BGPKIT INCIDENT REPORT • IR-2026-0831 • PUBLIC DISTRIBUTION

The Virtualizor / Softaculous Supply-Chain BGP Hijack

Independent analysis of the 2026-08-28 – 08-30 routing incident from public BGP measurement data

2026-09-01 • Author: Mingwei Zhang <mingwei@bgpkit.com>, BGPKIT

All routing facts in this report were re-verified against raw MRT archives from RIPE RIS and Route Views; vendor-reported quantities are attributed as such and cross-checked where possible.

1. Executive summary

What happened. Between 2026-08-28 20:57:30 UTC and 2026-08-30 ∼06:10 UTC, the network AS62390 (NexonHost) fraudulently announced `162.55.80.0/24` – a block of Hetzner (AS24940) address space hosting Softaculous' software-update and billing infrastructure – through transit provider AS6204 (ZET.NET). Because the forged route was *more specific* than Hetzner's legitimate `162.55.0.0/16`, it won route selection across every network that received it. The attacker kept the real origin (AS24940) at the end of the AS path, which made the hijacked route **validate as RPKI-Valid**: origin validation by any network would not have rejected it anywhere. Over the diversion window, the attacker passed Let's Encrypt's automated domain validation, obtained a technically valid TLS certificate covering 26 Softaculous product domains, impersonated the update infrastructure, and delivered a backdoored Virtualizor update that compromised hypervisor servers at root level.

Key numbers. Incident window: ≈33 hours across two attack waves, of which traffic was actively divertible for ≈22 hours. Peak diversion reached ≈72% of RIPE RIS's 368 collector peers (vendor measurement via RIPEstat; BGPKIT cross-checked onset, countermeasure, and both wave boundaries). The first effective countermeasure, Hetzner announcing the /24 directly, took effect ≈12 hours after onset and only after escalation by the victim. The fraudulently issued certificate remained valid until revoked: a worst case of ≈31 hours.

Why the defenses folded, in order:

- **RPKI/ROV: bypassed by construction.** The ROA for `162.55.0.0/16` authorizes origin AS24940 with `maxLength 24`; the hijacked /24 announced exactly that origin at exactly that length. Full ROV deployment would not have blocked this route.
- **IRR prefix filtering at the hijacker's transit: absent.** AS62390's authorized prefix set has no plausible member that is a Hetzner /24; strict customer filtering at AS6204 would have stopped the route from propagating beyond one hop. This decades-old control was the one that mattered.
- **TLS: subordinated.** The CA's domain-control validation ran over the hijacked path, so the attacker obtained a certificate no browser or update client would question.
- **Update-channel integrity: absent.** Virtualizor's update client did not verify package signatures, so a valid-looking HTTPS response sufficed to deliver the payload.

Recommendations (decision-maker view). Treat RPKI ROV as necessary but insufficient; the residual exposure is precisely "who may be your upstream," which ASPA (draft-ietf-sidrops-aspa-verification) addresses and IRR-based customer filtering mitigates today. Any software distribution channel that relies only on DNS+TLS is hostage to routing; cryptographic package signing is the control that severs the routing-to-supply-chain chain. Continuous monitoring for *new more-specifics inside RPKI-valid blocks with an unseen origin-adjacent AS* flags this exact attack class within one minute of onset (Section 6).

1 Scope, assets, and evidence basis

sffamily Dimension	Value
Prefix attacked	`162.55.80.0/24` (IPv4), normally covered only by `162.55.0.0/16`
Legitimate origin	AS24940 Hetzner Online GmbH
Forged-route announcer	AS62390 NexonHost (announced via AS6204 ZET.NET transit)

Assets on the /24	Softaculous update endpoints (`files.{virtualizor,softaculous}.com` et al.), client area / billing (`softaculous.com/clients`); `softaculous.com` resolved to 162.55.80.8 during the incident
Observation window	2026-08-28T20:50Z → 2026-08-30T06:15Z (baseline begins 20:50)
Measurement plane	BGP control plane only. *This report makes no data-plane traffic or forwarding claims beyond what routed-diversion makes possible.*
Primary panel	RIPE RIS rrc00, rrc01; cross-project corroboration Route Views route-views2
Vendor measurements cited	RIPE RIS 368-peer set via RIPEstat `bgp-state`/`bgp-updates`/BGPlay (disclosure's methodology), re-verified by BGPKIT at key instants (Section 3)

