FULL REPORT · C2 INFRASTRUCTURE · MONOGRAPH 68 / 67 · 18 MIN READ

# AS40065 Multi-Service Cloud Nodes and Rackspace Spot Overlap

Mapping 598 candidate endpoints across 113 multi-service CloudRadium hosts, examining the 24 November 2026 Rackspace Spot overlap, and hardening Kubernetes Secrets and pod admission


BY
Odin by Brandefense



PUBLISHED
19 November 2026

READING TIME
18 min read

EDITION
Version 1.1 · TLP:CLEAR

SCOPE
Prepared November 2026


## 1. Executive Summary


IN SHORT

Through 25 November 2026, Odin by Brandefense evaluated 598 candidate endpoints across 401 IPv4s on AS40065 (CNSERVERS LLC). Thirty-two endpoints were active in the two-week window, and 113 multi-service hosts held 310 endpoints. The intrusion IPv4 reported in the November 2026 Rackspace Spot Kubernetes breach falls within AS40065. Restrict Secret RBAC, enforce Pod Security and volume admission policy, and block high-port egress.


Cloud providers that allow fast server setup and broad port binding host both benign test labs and attacker tools. Reading threat feeds one port at a time hides how users stack scanners, relays, autonomous panels, and C2 listeners onto a single IPv4 host. Whereas Monograph 60 covered seven-framework growth and Storm-0940 overlaps, Monograph 68 examines 598 AS40065 endpoints (113 multi-service hosts) and Rackspace Spot Kubernetes Secret and pod hardening.

During a two-week observation window ending 25 November 2026, Odin by Brandefense reviewed 598 candidate endpoints on 401 IPv4s in AS40065 ( `CNSERVERS LLC` / `cloudradium[.]com` ). Within those 598 indexed records, 32 endpoints across 18 hosts carried active last-seen timestamps from 12 to 25 November 2026. The other 566 records carried earlier baseline timestamps before 12 November 2026.

Every one of the 598 records carries candidate status without independent C2 beacon captures. Odin by Brandefense therefore withholds all 401 IPv4 addresses. Zero blocklist IPs or file SHA-256 hashes appear in our feeds.

1. **Multi-service hosting places 51.8 percent of AS40065 endpoints onto 113 servers.**
   Across 401 unique IPv4s, 288 hosts (71.8 percent) exposed one candidate endpoint, while 113 hosts (28.2 percent) exposed two to eight endpoints each. Those 113 multi-service hosts carry 310 of the 598 endpoints (51.8 percent), peaking at eight endpoints on one IPv4. Ninety-four hosts bound multiple ports of one tool, and 19 hosts paired two distinct tools on one box.

2. **Six framework groups reconcile all 598 endpoints and show a November shift to autonomous panels.**
   Cobalt Strike and unverified profiles lead with 294 endpoints (49.2 percent), followed by recon scanners with 143 endpoints (23.9 percent, including 108 ARL panels). Post-exploitation C2s add 54 endpoints, RATs and other panels add 52, autonomous consoles add 33, and relays add 22.

Among the 32 endpoints active from 12 to 25 November 2026, autonomous consoles account for 24 endpoints (75.0 percent). Nineteen of those active ports ran ARTEX, alongside three AutoCVE ports and two CyberStrikeAI ports.

3. **The single reported Rackspace Spot intrusion IPv4 falls within AS40065.**
   In late November 2026, Rackspace customer notices disclosed a 17 November 2026 credential-based intrusion into its managed Kubernetes service Rackspace Spot. Intruders entered customer clusters, created unauthorized pods, listed nodes and volumes, and may have read Kubernetes Secrets and storage-volume files.

The reported Rackspace Spot intrusion IP falls within AS40065 ( `CNSERVERS LLC` / `cloudradium[.]com` ); no evidence links it to a specific console or service.

4 **Shared ASN overlap neither attributes the breach to tool authors nor names a threat group.** Open-source testing consoles, scanners, and relays run across shared clouds, and 113 AS40065 hosts expose multiple candidate endpoints across standard and high ports. Shared ASN hosting alone cannot show which service or tenant sent the 17 November cluster API calls. Defenders should skip blanket ASN IP blocks and harden Kubernetes clusters directly.

## Three Decisions for Security Leaders

1. **Lock down Kubernetes Secret RBAC and storage encryption.** Remove cluster-wide `get` , `list` , and `watch` verbs on `Secrets` to stop bulk API reads, and enable KMS v2 encryption for `etcd` at rest. Disable automatic `ServiceAccount` token mounting on pods to reduce credentials in files exposure.
2. **Enforce Restricted Pod Security, RBAC, and volume admission policy.** Use `Restricted` Pod Security to reject `privileged` pods, `hostNetwork` , and `hostPath` mounts, and use RBAC and admission policy to restrict pod creation and `persistentVolumeClaim` mounts. Alert on imperative `kubectl run` calls outside approved CI pipelines.
3. **Sweep cloud ports and block non-standard high-port egress.** Audit corporate cloud subnets across standard web ports and high scanner ports (**8787**, **50003**, **5003**, **7880**, **13000**). Block server and worker VLANs from direct outbound calls to uncategorized high ports.

## The Study in Numbers

| Number | What it is |
|---|---|
| **598** | Cumulative deduplicated candidate endpoints on AS40065 ( `CNSERVERS LLC` / `cloudradium[.]com` ) evaluated through 25 November 2026 |
| **401** | Unique IPv4 addresses across AS40065 (**288** single-endpoint hosts and **113** multi-service hosts; all **401** IPs withheld) |
| **310** | Candidate endpoints (**51.8%** of **598**) concentrated on the **113** multi-service hosts (**2** to **8** endpoints per IP) |
| **32** | Candidate endpoints active during the `12-25 November 2026` two-week window across **18** IPs (**24** autonomous console endpoints) |
| **94 + 19** | Ninety-four same-tool multi-port hosts plus **19** cross-tool hosts reconciling all **113** multi-service `AS40065` servers |
| **0** | Published blocklist IPv4 addresses or file SHA-256 hashes in our feeds (**10** protocol, CLI, and Kubernetes hunt markers) |

## 2. Six Framework Groups and Port Density on AS40065

Single-ASN analysis shows how diverse toolkits share commercial cloud space ( `[REF-10]` ). On `AS40065` ( `CNSERVERS LLC` / `cloudradium[.]com` ), Odin by Brandefense reviewed **598** candidate endpoints on **401** unique IPv4s ( `CloudRadium L.L.C` ) through `25 November 2026` . Teams can use external attack surface management to audit cloud subnets on both standard HTTPS ports and high management ports. Figure 1 maps the four study tracks.

