脅威分析のための基本的な軌道検索クエリー

内容

はじめに

このドキュメントでは、脅威分析のための基本的なオービタル検索クエリについて説明します。

前提条件

要件

脅威とマルウェアについて理解し、Structured Query Language（SQL；構造化照会言語）テーブルについて基本的な知識があることが推奨されます。

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Secure Endpoint Connectorバージョン7.1.5以降（Windows用）
- Mac用のSecure Endpoint Connectorバージョン1.16以降
- Secure Endpoint Connectorバージョン1.17以降（Linux用）
- Orbitalを展開するには、セキュアエンドポイントユーザにadminのロールを割り当てる必要があります

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このド

キュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

カスタムクエリは、脅威ハンティングのためのOrbitalとosqueryのパワーをすばやく学ぶのに役立つ必要があります。

Orbitalは、Orbital固有のテーブルに加えて、osquerysストックテーブルを使用します。Orbitalを通じて返された結果は、Secure Endpoint、Secure Malware Analytics、SecureX Threat Responseなどの他のアプリケーションに送信でき、Amazon S3、Microsoft Azure、Splunkなどのリモートデータストア(RDS)に保存できます。

エンドポイントからより多くの情報を収集するために、エンドポイントでライブのライブクエリを構築して実行するには、Orbital Investigateページを使用します。Orbitalはosqueryを使用します。これにより、基本的なSQLコマンドを使用したデータベースのようにデバイスを照会できます。

以下に簡単な例を示します。 SELECT column1, column2 FROM table1, table2 WHERE column2='value'.

この例では、列1と列2は、データの選択元となるテーブルのフィールド名です。テーブルで使用可能なすべてのフィールドを選択するには、SELECT * FROM table1構文を使用します。

