Grundlegende Suchanfragen in Umlaufbahnen für Bedrohungsanalysen

Inhalt

Einleitung

Voraussetzungen

- Anforderungen
- Verwendete Komponenten

Hintergrundinformationen

Zugriff

Benutzerdefinierte Abfragen

- 1. Startelemente
- 2.Sha256 Hashes von laufenden Prozessen
- 3. Prozess mit Netzwerkverbindungen
- 4. Privilegierter Prozess mit Netzwerkverbindung ohne lokalen Host
- 5.Überwachung der Registrierung sichern/wiederherstellen
- 6. Dateisuche
- 7.PowerShell-Verlaufsüberwachung
- 8.Prefetch-Abfrage
- 9.Address Resolution Protocol (ARP) Cache Inspection

Einleitung

In diesem Dokument werden die grundlegenden Suchanfragen auf orbitaler Ebene für die Bedrohungsanalyse beschrieben.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse im Hinblick auf das Verständnis von Bedrohungen und Malware sowie grundlegende Kenntnisse von SQL-Tabellen (Structured Query Language) verfügen.

Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- Secure Endpoint Connector Version 7.1.5 oder höher für Windows
- Secure Endpoint Connector Version 1.16 oder höher für Mac

- Secure Endpoint Connector Version 1.17 oder höher für Linux
- Secure Endpoint-Benutzer muss die Rolle des Administrators zugewiesen werden, um Orbital bereitzustellen.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

Hintergrundinformationen

Die benutzerdefinierten Abfragen müssen Ihnen dabei helfen, schnell die Leistungsfähigkeit von Orbital und Soquery für die Jagd nach Bedrohungen zu erlernen.

Orbital verwendet zusätzlich zu Orbital-spezifischen Tabellen osquerys-Lagertabellen. Die von Orbital zurückgegebenen Ergebnisse können an andere Anwendungen gesendet werden, z. B. Secure Endpoint, Secure Malware Analytics und SecureX Threat Response, und können in Remote Data Stores (RDS) wie Amazon S3, Microsoft Azure und Splunk gespeichert werden.

Verwenden Sie die Seite Orbital Investigate (Orbitale Untersuchung), um Live-Abfragen von Endpunkten zu erstellen und auszuführen, um weitere Informationen von diesen zu sammeln. Orbital verwendet osquery, mit dem Sie Ihre Geräte wie eine Datenbank mit grundlegenden SQL-Befehlen abfragen können.

Hier ein einfaches Beispiel: SELECT column1, column2 FROM table1, table2 WHERE column2='value'.

In diesem Beispiel sind column1 und column2 die Feldnamen der Tabelle, aus der Sie Daten auswählen möchten. Verwenden Sie folgende Syntax, um alle in der Tabelle verfügbaren Felder auszuwählen: WÄHLEN SIE * AUS Tabelle 1.

Zugriff

Open Orbital direkt an diesen Standorten:

