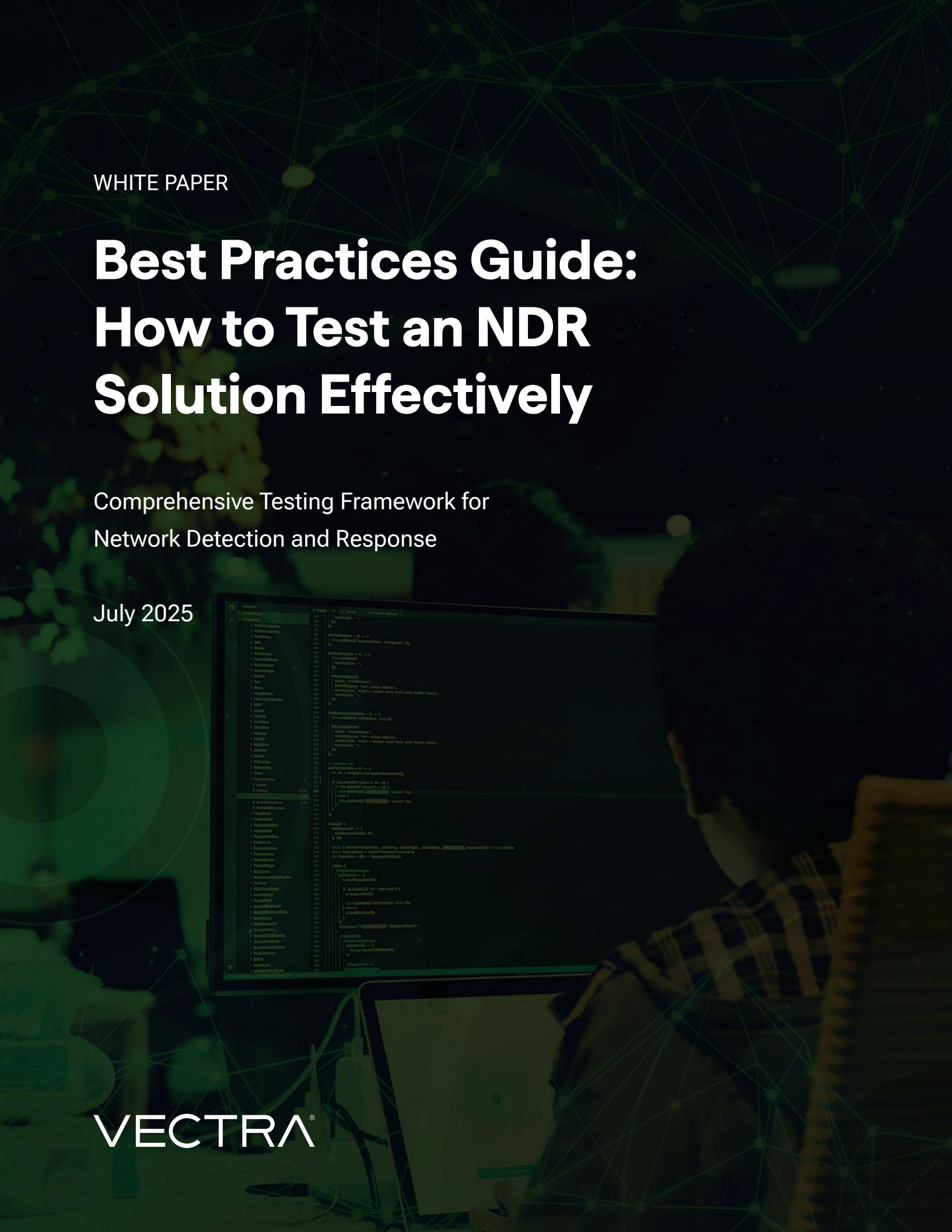
WHITE PAPER

Best Practices Guide: How to Test an NDR Solution Effectively

Comprehensive Testing Framework for Network Detection and Response

July 2025



VECTRA®

Table of Contents

Introduction

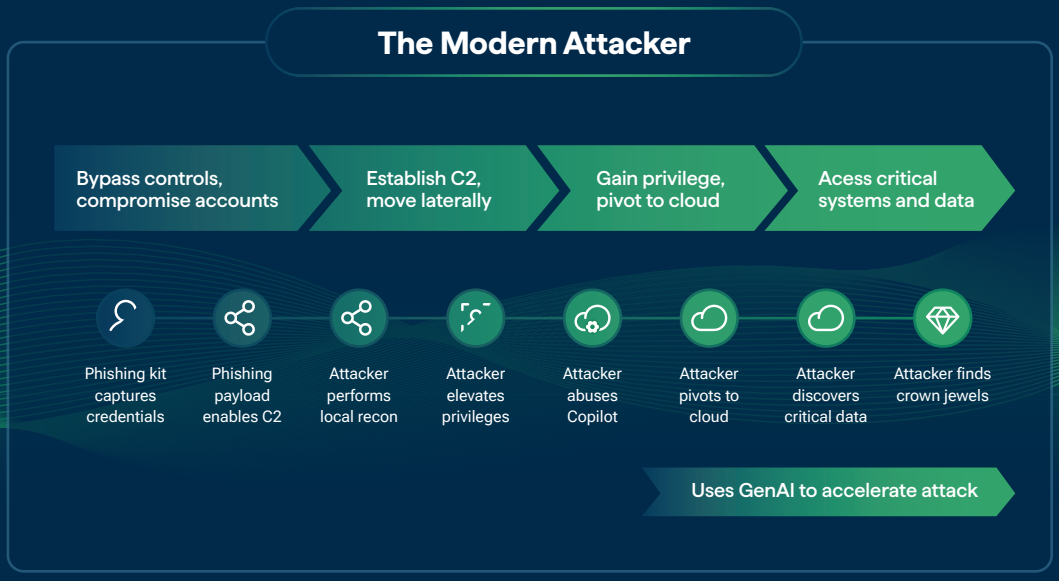
Network Detection and Response (NDR) solutions are designed to detect, correlate, and prioritize security incidents across the entire cyberattack lifecycle. Unlike tools that alert on isolated events, An NDR solution must analyze behavioral patterns across various entities and correlate them to identify and escalate critical threats. Its scope extends beyond just tracking hosts (IP-connected devices); it can also prioritize user accounts that exhibit suspicious or risky behavior.

Ultimately, conducting isolated tests that don't reflect a realistic attack progression provides limited insight into an NDR solution's effectiveness. For example, if an APT-style attack involves 10 distinct steps, what truly matters is not whether the NDR detects every single one, but whether it gathers enough signal to reliably identify and alert on the overall threat activity. A perfect 100% detection rate is unrealistic—what's essential is the solution's ability to surface and correlate meaningful indicators that expose the attacker's presence.