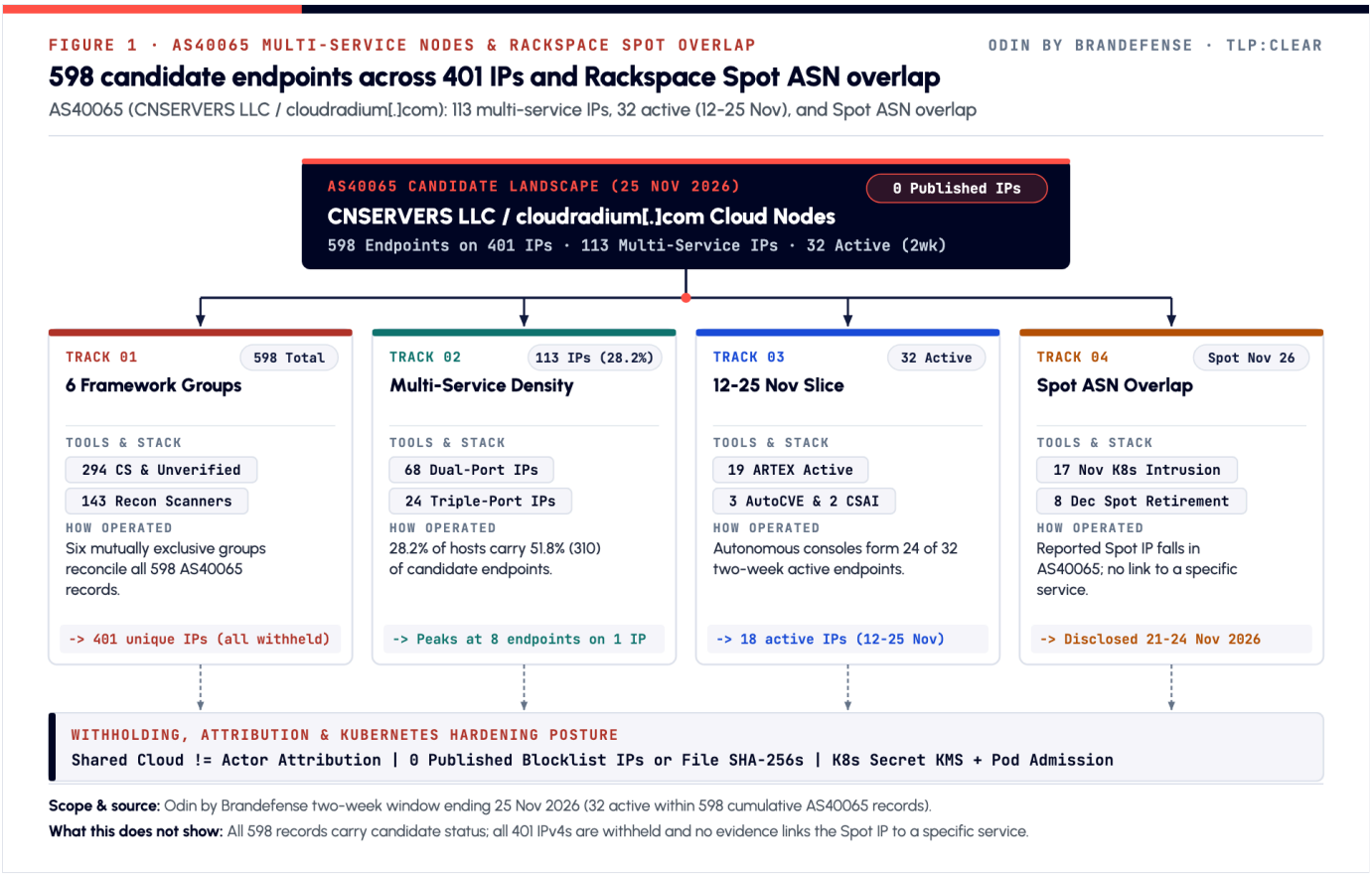


Figure 1: 598 candidate endpoints across 401 IPs and Rackspace Spot ASN overlap

All **598** records (**100.0%**) hold `candidate` status (**0** confirmed beacon captures). By country, **518** endpoints (**86.6%**) map to the United States and **67** (**11.2%**) to Hong Kong, plus **8** (**1.3%**) in China and **5** (**0.8%**) in Mauritius. Upstream role tags span `c2` (**300**), `red-team-tools` (**162**), `team-server` (**77**), `management` (**29**), and `admin-panel` (**22**). The remaining eight rows carry `phishing` (**4**), `victim` (**3**), and `exploit-server` (**1**) tags.

## Separating 32 Two-Week Active Endpoints from 566 Earlier Records

To keep time boundaries clear, we separate recent two-week activity from older upstream record dates. Within the **598** `AS40065` records reviewed on `25 November 2026`, **32 endpoints on 18 IPv4s** carried last-seen dates from `12 to 25 November 2026`. Twenty-two of those **32** active endpoints (`across 8 IPs`) were first seen during that same `12-25 November` window.

The other **566 candidate endpoints across 387 unique IPv4s** carry earlier upstream first-seen and last-seen dates before `12 November 2026` (`32 + 566 = 598`). Four IPv4 hosts held both an earlier baseline record and a `12-25 November` active record, reconciling `18 + 387 - 4 = 401` unique ASN IPv4s. Keeping those slices distinct stops older `Cobalt Strike` and scanner churn from hiding late-November console trends.

## Reconciling All 598 Endpoints Across Six Framework Groups

Table 1 groups all **598** `AS40065` records into six mutually exclusive framework groups (`294 + 143 + 54 + 52 + 33 + 22 = 598`). Pairing this breakdown with cyber threat intelligence helps defenders weigh historical C2 density against active autonomous testing panels.

| Framework group | Cumulative endpoints | Share (%) | Unique IPs in group | Active (`12-25 Nov`) | Observed tool breakdown |
|---|---|---|---|---|---|
| Cobalt Strike & Unverified Profiles | 294 | 49.2 % | 143 | 1 | `Cobalt Strike` (276), `CS Unverified` (18). |

| Framework group | Cumulative endpoints | Share (%) | Unique IPs in group | Active (12-25 Nov) | Observed tool breakdown |
|---|---|---|---|---|---|
| Recon & Vulnerability Scanners | 143 | 23.9% | 133 | 3 | ARL (108), Acunetix (21), Yakit (13), reNgine (1). |
| Interactive Post-Exploitation C2s | 54 | 9.0% | 54 | 0 | Viper (23), VShell (13), Brute Ratel C4 (7), Metasploit (4), Havoc (3), Sliver (2), Supershell (2). |
| RATs, Loaders, Phishing & Other Panels | 52 | 8.7% | 50 | 1 | PlugX (9), AsyncRAT (6), Kaiji (5), GoPhish (4), PrismX (4), Emotet (4), QakBot (4), XenoRAT (3), Ramnit (3), BeEF (3), plus 7 single-endpoint tools. |
| Autonomous & AI Testing Consoles | 33 | 5.5% | 17 | 24 | ARTEX (22 on 8 IPs), CyberStrikeAI (7 on 6 IPs), AutoCVE (3 on 3 IPs), PentAGI (1 on 1 IP). |
| C2 Fronting, Reverse SSH & Relays | 22 | 3.7% | 16 | 3 | RedGuard (13), ReverseSSH (8), FRP (1). |
| Total | 598 | 100.0% | 401 unique ASN IPs | 32 | Group IPs sum to 413 due to 12 cross-group IP overlaps. |

Why do the six group IP counts add up to **413** (**143 + 133 + 54 + 50 + 17 + 16**) when `AS40065` has **401** unique IPv4s? Twelve cross-group multi-service hosts appear in two groups ( `413 - 12 = 401` ). In the fourth row, the seven single-endpoint tools (**1** each) are `Hashtopolis`, `Moobot`, `Hookbot`, `Responder`, `PupyC2`, `Meterpreter`, and `Bumblebee`. Figure 5 compares cumulative endpoints with group IP counts and two-week active totals.

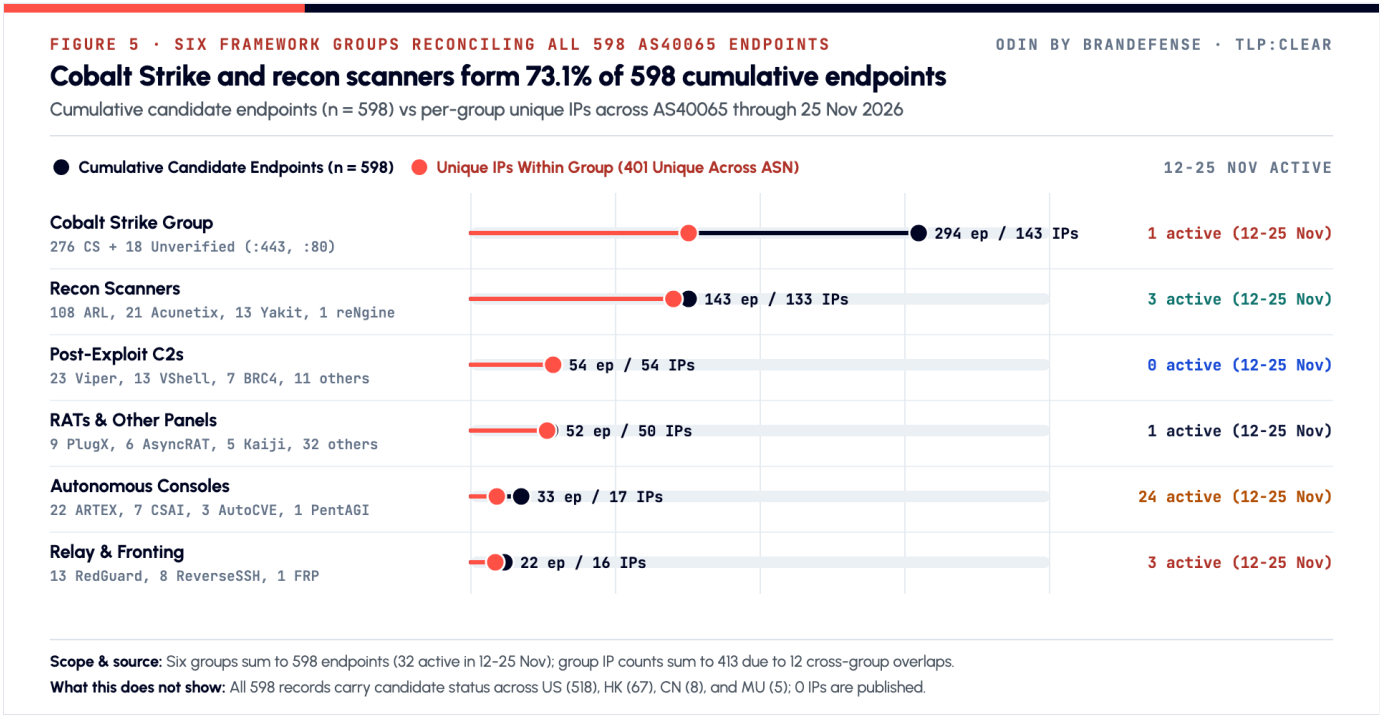


Figure 5: Cobalt Strike and recon scanners form 73.1% of 598 cumulative endpoints