Nordamerika - https://orbital.amp.cisco.com

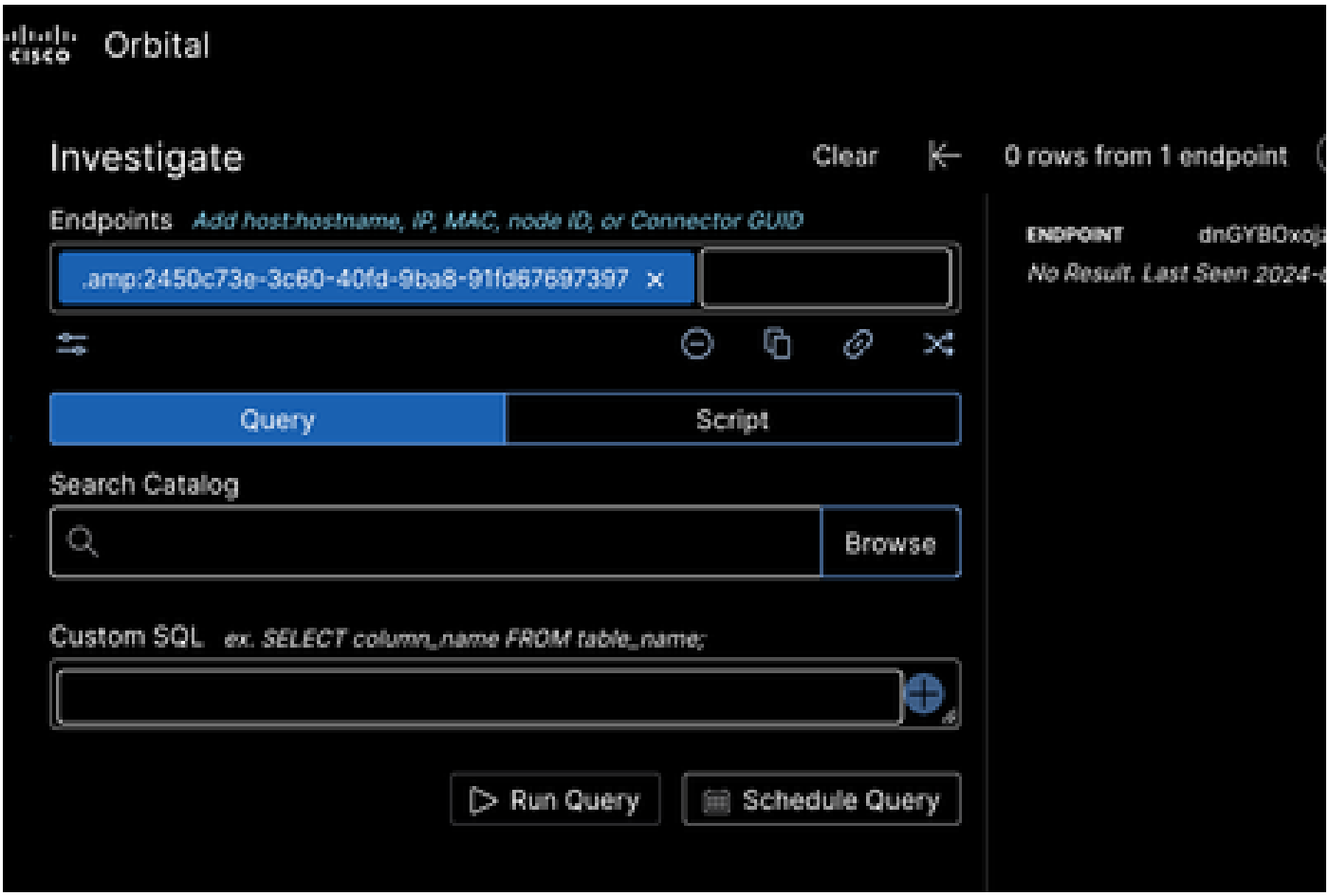
Europa - https://orbital.eu.amp.cisco.com