To accurately assess an NDR product, tests must reflect the progression of real-world attacks, not isolated or simplistic behaviors. A valid evaluation should simulate multiple stages of the attack kill chain, ideally using real tools, tactics, and procedures (TTPs) that mirror actual threat actor behavior.

Key Principle

A comprehensive test must cover the entire kill chain—from initial access and command and control (C2), through lateral movement, and ending with data exfiltration.



High-Level Guidance for Testing NDR

1. Realistic red team scenarios should reflect a cohesive attack flow that mirrors how adversaries operate.

2. Ensure tests reflect attack progression (TTPs across MITRE ATT&CK stages).

3. Ensure each phase of the attack simulation builds upon the previous one. The information gathered at each stage (e.g., reconnaissance, credential harvesting) should enable logical progression to the next step. Avoid isolated test steps.

4. Do not rely on simplistic, standalone actions (e.g., a port scan or single exploit).

5. Use real attacker frameworks and tools (e.g., Cobalt Strike, Sliver, Impacket).

6. Simulate real traffic and timing, including sleep beacons and low-and-slow behaviors.

7. Include environmental context: active users, internal assets, normal background traffic.

Before You Start

Consider your environment carefully! Use production where possible, or simulate a realistic enterprise network in test environments. Setting up new accounts for the adversary emulation is also not recommended as nothing could be learned if not used.

