Threat Hunt Report: New Local Administrator Account Creation from Unauthorized IP

Detection of new administrator account being created on `peter-vm` from net.exe or net1.exe, added by a non-whitelisted IP.

Scenario:

In our ongoing efforts to secure endpoints and maintain strict access control per Management's guidance, we are focusing on detecting unauthorized local admin account creation across our systems. This threat hunt specifically targets events where local administrator accounts are created on `peter-vm` originating from IP addresses outside of the approved whitelist. Such actions may indicate an external threat actor attempting to establish persistent access to the system.

The goal of this hunt is to monitor and detect the following:

- Creation of new **local administrator accounts** (`net user /add` and `Add-LocalGroupMember -Group "Administrators" -Member` commands for cmd.exe and powershell.exe, respectively).

- Detection of these events when originating from **non-whitelisted IPs (for a list we will provide, just one IP currently)**.

This is an important indicator of compromise (IoC) for us as defenders. This activity could indicate that an attacker has gained access to our environment, and is attempting to establish persistence, evade detection, or further escalate privileges. Attackers often leverage administrative access for persistence in compromised environments. These tactics are consistent with MITRE ATT&CK tactics & techniques:

Tactic: Persistence (TA0003)

- **Technique: Create Account (T1136)** (specifically **Local Account** as a sub-technique)

Tactic: Privilege Escalation (TA0004)

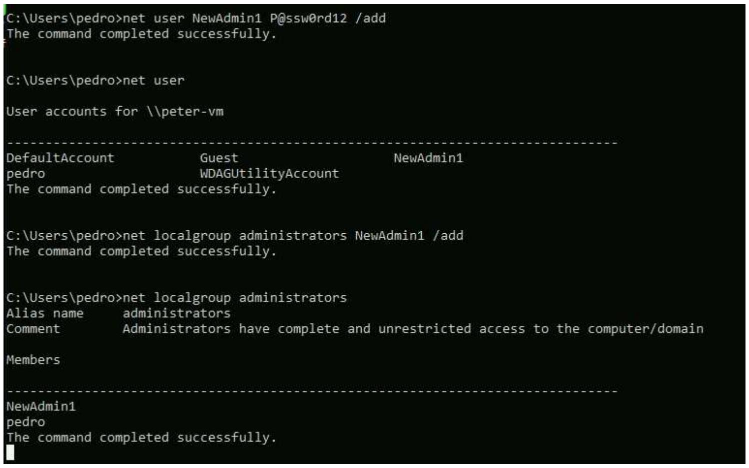
- **Technique: Abuse Elevation Control Mechanism (T1548)** (Create or Modify System Process)

High-Level IoC Discovery Plan

- Define a list of 'whitelisted' IPs to serve as safe, known IPs.
- Check for use of net.exe or net1.exe on endpoint `peter-vm`.
- Check **`DeviceProcessEvents`** for any signs of command strings which add new users via cmd or Powershell.
- Determine if the new user was added from an IP on the whitelist, or not.

