Timeline Summary and Findings:

Instinct-mde-te was found failing several connection requests against itself and another host on the same network

```
DeviceNetworkEvents
| where DeviceName startswith "instinct"
| where ActionType == "ConnectionFailed"
| summarize ConnectionCount = count() by DeviceName, ActionType, LocalIP, RemoteIP
| order by ConnectionCount
```

Filters: Add filter

DeviceName	ActionType	LocalIP	RemoteIP	ConnectionCount
instinct-mde-te	ConnectionFailed	10.1.0.147	10.0.0.5	23
instinct-mde-te	ConnectionFailed	10.1.0.147	168.63.129.16	11
instinct-mde-te	ConnectionFailed	::1	::1	5
instinct-mde-te	ConnectionFailed	10.1.0.147	169.254.169.254	5
instinct-mde-te	ConnectionFailed	127.0.0.1	127.0.0.1	5
instinct-mde-te	ConnectionFailed	10.1.0.147	52.123.129.14	2
instinct-mde-te	ConnectionFailed	10.1.0.147	23.54.127.164	2
instinct-mde-te	ConnectionFailed	10.1.0.147	208.89.73.147	2

—--

After observing failed connection requests from a suspected host (10.1.0.147) in chronological order, I noticed a port scan was taking place due to the sequential order of the ports. There were several port scan being concluded:

```
// Observe all failed connections for the IP in question. Notice anything?
let IPInQuestion = "10.1.0.147";
DeviceNetworkEvents
| where ActionType == "ConnectionFailed"
| where LocalIP == IPInQuestion
| order by Timestamp desc
```

—--

I pivoted to the DeviceProcessEvents table to see if I could see anything that was suspicious around the time the port scan started. I noticed a PowerShell named portscan.ps1 launch at 2025-08-11T10:48:10.0834279Z

```
// Observe DeviceProcessEvents for the past 10 minutes of the unusual activity
 found
let VMName = "instinct-mde-te";
let specificTime = datetime(2025-08-11T10:48:27.8362168Z);
DeviceProcessEvents
| where Timestamp between ((specificTime - 10m) .. (specificTime + 10m))
| where DeviceName == VMName
| order by Timestamp desc
| project Timestamp, FileName, InitiatingProcessCommandLine
```

——---

I logged into the suspect computer and observed the PoweerShell script that was used to conduct the port scan:

```powershell
# Define the log file path
$logFile = "C:\ProgramData\entropygorilla.log"
$scriptName = "portscan.ps1"

# Function to log messages
function Log-Message {
    param (
        [string]$message,
        [string]$level = "INFO"
    )
    $timestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
    $logEntry = "$timestamp [$level] [$scriptName] $message"
    Add-Content -Path $logFile -Value $logEntry
}

# Define the range of IP addresses to scan
$startIP = 4
$endIP = 10
$baseIP = "10.0.0."

# Expanded list of common ports (well-known port numbers 0-1023 + some higher)
$commonPorts = @(21, 22, 23, 25, 53, 69, 80, 110, 123, 135, 137, 138, 139, 143, 161, 194, 443, 445, 465, 587, 993, 995, 3306, 3389, 5900, 8080, 8443)

# Log the start of the scan
Log-Message "Starting port scan on IP range $baseIP$startIP to $baseIP$endIP."

# Function to test a single IP and all its common ports
function Test-Ports {
```

——

We observed the port scan script was launched by the SYSTEM account, this is not expected behaviour and is not something that was setup by the admins, so I Isolated the device and ran a malware scan

——

The malware scan produced no result, so out of caution, I kept the device isolated and put in a ticket to have it reimage/rebuit

——

MITRE ATT&CK Framework Related TTPs:

- T1046: Network Service Discovery – Port scanning activity across sequential ports.

- T1059.001: Command and Scripting Interpreter: PowerShell – Execution of "portscan.ps1".

- T1078: Valid Accounts – Use of SYSTEM account to execute malicious script.

- T1105: Ingress Tool Transfer – Presence of a custom port scanning script on host.

- T1569.002: System Services: Service Execution – Script executed in a system-level context.