Measuring Success

Adversary emulation exercises are only as valuable as your ability to measure their outcomes. Success is not just about generating alerts. it's about how well the NDR solution detects, prioritizes, and enables a timely response to real threats.

Focus on MTTD and MTTR

Adopt industry-standard metrics such as:

MTTD
(Mean Time to Detect)
How long does it take the NDR solution to surface meaningful alerts once attacker activity begins?

MTTR
(Mean Time to Respond)
Once detected, how quickly can your team investigate, validate, and respond to the threat?

These metrics provide quantifiable insight into how well your detection and response process is working—not just whether the solution triggers an alert.

Go Beyond "Did It Alert?"

It's not enough to ask "Was this activity detected?" Instead, ask:

Did the NDR solution prioritize the threat appropriately?

Was the attacker activity surfaced clearly and early, or buried among low-fidelity noise?

Did detection support timely investigation and escalation, or was it missed or deprioritized?

Many teams fall into the trap of searching for buried alerts post-simulation—treating delayed discovery as a detection win. But if the signal is not actionable in real time, it might as well not exist.

Detection that is lost in noise or flagged with low confidence is operationally useless.

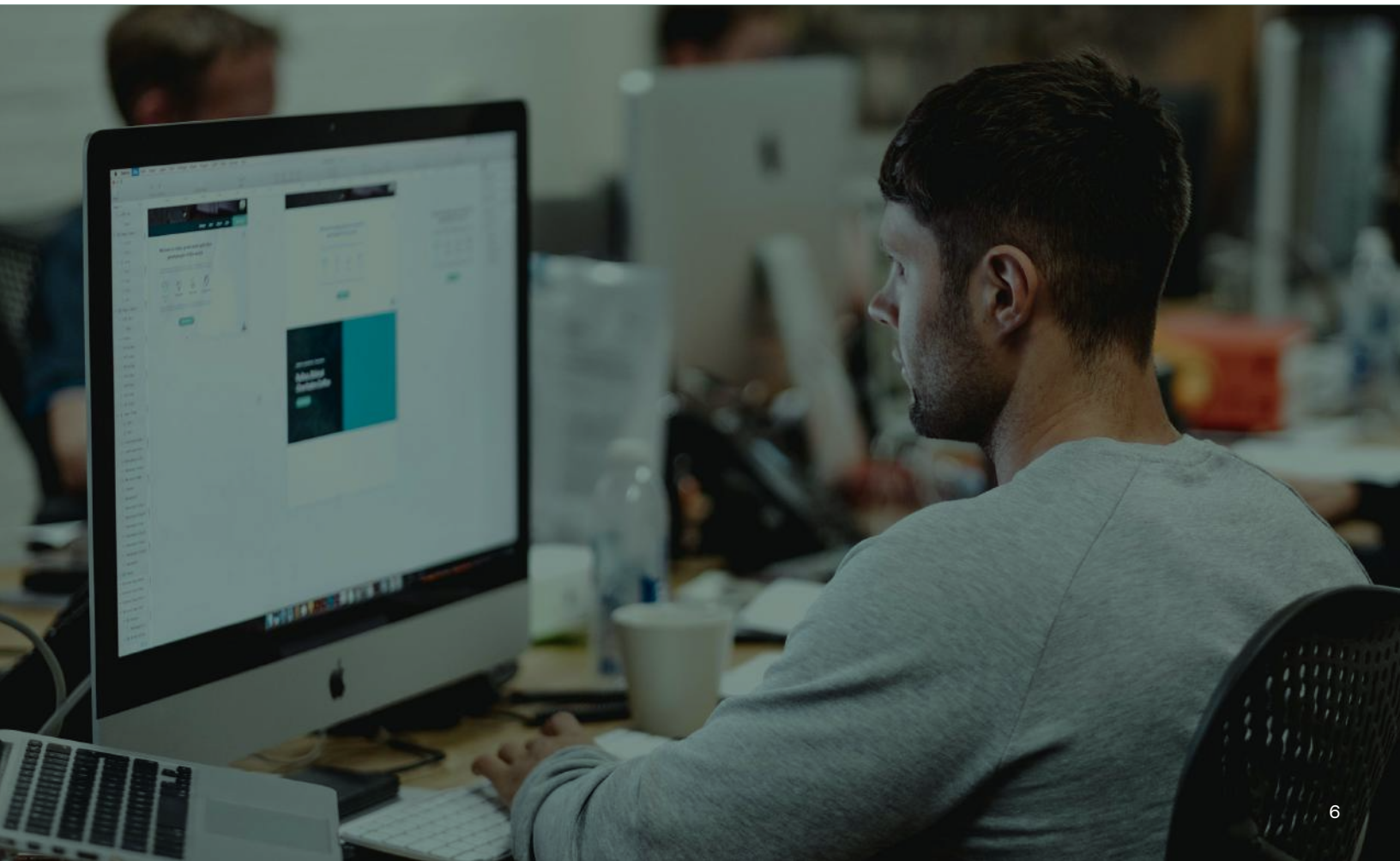
Evaluating the Signal

A strong NDR solution should:

Correlate attacker activity across phases, not trigger isolated alerts.

Elevate high-risk entities, so that a compromised host or account stands out clearly.

Suppress background noise, ensuring that relevant detections are not diluted.



Adversary Emulation

Command and Control (C2)

☆ Goal ☆

Assess NDR's ability to detect stealthy C2 channels used by sophisticated adversaries.

Most targeted attacks involve command and control (C2) activity shortly after initial access. In fact, an estimated 60–80% of sophisticated attacks leverage C2 at some stage—especially in ransomware campaigns, advanced persistent threats (APTs), and supply chain compromises. Since NDR solutions typically assume a compromise has already occurred, they focus on detecting post-compromise behavior, not the initial intrusion itself.

