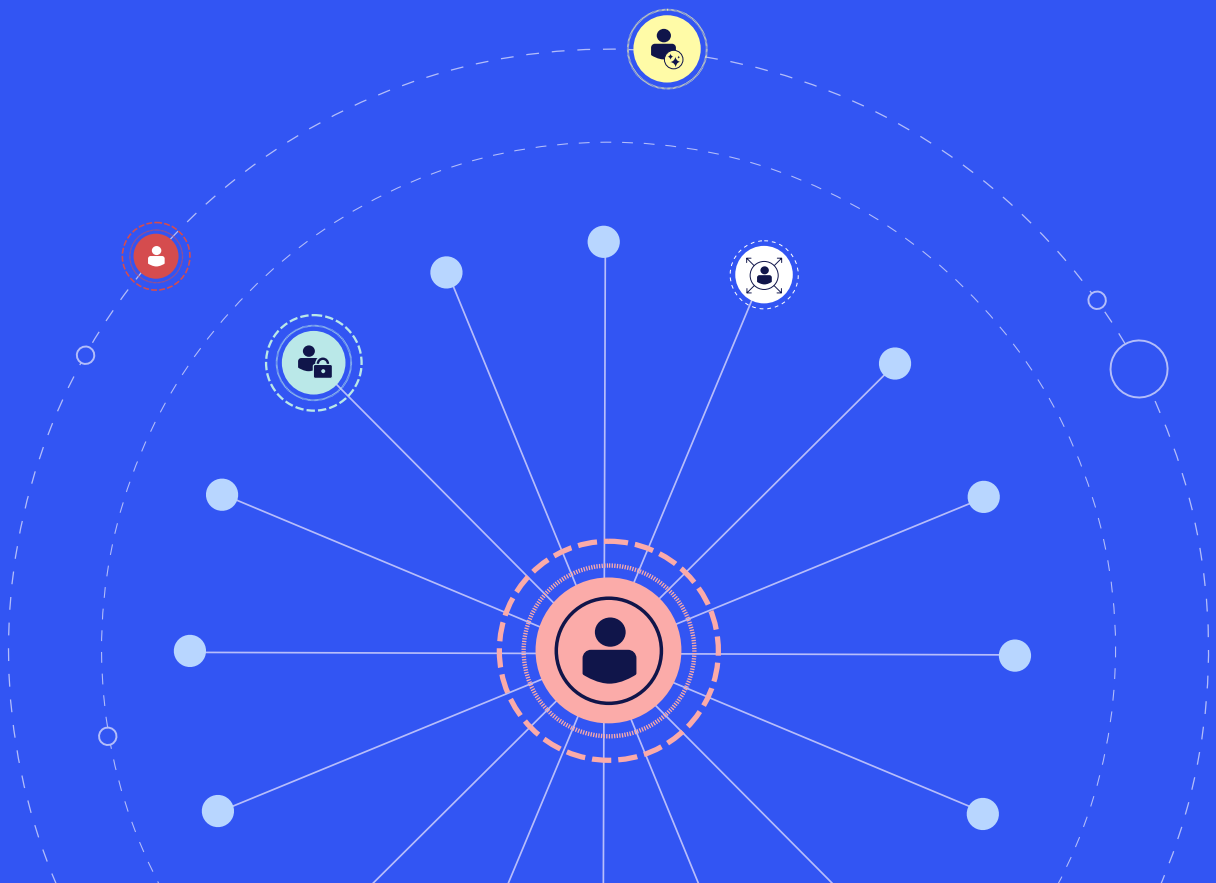
XM Cyber
A Brand of Schwarz Digits

How Identity Becomes the Attack Path:

11 Real-Life Stories

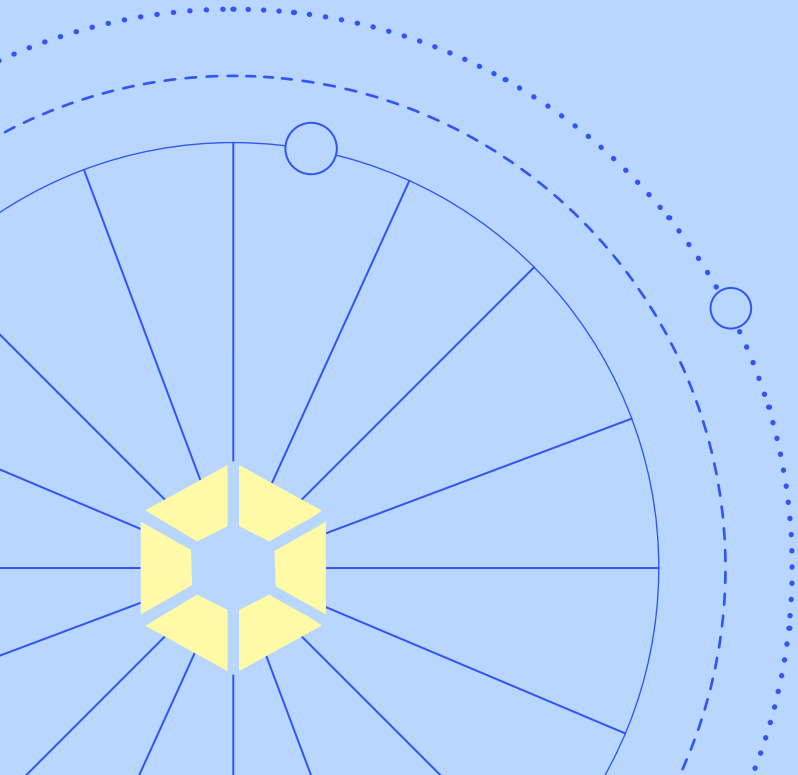


Identity Isn't a Perimeter. It's a Highway.

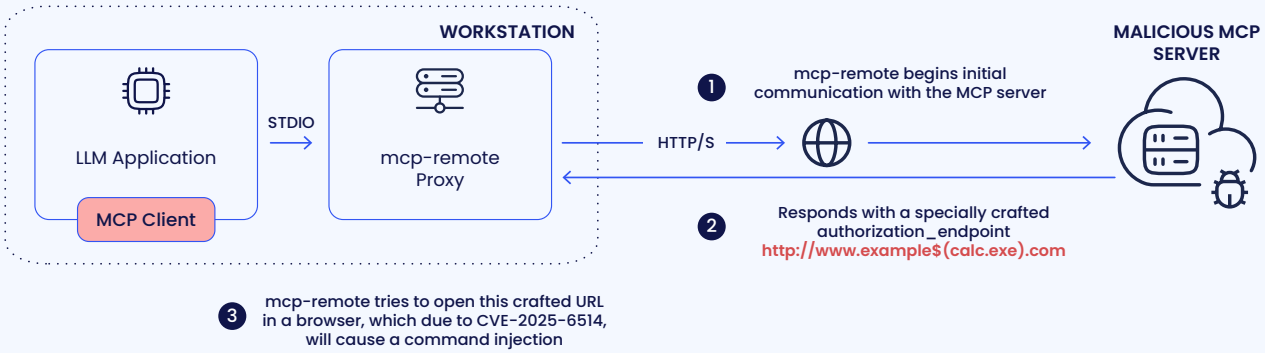
Identities touch every layer of a hybrid environment - Active Directory, cloud, service accounts, machine identities, AI agents. Attackers exploit that reach to break in, move laterally, and cross the boundary between on-prem and cloud to compromise critical assets. A stolen credential hands the attacker a legitimate identity - and every permission attached to it. Indeed, Verizon found that credential abuse was responsible for 22% of confirmed breaches and that stolen credentials were the root of 88% of attacks against web applications.

Most exposure management platforms scan for CVEs and flag misconfigurations without examining how permissions, credentials, and access controls connect those findings into exploitable paths. Identity-specific tools like IGA, PAM, and IAM are critical, but they typically manage specific identity security requirements in their own silos - without connecting to the broader attack surface.

This ebook walks through 11 real-life attack path scenarios that XM Cyber's in-house experts found in customer hybrid environments. Every story centers on identity as the key enabler - a credential, permission, or access control that let the attacker advance.



Attack Path #1: Mapping MCP Server Risk



Who was the customer?

A large enterprise whose development teams had adopted AI coding assistants and MCP-based tooling across their workflows.

What was the scenario?

The security team asked XM Cyber to assess the risk introduced by AI tools running in their environment - specifically, the MCP servers their developers had configured to connect AI agents to internal systems.

What was the attack path?

Several MCP servers across the environment had been configured with elevated privilege commands, giving the AI agents full administrative access to the host machines. A critical vulnerability in one of the MCP tools (CVE-2025-6514) allowed a malicious or compromised MCP server to execute arbitrary OS commands on any machine that connected to it. Because the MCP server process ran with admin-level permissions on the host, the attacker didn't just gain access - they inherited full control of the machine. From there, the attacker could steal cached AWS access keys stored on the machine, use them to compromise an IAM user in the cloud environment, and reach sensitive resources including S3 buckets and RDS databases.

