Iron Net: Proactive Remediation of Network-Level Lateral Movement Attack Paths in Windows Active Directory Environments

CS6727 Final Report
April 15, 2026

Joshua Arnold
Georgia Institute of Technology
jarnold62@gatech.edu

***Abstract*—Lateral movement remains a key stage in the attack lifecycle and allows for adversaries to bury deep into compromised networks and achieve their primary objectives. When leveraged and properly configured, host-level firewalls provide an opportunity to limit lateral movement, increase the chance of detecting adversary actions, and prevent an attacker from further expanding their access. Unfortunately, host-level firewalls are commonly an afterthought, misconfigured, and policies controlling firewall implementation can be lost in the shuffle leading to reduced firewall effectiveness.**

This paper provides a semi-automated, human-in-the-loop, system designed to enumerate the presence of network-level lateral movement attack paths in Active Directory, analyze root cause, and recommend and execute approved remediations. By integrating with Active Directory, continuously monitoring for firewall policy issues, and scanning for open ports related to common lateral movement port communication, the system identifies root cause of lateral movement port communication success in the environment. Remediations linked to an enumerated root cause allow for the system to recommend an appropriate remediation to a user for action to mitigate the issue and related network-level attack paths.

Keywords – Active Directory, firewall policy, Group Policy remediation, lateral movement, attack path analysis, network security automation

AI Disclosure: The author used Claude (claude-sonnet-4-6) [21] as a writing assistance tool and development aid during the preparation of this manuscript and the implementation of Iron Net.

I. Introduction

Enterprise networks face a persistent and well-documented threat: adversaries who, having established an initial foothold, move laterally through the network by exploiting host-to-host communication channels to escalate privileges and reach high-value assets. Despite broad awareness of this threat, organizations continue to struggle with a fundamental protection and enforcement problem: the network-level host firewall rules intended to restrict lateral movement ports. These rules are frequently absent, misconfigured, or silently failing to reach the hosts they are meant to protect. National security advisories consistently identify firewall misconfiguration and insufficient network segmentation as among the most exploited weaknesses in enterprise environments [1], and post-breach after action reports from real-world incidents confirm that inadequate host firewall enforcement directly enabled adversarial lateral movement at scale [6][7][8].

In Windows Active Directory environments, host-based firewall policy is defined and delivered through Group Policy Objects (GPO) [18]. This delivery mechanism introduces a class of failure modes: blocked inheritance, disabled GPO links, missing security filtering, WMI filter exclusions, and misconfigured firewall rules. These failures could silently prevent policy from reaching hosts without producing any visible error to administrators [18][19]. Existing tools address adjacent problems: network vulnerability scanners assess exposed services, AD security platforms identify credential-based attack paths, and firewall policy analyzers detect rule-level conflicts within delivered configurations. However, none of these tools automate the full chain from observed host port state, to GPO policy intent, to OU inheritance root cause analysis, to targeted remediation through the AD management plane.

This paper presents Iron Net, a platform that fills this gap. Iron Net ingests authoritative GPO configurations, correlates observed network port state against policy intent on a per-host basis, traces enforcement failures to their root cause in the GPO inheritance hierarchy, and executes or recommends targeted remediation through the AD management plane. A lightweight validation agent deployed to monitored hosts enables active port scanning, ‘gpupdate’ delivery, and post-remediation port confirmation, closing the loop between AD-level policy change and host-level enforcement verification. Iron Net was evaluated against a ten-phase framework in a purpose-built Active Directory lab environment, achieving a composite weighted score of 82 out of 100 and demonstrating end-to-end detection, remediation, and verification of GPO-driven firewall enforcement failures across sixteen misconfiguration categories.

The remainder of this paper is organized as follows. Section II provides background on Active Directory

infrastructure, Group Policy processing, lateral movement as an attack surface, and the limitations of existing approaches. Section III describes Iron Net's system architecture and design decisions. Section IV presents the implementation including GPO misconfiguration detection, attack path computation, the port violation pipeline, the decision engine, the GPO remediator, and the verification chain. Section V describes the evaluation methodology and results. Section VI discusses the findings, practical applicability, limitations, and future work. Section VII provides the conclusion and Section VIII provides acknowledgements.

II. BACKGROUND AND RELATED WORK

A. Active Directory and Group Policy Infrastructure

In an Active Directory (AD) environment, domains contain components such as organizational units (OUs), users, and computers, known as objects [17]. A forest is comprised of at least one domain and forms the outermost structure and trust boundary between other forests and domains [17]. A forest may be comprised of multiple domains while a singular domain constitutes a forest. At its core, a domain is comprised of a hierarchy of containers and a special type of container known as an Organization Unit (OU) [17]. Together, containers and OUs store all objects that exist within a domain. OUs, however, allow for the grouping and collection of similar objects in which access control and configuration policies can be defined [17].

Administrators of a domain may choose to group subsets of organizational users and computers in specific OUs to tightly control administrative access and apply differing policies to the various subsets of objects represented across OUs. For example, two servers of differing criticality may be split into two OUs to be able to specify two different domain user groups as administrators of the servers in one of OUs, but not both. When an object is placed within an OU, by default, it will inherit the security settings and policies defined at the OU [17]. This relationship between an object and its parent OU is key to understanding core AD functionality and how misconfiguration of OUs, their settings, and linked policies, can lead to problems pertaining to the objects within an OU. The main functionality in which objects within OUs are configured is via another important AD object, the Group Policy Object (GPO). Most importantly, Group Policy is the primary mechanism by which security configuration is distributed across the AD hierarchy [18].

B. Group Policy Objects

Group Policy Objects (GPOs) are another type of object within AD that describe a collection of configuration settings [18]. A GPO is comprised of two elements: an object that resides in AD and a folder structure stored in the AD file share, known as SYSVOL [18]. By default, all user and computer objects in a domain have read access to the GPOs stored within SYSVOL [18].

GPOs are created and modified to apply configuration settings to computers, users, or both, and are able to be linked to a site, an entire domain, or an OU [18]. Administrators can also define a few different types of filters that provide additional segmentation between subsets of systems located within the same OU. Windows Management Instrumentation (WMI) Filters, the policy application flag, and security filtering can be configured to further segment computers intended to apply or deny the GPO within an OU [18].

AD administrators may specify a WMI Filter to ensure all matching Windows 10 workstations apply the settings of a Windows 10-specific GPO while Windows 11 workstations within the same OU are exempted [18]. The policy application flag defines the category of settings within the GPO that should apply: user, computer, both, or neither settings category in which case the GPO would be disabled [18]. Finally, administrators can leverage security filtering to define specific users, computers, or groups allowed or denied read access to a GPO, for example [18].

When an administrator creates a GPO, the object is created in AD and the GPO folder structure is created in the SYSVOL file share [18]. Within this folder structure, Machine and User directories store the policies intended to apply to computers and users, respectively [18]. Depending on the object type the GPO is intended to target, a Registry.pol file is created within these folders to store the settings defined in the GPO [19]. When a GPO is linked to an OU, computers in the OU will apply the settings defined in the GPO provided they aren't filtered from reading or applying the GPO [18]. Considering GPOs are the mechanism in which settings are applied to computers within a domain, any misconfiguration of GPO or a related component can lead to host configuration or security issues.

C. Group Policy Processing and Enforcement

Previously stated, a domain is comprised of a hierarchy of containers, most notably, the OU container. In AD, this hierarchy implements the concepts of inheritance and precedence [18]. GPOs linked at the top of the hierarchy, the root of the domain, are inherited by objects within OUs underneath the root of the domain [18]. In AD, sub-OUs can be created under a parent OU, and GPOs can be linked to a parent to propagate settings throughout sub-OUs to inherit [18]. GPO precedence applies at and can be controlled at each OU [18]. AD hosts, by default, will process Group Policy on a configured interval but administrators can also force a computer to process Group Policy, as needed [18].

Application of GPOs can go awry for several reasons. OUs can be configured to block inheritance which results in GPOs defined above the OU being prevented from applying [18]. Incorrect GPO precedence defined at an OU may result in incorrect settings being applied [18]. GPO links can be disabled for a given OU resulting in the GPO's settings not being applied [18]. WMI and Security Filtering may incorrectly prevent a computer from reading and/or applying a GPO [18]. AD can also experience SYSVOL replication issues that can affect application of GPOs [17].

AD includes a query engine known as Resultant Set of Policy (RSoP), that helps resolve the complexity of Group Policy propagation [18]. RSoP queries GPOs and calculates the combination of applicable GPOs for a given host based on its OU placement and related GPO inheritance, precedence, and filtering [18]. While RSoP can assist in troubleshooting

and determining the computed GPOs that should be applied to a host, neither RSoP nor any other AD-native tools are able to track the enforcement status of the settings within the actual system state. RSoP may report that a GPO's settings are applied to an OU and the hosts it contains but there's a gap that exists between what should be applied to a host and what the host's actual system state is.

Real-time system state can be queried using Windows-native tools such as the Registry Editor or PowerShell, or third-party tools such as Tenable Nessus or Tanium. However, these tools have no awareness or concept of the actual policy intent in relation to the current system state. Together, these tools may be used by an administrator to bridge the gap between policy intent and real-time system state but due to the complexity of an AD environment, more than likely it's a manual and time-consuming process. This gap between policy intent and system state is not merely an operational inconvenience, it can result in a weakened security posture adversaries can exploit [1][6].

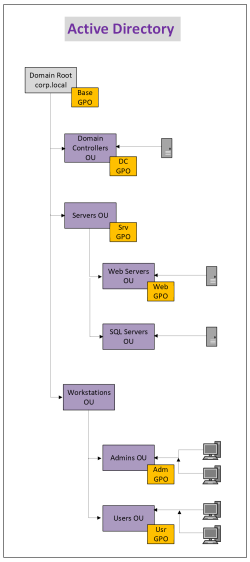

Fig. 1. Conceptual Active Directory OU hierarchy showing GPO linkage scope.

D. *Lateral Movement in Enterprise Attack Methodology*

In any cyber-attack, the lateral movement phase is key to an adversary achieving their primary objectives [14]. Once an adversary gains initial access, they enumerate the environment to determine any attack paths to exploit to advance their current access and standing in the environment, collect additional credentials, and sensitive information [14]. To do so, once an adversary compromises information or credentials that grant them access to other hosts in the environment, they'll begin pivoting between hosts to collect more sensitive credentials, information, and access critical to compromising the most sensitive data or systems within the environment [9][14]. Table I lists the common protocols used for lateral movement.

TABLE I. COMMON LATERAL MOVEMENT PROTOCOLS

Port	Protocol/Service	Lateral Movement Use
135	RPC/DCOM	Remote procedure calls; exploited for WMI-based execution and service enumeration
445	SMB	File sharing and named pipes; primary vector for pass-the-hash, PsExec, and ransomware propagation
3389	RDP	Remote desktop access; interactive lateral movement
5985	WinRM HTTP	Windows Remote Management over HTTP; PowerShell remoting and remote command execution
5986	WinRM HTTPS	Windows Remote Management over HTTPS; encrypted PowerShell remoting and remote execution

E. *The Case for Host Firewall Enforcement*

National cybersecurity agencies across the world, including the National Security Agency (NSA), Cybersecurity and Infrastructure Security Agency (CISA), National Institute of Standards and Technology (NIST), National Cyber Security Centre (NCSC), and Australian Signals Directorate (ASD), formally recommend enablement and configuration of host firewalls to limit and contain adversarial lateral movement [1][2][3][4][5]. In a joint NSA/CISA Cybersecurity Advisory AA23-278A, the lack of network segmentation is described as a systemic weakness across many networks [1]. NCSC guidance for preventing lateral movement states that local firewalls on hosts should be enabled to restrict unnecessary inbound and outbound traffic [2]. The NSA Methodology for Adversary Obstruction recommends host firewall rules as a mechanism to limit lateral movement and increase the survivability and defensibility of a network under attack [3]. NIST SP 800-41 and the ASD Information Security Manual further establish host firewall configuration as a recognized security control within their respective national frameworks [5][4].