Access

Orbitalを次のサイトで直接開きます。

北米：https://orbital.amp.cisco.com

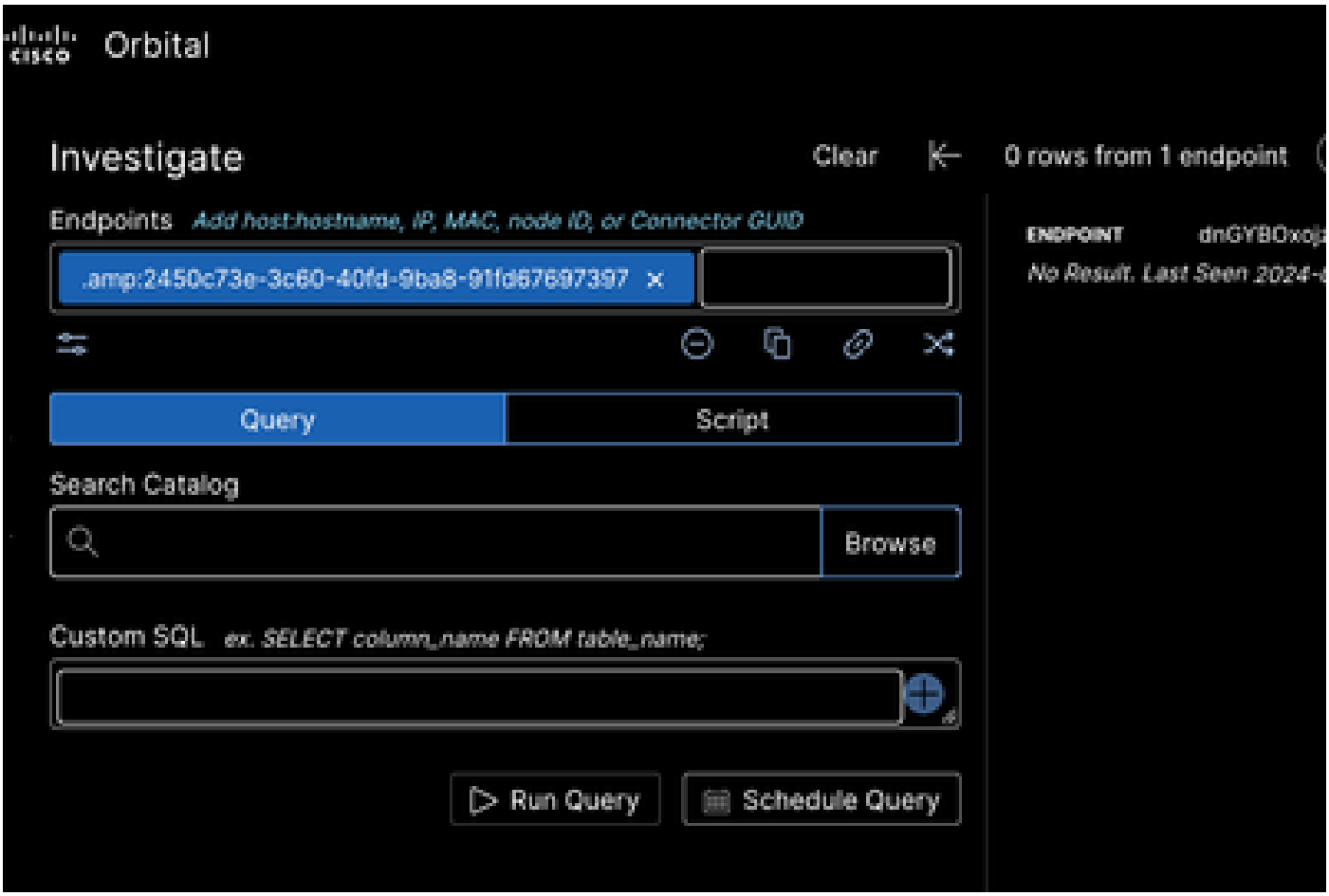
ヨーロッパ：https://orbital.eu.amp.cisco.com

アジア太平洋 – https://orbital.apjc.amp.cisco.com

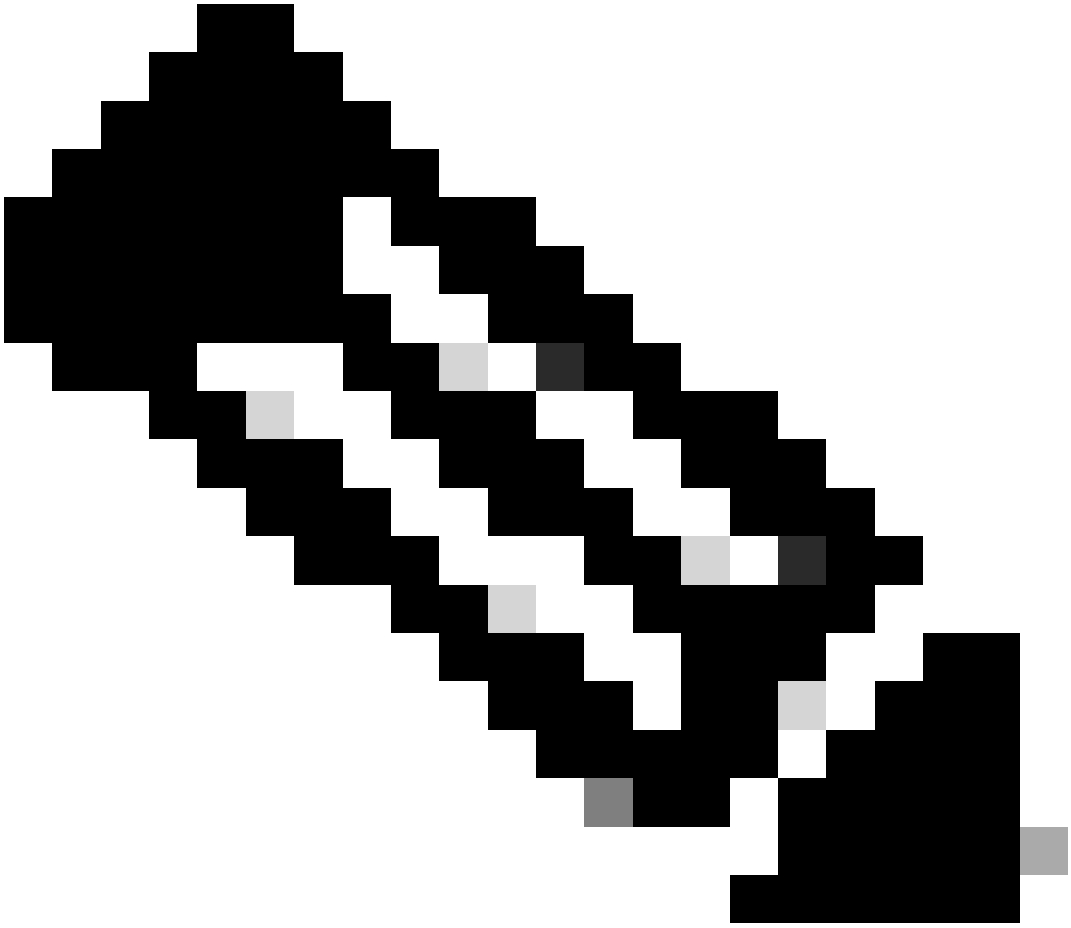
または

Secure Endpoint Consoleで、影響を受けるホストシステムを選択し、Investigate in Orbitalをクリックします。

