TUTORIAL

How to Detect Lateral Movement with Network Traffic Analysis

Lateral movement often hides inside ordinary internal traffic, so detection depends on knowing which patterns matter, how to validate them, and how to operationalize the results. This tutorial shows a practical workflow for detecting lateral movement with network traffic analysis, from prerequisites and collection to validation and follow-up.

Security - July 22, 2026

Why lateral movement is hard to see in network traffic

Lateral movement is the stage where an intruder shifts from one internal system to another after the initial foothold. Operationally, this matters because the traffic often looks like normal administration, file sharing, authentication, or remote management. If you only monitor perimeter traffic, you can miss the internal pivot that turns a single compromised host into a domain-wide incident.

This tutorial shows you how to detect lateral movement with network traffic analysis in a way that is practical for system and security engineers. By the end, you should be able to decide whether your environment has enough telemetry for this approach, build a detection workflow, validate suspicious patterns, and understand what to confirm before using the results in production response.

What you are building

You are not trying to classify every internal packet as malicious. You are building a repeatable process that surfaces suspicious internal connections, enriches them with context, and reduces false positives until you have a defensible alert or hunt signal.

A finished implementation should give you:

- visibility into east-west traffic between internal hosts
- a baseline of normal connection patterns by subnet, protocol, and service
- detection logic for anomalous pivots, rare service use, credentialed access bursts, and scan-like internal behavior
- validation evidence that distinguishes benign administration from likely movement
- a follow-up process for tuning, incident triage, and reporting

If you already have zero trust network segmentation work in progress, it can help narrow what ?normal? internal access should look like. If not, you can still proceed; just be prepared for a broader baseline and more tuning noise. For segmentation design context, see How to Implement Zero Trust Network Access in Cyber Security.

Prerequisites and stop-here warnings

Before you start, confirm that your network visibility and data retention are sufficient to support internal traffic analysis.

Minimum prerequisites

- Internal network telemetry from a span port, tap, flow records, firewall logs, or NDR sensor
- Time synchronization across sensors, hosts, and log sources
- Asset inventory with hostnames, roles, IP ranges, and business-critical segments
- A way to map source and destination pairs over time
- A ticketing or case workflow for triage and follow-up

Stop-here-if warnings

Stop if any of the following are true:

- you only collect north-south perimeter traffic and have no east-west visibility
- timestamps are not reliable across sources
- internal NAT obscures the true source host and you cannot correlate it to endpoint telemetry
- encryption prevents useful metadata collection and you have no flow or connection logs
- you cannot distinguish servers, user endpoints, and management systems in inventory

Without these basics, lateral movement detection will generate too many false leads to be operationally useful.

Validation before proceeding

Pick a small internal segment and confirm you can answer these questions from your telemetry:

- Which host initiated the connection?
- Which internal host received it?
- What port, protocol, and session timing were used?
- Is the source a workstation, server, or admin host?
- Has this source-destination pair been seen before?

If you cannot answer at least four of the five, fix the collection and enrichment layer first.

Step 1: Build a baseline of normal internal traffic

Goal

Separate normal operational patterns from suspicious pivots. You need a baseline before anomaly rules become meaningful.

Action

Collect a representative period of internal traffic, ideally covering business hours, maintenance windows, and at least one change cycle. Group the data by source asset, destination asset, destination port, protocol, and connection frequency.

Focus on questions such as:

- Which systems regularly talk to domain controllers, file servers, database servers, or management tools?
- Which subnets initiate the most east-west connections?
- Which ports are common between specific roles?
- Which hosts only communicate with a small set of peers?

A simple baseline table often helps more than complex scoring at the start:

Expected output

You should have a reference model that tells you what is ordinary for each role and subnet.

Validation

Check a random sample of known-benign events against the baseline. If common admin activity appears anomalous, the baseline is too narrow or the asset labeling is wrong.

Common failure

The most common mistake is baselining on too short a window. That creates false positives when backups, patching, or maintenance tasks start generating unusual internal traffic.

Step 2: Identify lateral movement patterns in traffic

Goal

Create a detection hypothesis for traffic that does not fit normal internal behavior.

Action

Look for patterns that are strong indicators of internal pivoting or reconnaissance. The most useful ones are usually behavioral rather than protocol-specific:

- one source host touching many internal destinations in a short period
- a workstation connecting to server-only ports on multiple hosts
- rare use of remote management protocols from non-admin endpoints
- authentication-heavy bursts to several systems after a single compromise event
- short-lived connections to many ports on the same host, suggesting internal scanning
- unusual access to file shares, admin shares, or remote shell services
- new source-destination pairs in segments that normally do not interact

Network traffic alone rarely proves malicious intent. It provides evidence that should be corroborated with endpoint, authentication, and identity context. If you need a disciplined way to validate identity and suspicious communication patterns, the same method used to inspect routing and authentication signals in mail can be useful as a thinking model; a comparable evidence-first workflow is covered in How to Detect Phishing Attacks Using Email Header Analysis.

Expected output

You should have a short list of traffic behaviors that are unusual for your environment and plausible for lateral movement.

Validation

Test each hypothesis against a benign event set. For example, compare the pattern of a patch-management job, a software deployment, and an actual administrator session. A useful hypothesis should flag the suspicious pattern while leaving routine automation explainable.

Common failure

A frequent mistake is treating ?internal traffic? as inherently suspicious. Most enterprise environments have plenty of legitimate east-west movement. The signal comes from rarity, role mismatch, and sequence, not from internal traffic by itself.