Beyond formal guidance, real-world incidents demonstrate that adversaries actively exploit misconfigured or absent firewall enforcement to achieve lateral movement. In CISA Cybersecurity Advisory AA23-059A, the CISA Red Team describes an attack in which they were able to move laterally within a mature critical infrastructure organization's network due to non-uniform firewalls [6]. In the SolarWinds investigative update, the company identifies tightening firewall policies to further limit east/west traffic as one of the remediations targeted for increased security of their environment following a successful compromise [7]. In a Google Threat Intelligence report detailing recommendations to protect against destructive malware, wipers, or ransomware, one of the five focus areas provides guidance for on-premises lateral movement protections [8]. This guidance provides detailed Windows Defender host firewall configuration, via GPO, to prevent lateral movement and the risk posed by destructive malware based on in-the-wild threats tracked by Google Threat Intelligence [8]. These attacks illustrate a pattern in which simulated or real-world threat actors successfully leverage lateral movement to reach their objective due to a lack of segmentation via host firewalls.

F. Firewall Policy Enforcement in Enterprise AD Environments

Windows hosts provide host-based firewall functionality through Windows Defender Firewall, configurable at scale in Active Directory environments via Group Policy [19][20]. A GPO defines firewall rules specifying permitted and blocked traffic by port, protocol, and source address [19]. Firewall rules defined in a GPO are stored in the Registry.pol file within the GPO's directory structure [19]. When a host processes Group Policy, the host reads the firewall rules within the GPO's Registry.pol file and applies the rules to the host firewall [18][19]. Considering host firewall policy is delivered through GPO, the same inheritance, precedence, and filtering mechanics described in Section II-B and II-C govern whether rules apply to their intended hosts [18]. RSoP can be used to determine which firewall-related GPOs should apply to hosts within a given OU; however, it does not verify enforcement of those rules at the network level [18].

This paper categorizes the issues regarding firewall-related GPO enforcement into two distinct classes: implementation of firewall policy and delivery of firewall policy. The firewall implementation class describes issues in which the core firewall settings are misconfigured or non-existent. Firewall implementation issues present via overly permissive "any-any" rules, conflicting local rules, firewall profile state, and a lack of implementation of a firewall GPO altogether [19][20]. The delivery class describes issues regarding firewall-related GPOs not being applied or processed by the hosts intended for the policies [18]. Delivery-related issues could result due to misconfigured delegation permissions, inheritance, links, GPO state, security filtering, or WMI Filtering [18]. Any of these issues result in a weakened security posture for affected hosts and require manual and time-consuming root cause analysis and remediation efforts [1][6]. Despite authoritative guidance and demonstrated adversary exploitation, no existing tooling analyzes policy intent and the relationship to observed port state while recommending and executing remediations to limit the existence of network-level lateral movement paths.

G. Limitations of Existing Approaches

While a range of commercial and open-source tools address aspects of AD security and network compliance, no existing solution automates the full chain from observed host port state, to GPO policy intent, to root cause identification, and remediation. Table II summarizes current capability gaps of existing approaches relative to Iron Net's full validation chain.

TABLE II. GAP ANALYSIS

Tool	Category	Gap
Tenable Nessus	Vulnerability Scanner	Operates at the network/vulnerability layer; no awareness of GPO policy intent
BloodHound	Attack Path Mapping	Focused on credential-based attack paths and AD permission abuse; does not validate host firewall enforcement
PingCastle	AD Posture Assessment	Focused on identifying structural AD weaknesses; no awareness of observed network-level port state
gpresult/GPMC	Native AD Tooling	Shows policy intent; cannot verify firewall policies enforced at the network level
Tanium/SCCM	Endpoint Management	Can verify the presence of firewall registry keys defining firewall policy; no correlation of local firewall registry keys to policy intent

Table II shows that, across these tools, each of them may be used to address and verify a specific layer of the full problem domain but overall, each tool has been designed to treat a separate issue within the problem domain. As a result, an open port that violates intended firewall policy may be independently visible to a vulnerability scanner and independently explainable through manual GPO analysis, but no existing tool connects the observed port violation to the specific firewall implementation or delivery issue responsible for it [10][13][15]. Additionally, none of the reviewed tools provide automated or guided remediation of the underlying firewall or AD misconfiguration [13]. For a given identified enforcement gap issue, administrators must separately determine the correct firewall or AD-level remediation and execute it without tooling support.

H. Related Academic Work

Academic research relevant to automated firewall policy enforcement in enterprise networks spans three areas: lateral movement attack modeling, automated Active Directory security configuration assessment, and firewall policy analysis. Chen et al. model lateral movement as a contagion spreading across enterprise networks through the relationships between users, hosts, and applications, and show that targeted changes to network access configuration can meaningfully reduce how far an attacker can reach [9]. Powell applies the same epidemic framing to real Windows networks, mapping out which systems are connected through shared administrator credentials and active sessions. Powell also finds that relatively simple policy changes such as restricting accounts that can authenticate remotely, dramatically shrinks the paths available to an attacker [14]. Both papers establish that host-to-host lateral movement is a measurable and reducible attack surface, but neither asks whether the firewall rules that would block those paths are enforced on each host. In AD environments, this question depends both on whether GPO delivery succeeded and whether correct firewall rules are configured.

Nguyen et al.'s TriAD framework automates AD security auditing by checking domain configurations against established security benchmarks and visualizing potential attack paths, representing the closest prior work to Iron Net in scope and intent [13]. Ngo et al. address how to minimize the negotiation burden between security and IT operations teams when removing attack paths from an AD environment [11]. This adaptive approval problem's human-in-the-loop structure informed Iron Net's remediation design. What neither system addresses is the enforcement layer: TriAD checks whether settings conform to a checklist but cannot

determine whether firewall rules are actually reaching hosts [13], and Ngo et al. focus on credential-based privilege paths rather than network-layer enforcement gaps [11]. Neither traces a misconfiguration to its root cause in the GPO hierarchy or provides automated remediation of the underlying AD delivery failure.

Hamed and Al-Shaer classify the full range of conflicts that can exist within and across network security policy devices, and show that even experienced administrators produce these errors at high rates, making the case for automated detection [15]. Yuan et al. extend this with a toolkit that statically analyzes firewall configurations across distributed enterprise networks and has found real misconfigurations in production environments [10]. Both bodies of work assume the firewall rules are already present on the device being analyzed [10][15]. In AD environments, that assumption frequently does not hold considering firewall rules may never reach a host due to the lack of firewall implementation, blocked inheritance, disabled GPO links, or misconfigured filtering, and no existing academic work addresses this delivery failure or how to remediate it.

I. Research Gap and Contribution

The preceding survey reveals a consistent gap. While researchers have modeled lateral movement path remediation [9][14], audited AD security configurations [11][13], and analyzed firewall rule consistency [10][15], no existing work addresses the chain from observed host port state, to policy intent, and through to the issue's root cause. Additionally, no existing work provides automated remediation of AD firewall implementation and delivery failures that are the cause of enforcement gaps. Iron Net addresses this gap by analyzing firewall, Group Policy, and AD configuration to derive policy intent, correlate policy intent to observed network port state on a per-host basis, trace enforcement failures to their root cause in Active Directory, and execute or recommend targeted remediation through modification of Active Directory Group Policy. The following section describes Iron Net's architecture and the design decisions considered during development.

III. SYSTEM DESIGN

A. Architecture Overview

Iron Net is designed as a centralized security orchestration platform composed of three primary tiers: a fleet of lightweight endpoint agents, a central bridge application, and the Active Directory (AD) infrastructure it monitors and remediates. Figure 1 illustrates the high-level system architecture and the communication flows between these tiers.

The platform follows a detect, decide, remediate, and verify workflow. Misconfigured or entirely absent Group Policy Objects (GPOs) that permit lateral movement are detected through continuous AD analysis, presented to a security administrator for a remediation decision, executed against the AD infrastructure, and verified end-to-end through a coordinated agent-driven confirmation pipeline.

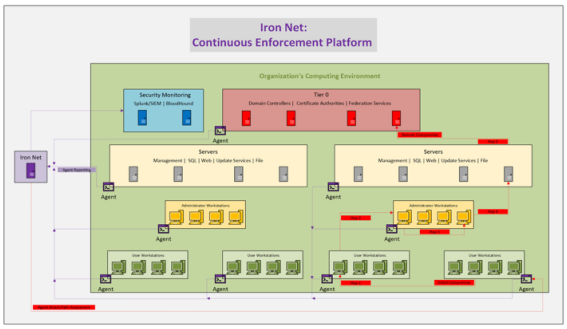

Fig. 2. Iron Net system architecture showing component relationships and data flow.

B. Project Scope and Constraints

Iron Net is scoped to Windows-based Active Directory environments where host firewall policy is delivered through Group Policy Objects [18]. The platform targets the specific class of misconfiguration in which GPOs governing host-based firewall behavior are absent, insufficiently restrictive, or failing to reach hosts, leaving lateral movement ports such as SMB, RDP, RPC, and WinRM reachable between hosts that should not communicate on those channels [1][6]. All GPO analysis, remediation, and verification operations assume a functioning AD domain with standard Group Policy processing on managed endpoints [18].

Several constraints bound the current implementation. GPO remediation is scoped exclusively to Windows Defender Firewall delivered through Group Policy [19][20]. Network-layer controls such as router ACLs and third-party host firewall products are outside the remediation scope, though port scan-based detection remains applicable in those environments. Agent deployment is currently limited to Windows hosts, as the validation and verification mechanisms are tightly coupled to the Windows Group Policy infrastructure [18]. All remediation decisions require explicit administrator approval before execution and at this point, no action is taken autonomously. BloodHound and Splunk integrations are supported but optional considering Iron Net performs its own lateral movement path computation and activity logging, independently of both.

C. Iron Net Bridge Console

The bridge is a Python-based Flask web application that serves as the central intelligence and coordination layer of the platform. It is responsible for agent lifecycle management, GPO analysis, remediation orchestration, and attack path tracking. The bridge exposes both a web-based administrative interface and a REST API consumed by the CVA agents. Its internal architecture is composed of five major functional subsystems.

1) Agent Registry and Task Dispatcher

The Agent Registry maintains the real-time operational state of all deployed CVA agents. It processes incoming heartbeats, updates agent status, and resolves agents as offline if no heartbeat is received within a configurable threshold. The task dispatcher queues and delivers task

payloads such as port scans and Group Policy update commands through the heartbeat response channel.

2) GPO Analyzer

The GPO Analyzer connects to AD via LDAP and performs continuous inspection of Group Policy Object configurations across all organizational units (OU) [17][18]. It evaluates each GPO's Registry.pol content, parsed from SYSVOL, to identify firewall policy conditions that permit lateral movement [19]. Specifically, it detects any-any inbound allow rules on lateral movement ports, missing or misconfigured firewall profiles, orphaned or disabled GPO links, blocked policy inheritance, and absent WMI filters [18][19]. Each identified condition is recorded as a 'DetectedIssue' record with full context, including the affected GPO distinguished name, matched rule details, and the set of AD containers to which the GPO is linked [17].

3) Decision Engine

The Decision Engine evaluates newly detected issues and generates 'Decision' records that are surfaced to the security administrator through the remediation decision queue. Each decision presents the administrator with one or more 'RemediationOption' entries, each carrying a risk classification, detailed affected objects, related attack paths, a plain-language description of the proposed change, and a reference to the handler that will execute it. All remediation actions require explicit administrator approval before execution, ensuring human oversight of every GPO modification [11].

4) GPO Remediator

The GPO Remediator executes approved remediation decisions by applying targeted changes to the AD and SYSVOL infrastructure [17][18]. Supported remediation operations include restricting source IP addresses on overly permissive inbound firewall rules, creating per-OU restriction GPOs that override inherited any-any rules via GPO precedence, configuring firewall profile defaults, and creating new baseline firewall GPOs for OUs lacking any firewall policy [19][20]. After each write operation, the remediator increments the GPO version number in both the AD object 'versionNumber' attribute and the SYSVOL GPT.INI file to ensure domain clients detect and download the updated policy [18]. It also ensures the Windows Firewall Client-Side Extension is registered in the GPO's 'gPCMachineExtensionNames' attribute, which is required for the firewall policy to be applied without a system reboot [18][19].

5) Verification Pipeline

The Verification Pipeline orchestrates a multi-step confirmation sequence following each executed remediation. The pipeline first re-scans the affected GPO in AD to confirm the change persisted. Upon confirmation, it dispatches the 'gpupdate /force' command task to all active agents in three rounds at 0, 10, and 30 seconds. This ensures accelerated Group Policy propagation across the endpoint fleet [18]. Following policy propagation, agents perform targeted port scans of the previously violating port connections. The pipeline evaluates the resulting scan reports against the original violation and marks the decision as `verified` if the violating ports are confirmed closed, or `failed` if the ports remain open, indicating the policy did not propagate correctly.

