HORIZON3.ai

~~TRUST BUT~~ VERIFY

WHITE PAPER

FROM HONEYPOTS TO ACTIVE DIRECTORY TRIPWIRES

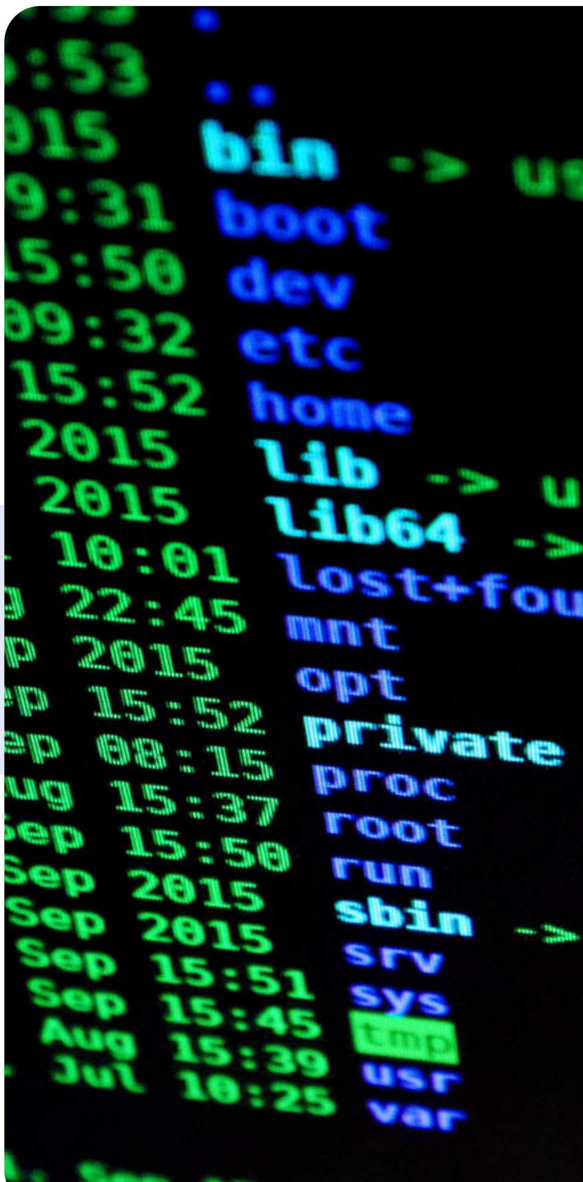
THE EVOLUTION OF DECEPTION IN CYBERSECURITY

CYBERSECURITY HAS ALWAYS BEEN A CONTEST OF VISIBILITY

Defenders install controls and monitoring systems, attackers look for blind spots. For decades, one of the most compelling ideas for defenders was simple: create a trap. If an attacker interacts with something that no legitimate user ever should, you know immediately that something is wrong. This is the idea behind deception in cybersecurity.

It started with honeypots—full systems set up as decoys. Later, the concept was refined into honey tokens—small pieces of fake data designed to trip alarms if touched. Each step brought promise but also new limitations. Honeypots were too heavy and costly, honey tokens too uncertain and difficult to operationalize.

Today, the same idea has reemerged in a far more practical form. NodeZero Tripwires™ and the introduction of Active Directory (AD) Tripwires apply the lessons of the past, solve the limitations of early deception technology, and deliver high-fidelity detection in places where attackers can least afford to be caught. To understand why this matters, it helps to trace the history of deception in security.



The Rise of Honeypots

The idea of luring an attacker into a decoy system is nearly as old as networked computing itself. Honeypots emerged in the 1990s as one of the first practical ways to observe attackers in real time. The concept was popularized by researchers such as Clifford Stoll in The Cuckoo's Egg (1989), and later formalized through organized efforts such as the Honeynet Project, founded by Lance Spitzner around 1999–2000.

A honeypot was typically a system or network deliberately exposed and designed to look attractive to intruders. Once an attacker engaged, defenders could monitor every move, study tactics, and sometimes even trace the origin of the attack.

For researchers, honeypots were invaluable. They provided forensic insight and a window into adversary behavior that no log analysis or intrusion detection system could provide at the time. They helped build early knowledge about malware, worms, and attacker tradecraft.

But honeypots had serious drawbacks:

HIGH COST AND COMPLEXITY
Running a convincing honeypot meant standing up and maintaining fulloperating systems and networks.

SCALABILITY ISSUES
One or two honeypots could be managed, but deploying them widely across an enterprise was unrealistic.

LIMITED DETECTION COVERAGE
Honeypots were usually isolated. If attackers did not interact with them, they provided no visibility at all.

RISK OF MISUSE
Poorly configured honeypots could be hijacked and used to launch further attacks.

Honeypots proved that deception was useful, but they were too heavy to become a standard enterprise defense.

HONEYPOT LESSONS

Valuable for research, not scalable for enterprise defense.

High maintenance costs, limited adoption.

Insightful but reactive, offering little proactive protection.

From Honeypots to Honey Tokens

In the early 2000s, researchers began asking: what if the bait didn't need to be a whole system? What if it could be a single piece of data?