Three clear patterns stand out in Table 1 and Figure 5:

- **High port-per-host density in `Cobalt Strike` and `ARL`.** `Cobalt Strike` (**294** endpoints on **143** IPs, or **2.06** per IP) and recon scanners (**143** endpoints) form **73.1%** (**437** of **598**) of cumulative records.
- **Autonomous consoles dominate the `12-25 November` active slice.** Autonomous testing consoles form just **5.5%** (**33** of **598**) of cumulative records, yet they drive **75.0% (24 of 32)** of two-week active endpoints. Those **24** active ports span **19** `ARTEX`, **3** `AutoCVE`, and **2** `CyberStrikeAI`

bindings. Eight other active ports span **3** `Yakit`, **2** `ReverseSSH`, **1** `FRP`, **1** `Cobalt Strike`, and **1** `Hashtopolis` listener (`24 + 8 = 32`).

- **Dedicated fronting and tunnel relays.** Twenty-two relay endpoints span **13** `RedGuard` C2-fronting reverse proxies, **8** `ReverseSSH` services, and **1** `FRP` tunnel port across **16** hosts.

## Port Distribution Across 120 Distinct TCP Ports

Across all **598** candidate endpoints, services span **120 distinct TCP ports**. Table 2 reconciles the top **12** ports (**366** endpoints, **61.2%**) and the other **108** ports (**232** endpoints, **38.8%**).

| TCP port | Candidate endpoints | Share (%) | Primary observed tools on port |
|---|---|---|---|
| `TCP 443` | 92 | 15.4% | `Cobalt Strike` (51), `ARTEX` (6), `CS Unverified` (6), `ARL` (5), others (24). |
| `TCP 50003` | 58 | 9.7% | `ARL` (58) alternate management console port. |
| `TCP 5003` | 45 | 7.5% | `ARL` (45) default web console port. |
| `TCP 80` | 43 | 7.2% | `Cobalt Strike` (31), `ARTEX` (4), `Ramnit` (2), `VShell` (1), others (5). |
| `TCP 50050` | 28 | 4.7% | `Cobalt Strike` (28) default team-server port. |
| `TCP 60000` | 23 | 3.8% | `Viper` (23) post-exploitation web panel port. |
| `TCP 8080` | 19 | 3.2% | `CyberStrikeAI` (6), `Cobalt Strike` (6), `ARTEX` (5), `PlugX` (2). |
| `TCP 8443` | 19 | 3.2% | `Cobalt Strike` (14), `Havoc` (2), `PentAGI` (1), others (2). |
| `TCP 4433` | 13 | 2.2% | `Cobalt Strike` (12), `CS Unverified` (1). |
| `TCP 7880` | 10 | 1.7% | `Acunetix` (10) vulnerability scanner port. |
| `TCP 8085` | 8 | 1.3% | `Yakit` (5), `CS Unverified` (2), `Cobalt Strike` (1). |
| `TCP 11111` | 8 | 1.3% | `RedGuard` (6), `Brute Ratel C4` (2). |
| `Other 108 TCP ports` | 232 | 38.8% | Includes `TCP 8787` (`ARTEX`: 7), `TCP 13000` (`AutoCVE`: 3), and 106 other ports. |
| **Total (120 ports)** | **598** | **100.0%** | **366 on top 12 ports (61.2%) + 232 on other 108 ports (38.8%).** |

Standard web ports (**:443**, **:80**, **:8080**, **:8443**) carry **173** endpoints (**28.9%**). Non-standard and high ports carry **425** endpoints (**71.1%**), showing why server VLANs should block direct outbound calls to uncategorized high TCP ports.

## 3. Multi-Service Multiplexing and the Rackspace Spot Overlap

Grouping `AS40065` by unique IPv4 host shows that **113 multi-service hosts (28.2% of 401 IPs) hold 310 of the 598 candidate endpoints (51.8%)**. Table 3 breaks all **401** hosts into four density bands.

| Host density band | Unique IPs | IP share (%) | Candidate endpoints | Endpoint share (%) | Same-tool multi-port IPs | Cross-tool IPs |
|---|---|---|---|---|---|---|
| 1 endpoint per IP | 288 | 71.8% | 288 | 48.2% | 0 | 0 |
| 2 endpoints per IP | 68 | 17.0% | 136 | 22.7% | 55 | 13 |
| 3 endpoints per IP | 24 | 6.0% | 72 | 12.0% | 22 | 2 |

| Host density band | Unique IPs | IP share (%) | Candidate endpoints | Endpoint share (%) | Same-tool multi-port IPs | Cross-tool IPs |
|---|---|---|---|---|---|---|
| `4 to 8` endpoints per IP `(peak 8 on 1 IP)` | 21 | 5.2% | 102 | 17.1% | 17 | 4 |
| Multi-endpoint subtotal (2-8) | 113 | 28.2% | 310 | 51.8% | 94 | 19 |
| Total | 401 | 100.0% | 598 | 100.0% | 94 | 19 |

Inside the **4-8** endpoint band (**21** hosts, **102** endpoints), **13** hosts ran **4** endpoints (**52**) and **2** hosts ran **5** endpoints (**10**). Three hosts ran **6** endpoints (**18**), **2** hosts ran **7** (**14**), and **1** peak host ran **8** endpoints (**8**).

Ninety-four multi-endpoint servers bound multiple ports of one tool, led by **72** `Cobalt Strike`, **6** `RedGuard`, **5** `Yakit`, and **5** `ARTEX` hosts. The other six same-tool hosts ran `Acunetix` (**2**), `CS Unverified` (**2**), `CyberStrikeAI` (**1**), and `PlugX` (**1**).

Nineteen servers paired two distinct tools, led by `Cobalt Strike` with `CS Unverified` (**4**) and `ARL` with `Cobalt Strike` (**3**). Other pairs linked `Acunetix` with `ARL` (**2**), `AsyncRAT` (**2**), or `CS Unverified` (**2**), plus six single-host tool pairs including `ARTEX` with `CyberStrikeAI` (**1**).

## `AS40065` Overlap and the November 2026 Rackspace Spot Disclosures

In late November 2026, Rackspace customer notices disclosed a security incident on **Rackspace Spot** (`[REF-01]`, `[REF-02]`), its managed cloud and Kubernetes service. Public reporting states that intruders signed in with stolen or valid credentials on `17 November 2026` and reached customer Kubernetes clusters. Inside affected clusters, intruders created unauthorized pods (`[REF-07]`), listed nodes and persistent volumes (`[REF-06]`), and may have read Kubernetes `Secret` objects and storage-volume files (`[REF-08]`).

Rackspace detected the activity on `18 November 2026` and turned off the Rackspace Spot management console (`[REF-01]`, `[REF-02]`). Console access returned on `21 November 2026` for customer migration (`[REF-01]`), and on `24 November 2026` Rackspace announced permanent Spot retirement set for `8 December 2026` (`[REF-02]`). Figure 4 summarizes the four `AS40065` host density bands across **113** multi-service cloud nodes.