6) Persistent State Layer

All bridge subsystems share a common PostgreSQL relational database that provides durable state across the full detect-to-verify workflow. The schema is organized around five primary entity groups: agent operational state, GPO issue lifecycle, remediation audit trail, attack path records, and port violation history. The relational structure enforces referential integrity between these entities. A `Decision` record, for example, is directly foreign-keyed to both a `DetectedIssue` and a `RemediationOption`, and a `PortViolation` is linked to the specific `AttackPathNode` it activates to ensure that state transitions across subsystems remain consistent throughout the remediation lifecycle.

D. Validation Agents

The Continuous Validation Agent (CVA) is a Windows C# binary deployed to domain-joined endpoints across the monitored environment. Each agent is multi-purpose: it performs active port connectivity scans against peer hosts to measure lateral movement reachability and enforces the application of Group Policy when tasked.

Agents communicate with the bridge via a periodic heartbeat mechanism. Each heartbeat carries telemetry including the agent's hostname, IP address, operating system, organizational unit (OU) path, and the most recent scan statistics. The bridge responds to each heartbeat with a task payload, enabling the delivery of on-demand instructions such as targeted port scans and Group Policy refresh commands [18]. This pull-based task delivery model avoids the need for inbound network access to endpoints and accommodates firewall boundaries between the agent and the bridge.

Port scans are performed directly between endpoints. Each agent probes its assigned target hosts on the configured set of lateral movement ports, e.g., TCP 445/SMB, TCP 3389/RDP, TCP 135/RPC, TCP 5985–5986/WinRM [14]. Scan results are reported to the bridge where they are evaluated against defined port connection rules between host groups and correlated with known attack paths.

E. Active Directory and SYSVOL

Active Directory serves as both the authoritative source of GPO configuration and the target of all remediation actions [17]. The bridge interacts with AD via LDAP for read and write operations and via SMB for direct SYSVOL file access

to modify or create Group Policies [17][18]. Remediation writes are performed using the credentials of a configured service account, preferably with delegated GPO write permissions to follow the principle of least privilege [17].

F. Attack Path Engine

The Attack Path Engine computes lateral movement reachability across the environment using a breadth-first search (BFS) algorithm applied to the port scan result graph. Given a set of source hosts and a set of Tier 0 targets such as a domain controller, the engine identifies multi-hop paths along which an adversary could traverse the network using open lateral movement ports [9][14]. Each discovered path is represented as an `AttackPath` record composed of ordered `AttackPathNode` hops, each capturing the source and destination hosts, the open port enabling the hop, and the current validation status.

Attack path states follow a defined lifecycle: `pending`, in which the path was discovered but not yet validated by live scan data, `active`, in which one or more hops was confirmed open by agent scan results, `partially mitigated` in which a singular hop in the attack path is mitigated, and `mitigated`, in which all hops were confirmed closed. Exposure time is recorded from the moment a path first enters the `active` state to the moment it transitions to `mitigated`, providing a quantitative measure of remediation effectiveness [11]. If a previously mitigated hop subsequently reopens, due to GPO conflict, policy regression, or new GPO linkage, for example, the path automatically reverts to `active` and a new violation pipeline execution is initiated.

G. Port Classification Model

Lateral movement reachability is evaluated relative to a defined connectivity policy model. Hosts are assigned to `HostGroups` that reflect their operational tier and role. Common host groups may be Tier 0 systems, administrative workstations, user workstations, and application servers, for example [14]. `PortConnectivityRules` specify the expected connectivity posture between host group pairs for each monitored port that should be permitted or blocked. When an agent reports an open port between a source and destination host, the bridge resolves the applicable rule for that host group pair and port. An open port that violates a block rule generates a `PortViolation` record and contributes to the activation of any attack paths that traverse that hop. A closed port on a previously violating connection advances the corresponding hop toward the `mitigated` state.

IV. IMPLEMENTATION

A. GPO Misconfiguration Detection

The GPO Analyzer performs continuous inspection of Active Directory Group Policy Objects by querying the domain controller via LDAP and parsing policy content directly from SYSVOL [17][18]. For each GPO, the analyzer retrieves the associated Registry.pol file, the binary encoding of Windows Registry-based policy settings, and deserializes it to extract configured firewall rules and profile settings [19]. Detection logic is organized into two failure classes: firewall implementation issues, in which GPO firewall policy content is misconfigured or absent, and firewall delivery issues, in which correctly defined policy fails to reach its intended hosts due to GPO inheritance, link, security filtering, or WMI filter misconfiguration.

Firewall implementation detection evaluates each inbound firewall rule against a defined set of lateral movement ports, TCP 135, 139, 445, 3389, 5985, and 5986 [14]. A rule is flagged when it specifies an allow action with no source address restriction, referred to as an any-any rule, on one or more of these ports [19]. The analyzer also inspects firewall profile settings for permissive defaults such as disabled inbound blocking or enabled local rule merging, which can override administratively defined restrictions [19][20]. OUs with no linked firewall GPO are flagged as unprotected, with the analyzer querying the `gPLink` attribute of each organizational unit and cross-referencing linked GPOs against those containing firewall policy [18].

Firewall delivery detection evaluates the GPO inheritance and link configuration across the OU hierarchy. Checks evaluate inheritance blocking at the OU level, disabled GPO links, GPO status flags, security filtering configuration, delegation permissions, and WMI filter evaluation against target host properties [18]. Each detected condition is recorded as a `DetectedIssue` with the affected GPO distinguished name, matched rule or configuration details, and the set of AD containers to which the GPO is linked [17]. GPOs linked at the domain root are flagged in the associated `GpoContainerLink` record and surfaced prominently in the remediation interface, as domain root linkage affects all hosts domain-wide.

B. Port Violation Pipeline

CVA agents perform TCP connect scans against assigned target hosts on the configured lateral movement port set [14]. Scan results are submitted to the bridge as structured payloads containing per-host, per-port open or closed verdicts. Upon ingestion, each result is evaluated against the `PortConnectivityRules` defined for the source and destination host group pair. A result is classified as a violation when a port is observed open between two hosts for which the applicable rule specifies a block posture.

Confirmed violations are recorded as `PortViolation` records and enter a lifecycle: `detected` on first observation, `mitigated` when a subsequent scan confirms the port closed, and `regressed` if a previously mitigated port reopens. Regression detection is significant considering it identifies cases where a remediated GPO was overridden by a conflicting policy, a new GPO link was introduced, or Group Policy failed to reapply following a system event [18]. On each violation state transition, the attack path engine is notified to recompute the affected path segments.

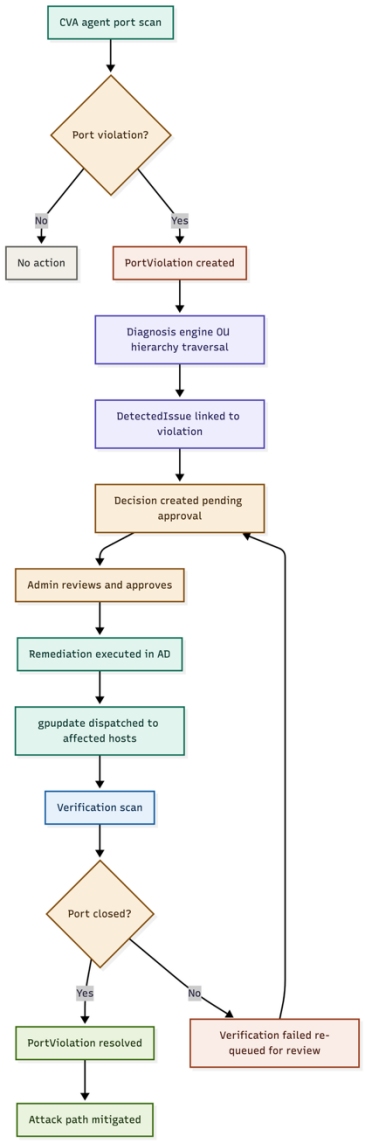

Fig. 3. Iron Net data flow pipeline from scan discovery to remediation verification.

C. Attack Path Computation

Attack path computation is performed by a breadth-first search (BFS) algorithm operating over the graph of observed host-to-host port reachability [9]. Source nodes are drawn from hosts assigned to non-Tier-0 host groups; target nodes are Tier 0 assets such as domain controllers, certificate authorities, or other administratively designated high-value hosts [14]. The BFS traverses open-port edges, identifying all paths along which an adversary could chain lateral movement steps to reach a Tier 0 target [9][14].

Each discovered path is stored as an `AttackPath` record composed of ordered `AttackPathNode` hops. Each hop records the source host, destination host, and the open port enabling traversal. Path state follows a three-value lifecycle: `pending`, a path was identified but awaiting scan confirmation, `active`, one or more hops confirmed open by live agent data, `partially mitigated`, in which some hops have been mitigated, and `mitigated`, in which all hops have been confirmed closed. The transition from `active` to `mitigated` records an exposure duration, computed as the elapsed time from first activation, providing a quantitative remediation metric [11]. Paths revert from `mitigated` to `active` upon detection of any hop regression, triggering re-entry into the violation pipeline.

D. Decision Engine

The Decision Engine translates detected GPO issues into structured remediation decisions surfaced to the security administrator. For each `DetectedIssue`, the engine creates a `Decision` record associating the issue with one or more `RemediationOption` entries. Each option carries a plain-language description of the proposed change, a risk classification, low, medium, high, or critical, and a reference to the handler within the GPO Remediator that will execute it. Options are ordered by recommended approach, with safer, more targeted options such as application of per-OU restriction GPOs ranked above broader modifications that could affect multiple hosts across varying OUs [11].

The administrator reviews each decision in a queue interface that surfaces the affected GPO, the specific matching rules or conditions, the linked computer objects, and the linked AD OUs. Upon selection of an option and explicit approval, the decision transitions from `pending` to `approved` and is passed to the GPO Remediator for execution. Decisions may also be rejected, deferring the issue without action. The engine monitors for cases where an issue resolves between detection and approval, for example, due to a manual modification made by an administrator, and automatically dismisses the associated decision to prevent unnecessary remediation.

E. GPO Remediator

The GPO Remediator executes approved remediation decisions by applying targeted writes to Active Directory and SYSVOL [17][18]. Remediation operations are organized across the same two failure classes as detection: firewall implementation remediations that correct or provision GPO firewall policy content, and firewall delivery remediations that restore correct GPO inheritance, link configuration, security filtering, and delegation. The full remediation catalog is provided in Appendix B.

Implementation remediations include restricting source addresses on any-any allow rules by rewriting Registry.pol with explicit IP range restrictions [19], creating per-OU restriction GPOs that override inherited permissive policy via GPO precedence [18][19], setting firewall profile defaults to block inbound connections [19][20], and provisioning baseline firewall GPOs for OUs with no existing firewall policy [18][19]. Delivery remediations include removing block inheritance flags from OUs, enabling disabled GPO links, enabling disabled GPO status, restoring Authenticated Users security filtering, removing explicit Deny ACEs from GPO DACLs, correcting WMI filter configuration, and granting missing read permissions to affected principals [17][18].

Following each write, the remediator increments the GPO version counter in both the AD object's `versionNumber` attribute and the SYSVOL GPT.INI file, ensuring domain clients detect the change during their next Group Policy polling cycle [18]. It applies the GPO's security descriptor to all newly written SYSVOL files, ensuring domain computers have read access [17]. Finally, it verifies that the Windows Firewall with Advanced Security Client-Side Extension GUID is registered in the GPO's `gPCMachineExtensionNames` attribute, which is required for the firewall policy to be applied without a host reboot [18][19].

F. Verification Chain

Following successful execution of a remediation, the Verification Chain confirms that the policy change propagated from AD through to the endpoint firewall layer. The chain proceeds through three sequential steps managed by a scheduler-driven state machine.

In the first step, the GPO Analyzer re-scans the affected GPO in AD. If the previously detected issue is no longer present, confirming the write persisted and AD replication completed, the chain advances [17]. If the issue is still detected, the decision is marked as failed, indicating the AD write did not take effect.