What was the impact?

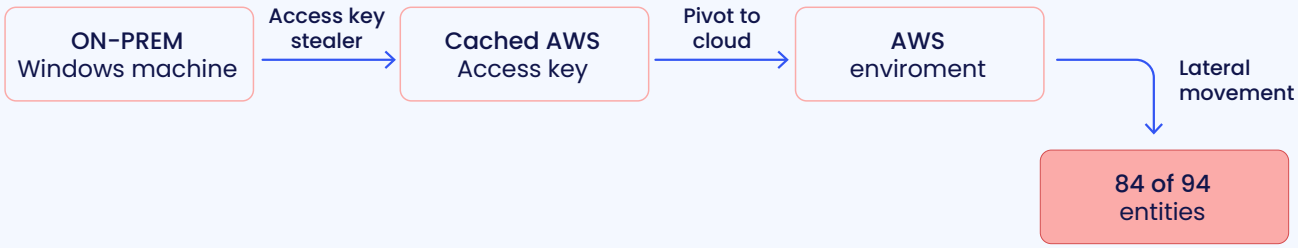
An outsider exploiting this path could move from a developer's workstation into the cloud environment and access sensitive data stored in S3 buckets and RDS databases - all through the permissions the AI agent already held. The AI agent's identity had, in fact, become the attack path.

How was it remediated?

The team used XM Cyber to scan all MCP configurations across the environment and identify which servers had been configured with elevated privileges. They removed the excessive permissions and locked down the configurations before the vulnerability could be exploited. The team also patched the affected MCP tool to close CVE-2025-6514. They deleted the cached AWS access keys from the developer workstations and tightened the user permissions to remove the path to sensitive cloud resources.

**Developers gave their AI tools admin access to move fast.
But an attacker only needed to connect to move faster.**

Attack Path #2: Unearthing the Access Key Cache



Who was the customer?

A major SaaS company running nearly all of its infrastructure on AWS.

What was the scenario?

The team wanted to test a specific question: if an attacker compromised an on-prem machine, could they reach the cloud environment - and how?

What was the attack path?

An AWS access key was cached on an on-prem Windows machine. This is standard AWS behavior - when a user uses the AWS CLI, the key gets cached automatically in the local directory. An attacker who gained access to that machine could steal the key using an Access Key Stealer technique, pivot into the AWS environment, and move laterally from there. One access key, belonging to an identity with excessive administrative permissions, opened a path to 84 out of 94 entities in the customer's AWS environment, including critical cloud workloads.

What was the impact?

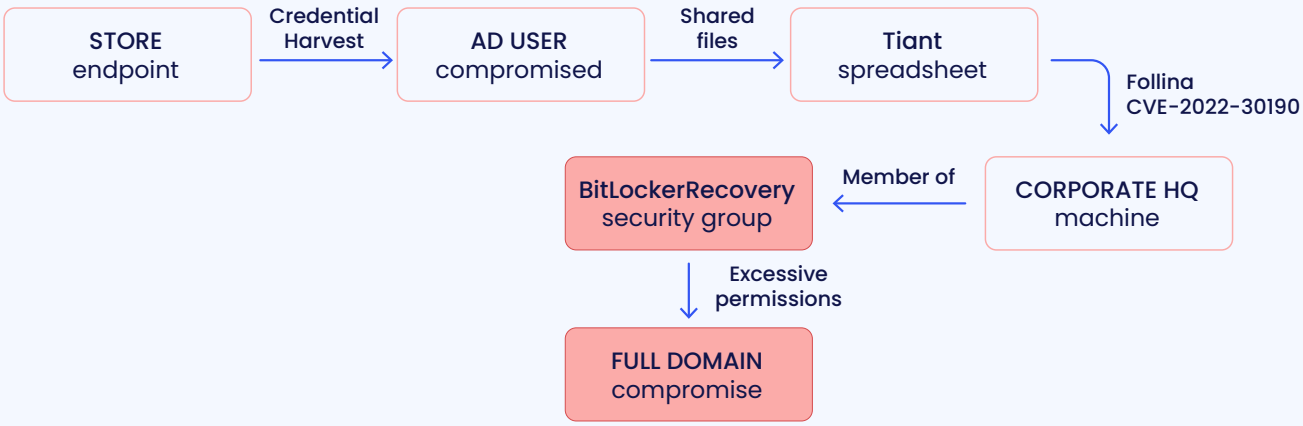
For a company whose critical infrastructure runs almost entirely in the cloud, this represented near-total compromise. An attacker with that single key could reach virtually every asset in their AWS environment.