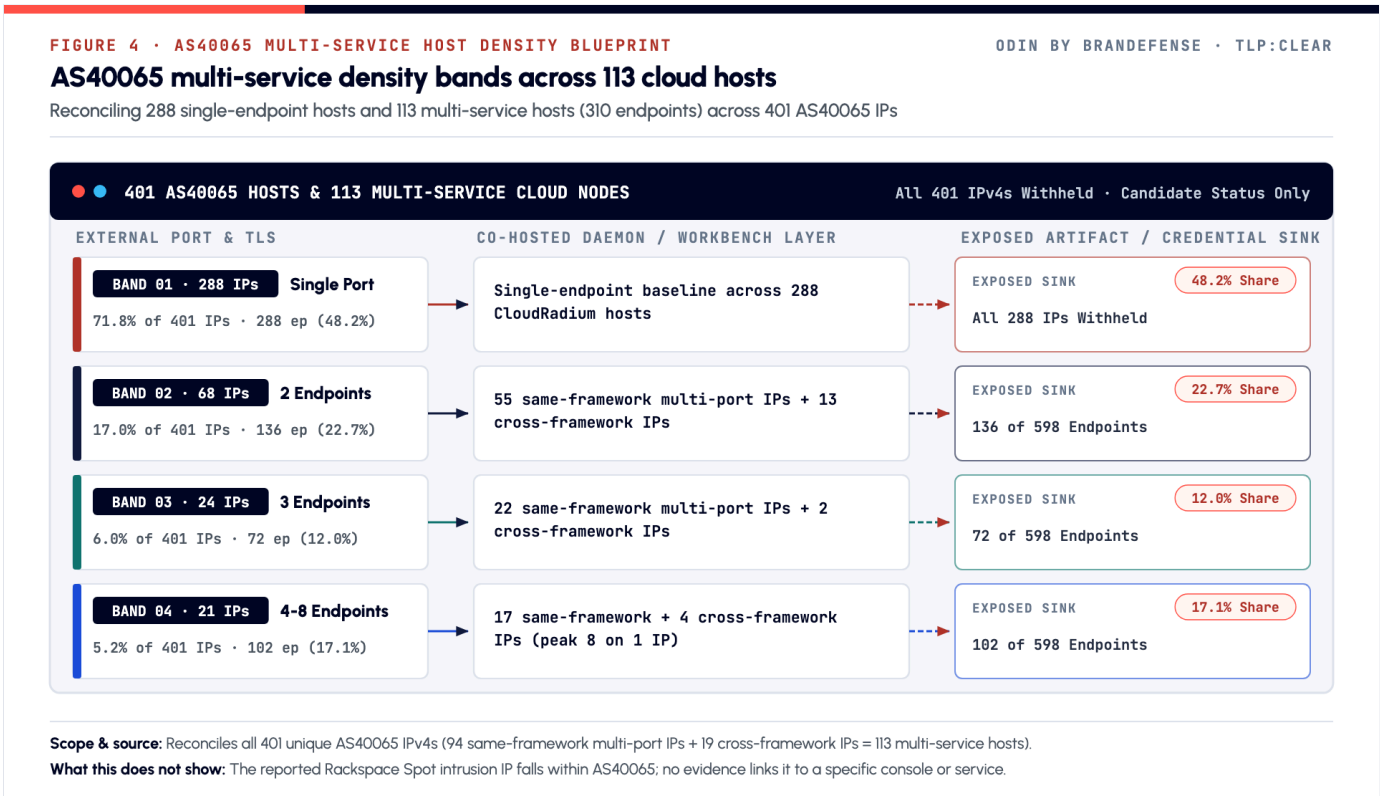


Figure 4: AS40065 multi-service density bands across 113 cloud hosts

The reported Rackspace Spot intrusion IP falls within AS40065; no evidence links it to a specific console or service. That United States host IP space is withheld because the candidate addresses lack independent adversary confirmation. Table 4 lines up our `12-25 November 2026` `AS40065` observation window with the disclosed Spot timeline.

| Date (`Nov-Dec 2026`) | Source | Observed `AS40065` window or disclosed Rackspace Spot event |
|---|---|---|
| `12-25 Nov 2026` | Odin observation | Two-week active window on `AS40065`: **32** active candidate endpoints across **18** IPs (**24** autonomous console ports) within **598** cumulative records. |
| `17 Nov 2026` | Public Spot notice | Credential-based entry into Rackspace Spot from an IP within `AS40065`. Unauthorized pods created; `Secrets` and storage-volume files may have been read (`[REF-01]`, `[REF-02]`). |
| `18 Nov 2026` | Public Spot notice | Rackspace detects unauthorized cluster activity and disables the Rackspace Spot management console (`[REF-01]`, `[REF-02]`). |
| `21 Nov 2026` | Public Spot notice | Spot console restored for credential rotation and customer migration; customer breach notice reported (`[REF-01]`). |
| `24 Nov 2026` | Public Spot notice | Permanent Rackspace Spot platform retirement announced for `8 December 2026` (`[REF-02]`). |

Three strict boundaries guide how analysts should read the `AS40065` overlap:

- **ASN overlap does not link the IP to a specific console or service.** The reported Rackspace Spot intrusion IP falls within `AS40065`, where **32** candidate endpoints (**24** autonomous consoles) were active from `12 to 25 November`. However, no evidence links it to a specific console or service, and external scans hold no cluster API payloads.
- **Zero attribution to tool builders or named threat groups.** Public `ARTEX` forks appear across **22** `AS40065` endpoints (**8** IPs), alongside **113** multi-service hosts running diverse scanners, relays, and C2 listeners. Odin by Brandefense makes **zero attribution** to `ARTEX` authors or any named actor.
- **Withholding candidate IPs.** Because the candidate addresses lack independent adversary confirmation, all **401** IPv4s stay withheld.

## 4. Managed Kubernetes Secret and Pod Hardening Architecture

When an intruder signs in with stolen cloud portal credentials or a leaked `kubeconfig` file, cluster actions follow a clear path (`[REF-03]`, `[REF-06]`). Figure 3 shows an analytical model pairing `AS40065` multi-service hosting context (`Step 01`) with the Kubernetes actions reported in Rackspace Spot customer notices (`Steps 02-06`).

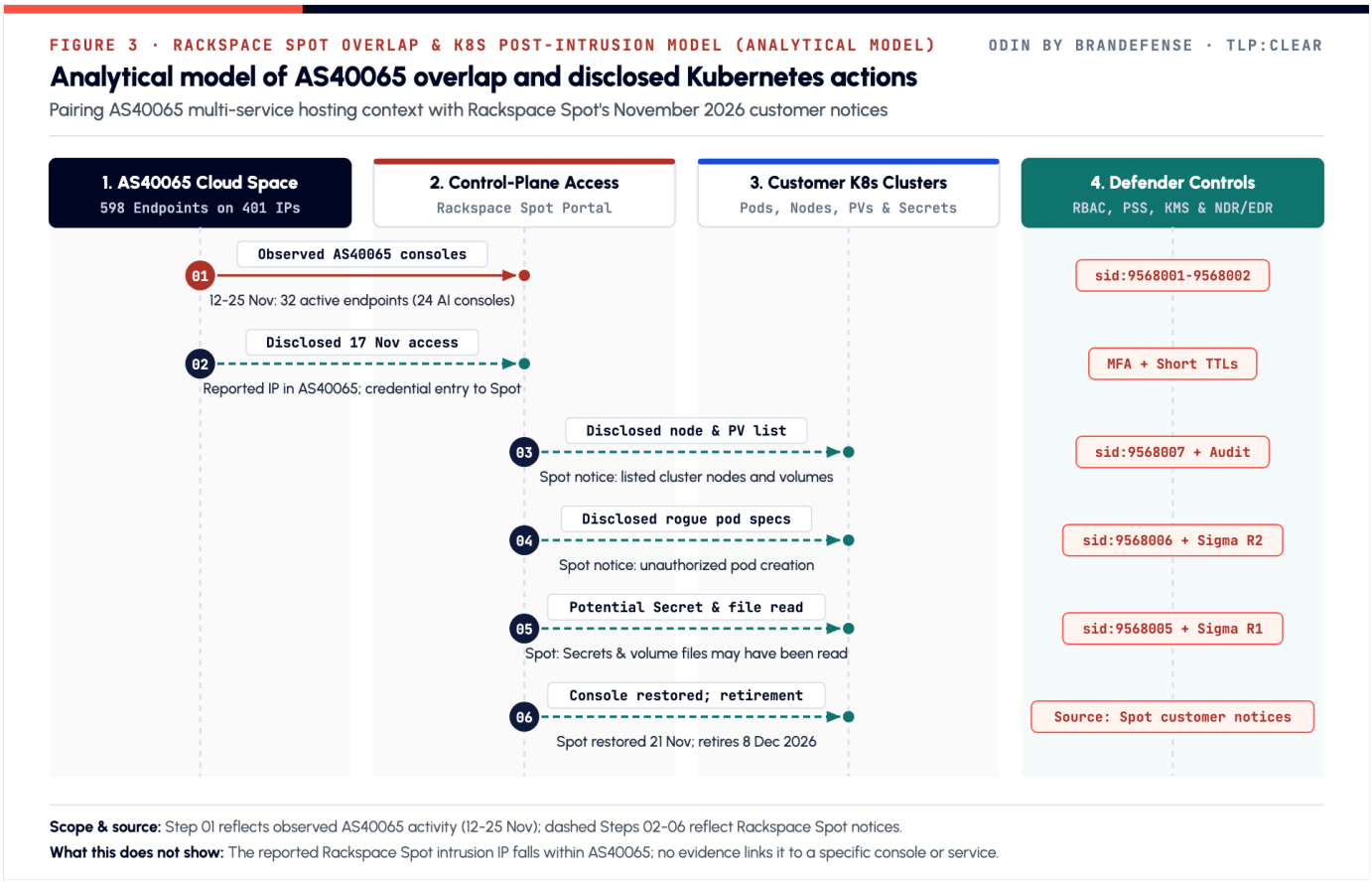


Figure 3: Analytical model of AS40065 overlap and disclosed Kubernetes actions

## How Credential Access Reaches Cluster Pods, Volumes, and Secrets

The Spot notices confirm unauthorized pod creation and node and volume listing. They say Secret and storage-volume files may also have been read.