Verification method. BGPKIT independently downloaded the raw MRT UPDATE files covering onset, Hetzner's countermeasure, second-wave onset, and incident end from both collector projects, parsed them with `bgpkit-parser` (Rust) and `monocle`, and re-derived every timestamp and AS path quoted below. Vendor-claimed quantities that BGPKIT did not independently recompute (notably the full 10-minute visibility curve and total withdrawal count) are attributed to the disclosure throughout.

2 The hijack technique, read from the raw bytes

The first collector sighting of the forged route is a single MRT record at RIPE RIS rrc00, timestamped **2026-08-28 20:57:30 UTC** (unix 1787950650):

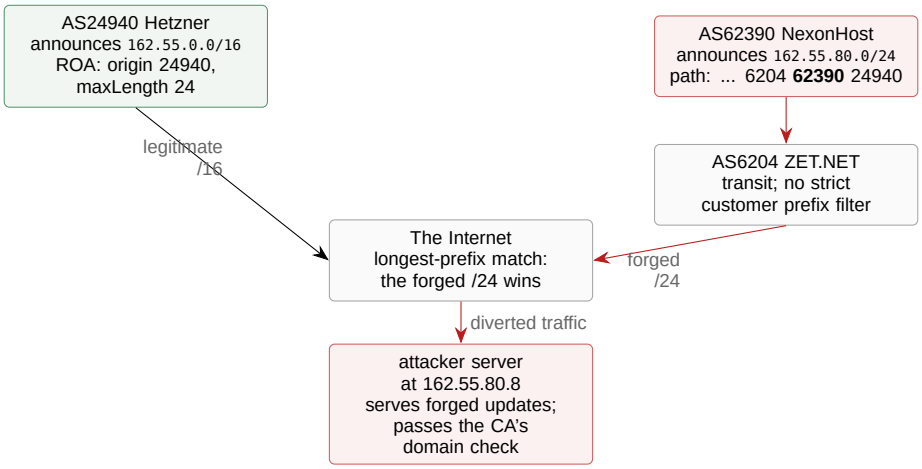
```
BGP4MP|1787950650|A|185.193.84.191|29504|162.55.80.0/24|29504 15935 6204 62390 24940|IGP|...
```

The record (144 bytes, complete with MRT and BGP4MP envelope) is preserved verbatim and can be dissected field-by-field in any browser:

wirescope.labs.bgpkit.com, hijack message dissection

Three elements of this message carry the entire attack:

- **NLRI** `18 a2 37 50` – prefix length 24, address 162.55.80.0. Longest-prefix match guarantees the forged route beats the legitimate /16 everywhere it propagates, independent of path length, policy, or relationship.
- **AS_PATH** `29504 15935 6204 62390 24940` – the attacker inserted *one* ASN (62390) immediately before the real origin (24940). The real origin is retained, so the announcement looks like ordinary customer space being transited. This is the "forged-origin" path pattern that RPKI origin validation is expressly unable to see.
- **Communities** `6204:1000 6204:1002` (stamped by the hijacker's own transit), `15935:100 15935:103`, `31230:0 31230:2` picked up en route. Communities accumulate hop-by-hop and later let analysts reconstruct the propagation paths the route took through different transit chains.



3 Verified timeline

Every "BGPKIT-verified" row was re-derived from the raw RIS/Route Views update files (public archives, 2026.08 directories; file names cited per row); vendor rows are from the disclosure's RIPEstat reconstruction.