Moreover, C2 evasion techniques (such as DNS tunneling, use of encrypted or covert channels) are now standard practice among threat actors, significantly increasing the difficulty of detection. This reinforces the importance of simulating realistic, stealthy C2 activity when evaluating an NDR solution's effectiveness.

Test Setup Recommendations

- Use CDNs or reputable hosting providers to simulate trusted infrastructure (Always Externally hosted!)
- Acquire a domain with a high reputation score to bypass category filters (e.g. The Domain Robot).
- Use modern and popular C2 frameworks (The C2 Matrix).
- Generate trusted TLS certificates using services like Let's Encrypt.

Execution

- Simulate beaconing with realistic sleep intervals (30–60 seconds).
- Execute post-compromise activities over the tunnel for at least 20–30 minutes.
- Vary protocols: HTTPS, HTTP, DNS.

Do

- Simulate realistic, ongoing attacker behavior.
- Customize C2 profiles to evade default detections.
- Measure detection timing, alert fidelity, and correlation depth.

Don't

- Use idle or inactive tunnels (no traffic = no detection).
- Rely on threat intel or user-agent rarity alone for detection.
- Test only one phase (e.g., beaconing without post-access activity).

Network Reconnaissance

☆ Goal ☆

Evaluate NDR's ability to detect early-stage network mapping and enumeration.

Network reconnaissance is how attackers learn about the environment post-access. They scan for live hosts, open ports, and services to plan lateral movement and privilege escalation.

Tools to Use

- Nmap (port scanning, OS fingerprinting)
- Responder (LLMNR, NBT-NS poisoning)
- NetScanTools, Angry IP Scanner, manual scripts

Common Activities

- Scanning subnets for active hosts
- Enumerating open ports and common services
- Capturing broadcast traffic, probing for weak protocols.

Simulate activity gradually across IP ranges and networks to reflect real threat actor behavior.

Domain Reconnaissance

☆ Goal ☆

Test NDR detection of internal AD reconnaissance.

After network mapping, attackers turn to Active Directory to locate high-value targets. This includes enumerating users, groups, policies, and trust relationships.