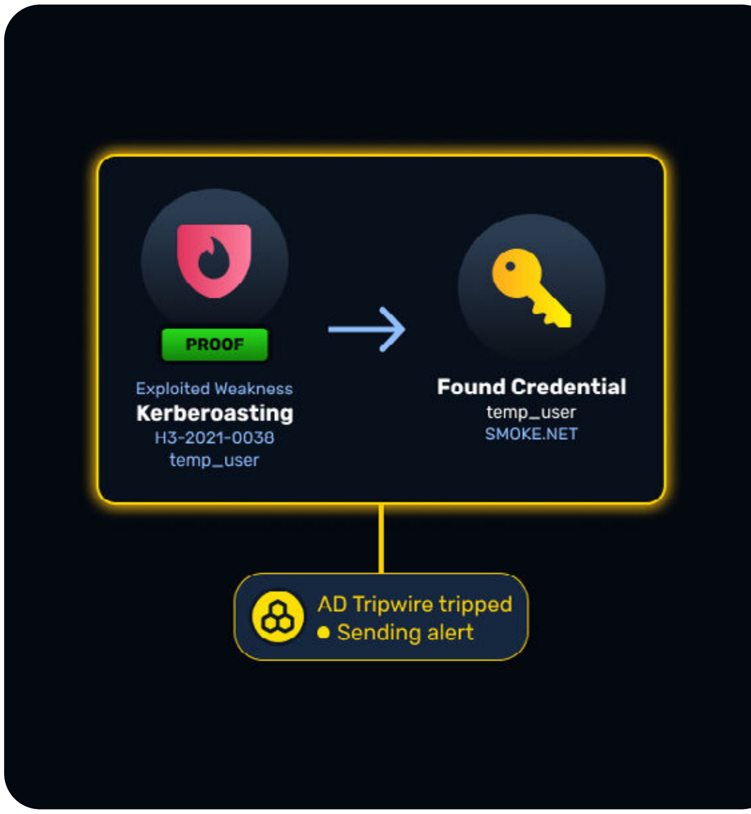
This led to the development of honey tokens, a term widely credited to Lance Spitzner of the Honeynet Project around 2003. A honey token could be a fake username and password, a dummy credit card number, a phony spreadsheet, or even an email address. The key idea was that the token had no legitimate use. If anyone touched it, it meant something suspicious was happening.

Honey tokens were attractive because they solved some of the problems honeypots could not:

- They were lightweight and easy to create.
- They could be distributed widely across an organization.
- They required fewer resources to maintain.

Common use cases emerged quickly. Companies planted dummy customer records to detect insider abuse. Security teams generated fake cloud access keys to catch external attackers who gained file system access. Some organizations embedded bogus email addresses to detect spam campaigns or data leaks.

Compared to honeypots, honey tokens were more practical. They brought the promise of high-fidelity alerts without the overhead of running full decoy systems.



The Drawbacks of Honey Tokens

As honey tokens moved from concept to practice, their weaknesses also became clear. For all their simplicity, they often failed to deliver consistent, actionable results.

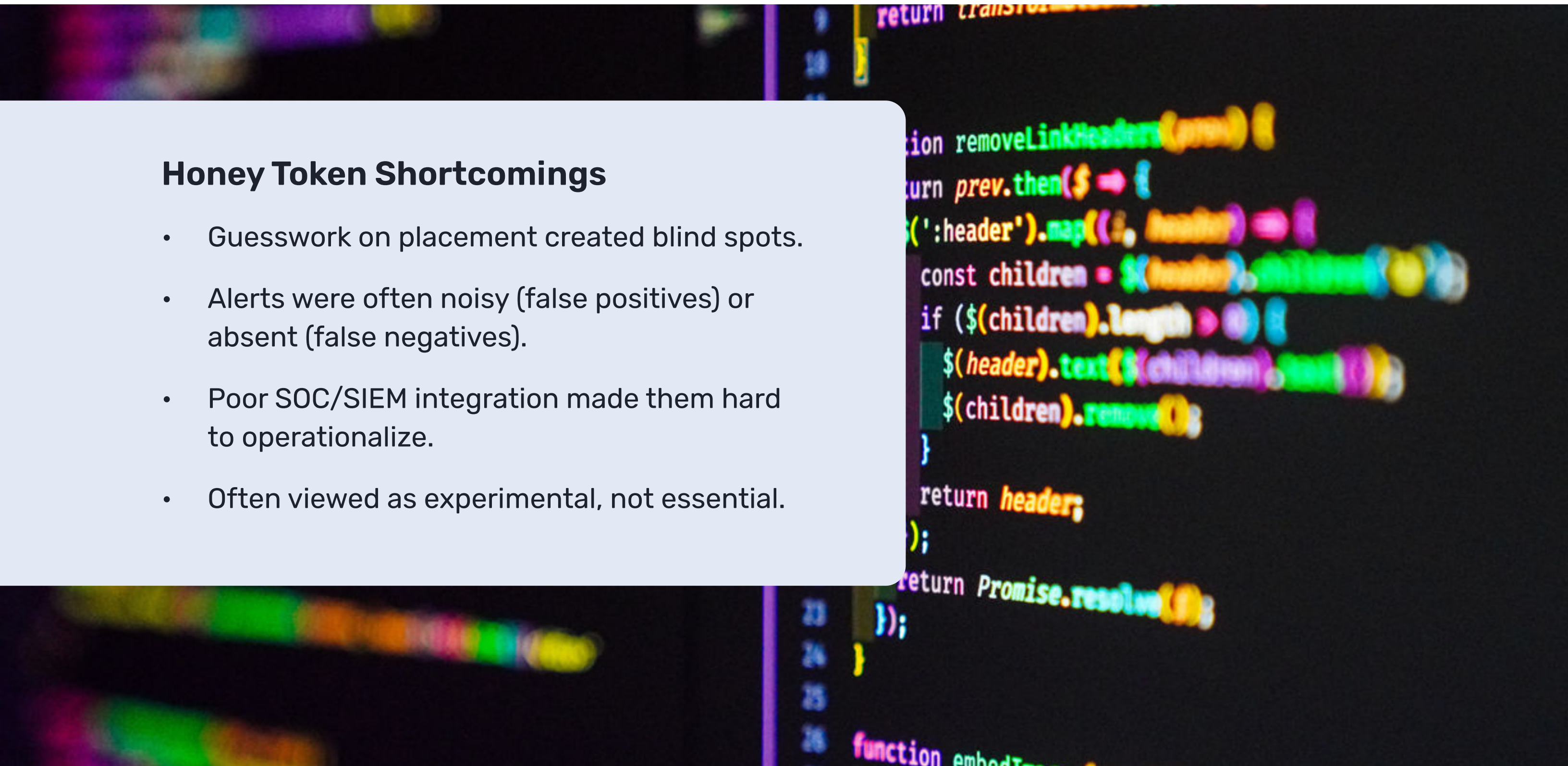
Placement Uncertainty	The first problem was where to put them. If a token was placed in a directory no attacker ever reached, it was useless. If tokens were scattered too widely, they generated clutter or became obvious to an intruder. Security teams were left guessing about coverage, with no guarantee attackers would ever trip one.
False Positives	Honey tokens were designed to never be touched by legitimate users, but in practice, that wasn't always true. IT staff sometimes stumbled across them during maintenance. Automated scripts or vulnerability scanners occasionally triggered them. Even backup systems or indexing tools could interact with tokens, generating noise.