Orbitalカタログを使用する(Browseをクリックする)か、前述のようにCustom SQLセクションにカスタムクエリを入力するオプションがあります。



カスタムクエリ

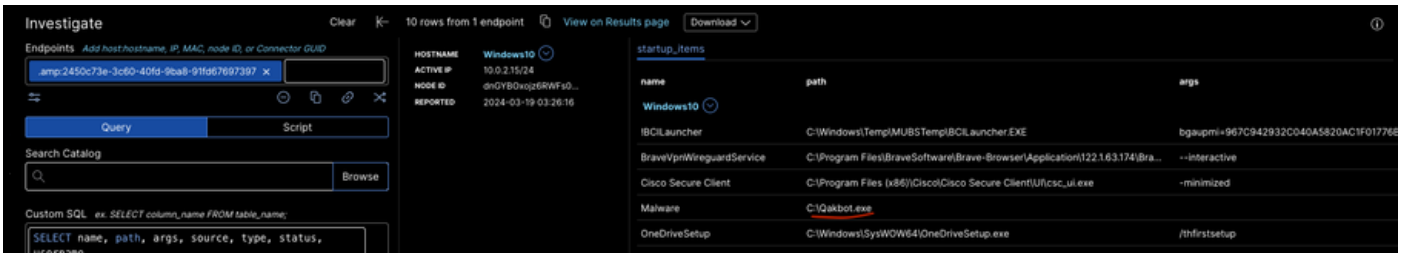


注：ホストシステムはラボネットワーク内にあり、システムやネットワークに影響が及ばないように試みられています。

1. スタートアップ項目

スタートアップ項目は、攻撃者によって悪用され、侵害されたシステムで持続性を維持される可能性があります。つまり、システムを再起動するたびに、悪意のあるソフトウェアが自動的に実行または再起動され続けます。次の例では、ホストシステムでQakbot.exeが実行されています。