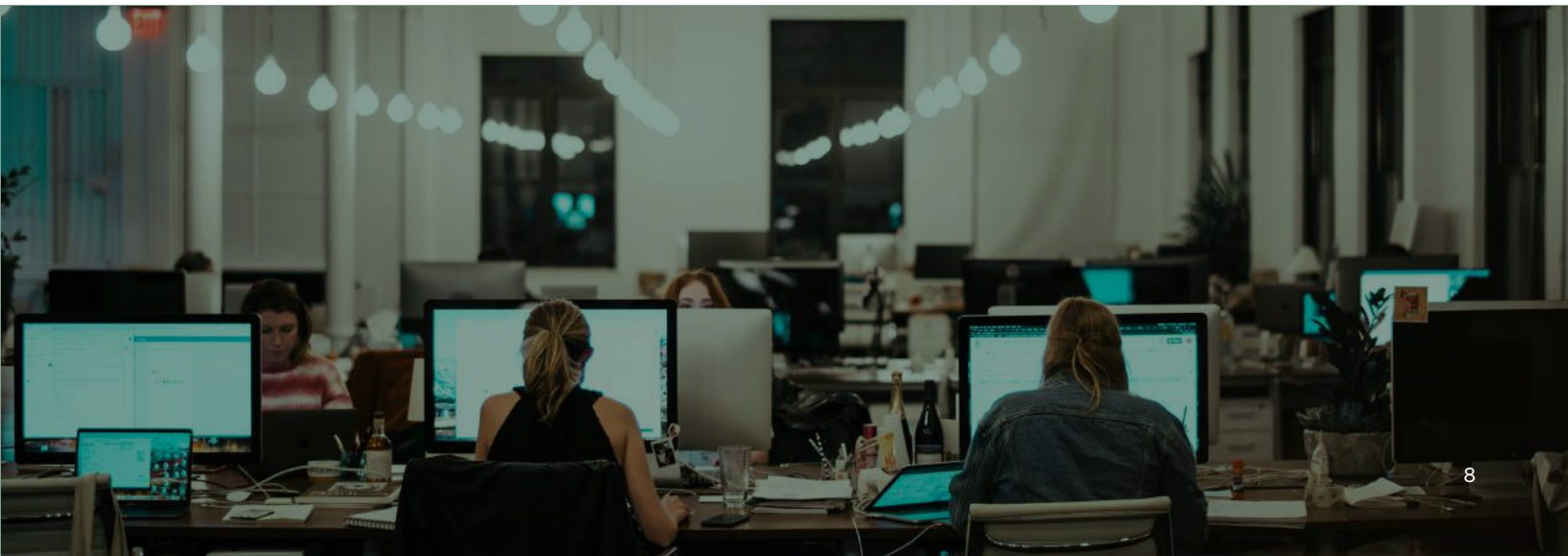
Tools to Use

- ADRecon
- BloodHound & SharpHound
- ldapsearch
- Rubeus

Common Activities

- Enumerating domain users, groups, and computers
- Querying group policies, OUs, and trust relationships
- Uncovering Kerberoastable accounts via SPN requests

Simulate over time and across multiple hosts to emulate realistic post-compromise behavior.



Lateral Movement Detection

☆ Goal ☆

Evaluate how well the NDR detects attacker movement inside the network.

Lateral movement involves navigating laterally between systems to escalate privileges or access sensitive data. It often mimics admin behavior and relies on existing credentials.

Common Tools

- Netexec (supports WMI, PsExec)
- LOLBins (e.g., wmic, psexec, schtasks, etc.)
- Remote Desktop Protocol (RDP)
- Impacket (e.g., wmiexec.py, smbexec.py)

Ensure tests simulate logins, session reuse, and native tool abuse (“Living off the Land”).

Data Exfiltration Detection

☆ Goal ☆

Validate NDR’s ability to detect outbound data theft.

Data exfiltration is the final and often most damaging stage of an attack. NDR must detect stealthy data movement—especially when attackers use encrypted or obfuscated channels.

Simulate the Following

- Exfil over HTTPS/HTTP (blend with legitimate traffic)
- Exfil via DNS tunneling (e.g., dnscat2)
- Exfil to cloud services (e.g., OneDrive, Dropbox, Google Drive via rclone)

Ensure large enough volumes or chunked transfers to trigger behavioral/statistical models.

