Cloned, Loaded, and Stolen: How 109 Fake GitHub Repositories Delivered SmartLoader and StealC

By Maurice Fielenbach

Published: 2026-04-18 · Archived: 2026-05-06 02:01:29 UTC

Executive Summary

- We uncovered a malware distribution campaign built around **fake GitHub SmartLoader StealC** staging, where cloned open source repositories are used to deliver a LuaJIT-based loader and a follow-on stealer.
- We identified **109 malicious GitHub repositories** across **103 accounts** that impersonate legitimate projects and redirect users to embedded ZIP files containing a **LuaJIT-based SmartLoader stage**.
- The campaign appears to be operated by a **single threat actor or tightly controlled cluster** based on infrastructure overlap, synchronized repository updates, and consistent tooling.
- The staged malware is a **Prometheus-obfuscated Lua script executed by LuaJIT**. In the samples we analyzed, it uses the Windows FFI to call native APIs directly, hide execution, fingerprint the host, capture screenshots, and execute follow-on content in memory.
- SmartLoader resolves its active C2 through a **Polygon smart contract**, allowing the operator to rotate infrastructure without rebuilding the loader or updating every staged sample.
- **Collected host data** is exfiltrated to **bare-IP C2 servers** via multipart POST. The server then returns encrypted follow-on instructions and tasking.
- Persistence is established through **two scheduled tasks with separate recovery paths**. One executes a cached local copy. The other re-downloads a fresh Lua stage from GitHub.
- The same GitHub staging repository also hosted an encrypted **StealC** payload. SmartLoader's PE parsing and in-memory execution path are consistent with its use as the delivery and reflective loading layer for that follow-on stealer.
- Based on **PE compilation timestamps, ZIP metadata, and GitHub commit history**, the campaign has been active for at least **seven weeks**, with new repositories still appearing as of **2026-04-12**.

After someone impersonated one of our recent projects, PyrsistenceSniper, on GitHub, we uncovered a broader malware distribution campaign built around cloned open source repositories. The operator copies legitimate projects, republishes them under different accounts, strips the README of its technical content, and replaces it with prominent download buttons. Those buttons point to ZIP files hidden inside the repository tree rather than to GitHub releases or tagged source packages.

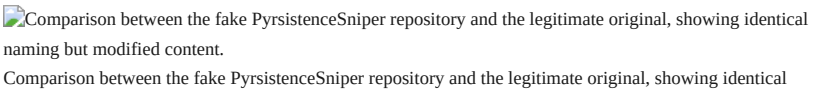
The source code is usually left mostly intact. That is what makes the lure work. At a glance, the repository still looks legitimate. The actor changes only the parts that influence user behavior: the README, the repository metadata, and the embedded archive. A user who trusts the project name or skims the source tree can easily be pushed toward a malicious download.

As of **2026-04-12**, we identified **109 malicious repositories** across **103 GitHub accounts**. Based on PE compilation timestamps, ZIP metadata, and GitHub commit history, the campaign has been active for at least **seven weeks**. New repositories continued to appear during our review.

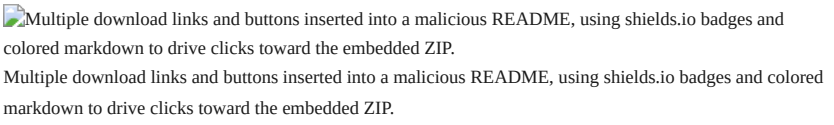
The infection chain is simple and effective. The victim extracts the ZIP and launches a batch file. That launcher starts a LuaJIT interpreter with an obfuscated Lua script as its argument. In the samples we analyzed, that SmartLoader stage hides execution, performs a native anti-debug check, resolves its current C2 through a Polygon smart contract, downloads a functionally overlapping second-stage Lua script from a separate GitHub repository tied to the same campaign, fingerprints the host, captures a screenshot, exfiltrates the collected data, and receives encrypted tasking in return. SmartLoader also contains the structures and execution primitives needed to decrypt and load PE payloads directly in memory, which is relevant because the same staging repository also hosted an encrypted StealC sample. It then establishes persistence through two scheduled tasks designed to survive partial cleanup.