In the second step, the bridge dispatches `gpupdate /force` tasks to all active CVA agents through the heartbeat task channel [18]. To improve propagation reliability, tasks are issued in three rounds at intervals of zero, ten, and thirty seconds, accommodating environments where a single policy refresh may not immediately apply due to background processing or CSE sequencing [18].

In the third step, agents perform targeted port scans of the previously violating connections and submit results to the bridge. If the scan confirms the relevant ports are closed, the decision is marked `verified`, the associated `PortViolation` records are transitioned to `mitigated`, and the attack path engine recomputes the affected paths. If the ports remain open, the decision is marked `failed`, the issue is reset to an actionable state, and the administrator is notified for follow-up.

V. EVALUATION

Iron Net was evaluated against three primary questions: whether the system correctly enumerated and detected GPO misconfigurations responsible for firewall enforcement failures, whether remediation actions correctly process and resolve those failures to restore expected port state, and whether the system performed proper tracking and validation of identified attack paths.

A. Experimental Environment

Iron Net was developed, tested, and evaluated in a virtual lab hosted on a VMware vSphere instance consisting of sixteen hosts across a mix of Debian Linux, Windows Server 2025, and Windows 10. The environment included two domain controllers, a Windows Certificate Authority, file server, jump host, management host, SQL server, web/IIS server, WSUS host, Splunk host, two administrator workstations, and two user workstations. The bridge service and BloodHound were hosted on dedicated Debian instances. The second domain controller was added during evaluation to test Iron Net's agent registry and posture change tracking under a tier 0 attack path scenario.

The Active Directory environment was purpose-built to reflect realistic enterprise complexity [12]. Twenty OUs were created covering computer objects, administrative users, and non-administrative users across a multi-tier hierarchy with sub-OUs reflecting realistic organizational structure. Seventeen GPOs were created, modified, and remediated over the course of development. Each GPO misconfiguration category detectable by Iron Net was deliberately injected multiple times, with misconfigurations drawn from operational experience, published research [1][6][8], and systematic manual testing of GPO application failure modes. This iterative injection process produced strong ground truth coverage across all detection categories. The full environment details are provided in Appendix D.

B. Evaluation Methodology

Iron Net was evaluated against a ten-phase framework covering the full detection-to-remediation lifecycle, from environment visibility and scan coverage through GPO issue discovery, violation correlation, decision engine behavior, remediation execution, verification, and attack path validation. The framework was designed to answer four primary questions: whether Iron Net correctly identifies GPO misconfigurations responsible for firewall enforcement failures, whether the violation-to-issue correlation accurately traces failures to their root cause in the GPO hierarchy, whether remediation actions correctly modify AD and close the relevant ports, and whether attack paths are accurately tracked and marked mitigated following successful remediation.

Each phase was scored on a zero-to-100 scale and weighted by operational significance, with detection and remediation phases weighted most heavily at 15% each, reflecting their centrality to Iron Net's contribution. Table III presents the phase weights and scoring criteria. Two phases received modified treatment: scan coverage automation was not fully implemented within the project timeline and was validated manually via an Iron Net Security Posture dashoard during development rather than through an automated pipeline. The operational safety phase was assessed implicitly throughout evaluation, as all remediation testing was conducted in a live AD environment where any safety failures would have produced observable AD functional disruptions. Phases 1 and 3 through 9 were evaluated as designed against the deliberately injected misconfiguration ground truth described in Section V.A.

TABLE III. EVALUATION PHASE WEIGHTS AND SCORES

Phase	Weight	Preliminary Score	Final Score
Environment Visibility	10%	10	10
Scan Coverage	10%	N/A – final target	1.2

Phase	Weight	Preliminary Score	Final Score
Violation Detection	15%	12.8	12.8
GPO Issue Discovery	15%	15	15
Violation Pipeline Processing	10%	5	8.3
Recommended Decisions	10%	7.9	8.3
Remediation Execution	15%	10.5	10.9
Remediation Verification	5%	4.3	5
Attack Path Validation	5%	4.3	5
Operational Safety	5%	N/A – final target	5
Total	**100%**	**~70**	**~82**

C. Evaluation Results and Analysis

Iron Net achieved a composite weighted score of 82 out of 100 in final evaluation, improving from a preliminary score of 70 following targeted implementation fixes across four phases. The results demonstrate that the platform's core detection and remediation capabilities are sound, with the strongest performance in environment visibility, GPO issue discovery, and end-to-end verification.

Phases 1 and 4 achieved maximum scores in both preliminary and final evaluation. Environment visibility was complete across all 16 lab hosts with zero heartbeat failures and accurate OU path reporting throughout the evaluation period. GPO issue discovery achieved a perfect score reflecting the iterative misconfiguration injection methodology described in Section V.A. Each of the 16 detectable issue categories was exercised multiple times over the course of development, producing robust detection coverage that held consistently across evaluation rounds. Operational safety, assessed implicitly through the absence of AD functional disruptions across all remediation operations, and attack path validation and remediation verification, which reached maximum scores in final evaluation, confirm that Iron Net executes changes correctly and safely within a live AD environment.

The most analytically significant results are the phases that improved between preliminary and final evaluation. Violation pipeline processing improved from 5.0 to 8.3, reflecting fixes to the correlation engine that addressed cases where violations were not prioritized or diagnosis was incorrect, the primary gap identified during preliminary evaluation. Recommended decisions improved from 7.9 to 8.3 following corrections to decision creation and dismissal logic for edge cases in multi-OU GPO scope analysis. Remediation execution improved from 10.5 to 10.9 with the resolution of source address restriction issues identified during preliminary testing. These improvements help account for the 12-point gain between preliminary and final composite scores and validate the iterative evaluation approach.

Two phases did not achieve full scores in final evaluation. Violation detection scored 12.8 out of 15 in both rounds, indicating that while the detection pipeline correctly identifies the majority of violations, a subset of edge cases, primarily involving complex overlapping rule sets, were not fully resolved within the project timeline. Remediation execution similarly did not reach its maximum, reflecting cases where remediation options exist in the decision engine but the execution path requires additional hardening. Both represent known gaps documented in Section VI.C rather than unexpected failures. Scan coverage scored 1.2 out of 10, reflecting the partial implementation status described in Section V.B. Manual validation during development confirmed that scan rules, port coverage, and agent matching operate correctly, but the automated coverage pipeline was not completed within the project timeline.

VI. DISCUSSION

A. Analysis of Results

Iron Net achieved a composite weighted score of 82 out of 100 in final evaluation, reflecting a functionally sound platform that correctly executes the detect-decide-remediate-verify workflow across a realistic AD environment. The results validate the core thesis that GPO-layer misconfiguration is a detectable, attributable, and remediable root cause of host firewall enforcement failures. The results also validate that automating the full chain from observed port state to AD remediation is both technically feasible and operationally meaningful.

The perfect scores in environment visibility and GPO issue discovery are particularly significant. Environment visibility underpins every downstream capability: detection, correlation, remediation, and verification. These capabilities depend on accurate host enumeration, OU path resolution, and its perfect score confirms that Iron Net's AD integration is reliable enough to serve as an authoritative foundation for security decision-making. GPO issue discovery achieving maximum score across all 16 detectable misconfiguration categories reflects the depth of the detection engine and the breadth of the ground truth coverage established during development. That these two foundational phases scored perfectly in both preliminary and final evaluation suggests the detection architecture is stable. The architecture did not degrade under the conditions of evaluation and produced consistent results across repeated scan cycles.

The 12-point improvement between preliminary and final composite scores tells a specific story about where the hardest implementation problems in this space lie. The largest single gain was in violation pipeline processing, which improved from 5.0 to 8.3 following fixes to the correlation engine. This improvement reflects a finding that is broader than a bug fix: correctly attributing an observed port violation to its root cause in the GPO hierarchy is the most technically demanding step in Iron Net's workflow, requiring accurate OU traversal, inheritance resolution, and issue prioritization to produce a diagnosis that is both correct and actionable. The remaining gap in violation detection and remediation execution scores, neither of which reached their maximum in final evaluation, similarly points to edge cases at the boundary between policy intent and observed host behavior as the most challenging problems in automated GPO firewall validation.

The result that most directly advances the research contribution is the end-to-end verification finding. Prior work in firewall policy analysis operates on rule sets as static artifacts and has no mechanism for confirming that a policy change produced the intended network behavior on a specific host [10][15]. Iron Net's verification chain, coupling AD-level GPO remediation with agent-driven port confirmation, demonstrated in evaluation that this confirmation is achievable within a five-minute cycle from approval to verified port closure. The successful mitigation of attack paths reaching Tier 0 assets through this mechanism confirms that the GPO delivery layer is not merely a detection surface but a viable remediation surface, one that produces observable, verifiable security improvements rather than configuration changes whose host-level effect remains unknown. No tool in the landscape surveyed in Section II provides this capability, and the evaluation results establish that it is both implementable and effective within the constraints of a realistic AD environment.

B. Practical Applicability

Iron Net is packaged as a Docker Compose stack requiring only a domain service account with read permissions to begin producing results. An operator can move from a fresh deployment to a prioritized list of actionable GPO misconfigurations within a single scan cycle, without installing anything on endpoints or modifying existing configuration. The browser-based decision queue requires no client-side installation and presents detected issues in terms AD administrators already recognize such as GPO distinguished names, link structures, and OU scope, with a pre-selected remediation option and single-action approval. Changes applied through Iron Net are immediately visible in GPMC, eliminating any divergence between what Iron Net executes and what administrators observe in their existing tools [18].

The end-to-end verification chain addresses a gap that existing AD tooling does not fill. GPMC confirms that a policy is correctly configured in AD but provides no visibility into whether it was applied by the client and resulted in the intended port closure [18]. Iron Net closes this loop by coupling the AD-level change with agent-driven port confirmation, giving the operator positive evidence that an attack path has been closed rather than simply that a policy has been updated. In environments where firewall GPO hygiene has not been systematically reviewed, the detection capability alone, requiring only a read service account, is likely to surface multiple high-priority findings within the first scan [1][6]. Production deployment would require additional hardening including TLS enforcement, formal change management integration, and enterprise credential management for the service account; these considerations are addressed in Section VI.C.

C. Limitations

Iron Net's current implementation carries several limitations that distinguish the platform from production readiness. The most significant is scan coverage dependency. Attack path detection is bounded by agent deployment breadth, and hosts without an installed agent cannot contribute active scan results. In a large domain, meaningful reachability coverage requires coordinated agent rollout across representative hosts in each OU tier, which in production would require integration with endpoint management tooling such as SCCM or GPO-based software deployment [18].

The remediation subsystem carries its own constraints. GPO write operations are applied directly through the service account without integration into formal change management workflows, meaning Iron Net's approval gate is not a substitute for a change ticket in organizations that require one. The service account credentials are stored encrypted at rest in the database, which is appropriate for a lab environment but should be evaluated against organizational credential management standards, ideally replaced by an enterprise secrets manager, before production use. Remediation scope is also bounded to Windows Defender Firewall policy delivered through Group Policy [19][20]. Network-layer controls such as router ACLs and third-party host firewall products are outside the current remediation boundary.

Several security hardening items remain to be implemented. The bridge web interface and agent API channel do not enforce TLS in the current implementation, and the administrative interface lacks authentication beyond what the current build provides. The agent API channel, which accepts heartbeats and scan results from endpoints, should require mutual authentication or token-based verification in production to prevent spoofed result injection. Bridge placement also requires careful network planning in segmented environments, as the host could hold AD write credentials and its exposure should be limited to the minimum required communication paths [17].

Finally, the platform is currently scoped to Windows hosts. Linux and macOS endpoints are not supported, as the validation and verification mechanisms are tightly coupled to the Windows Group Policy infrastructure [18]. Extension to non-Windows platforms is architecturally feasible but outside the current project scope.

D. Future Work

Several directions would meaningfully extend Iron Net's capabilities beyond the current implementation. The most impactful near-term addition would be hardened communications throughout the platform. Enforcing TLS on the bridge web interface and agent API channel, implementing token-based or certificate-based mutual authentication between CVA agents and the bridge console, and replacing service account credential storage with an enterprise secrets manager would help reduce current security risks. These changes would close the primary security hardening gaps identified in Section VI.C and are prerequisites for production deployment.

A significant capability extension would be the integration of machine learning models for automated host rule classification and port usage profiling. By analyzing observed traffic patterns between hosts, host groups, and privilege tiers over time, Iron Net could learn what communication is normal for a given environment and surface deviations without requiring administrators to manually define expected flows.

