Secure Systems Design and Technology Development. Veliko Tarnovo, Bulgaria
Proceedings of the International Scientific Conference, 2026. Volume III, 285-289

Automated Vulnerability Assessment in Windows Active Directory and Oracle Database Environments using PowerShell and SQL*Plus

Vasil Slavyanov
PhD Student
Vasil Levski National Military University
Veliko Tarnovo, Bulgaria
vasil.slavyanov@gmail.com

***Abstract*—Enterprise IT environments are rarely built around a single technology. In most mid-to-large organisations, identity management runs through Microsoft Active Directory while business-critical data sits in relational databases such as Oracle. This combination is operationally essential and a recurring focus of security incidents. Reviewing such environments manually is slow, inconsistent across reviewers, and leaves real gaps in coverage. This paper describes a practical approach to automating vulnerability assessment across both layers using tools already present in the environment: PowerShell for the Windows and Active Directory side, and SQL*Plus for Oracle Database. The focus is on configuration weaknesses, access control gaps, and governance failures that accumulate quietly in production systems. The paper presents a methodology, discussion of typical findings, and an honest account of where the approach runs into its limits.**

Keywords— Active Directory, configuration auditing, internal control testing, Oracle Database, PowerShell, security automation, SQL*Plus, vulnerability assessment

I. Introduction

Security problems in enterprise environments tend not to arrive through the front door. They accumulate — in stale accounts, overpermissioned service users, password policies that were never enforced, audit settings that were configured once and never revisited. The gap between what an organisation's policy documents say and what is actually running in production is often wider than anyone realises, and it is rarely visible without looking directly at the technical configuration.

Active Directory and Oracle Database together represent a substantial portion of the attack surface in most enterprise environments. Active Directory controls who can authenticate, what they can access, and which policies apply to which systems. Oracle Database holds the data that organisations are most likely to be contractually and legally obligated to protect. Weaknesses in either layer carry serious consequences; weaknesses in both, which is common, compound each other.

The conventional approach to reviewing these environments relies heavily on the knowledge and diligence of whoever conducts the review. A reviewer who knows what to look for in Active Directory may know considerably less about Oracle, and vice versa. Even where competence is not an issue, manual review is slow, and a slow process tends to get compressed or skipped when audit timelines tighten.

Automation addresses the most persistent part of the problem: whether the data was actually collected, consistently, in a form that can be reviewed. PowerShell and SQL*Plus are available in any domain-joined Windows environment and any Oracle installation. Using them systematically produces a complete, comparable, timestamped picture of what the environment actually looks like.

Section 2 establishes the problem context; Section 3 describes the methodology. The Active Directory and Oracle assessment layers are treated in Sections 4 and 5 respectively. Section 6 characterises typical findings, Section 7 examines limitations, Section 8 addresses practical application within audit and compliance contexts, and Section 9 concludes

II. Problem Statement and Motivation

This paper addresses a specific kind of problem: not zero-days or sophisticated exploitation techniques, but the mundane security failures that cause the most damage in enterprise environments. A domain administrator account belonging to an employee who left three years ago. A service account with DBA rights that nobody remembers creating. A password policy applied to ninety percent of users but not the accounts that actually matter. These are

Online ISSN 3044-7771
https://doi.org/10.68302/std2026.vol3.146

configuration and governance failures, and they persist because nobody looks for them systematically.

Active Directory is a particularly productive source of this kind of problem. Because it underpins authentication and policy enforcement across the entire domain, a misconfiguration does not stay local — it propagates. Mokhtar et al. [7] point out that the breadth of adoption of Active Directory services has made them a consistent target, and that attack techniques have evolved specifically to exploit misconfigurations in group memberships, delegation settings, and authentication configurations.

The Oracle Database side of the problem is similar in origin. Database environments tend to accumulate excessive privilege over time as application requirements change, developers request access and move on, and DBAs apply workarounds that remain in place long after the original problem is resolved. Omotunde and Ahmed [8] identify improper access control enforcement as among the primary contributors to database security incidents in contemporary organisations.

Three practical requirements follow. First, assessment needs to be repeatable. Second, the evidence needs to be defensible — findings based on structured script output tied to specific configuration objects are not easily challenged. Third, the process needs to be efficient enough to actually happen within normal audit planning constraints.

III. Methodology

The methodology is built around four principles that reflect deliberate choices about what kind of assessment this is.

3.1 Assessment Principles

The first principle is to use only what is already there. PowerShell and SQL*Plus are available in any environment where Windows and Oracle are deployed. Scripts based on native capabilities avoid deployment friction and questions about authorisation.