Time (UTC)	Event	Source	Verification
08-28 20:50–20:55	Negative control: zero `162.55.80.0/24` records at rrc00 – the /24 had never been announced	rrc00 2050 file	BGPKIT-verified
08-28 20:57:30	**Onset.** First forged announce at rrc00: `29504 15935 6204 62390 24940`; same second at rrc01 via a different transit chain `6830 6204 62390 24940`	rrc00/rrc01 2055 files	BGPKIT-verified
08-28 21:01:04	Cross-project corroboration: Route Views route-views2, peer AS57866, `57866 5511 6204 62390 24940`	route-views2 2100 file	BGPKIT-verified
08-28 21:00–08-29 08:50	**Wave 1** (≈12 h). Route flaps violently; transit flap-dampening repeatedly suppresses it; peak diversion ≈72% of routed RIS peers (median 266, range 145–272 of 368)	disclosure/RIPEstat	vendor, onset cross-checked
08-29 08:00–08:50	Active interception independently confirmed: impostor host at the diverted address answers for Softaculous domains with the fraudulently issued certificate	disclosure + LET operators	external report
08-29 08:50	**Countermeasure.** Hetzner begins announcing the /24 directly (e.g. `852 3320 3320 3320 24940`); diversion drops to zero within minutes	rrc00 0850 file	BGPKIT-verified
08-29 09:00–14:30	Quiet: clean /24 held by Hetzner, then withdrawn; network falls back to the legitimate /16	disclosure table	vendor
08-29 19:55:25	**Wave 2 onset.** Forged route re-announced, same path signature, minutes after the clean /24 disappeared: `50300 3257 6204 62390 24940`	rrc00 1955 file	BGPKIT-verified
08-30 05:45:16	**End.** Last forged-route record at rrc00 (`202297 47232 3257 6204 62390 24940`); zero 62390 sightings after 05:50; clean by 06:10	rrc00 0545/0550 files	BGPKIT-verified

Effective-interception window (the quantity that matters for impact):

```
08-28 20:57  forged /24 first propagates --------------------------------
             wave 1 (~12 h): flapping, dampening, peak 266/368 RIS peers
08-29 08:50  Hetzner clean /24 up  ->  diversion ~0      [12 h after onset]
08-29 14:30  countermeasure withdrawn; net falls back to the /16
08-29 19:55  wave 2 (~10 h): same forged path returns
08-30 05:45  last forged record; clean by 06:10
             ------------------------------------------------------------
diverted:      ~22 of 33 hours     effective-defense lag: ~12 h
certificate:   valid until revoked; no revocation time disclosed ->
               worst-case usable ~31 h from onset (22 h with routing)
```

The second wave is operationally significant: the attacker watched Hetzner's countermeasure appear and *be withdrawn*, then re-announced into the gap. The routing data alone demonstrates sustained, adaptive operation rather than a one-shot misconfiguration.

Propagation corroboration census (onset files)

BGPKIT census of distinct collector peers already carrying a 62390 path in the first files covering onset: rrc00 **36 peers**, rrc01 **20 peers**, route-views2 **4 peers** (RV file begins 21:00, 3.5 min after onset). Distinct transit chains observed carrying the forged route: via AS6204 reached through `15935`, `3257`, `5511`, `6830`, `6939`, `3320` – i.e. the route entered multiple major transit backbones within seconds.

4 Why each defense layer failed

4.1 RPKI origin validation: bypassed by construction

Hetzner's ROA (verified continuously in force through the incident window via BGPKIT's ROA history service, which retains daily snapshots back to 2011):

```
$ curl "https://api.bgpkit.com/v3/roas/search?prefix=162.55.0.0/16"
{"prefix":"162.55.0.0/16","max_len":24,"asn":24940,
 "date_ranges":[["2021-03-17","2025-05-04"],["2025-05-06","2026-08-31"]],"current":true}
```

ROV decision for the forged route, exactly as a validating router computes it:

Check	Forged-route value	Result
Covering ROA exists	162.55.0.0/16 (AS24940, maxlen 24)	match
Origin ASN authorized?	announced origin = **24940** (the real one, kept)	authorized
Prefix length $\leq$ maxlen?	**24** $\leq$ 24	within
RPKI validation outcome		**VALID – route accepted everywhere**

This is not an implementation gap that wider ROV deployment closes; it is a *scope* gap. ROV authorizes origins, not paths. The attack inserts an unauthorized *upstream* (AS62390 before AS24940), which origin validation cannot see. The controls that do see it:

- **IRR-based customer prefix filtering at transit** (available today): AS62390's authorized set cannot plausibly contain a Hetzner /24; strict filtering at AS6204 confines the forgery to one AS. This was the single control that would have stopped propagation.
- **ASPA** (RPKI-adjacent, `draft-ietf-sidrops-aspa-verification`): path verification would reject the forged path as soon as the victim publishes one object – see the replay in Section 4.2. The retained-origin trick does not help against ASPA.

