Rocket® software

Mainframe vs. Network Penetration Testing

Why traditional infrastructure testing is not enough to assess business-critical mainframe risk.

Contents

Executive summary

Network penetration testing and mainframe penetration testing share the same broad objective: identify exploitable weaknesses before an attacker does. However, they differ significantly in scope, skill requirements, attack paths, tools, evidence, and business impact. Network penetration testing typically evaluates the exposure of distributed infrastructure such as servers, endpoints, firewalls, directories, web services, cloud connections, and network protocols. Mainframe penetration testing focuses on the security of z/OS® or related mainframe environments, including external security manager configuration, privileged access, datasets, batch processing, 3270 access, JES2, IBM® CICS®, Db2®, UNIX® System Services, and logical partitions.

The most important difference is that network testing often answers, “Can an attacker reach or compromise systems on the network?” Mainframe testing answers a higher-stakes question: “Can an attacker gain unauthorized access to the enterprise systems of record, elevate privileges, manipulate production data, submit jobs, or bypass controls that protect regulated business processes?” For organizations that rely on mainframes for financial transactions, claims processing, logistics, payments, customer records, or regulatory reporting, mainframe testing should be treated as a distinct and specialized discipline, not a subset of standard network testing.

Why the distinction matters

Mainframes are often deeply integrated with an organization’s most critical and regulated business processes. They may not have the same visible attack surface as modern distributed systems, but they frequently hold the data, authorization logic, and transaction processing that attackers ultimately want to reach. A network test may discover open services, weak perimeter controls, exposed protocols, or misconfigured segmentation. A mainframe test goes further by evaluating whether access to the platform can be abused once a user, service account, application, or network path reaches the mainframe environment.

Core differences at a glance:

Dimension	Network penetration testing	Mainframe penetration testing
Primary focus	Distributed infrastructure, connectivity, services, segmentation, perimeter exposure, and exploitable system vulnerabilities.	z/OS and mainframe subsystems, identity and authorization controls, datasets, jobs, privileged functions, and transaction environments.
Typical assets	Servers, routers, firewalls, endpoints, directories, cloud gateways, VPNs, APIs, web applications, and exposed ports.	LPARs, Sysplexes, RACF®, ACF2, Top Secret, TSO, JES2, CICS, Db2, VTAM®, TN3270, USS, JCL, and sensitive datasets.
Attack model	External or internal attacker attempts to discover and exploit reachable hosts, services, credentials, and trust paths.	Adversary attempts to gain mainframe access, escalate privileges, manipulate jobs or datasets, abuse security manager rules, or reach high-value transaction systems.
Specialized expertise	TCP/IP networking, operating systems, identity systems, cloud infrastructure, vulnerability exploitation, and web technologies.	z/OS architecture, external security manager logic, mainframe commands, batch processing, dataset permissions, subsystem security, and audit evidence.
Tools and techniques	Network scanners, vulnerability platforms, packet analysis, password testing, exploit frameworks, web application testing, segmentation validation, and lateral movement analysis.	Mainframe-native security reviews, RACF/ACF2/Top Secret rule analysis, dataset validation, TSO and JES2 testing, CICS and Db2 transaction review, APF library analysis, job submission testing, and audit log validation.
Common findings	Missing patches, weak passwords, exposed services, insecure protocols, misconfigured firewalls, privilege escalation paths, and weak segmentation.	Overprivileged IDs, weak dataset protections, excessive SPECIAL or OPERATIONS authority, insecure APF libraries, JES2 or TSO weaknesses, CICS transaction exposure, and insufficient logging.
Business impact	Compromise of infrastructure, lateral movement, data exposure, service disruption, or entry into critical environments.	Unauthorized access to systems of record, fraudulent transactions, production data manipulation, audit failures, regulatory exposure, and operational disruption.

How network penetration testing works

Network penetration testing evaluates how an attacker could discover, access, exploit, and move through an organization's distributed technology environment. It usually begins with scoping and rules of engagement, followed by reconnaissance, port scanning, service enumeration, vulnerability analysis, controlled exploitation, post-exploitation analysis, and reporting. Typical test activities include validating exposed services, checking protocol security, testing segmentation, attempting credential attacks within agreed limits, validating patch levels, reviewing firewall rules, and identifying paths that could lead to sensitive systems.