The second principle is read-only operation. Nothing in the assessment modifies the target environment. This is non-negotiable for any review conducted in a production context, and it is what separates a configuration assessment from a penetration test.

The third principle is control orientation. The scripts are structured to answer specific questions: is this setting within the range the policy requires? Does this account have attributes that should have triggered a review? Are these group memberships consistent with the documented access model?

The fourth principle is structured output. Results are exported in formats — CSV, JSON, or timestamped text — that can be reviewed by a human, processed analytically, and stored as audit evidence.

3.2 Assessment Stages

Stage 1: Scope Definition

Before any script runs, the reviewer defines what is in scope: which domains, which domain controllers, which member servers, which Oracle instances. Scope decisions determine what the assessment can and cannot conclude, and an ambiguous scope is a common reason why findings get disputed after the fact.

Stage 2: Automated Data Collection

Scripts are executed against the defined scope to extract configuration data. For Active Directory and Windows, this covers user and group objects, Group Policy settings, local administrator membership, installed roles, open ports, running services, event log configuration, and registry-based security parameters. For Oracle, it covers user account status, role and privilege assignments, profile settings, password controls, and audit configuration.

Stage 3: Control Evaluation

The collected data is reviewed against control expectations. Some evaluations can be automated — accounts inactive for more than ninety days, passwords set to never expire, users with DBA role. Others require human judgement — whether a particular group membership is justified, whether a deviation from password policy is covered by a documented exception.

Stage 4: Reporting and Follow-Up

Findings are documented with a description of what was observed, a statement of why it matters, the specific evidence that supports it, and a recommendation. Technical outputs are attached as appendices. The whole package supports both management reporting and remediation tracking.

3.3 Evaluation Dimensions

Across both layers, assessment logic is organised around the same dimensions: identity and access management; privilege management; authentication and password controls; configuration hardening; logging and monitoring; audit readiness; and policy consistency across systems.

IV. Automated Assessment of Windows Active Directory and Server Environments

PowerShell's usefulness comes from the breadth of what it can reach. A single script, run from a domain-joined machine with appropriate read permissions, can enumerate user and group objects across the entire directory, pull Group Policy Object settings, query local configuration on individual servers, and export the results to structured files — all without installing anything, without modifying anything, and without requiring administrator cooperation.

4.1 Data Sources and Collection Areas

4.1.1 User Accounts

User account data is where most Active Directory assessments find their first substantive problems. The relevant attributes are enabled status, password last set date, last logon timestamp, whether the account has a non-expiring password, and whether it carries any administrative group membership. From these, it is possible to identify accounts not used in months but remaining enabled, accounts whose passwords have not changed in years, and service accounts that have

accumulated group memberships beyond what any documented role would justify.

4.1.2 Group Membership and Privilege Exposure

Privileged group membership in Active Directory — Domain Admins, Enterprise Admins, Account Operators, Backup Operators — deserves more scrutiny than it typically receives. Bu Haimed et al. [9] show experimentally that even modest initial access rights can be escalated to full domain administrative control when group and delegation configurations contain the kinds of misconfiguration that automated review is well placed to detect.

4.1.3 Organisational Units and Delegation

Delegation settings in the organisational unit structure are a less commonly reviewed area but can carry significant risk. Permissions delegated at the OU level for helpdesk workflows, password resets, or computer account management can, if misconfigured, provide pathways to privilege that bypass the standard administrative groups entirely.

4.1.4 Group Policy Objects

Group Policy is where the security baseline is supposed to live. The relevant settings — password and lockout policy, audit configuration, user rights assignment, local administrator control, remote access restrictions — are the documented intent of whoever designed the security architecture. The automated review extracts the actual GPO configuration and checks it against that intent. In practice, the two are frequently inconsistent.

4.1.5 Local Server Configuration

The local Administrators group on individual servers is particularly worth examining: it is not uncommon to find it containing individual user accounts added during incidents, test accounts never removed, or service accounts whose presence was never intended. Beyond group membership, the review covers running services, listening ports, firewall configuration, and event log settings.

4.2 Sample Assessment Logic

The following checks represent a practical starting set: identifying enabled accounts inactive beyond a defined threshold; listing accounts with non-expiring passwords; extracting and resolving nested membership of all privileged groups; comparing the effective domain password policy against documented requirements; identifying services running under privileged domain accounts; and verifying that audit policy configuration is centrally managed and complete.

4.3 Evidence and Output Structure

Each script output should carry at minimum a timestamp, a scope identifier, and a reference to the module or query that produced it. CSV is usually the most practical format for Active Directory exports. The discipline of saving outputs systematically — rather than reviewing them interactively and discarding them — is what makes it possible to produce defensible findings rather than recollections.