- **Potential cluster-wide `Secret` access ( `T1552.007` , `[REF-03]` , `[REF-08]` ).** Any role holding cluster-wide `get` , `list` , or `watch` verbs on `secrets` ( `GET /api/v1/secrets` ) can export database logins, TLS keys, and `.dockerconfigjson` tokens through the API ( `[REF-05]` ). KMS envelope encryption protects stored `Secrets` at rest in `etcd` , not authorized API responses ( `[REF-03]` ).
- **Rogue pod creation and volume mounts ( `T1610` , `[REF-04]` , `[REF-07]` ).** Calling `POST /api/v1/namespaces/*/pods` schedules new containers onto worker nodes. `Restricted` Pod Security blocks `privileged` and `hostPath` pods, but still permits `persistentVolumeClaim` mounts and does not stop pod creation or `ServiceAccount` token automounting ( `[REF-04]` ). Limit pod creation and volume use through RBAC and admission policy ( `[REF-05]` ), and disable token automounting explicitly ( `[REF-06]` ).
- **Node and volume listing ( `[REF-06]` ).** Querying `/api/v1/nodes` and `/api/v1/persistentvolumes` maps worker hosts, CSI storage paths, and backing cloud volumes across zones.

Figure 2 links all five console and cluster layers to the defender actions in Section 7 and the rules in Section 8.

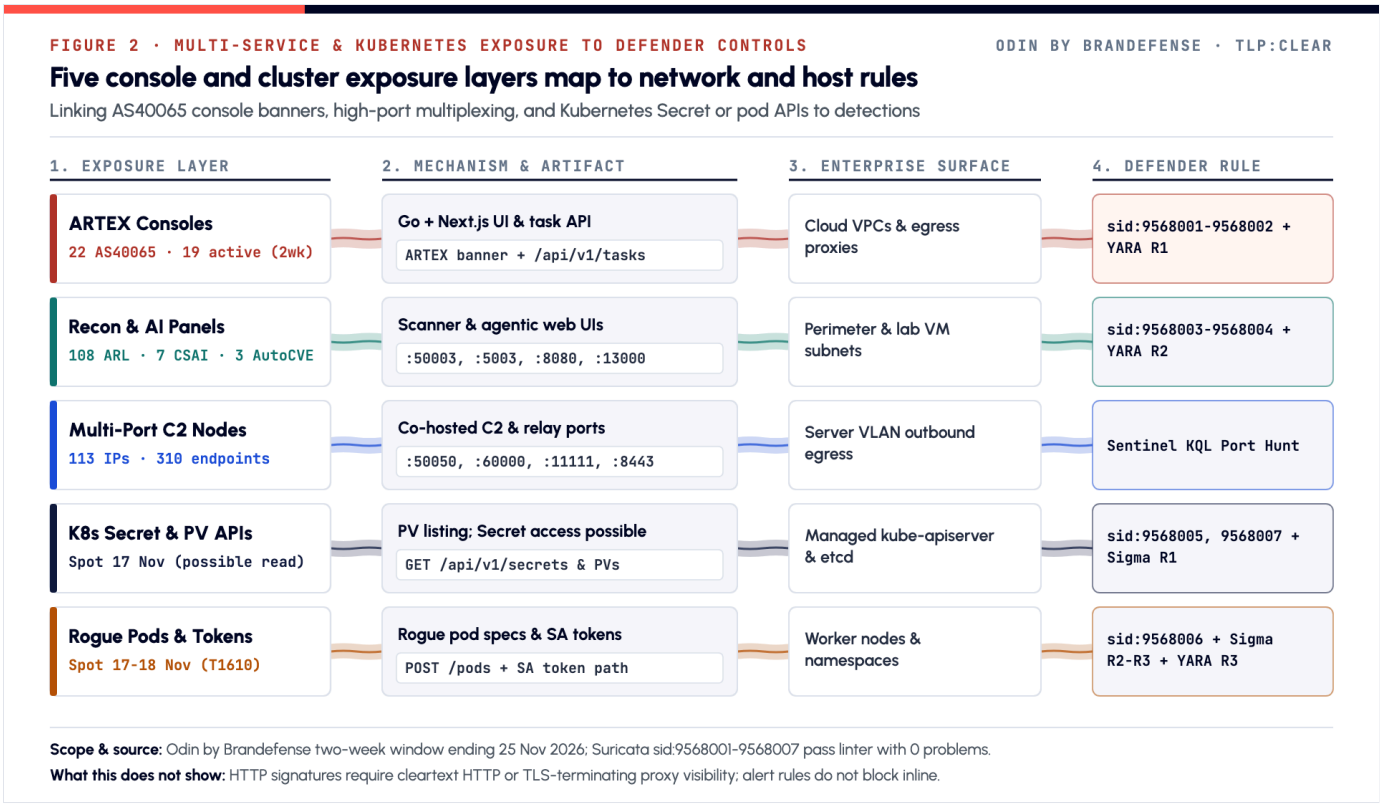


Figure 2: Five console and cluster exposure layers map to network and host rules

## 5. Enterprise Exposure and Control Verification

Teams should check five concrete controls across cloud subnets and Kubernetes clusters rather than assuming unverified root causes:

| Exposure layer | Observed or disclosed evidence | Control to verify | Operational risk |
|---|---|---|---|
| Exposed scanner & AI panels | 598 `AS40065` candidates (108 `ARL`, 22 `ARTEX`, 21 `Acunetix`, 13 `Yakit`, 7 `CyberStrikeAI`). | Verify whether cloud VPCs expose :8787, :50003, :5003, :7880, :8080, or :443 panels without mTLS. | Open panels leak scan scopes, API keys, and to routes ( `[REF-10]` ). |
| Multi-port high-port egress | 113 multi-service IPs (310 endpoints); 71.1% (425/598) sit outside :443/:80/:8080/:8443. | Verify whether worker nodes and server VLANs block direct outbound calls to random high ports. | Pods or hosts can reach co-hosted C2, `RedGuar` or scanner ports. |
| Kubernetes `Secret` RBAC & KMS | Spot customer notices: intruders may have read Kubernetes `Secrets` in customer clusters ( `[REF-01]`, `[REF-02]`, `[REF-08]` ). | Verify whether RBAC blocks cluster-wide `get`, `list`, and `watch` on `secrets` ( `[REF-05]` ) and `etcd` uses KMS v2 encryption at rest ( `[REF-03]` ). | Broad `Secret` RBAC exposes credentials via the API, while unencrypted `etcd` snapshots expose store Secrets at rest. |
| Rogue pod & volume admission | Spot customer notices: unauthorized pods created and volume files may have been read ( `[REF-01]`, `[REF-02]`, `[REF-07]` ). | Verify whether namespaces enforce `Restricted` Pod Security ( `[REF-04]` ) and whether RBAC and admission policies restrict pod creation and `persistentVolumeClaim` mounts ( `[REF-05]` ). | `Restricted` Pod Secur blocks `privileged` an `hostPath` pods, but sti permits `persistentVolumeCla` mounts if a role can create pods. |
| Pod `ServiceAccount` tokens | Default pods mount JWTs at `/var/run/secrets/kubernetes.io/serviceaccount/token` ( `[REF-06]`, `[REF-09]` ). | Verify whether `automountServiceAccountToken: false` is set on pods that do not call `kube-apiserver`. | Container shell bugs grant instant authenticated cluster A queries. |

## 6. Alternative Hypotheses

| Hypothesis | Evidence for | Evidence against | Assessment |
|---|---|---|---|
| **`ARTEX` authors or a named APT ran the Spot breach** | The reported Spot intrusion IP falls within `AS40065`, where 22 candidate `ARTEX` endpoints (19 active in `12-25 Nov`) operate. | `ARTEX` is open-source, `AS40065` hosts 598 endpoints across 401 commercial VPS IPs (`[REF-10]`), and no evidence links the intrusion IP to a specific console or service. | **Unsupported.** Zero attribution to tool authors or named groups. |
| **An `ARTEX` console on `AS40065` issued the `17 Nov` K8s API calls** | Autonomous consoles account for 24 of 32 active `AS40065` endpoints (`12-25 Nov`), including 19 `ARTEX` ports. | No evidence links the reported intrusion IP to a specific console or service, and external scans hold no API payloads. | **Unproven mechanism.** ASN overlap cannot identify which service sent the cluster API traffic. |
| **Blocking all 401 `AS40065` IPs solves the risk** | 113 multi-service hosts hold 310 candidate endpoints across C2 and scanner tools. | All 598 records hold candidate status without beacons, and shared cloud IPs churn fast. | **Rejected for static blocking.** Withhold all 401 IPs and harden K8s RBAC and PSS. |
| **`AS40065` mostly runs active `Cobalt Strike` today** | `Cobalt Strike` and unverified profiles form 49.2% (294/598) of cumulative records. | Only 1 of 294 `Cobalt Strike` ports was active in `12-25 Nov`, versus 24 autonomous console ports. | **Refuted by date slicing.** Splitting 32 active ports from 566 older records avoids bias. |