HORIZON3.ai ~~TRUST BUT~~ VERIFY

False Negatives	Even worse, attackers often bypassed tokens altogether. A skilled adversary might target a very specific set of assets or credentials, never touching the decoys. When that happened, defenders were lulled into a false sense of security—assuming no alerts meant no threat.
Operational Overhead	Tokens had to be seeded, rotated, and monitored. At a small scale, this was easy. At enterprise scale, it became a challenge. Without automation, tokens went stale, or their alerts became disconnected from other parts of the security stack.
Integration Gaps	Honey tokens rarely integrated smoothly with SIEM, SOAR, or SOC workflows. An alert would fire, but it was often just a binary signal with no context. Analysts had to dig for details—who triggered it, how, and why. Without that context, honey tokens created more work rather than less.
Perception of Gimmickry	Finally, honey tokens were sometimes dismissed as clever tricks rather than serious tools. Because they often sat on the edges of security programs, they weren't prioritized or resourced.

For many organizations, they remained proof-of-concept experiments rather than core defenses.

Taken together, these drawbacks meant honey tokens delivered high signal in theory but unreliable value in practice.



Honey Token Shortcomings

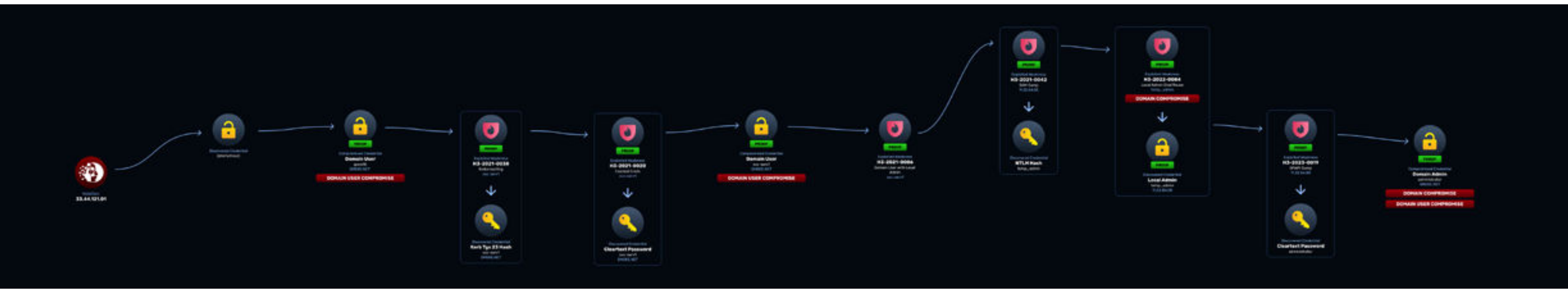
- Guesswork on placement created blind spots.
- Alerts were often noisy (false positives) or absent (false negatives).
- Poor SOC/SIEM integration made them hard to operationalize.
- Often viewed as experimental, not essential.

A New Approach to Deception

The shortcomings of honeypots and honey tokens didn't mean the idea of deception was wrong. It meant the execution was incomplete. What was missing was intelligence about where to place the decoys, how to tune them to attacker behavior, and how to integrate them into real defensive workflows.