V. Automated Assessment of Oracle Database Environments Using SQL*Plus

Oracle Database assessment through SQL*Plus works on the same principle: use what is already there, read only, compare against expected controls. The data dictionary views exposed through SQL*Plus provide access to everything a security reviewer needs — user account status, privilege and role assignments, profile settings, audit configuration — without requiring additional tooling and without leaving a footprint beyond the query itself in the audit log.

5.1 Security-Relevant Review Areas

5.1.1 Database Users and Account Status

The user inventory is the natural starting point. Which accounts are open, which are locked, which have never logged in recently, which are default Oracle accounts not secured at installation, and which application or service accounts lack an identifiable owner. An account with DBA privileges attached to an application decommissioned two years ago represents a real exposure, not a theoretical one.

5.1.2 Roles and Privileges

Privilege accumulation in Oracle environments follows a recognisable pattern. The DBA role gets assigned during a project because someone needed broad access temporarily and is never removed. A developer gets GRANT ANY PRIVILEGE during testing and the account is kept for support purposes. In aggregate, these describe a database where the privilege model provides very little actual constraint.

5.1.3 Profiles and Password Controls

Oracle profiles control password lifetime, reuse restrictions, failed login limits, and session timeout behaviour. The DEFAULT profile often retains its out-of-box settings — UNLIMITED password lifetime, no lockout after failed attempts, no verification function — long after it should have been hardened. Checking which accounts are on which profile, and what those profiles actually specify, is a straightforward query that frequently produces findings of immediate practical significance.

5.1.4 Audit and Logging Configuration

Database auditing is where Oracle assessments most consistently find problems. The question is not merely whether auditing is enabled — it is whether it captures the actions that would matter in an incident investigation: privileged user activity, DDL on sensitive objects, logon failures, modifications to audit configuration itself. An audit trail that records SELECT on a lookup table but not GRANT ANY PRIVILEGE is not providing meaningful assurance.

5.1.5 Configuration Review and Exposure Indicators

Beyond users and privileges, configuration parameters and structural features can increase exposure. Public synonyms that expose internal objects to any authenticated user, external procedure configurations, and initialisation parameters that weaken access enforcement are all worth checking.

5.2 Assessment Questions and Query Logic

The SQL*Plus queries are structured around specific questions rather than generic data dumps: which users have system privileges granted directly rather than through roles? Which accounts are in OPEN status but have not authenticated in the last ninety days? Which profiles have no failed login limit or no password verification function? Framing the queries this way produces output that maps directly to findings.

5.3 Assessment Value in the Audit Context

The practical benefit in audit terms is independence. When a finding is based on SQL*Plus output that the reviewer ran themselves against the production database, it is not dependent on what an administrator chose to show during a demonstration. It can be re-run to verify that remediation was applied. It can be compared against last year's output to show that a problem is not new.

VI. Typical Findings and Reporting Model

The findings that emerge from this kind of assessment are not surprising in type, but their prevalence usually is. Organisations that believe their access governance is broadly in order tend to discover, when they look at the actual configuration data, that the gap between policy and practice is substantially larger than expected.

6.1 Common Findings in Active Directory and Windows Environments

- Privileged groups contain accounts that cannot be justified against current role assignments, often because the access was granted for a specific purpose and never removed.
- Accounts remain enabled well past any reasonable inactivity threshold, including accounts belonging to former employees.
- The effective password policy, once GPO inheritance is resolved, applies weaker controls to certain account types than the documented policy specifies.
- Fine-grained password policies, where they exist, are applied inconsistently or to groups that do not reflect current account classification.
- Local Administrators group membership on servers has drifted from any documented baseline and is not periodically reviewed.
- Audit policy configuration is incomplete, often because it was set at domain level but overridden by local policy on critical servers.
- Service accounts carry privileges inherited from earlier configurations and have no documented owner accountable for reviewing them.

6.2 Common Findings in Oracle Database Environments

- System privileges are assigned directly to users rather than through roles, in quantities that have no relationship to current operational requirements.
- Application and service accounts lack an identifiable owner and in some cases are associated with applications that no longer exist.
- The DEFAULT profile applies to accounts that should be on a more restrictive profile, and the DEFAULT profile itself retains weak or unlimited settings.
- DBA role assignments are broader than current operational needs.
- Audit configuration does not capture the actions needed to reconstruct a security incident.
- Privilege combinations exist that effectively bypass separation of duties in ways not reflected in any risk register or compensating control documentation.

6.3 Reporting Structure

The output from this assessment is structured as four components. An executive summary provides an overall picture of risk exposure, key themes across findings, and priority actions. A findings register documents each observation with a description of what was found, the risk it represents, evidence that supports it, and a specific recommendation. Technical appendices attach the structured script and query outputs. A remediation tracker assigns ownership, target dates, and a mechanism for follow-up verification.