4.2 How ASPA would have changed the outcome

ROV authorizes *origins*; it says nothing about *who may sit immediately upstream of the origin* – which is exactly the degree of freedom this attack used. ASPA (`draft-ietf-sidrops-aspa-verification`) closes that gap: a customer publishes a signed list of its providers, and validators run an up/down algorithm over the AS path. We replayed the incident's paths through the draft's algorithm (§5.2–5.6) under three ASPA states, using the real global ASPA corpus (rpki-client export, 2026-09-01 build: **2,820 ASPA objects worldwide; none published by any AS on these paths; no object anywhere lists AS62390 as a provider**):

ASPA state	Hijack /24	Legit /16	Hetzner /24	Basis
Today (no ASPA on path)	Unknown	Unknown	Unknown	every hop "No Attestation"
ASPA(24940) published	**Invalid**	Unknown	Unknown / Valid†	hop 24940→62390 is "Not Provider+"; max_up_ramp < N
ASPA(24940) + ASPA(6204)	**Invalid**	Unknown	Unknown / Valid†	same; 6204's object further isolates the forgery

Paths: hijack `29504 15935 6204 62390 24940`; legitimate /16 as observed at RIS (e.g. `14907 1299 2914 24940`); Hetzner's countermeasure /24 (`852 3320 24940`). †Upstream (customer/peer-received) verdict / downstream (provider-received) verdict; the recommended mitigation policy rejects Invalid only.

Three points make this concrete for decision makers:

1. **One object from the victim flips the attack to Invalid everywhere ASPA verification runs.** Hetzner publishing a single ASPA listing its real transit (which by definition excludes AS62390) makes the hop $24940 \rightarrow 62390$ provably "Not Provider+". The up-ramp from the origin collapses to zero hops, and both the upstream and downstream variants of the algorithm halt at Invalid – no cooperation from the attacker, the attacker's transit, or any other network on the path is required.
2. **Zero collateral damage on Hetzner's legitimate routes.** The legitimate /16 and the countermeasure /24 remain Unknown (not Invalid) because their un-attested hops are "No Attestation," not "Not Provider+." Under the draft's recommended mitigation policy – reject Invalid only – none of Hetzner's legitimate traffic is affected by its own ASPA publication.
3. **The deployment story is one-sided.** Today the path is Unknown for the boring reason that nobody on it has published ASPA yet; AS62390 appears in no ASPA anywhere as a provider, so the attacker has prepared no cover for itself. The asymmetry is the argument: defenders opt in one object at a time, and each object hardens its own origin against forged-upstream hijacks regardless of what anyone else does.

The counterfactual is stark when set against what actually happened: with ASPA(24940) in place and ASPA verification enabled at even the major transit networks that carried the forged route (AS3257, AS5511, AS6830, AS3320 among them were all on observed paths), the 20:57:30 announcement would have been rejected at the first validating hop – no two-wave 22-hour diversion, no certificate minted through the hijacked path, no poisoned update channel.

4.3 TLS: the certificate authority validated through the hijack

With traffic for `files.virtualizor.com` (162.55.80.8) diverted, the attacker requested certificates from Let's Encrypt; the CA's automated HTTP domain-control validation also traversed the hijacked path and succeeded against the impostor. The resulting certificate was technically valid for **26 names** across the product family:

`virtualizor.com`	`softaculous.com`	`webuzo.com`	`sitepad.com`
`ampps.com`	`backuply.com`	`pagelayer.com`	`popularfx.com`
`api.*.com` (4 products)	`files.*.com` (5 products)	`server.softaculous.com`	`www.*` variants

(full list: disclosure Appendix A). No browser or API client presented a warning. The impostor host carried reverse DNS `server.softaculous.com`. A certificate obtained this way remains valid until revoked; the disclosure does not state a revocation time, hence the ≈31-hour worst-case availability above.