Asien-Pazifik-Raum - https://orbital.apjc.amp.cisco.com

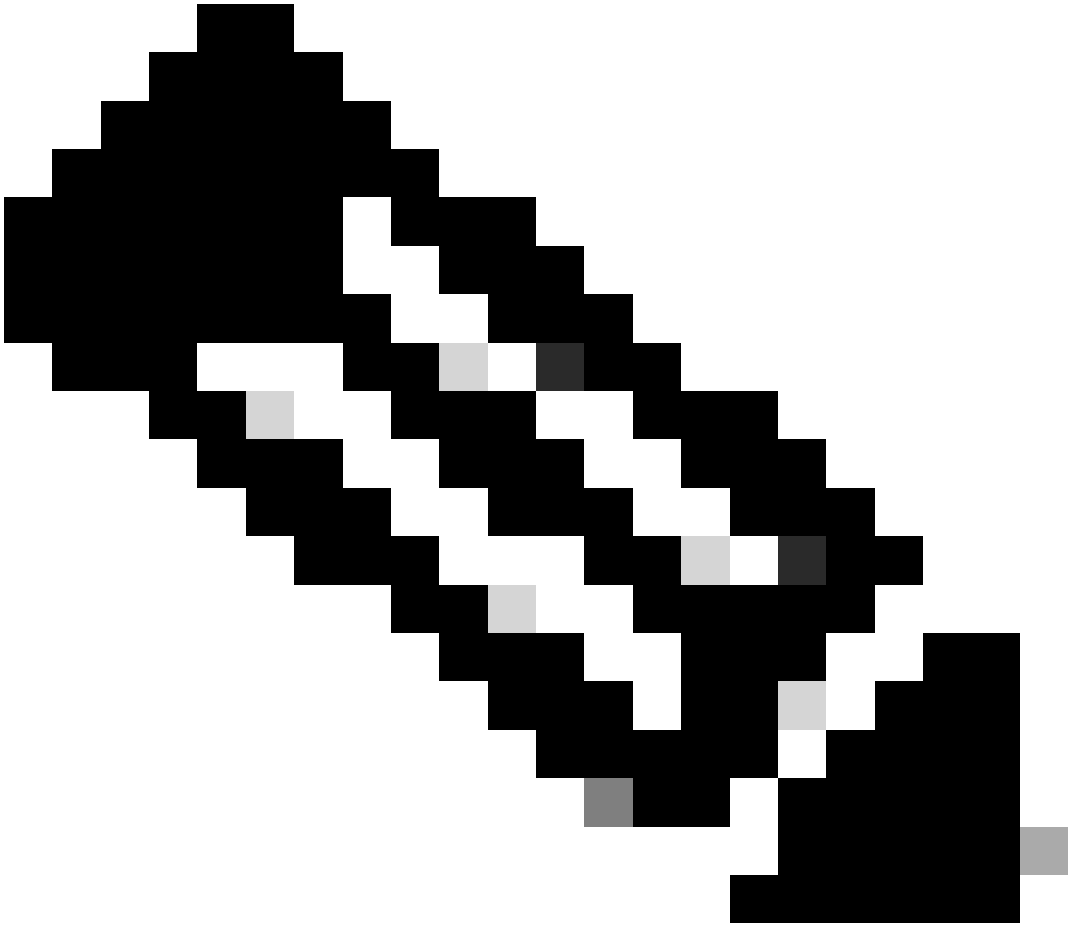
ODER

Wählen Sie in der Konsole für sichere Endgeräte das betroffene Hostsystem aus, und klicken Sie auf Investigate in Orbital (In Orbit untersuchen).

Es gibt Optionen für die Verwendung des orbital-Katalogs (Klicken Sie auf Durchsuchen) oder die Eingabe der benutzerdefinierten Abfragen unter Custom SQL, wie erwähnt:



Benutzerdefinierte Abfragen

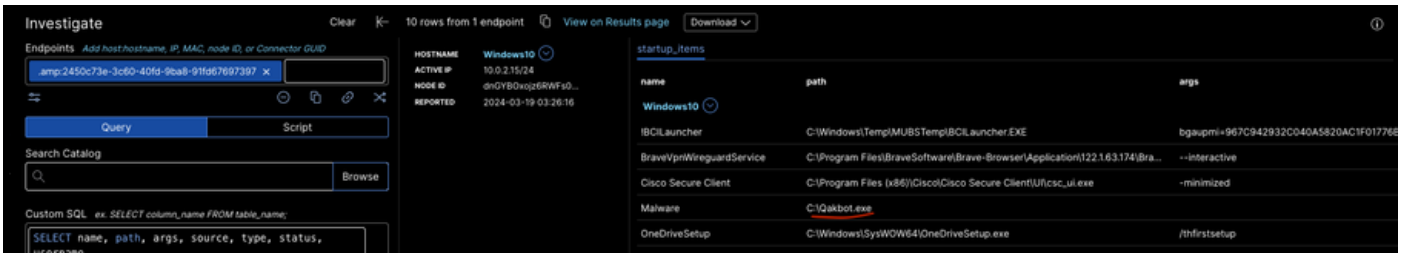


Anmerkung: Das Hostsystem befindet sich im Labornetzwerk und versucht, das System/Netzwerk nicht zu beschädigen.