VII. Benefits and Limitations

7.1 Benefits

The most significant benefit is that the assessment actually gets done, consistently, across the full scope, every time it runs. Manual reviews tend to compress under time pressure, focus on systems where the reviewer has the most knowledge, and produce varying output depending on who conducts them. Scripts do not have these properties.

The cost profile is also worth noting. Once scripts are built and the reporting structure established, the marginal cost of running an assessment is low — lower than the equivalent manual review by a considerable margin — and the output is more complete and consistent.

7.2 Limitations

The most fundamental limitation is that this approach finds what it looks for. A script that checks password policy settings will find password policy problems. It will not find a social engineering attack in progress or a compromised credential being used within normal parameters. This is a configuration assessment, not threat detection.

The second significant limitation is interpretation. Automated extraction produces data; it does not produce judgement. Whether a particular group membership is justified, whether a deviation from password policy is

covered by a documented exception — these questions cannot be answered by a script.

There is also the maintenance question. Environments change, and scripts that are not updated will eventually produce incorrect or misleading output. Scripts should be treated as assets that require the same kind of lifecycle management as the controls they are designed to assess.

VIII. Practical Application in Internal Audit and Compliance

Internal audit is the context where this approach delivers the most immediate practical value. Audit teams working in regulated environments are expected to provide assurance over access governance, privilege management, and configuration controls — often across multiple systems, within a constrained timeline, with findings that will be scrutinised by external reviewers and potentially by regulators.

Embedding automated extraction into the audit workflow changes the nature of the audit engagement. When an auditor arrives at an interview already holding structured data about the actual configuration, the conversation is different. The system owner cannot simply describe how the controls work in principle; they have to account for what the data shows. Findings based on script output are harder to dismiss and easier to track through to closure.

Beyond internal audit, the same methodology supports compliance work more broadly — regulatory reviews, third-party assessments, and continuous control monitoring programmes. The evidence requirements differ across these contexts but the underlying need — structured, traceable, reproducible technical data — is consistent.

IX. Conclusion

The argument of this paper is straightforward: configuration and governance weaknesses in Active Directory and Oracle Database environments are common, consequential, and largely invisible without systematic technical review. Manual review is insufficient to provide that visibility at the scale and consistency that regulated environments require. Automation using PowerShell and SQL*Plus can provide it, at a cost substantially lower than the equivalent manual effort and with output that is more useful as audit evidence.

The methodology described here is not a replacement for professional judgement or for the broader suite of security testing tools that a mature security programme would employ. It is a foundation: a systematic way of ensuring that the basic configuration data is collected, consistently, in a form that supports both operational decision-making and audit assurance. In environments where that foundation does not currently exist, establishing it produces immediate and tangible improvement in security visibility.

Future development might productively focus on automated comparison of results against documented baselines, scoring mechanisms that weight findings by organisational context rather than just technical severity, and integration with continuous monitoring infrastructure.

References

[1] Microsoft Corporation. Windows Server and Active Directory documentation. Microsoft Docs. https://docs.microsoft.com

[2] Oracle Corporation. Oracle Database Security Guide. Oracle Documentation. https://docs.oracle.com/en/database/oracle/oracle-database/

[3] Center for Internet Security (CIS). CIS Benchmarks for Microsoft Windows Server and Oracle Database. https://www.cisecurity.org

[4] ISACA. IT Audit Framework (ITAF): A Professional Practices Framework for IT Audit, 4th Edition. ISACA, 2020.

[5] NIST. Special Publication 800-53: Security and Privacy Controls for Information Systems and Organizations. National Institute of Standards and Technology, 2020. https://doi.org/10.6028/NIST.SP.800-53r5

[6] The Institute of Internal Auditors. International Standards for the Professional Practice of Internal Auditing. IIA, 2017.

[7] Mokhtar, B. I., Jurcut, A. D., ElSayed, M. S., & Azer, M. A. (2022). Active Directory Attacks: Steps, Types, and Signatures. Electronics, 11(16), 2629. https://doi.org/10.3390/electronics11162629

[8] Omotunde, H., & Ahmed, M. (2023). A Comprehensive Review of Security Measures in Database Systems. Mesopotamian Journal of CyberSecurity, 2023, 115-133. https://doi.org/10.58496/MJCSC/2023/016

[9] Bu Haimed, I., Albahar, M., & Alzubaidi, A. (2023). Exploiting Misconfiguration Vulnerabilities in Microsoft's Azure Active Directory for Privilege Escalation Attacks. Future Internet, 15(7), 226. https://doi.org/10.3390/fi15070226