## 7. What Defenders Should Do

Teams migrating off retiring platforms (Rackspace Spot retires `8 December 2026`, `[REF-02]`) or running Kubernetes clusters can close post-credential gaps in five steps:

1. **Rotate Spot secrets and tighten `Secret` RBAC (`0-48 Hours`).** Treat every `Secret`, registry token, TLS key, and volume credential from November 2026 Spot clusters as exposed and rotate them (`[REF-01]`, `[REF-02]`, `[REF-03]`). Remove cluster-wide `get`, `list`, and `watch` verbs on `secrets` from human and CI roles, scoping reads to single named objects (`[REF-05]`).
2. **Enforce `Restricted` Pod Security, RBAC, and volume admission (`0-72 Hours`).** Label production namespaces with `pod-security.kubernetes.io/enforce: restricted` to reject `privileged` pods, `hostNetwork`, and `hostPath` mounts (`[REF-04]`, `[REF-07]`). Because `Restricted` Pod Security still permits `persistentVolumeClaim` mounts, limit pod creation and volume claims through RBAC and admission policy (`[REF-05]`, Sigma Rule 2).
3. **Turn on KMS v2 `etcd` encryption and disable token automounts (`Days 1-7`).** Configure KMS v2 envelope encryption so `Secrets` stay encrypted at rest in `etcd` backups (`[REF-03]`). Set `automountServiceAccountToken: false` on default `ServiceAccounts` and app pods (`[REF-06]`, `[REF-09]`), using short-lived projected tokens where needed (Sigma Rule 3).
4. **Lock down `kube-apiserver` access and storage volumes (`Days 7-14`).** Put `kube-apiserver` behind private VPC links or mTLS proxies and require phishing-resistant MFA on cloud portals (`[REF-06]`). Restrict CSI volume binds and audit `GET /api/v1/persistentvolumes` calls (`[REF-06]`, `sid:9568007`).
5. **Sweep cloud ports and block high-port worker egress (`Days 14-21`).** Scan cloud ranges on web ports and high ports (**8787**, **50003**, **5003**, **7880**, **13000**) for open `ARTEX`, `ARL`, `CyberStrikeAI`, and `AutoCVE` panels (`sid:9568001`-**9568004**, YARA Rules 1-2). Block worker VLANs from direct calls to high ports (**50003**, **5003**, **50050**, **60000**, **11111**, **8787**, **13000**) and hunt with Sentinel KQL.

## 8. Detection Rules

The full rules are in the Detection rules tab. Because all **401** candidate IPv4s are withheld, every rule targets HTTP banners, Kubernetes API paths, CLI flags, or manifest strings.

Verification scope: the Suricata file passed the Odin Suricata linter (**0** problems). YAML checks confirmed all three Sigma rules, and structural checks verified the Sentinel KQL and YARA files. Live engine binaries (`suricata -T`, `Kusto.Language`, `pySigma`, and `yara`) were not installed on the build host. Rules were not replayed on live cluster traffic, so test and tune them before rollout.

| Rule file | What it catches |
| --- | --- |
| `odin_as40065_consoles_and_k8s_api_hunt.rules` | Suricata 7.x ( `sid:9568001` -9568007). Flags `ARTEX` , `ARL` , and `CyberStrikeAI` banners, `POST /api/v1/tasks` , and inbound K8s `Secret` , `Pod` , and `PV` API calls. |
| `sentinel_k8s_secret_pod_and_multiservice_hunt.kql` | Sentinel KQL (7-day lookback). Links `kubectl` `Secret` exports, imperative pod calls, and SA token reads with high-port console egress. |
| `sigma_k8s_secret_read_pod_deploy_and_token_access.yml` | Portable Sigma (3 Linux rules). Flags `kubectl` bulk `Secret` exports, imperative pod creation or overrides, and CLI reads of mounted SA tokens. |
| `odin_as40065_consoles_and_k8s_artifacts.yar` | Structural YARA (4 `filesize` -bounded rules). Flags `ARTEX` and `ARL` console bundles, privileged `hostPath` `Pod` specs, and `SecretList` dumps. |

## How the Suricata 7.x Signatures Work ( `sid:9568001` -9568007)

All seven `http` signatures require cleartext HTTP or placement behind a TLS-terminating reverse or API proxy (required to inspect `kube-apiserver` HTTPS requests). Because `alert` rules do not drop traffic inline, enforce blocks in Kubernetes RBAC, Pod Security Admission, or your API gateway.

| SID | Direction & flow | What it matches | Defensive purpose |
| --- | --- | --- | --- |
| 9568001 | `$HOME_NET any -> $EXTERNAL_NET any` ( `to_client` ) | `http.response_body` containing `ARTEX - Autonomous Red-Team Expert` | Flags exposed home-network `ARTEX` panels answering external clients |
| 9568002 | `$HOME_NET any -> $EXTERNAL_NET any` ( `to_server` ) | `POST` request with `http.uri` starting with `/api/v1/tasks` | Flags outbound task dispatch calls to an `ARTEX` controller |
| 9568003 | `$HOME_NET any -> $EXTERNAL_NET any` ( `to_client` ) | `http.response_body` containing `Asset Reconnaissance Lighthouse` | Flags exposed `ARL` recon scanner panels (108 on `AS40065` ) |
| 9568004 | `$HOME_NET any -> $EXTERNAL_NET any` ( `to_client` ) | `http.response_body` containing `CyberStrikeAI` | Flags exposed `CyberStrikeAI` panels (7 on `AS40065` ) |
| 9568005 | `$EXTERNAL_NET any -> $HOME_NET any` ( `to_server` ) | Inbound `GET` with `http.uri` starting with `/api/v1/secrets` | Flags cluster-wide `Secret` list queries via TLS proxy ( `T1552.007` ) |
| 9568006 | `$EXTERNAL_NET any -> $HOME_NET any` ( `to_server` ) | Inbound `POST` with `http.uri` starting `/api/v1/namespaces/` and containing `/pods` | Flags pod-route POSTs; confirm the Kubernetes audit verb is `create` before classifying pod creation. |
| 9568007 | `$EXTERNAL_NET any -> $HOME_NET any` ( `to_server` ) | Inbound `GET` with `http.uri` starting with `/api/v1/persistentvolumes` | Flags cluster storage-volume discovery calls |

## How the Sentinel KQL, Sigma, and YARA Rules Work

The Sentinel KQL hunt unions two branches over a **7**-day lookback. Replace the sample hostnames in `ApprovedAdminHosts` ( `k8s-admin-01.corp.example` , `sec-scanner-01.corp.example` ) with your own admin bastions before scheduling.

The first branch checks `DeviceProcessEvents` for `kubectl` commands pairing `secrets` with `-A`, `--all-namespaces`, `-o yaml`, or `-o json`, or imperative pod flags (`create pod`, `--overrides=`, `run`). That same branch also flags `curl`, `wget`, `cat`, and `base64` reading `/var/run/secrets/kubernetes.io/serviceaccount/token`. The second branch checks `DeviceNetworkEvents` for `RemotePort in (MultiServicePorts)` (**8787**, **50003**, **5003**, **50050**, **60000**, **13000**, **11111**) or `RemoteUrl contains "/api/v1/tasks"`.

| Host or file rule | Engine | Trigger condition |
|---|---|---|
| `Linux Kubectl Bulk Secret Export or All-Namespaces Enumeration (R1)` | Sigma (`linux`) | `/kubectl` with `secrets` and `-A`, `--all-namespaces`, `-o yaml`, or `-o json` outside `k8s-admin-01.corp.example` (`T1552.007`) |
| `Linux Kubectl Imperative Pod Creation or Spec Override (R2)` | Sigma (`linux`) | `/kubectl` with `run`, `create pod`, or `--overrides=` outside `ci-deployer-01.corp.example` (`T1610`) |
| `Linux Container ServiceAccount Token Read via CLI Utilities (R3)` | Sigma (`linux`) | `/curl`, `/wget`, `/cat`, or `/base64` referencing `/var/run/secrets/kubernetes.io/serviceaccount/token` (`T1552.001`, `T1552.007`, `[REF-09]`) |
| `ODIN_R68_ARTEX_Console_Bundle_and_Task_API (YARA R1)` | YARA (< 5MB) | `ARTEX - Autonomous Red-Team Expert` paired with `/api/v1/tasks` or `artex-swarm-root` |
| `ODIN_R68_ARL_Recon_Lighthouse_Bundle (YARA R2)` | YARA (< 5MB) | `Asset Reconnaissance Lighthouse` paired with `/api/user/login` or `arl_web` |
| `ODIN_R68_Privileged_HostPath_Pod_Manifest (YARA R3)` | YARA (< 512KB) | `kind: Pod` paired with `hostPath` or `hostNetwork: true` and `privileged: true` |
| `ODIN_R68_Exported_K8s_Secret_List_Dump (YARA R4)` | YARA (< 10MB) | `SecretList` and `apiVersion: v1` paired with `kubernetes.io/service-account-token` or `dockerconfigjson` |