How mainframe penetration testing works

Mainframe penetration testing evaluates whether technical, administrative, and operational controls adequately protect the mainframe environment from unauthorized access, privilege escalation, data exposure, or manipulation of critical business processes. Testing may begin by identifying mainframe-facing services such as TN3270, FTP, SSH, web interfaces, APIs, or network job entry connections. From there, the test may move into authenticated or presumed-breach scenarios to assess what a user, application account, or compromised credential could do inside the environment.

Key testing scenarios:

Testing scenario	Testing objective	Example focus areas
External exposure	Identify whether mainframe services are visible from untrusted or semi-trusted networks.	TN3270, FTP, SSH, web services, APIs, and network job entry.
Credential and access abuse	Test whether weak passwords, shared IDs, excessive access, or poorly governed service accounts could provide unauthorized entry.	Password policy, shared IDs, service accounts, orphaned IDs, excessive permissions, and authentication controls.
Privilege escalation	Assess whether users can gain elevated control through privileged attributes or misconfigured security rules.	SPECIAL, OPERATIONS, surrogate access, APF-authorized libraries, started tasks, and security manager definitions.
Dataset and file access	Determine whether sensitive datasets, USS files, configuration libraries, source code, or credentials are insufficiently protected.	Production datasets, PARMLIB, PROCLIB, source libraries, USS directories, credential stores, and configuration files.
Batch and job control abuse	Evaluate whether JCL, JES2, scheduling, or job submission paths could be manipulated to execute unauthorized actions.	JCL libraries, JES2 controls, scheduler definitions, job submission authority, and production execution paths.
Transaction and application abuse	Test mainframe application environments for unauthorized transactions, weak region configuration, or excessive access.	CICS, Db2, IBM® IMS, transaction access, region configuration, application permissions, and database privileges.
Audit and detection gaps	Verify whether security events are logged, monitored, and actionable enough to detect mainframe-specific attack behavior.	SMF records, ESM logging, alerting rules, privileged activity monitoring, failed access attempts, and security operations workflows.

Methodology comparison

Both testing disciplines should follow a structured approach: define scope, establish rules of engagement, gather information, analyze vulnerabilities, validate exploitability, assess impact, and report remediation priorities. The difference is in the evidence collected and the expertise required. Network testing evidence often includes host inventories, open ports, screenshots, vulnerability data, exploit validation, and lateral movement paths. Mainframe testing evidence may include security manager rule analysis, dataset access proof, job submission results, transaction access validation, privileged attribute review, subsystem configuration issues, and audit trail gaps.

A mature security testing program should treat the two assessments as complementary. Network testing helps determine whether attackers can reach mainframe-connected services or move laterally toward protected environments. Mainframe testing determines what damage could occur if unauthorized access is obtained and whether mainframe-native controls would prevent, detect, or limit that activity.

Regulatory alignment: Why mainframe penetration testing matters

Regulatory frameworks increasingly emphasize risk-based security testing, operational resilience, vulnerability management, and evidence that critical systems are protected in practice. For organizations that rely on mainframes to process regulated transactions or store sensitive data, mainframe penetration testing helps extend compliance validation beyond the perimeter and into the systems of record where business impact is highest.

Framework	Relevant expectation	How mainframe penetration testing supports alignment	Evidence to capture
DORA	Requires ICT risk management, digital operational resilience testing, detection, response, recovery, and advanced threat-led penetration testing for certain financial entities.	Validates whether critical mainframe services, transaction systems, privileged access paths, and operational controls can withstand realistic attack scenarios that could disrupt regulated financial services.	Test scope, threat scenarios, findings, exploit validation, resilience impact, remediation plan, retest results, and management sign-off.
PCI DSS 4.x	Requires defined penetration testing methodology, internal and external testing, testing after significant change, segmentation validation, remediation, and retesting for environments that store, process, transmit, or affect cardholder data security.	Determines whether mainframe-connected applications, payment processing paths, privileged accounts, datasets, and segmentation assumptions could expose or affect cardholder data environments.	Documented methodology, internal and external test results, CDE boundary validation, segmentation test evidence, remediation records, and retest confirmation.
NYDFS 23 NYCRR 500	Requires a risk-based cybersecurity program, annual penetration testing from inside and outside information system boundaries, vulnerability management, timely remediation, incident response, and annual certification of compliance.	Provides evidence that mainframe information systems and non-public information are tested against realistic compromise scenarios, including internal misuse, privileged access abuse, and unauthorized access to regulated data.	Risk assessment linkage, inside/outside test scope, vulnerability findings, remediation prioritization, retest documentation, and compliance certification support.
FFIEC	Emphasizes risk identification, cybersecurity maturity, controls, threat intelligence, external dependency management, incident management, resilience, and examination-ready evidence for financial institutions.	Supports examiner expectations by validating the effectiveness of controls protecting core banking platforms, high-risk connection types, third-party access paths, privileged IDs, and critical infrastructure dependencies.	Control test results, inherent risk linkage, maturity improvement actions, third-party access evidence, incident-detection gaps, remediation tracking, and board-ready reporting.