1. Startelemente

Startelemente können von Angreifern ausgenutzt werden, um die Persistenz auf einem kompromittierten System aufrechtzuerhalten. Das bedeutet, dass die bösartige Software weiterhin ausgeführt wird oder bei jedem Systemneustart automatisch neu gestartet wird. Im nächsten Beispiel wird Qakbot.exe auf dem Hostsystem ausgeführt.

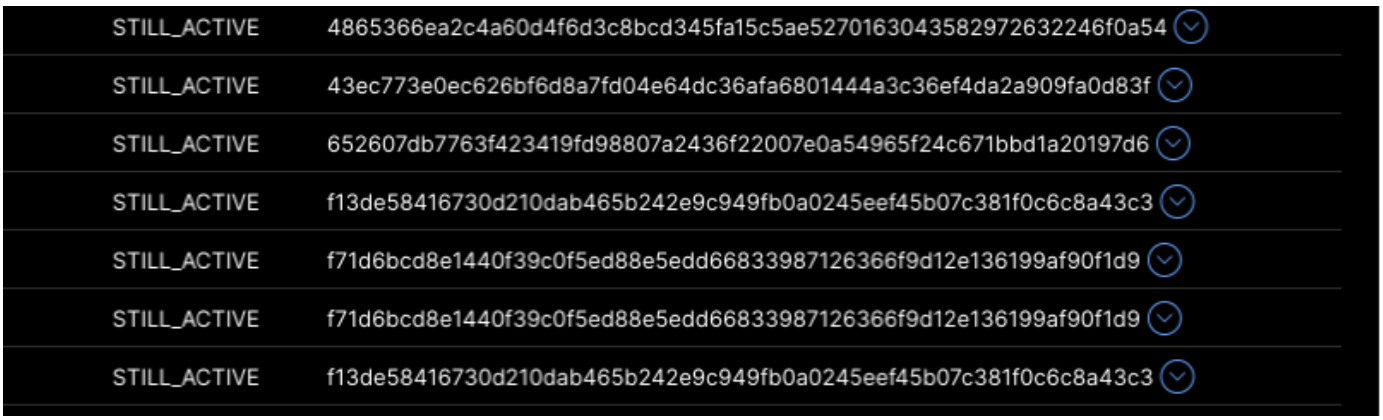
```
SELECT name, path, args, source, type, status, username
FROM startup_items;
```



2. Sha256 Hashes von laufenden Prozessen

SHA256-Hashes sind nicht automatisch mit Prozessen in ihrem natürlichen Zustand verknüpft. Sicherheitssoftware und Systemüberwachungstools können jedoch den SHA256-Hash eines laufenden Prozesses der ausführbaren Datei berechnen, um deren Integrität und Authentizität zu überprüfen.

```
 SELECT
p.pid, p.name, p.path, p.cmdline, p.state, h.sha256
FROM processes p
INNER JOIN hash h
ON p.path=h.path;
```

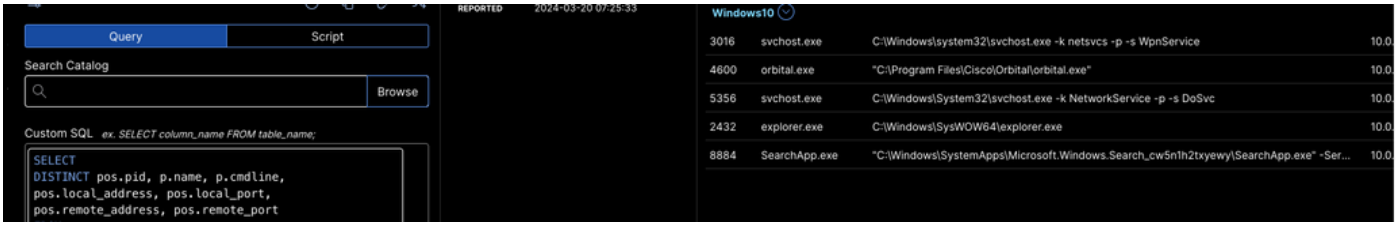


Wenn der zugehörige Hash einer Datei schädlich ist, können Sie sich mit dieser Abfrage identifizieren.