Imagine if deception wasn't based on guesswork but on live knowledge of how attackers actually move through your environment. Imagine if decoys were placed precisely where attackers were likely to look next, not scattered randomly. Imagine if each alert came with full context, like what was touched, how it was placed, and what attack path it represented.

That is the shift that turns deception from a cool technology into a reliable detection strategy.



NodeZero Tripwires™

The NodeZero® Offensive Security Platform applies this new approach. NodeZero doesn't simulate attacks; it performs them safely inside customer environments. In doing so, it reveals how real attackers can gain access, move laterally, and escalate privileges.

NodeZero Tripwires extend these offensive insights into defensive detection. When NodeZero identifies a successful attack path, it can automatically deploy a decoy—a Tripwire—along that path. These may be fake credentials, secrets, or sensitive-looking files placed exactly where an intruder is most likely to explore next.

The difference from early honey tokens is clear:

- Intelligent placement. Tripwires are dropped on exploited hosts and vulnerable systems, not scattered at random.
- Context-rich alerts. Each alert shows where the Tripwire was placed, which weakness it defended, and what test led to its creation.
- SOC integration. Tripwire alerts can feed directly into SIEMs, giving analysts the context they need for immediate response.
- Customizable deployment. Security teams also have the flexibility to deploy tripwires to additional critical and/or NodeZero-recommended systems & data stores.

HORIZON3.ai ~~TRUST BUT~~ VERIFY

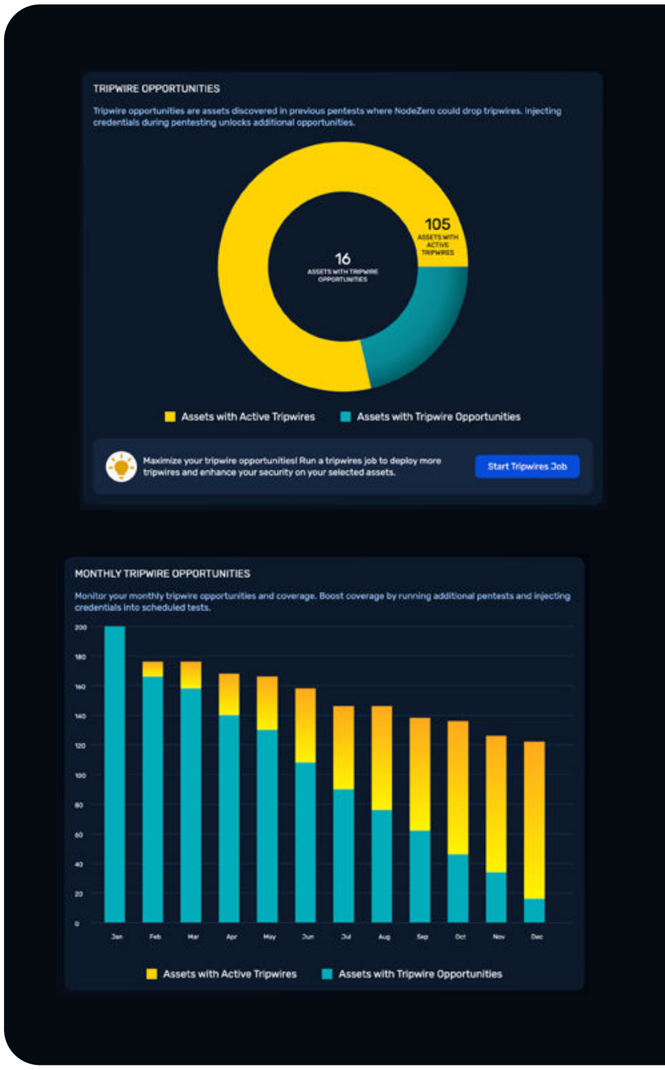
NODEZERO ACTIVE DIRECTORY (AD) TRIPWIRES

Why Active Directory is the Prime Target

Active Directory (AD) is one of the most common and critical technologies in enterprise environments. Nearly 90% of organizations worldwide rely on AD to manage authentication and authorization[1].

That ubiquity makes AD a universal attack surface. Research shows that nearly half of organizations have experienced AD attacks in the past two years, and more than 40% of those attacks resulted in successful compromise[1].

Sophos X-Ops research has found that attackers can escalate privileges into AD in as little as 11 hours following initial compromise[2]. Ransomware groups and advanced persistent threats alike rely heavily on this rapid escalation.


105
ASSETS WITH ACTIVE TRIPWIRES
16
ASSETS WITH TRIPWIRE OPPORTUNITIES
🟨 Assets with Active Tripwires
🟦 Assets with Tripwire Opportunities
Start Tripwires Job
200
180
160
140
120
100
80
60
40
20
0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
🟨 Assets with Active Tripwires
🟦 Assets with Tripwire Opportunities

TRIPWIRE OPPORTUNITIES

Tripwire opportunities are assets discovered in previous pentests where NodeZero could drop tripwires. Injecting credentials during pentesting unlocks additional opportunities.

Maximize your tripwire opportunities! Run a tripwires job to deploy more tripwires and enhance your security on your selected assets.

MONTHLY TRIPWIRE OPPORTUNITIES

Monitor your monthly tripwire opportunities and coverage. Boost coverage by running additional pentests and injecting credentials into scheduled tests.