The Lure – How the Fake GitHub Repositories Work

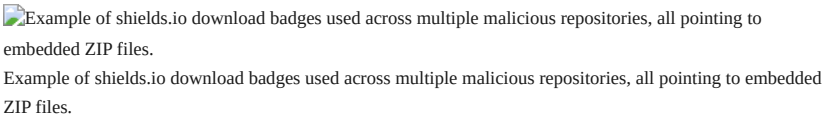
The operator appears to target repositories that are beginning to gain traction. New projects with recent activity and growing star counts are attractive because users are actively searching for them. By cloning those repositories under different accounts, the fake copies can appear in search results beside the real project.


Comparison between the fake PyrsistenceSniper repository and the legitimate original, showing identical naming but modified content.

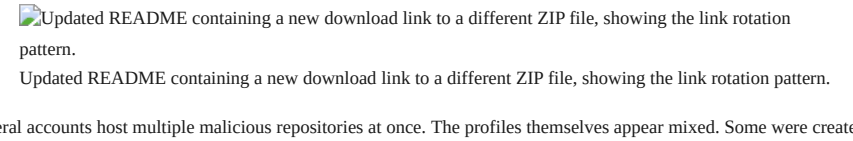
The README modifications follow a consistent pattern. Technical content is stripped out. Installation steps, prerequisites, development notes, and contribution guidance are removed. In their place, the actor inserts prominent download buttons using shields.io badges, colored markdown buttons, and direct links. Every link in the README points to the same ZIP file buried deep in the repository.


Multiple download links and buttons inserted into a malicious README, using shields.io badges and colored markdown to drive clicks toward the embedded ZIP.

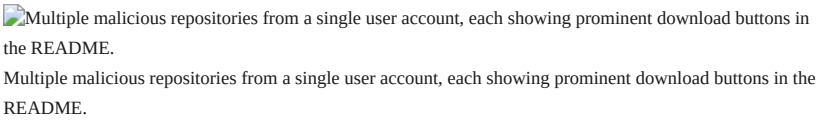
The actor also modifies repository descriptions and topics. In several cases, unrelated SEO terms were appended to increase search visibility. Combined with a cloned codebase and a plausible-looking repository structure, those additions help the fake project appear credible long enough to drive a download.


Example of shields.io download badges used across multiple malicious repositories, all pointing to embedded ZIP files.

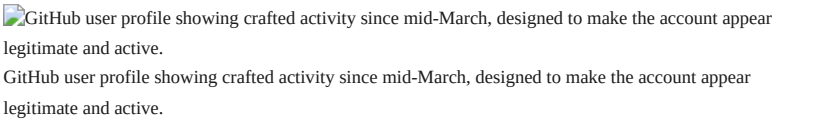
We assess this activity is primarily tied to a single threat actor or tightly controlled cluster. The strongest signals are operational consistency and infrastructure overlap. Repositories across different accounts are updated in batches when download links rotate to new ZIP files. The archive layout, README structure, staging pattern, and malware family remain stable throughout the campaign. That points to centralized control and at least partial automation.


Updated README containing a new download link to a different ZIP file, showing the link rotation pattern.

Several accounts host multiple malicious repositories at once. The profiles themselves appear mixed. Some were created recently and show little or no prior history. Others display older benign contributions before pivoting into malware distribution.

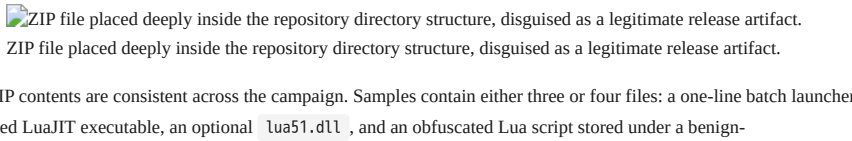

Multiple malicious repositories from a single user account, each showing prominent download buttons in the README.