3. Prozess mit Netzwerkverbindungen

Prozesse mit Netzwerkverbindungen sind Programme oder Systemdienste, die die Netzwerkschnittstelle aktiv nutzen, um mit anderen Geräten in einem Netzwerk oder über das Internet zu kommunizieren.

```
SELECT
DISTINCT pos.pid, p.name, p.cmdline, pos.local_address, pos.local_port, pos.remote_address, pos.remote_
FROM processes p
JOIN process_open_sockets pos USING (pid)
WHERE
pos.remote_address NOT IN ("", "0.0.0.0", "127.0.0.1", "::", "::1", “0");
```



4. Privilegierter Prozess mit Nicht-Localhost-Netzwerkverbindung

Ausführung von Programmen oder Diensten mit erhöhten Berechtigungen (wie denen eines Administrator- oder Systemkontos), die über das Netzwerk mit einem externen Gerät oder Dienst kommunizieren, d. h. mit jeder anderen IP-Adresse als 127.0.0.1 (localhost) oder ::1 (IPv6 localhost).

```
SELECT DISTINCT p.name, p.cmdline, pos.pid, pos.local_address, pos.local_port, pos.remote_address, pos.
FROM processes p JOIN process_open_sockets pos USING (pid)
WHERE pos.remote_address NOT IN ("", "0.0.0.0", "127.0.0.1", "::", "::1")
```

Sobald Sie die Liste Packet Identifier (PID) haben, können Sie sie in den benutzerdefinierten Abfragen entsprechend hinzufügen.

```
SELECT DISTINCT p.name, p.cmdline, pos.pid, pos.local_address, pos.local_port, pos.remote_address, pos.
FROM processes p JOIN process_open_sockets pos USING (pid)
WHERE pos.remote_address NOT IN ("", "0.0.0.0", "127.0.0.1", "::", "::1") and p.uid=1436
```

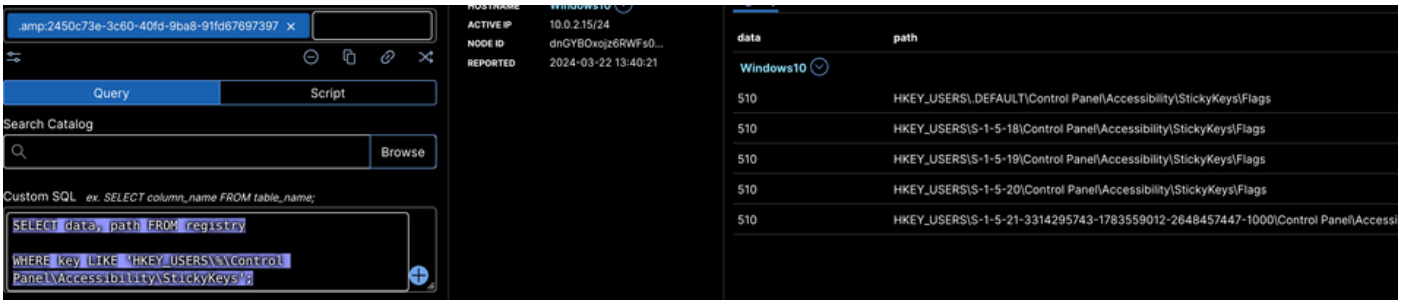
5. Sichern/Wiederherstellen der Registrierungsüberwachung

Nachverfolgung von Ereignissen, bei denen Änderungen an der Windows-Registrierung vorgenommen werden, durch Sicherungs- oder Wiederherstellungsvorgänge. Die Windows-