This would reduce dependence on human-defined scan rules and allow the platform to adapt to organizational topology changes automatically, moving Iron Net toward continuous autonomous compliance monitoring rather than scan-cycle-driven detection.

The current remediation boundary, the Windows Defender Firewall delivered through Group Policy [19][20], could be extended in two directions. First, integration with network-level firewall platforms would make Iron Net's enforcement model host-and-network encompassing, correlating host-level port state with perimeter and internal network ACL configurations to provide a unified view of lateral movement exposure. Second, extending CVA agent support to Linux and macOS endpoints would allow Iron Net to operate across heterogeneous enterprise environments where Windows-only coverage leaves meaningful blind spots.

A longer-term research direction is the application of moving target defense principles to Iron Net's GPO enforcement model. Moving target defense creates asymmetric uncertainty by continuously shifting attack surface properties to increase attacker cost and degrade reconnaissance reliability [16]**.** Iron Net is well-positioned to implement dynamic firewall rule diversification across two complementary layers. GPO-based rotation sets the authoritative policy baseline while the CVA agent, already resident on monitored hosts, executes rule changes locally without waiting for Group Policy propagation. This two-layer approach enables MTD schedules that outpace attacker reconnaissance timelines without requiring router or DHCP integration [16]**.** Iron Net's separation between GPO policy intent and observed port state provides the ground truth model needed to distinguish legitimate MTD-driven port openings from genuine violations.

VII. CONCLUSION

This paper presented Iron Net, a platform for proactive detection and remediation of network-level lateral movement attack paths in Windows Active Directory environments. Iron Net addresses a gap that existing tools leave unaddressed: the silent failure of GPO-delivered host firewall policy to reach the hosts it is intended to protect [18][19], and the absence of any automated mechanism to trace those failures to their root cause in the AD GPO hierarchy and remediate them through the AD management plane [10][13][15].

Evaluated against a ten-phase framework in a purpose-built Active Directory lab environment, Iron Net achieved a composite weighted score of 82 out of 100, with perfect scores in environment visibility and GPO issue discovery and measurable improvement across violation pipeline processing, decision engine behavior, and attack path validation between preliminary and final evaluation rounds. The end-to-end verification chain, coupling AD-level GPO remediation with agent-driven port confirmation, demonstrated that policy delivery failures are not only detectable and attributable but correctable, with positive host-level evidence of enforcement improvement rather than configuration change alone.

Iron Net's contribution is a validated capability chain that no existing tool in the surveyed landscape provides from observed port state to GPO policy intent, OU inheritance root cause, targeted AD remediation, and verified port closure. Future work includes hardened agent communications, machine learning-based port usage profiling for autonomous compliance monitoring, network-layer firewall integration, cross-platform agent support, and the application of moving target defense principles to Iron Net's GPO enforcement model [16]. The platform represents a foundation for a broader class of AD-native security automation tools that operate at the policy delivery layer rather than the detection or access control layers that existing research predominantly addresses [9][11][13][14].

VIII. ACKNOWLEDGMENTS

The author used Claude (claude-sonnet-4-6) [21] as both a writing assistance tool and a development aid during the preparation of this manuscript and the implementation of Iron Net. All system architecture, technical design decisions, security research, and analytical conclusions represent the author's original ideas and synthesis; Claude assisted with prose drafting, structural organization, literature positioning, and implementation guidance during development.

The author would like to thank the course professors and teaching assistants for their feedback throughout the semester. Two feedback questions in particular shaped the direction of this work. The first questioned how the virtual testing environment would be representative of real-world systems, which prompted a more deliberate approach to lab environment design. This resulted in a lab environment engineered to reflect host roles, OU structures, GPO complexity, and misconfiguration patterns common in enterprise AD environments. Production environment testing remains outside the current project scope and is identified as a near-term next step.

The second question asked how the problem manifests in real-world systems and what attacks result from it. Addressing this feedback led to additional research that substantially strengthened both the motivation for Iron Net and the breadth of its detection capabilities. Security best practices, red team post-assessment findings, post-breach after action reports, and adversary obstruction guidance established that successful lateral movement through inadequately enforced host firewalls is a recurring and well-documented operational problem. This research also surfaced additional misconfiguration classes and GPO delivery failure modes that had not been initially considered, directly expanding the set of detectable issues Iron Net targets and informing the final catalog of detection categories. The author additionally thanks peers for their feedback during the semester. Several peer suggestions were incorporated into the current implementation and others have been identified as valuable directions for future development.

REFERENCES

[1] CISA and NSA, "NSA and CISA Red and Blue Teams Share Top Ten Cybersecurity Misconfigurations," *CISA.gov*, Oct. 5, 2023. [Online].

Available:https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-278a.

[2] NCSC, "Preventing Lateral Movement," *NCSC.gov.uk*, Feb. 8, 2018. [Online]. Available:https://www.ncsc.gov.uk/guidance/preventing-lateral-movement.

[3] NSA, "NSA Methodology for Adversary Obstruction," *National Security Agency*, Aug. 1, 2015. [Online]. Available: https://www.cdse.edu/Portals/124/Documents/webinars/nsa-methodology-for-adversary-obstruction.pdf.

[4] Australian Signals Directorate, "Information Security Manual," *cyber.gov.au*, Mar. 2026. [Online]. Available: https://www.cyber.gov.au/sites/default/files/2026-03/Information%20security%20manual%20%28March%202026%29.pdf.

[5] K. Scarfone and P. Hoffman, "Guidelines on Firewalls and Firewall Policy," NIST Special Publication 800-41 Rev. 1, National Institute of Standards and Technology, Gaithersburg, MD, USA, Sep. 2009. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-41r1.pdf.

[6] CISA, "CISA Red Team Shares Key Findings to Improve Monitoring and Hardening of Networks," *CISA.gov*, Feb. 28, 2023. [Online]. Available: https://www.cisa.gov/sites/default/files/2023-03/aa23-059a-cisa_red_team_shares_key_findings_to_improve_monitoring_and_hardening_of_networks_1.pdf.

[7] SolarWinds, "An Investigative Update of the Cyberattack," *SolarWinds Blog*, May. 7, 2021. [Online]. Available: https://www.solarwinds.com/blog/an-investigative-update-of-the-cyberattack.

[8] Google Threat Intelligence, "Preparation and Hardening Against Destructive Attacks," *Google Cloud Blog*, Mar. 6, 2026. [Online]. Available: https://cloud.google.com/blog/topics/threat-intelligence/preparation-hardening-destructive-attacks.

[9] Chen, P. Y., Choudhury, S., Rodriguez, L., Hero, A., & Ray, I. (2019). Enterprise cyber resiliency against lateral movement: A graph theoretic approach. *arXiv preprint arXiv:1905.01002*.

[10] Yuan, L., Chen, H., Mai, J., Chuah, C. N., Su, Z., & Mohapatra, P. (2006, May). Fireman: A toolkit for firewall modeling and analysis. In *2006 IEEE Symposium on Security and Privacy (S&P'06)* (pp. 15-pp). IEEE.

[11] Ngo, H. Q., Guo, M., & Nguyen, H. (2025). Adaptive Wizard for Removing Cross-Tier Misconfigurations in Active Directory. *arXiv preprint arXiv:2505.01028*.

[12] Nguyen, N. L., Falkner, N., & Nguyen, H. (2024, June). ADSynth: Synthesizing Realistic Active Directory Attack Graphs. In *2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (pp. 66-74). IEEE.

[13] Nguyen, G. H., Nguyen, K., Luu, M. T., Nguyen, A. N., & Ngo, T. S. (2025, January). Automated Framework for Active Directory Security and Compliance through Continuous Monitoring and Attack Path Validation. In *2025 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)* (pp. 287-292). IEEE.

[14] Powell, B. A. (2020). The epidemiology of lateral movement: exposures and countermeasures with network contagion models. *Journal of Cyber Security Technology*, *4*(2), 67-105.

[15] Hamed, H., & Al-Shaer, E. (2006). Taxonomy of conflicts in network security policies. *IEEE Communications Magazine*, *44*(3), 134-141.

[16] Sengupta, S., Chowdhary, A., Sabur, A., Alshamrani, A., Huang, D., & Kambhampati, S. (2020). A survey of moving target defenses for network security. *IEEE Communications Surveys & Tutorials*, *22*(3), 1909-1941.

[17] Microsoft, "Active Directory Technical Specification [MS-ADTS]," *Microsoft Open Specifications*, ver v20260309 [Online]. Available: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-adts/d2435927-0999-4c62-8c6d-13ba31a52e1a.

[18] Microsoft, "Group Policy: Core Protocol Specification [MS-GPOL]," *Microsoft Open Specifications*, ver 20240423. [Online]. Available: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-gpol/62d12924-6252-4052-996f-161d2b9019f4.

[19] Microsoft, "Group Policy: Firewall and Advanced Security Data Structure Specification [MS-GPFAS]," *Microsoft Open Specifications*, ver 20240423. [Online]. Available: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-gpfas/46e8d583-a4ce-4c43-b399-566afb1eec7f.

[20] Microsoft, "Group Policy: Security Protocol Extension Specification [MS-GPSB]," *Microsoft Open Specifications*, ver 20240423. [Online]. Available:https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-gpsb/6a07a06b-e628-4765-9d91-0d63ba47fdc0.

[21] Anthropic, "Claude (claude-sonnet-4-6)," *Anthropic*, 2026. [Online]. Available: https://www.anthropic.com.

APPENDIX A

POSTGRESQL SCHEMA DIAGRAM

The following diagram presents the Iron Net PostgreSQL schema. Tables are organized around the platform's core workflow: agent and host group tables represent the monitored environment; port violation and scan result tables capture reachability findings; detected issue and GPO container link tables record GPO analyzer output; decision and remediation log tables track administrator approvals and AD changes; and attack path and hop tables model lateral movement exposure. Foreign key relationships reflect the sequential dependency between detection, diagnosis, decision, execution, and verification.

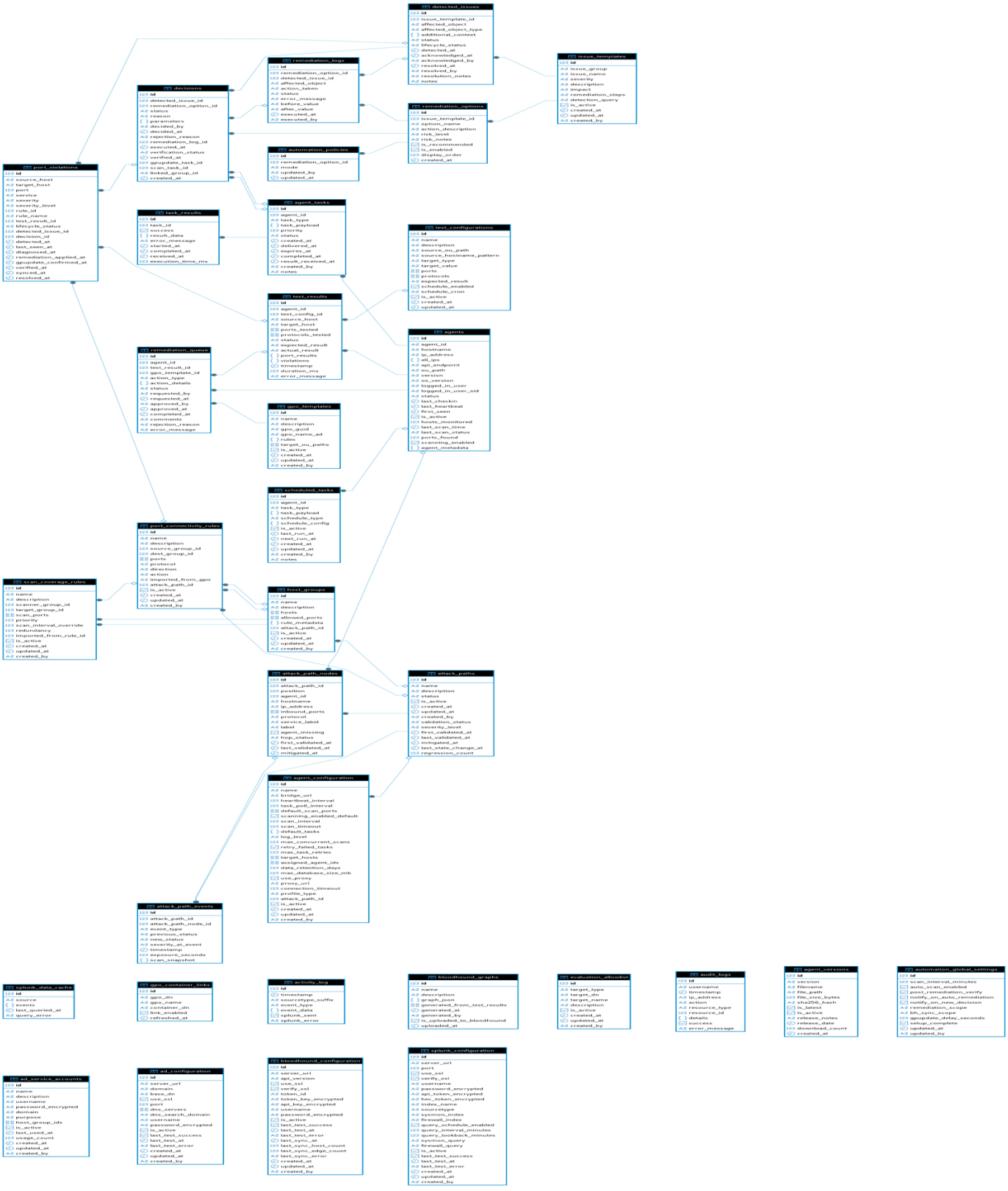

Fig. A1. Iron Net PostgreSQL database Entity-Relationship Diagram.