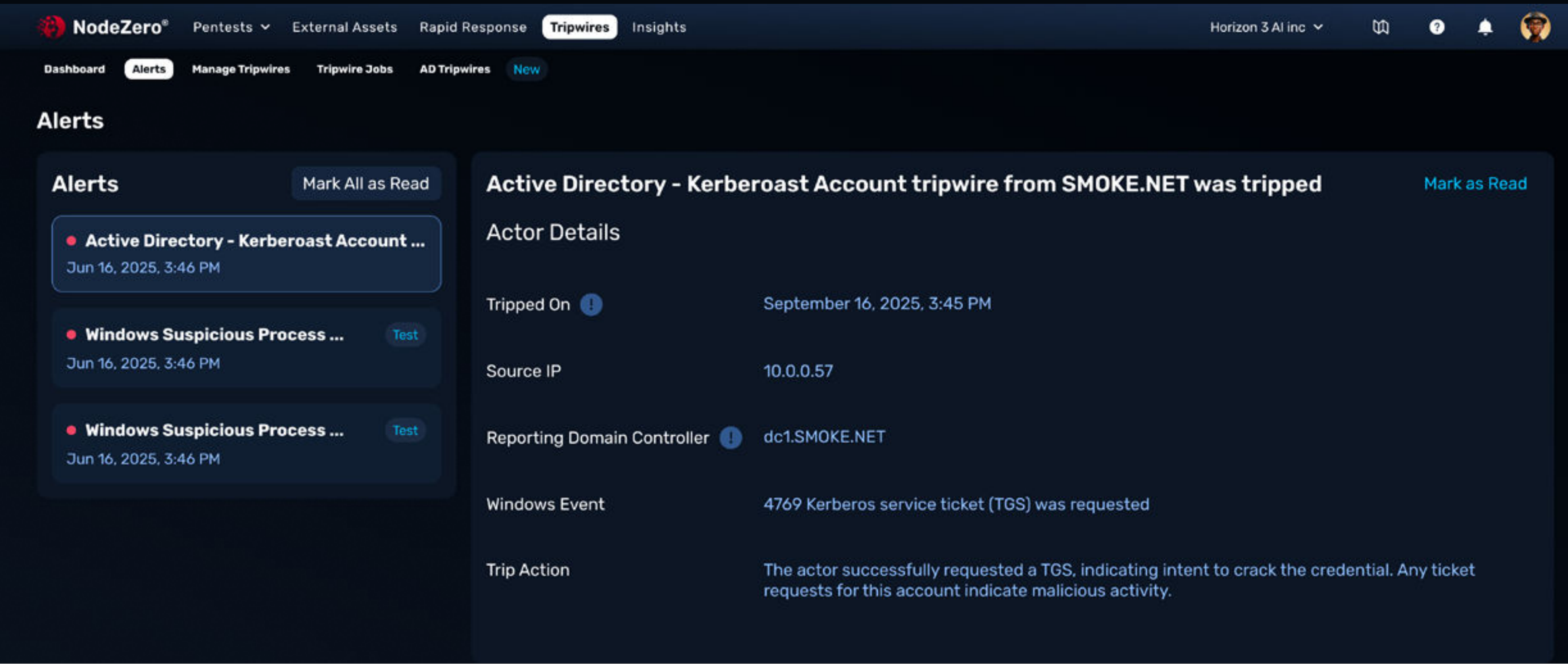
Why Traditional Monitoring Fails

AD is inherently difficult to defend. Its default settings are permissive, its trust relationships complex, and its support for legacy protocols creates exploitable vectors[3].

Common attack techniques such as kerberoasting, AS-REP roasting, and metadata scraping generate logs that often appear indistinguishable from normal user behavior. SIEM, EDR, and ITDR tools rely on correlation rules and thresholds, which bury subtle identity attacks in the noise of daily operations[3].

SOC analysts, already overwhelmed, rarely detect these attacks until it is too late. By the time a breach a has been detected, adversaries may already have domain admin rights, access to sensitive data, and established persistence for weeks or even months[3].

1. Semperis / Enterprise Management Associates (EMA). Rise of Active Directory Exploits: Is It Time to Sound the Alarm? 2023–2024 survey report. semperis.com
2. Sophos. It takes two: The 2025 Sophos Active Adversary Report. New Sophos X-Ops Research on What Happens After Attackers Breach a Company. 2025. news.sophos.com
3. Australian Signals Directorate (ASD), Australian Cyber Security Centre (ACSC), et al. Detecting and Mitigating Active Directory Compromises. September 26, 2024. cyber.gov.au



Alerts

Alerts

- Active Directory - Kerberoast Account ... Jun 16, 2025, 3:46 PM
- Windows Suspicious Process ... Test Jun 16, 2025, 3:46 PM
- Windows Suspicious Process ... Test Jun 16, 2025, 3:46 PM

Active Directory - Kerberoast Account tripwire from SMOKE.NET was tripped

Actor Details

Tripped On	September 16, 2025, 3:45 PM
Source IP	10.0.0.57
Reporting Domain Controller	dc1.SMOKE.NET
Windows Event	4769 Kerberos service ticket (TGS) was requested
Trip Action	The actor successfully requested a TGS, indicating intent to crack the credential. Any ticket requests for this account indicate malicious activity.