Across the accounts we reviewed, recent contribution activity often appeared clustered rather than organically distributed over time. That pattern may reflect the operator's routine maintenance workflow, including repeated README and link updates across staged repositories, rather than meaningful long-term development activity.

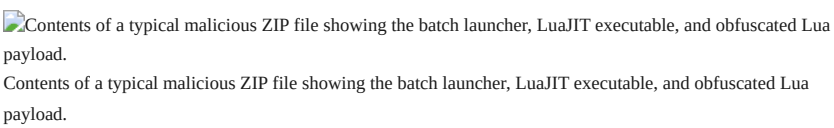

GitHub user profile showing crafted activity since mid-March, designed to make the account appear legitimate and active.

The ZIP Files – Bundled SmartLoader Archives

All 109 identified repositories contain a single ZIP file buried deep in the directory structure. The path often resembles an ordinary project archive, such as `repo/some/deep/path/project-name-version.zip`. That placement appears deliberate, intended to make the archive blend into the repository tree rather than stand out as the actual lure.


ZIP file placed deeply inside the repository directory structure, disguised as a legitimate release artifact.

The ZIP contents are consistent across the campaign. Samples contain either three or four files: a one-line batch launcher, a renamed LuaJIT executable, an optional `lua51.dll`, and an obfuscated Lua script stored under a benign-looking `.txt` or `.log` filename.


Contents of a typical malicious ZIP file showing the batch launcher, LuaJIT executable, and obfuscated Lua payload.

We identified six cosmetic variants across all repositories. Some use four files, with a small frontend that dynamically loads `lua51.dll`. Others collapse to three files by statically linking the Lua runtime. Executable names rotate between generic labels such as `loader.exe`, `unit.exe`, `boot.exe` and `java.exe`. Script filenames vary between `.txt` and `.log` extensions. However, the execution model is the same in every case.

Across all variants, the executables are LuaJIT 2.1.0-beta3 interpreter builds compiled as GUI subsystem PEs, so no console window appears when they run. The malicious logic sits in the bundled Lua script. File-level reuse is common. We observed byte-for-byte identical ZIP files reused across repositories in the same batch. ZIP timestamps ranged from **2026-02-19** to **2026-04-05**.

Technical Analysis

Execution Chain

The initial launcher is minimal. It is typically a single-line batch file:

```
start <exe> <payload>
```

The use of `start` detaches the new process, while the GUI subsystem flag suppresses the console window. The Lua stage then calls `GetConsoleWindow` and `ShowWindow(SW_HIDE)` through LuaJIT FFI as a second layer of concealment. From the victim's perspective, nothing visible happens on screen.


Content of the batch launcher showing the single-line start command.

The Executables

All observed executables are LuaJIT 2.1.0-beta3 PE64 GUI builds. None were signed and none contained version metadata. In the four-file variants, a small frontend loads `lua51.dll` dynamically. In the three-file variants, the Lua runtime is linked statically. We observed multiple compilation generations with regular refreshes, which aligns with the repeated ZIP and link rotation visible across the campaign.

The Obfuscated Lua Scripts

All staged Lua scripts are single-line files between roughly **296 KB and 309 KB**, obfuscated with **Prometheus**. Constants are hidden behind arithmetic identities, strings are permutation-encoded, variable names are randomized, and execution is routed through a custom VM dispatcher. C2 addresses and download URLs are stored inside encrypted blobs that require runtime decryption.


Excerpt of a Prometheus-obfuscated SmartLoader Lua script, showing the single-line format, randomized identifiers, encoded strings, and VM-style control flow.

The obfuscation is not decorative. It is central to how SmartLoader delays analysis and hides operational details until execution.

SmartLoader Behavior

SmartLoader uses LuaJIT's foreign function interface to call the Windows API directly from Lua. Early in execution, it loads `ffi` and declares a broad set of native structures and function prototypes, giving the script access to memory management, process control, windowing, and graphics APIs without requiring additional native components on disk.


LuaJIT FFI declarations used by SmartLoader to access Windows APIs directly, including memory management, process control, window handling, and graphics functions.