Rocket's point of view: From point-in-time testing to continuous assurance

Rocket's POV is that mainframe penetration testing is essential, but it should not stand alone. A penetration test provides valuable point-in-time validation of exploitable risk; continuous assurance extends that value by helping organizations identify, validate, prioritize, remediate, and demonstrate that IBM Z® risk is under control over time. This is especially important as mainframes become more connected through APIs, hybrid architectures, AI-enabled workflows, third-party access, and modernization initiatives.

Rocket® z/Assurance brings purpose-built IBM Z software, expert-led security services, and mainframe-specific risk insight together in one assurance model. The portfolio helps security, risk, compliance, and technology leaders move from assumed protection to evidence-backed assurance by discovering vulnerabilities and control gaps, validating policy adherence, prioritizing remediation, and producing clearer evidence for auditors, regulators, and executives.

- **Purpose-built for IBM Z:** Rocket focuses on the distinct security architecture, operating model, and control evidence requirements of mainframe environments rather than treating IBM Z as just another endpoint on the network.
- **Penetration testing plus assurance:** Rocket mainframe security services can validate exploitable paths, while the broader z/Assurance portfolio helps maintain visibility across vulnerabilities, patch exposure, compliance posture, cryptographic risk, and emerging exposures.
- **Evidence that leaders can defend:** Rocket helps translate technical findings into business-relevant risk insight, audit-ready evidence, and remediation priorities that CISOs, CIOs, risk leaders, and boards can use.
- **Continuous proof of control:** Rocket's assurance model supports a shift from periodic assessments and manual audit preparation toward ongoing control validation, executive reporting, and demonstrable progress.
- **Secure modernization:** Rocket helps organizations modernize IBM Z with confidence by ensuring security, compliance, and governance advance alongside AI, API, cloud, and hybrid integration initiatives.

The practical takeaway is simple: network penetration testing can reveal how attackers may reach critical systems, and mainframe penetration testing can show what could happen once IBM Z is in scope. Rocket z/Assurance helps organizations go further by turning those findings into a continuous, defensible operating model for mainframe risk management, compliance readiness, operational resilience, and secure modernization.

Best-practice recommendations

- **Do not assume network testing covers the mainframe.** Include mainframe-native scope, skills, and evidence requirements in the testing plan.
- **Use specialist testers.** Require demonstrated expertise in z/OS, RACF, ACF2, Top Secret, JES2, CICS, Db2, USS, and mainframe operations.
- **Include authenticated and presumed-breach testing.** Many meaningful mainframe risks emerge only after a user or service account has access.
- **Map findings to business processes.** Tie technical findings to transaction integrity, data confidentiality, operational resilience, compliance obligations, and audit evidence.
- **Coordinate with operations teams.** Mainframe testing must be carefully planned to avoid disrupting production workloads, batch windows, or critical transaction processing.
- **Retest after remediation.** Validate that access changes, rule updates, monitoring improvements, and configuration fixes have reduced risk.

Conclusion

Network penetration testing remains essential for understanding exposure across modern infrastructure, but it is not sufficient to assess the full risk of enterprise mainframe environments. Mainframe penetration testing requires a specialized lens because the platform's security model, operational patterns, privileged controls, and business impact are fundamentally different. Organizations that rely on mainframes should include dedicated mainframe penetration testing as part of their broader assurance strategy to validate that critical systems of record remain protected against both external compromise and insider or presumed-breach scenarios.

About Rocket Software

Rocket Software is a global technology leader in modernization and a partner of choice that empowers the world's leading businesses on their modernization journeys, spanning core systems to the cloud. Trusted by over 12,500 customers and 750 partners, and with more than 3,200 global employees, Rocket Software enables customers to maximize their data, applications, and infrastructure to deliver critical services that power our modern world. Rocket Software is a privately held U.S. corporation headquartered in the Boston area with centers of excellence strategically located throughout North America, Europe, Asia and Australia. Rocket Software is a portfolio company of Bain Capital Private Equity.

Rocket® software

Modernization.
Without Disruption.™

Visit RocketSoftware.com

MAR-19197_WP_BrandPenTesting_V2