Microsoft Security

CYBERATTACK SERIES

Beyond source code: A path to the keys to the kingdom

How a password reset became a pathway to the cloud

In our latest Cyberattack Series report, we examine how Microsoft Defender Experts Cybersecurity Incident Response, delivered by the Detection and Response Team (DART), investigated activity by STORM-3068—a threat actor that turned a compromised identity into a gateway to Azure DevOps, development pipelines, and Kubernetes environments by abusing legitimate cloud and identity services rather than relying on malware or exploits.

The incident began with a successful self-service password reset and quickly expanded into the organization's development environment. Using valid credentials, the threat actor registered new authentication methods, established persistent access, and leveraged Azure DevOps administrative capabilities to enumerate repositories, access project data, create malicious pipelines, and obtain Kubernetes credentials. With those credentials, the actor gained a pathway beyond Azure DevOps and into connected cloud infrastructure.

This case highlights a growing challenge for defenders: development platforms, cloud management tools, and the identities that connect them have become high-value targets. When identity, source code, pipelines, and production environments are tightly linked, a single compromised account can provide a path to much broader access across an organization.

What happened?

The attack flow

How did Microsoft respond?

What can customers do to strengthen their defenses?

What happened?

The attack began when STORM-3068 gained access to a user account through a self-service password reset process and then took control of the account by registering its own authentication methods. With access established, the threat actor turned its attention to Azure DevOps, gathering information about repositories, projects, pipelines, and deployment environments using both automated scripts and legitimate administrative tools.

Azure DevOps was a particularly valuable target because it sat at the intersection of identity, development, and cloud operations. Beyond source code, it provided visibility into how the organization built, deployed, and managed its environment. By gaining access, the threat actor was able to identify trusted deployment paths, discover connected cloud resources, and leverage existing administrative relationships to further expand access. In many organizations, development platforms function as a central hub for business-critical systems, making them a high-value target for attackers seeking broader control of the environment.

Using the permissions, the actor deployed the pipeline that was authorized to access over 50 resources. The main purpose of the pipeline is accessing and retrieving Kubernetes 'kubeconfig' files which contain information pertaining to Kubernetes clusters, users, namespaces, and authentication mechanisms. Using Azure DevOps audit data, pipeline-related actions could be tracked in detail.

Then, with the audit data alongside the default Git version tracking history, a clear picture of subsequent threat actor actions could be rebuilt.

INSIGHT

Azure DevOps can reveal much more than source code

Repositories, pipelines, service connections, and deployment settings can provide attackers with a roadmap to an organization's broader environment.

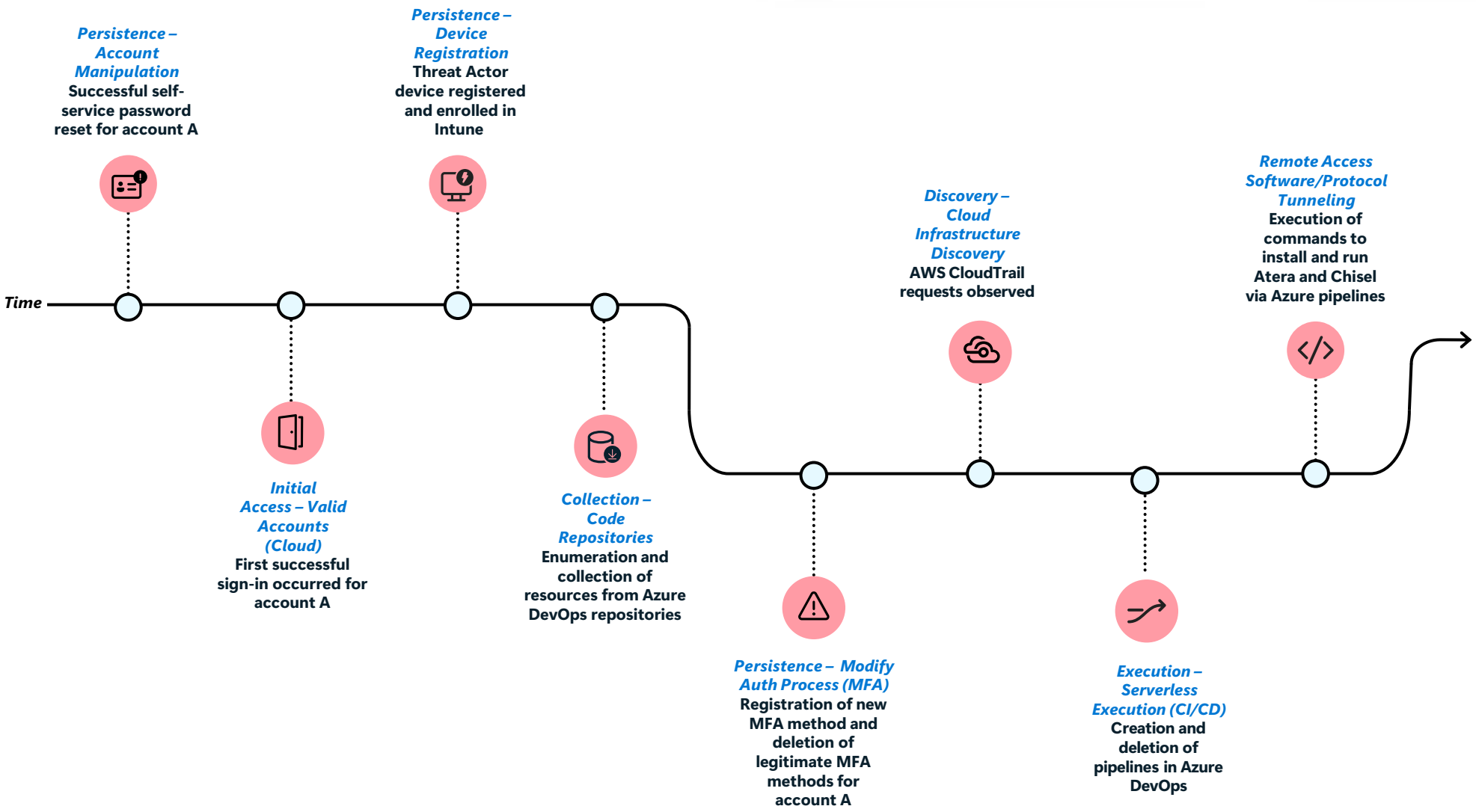
The threat actor initially deployed a kube agent by downloading and executing a third-party script, introducing multiple jobs using the DUMP_CLUSTER_<NAME> pattern, obtaining the contents of the kubeconfig files, and saving them locally.