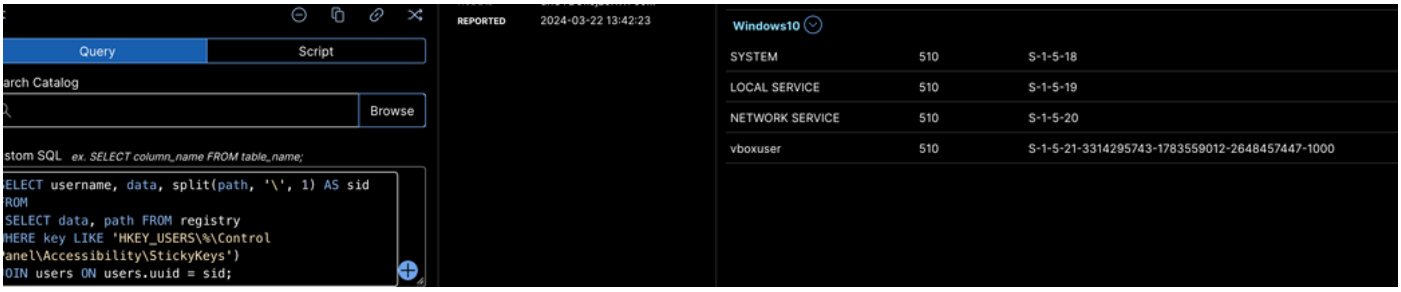
Registrierung ist eine hierarchische Datenbank, die Konfigurationseinstellungen und -optionen unter Microsoft Windows-Betriebssystemen speichert.

```
SELECT key AS reg_key, path, name, data, DATETIME(mtime, "unixepoch") as last_modified
FROM registry
WHERE key LIKE "HKEY_LOCAL_MACHINE\system\currentcontrolset\control\backuprestore\filesnottosnapshot";
```

```
SELECT data, path FROM registry
WHERE key LIKE 'HKEY_USERS\%\Control Panel\Accessibility\StickyKeys';
```




```
SELECT username, data, split(path, '\', 1) AS sid
FROM
(SELECT data, path FROM registry
WHERE key LIKE 'HKEY_USERS\%\Control Panel\Accessibility\StickyKeys')
JOIN users ON users.uuid = sid;
```



6. Dateisuche

Ermöglicht es Benutzern, Dateien und Ordner auf ihrem Computer anhand verschiedener Kriterien wie Dateiname, Inhalt, Eigenschaften oder Metadaten zu suchen.

```
SELECT
f.directory, f.filename, f.uid, f.gid,
f.mode, f.device, f.size,
```

```
f.atime,
f.mtime,
f.ctime,
f.btime,
f.hard_links, f.symlink, f.file_id, h.sha256
FROM file f
LEFT JOIN hash h on f.path=h.path
WHERE
f.path LIKE (SELECT v from __vars WHERE n="file_path") AND
f.path NOT LIKE (SELECT v from __vars WHERE n="not_file_path");
```

Navigieren Sie zu PARAMETERS > File Path, und klicken Sie auf %.dll, %.exe oder %.png.

7. PowerShell-Verlaufsüberwachung

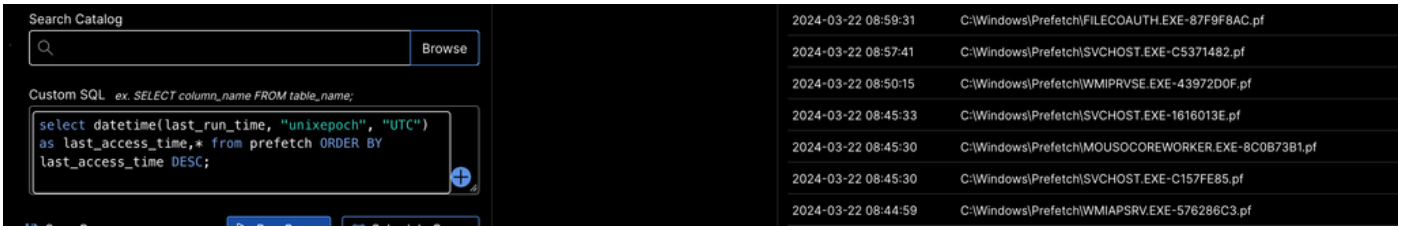
Praktische Vorgehensweise zum Nachverfolgen der Befehle, die in PowerShell-Sitzungen ausgeführt wurden. Die Überwachung des PowerShell-Verlaufs kann aus Sicherheits- und Compliance-Gründen besonders wichtig sein.

```
SELECT time, datetime, script_block_id, script_block_count, script_text, script_name, script_path
FROM orbital_powershell_events
ORDER BY datetime DESC
LIMIT 500;
```