How was it remediated?

In the short term, the team deleted the cached key - and the attack paths disappeared immediately. The longer-term fix required implementing a credential management solution to eliminate the caching mechanism entirely, so keys would not reappear the next time a user logged in

One cached access key.
84 out of 94 cloud assets compromised.

Attack Path #3: Reining in Excessive Permissions



Who was the customer?

A large convenience store and fuel retail chain with hundreds of locations across the United States.

What was the scenario?

The company asked XM Cyber to validate that its store and corporate networks were fully segregated.

What was the attack path?

The path started at a store endpoint. An attacker could harvest credentials from that device, compromise an Active Directory user, and reach shared files on the corporate network. From there, the attacker could inject malicious content into a shared spreadsheet. When a headquarters user opened the file, it triggered the Follina vulnerability (CVE-2022-30190), giving the attacker control of that machine. That machine's user held excessive permissions through membership in the BitLockerRecovery security group - a single group membership that opened a direct path to full domain compromise.

What was the impact?

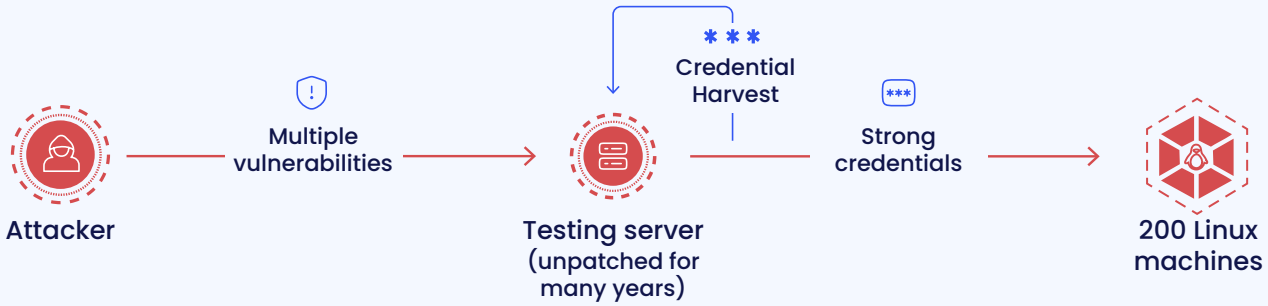
In 81% of completed attack simulations, the main domain was compromised.

How was it remediated?

The team removed the excessive permissions from the BitLockerRecovery group. The domain compromise rate dropped from 81% to 18% within one month.

Store and corporate were supposed to be segregated. One overlooked group membership put the entire domain at risk.

Attack Path #4: Shining a Light on Shadow IT



Who was the customer?

A large travel company that does getaway bookings.

What was the scenario?

The company had just merged with another company in the same industry and brought their infrastructure across.

What was the attack path?

The company had a product testing server that ran continuously and carried no particular importance. The team believed they were patching it, but patches had not applied correctly since 2017. The server carried a long list of vulnerabilities, including PrintNightmare and EternalBlue. An attacker who compromised the server could harvest cached credentials with elevated privileges - credentials that opened a direct path to 200 Linux machines across the environment.

What was the impact?

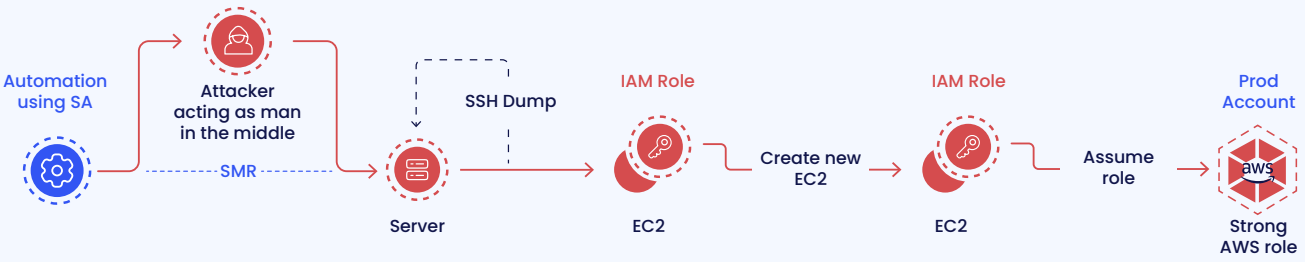
An unimportant, forgotten server held the keys to a significant portion of the production environment. When the team actually looked at the server, they realized they didn't even need it - so they just turned it off.