Steps Taken

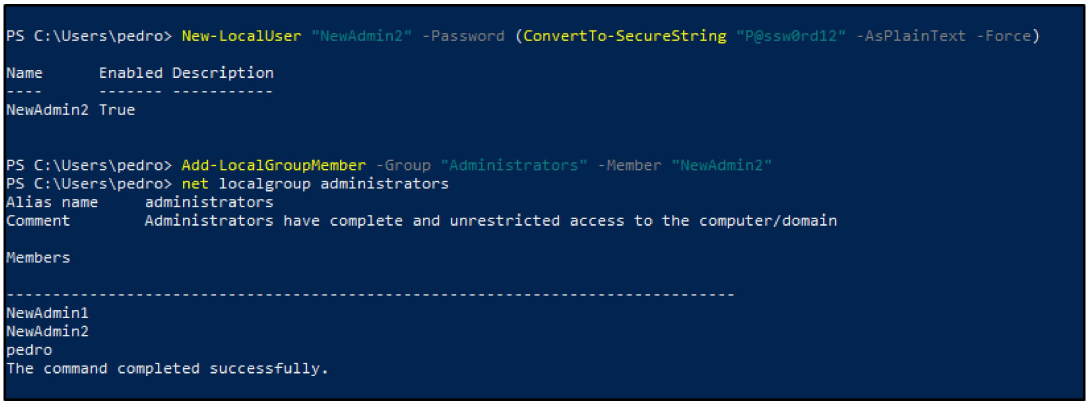
1. Started the VM and created accounts using cmd.exe and powershell.exe, then promoted both accounts to administrator using the same shells (2 new admins).



```
C:\Users\pedro>net user NewAdmin1 P@ssw0rd12 /add
The command completed successfully.


C:\Users\pedro>net user

User accounts for \\peter-vm

-------------------------------------------------------------------------------
DefaultAccount           Guest                    NewAdmin1
pedro                    WDAGUtilityAccount
The command completed successfully.


C:\Users\pedro>net localgroup administrators NewAdmin1 /add
The command completed successfully.


C:\Users\pedro>net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
NewAdmin1
pedro
The command completed successfully.
```

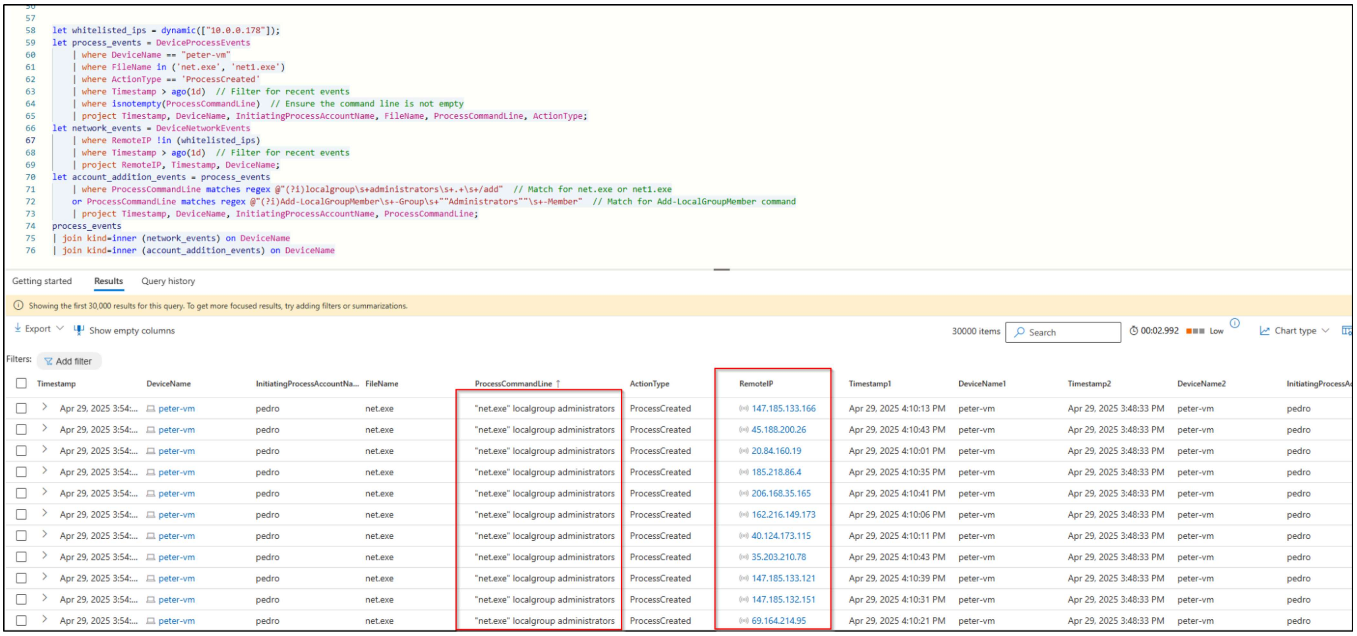
Above: New user created with cmd.exe, then added to administrators group, administrators group checked.