## 9. Indicators

The Indicators tab holds **0** published blocklist IPv4 addresses and **0** file SHA-256 hashes. All **401** candidate IPv4s on `AS40065` are withheld because the candidate addresses lack independent adversary confirmation.

| Indicator or marker group | Published in Indicators tab | Status in this report |
|---|---|---|
| **Candidate `AS40065` host IPv4s (401 IPs across 598 endpoints)** | 0 | Withheld (candidate addresses lack independent adversary confirmation) |
| **Rackspace Spot intrusion IPv4 (`AS40065` overlap)** | 0 | Withheld (no evidence links the IP to a specific console or service) |
| **File SHA-256 hashes** | 0 | None published (**0** malware payload files in scope) |
| **Behavioral, protocol, and Kubernetes hunt markers** | 0 (text-only table below) | 10 markers published in the table below |

## Behavioral, Protocol, and Kubernetes Hunt Markers

| Marker category | Pattern / protocol artifact | Layer | Defensive hunt & control |
|---|---|---|---|
| ARTEX icon MD5 | `cd3ebd16437d6d3963592f7365dd3980` | All web ports | Scan cloud subnets for `/favicon.ico` (15,406 B) on 8787, 443, 8080, and 80. |
| ARTEX HTML banner | `ARTEX - Autonomous Red-Team Expert` | HTTP / TLS proxy | Alert on exposed panels ( `sid:9568001` , YARA R1). Gate approved tools behind mTLS. |
| ARTEX task API | `POST /api/v1/tasks` | HTTP / TLS egress | Alert on outbound task calls ( `sid:9568002` ) and Sentinel `DeviceNetworkEvents` . |
| ARL scanner banner | `Asset Reconnaissance Lighthouse` | 50003, 5003, 443 | Sweep cloud VPCs for exposed `ARL` consoles ( `sid:9568003` , YARA R2; 108 on `AS40065` ). |
| CyberStrikeAI banner | `CyberStrikeAI` | 8080, 8445 | Sweep external assets and alert on exposed home responses ( `sid:9568004` ). |
| K8s Secret API route | `GET /api/v1/secrets` | K8s API / TLS proxy | Audit cluster-wide `Secret` list calls ( `sid:9568005` , K8s audit logs, YARA R4). |
| K8s Pod route POST | `POST /api/v1/namespaces/*/pods` | K8s API / Admission | Restrict pod creation and volume use via RBAC and admission policy ( `[REF-04]` , `[REF-05]` ); alert via `sid:9568006` and YARA R3. |
| K8s PV list route | `GET /api/v1/persistentvolumes` | K8s API / TLS proxy | Restrict storage-discovery RBAC verbs ( `[REF-05]` ) and alert via `sid:9568007` . |
| Bulk Secret CLI | `kubectl get secrets --all-namespaces` | Linux / admin EDR | Hunt `-A` , `-o yaml` , and `-o json` exports with Sentinel KQL and Sigma Rule 1. |
| Pod SA token path | `/var/run/secrets/kubernetes.io/serviceaccount/token` | Container / node EDR | Set `automountServiceAccountToken: false` and hunt CLI reads (Sigma Rule 3, KQL). |

## 10. Scope and Limits

Odin by Brandefense reviewed **598** candidate endpoints across **401** IPv4s on `AS40065` ( `CNSERVERS LLC` / `cloudradium[.]com` ) during a two-week observation window ending `25 November 2026` . Thirty-two endpoints on **18** IPs carried active dates from `12 to 25 November 2026` , including **22** newly seen on **8** IPs. Another **566** records across **387** IPs carried earlier upstream dates before `12 November 2026` , with **4** IPs appearing in both slices. Whereas Monograph 60 covered seven-framework growth and `Storm-0940` overlaps, Monograph 68 examines single-ASN multi-framework density on `AS40065` and managed Kubernetes post-compromise defense.

All **598** `AS40065` records hold candidate status without independent C2 beacon captures, so all **401** IPv4s are withheld. Public Rackspace Spot dates span the `17 Nov` intrusion, `18 Nov` shutdown, `21 Nov` restore, and `24 Nov` retirement notice for `8 Dec 2026` . Those milestones come from public reporting of customer notices ( `[REF-01]` , `[REF-02]` ).

The reported Rackspace Spot intrusion IP falls within `AS40065`; no evidence links it to a specific console or service. Shared ASN overlap does not identify which service sent the cluster API calls. Odin by Brandefense makes zero attribution to tool authors or named threat groups.

## 11. Sources


1. `[REF-01]` BleepingComputer, *Rackspace Spot breach let hackers access customer Kubernetes secrets* (21 November 2026; verified on 25 November 2026).
2. `[REF-02]` BleepingComputer, *Rackspace shuts down Spot cloud platform after security breach* (24 November 2026; verified on 25 November 2026).
3. `[REF-03]` Kubernetes Documentation, *Secrets Configuration and Encryption at Rest* (verified on 25 November 2026).
4. `[REF-04]` Kubernetes Documentation, *Pod Security Standards (Baseline and Restricted Profiles)* (verified on 25 November 2026).
5. `[REF-05]` Kubernetes Documentation, *Role-Based Access Control (RBAC) Good Practices (Restricting Secret and Pod Verbs)* (verified on 25 November 2026).
6. `[REF-06]` Kubernetes Documentation, *Securing a Cluster* (verified on 25 November 2026).
7. `[REF-07]` MITRE ATT&CK, *Deploy Container (T1610)* (verified on 25 November 2026).
8. `[REF-08]` MITRE ATT&CK, *Unsecured Credentials: Container API (T1552.007)* (verified on 25 November 2026).
9. `[REF-09]` MITRE ATT&CK, *Unsecured Credentials: Credentials In Files (T1552.001)* (verified on 25 November 2026).
10. `[REF-10]` MITRE ATT&CK, *Acquire Infrastructure: Virtual Private Server (T1583.003)* (verified on 25 November 2026).


---


BRANDEFENSE CAPABILITIES RELATED TO THIS RESEARCH

**Cyber Threat Intelligence**
Track the actors, infrastructure and campaigns behind threats like these.

**Threat Actor Landscape**
Profiles of the groups that run this kind of infrastructure.

**Threat Search**
Pivot on an indicator across the Brandefense dataset.

DEFENDER KIT

# Indicators of compromise

Verified adversary infrastructure and tool hashes from this study. Every value below is also discussed in the report.

### No network or file indicators are published for this study

The hosts in this report are exposed or unattributed systems rather than confirmed adversary infrastructure, so they stay anonymous. Use the detection rules below to find the same exposure in your own estate.

10 records from this study are not published in this table: candidate, suspected and dual-use hosts, victim-related values, and hashes of exposed secrets or files.

DEFENDER KIT

# Detection rules

The rule files exactly as printed in the report. Test them in your own engine before deployment.