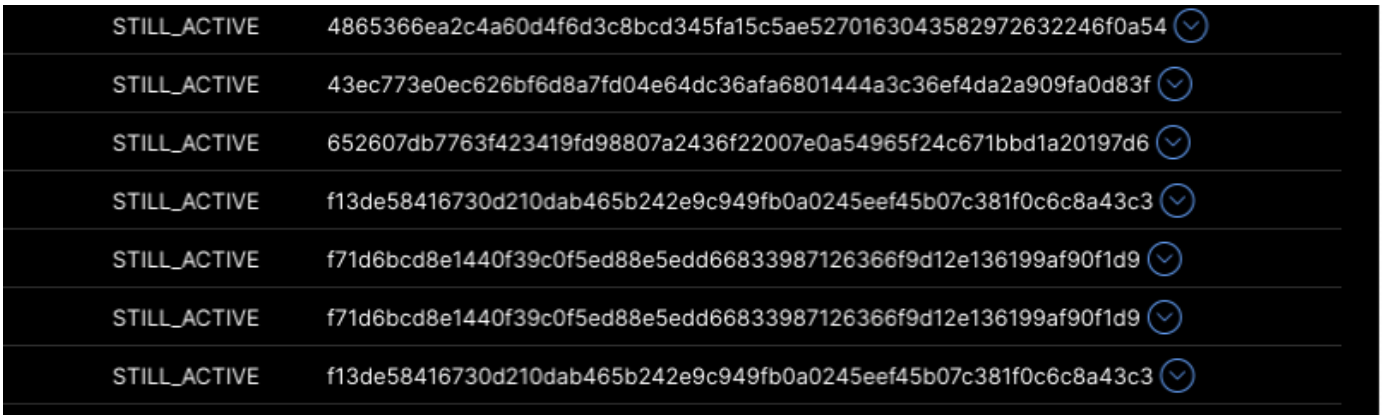
```
SELECT name, path, args, source, type, status, username
FROM startup_items;
```



2. 実行中のプロセスのSha256ハッシュ

SHA256ハッシュは、本質的にプロセスを自然な状態で実行することに関連付けられていません。ただし、セキュリティソフトウェアおよびシステム監視ツールは、実行可能ファイルの実行中のプロセスのSHA256ハッシュを計算して、その整合性と信頼性を確認できます。

```
 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;
```

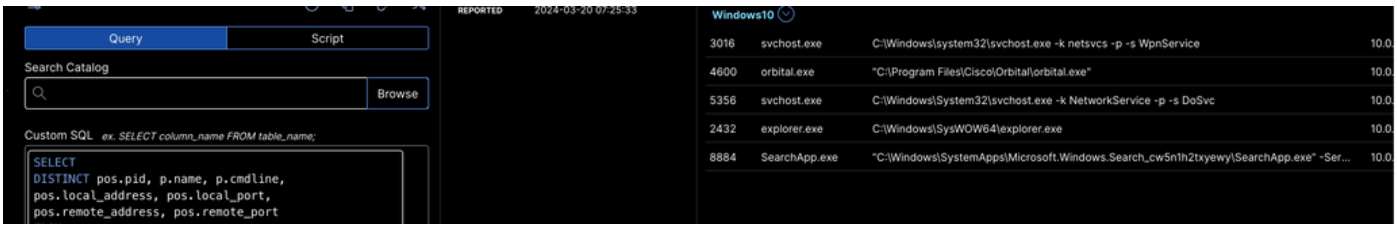


ファイルの関連するハッシュが悪意のあるハッシュである場合、このクエリで識別できます。

3. ネットワーク接続を使用したプロセス

ネットワーク接続を使用するプロセスは、ネットワーク上またはインターネット上の他のデバイスと通信するためにネットワークインターフェイスをアクティブに使用しているプログラムまたはシステムサービスです。

```
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. 非ローカルホストネットワーク接続での特権プロセス

アクセス許可が昇格されたプログラムまたはサービス (管理者アカウントまたはシステムアカウントのアクセス許可など) を実行しており、外部のデバイスまたはサービスとネットワーク経由で通信している。つまり、127.0.0.1 (localhost)または ：:1 (IPv6 localhost)以外の任意のIPアドレス。