APPENDIX B

GPO ISSUE DETECTION AND REMEDIATION CATALOG

The following tables present the complete catalog of GPO misconfiguration categories detectable by Iron Net. Table B-I enumerates the sixteen detection types, including the detection logic applied by the GPO analyzer and the condition that triggers issue creation. Table B-II presents the twenty-one remediation options available across those detection categories, including the AD operation performed and the associated risk classification used by the decision engine to determine approval requirements.

Detection categories are organized by failure class: firewall delivery failures covering inheritance, link, GPO status, security filtering, delegation, and WMI filter issues; and firewall implementation gaps covering missing or misconfigured policy content. Remediation options are mapped to their corresponding detection categories; a single detection type may have multiple remediation options where the appropriate corrective action depends on root cause context identified during diagnosis.

TABLE B-I

GPO Misconfiguration Detection Catalog

Issue type	Failure class	Detection logic	Trigger condition
OU Blocking GPO Inheritance	Firewall Delivery	Query all OUs for inheritance_blocked=True; verify parent firewall GPOs are not locally linked to the blocked OU	OU has Block Inheritance set AND at least one ancestor-linked firewall GPO is not re-linked locally
Authenticated Users Read Access Blocked	Firewall Delivery	Parse GPO DACL for Deny ACEs on SIDs S-1-5-11 (Authenticated Users) or S-1-1-0 (Everyone); check for missing Allow Read	Deny ACE present on broad principal SID OR Allow Read absent for Authenticated Users
Explicit Deny ACE on GPO Principal	Firewall Delivery	Parse GPO nTSecurityDescriptor DACL; flag any non-default Deny ACE on a specific user, computer, or group SID	Deny ACE found for a principal other than the broad Authenticated Users / Everyone SIDs
Missing Read Permissions	Firewall Delivery	Enumerate GPO DACL entries; identify principals holding Apply Group Policy right without corresponding Read (DS_READ_PROP)	Principal has Apply Group Policy ACE but no Read ACE on the same GPO
Disabled GPO Link	Firewall Delivery	Read gPLink attribute of each container; parse link flag bits	Link flags bit 0 is set (flags & 1 == 1) on a firewall GPO link
Conflicting Link Order	Firewall Delivery	Enumerate all container GPO links; classify each GPO as firewall-related; detect multiple firewall GPOs in effective scope	Container has more than one firewall GPO in its effective set (same-level, cross-hierarchy, or both)
GPO Status: All Settings Disabled	Firewall Delivery	Read GPO flags AD attribute; interpret as 2-bit integer	flags == 3 (both Computer and User Configuration disabled)
GPO Status: Computer Configuration Disabled	Firewall Delivery	Read GPO flags AD attribute	flags == 2 (Computer Configuration disabled; all firewall rules inactive)
Authenticated Users Missing from Security Filter	Firewall Delivery	Check GPO DACL for Apply Group Policy right granted to S-1-5-11	Authenticated Users (S-1-5-11) lacks Apply Group Policy permission in the GPO security filter
Failed WMI Filter Evaluation	Firewall Delivery	Evaluate WQL query of associated WMI filter against OS version attributes of computers in linked OUs	WMI filter excludes all or a significant portion of computers in the linked OU scope
Missing WMI Filter	Firewall Delivery	Resolve the WMI filter DN referenced in GPO gPCWQLFilter attribute against AD objects	Referenced WMI filter object DN does not exist in AD
No Firewall GPOs Found	Firewall Implementation	Enumerate all GPOs domain-wide; classify by presence of firewall-related CSE extension GUIDs in gPCMachineExtensionNames	No GPO in the domain contains a Windows Firewall with Advanced Security CSE extension reference
Inbound Connections Not Set to Block	Firewall Implementation	Read Registry.pol via SYSVOL SMB; parse DefaultInboundAction per profile (Domain, Private, Public)	Any enabled profile has DefaultInboundAction = Allow or the value is not configured
Apply Local Firewall Rules Allowed	Firewall Implementation	Read Registry.pol; parse AllowLocalPolicyMerge per profile	Any enabled profile has AllowLocalPolicyMerge = True

All Profile Firewall States Disabled	Firewall Implementation	Read Registry.pol; parse EnableFirewall per profile	All configured profiles have EnableFirewall = False
Firewall GPO Contains Any-Any Rules on Lateral Movement Ports	Firewall Implementation	Read Registry.pol FirewallRules key; parse each rule's Action, RemoteAddresses, and LocalPort/RemotePort fields	Rule is enabled, action is Allow, RemoteAddresses is Any/*/empty, and port matches LM port set {22, 135, 139, 445, 1433, 3389, 4444, 5985, 5986, 47001, 593}

Table B-II presents the remediation options corresponding to the detection categories in Table B-I, including the specific AD operation executed and risk classification. Risk level informs the decision engine approval requirement: Low and Medium risk options may be queued for single-administrator approval; High risk options require elevated review.

TABLE B-II

GPO Remediation Options Catalog

Remediation option	Target detection category	AD operation performed	Risk level
Link Firewall GPO Locally	OU Blocking GPO Inheritance	Reads OU gPLink; appends parent firewall GPO DN; writes updated gPLink; reorders links by container proximity	Low
Remove Block Inheritance	OU Blocking GPO Inheritance	Writes gPOptions = 0 on the OU object	High
Restore Authenticated Users (Security Filtering)	Authenticated Users Missing from Security Filter	Adds Allow ACE granting Read + Apply Group Policy to S-1-5-11 on GPO DACL via nTSecurityDescriptor write	Low
Restore Authenticated Users (Delegation)	Authenticated Users Read Access Blocked	Removes Deny ACE for S-1-5-11 / S-1-1-0; adds Allow Read ACE if missing	Low
Remove Explicit Deny ACE	Explicit Deny ACE on GPO Principal	Removes targeted Deny ACE from GPO DACL via nTSecurityDescriptor write	Low
Grant Read Permission	Missing Read Permissions	Adds Allow Read (DS_READ_PROP) ACE for identified principal on GPO DACL	Low
Enable GPO Link	Disabled GPO Link	Reads container gPLink; clears bit 0 of link flags for the target GPO; writes updated gPLink	Medium
Promote Firewall GPO Link Order	Conflicting Link Order	Reads gPLink; sorts firewall GPO entries by container proximity (closest to OU = highest precedence); writes reordered gPLink	Low
Enable GPO	GPO Status: All Settings Disabled	Writes flags = 0 on GPO object	Medium
Enable Computer Configuration	GPO Status: Computer Configuration Disabled	Writes flags = 0 (or flags = 1 to preserve User Configuration disabled state) on GPO object	Medium
Update WMI Filter to Match OU	Failed WMI Filter Evaluation	Creates new msWMI-Som AD object with generated WQL targeting OU computer OS versions; links to GPO via gPCWQLFilter attribute write	Medium
Unlink WMI Filter from GPO	Failed WMI Filter Evaluation	Clears gPCWQLFilter attribute on GPO object	Low
Remove Missing WMI Filter Reference	Missing WMI Filter	Clears gPCWQLFilter attribute on GPO object	Low
Create Baseline Firewall GPO	No Firewall GPOs Found	Creates GPO AD object (groupPolicyContainer); writes SYSVOL directory structure and Registry.pol with default-block inbound profile settings via SMB; links to domain root via gPLink	Medium
Set Default Inbound to Block	Inbound Connections Not Set to Block	Reads and modifies Registry.pol via SYSVOL SMB; sets DefaultInboundAction = 1 per affected profile; increments GPO version	Medium
Disable Local Rule Merging	Apply Local Firewall Rules Allowed	Reads and modifies Registry.pol; sets AllowLocalPolicyMerge = 0 per affected profile; increments GPO version	Medium
Enable Windows Firewall	All Profile Firewall States Disabled	Reads and modifies Registry.pol; sets EnableFirewall = 1 per affected profile; increments GPO version	Low
Restrict Source Addresses on Broad Rules	Firewall GPO Contains Any-Any Rules on LM Ports	Reads Registry.pol; modifies matching Allow rules to set RA4= to authorized IP list; adds complement Block rules for all other IPv4 ranges; increments GPO version	Medium

Create Per-OU Restriction GPOs	Firewall GPO Contains Any-Any Rules on LM Ports	Creates per-OU GPO AD object; writes Registry.pol with complement Block rules (all IPv4 except authorized IPs) for LM ports; applies DACL to SYSVOL dirs; prepends GPO to OU gPLink	Low
Create Per-OU Profile GPO	Inbound Connections Not Set to Block / Apply Local Firewall Rules Allowed / All Profile Firewall States Disabled	Creates per-OU GPO AD object; writes Registry.pol with profile-specific fix; applies DACL; prepends to OU gPLink	Low
Migrate to Per-OU GPOs (Copy & Rename)	Firewall GPO Contains Any-Any Rules on LM Ports	Reads source GPO Registry.pol bytes and gPCMachineExtensionNames via SYSVOL SMB; creates new GPO per selected OU with OU-specific name; copies SYSVOL structure and Registry.pol content; prepends to OU gPLink; triggers background GPO re-scan	Low

APPENDIX C

FULL EVALUATION PHASE TABLES

The following tables present the complete evaluation framework applied to Iron Net across ten phases. Each table lists the individual tests within that phase, the method or pass criteria, and the preliminary and final evaluation results. Two phases received modified treatment: Phase 2 (Scan Coverage) was not fully automated within the project timeline and tests were validated manually during development; Phase 10 (Operational Safety) was assessed implicitly throughout evaluation via the absence of AD functional disruptions across all remediation operations.