This execution flow is consistent with known SmartLoader behavior. It hides its console window, performs an anti-debug check using native shellcode copied into executable memory, resolves its C2, downloads a second Lua stage from GitHub, captures a screenshot through the GDI pipeline, fingerprints the system, and exfiltrates the collected data to C2.

The FFI declarations also include PE header and export parsing structures together with thread creation primitives, which is consistent with SmartLoader's known support for in-memory PE loading. In that context, the observed behavior matches SmartLoader's established role as a loader that supports host profiling, data collection, persistence, and follow-on execution.

Blockchain Dead Drop Resolver

SmartLoader does not hardcode its live C2 address directly in the Lua stage. Instead, it performs a JSON-RPC `eth_call` against `polygon.drpc.org` and queries the Polygon smart contract at `0x1823A9a0Ec8e0C25dD957D0841e3D41a4474bAdc` using function selector `0x3bc5de30`.


SmartLoader blockchain request, showing the JSON-RPC eth_call request to polygon.drpc.org and contract query.

SmartLoader blockchain request, showing the JSON-RPC eth_call request to polygon.drpc.org and contract query.

The contract returns an ABI-encoded string containing the active C2 URL. During analysis, we observed two returned addresses:

- `http://144.31.57.65`
- `http://144.31.57.67`

Both sit in the same `/24`.


Decoded contract request returning the active SmartLoader C2 address.

This functions as a dead drop resolver. The operator can rotate infrastructure by updating a single on-chain value rather than rebuilding the malware, modifying every staged sample, or changing DNS. It is a practical way to decouple malware distribution from C2 management and reduce the value of single-node takedowns.

C2 Communication

Once SmartLoader resolves the active server, it sends a `POST` request to `/api/<base64_victim_id>` on the bare-IP C2. The request body is `multipart/form-data` and includes host metadata together with collected data from the infected system. In the samples we analyzed, the exfiltrated content included screenshots and host fingerprinting details.


Captured SmartLoader HTTP POST request to the bare-IP C2, showing the multipart upload structure used to send host metadata and collected data.

The server responds with JSON containing two encrypted fields:

```
{
  "loader": "<base64-encoded encrypted instructions>",
  "tasks": "<base64-encoded encrypted task list>"
}
```

Follow-on task completion is later reported back through `POST /task/<base64_victim_id>`, which returns HTTP 204.

Persistence

SmartLoader establishes persistence through two daily scheduled tasks with separate recovery paths. The task names and local staging directory names vary across samples, but the persistence design remains consistent. In the samples we reviewed, names such as `AudioManager_ODM3` and `OfficeClickToRunTask_7d7757` were used to masquerade as legitimate software activity.

In one path, SmartLoader copies the LuaJIT binary, the downloaded second-stage Lua file, and `lua51.dll` into a directory under `%LOCALAPPDATA%` and executes the locally cached Lua stage on subsequent runs. That path continues to work even if the GitHub staging repository is removed after the initial infection.

In the second path, SmartLoader copies the interpreter and original payload components into a separate directory under `%LOCALAPPDATA%`, but does not rely only on the cached copy. On each execution, it reaches back to GitHub, downloads a fresh encrypted staging file, decrypts it in memory, and continues execution with the recovered Lua stage. That path restores access if the local second stage is deleted or corrupted.

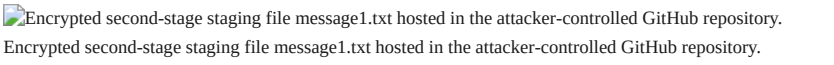

Scheduled tasks visible in Task Scheduler, masquerading as audio management and Office Click-to-Run.

Second-Stage Staging on GitHub

The second-stage repository observed in this infection chain was `deepanshugoel99/long`, a sparse GitHub repository with a single commit and minimal visible activity. It hosted two encrypted staging files under `long/long/` rather than plaintext payloads.

The file SmartLoader actively retrieved was `https://raw.githubusercontent.com/deepanshugoel99/long/refs/heads/main/long/long/message1.txt`.