```
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")
```

パケット識別子(PID)リストを作成したら、それに応じてカスタムクエリに追加できます。

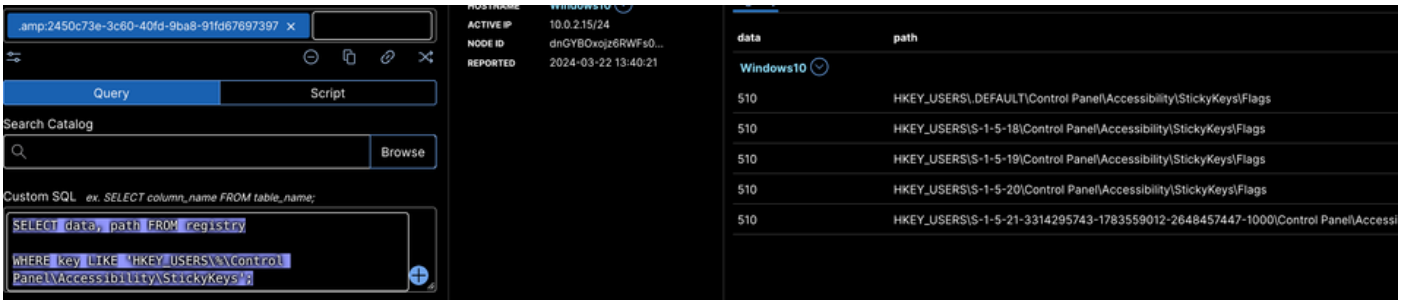
```
SELECT DISTINCT p.name, p.cmdline, pos.pid, pos.local_address, pos.local_port, pos.remote_address, pos.
FROM processes p JOIN process_open_sockets pos USING (pid)
WHERE pos.remote_address NOT IN ("", "0.0.0.0", "127.0.0.1", "::", "::1") and p.uid=1436
```

5. レジストリのバックアップ/復元の監視

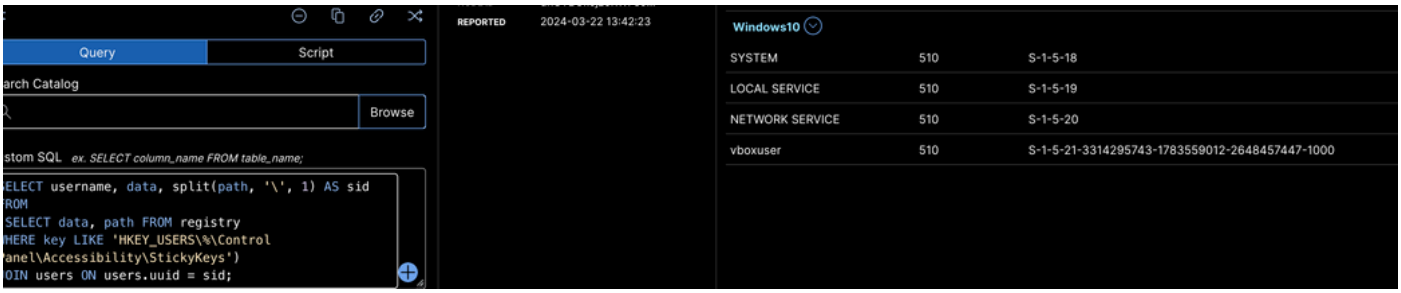
バックアップ操作または復元操作によってWindowsレジストリが変更された場合のイベントの追跡。Windowsレジストリは、Microsoft Windowsオペレーティングシステムの構成設定とオプションを格納する階層データベースです。

```
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. ファイル検索

ユーザーは、ファイル名、コンテンツ、プロパティ、メタデータなどのさまざまな条件を使用して、コンピューター上のファイルやフォルダーを検索できます。

```
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");
```