Phase 1: Environment Visibility (weight: 10%)			
Test	**Method / pass criteria**	**Preliminary**	**Final**
New host detection	Add computer to AD; deploy agent → agent appears with correct hostname, IP, OU within one heartbeat cycle	Pass	Pass
Host removal handling	Remove/disable AD computer → Iron Net marks agent stale/offline; host flagged in coverage gaps	Pass	Pass
OU move tracking	Move computer to different OU → ou_path updates on next heartbeat; host group and GPO linkage recalculate	Pass	Pass
IP change tracking	Change host IP → agent record updates; scan results reference correct address	Pass	Pass
AD computer count accuracy	Compare Iron Net count against Get-ADComputer -Filter * → counts match within tolerance	Pass	Pass
Multi-homed host handling	Host with multiple NICs → agent tables contain all addresses; scans target correct reachable IP	Pass	Pass
Tier 0 categorization	Add new Tier 0 host → Iron Net labels host as Tier 0; related attack path classified Critical	Added during preliminary	Pass

Phase 2: Scan Coverage (weight: 10%)			
Test	**Method / pass criteria**	**Preliminary**	**Final**
Rule-to-assignment expansion	Create scan rule between two groups → all (scanner, target) pairs generated correctly	Not tested – not fully implemented	Not tested – not fully implemented
Auto-import from host rules	Use Import from Host Groups → scan rules created for every group pair with correct ports	Not tested – not fully implemented	Not tested – not fully implemented
New host in group triggers coverage	Add host to host group → new host appears in expected scan pairs	Not tested – not fully implemented	Not tested – not fully implemented
Removed host drops from coverage	Remove host from group → host no longer appears in expected pairs	Not tested – not fully implemented	Not tested – not fully implemented

Agent loss detection	Take agent offline (sole scanner) → coverage gap created; target flagged unreachable	Not tested – not fully implemented	Not tested – not fully implemented
Profile drift detection	Change agent profile to exclude target → drift detected and flagged	Not tested – not fully implemented	Not tested – not fully implemented
Scan staleness detection	Pause scanning → missing scan pairs increase; coverage percentage decreases	Not tested – not fully implemented	Not tested – not fully implemented
New scans reflect correct coverage	Create scans for new hosts → Security Posture dashboard reflects correct hosts scanned	Pass – manually monitored	Pass – manually monitored

Phase 3: Violation Detection (weight: 15%)			
Test	**Method / pass criteria**	**Preliminary**	**Final**
Block rule violation	Open port matching active block rule → PortViolation created with violation severity	Pass	Pass
Lateral movement warning	Open LM port with no block rule → PortViolation created with warning severity	Pass	Pass
Allowed traffic: no alert	Open port matching allow rule or non-LM port → no violation or warning generated	Pass	Pass
Structurally required traffic	Workstation→DC on 445 with allow rule for DC group → no violation (SYSVOL preserved)	Fail	Fail
Port closes: violation resolves	Close violating port → next scan marks PortViolation as resolved	Pass	Pass
Port reopens: new violation	Re-open resolved port → new PortViolation created; regression tracked	Pass	Pass
Duplicate prevention	Scan same open port multiple times → only one active PortViolation per (source, target, port)	Pass	Pass
Severity classification	Open SMB (445) vs SSH (22) → correct classification per LATERAL_MOVEMENT_PORTS	Not tested – methodology changed	Not tested – methodology changed

Phase 4: GPO Issue Discovery (weight: 15%)			
Test	**Method / pass criteria**	**Preliminary**	**Final**
Any-any firewall rule	GPO with default inbound allow on LM port → issue detected: Any-Any Rules on LM Ports	Pass	Pass
Blocked inheritance	OU blocks inheritance of firewall GPO → issue detected: OU Blocking GPO Inheritance	Pass	Pass
Disabled GPO	Disable all settings on firewall GPO → issue detected with correct GPO DN and status	Pass	Pass
Disabled link	Disable GPO link on OU → issue detected: Disabled GPO Link	Pass	Pass
Missing security filtering	Remove Authenticated Users from security filter → issue detected	Pass	Pass
Deny ACE	Add explicit Deny ACE to GPO DACL → issue detected: Explicit Deny ACE on GPO Principal	Pass	Pass
WMI filter exclusion	Attach WMI filter excluding target OS → issue detected: Failed WMI Filter Evaluation	Pass	Pass
Auto-resolve on fix	Manually fix any issue in AD then re-scan → issue transitions to Resolved automatically	Pass	Pass
No false issues on clean GPO	Create correctly configured firewall GPO → no issues detected for that GPO	Pass	Pass
Container link accuracy	Link GPO to multiple OUs then run scan → GpoContainerLink records match actual gPLink attributes	Pass	Pass

Phase 5: Violation Pipeline Processing (weight: 10%)			
Test	**Method / pass criteria**	**Preliminary**	**Final**
Direct link correlation	Violation on host in OU with misconfigured GPO linked directly → violation linked to correct issue	Pass	Pass
Inherited GPO correlation	Violation caused by GPO linked at parent OU → diagnosis walks hierarchy and finds correct issue	Pass	Pass
Nested OU hierarchy	Sub-OUs several levels deep → correct issue found despite host being nested deeply	Pass	Pass
Multiple issues: prioritization	Host affected by blocked inheritance and disabled link → most relevant issue linked (Implementation > Inheritance > Links)	Fail	Pass – implemented during preliminary evaluation
Diagnosis triggers scan	Violation with no matching issue in cache → targeted GPO scan triggered; issue discovered and linked	Fail	Pass – edge case identified: violation may remain unlinked
Issue resolved: violation re-diagnosed	Resolve linked issue; violation still open → violation re-enters diagnosis for next root cause	Fail	Fail

Phase 6: Recommended Decisions (weight: 10%)			
Test	**Method / pass criteria**	**Preliminary**	**Final**
Decision created per issue	Introduce new issue → decision record created with correct issue and option linkage	Partial	Pass
Correct options offered	Blocked inheritance issue → options include Link GPO Locally and Remove Block Inheritance	Pass	Pass
Risk-appropriate automation	Low risk + recommended + automated policy → decision auto-approved and executed	Not implemented	Not implemented
Human-required gating	High/Critical risk option → decision stays pending regardless of automation policy	Not implemented	Not implemented
Impact analysis: affected OUs	Decision for GPO linked to 3 OUs → impact shows all 3 OUs with hosts	Pass	Pass
Impact analysis: attack paths	Violation on hop of active attack path → impact surfaces related attack path with status	Pass	Pass
Shared GPO awareness	GPO linked at domain root or multiple OUs → warning displayed; per-OU option offered	Pass	Pass
Posture change on approve	Approve and execute decision → open issues decrease; resolved increase; Security Posture reflects change	Pass	Pass
Decision resolved on manual fix	Manually remediate GPO issue → pending decision linked to issue removed on next scan	Added during preliminary	Pass
Re-introduction of remediated decision	Re-introduce previously remediated GPO issue → pending decision re-populates on next scan	Added during preliminary	Pass
Ranked remediation decisions	Multiple issues as root cause for attack path → decisions ranked by most impactful remediation	Added during preliminary	Pass
Linked remediation decisions	Multiple remediations required to fully remediate violation → linked remediation group shown in each decision	Added during preliminary	Pass

Phase 7: Remediation Execution (weight: 15%)			
Test	**Method / pass criteria**	**Preliminary**	**Final**
Remove Block Inheritance	Approve decision on OU with blocked inheritance → gPOptions cleared; parent GPO now applies to hosts in OU	Pass	Pass
Link GPO Locally	Approve decision for missing GPO on OU → gPLink updated with firewall GPO DN at correct position	Pass	Pass

Enable GPO Link	Approve decision on disabled link → link flags changed to enabled in gPLink attribute	Pass	Pass
Enable GPO Status	Approve decision on disabled GPO → flags attribute set to 0 (all settings enabled)	Pass	Pass
Restrict Source Addresses	Approve with specific IPs → Registry.pol updated with LM ports restricted to specific IPs only	Fail	Pass
Create Baseline Firewall GPO	Approve on OU with no firewall GPO → new GPO created, linked, and confirmed in gpresult	Pass	Pass
Restore Authenticated Users	Approve on GPO missing security filter → DACL updated with Read+Apply for Authenticated Users	Pass	Pass
Remove Deny ACE	Approve on GPO with explicit Deny → Deny ACE removed; non-Deny ACEs preserved	Fail	Fail
Audit trail completeness	Execute any remediation → log captures value change, affected object, timestamp	Fail	Fail
Partial failure isolation	Multi-step remediation with one failing step → completed steps persist; AD not left broken	Not tested	Not tested
Idempotency	Execute same remediation twice → second execution is no-op; no duplicate GPOs or links	Pass	Pass

Phase 8: Remediation Verification (weight: 5%)			
Test	**Method / pass criteria**	**Preliminary**	**Final**
gpupdate task delivery	Execute remediation → agent gpupdate task created and delivered to affected hosts on next heartbeat	Pass	Pass
Verification scan triggered	gpupdate completes → port scan task queued targeting same ports as original violation	Pass	Pass
Successful verification	Port closed after remediation → proper verification tracking attributes set	Pass	Pass
Failed verification	Port still open after remediation → verification failed; violation remains active; issue not auto-resolved	Pass	Pass
Posture delta	Successful verification completes → Security Posture: active violations decrease, resolved increase	Pass	Pass
Full loop timing	Measure approve → execute → gpupdate → verify cycle → completes within expected window	Pass	Pass

Phase 9: Attack Path Validation (weight: 5%)			
Test	**Method / pass criteria**	**Preliminary**	**Final**
Path activation	All hops have open LM ports confirmed by scan → path registered as active; all sub-paths show confirmed	Pass	Pass
Single hop remediation breaks path	Remediate one hop in multi-hop path → hop transitions to mitigated; entire path transitions to mitigated	Pass	Pass
Partial remediation: path stays active	Remediate one hop in 3-hop path; other 2 still open → remediated hop mitigated; path mitigated (one broken link breaks chain)	Pass	Pass
Regression detection	After mitigation, port reopens (GPO reverted) → hop and path transition back to active; regression count increments	Pass	Pass
Exposure tracking	Path active for period then mitigated → attack path records active and mitigated timestamps with exposure time	Fail	Pass
Severity recalculation	Path reaches Tier 0 server → severity = Critical; non-Tier-0 path not Critical	Pass	Pass
Posture delta	Active path mitigated → Security Posture: active paths decrease, mitigated increase	Pass	Pass

Phase 10: Operational Safety (weight: 5%)			
Test	**Method / pass criteria**	**Preliminary**	**Final**
GPO application preserved	Execute remediation, then gpresult /R on affected hosts → all expected GPOs still apply; no unintended GPOs removed	Not tested	Pass
SYSVOL access preserved	After blocking peer SMB, test workstation SYSVOL access → \\domain\SYSVOL reachable; Group Policy processing unaffected	Not tested	Pass
Rollback capability	Unlink or disable Iron Net-created GPO → original behavior restored; Iron Net detects change on next scan	Not tested	Pass
Allowed application traffic preserved	Remediate LM ports; check explicitly allowed ports (SQL, custom apps) → non-LM traffic unaffected	Not tested	Pass
No collateral modification	Execute remediation targeting one OU → GPOs linked to other OUs untouched; violation pipeline confirms scope	Not tested	Pass