How was it remediated?

Turning off the server eliminated the attack path entirely and had an immediate impact on reducing risk across the environment.

A server no one cared about, holding credentials everyone should have cared about - for six years.

Attack Path #5: Seeing the Big Picture: On-Prem to Cloud Lateral Movement



Who was the customer?

A global financial institution.

What was the scenario?

The team wanted to understand whether an attacker with on-prem access could reach critical assets in their AWS environment.

What was the attack path?

XM Cyber identified a service account running an automation over the SMB port. An attacker could intercept that account's credentials through a Man-in-the-Middle attack and use them to move laterally to another workstation on the network. That workstation held SSH private keys to a server running on an EC2 instance in AWS. The EC2 had an attached IAM role with permissions to spin up new instances - and from there, the attacker could compromise one of the bank's most critical assets, a system used for production deployment.

What was the impact?

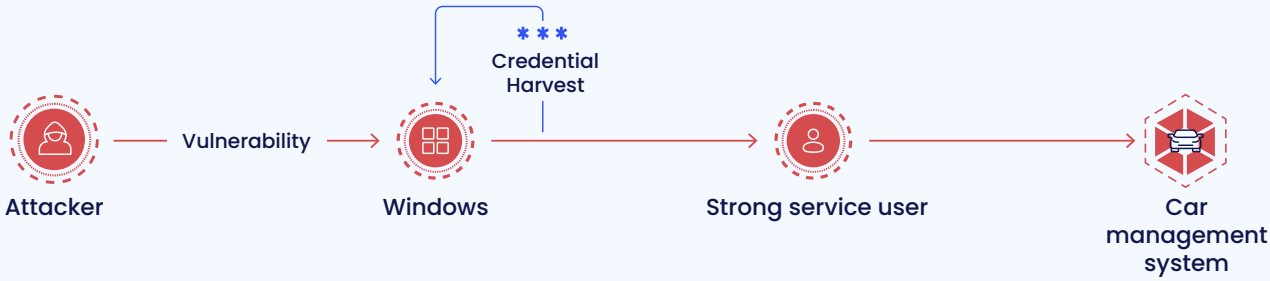
This path chains service account credentials, SSH keys, and IAM role permissions into a single attack path that crosses from on-prem into AWS. A few years ago, the complexity alone might have made it seem unlikely. That assumption no longer holds - AI-powered tools can map and traverse multi-step paths like this one in minutes.

How was it remediated?

The team immediately remediated each part of the attack path by removing private SSH keys, resetting IAM role permissions, and removing specific users.

Three identity exposures chained across two environments.
Complex enough to seem unlikely - until you consider what AI can find.

Attack Path #6: Identity is not a Silo: Combining Credentials and Vulnerabilities



Who was the customer?

A large manufacturer in EMEA.

What was the scenario?

Heavy goods moved through the factory on unmanned vehicles - a standard setup for industrial environments. The company had a vulnerability management solution in place and had done some hardening. XM Cyber ran a POC to assess whether those measures covered the systems controlling the vehicles.

What was the attack path?

A server in the factory controlled the uncrewed vehicles transporting goods. XM Cyber located an attack path that started with a software vulnerability on a Windows machine in the environment, led to credential harvest from a service account with elevated privileges, and ended with that compromised identity logging directly into the vehicle management system.

What was the impact?