How AD Tripwires Work

AD Tripwires are real accounts created directly inside Active Directory. They are designed to look exactly like production accounts, blending seamlessly into the environment, but they are never used in business operations. That makes them irresistible to attackers scanning AD for escalation opportunities, but any interaction with them is suspicious by definition.

The accounts are tuned to mimic common AD weaknesses and detect specific attack techniques:

KERBEROASTING

In a Kerberoasting attack, adversaries request service tickets for service accounts. They then attempt to crack the ticket offline to recover plaintext credentials. An AD Tripwire configured for kerberoasting will trigger as soon as an attacker requests a ticket, producing a high-fidelity alert.

AS-REP ROASTING

Attackers exploit accounts that have “Do not require Kerberos pre-authentication” enabled. They request authentication responses and crack them offline. An AS-REP Roast AD Tripwire created without pre-authentication will trigger immediately when such a request occurs.

METADATA SCRAPING

Attackers often scrape AD attributes such as the “description” field in search of embedded credentials or hints. The Exposed Credential AD Tripwires are seeded with fake password strings. If an attacker attempts to use that password, the Tripwire triggers and exposes the intrusion.

By targeting techniques attackers cannot resist, AD Tripwires deliver signals that are both early and reliable.

HORIZON3.ai ~~TRUST BUT~~ VERIFY

Operational Model and Deployment

Deploying AD Tripwires is designed to be straightforward and secure. For each domain, up to three Tripwire accounts can be created, one for each of the described attack types. A single lightweight AD Agent runs on the selected NodeZero host. This agent has read-only permissions to the relevant subdirectory in SYSVOL and checks in at regular intervals, transmitting log data back to NodeZero for immediate alerting.

Several key design decisions ensure ease of use and security:

MINIMAL PRIVILEGES

The AD Agent is restricted to read-only access, with no ability to view legitimate credentials or monitor production user activities.

NO SENSITIVE DATA

Only Tripwire-related activity is monitored. No business communications or user data are collected.

LIGHTWEIGHT FOOTPRINT

The system requires few additional resources and relies only on existing Windows event logs.

SECURE STORAGE

Logs are stored in a specific subdirectory of SYSVOL, eliminating the need for additional systems and avoiding disruption to production services.

ERROR HANDLING

The agent monitors its own health and provides robust error messaging and documentation for easy troubleshooting.

EASE OF SETUP

Deployment requires a one-time PowerShell command, application of a domain policy template, and selection of usernames for the Tripwire accounts. Once installed, the Tripwires require no further maintenance.

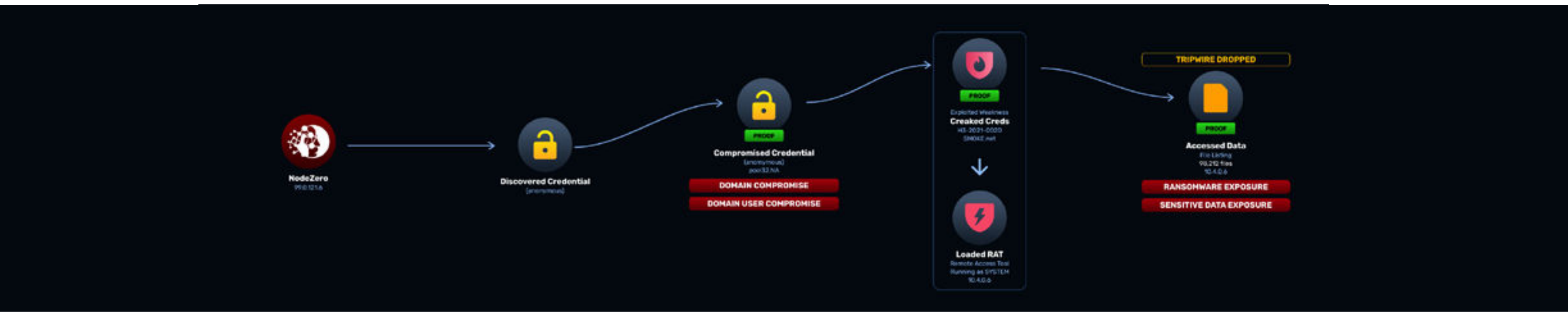
Importantly, AD Tripwires can also be validated through built-in end-to-end testing. Security teams can simulate an attack, see the Tripwire trigger, review the Windows logs generated, and practice their response. These tests not only confirm that the system works as expected but also train SOC analysts to recognize and react to alerts in production.

Strategic Fit

AD Tripwires embody the vision behind NodeZero Tripwires: use offense to inform defense. Traditional detection tools focus on probability. They generate alerts based on unusual patterns or correlation rules, which requires constant fine-tuning and still produces noise. AD Tripwires uses knowledge of attacker behavior against attackers, presenting as irresistible lures to generate real signal.

This distinction is powerful. Instead of sifting through false positives, SOC analysts receive a high-fidelity signal with full context: which identity was touched, how it was attacked, and what the adversary likely was attempting. That context enables faster, more confident response and significantly reduces attacker dwell time.

The messaging is clear: proof of attack, not probability.