APPENDIX D

LAB ENVIRONMENT TOPOLOGY

The following output was generated by the Iron Net lab environment export script (Export-IronNetLabEnvironment.ps1), executed on a domain-joined host at the conclusion of the evaluation period. The script queries Active Directory via PowerShell and the Group Policy module to capture a point-in-time snapshot of the lab environment structure. Output is organized into eight sections: domain summary, organizational unit hierarchy with block inheritance flags, computer objects with operating system and OU placement, GPO inventory with creation dates and status, GPO links per container with link order and enforcement state, WMI filter objects with associated WQL queries and GPO assignments, GPO security filtering principals, and domain controller inventory with FSMO role assignments. The export script is available in the Iron Net repository at https://github.com/huskersec/IronNet.

```
Iron Net Lab Environment -- Active Directory Structure Export
Domain : deathstar.local
Generated : 2026-04-26 22:17:29


==============  SECTION 1: DOMAIN SUMMARY  ==============
Domain Name            : deathstar.local
NetBIOS Name           : DEATHSTAR
Domain Mode            : Windows2025Domain
Forest Name            : deathstar.local
Forest Mode            : Windows2025Forest
Domain Controllers     : srv-dc1.deathstar.local
PDC Emulator           : srv-dc1.deathstar.local
Domain DN              : DC=deathstar,DC=local


==============  SECTION 2: ORGANIZATIONAL UNIT HIERARCHY  ==============
OU Name                  Block Inheritance   Distinguished Name
-----------------------  -----------------   ----------------------------------------
Admin                    No                  OU=Admin,OU=DSTR WORKSTATIONS,DC=deathstar,DC=local
CA                       No                  OU=CA,OU=DSTR SERVERS,DC=deathstar,DC=local
Domain Admins            No                  OU=Domain Admins,OU=DSTR ADMINS,DC=deathstar,DC=local
```

```
Domain Controllers        No                  OU=Domain Controllers,DC=deathstar,DC=local
DSTR ADMINS               No                  OU=DSTR ADMINS,DC=deathstar,DC=local
DSTR GROUPS               No                  OU=DSTR GROUPS,DC=deathstar,DC=local
DSTR SERVERS              No                  OU=DSTR SERVERS,DC=deathstar,DC=local
DSTR SERVICE ACCTS        No                  OU=DSTR SERVICE ACCTS,DC=deathstar,DC=local
DSTR USERS                No                  OU=DSTR USERS,DC=deathstar,DC=local
DSTR WORKSTATIONS         No                  OU=DSTR WORKSTATIONS,DC=deathstar,DC=local
Enterprise Admins         No                  OU=Enterprise Admins,OU=DSTR ADMINS,DC=deathstar,DC=local
File                      No                  OU=File,OU=DSTR SERVERS,DC=deathstar,DC=local
Jump                      Yes                 OU=Jump,OU=DSTR SERVERS,DC=deathstar,DC=local
Management                No                  OU=Management,OU=DSTR SERVERS,DC=deathstar,DC=local
Server Admins             No                  OU=Server Admins,OU=DSTR ADMINS,DC=deathstar,DC=local
Splunk                    No                  OU=Splunk,OU=DSTR SERVERS,DC=deathstar,DC=local
SQL                       No                  OU=SQL,OU=DSTR SERVERS,DC=deathstar,DC=local
User                      No                  OU=User,OU=DSTR WORKSTATIONS,DC=deathstar,DC=local
Web                       No                  OU=Web,OU=DSTR SERVERS,DC=deathstar,DC=local
Workstation Admins        No                  OU=Workstation Admins,OU=DSTR ADMINS,DC=deathstar,DC=local
WSUS                      No                  OU=WSUS,OU=DSTR SERVERS,DC=deathstar,DC=local

Total OUs: 21

===============  SECTION 3: COMPUTER OBJECTS  ===============
Hostname                  Operating System                    Enabled    OU Path
------------------------  ----------------------------------  ---------  --------------------------------
SRV-CA                    Windows Server 2025 Datacenter      Yes        OU=CA,OU=DSTR SERVERS
SRV-DC1                   Windows Server 2025 Datacenter      Yes        OU=Domain Controllers
SRV-EVAL                  Windows Server 2025 Datacenter      Yes        CN=Computers
SRV-FS                    Windows Server 2025 Datacenter      Yes        OU=File,OU=DSTR SERVERS
SRV-JUMP                  Windows Server 2025 Datacenter      Yes        OU=Jump,OU=DSTR SERVERS
SRV-MGMT                  Windows Server 2025 Datacenter      Yes        OU=Management,OU=DSTR SERVERS
SRV-SPLUNK                Windows Server 2025 Datacenter      Yes        OU=Splunk,OU=DSTR SERVERS
SRV-SQL                   Windows Server 2025 Datacenter      Yes        OU=SQL,OU=DSTR SERVERS
SRV-WEB                   Windows Server 2025 Datacenter      Yes        OU=Web,OU=DSTR SERVERS
SRV-WSUS                  Windows Server 2025 Datacenter      Yes        OU=WSUS,OU=DSTR SERVERS
WKST-ADM1                 Windows 10 Education                Yes        OU=Admin,OU=DSTR WORKSTATIONS
WKST-ADM2                 Windows 10 Education                Yes        OU=Admin,OU=DSTR WORKSTATIONS
WKST-PROD2                Windows 10 Education                Yes        OU=User,OU=DSTR WORKSTATIONS

Total Computer Objects: 13

===============  SECTION 4: GPO INVENTORY  ===============
GPO Name                                          Status                   Created
------------------------------------------------  -----------------------  --------------------
Default Domain Controllers Policy                 AllSettingsEnabled       2026-01-28
Default Domain Policy                             AllSettingsEnabled       2026-01-28
```

```
DSTR Auditing                                   AllSettingsEnabled        2026-04-10
DSTR Autologon - Admin Workstations             AllSettingsEnabled        2026-02-27
DSTR Autologon - User Workstations              AllSettingsEnabled        2026-02-27
DSTR Firewall - Admin Workstations              AllSettingsEnabled        2026-04-13
DSTR Firewall - Admin Workstations 2            AllSettingsEnabled        2026-04-25
DSTR Firewall - Base                            AllSettingsEnabled        2026-04-10
DSTR Firewall - Base - Web                      AllSettingsEnabled        2026-04-25
DSTR Firewall - CAs                             AllSettingsEnabled        2026-04-13
DSTR Firewall - DCs                             AllSettingsEnabled        2026-04-10
DSTR Firewall - MGMT                            AllSettingsEnabled        2026-04-25
DSTR Firewall - Splunk                          AllSettingsEnabled        2026-04-10
DSTR Remote Management                          AllSettingsEnabled        2026-02-27
DSTR Server Administrators                      AllSettingsEnabled        2026-01-30
DSTR WFP Auditing                               AllSettingsEnabled        2026-02-20
DSTR Workstation Administrators                 AllSettingsEnabled        2026-01-31
DSTR WSUS                                       AllSettingsEnabled        2026-01-31
IronNet - LM Restriction - File                 AllSettingsEnabled        2026-04-25

Total GPOs: 19

===============  SECTION 5: GPO LINKS PER CONTAINER  ===============
Container: DC=deathstar,DC=local
  Block Inheritance:
  Linked GPOs:
    [1] Default Domain Policy                           Enabled      Not Enforced
    [2] DSTR WSUS                                       Enabled      Not Enforced
    [3] DSTR WFP Auditing                               Enabled      Not Enforced
    [4] DSTR Remote Management                          Enabled      Not Enforced
    [5] DSTR Auditing                                   Enabled      Not Enforced
    [6] DSTR Firewall - Base                            Enabled      Not Enforced

Container: OU=Admin,OU=DSTR WORKSTATIONS,DC=deathstar,DC=local
  Block Inheritance:
  Linked GPOs:
    [1] DSTR Autologon - Admin Workstations             Enabled      Not Enforced
    [2] DSTR Firewall - Admin Workstations 2            Enabled      Not Enforced

Container: OU=CA,OU=DSTR SERVERS,DC=deathstar,DC=local
  Block Inheritance:
  Linked GPOs:
    [1] DSTR Firewall - CAs                             Enabled      Not Enforced

Container: OU=Domain Controllers,DC=deathstar,DC=local
  Block Inheritance:
  Linked GPOs:
```

```
    [1] Default Domain Controllers Policy                  Enabled      Not Enforced
    [2] DSTR WSUS                                          Enabled      Not Enforced
    [3] DSTR Firewall - DCs                                Enabled      Not Enforced

Container: OU=DSTR SERVERS,DC=deathstar,DC=local
  Block Inheritance:
  Linked GPOs:
    [1] DSTR Server Administrators                         Enabled      Not Enforced

Container: OU=DSTR WORKSTATIONS,DC=deathstar,DC=local
  Block Inheritance:
  Linked GPOs:
    [1] DSTR Workstation Administrators                    Enabled      Not Enforced

Container: OU=File,OU=DSTR SERVERS,DC=deathstar,DC=local
  Block Inheritance:
  Linked GPOs:
    [1] IronNet - LM Restriction - File                    Enabled      Not Enforced

Container: OU=Jump,OU=DSTR SERVERS,DC=deathstar,DC=local
  Block Inheritance:
  Linked GPOs:
    [1] DSTR Firewall - Base                               Enabled      Not Enforced

Container: OU=Management,OU=DSTR SERVERS,DC=deathstar,DC=local
  Block Inheritance:
  Linked GPOs:
    [1] DSTR Firewall - MGMT                               Enabled      Not Enforced

Container: OU=Splunk,OU=DSTR SERVERS,DC=deathstar,DC=local
  Block Inheritance:
  Linked GPOs:
    [1] DSTR Firewall - Splunk                             Enabled      Not Enforced

Container: OU=User,OU=DSTR WORKSTATIONS,DC=deathstar,DC=local
  Block Inheritance:
  Linked GPOs:
    [1] DSTR Autologon - User Workstations                 Enabled      Not Enforced

Container: OU=Web,OU=DSTR SERVERS,DC=deathstar,DC=local
  Block Inheritance:
  Linked GPOs:
    [1] DSTR Firewall - Base - Web                         Enabled      Not Enforced

==============  SECTION 6: WMI FILTERS  ==============
```

```
   WMI Filter Name : IronNet - Auto - DSTR SERVERS

   Description: Auto-generated by IronNet for OU: DSTR SERVERS

   WQL Query: 1;3;10;90;WQL;root\CIMv2;SELECT * FROM Win32 OperatingSystem WHERE (Version LIKE "10.0.26100%"
AND ProductType="3");

   DN: CN={8F389B56-1646-4AE7-8EF3-62B44733B88B},CN=SOM,CN=WMIPolicy,CN=System,DC=deathstar,DC=local


   WMI Filter Name : Windows 10

   Description:

   WQL Query: 1;3;10;84;WQL;root\CIMv2;SELECT * FROM Win32_OperatingSystem where Version like "10.0.1%" and
ProductType="1";

   DN: CN={80DA14A2-EE4F-4188-BA8E-71AB24DFE57B},CN=SOM,CN=WMIPolicy,CN=System,DC=deathstar,DC=local


   WMI Filter Name : Windows Server 2025

   Description:

   WQL Query: 1;3;10;109;WQL;root\CIMv2;SELECT * FROM Win32_OperatingSystem WHERE Version LIKE "10.0.26100%"
AND (ProductType="2" OR ProductType="3");

   DN: CN={0D69586C-887C-47A7-B6A4-63BE5084F725},CN=SOM,CN=WMIPolicy,CN=System,DC=deathstar,DC=local


   WMI Filters Associated with GPOs:


   GPO Name                                          WMI Filter
   ------------------------------------------------  ---------------------------------------
   Default Domain Controllers Policy                 None
   Default Domain Policy                             None
   DSTR Auditing                                     None
   DSTR Autologon - Admin Workstations               None
   DSTR Autologon - User Workstations                None
   DSTR Firewall - Admin Workstations                None
   DSTR Firewall - Admin Workstations 2              None
   DSTR Firewall - Base                              None
   DSTR Firewall - Base - Web                        None
   DSTR Firewall - CAs                               None
   DSTR Firewall - DCs                               None
   DSTR Firewall - MGMT                              None
   DSTR Firewall - Splunk                            None
   DSTR Remote Management                            None
   DSTR Server Administrators                        None
   DSTR WFP Auditing                                 None
   DSTR Workstation Administrators                   None
   DSTR WSUS                                         None
   IronNet - LM Restriction - File                   None


   Total WMI Filters: 3


   ==============  SECTION 7: GPO SECURITY FILTERING  ==============
   GPO Name                                          Security Filter Principals
```

```
   ------------------------------------------------  ------------------------------------------------

   Default Domain Controllers Policy                 Authenticated Users
   Default Domain Policy                             Authenticated Users
   DSTR Auditing                                     Authenticated Users
   DSTR Autologon - Admin Workstations               Authenticated Users
   DSTR Autologon - User Workstations                Authenticated Users
   DSTR Firewall - Admin Workstations                Authenticated Users
   DSTR Firewall - Admin Workstations 2              Authenticated Users
   DSTR Firewall - Base                              Authenticated Users
   DSTR Firewall - Base - Web                        Authenticated Users
   DSTR Firewall - CAs                               Authenticated Users
   DSTR Firewall - DCs                               Authenticated Users
   DSTR Firewall - MGMT                              Authenticated Users
   DSTR Firewall - Splunk                            Authenticated Users
   DSTR Remote Management                            Authenticated Users
   DSTR Server Administrators                        Authenticated Users
   DSTR WFP Auditing                                 Authenticated Users
   DSTR Workstation Administrators                   Authenticated Users
   DSTR WSUS                                         Authenticated Users
   IronNet - LM Restriction - File                   Authenticated Users


   ===============  SECTION 8: DOMAIN CONTROLLER SUMMARY  ===============

   DC Name         IP Address   OS                          Roles
   ---------       -------      --------------              ---------------------------------------

   SRV-DC1        2.1.1.50   Windows Server 2025 Datacenter SchemaMaster, DomainNamingMaster, PDCEmulator,
RIDMaster, InfrastructureMaster


   Total Domain Controllers: 1

   ============================================================

     End of Export

     Iron Net Lab Environment -- deathstar.local

     2026-04-26 22:17:29

============================================================
```

APPENDIX E

CODE REPOSITORY AND TECHNICAL ARTIFACTS

The Iron Net source code, configuration templates, and agent deployment packages are available at the following repository:

https://github.com/huskersec/IronNet

The repository includes all components required to run Iron Net in a Docker environment, including, the Flask bridge service, PostgreSQL schema definitions, CVA agent source and compiled executable within the Bridge.