That file is not a readable Lua script on its own. To understand what it contained, we traced the staged-content decryption routine in SmartLoader and recovered the key material used for the GitHub-hosted blobs. Applying that same routine to `message1.txt` recovered `a.lua`, a functionally overlapping Lua stage.


Encrypted second-stage staging file message1.txt hosted in the attacker-controlled GitHub repository.

That recovered Lua stage behaved almost identically to the original SmartLoader Lua payload. It resolved the same C2 through the same Polygon contract, communicated with the same infrastructure, and created the same scheduled tasks. The small differences were operational and packaging-related, not architectural. In practical terms, `a.lua` is best understood as a redundant SmartLoader stage used for persistence and re-delivery.

Staged StealC Payload

A second encrypted file, `message2.txt`, was present in the same staging repository and was fetched during the observed infection chain. We analyzed it using the same staged-content decryption logic and key base recovered from SmartLoader and used to decode `message1.txt`. That process recovered a packed x64 PE consistent with StealC. The decrypted binary has SHA256 `87de3e5a8ef669589c421220cd392ae8027a8f8d3cd97d35ac339f87dcff12c8` and a PE compilation timestamp of **2026-04-10 20:06 UTC**, roughly twelve minutes before the staging repository was created.

Operationally, that matters because it matches the execution model SmartLoader already exposes. The Lua stage defines PE parsing structures, allocates executable memory, resolves imports, and starts execution using thread creation primitives. In other words, the same loader used to retrieve and execute encrypted Lua content from GitHub is also capable of decrypting and reflectively loading a PE such as StealC without writing it to disk.

Detection

- **GitHub raw content downloads** — Monitor for outbound requests to `raw.githubusercontent.com` and `github.com` that fall outside established developer baselines. This detection carries inherent noise in engineering-heavy environments. It is most viable in organizations that already enforce strict egress controls and maintain an approved software inventory. Without that baseline, expect a high false-positive rate that limits operational value.
- **Batch-launched unsigned executables with script arguments** — Look for `cmd.exe` process creation events where the command line includes `start` followed by an unsigned executable and an argument ending in `.txt` or `.log`. Execution from user-writable locations such as `%TEMP%`, `Downloads`, or paths consistent with recently extracted archives increases confidence. The combination of an unsigned binary, a non-standard script extension, and a transient directory is unusual enough to warrant investigation even in noisy environments.
- **Lua runtime loaded from non-standard paths** — Track DLL load events for `lua51.dll` originating from directories outside expected software installation paths such as `Program Files`. Loading from `%TEMP%`, `%LOCALAPPDATA%`, or arbitrary user profile subdirectories has no legitimate justification in most enterprise environments and directly reflects the staging layout used throughout this campaign.
- **Blockchain RPC resolution from non-browser processes** — Identify DNS queries or outbound connections to `*.drpc.org` and known Polygon or Ethereum JSON-RPC endpoints initiated by processes other than web browsers. Outside of cryptocurrency wallets and blockchain development tooling, this activity has a very narrow legitimate footprint and serves as a reliable early indicator of dead drop resolver behavior.
- **Smart contract C2 resolution query** — Inspect HTTP POST bodies for `eth_call` requests targeting contract address `0x1823A9a0Ec8e0C25dD957D0841e3D41a4474bAdc` or referencing function selector `0x3bc5de30`. This is a high-fidelity indicator directly tied to the observed infrastructure and is unlikely to appear in benign traffic.
- **Multipart POST to bare-IP addresses** — Flag outbound `multipart/form-data` HTTP POST requests directed at bare IP addresses where the URI path starts with `/api/`. The absence of a hostname, combined with multipart encoding and a generic API path, is a pattern rarely seen in legitimate application traffic and aligns with SmartLoader's exfiltration behavior.
- **Task completion callbacks to bare-IP destinations** — Monitor for HTTP POST requests to URI paths matching `/task/` on bare-IP servers. In this campaign, SmartLoader uses this endpoint to confirm completed tasking. The combination of a bare IP, a short predictable path, and a POST with no meaningful response body is distinctive.
- **Scheduled task creation referencing %LOCALAPPDATA%** — Alert on `schtasks.exe /create` invocations where the scheduled action points to an executable under `%LOCALAPPDATA%`, particularly when the associated command-line arguments include `.txt`, `.log`, or `raw.githubusercontent.com`. The creation of two or more such tasks within a short window is a strong compound signal. After baselining legitimate scheduled task behavior, this detection should produce minimal noise in most environments.
- **ZIP archives pairing Lua runtime components with batch launchers** — Detect the co-occurrence of `lua51.dll` or a LuaJIT interpreter binary alongside `.bat` or `.cmd` files within the same archive or newly created directory tree. This combination is structurally consistent across every observed variant in the campaign and is uncommon enough in legitimate software distribution to serve as a reliable static or on-access indicator.