8. Abfrage vorab abrufen

Leistungsmerkmal, das das Laden von Anwendungen beschleunigt. Beim Prefetching wird analysiert, wie Software auf einem System geladen und ausgeführt wird, und die entsprechenden Informationen werden dann in bestimmten Dateien gespeichert.

```
select datetime(last_run_time, "unixepoch", "UTC") as last_access_time,*
from prefetch
ORDER BY last_access_time DESC;
```

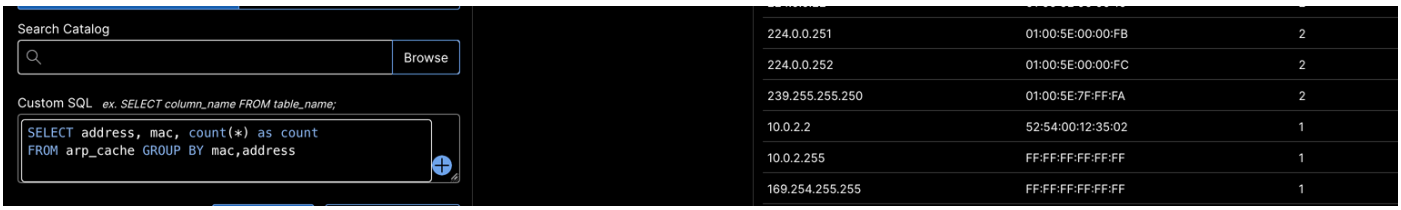


Prefetch ist ein Mechanismus, mit dem SQL Server viele E/A-Anfragen parallel für einen Nested Loop-Join auslösen kann.

9. Address Resolution Protocol (ARP) Cache Inspection

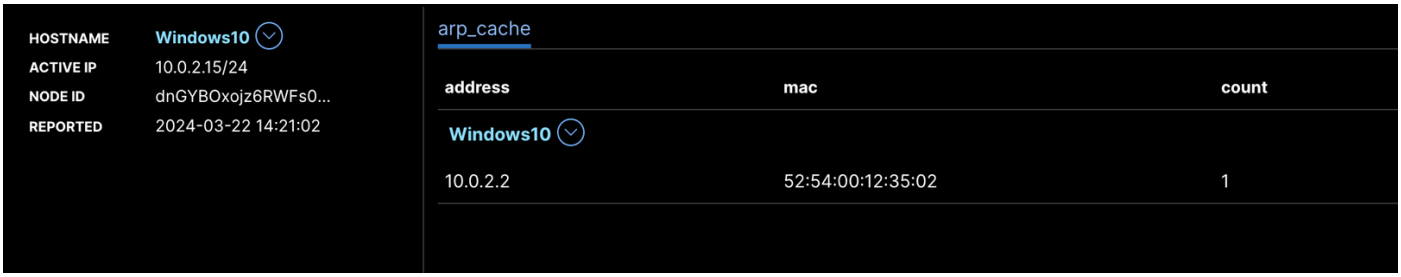
Untersuchen des Inhalts des ARP-Cache auf einem Computer oder Netzwerkgerät. Der ARP-Cache ist eine Tabelle, in der Zuordnungen zwischen IP-Adressen und den entsprechenden MAC-Adressen gespeichert werden.

```
SELECT address, mac, count(*) as count
FROM arp_cache GROUP BY mac,address;
```



Im nächsten Beispiel werden die verdächtige MAC-Adresse und deren Anzahl aus dem ARP-Cache ermittelt.

```
SELECT address, mac, count(*) as count
FROM arp_cache GROUP BY mac,address
HAVING COUNT(mac) >= (SELECT count FROM arp_cache WHERE count>=1)
AND mac LIKE (SELECT mac FROM arp_cache WHERE mac="52:54:00:12:35:02");
```



Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.