Below: New user created with powershell.exe, added to administrators group, then administrators group membership checked.



```
PS C:\Users\pedro> New-LocalUser "NewAdmin2" -Password (ConvertTo-SecureString "P@ssw0rd12" -AsPlainText -Force)

Name      Enabled Description
----      ------- -----------
NewAdmin2 True


PS C:\Users\pedro> Add-LocalGroupMember -Group "Administrators" -Member "NewAdmin2"
PS C:\Users\pedro> net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
NewAdmin1
NewAdmin2
pedro
The command completed successfully.
```

2. Started forming a KQL query chain by checking for DeviceProcessEvents, specifically for the VM `peter-vm`. Adjusted the query to pull in Events with FileName 'net.exe' or net1.exe', which are system administration binaries for Windows that could add/modify/delete users/groups/admins and more.
3. Created a variable for the whitelisted IPs, this is declared at the top of the query. This will contain our whitelisted IP, and any IPs we deem worthy of being on our admin whitelist (part of the organization or 'safe'). The next condition in the KQL query checks to see if the RemoteIP is on the 'whitelist' or not.
4. Now that there were multiple conditions, the query needed restructured into a more helpful format, where each overall condition for detection was separated. Basically we want to check for whether the Events matched all three conditions, and only then include them in our results for the query.
5. The third condition became a regex match. So far we checked if the Event used files net or net1, was it from a non-whitelisted IP, and finally we want to know if the ProcessCommandLine matches a command which would add a new admin to the local admins group.
6. The condition blocks were merged/joined.
7. The query was tested and tweaked. The VM was remade, many revisions of the rule followed. There continued to be many IPs showing up in the RemoteIP field, which seemed unusual, but is likely part of Azure's backbone or Microsoft infrastructure:



```
let whitelisted_ips = dynamic(["10.0.0.178"]);
let process_events = DeviceProcessEvents
    | where DeviceName == "peter-vm"
    | where FileName in ('net.exe', 'net1.exe')
    | where ActionType == 'ProcessCreated'
    | where Timestamp > ago(1d)  // Filter for recent events
    | where isnotempty(ProcessCommandLine)  // Ensure the command line is not empty
    | project Timestamp, DeviceName, InitiatingProcessAccountName, FileName, ProcessCommandLine, ActionType;
let network_events = DeviceNetworkEvents
    | where RemoteIP !in (whitelisted_ips)
    | where Timestamp > ago(1d)  // Filter for recent events
    | project RemoteIP, Timestamp, DeviceName;
let account_addition_events = process_events
    | where ProcessCommandLine matches regex @"(?i)localgroup\s+administrators\s+.+\s+/add"  // Match for net.exe or net1.exe
   or ProcessCommandLine matches regex @"(?i)Add-LocalGroupMember\s+-Group\s+""Administrators""\s+-Member"  // Match for Add-LocalGroupMember command
    | project Timestamp, DeviceName, InitiatingProcessAccountName, ProcessCommandLine;
process_events
| join kind=inner (network_events) on DeviceName
| join kind=inner (account_addition_events) on DeviceName
```

ⓘ Showing the first 30,000 results for this query. To get more focused results, try adding filters or summarizations.