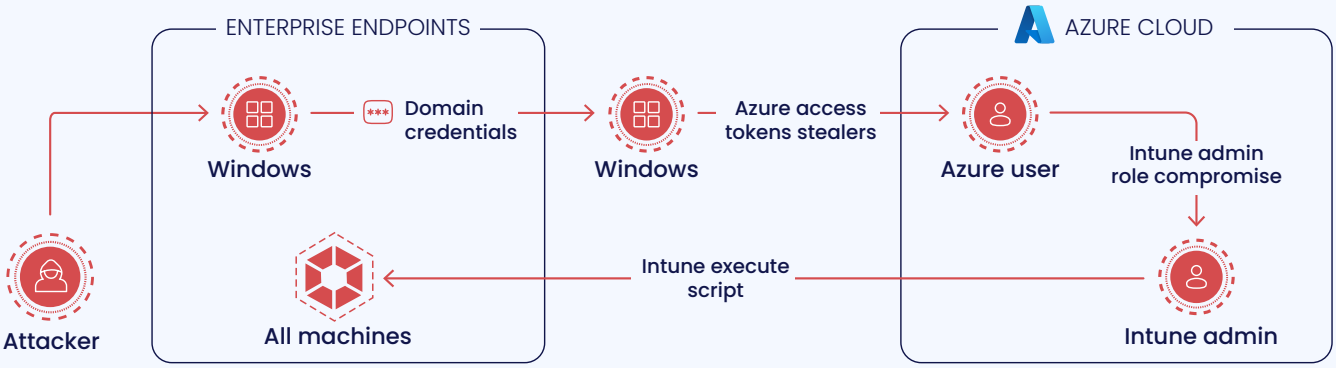
If an attacker got access, they'd be able to gain control of the vehicles and cause physical harm and damage – meaning IT Security could have a direct impact on the safety of personnel. The gap between a compromised identity and a physical safety incident was just two steps.

How was it remediated?

The team quickly realized that they needed a deeper understanding of their networks and created plans to onboard attack path modeling tools.

One overprivileged service account turned a software vulnerability into a physical safety risk.

Attack Path #7: From Credential Dump to Admin Privileges



Who was the customer?

A telecom enterprise with many subsidiaries.

What was the scenario?

The team wanted to assess whether their admin account structure created lateral movement risk across the enterprise.

What was the attack path?

Each admin used a single account for everything - domain admin, software installation, database administration. From any client machine that admin had logged into, XM Cyber was able to dump cached credentials, move laterally through the domain controller, and demonstrate how the entire domain could be compromised. The path extended into Azure: by harvesting valid access tokens from the compromised machine, an attacker could pivot to the cloud, compromise an Intune Administrator identity, and use that role to execute scripts back on every managed endpoint in the enterprise.

What was the impact?

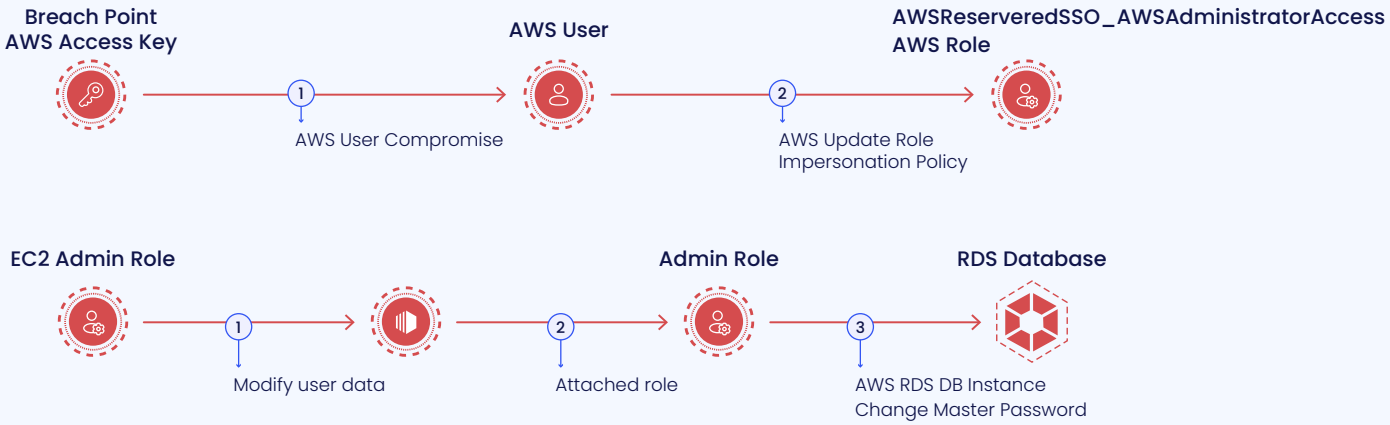
The team already had an initiative to address this - new accounts for admins and hardware tokens for authentication to prevent cached credentials - but the project had stalled. XM Cyber's findings gave them the evidence to take to management and get it prioritized.

How was it remediated?

The team removed risky credentials and re-established best practices for creating roles and permissions across both Active Directory and Azure.

One overprivileged admin identity bridged on-prem and cloud - turning a credential dump into full enterprise compromise

Attack Path #8: Escalating Privileges Across the Cloud



Who was the customer?

