external dependency in the replacement is chosen deliberately and auditable. The total dependency count for a typical replacement application is in single figures, not the dozens of registered COM objects and OCX files that characterise a VB6 deployment.

Data sovereignty by architecture. The VB6 application stores credentials in the registry in plaintext because that's what the tooling made easy in 2008. The replacement uses proper credential management — environment variables, encrypted configuration, or platform-native secret stores. The VB6 application scatters data across COM objects, shared drives, and embedded database references that nobody can fully map without forensic analysis. The replacement uses a single, auditable data store — SQLite for local state, PostgreSQL for networked access — with documented schemas and migration paths. Data stays on your infrastructure, under your control, in a format you can inspect and export.

Patchable infrastructure. This is the point of the entire exercise. Once the application no longer depends on COM, DCOM, or ActiveX, the host machine can receive Windows security updates normally. The anchor is removed. The machine rejoins your security baseline. The compounding vulnerability exposure documented in section 5.3 stops accumulating. And the next time the ICO asks about your patch management, you have an answer.

7.3 Migration

Implementation maintains business continuity throughout. The replacement is developed against the specification produced by the audit. Parallel running validates behavioural equivalence before cutover. Data migration is planned with full rollback capability. When the replacement is validated, the legacy application is decommissioned, its COM/DCOM registrations are removed, and the host machine is brought up to current patch levels.

The manufacturing firm's stock system becomes a single Go binary serving a web interface on localhost. Warehouse staff access it through a browser instead of a VB6 form. The SQL Server connection uses TLS with credentials in an encrypted

configuration file. The label printer communicates over a standard network protocol instead of DCOM. The nightly export runs as a documented, auditable scheduled task with appropriate file permissions. The Windows machine receives security updates on the same schedule as every other machine in the organisation.

The result is an application that any competent developer can read, build, and deploy. The source is version-controlled. The build is reproducible. The organisation is no longer dependent on a single person, a single machine, or abandoned technology.

8. Conclusion

Classic Visual Basic applications are a documented, active, and compounding security liability. The technology they depend on has been the direct exploitation vector for attacks causing tens of billions of dollars in damage. Microsoft's own efforts to harden these technologies break the applications that depend on them. The result is machines frozen in time on corporate networks, accumulating vulnerability exposure that grows every month.

The ICO is actively fining organisations for the exact security deficiencies that characterise legacy VB6 deployments: inadequate patch management, absent access controls, insufficient vulnerability scanning. The average breach costs a UK SME £75,000. For 28% of them, a single attack could end the business.

The modernisation project costs less than one breach. The question is not whether to modernise, but how much exposure to accumulate before doing so.

Notes

1. TIOBE Index for May 2026, https://www.tiobe.com/tiobe-index/
2. Google Project Zero, "Exploiting .NET Managed DCOM," April 2017
3. Kaspersky, "Evolution of Security: The Story of the ILOVEYOU Worm," 2022
4. Microsoft Security Bulletin MS03-026; Rapid7 Metasploit Module exploit/windows/dcerpc/ms03_026_dcom
5. Microsoft Security Blog, "Analyzing attacks that exploit CVE-2021-40444," September 2021
6. Zscaler, "ActiveX Vulnerabilities — Threat To Web Security"
7. Microsoft KB5004442, "Manage changes for Windows DCOM Server Security Feature Bypass (CVE-2021-26414)"
8. TXOne Networks, "Microsoft DCOM Hardening Patch — What You Need to Know"
9. MCB Systems, "April 2017 Monthly Rollup Breaks VB6 App," 2017
10. Infosecurity Europe, "What Have We Learned from NotPetya Six Years On," 2023
11. Kiteworks, "GDPR Fines Hit €7.1 Billion: Data Privacy Enforcement Trends in 2026," March 2026
12. DLA Piper, "GDPR Fines and Data Breach Survey: January 2026"
13. ICO Monetary Penalty Notice, Capita plc, October 2025; Skadden, "Recent ICO Data Breach Enforcement," February 2026; Ropes & Gray, "UK's ICO issues a £14 Million Fine for Poor Data Security," January 2026
14. ICO Monetary Penalty Notice, Advanced Computer Software Group Ltd, March 2025; Computer Weekly, "Advanced Software fined £3m over LockBit attack," March 2025
15. Naq Cyber, "GDPR Small Business Compliance: The 2026 UK Guide"
16. Forensic Control, "Data breach costs for UK SMEs reach record high," 2025; UK Government Cyber Security Breaches Survey 2025
17. AMVIA, "UK SME Cybersecurity Report 2026," January–February 2026
18. IBM, "Cost of a Data Breach Report 2025"
19. Cyber Sec Stats, "UK Cybersecurity Statistics 2026"
20. Sky Business, "SMEs miscalculate the cost of cyber attacks on their business"
21. UK Government, "Cyber Security Breaches Survey 2025/2026," DSIT & Home Office, April 2026
22. PolicyBee, "The average cost of a data breach for small businesses," March 2026
23. NCSC, "Preparing for a vulnerability patch wave," Ollie Whitehouse, May 2026
24. National Audit Office, Legacy IT systems in government departments, January 2025