Step 3: Enrich traffic with asset and identity context

Goal

Reduce ambiguity so you can tell compromise-driven movement from normal administration.

Action

Add context to each suspicious flow or session:

- asset role and ownership
- user logon activity near the event time
- administrative group membership

- known management jump hosts
- business function of the destination system
- vulnerability exposure or recent changes on the source or destination

This enrichment is what turns a raw flow record into a useful detection candidate. A workstation connecting to SMB on a file server may be normal; a finance endpoint connecting to SMB on six servers it has never touched is much more interesting.

Expected output

Each candidate event should carry enough context to answer whether the source is authorized to talk to the destination in that way.

Validation

Inspect several known-good events and confirm that the enrichment prevents obvious misclassification. If the data cannot distinguish user endpoints from admin jump hosts, fix inventory tagging or directory integration before tuning detections.

Common failure

False positives often come from missing role labels. If every host is just an IP address, the detection logic cannot account for standard operational exceptions.

Step 4: Write practical detection rules or hunt queries

Goal

Turn the baseline and hypotheses into repeatable detection logic.

Action

Start with simple, explainable conditions. For example:

- alert when a non-admin host connects to three or more internal systems on a remote management port within five minutes
- alert when a workstation initiates SMB or WinRM sessions to servers it has never contacted before
- alert when a single source touches many destinations on one port and the destination count exceeds its normal baseline
- alert when there is a sudden spike in internal authentication attempts followed by remote service connections

A usable rule should prefer precision over coverage at first. You can widen it later.

Example logic for a hunt query, expressed generically:

Expected output

You should have a small set of candidate detections or hunts that can run on your telemetry platform or SIEM.

Validation

Run the logic on historical data and inspect the top results manually. A good first-pass detection should produce understandable hits, not a wall of noise.

Common failure

A rule that fires on volume alone will often catch backup systems, monitoring tools, and patch orchestration. Add role, time window, destination type, and source history to keep the signal useful.

Step 5: Validate suspicious traffic with corroborating evidence

Goal

Decide whether the traffic truly suggests lateral movement before escalating.

Action

For each candidate, check:

- whether the source host had a legitimate reason to connect
- whether the account in use matches the host?s normal role
- whether the destination is a management target or a user system
- whether the event aligns with login activity, service creation, or unusual process launches
- whether the same source later reached additional systems

One strong indicator is sequence. A small scan, followed by authentication attempts, followed by remote service use, is much more consistent with movement than a single odd connection.

Expected output

You should be able to classify each hit into one of three buckets: benign, suspicious but explained, or likely malicious.

Validation

A reviewed case should include the time window, source and destination list, ports, frequency, and the contextual evidence that led to the decision. If you cannot explain the conclusion in a case note, the detection is not yet operationally ready.

Common failure

Do not validate only with packet or flow data if endpoint and identity logs are available. Lateral movement is a multi-signal problem, and network telemetry alone usually cannot confirm the actor or the mechanism.

Step 6: Tune, suppress, and operationalize

Goal

Make the detection durable enough for daily operations.

Action

After you confirm a detection is useful, tune it with explicit exceptions rather than broad exclusions. Examples include:

- known admin jump hosts
- approved patching windows
- sanctioned automation accounts
- managed backup systems

- approved remote support channels

Document each suppression so it can be audited later. If a suppression is temporary, give it an expiration date.

Expected output

You should end up with a detection that is actionable, a list of approved exceptions, and a clear triage path for new alerts.

Validation

Re-run the rule after tuning and confirm that benign operational activity is reduced without removing the suspicious cases you actually care about.

Common failure

The usual failure here is over-suppression. If every noisy case is permanently exempted, the rule stops detecting the behavior it was meant to find.

What good results look like in practice

A useful lateral movement detection does not just say ?odd internal traffic.? It gives you a concise story:

- a non-admin workstation started connecting to multiple internal servers on a remote service port
- the source had not previously contacted those destinations
- the activity occurred outside the expected support window
- authentication or host logs show a matching burst of access attempts
- the behavior continued to other systems after the first connection

That combination is usually enough to justify escalation, deeper investigation, or containment.

Operational follow-up and production checks

Before you rely on the workflow in production, verify three things.

First, confirm coverage. If a segment is not monitored, the absence of alerts is meaningless.

Second, confirm ownership. Someone needs to maintain baselines, exceptions, and asset labeling as networks change.

Third, confirm escalation criteria. Analysts should know which combinations of traffic, identity, and endpoint evidence warrant immediate incident handling.

If you want to prioritize which exposed systems or services deserve attention first, pairing network detections with vulnerability analysis can improve triage speed. In some environments, the best starting point is the set of hosts that are both highly exposed and operationally important, which is the approach used in How to Detect and Prioritize Critical Vulnerabilities.

Final check before production use

A lateral movement detection workflow is ready when it consistently identifies suspicious internal pivots, explains why each alert fired, and leaves you with enough evidence to act without guessing.

If you can answer these questions confidently, you are in good shape:

- Do we know what normal internal traffic looks like by role and segment?
- Can we spot rare remote service use, scan-like behavior, and unusual pivots?
- Can we enrich the traffic with host and identity context?
- Have we validated the logic against known-benign activity and tuned the obvious exceptions?
- Do we have a documented path from alert to investigation to containment?

If the answer to any of those is no, fix that gap before treating the detection as production-ready. The goal is not to observe more traffic; it is to reliably turn internal traffic patterns into evidence that helps you catch movement before it becomes a broader compromise.

Use this guidance together with Oracle privilege escalation detection to connect the workflow with related operational context already available on the site.