Infrastructure

A comprehensive, realistic environment is key to valid NDR testing. The closer your setup mirrors an enterprise network, the more actionable your results.

Environment Requirements

- Active Directory with real users and accounts
- DNS, DHCP, VPN, and endpoint agents
- Command and control infrastructure with redirection and encryption
- A mix of host roles: domain-joined, isolated, servers, workstations

If production isn't viable, use a well-structured test environment.

Vectra offers access to a Cyber-Range sandbox lab, a safe, enterprise-like environment ideal for full attack simulations and controlled testing.

Common Mistakes to Avoid

- Using an empty or overly simple lab
- Ignoring NDR's learning or baselining period
- Testing without normal traffic and user behavior

Critical Note

Most NDR platforms use machine learning—including unsupervised models—which require a training period to learn the environment. Skipping this period can produce misleading test results.

Always ask your vendor about learning time and configuration prerequisites. This is super critical when testing AI-driven NDR platforms.

Specialized Expertise and Internal Red Teams

Testing NDR solutions effectively is complex and resource-intensive.

It requires:

- Deep understanding of attacker behaviors, Adversaries and tooling
- Ability to replicate stealthy, evasive techniques
- Infrastructure that mirrors enterprise networks

At Vectra, for example, an independent internal red team runs advanced simulations that test detection and evasion across the stack—including EDR, firewalls, and proxies.

Ask your Vectra representative for support in designing realistic and impactful scenarios that is relevant for the industry and environment.

Whether using internal or external red teams:

- Respect the NDR's learning and tuning phases
- Ensure test traffic is labeled and validated for post-analysis
- Make sure stakeholders understand: NDR testing ≠ full-stack security testing

Expanding Red Team Scenarios

NDR solutions can also be evaluated against alternative attack paths that deviate from traditional C2-driven campaigns.

One valuable scenario involves a red team placing a rogue device (e.g., a drop box) inside the network that is accessed over an out-of-band channel such as LTE. In such cases, no C2 channel is required initially, yet the attacker can still perform reconnaissance, lateral movement, and exfiltration phases. These behaviors remain detectable and relevant for NDR evaluation.

Another important scenario starts with the attacker logging in directly using stolen credentials—perhaps obtained through a prior compromise, phishing, or brokered access. In these cases, C2 infrastructure might be optional at the onset, but once inside, the adversary still progresses through traditional kill chain stages. Even if the attacker enters through a VPN, they will often eventually deploy a remote access tool (RAT) or establish C2 for persistence in case their remote access is interrupted.