**Suricata** `odin_as40065_consoles_and_k8s_api_hunt.rules` 11 lines

```
# Odin by Brandefense - Monograph 68 Suricata 7.x Ruleset (sid:9568001-9568007)
# Syntax-checked with Odin Suricata linter (0 problems); requires HTTP or TLS proxy visibility.

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-R68. Exposed ARTEX Banner"; flow:established,to_client; http.response_body;
content:"ARTEX - Autonomous Red-Team Expert"; fast_pattern; classtype:trojan-activity; sid:9568001; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-R68. ARTEX Task API"; flow:established,to_server; http.method; content:"POS
T"; bsize:4; http.uri; content:"/api/v1/tasks"; startswith; fast_pattern; classtype:trojan-activity; sid:9568002; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-R68. Exposed ARL Scanner"; flow:established,to_client; http.response_body;
content:"Asset Reconnaissance Lighthouse"; fast_pattern; classtype:trojan-activity; sid:9568003; rev:1;)
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN-R68. CyberStrikeAI Banner"; flow:established,to_client; http.response_body;
content:"CyberStrikeAI"; fast_pattern; classtype:trojan-activity; sid:9568004; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R68. K8s Secret List"; flow:established,to_server; http.method; content:"GE
T"; bsize:3; http.uri; content:"/api/v1/secrets"; startswith; fast_pattern; classtype:attempted-admin; sid:9568005; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R68. K8s Pod Route POST (verify audit verb)"; flow:established,to_server; h
ttp.method; content:"POST"; bsize:4; http.uri; content:"/api/v1/namespaces/"; startswith; content:"/pods"; fast_pattern; classtype:a
ttempted-admin; sid:9568006; rev:1;)
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN-R68. K8s PV Enum"; flow:established,to_server; http.method; content:"GET";
bsize:3; http.uri; content:"/api/v1/persistentvolumes"; startswith; fast_pattern; classtype:attempted-recon; sid:9568007; rev:1;)
```

Show full rule

**YARA** `odin_as40065_consoles_and_k8s_artifacts.yar` 71 lines

```
/*
   Odin by Brandefense - Monograph 68 YARA Suite
   4 filesize-bounded structural rules covering exposed ARTEX and ARL console bundles,
   privileged hostPath Kubernetes pod manifests, and exported Kubernetes SecretList dumps.
*/

rule ODIN_R68_ARTEX_Console_Bundle_and_Task_API
{
    meta:
        author = "Odin by Brandefense"
        monograph = "68"
        description = "Detects exposed ARTEX autonomous testing console bundles and task API definitions"
        date = "2026-11-25"
    strings:
        $a1 = "ARTEX - Autonomous Red-Team Expert" ascii
        $a2 = "/api/v1/tasks" ascii
        $a3 = "artex-swarm-root" ascii
    condition:
        filesize < 5MB and $a1 and ($a2 or $a3)
}

rule ODIN_R68_ARL_Recon_Lighthouse_Bundle
{
    meta:
        author = "Odin by Brandefense"
        monograph = "68"
        description = "Detects exposed Asset Reconnaissance Lighthouse (ARL) web console bundles and login routes"
        date = "2026-11-25"
    strings:
        $l1 = "Asset Reconnaissance Lighthouse" ascii
        $l2 = "/api/user/login" ascii
        $l3 = "arl_web" ascii
    condition:
        filesize < 5MB and $l1 and ($l2 or $l3)
}

rule ODIN_R68_Privileged_HostPath_Pod_Manifest
{
```

```
    meta:
        author = "Odin by Brandefense"
        monograph = "68"
        description = "Detects Kubernetes Pod YAML or JSON manifests combining hostPath or hostNetwork with privileged execution"
        date = "2026-11-25"
    strings:
        $k1 = "kind: Pod" ascii
        $k2 = "\"kind\":\"Pod\"" ascii
        $m1 = "hostPath" ascii
        $m2 = "hostNetwork: true" ascii
        $p1 = "privileged: true" ascii
        $p2 = "\"privileged\":true" ascii
    condition:
        filesize < 512KB and ($k1 or $k2) and ($m1 or $m2) and ($p1 or $p2)
}

rule ODIN_R68_Exported_K8s_Secret_List_Dump
{
    meta:
        author = "Odin by Brandefense"
        monograph = "68"
        description = "Detects exported Kubernetes SecretList YAML or JSON dumps containing ServiceAccount or registry tokens"
        date = "2026-11-25"
    strings:
        $s1 = "SecretList" ascii
        $v1 = "apiVersion: v1" ascii
        $v2 = "\"apiVersion\":\"v1\"" ascii
        $t1 = "kubernetes.io/service-account-token" ascii
        $t2 = "kubernetes.io/dockerconfigjson" ascii
    condition:
        filesize < 10MB and $s1 and ($v1 or $v2) and ($t1 or $t2)
}
```

Sentinel KQL `sentinel_k8s_secret_pod_and_multiservice_hunt.kql` 28 lines

```
// Odin by Brandefense - Monograph 68
// Hunts Kubernetes Secret export, imperative pod creation, token reads, and multi-service C2 port egress.
let Lookback = 7d;
let ApprovedAdminHosts = dynamic(["k8s-admin-01.corp.example", "sec-scanner-01.corp.example"]);
let MultiServicePorts = dynamic([8787, 50003, 5003, 50050, 60000, 13000, 11111]);
let K8sSecretPodAndTokenActivity =
    DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAdminHosts))
    | where (
        (FileName =~ "kubectl" and (
            (ProcessCommandLine has "secrets" and ProcessCommandLine has_any ("-A", "--all-namespaces", "-o yaml", "-o json"))
            or ProcessCommandLine has_any ("create pod", "--overrides=")
            or ProcessCommandLine contains " run "
        ))
        or (ProcessCommandLine contains "/var/run/secrets/kubernetes.io/serviceaccount/token" and FileName in~ ("curl", "wget", "ca
t", "base64"))
    )
    | project Timestamp, DeviceName, AccountName, SignalType = "K8sSecretOrPodCLI", Detail = ProcessCommandLine;
let MultiServiceConsoleEgress =
    DeviceNetworkEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedAdminHosts))
    | where RemotePort in (MultiServicePorts) or RemoteUrl contains "/api/v1/tasks"
    | project Timestamp, DeviceName, AccountName = InitiatingProcessAccountName, SignalType = "MultiServiceConsoleEgress", Detail =
strcat(InitiatingProcessFileName, " -> ", RemoteIP, ":", tostring(RemotePort));
union K8sSecretPodAndTokenActivity, MultiServiceConsoleEgress
| summarize FirstSeen = min(Timestamp), LastSeen = max(Timestamp), EventCount = count(), Signals = make_set(SignalType, 5), SampleDe
tails = make_set(Detail, 5) by DeviceName, AccountName
| order by LastSeen desc
```

Sigma `sigma_k8s_secret_read_pod_deploy_and_token_access.yml` 88 lines

```
title: Linux Kubectl Bulk Secret Export or All-Namespaces Enumeration
id: 68a4d912-8b31-4e72-9c10-68b20f84d681
status: experimental
description: Detects kubectl commands enumerating Kubernetes Secrets across all namespaces or exporting Secret manifests to YAML or
JSON.
author: Odin by Brandefense
date: 2026-11-25
logsource:
  category: process_creation
  product: linux
detection:
  selection_bin:
    Image|endswith: '/kubectl'
    CommandLine|contains: 'secrets'
  selection_bulk:
    CommandLine|contains:
      - ' -A'
      - '--all-namespaces'
      - '-o yaml'
      - '-o json'
  filter_approved_hosts:
    ComputerName:
      - 'k8s-admin-01.corp.example'
  condition: selection_bin and selection_bulk and not filter_approved_hosts
falsepositives:
  - Approved cluster backup jobs or emergency incident-response bastions listed in ComputerName
level: high
tags:
  - attack.credential-access
  - attack.t1552.007
---
title: Linux Kubectl Imperative Pod Creation or Spec Override
id: 68a4d912-8b31-4e72-9c10-68b20f84d682
status: experimental
description: Detects imperative kubectl pod creation or inline JSON spec overrides bypassing declarative GitOps deployment pipeline
s.
author: Odin by Brandefense
date: 2026-11-25
logsource:
  category: process_creation
  product: linux
detection:
  selection_kubectl:
    Image|endswith: '/kubectl'
  selection_pod_action:
    CommandLine|contains:
      - ' run '
      - 'create pod'
      - '--overrides='
  filter_ci_runners:
    ComputerName:
      - 'ci-deployer-01.corp.example'
  condition: selection_kubectl and selection_pod_action and not filter_ci_runners
falsepositives:
  - Authorized break-glass troubleshooting sessions on designated bastion hosts
level: high
tags:
  - attack.execution
  - attack.t1610
---
title: Linux Container ServiceAccount Token Read via CLI Utilities
id: 68a4d912-8b31-4e72-9c10-68b20f84d683
status: experimental
description: Detects curl, wget, cat, or base64 reading the mounted Kubernetes pod ServiceAccount bearer token file directly.
author: Odin by Brandefense
date: 2026-11-25
logsource:
  category: process_creation
  product: linux
detection:
```

```
  selection_token_path:
    CommandLine|contains: '/var/run/secrets/kubernetes.io/serviceaccount/token'
  selection_reader:
    Image|endswith:
      - '/curl'
      - '/wget'
      - '/cat'
      - '/base64'
  filter_approved_hosts:
    ComputerName:
      - 'k8s-admin-01.corp.example'
  condition: selection_token_path and selection_reader and not filter_approved_hosts
falsepositives:
  - Minimal init containers using curl to query the Kubernetes API where client libraries are absent
level: high
tags:
  - attack.credential-access
  - attack.t1552.001
  - attack.t1552.007
```