4.4 The update channel: unsigned packages closed the chain

Virtualizor's update client did not cryptographically verify package signatures at the time of the incident (acknowledged in the disclosure). A valid HTTPS response from the cloned endpoint was therefore sufficient to deliver the payload. The fake update presented itself as version 3.2.9.8; affected hosts logged the update as successful while remaining on 3.2.9.7 – the version mismatch in local logs is itself a detectable compromise indicator.

5 Supply-chain impact and indicators

The malicious responses were served by the attacker's system and never reached vendor logs, so **no definitive victim list exists**; every Virtualizor installation that performed an update check inside the diversion window must be treated as potentially affected. The disclosure's census: "a handful" of installations received the malicious package; no evidence of compromised guest VPSes at disclosure time – though a rooted hypervisor must be assumed to reach its guests.

Primary indicator (vendor + merged YARA rules, signature-base PR #417):

```
/etc/systemd/system/java-jre-update.service     # systemd persistence (main check)
/tmp/.vz_svc_done                               # staging marker
/tmp/widdow.jar                                 # Java payload, SHA-256
  b81a4e1fab9fc4e404d57224fe71e2c143aa93942bd46998789bdc944a7870c7
ed25519 SSH key  AAAAC3NzaC1lZDI1NTE5AAAAIP13pPAm5jmInLQYD3XNb3HwrW4cAKDcphoT4kSKrnte
C2 observed:    31.77.220.138:2025 (Java implant), 193.32.127.248
update logs:    "3.2.9.8 update succeeded" while version stays 3.2.9.7
```

Vendor guidance (summarized): check for the systemd unit (do not delete – preserve evidence); reset and IP-restrict all Virtualizor API keys; audit SSH `authorized_keys`, accounts, cron; rotate client-area credentials used during the window; run the vendor scan script; rebuild affected hypervisors rather than clean them.

6 Detection: the signal was public and one minute old

The forged route was visible in public collector data at 20:57:30, the second it was announced. The supply-chain consequences were community-confirmed 2.5 days later. The detection predicate for this incident class is one sentence:

> *A never-before-announced more-specific appeared inside an RPKI-valid covering block, and the AS immediately before the origin had no history of adjacency to that origin.*

RPKI cannot be the discriminator (it validates); the discriminator is the **penultimate hop**. All inputs are public: daily RIB snapshots provide the has-this-/24-ever-existed baseline, and the live update stream provides the new-announcement trigger. On this incident:

F1. Baseline. Zero `162.55.80.0/24` records in rrc00's 20:50–20:55 update file (BGPKIT-verified), and none in preceding RIBs – a genuinely new more-specific.

F2. Penultimate-hop novelty. In every observed path the origin-adjacent AS was 62390; no historical adjacency between 62390 and 24940 exists in the collectors' record.

F3. Corroboration within seconds. 36 distinct peers at rrc00, 20 at rrc01, and Route Views (independent project, 4 peers in its first covering file) within 3.5 minutes – multi-vantage, multi-transit-chain confirmation.

F4. Flap-storm signature. ≈10,600 withdrawals across the window (vendor event reconstruction) with persistent re-announcement; dampening suppression visible as coverage dropping to 1–7 peers mid-wave. Route instability correlated with a newly-appeared more-specific is a high-precision anomaly pair.

F5. CT cross-reference. A burst certificate issuance covering 26 product domains within the diversion window (disclosure; independent CT re-verification was attempted but crt.sh was unavailable during analysis). Correlating CT issuance bursts with covering-block route changes needs no non-public data.

7 Recommendations

R1. For transit providers: enforce strict customer prefix filters (IRR- or contract-based) with prefix-count alarms. This single control, at AS6204, would have confined the incident to one AS.