A large enterprise operating critical workloads in AWS.

What was the scenario?

The team wanted to evaluate whether their role structure contained privilege escalation paths.

What was the attack path?

A compromised AWS access key - tied to a single user - opened two separate routes to full account takeover. In the first, the user had a custom managed policy that let them rewrite who could assume an administrative role. An attacker could simply add themselves to the trust list, assume the role, and take over the entire AWS account. In the second, an EC2 admin role could manage all instances by design, and one of those instances had an administrative role attached. That identity chain gave the attacker access to everything in the account, including an RDS database containing sensitive data.

What was the impact?

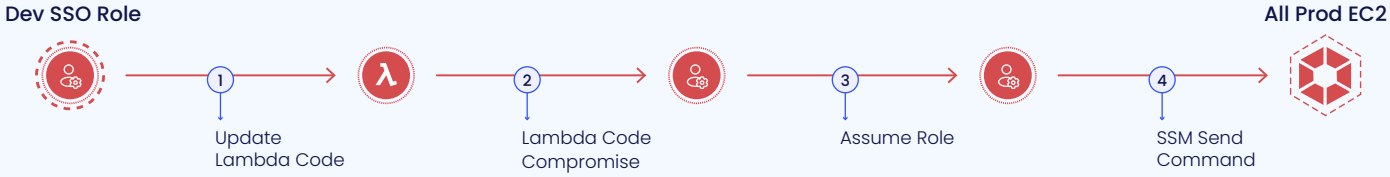
Every identity in the path - the user, the admin role, the EC2 role - carried permissions that the team had configured intentionally to support day-to-day workflows. From an attacker's perspective, those same identity relationships created two separate routes to full account takeover.

How was it remediated?

The team reviewed and tightened IAM policies across the environment, removing privilege escalation paths from roles and users that did not require them.

Two identity chains. Two routes to admin. Both built from permissions no one questioned.

Attack Path #9: The Forgotten Permissions that Led to Compromise



Who was the customer?

A Fortune 500 bank operating production workloads in AWS.

What was the scenario?

The bank wanted to understand whether low-level roles could reach production systems.

What was the attack path?

A developer SSO role - originally provisioned for a cloud migration project - retained permissions to update Lambda code long after the migration ended. An attacker who compromised that developer identity could inject malicious code into a Lambda function, compromise the function's execution role, and assume a system role with broad privileges across the AWS console. From there, a single SSM Send Command would give the attacker control over all production EC2 instances in the account.

What was the impact?

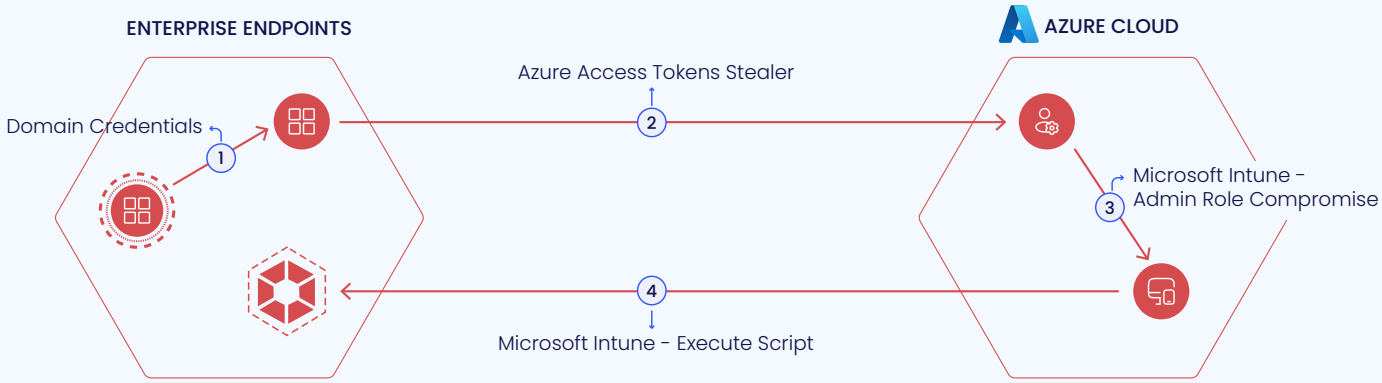
A low-level developer identity could escalate to full production access in four steps. The role's elevated permissions were a leftover from the migration - no system existed to detect that a high-privilege identity was still active after its original purpose had ended.