Subsequently, the threat actor added kubeconfig files to the targeted repository in an already existing 'kubeconfigs' folder. Each file contained a dedicated Kubernetes API server endpoint, certificate authority data, and cluster identification details—as well as authentication token associated with a dedicated service account. The files provided a pathway to interact directly with Kubernetes clusters outside of the Azure DevOps environment and represented a significant escalation in access.

The threat actor then modified the Azure DevOps pipeline to deploy additional tooling, including the Atera remote management agent and the Chisel tunneling utility. Atera provided a trusted remote administration capability, while Chisel was used to proxy Kubernetes API traffic and establish reverse tunnels to external infrastructure. These additions provided the threat actor with methods in an attempt to facilitate continued access and remote interaction with compromised systems.

The attack flow

A single compromised identity was used to abuse Azure DevOps, steal Kubernetes credentials, and pivot into connected cloud environments.



How did Microsoft respond?

DART immediately activated its containment and hardening playbook to investigate the intrusion and disrupt the threat actor's access. By analyzing telemetry across Microsoft's security platforms and specialized investigative tools, the team traced the attacker's activity across identity systems, Azure DevOps, and cloud infrastructure, helping build a clear picture of how the compromise unfolded.

As the investigation progressed, DART worked closely with the customer through daily briefings, sharing findings, providing visibility into observed activity, and delivering prioritized containment and remediation recommendations. This continuous coordination enabled the customer to make informed decisions and execute response actions quickly as new details emerged.

DART also collaborated with Microsoft Threat Intelligence to correlate investigative findings and place the activity into a broader threat context. This analysis identified multiple related streams of attacker activity, helping responders better understand the scope of the intrusion and refine response actions across affected environments.

Beyond containing the immediate threat, DART provided guidance to strengthen the organization's overall security posture. These recommendations focused on improving resilience, reducing opportunities for future compromise, and aligning security controls with Microsoft security best practices to help defend against similar attacks in the future.

TACTIC

Trusted pipelines were exploited

Rather than deploying malware, the threat actor modified development pipelines to collect Kubernetes credentials and expand access into cloud infrastructure.

INSIGHT

Identities are the new attack path

This incident demonstrates how a single compromised identity can provide access to development platforms, cloud resources, and production environments when those systems are tightly connected.

What can customers do to strengthen their defenses?

This incident serves as a reminder that when attacks move at speed, consistent execution of foundational security practices remains one of the most effective ways to limit attacker access and contain compromise.

Monitor for suspicious password reset activity

Implement monitoring for anomalous Self-Service Password Reset (SSPR) activity, including high volumes of reset attempts from a single IP address or attempts targeting multiple users within a short period.

Strengthen authentication for privileged accounts

Review whether administrative and privileged accounts can use SSPR and ensure they are protected with phishing-resistant multifactor authentication if enabled.

Require approval for code changes

Enforce branch protection policies and require pull request approvals to prevent unauthorized or unreviewed changes from being introduced into repositories.

Restrict direct commits to protected branches

Limit or block direct commits to critical branches so that all changes follow established review and approval workflows.

Control pipeline permissions

Limit who can create, modify, and execute build and deployment pipelines, reducing the risk that attackers can abuse pipelines to access sensitive resources or credentials.

Enforce least-privilege access

Regularly review and restrict user permissions to ensure accounts have only the access required for their role, minimizing the impact of a compromise.

Conclusion

This incident demonstrates how quickly a compromised identity can become a pathway into development and cloud environments when identities, repositories, pipelines, and infrastructure are closely connected. By abusing legitimate authentication workflows and administrative capabilities, the threat actor expanded access in a matter of hours rather than days.

The good news is that many of the techniques used in this attack can be mitigated through foundational security practices, including stronger authentication, monitoring for suspicious password reset activity, tighter pipeline controls, and least-privilege access. And when incidents do occur, Microsoft Defender Experts Cybersecurity Incident Response can help organizations investigate, contain, and recover while strengthening their defenses against future attacks.

→ Download incident response playbook resources from the official Microsoft Download Center.

→ Learn more about how DART can help you before, during, and after a security incident.

©2026 Microsoft Corporation. All rights reserved. The information in this document represents the current view of Microsoft on the content. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.