PARAMETERS > File Pathの順に移動し、%.dll、%.exe、または%.pngをクリックします。

7. Powershell履歴の監視

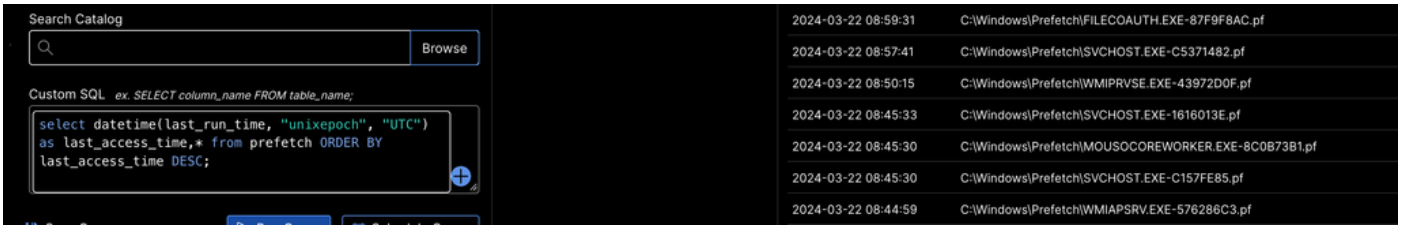
PowerShellセッションで実行されたコマンドを追跡する方法。 PowerShell履歴の監視は、セキュリティとコンプライアンスの観点から特に重要です。

```
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. プリフェッチクエリ

アプリケーションのロードを高速化するパフォーマンス機能。プリフェッチでは、ソフトウェアをシステムにロードして実行する方法を分析し、その情報を特定のファイルに保存します。

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

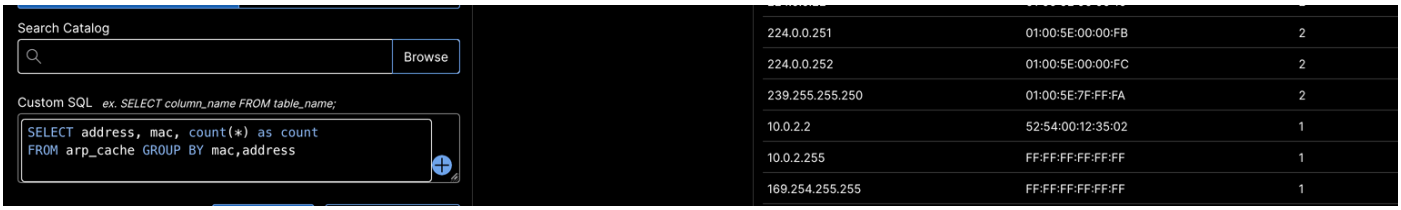


プリフェッチは、SQL Serverがネスト・ループ・ジョインに対して多数のI/O要求を並行して起動できるメカニズムです。

9. アドレス解決プロトコル(ARP)キャッシュインスペクション

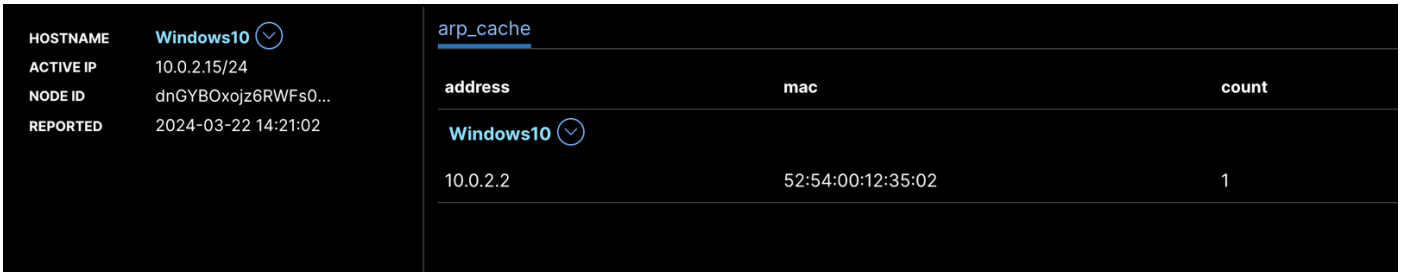
コンピュータまたはネットワークデバイス上のARPキャッシュの内容を確認する。ARPキャッシュは、IPアドレスとそれに対応するMACアドレス間のマッピングを格納するテーブルです。

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



次の例では、ARPキャッシュから不審なMACアドレスとその数を特定します。

```
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");
```



翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。