How was it remediated?

The team removed the excessive permissions from the developer's policy and implemented role lifecycle controls to prevent lingering high-privilege identities from going undetected.

A migration role that outlived its purpose. Four steps from developer to production admin - and no one knew the path existed.

Attack Path #10: The On-Prem Breach That Reached the Cloud



Who was the customer?

A large enterprise managing endpoints through Microsoft Intune.

What was the scenario?

The team wanted to understand whether a compromised on-prem endpoint could lead to exposure in their Azure environment.

What was the attack path?

The path started on-prem with a set of compromised domain credentials. From that foothold, the attacker could steal valid Azure access tokens from the same machine and pivot into the cloud environment. Once in Azure, the attacker could escalate privileges and compromise an Intune Administrator identity. That single identity carried permissions to execute scripts remotely on every device Intune managed - giving the attacker a path back into the on-prem environment and directly into Active Directory.

What was the impact?

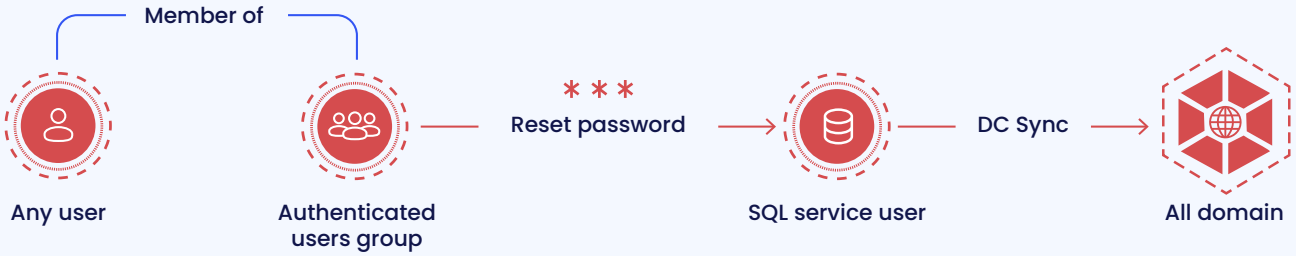
One compromised on-prem identity opened a path to Azure. One overprivileged cloud identity sent the attacker right back to on-prem - this time with administrative control over every managed endpoint. The round trip crossed the on-prem/cloud boundary twice, and the sensitive Active Directory sat at the end of the chain.

How was it remediated?

The team discovered more users than expected held the Intune Administrator role. They removed all unnecessary role assignments and tightened identity governance across both environments.

On-prem credentials got the attacker to Azure. An Intune admin identity brought them back - with the keys to Active Directory.

Attack Path #11: Every Employee Was an (Accidental) Admin



Who was the customer?

A hospital operating in a high-compliance healthcare environment.

What was the scenario?

The hospital experienced a breach that started with a phishing attack on a non-privileged user. The team brought in XM Cyber to understand how the attacker escalated from that foothold to full domain control.

What was the attack path?

A misconfiguration in Active Directory granted every user in the domain ForceChangePassword rights - the ability to reset any other user's password. The attacker compromised a non-privileged user and reset the password of a Tier-1 IT helpdesk account and then a Domain Admin account. The exposure carried no CVE and lived entirely in AD permissions - invisible to the vulnerability scanners already in the environment.

What was the impact?

A low-level compromise became full domain control within minutes. The attacker gained administrative control over patient records and hospital systems - all through a single legacy permission that no scanning tool had flagged.

How was it remediated?

XM Cyber identified the ForceChangePassword misconfiguration as the choke point. The team hardened AD inheritance to remove the excessive rights across the domain.

One legacy permission gave every employee in the domain the power to reset any password - including the Domain Admin's.

The Takeaway

Identity is what let the attacker advance in every scenario in this ebook. Cached credentials, overprivileged service accounts, forgotten role assignments, and AI agents with admin access all served as bridges between the initial foothold and a critical asset.

These exposures don't show up in a CVE scan. They live in Active Directory configurations, permissions policies for identities, group memberships, and cached access tokens scattered across hybrid environments. Tools that treat identity as a silo will miss how these exposures connect into full attack paths.

XM Cyber maps those connections. The platform traces how threat actors chain identity and access exposures with vulnerabilities, misconfigurations, and weak segmentation to reach critical assets across on-prem and cloud environments. It validates the choke points where a single fix breaks multiple attack paths. It prioritizes those fixes and provides guided remediation steps with the context teams need to deploy them.