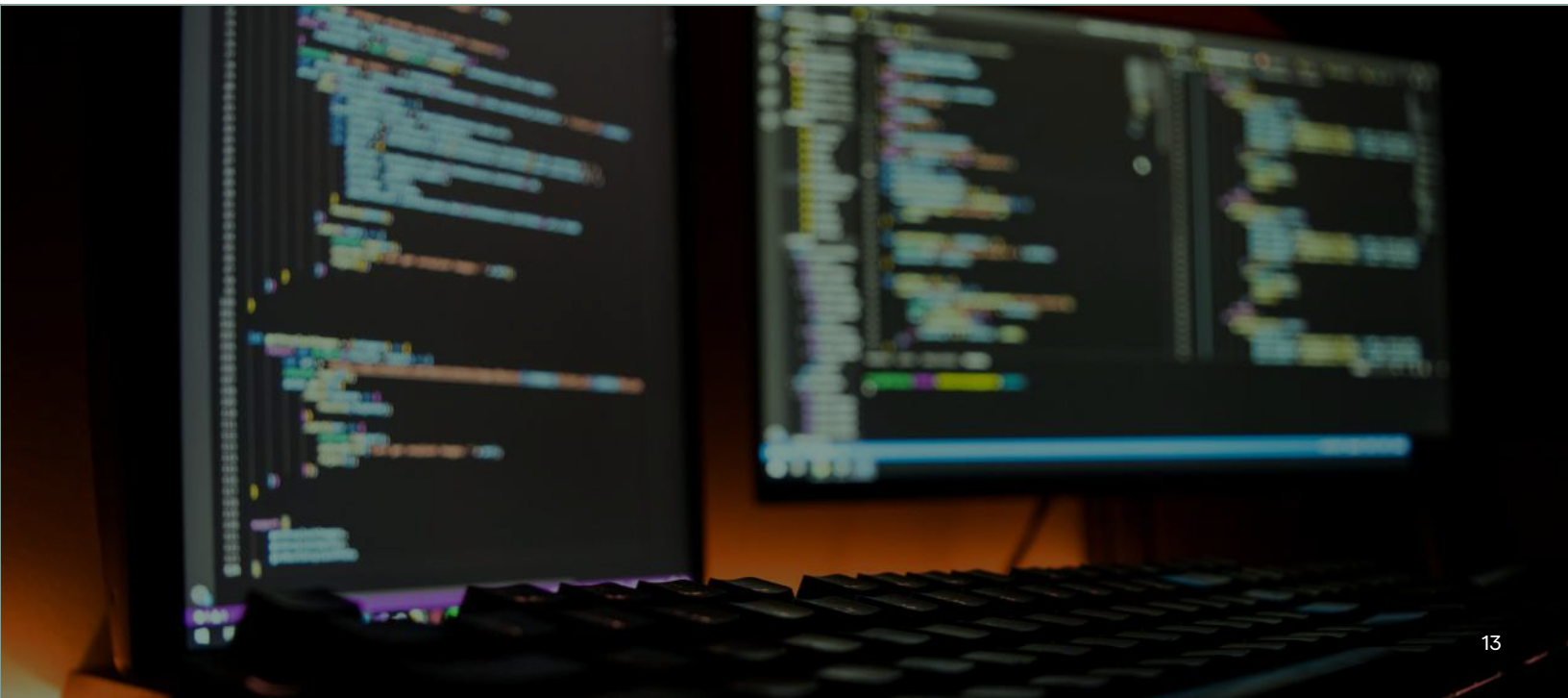
These non-C2 scenarios provide an opportunity to evaluate the NDR solution's ability to detect attack stages based on behavioral anomalies, lateral activity, and data movement—rather than relying solely on C2 beacon detection.