□	Timestamp	DeviceName	InitiatingProcessAccountNa...	FileName	ProcessCommandLine ↑	ActionType	RemoteIP	Timestamp1	DeviceName1	Timestamp2	DeviceName2	InitiatingProcessA
□	Apr 29, 2025 3:54:...	peter-vm	pedro	net.exe	"net.exe" localgroup administrators	ProcessCreated	147.185.133.166	Apr 29, 2025 4:10:13 PM	peter-vm	Apr 29, 2025 3:48:33 PM	peter-vm	pedro
□	Apr 29, 2025 3:54:...	peter-vm	pedro	net.exe	"net.exe" localgroup administrators	ProcessCreated	45.188.200.26	Apr 29, 2025 4:10:43 PM	peter-vm	Apr 29, 2025 3:48:33 PM	peter-vm	pedro
□	Apr 29, 2025 3:54:...	peter-vm	pedro	net.exe	"net.exe" localgroup administrators	ProcessCreated	20.84.160.19	Apr 29, 2025 4:10:01 PM	peter-vm	Apr 29, 2025 3:48:33 PM	peter-vm	pedro
□	Apr 29, 2025 3:54:...	peter-vm	pedro	net.exe	"net.exe" localgroup administrators	ProcessCreated	185.218.86.4	Apr 29, 2025 4:10:35 PM	peter-vm	Apr 29, 2025 3:48:33 PM	peter-vm	pedro
□	Apr 29, 2025 3:54:...	peter-vm	pedro	net.exe	"net.exe" localgroup administrators	ProcessCreated	206.168.35.165	Apr 29, 2025 4:10:41 PM	peter-vm	Apr 29, 2025 3:48:33 PM	peter-vm	pedro
□	Apr 29, 2025 3:54:...	peter-vm	pedro	net.exe	"net.exe" localgroup administrators	ProcessCreated	162.216.149.173	Apr 29, 2025 4:10:06 PM	peter-vm	Apr 29, 2025 3:48:33 PM	peter-vm	pedro
□	Apr 29, 2025 3:54:...	peter-vm	pedro	net.exe	"net.exe" localgroup administrators	ProcessCreated	40.124.173.115	Apr 29, 2025 4:10:11 PM	peter-vm	Apr 29, 2025 3:48:33 PM	peter-vm	pedro
□	Apr 29, 2025 3:54:...	peter-vm	pedro	net.exe	"net.exe" localgroup administrators	ProcessCreated	35.203.210.78	Apr 29, 2025 4:10:43 PM	peter-vm	Apr 29, 2025 3:48:33 PM	peter-vm	pedro
□	Apr 29, 2025 3:54:...	peter-vm	pedro	net.exe	"net.exe" localgroup administrators	ProcessCreated	147.185.133.121	Apr 29, 2025 4:10:39 PM	peter-vm	Apr 29, 2025 3:48:33 PM	peter-vm	pedro
□	Apr 29, 2025 3:54:...	peter-vm	pedro	net.exe	"net.exe" localgroup administrators	ProcessCreated	147.185.132.151	Apr 29, 2025 4:10:31 PM	peter-vm	Apr 29, 2025 3:48:33 PM	peter-vm	pedro
□	Apr 29, 2025 3:54:...	peter-vm	pedro	net.exe	"net.exe" localgroup administrators	ProcessCreated	69.164.214.95	Apr 29, 2025 4:10:21 PM	peter-vm	Apr 29, 2025 3:48:33 PM	peter-vm	pedro

The VM had been created about 15 minutes prior to the above screenshot, and a strong password and different username were chosen, so there does not seem to be any issue with outside activity.

Final KQL Query:

```
let whitelisted_ips = dynamic(["10.0.0.178"]);
let process_events = DeviceProcessEvents
    | where DeviceName == "peter-vm"
    | where FileName in ('net.exe', 'net1.exe')
    | where ActionType == 'ProcessCreated'
    | where Timestamp > ago(1d)  // Filter for recent events
    | where isnotempty(ProcessCommandLine)  // Ensure the
command line is not empty
    | project Timestamp, DeviceName,
InitiatingProcessAccountName, FileName, ProcessCommandLine,
ActionType;
let network_events = DeviceNetworkEvents
    | where RemoteIP !in (whitelisted_ips)
    | where Timestamp > ago(1d)  // Filter for recent events
    | project RemoteIP, Timestamp, DeviceName;
let account_addition_events = process_events
```

```
    | where ProcessCommandLine matches regex
@"(?i)localgroup\s+administrators\s+.+\s+/add"  // Match for
net.exe or net1.exe
    or ProcessCommandLine matches regex
@"(?i)Add-LocalGroupMember\s+-Group\s+""Administrators""\s+-Mem
ber"  // Match for Add-LocalGroupMember command
    | project Timestamp, DeviceName,
InitiatingProcessAccountName, ProcessCommandLine;
process_events
| join kind=inner (network_events) on DeviceName
| join kind=inner (account_addition_events) on DeviceName
```