Personas and Use Cases

Different organizations gain value from AD Tripwires in different ways:

Enterprises often have mature SOCs with SIEM, EDR, and ITDR investments, but they struggle with alert overload and lack visibility into AD. AD Tripwires cut through the noise and provide definitive proof of identity attacks, bridging the gap between red and blue teams.

Mid-market organizations may have only a few analysts, often relying heavily on Microsoft Defender or Sentinel. For them, AD Tripwires provide enterprise-grade detection capabilities without the cost and overhead of larger platforms.

SMBs often depend on outsourced detection and response. With AD Tripwires in place, even a small team gains protection against identity attacks with minimal maintenance.

MSPs and MSSPs benefit by delivering demonstrable protection to customers. AD Tripwires integrate seamlessly into SOC workflows, offering high-value alerts that reduce churn and strengthen upsell opportunities. Because they are lightweight and low-noise, they help protect margins while still improving service quality.

Across all personas, the value is the same: precise signals, early detection, and proof that defenses are working where it matters most.

Outcomes and Proof

NodeZero has already proven in benchmarks like GOAD that AD can be compromised in minutes. AD Tripwires flip that advantage. If NodeZero can exploit identity weaknesses, so can real attackers. By placing tripwires along those same attack paths, defenders gain immediate visibility into adversary behavior.

When an AD tripwire is triggered, the SOC does not just receive a binary alert. Each notification includes the targeted account, the attack, and the adversary's likely intent. This provides analysts with actionable intelligence in context, enabling faster triage and containment.

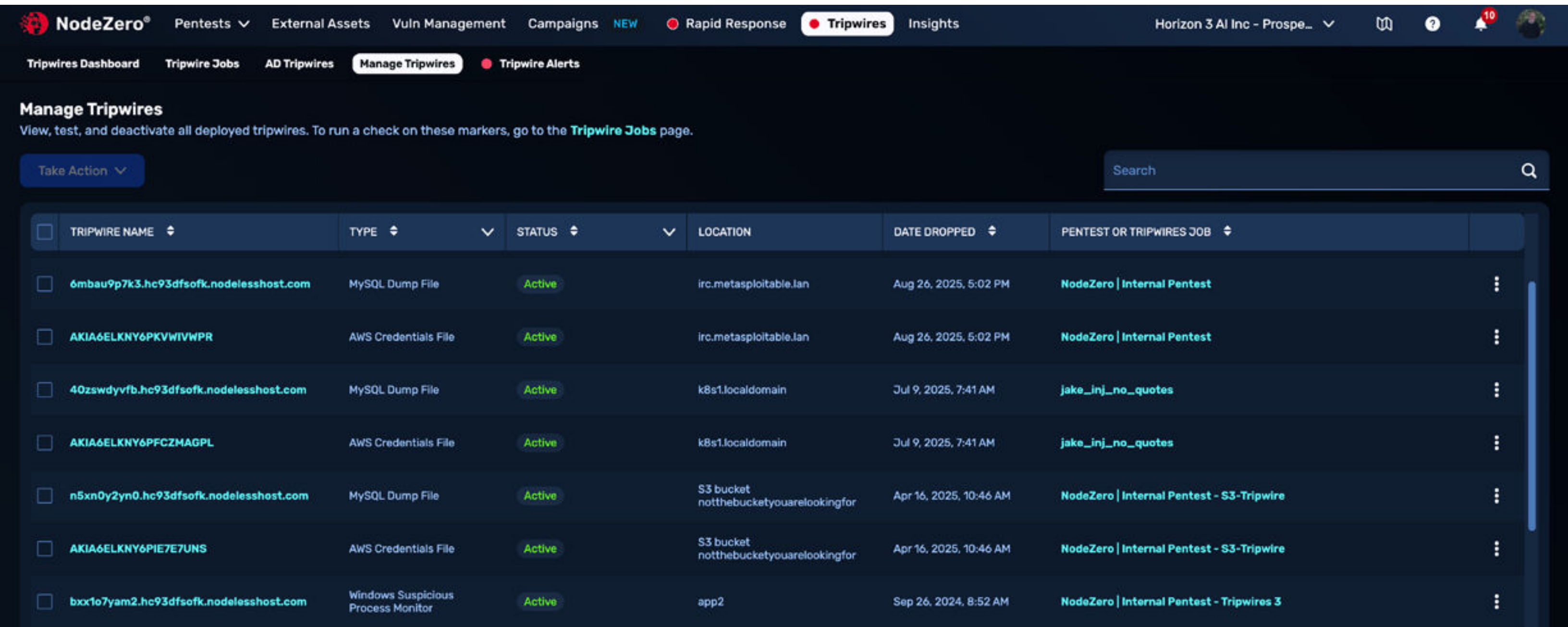
The outcome is a measurable reduction in dwell time. What once took weeks to detect can now be caught within minutes. This shift is not just technical. It validates security investment to boards and executives. It proves that defenses are not only deployed but working in production against the threats that matter most.

WHY AD TRIPWIRES MATTER

~90% of enterprises worldwide use AD.

~50% experienced AD attacks in the past 2 years, and >40% of those attacks were successful.

Attackers often escalate into AD in as little as 11 hours following initial access.



Manage Tripwires

View, test, and deactivate all deployed tripwires. To run a check on these markers, go to the Tripwire Jobs page.