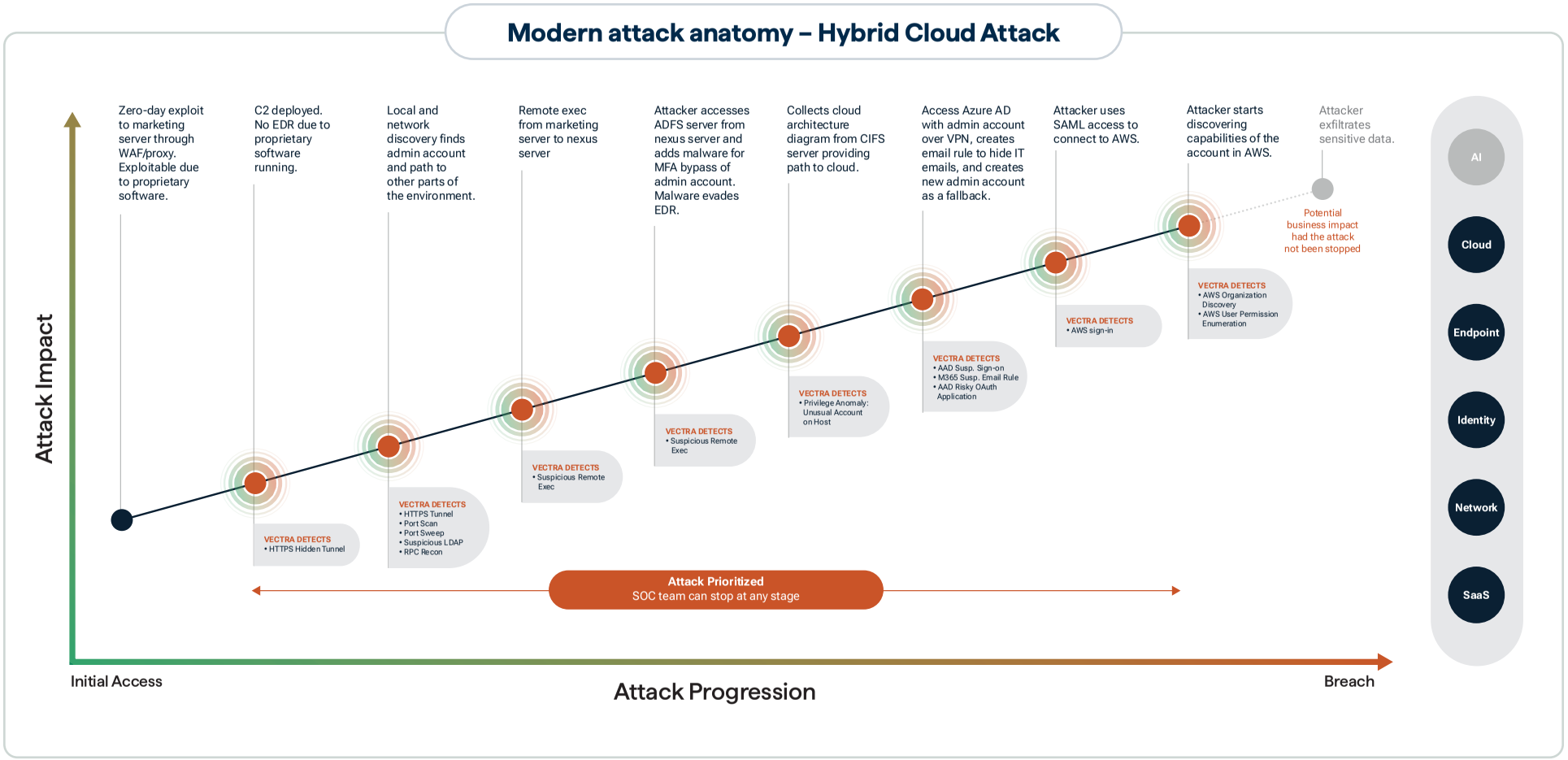
Beyond NDR: Attacking the Modern Hybrid Environment

Modern enterprise networks extend far beyond traditional on-premise infrastructure. While NDR remains a critical component for detecting threats within the network perimeter, today's attackers increasingly exploit a broader attack surface - including cloud platforms, SaaS applications, and identity systems (e.g. Microsoft Entra ID).

Adversaries don't stop at stealing local credentials or escalating privileges within the domain. From a compromised machine, attackers commonly extract browser-stored tokens, refresh tokens, and session cookies to gain persistent access to cloud services like Microsoft 365, Entra ID (Azure AD), and other identity providers. This enables lateral movement across cloud boundaries, bypassing traditional perimeter defenses entirely.

Visibility and detection capabilities must extend beyond the network layer to track attacker activity as it pivots through identity, API, and SaaS layers. If your red team scenarios stop at the edge of your internal network, you're missing a major component of the modern attack chain.





For expert guidance on emulating modern adversaries and expanding detection beyond the network layer, reach out to your Vectra representative. Our specialists can help design red team engagements that reflect the full scope of today's attack surface.

This guide provides a comprehensive framework for testing Network Detection and Response solutions effectively. For additional support and specialized testing environments, consult with your NDR vendor or security testing specialists.

For more information, contact us

About Vectra AI

Vectra AI, Inc. is the cybersecurity AI company that protects modern networks from modern attacks. When modern cyber attackers bypass existing controls, evade detection and gain access to customers' data center, campus, remote work, identity, cloud, and IoT/OT environments, the Vectra AI Platform sees their every move, connects the dots in real-time, and stops them from becoming breaches. With 35 patents in AI security and the most vendor references in MITRE D3FEND, organizations worldwide rely on Vectra AI to see and stop attacks their other tools can't. For more information, visit www.vectra.ai.

For more information please contact us:
Email: info@vectra.ai | vectra.ai