R2. For network operators: deploy ROV – it removes the large class of origin-hijacks – but do not credit it against forged-origin attacks. Track ASPA; verify paths, not just origins, as tooling matures.

R3. For software vendors: sign update packages and verify signatures against pinned keys in the client. This is the only layer that severs the routing-to-supply-chain chain once routing and TLS have both folded.

R4. For CAs: treat sudden covering-block route changes for a validated domain as a risk signal for DCV; multiple-perspective validation reduces a single-path hijack's ability to pass.

R5. For monitoring teams: implement the penultimate-hop novelty predicate over public MRT streams (baseline: daily RIBs; trigger: live updates). It fired on this incident within one minute of onset and requires no proprietary data.

R6. For hosting providers on shared panels: assume update-channel compromise is possible; version-pin, egress-filter update endpoints where feasible, and alert on version-mismatch-after-update.

8 Limitations and confidence

- **Control-plane only.** All diversion percentages are routing-visibility measures over the RIS peer sample (368 peers), not traffic volume, host counts, or measured forwarding. The flapping means per-network exposure was intermittent; the 72% figure is a peak-instant measure among peers holding a /24 route.
- **Vendor-derived quantities.** The 10-minute visibility curve, ≈10,600 withdrawal count, and 368-peer denominators are the disclosure's RIPEstat reconstruction; BGPKIT re-verified onset, countermeasure, wave-2, and end instants against raw MRT (matched to the second at onset; to the minute elsewhere) but did not recompute the full curve.
- **CT verification incomplete.** crt.sh returned HTTP 502 during analysis; certificate facts rest on the disclosure (Appendix A) and independent operator confirmation of the impostor host answering with a valid certificate.
- **Attribution.** BGP evidence establishes that AS62390 announced the forged route via AS6204. Whether NexonHost is complicit or itself compromised is *not established* by routing data; this report makes no attribution claim.
- **Panel scope.** Instant-level verification used rrc00, rrc01, route-views2; the full 368-peer aggregate derives from the vendor's RIPEstat methodology, whose 8-hourly RIB anchors (00:00/08:00/16:00 UTC rows) are its most reliable points.

Appendix A: anchor record (raw hex, 144 bytes)

The first forged `162.55.80.0/24` announcement as captured by RIS rrc00, MRT BGP4MP_MESSAGE_AS4 record, unix timestamp 1787950650 (2026-08-28 20:57:30 UTC). Paste into the wirescope toolbar to dissect offline:

```
6a91f63a001000040000008400007340000316e00000001b9c154bfc100041c
ffffffffffffffffffffffffffffffff007002000000554001010040021602
050000734000003e3f0000183c0000f3b60000616c400304b9c154bf800404
00000032c00818183c03e8183c03ea3e3f00643e3f006779fe000079fe0002
c0200c000079fe0000076c0000000118a23750
```

Sources

1. Virtualizor / Softaculous, “Security Incident – BGP Hijacking,” 2026-08-31. `https://www.virtualizor.com/blog/security-incident-bgp-hijacking/`
2. Neo23x0/signature-base PR #417, “Add rules for Virtualizor compromise” (YARA + IoCs), merged 2026-08-31. `https://github.com/Neo23x0/signature-base/pull/417`
3. LowEndTalk, “URGENT: Virtualizor Compromised (31st AUG)” – operator incident reports. `https://lowendtalk.com/discussion/220625/urgent-virtualizor-compromised-31st-aug/p1`
4. RIPE RIS raw MRT archives, rrc00/rrc01 2026.08 update files (Section 7 table). `https://data.ris.ripe.net/`
5. Route Views, route-views2 2026.08 update files. `https://archive.routeviews.org/`
6. RIPEstat `bgp-state` / `bgp-updates` / BGPlay, via the disclosure’s methodology. `https://stat.ripe.net/`
7. BGPKIT Broker v3, ROA history v2, monocle 1.4.0, bgpkit-parser, wirescope. `https://bgpkit.com`
8. PeeringDB records for AS62390 (NEXONHOST) and AS6204 (ZET.NET), retrieved 2026-08-31. `https://www.peeringdb.com/`