Chronological Events

1. The threat simulation for this hunt included the assumption that the threat actor already has access to the `peter-vm` device via CLI or GUI. The threat actor (TA) would have an IP not on our company whitelist, which is one of the criteria for the alert to fire.

2. The threat actor uses the net.exe / net1.exe binary, or the Add-LocalGroupMember cmdlet to add a user to the administrators group on Windows, which is another one of the alert criteria.

- This could be a freshly-created user, but should work with any user added to the administrators group, even if the promoted user wasn't created recently.

3. Command line strings match for commands identified by regex in our rule, which is the third criteria. Those regex rules are:

`@"(?i)localgroup\s+administrators\s+.+\s+/add"`   `//` Match for net.exe or net1.exe
`@"(?i)Add-LocalGroupMember\s+-Group\s+""Administrators""\s+-Member"` `//` Match for Add-LocalGroupMember command

4. When ALL 3 criteria above are positive, the rule fires. This will show our security team when an unknown IP is able to use account administration tools with specific command line syntax, and change group membership for users into the administrators group.

Response Taken

1. **Isolation of the Device**: Upon detecting the unauthorized IP's activity, the affected machine was **isolated** to prevent any further unauthorized access.

2. **Password Reset**: The password for the **new admin account** was immediately reset, and the account was deleted from the system.

3. **Investigative Package**: The device was subjected to a full **incident response investigation** to assess the extent of any potential breach or lateral movement within the network.

4. **Alert Tuning**: The alert was refined to include additional actions based on further findings during the investigation, ensuring that any future suspicious behavior is detected immediately.

Conclusion

The threat hunt for unauthorized local administrator account creation on the **peter-vm** device from a non-whitelisted IP was successfully conducted, which used a detection strategy based on critical system administration tools and commands.

The goal of this hunt was to identify potential compromise or lateral movement by external threat actors attempting to establish persistence on the system. The process was executed through detecting events triggered by the use of **net.exe** or **net1.exe**, tools commonly leveraged to modify user accounts, as well as **Add-LocalGroupMember** cmdlet in Powershell for adding users to administrative groups.

The detection strategy used three conditions:

1. **Use of net.exe or net1.exe**: These binaries were identified as key indicators for potential administrative actions.

2. **Source of the Action from a Non-Whitelisted IP**: Events originating from an unauthorized IP address, outside of the company's defined safe list, signified potential external access. Of note, Azure's Microsoft backbone seems to play a factor here, but in the future, benign IPs could be added to the whitelist to 'tune' the alert.

3. **Account Addition via Specific Commands**: The regex patterns captured the use of commands like localgroup administrators /add and Add-LocalGroupMember -Group "Administrators" -Member, both indicative of a user being added to the Administrators group.

Upon matching all three criteria, the detection rule fired and highlighted the suspicious activities involving account manipulation from an unknown source.

This exercise highlights the importance of monitoring administrative access closely, as it is often a critical entry point for attackers seeking to establish persistence and escalate privileges within an environment. The use of **MITRE ATT&CK tactics and techniques** (Tactic: Persistence (TA0003), Technique: Create Account (T1136)) further aligns this detection with industry-standard defense strategies.

By conducting this threat hunt, we not only detected an immediate threat but also strengthened the security posture by refining our detection capabilities to respond effectively to such activities in the future.