TRIPWIRE NAME	TYPE	STATUS	LOCATION	DATE DROPPED	PENTEST OR TRIPWIRES JOB
6mbau9p7k3.hc93dfsofk.nodelesshost.com	MySQL Dump File	Active	irc.metasploitable.lan	Aug 26, 2025, 5:02 PM	NodeZero | Internal Pentest
AKIA6ELKNY6PKVWIVWPR	AWS Credentials File	Active	irc.metasploitable.lan	Aug 26, 2025, 5:02 PM	NodeZero | Internal Pentest
40zswdyvfb.hc93dfsofk.nodelesshost.com	MySQL Dump File	Active	k8s1.localdomain	Jul 9, 2025, 7:41 AM	jake_inj_no_quotes
AKIA6ELKNY6PFCZMAGPL	AWS Credentials File	Active	k8s1.localdomain	Jul 9, 2025, 7:41 AM	jake_inj_no_quotes
n5xn0y2yn0.hc93dfsofk.nodelesshost.com	MySQL Dump File	Active	S3 bucket nottthebucketyouarelookingfor	Apr 16, 2025, 10:46 AM	NodeZero | Internal Pentest - S3-Tripwire
AKIA6ELKNY6PIE7E7UNS	AWS Credentials File	Active	S3 bucket notthebucketyouarelookingfor	Apr 16, 2025, 10:46 AM	NodeZero | Internal Pentest - S3-Tripwire
bxx1o7yam2.hc93dfsofk.nodelesshost.com	Windows Suspicious Process Monitor	Active	app2	Sep 26, 2024, 8:52 AM	NodeZero | Internal Pentest - Tripwires 3

Conclusion

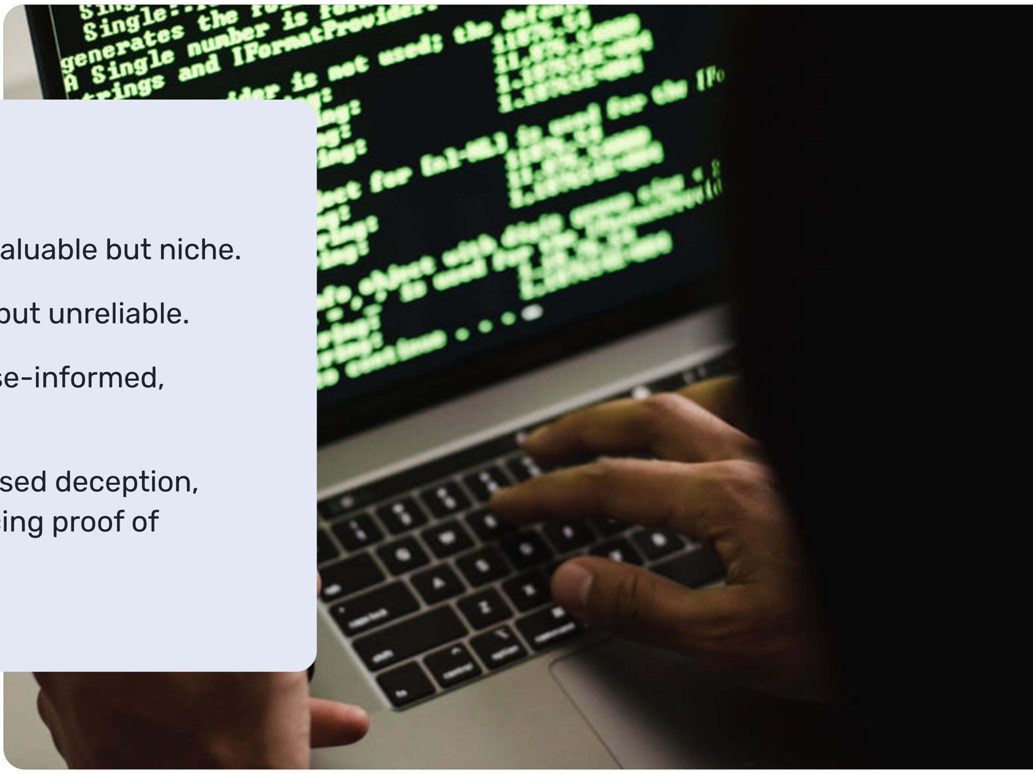
The history of deception in cybersecurity is a story of promise and frustration. Honeypots showed defenders they could observe attackers directly, but their cost and complexity kept them niche. Honey tokens promised lightweight traps, but uncertainty, noise, and integration gaps limited their effectiveness.

For years, deception was an intriguing concept without a practical path to widespread adoption.

NodeZero Tripwires and Active Directory Tripwires change that. By using offensive security insights to guide placement, they eliminate guesswork. By focusing on lures designed to catch true malicious behavior, they eliminate noise. By integrating with SOC workflows, they turn deception into actionable detection.

The result is a solution that delivers what honeypots and honey tokens could not: practical, scalable, high-fidelity detection where it matters most.

Deception is no longer a gimmick. It is a core part of modern defense.



Evolution of Deception

- **Honeypots:** heavy, costly, valuable but niche.
- **Honey tokens:** lightweight but unreliable.
- **NodeZero Tripwires:** offense-informed, context-rich, SOC-ready.
- **AD Tripwires:** identity-focused deception, cutting dwell time by surfacing proof of attacker behavior early.

Schedule a Demo