Prevention

This campaign depends on a user downloading a ZIP from a fake repository and running the included launcher, which makes source verification the most effective control. Users should validate the publisher and prefer official releases over archives buried in the repository tree. Traffic inspection at the proxy or TLS layer helps surface staged payload retrieval and bare-IP exfiltration. Application control blocks unsigned interpreters and script launchers from running out of user-writable paths. Egress rules that block Polygon and other blockchain RPC endpoints disrupt the dead drop resolver, and restricting raw GitHub downloads limits second-stage staging.

Wrap Up

This campaign is effective because it keeps the lure simple and the execution chain quiet. The operator cloned 109 repositories across 103 accounts, replaced legitimate documentation with download funnels, and delivered a LuaJIT-based SmartLoader stage that hides execution, resolves C2 through a blockchain dead drop, collects and exfiltrates host data, maintains persistence through two independent scheduled tasks, and supports in-memory follow-on payload execution.

The technical choices are pragmatic. GitHub provides both initial delivery and second-stage staging. A Polygon smart contract provides an updateable C2 resolver. Bare-IP servers handle collection and tasking. LuaJIT FFI provides direct access to native APIs without requiring additional dropped components. And the same staging workflow can support both redundant Lua stages and encrypted PE follow-on payloads such as StealC. Together, those choices make the campaign easy to replicate, easy to refresh, and harder to disrupt through single-point takedowns.

We will keep monitoring and submitting removal requests. But the broader lesson is straightforward. Repository removal is reactive. Impersonation is cheap, trusted project names are easy to reuse, and fake repositories can be recreated faster than they are taken down. Maintainers should monitor for lookalike projects, and users should verify the source of any GitHub-hosted tool before running it.

Indicators of Compromise

File Hashes

IoC	Type	Category	Filename
`2273702dfbcfd96a6ed7bdb42ba130291b653869256ec1325bc7fe30e8d9b70a`	SHA256	ZIP archive	`Sniper_Pyrsistence_v2.6.z`
`2d72abb33b8428a3a73fb64a03e6ac84595c4b1636f190f2936fadec3c8792f6`	SHA256	ZIP archive	`founders-kit-3.0-beta.5.z`
`b04db6cae604d2ab1542e3c0cf1a4a3bb8d76562556f7275efe25bb90fc1da19`	SHA256	ZIP archive	`score_global_health_assist`
`d92ac938494c2c74c73f3ca28c5c7148d0a03024b46630192a9348259b7b3665`	SHA256	ZIP archive	`Player_One_3.7.zip`
`3299b85734e03cbb767d10f89384f666c35d6863198a7c6c0004ef19fcc76bc3`	SHA256	ZIP archive	`nightlights-district-india`
`c324560d4310849fd6b86e126514b20512905eee7ee94a2152f4314bb4055649`	SHA256	ZIP archive	`dumper-cpp-1.7.zip`
`2c3c4f1e3401c7baa804c21164b17a2ab50b3462ab09fcdccec35c8faa8e17fb`	SHA256	ZIP archive	`client-qclaw-wechat-3.5-be`
`58b98acb7dc26d8130c20b38ad040e5e7042eac38f12205248595697143c4297`	SHA256	ZIP archive	`skills_product_ai_v3.9-bet`
`10cbcb3fb25205a53ea9fe4fad46f45a349f7da8de22dd53a1ce16a920059720`	SHA256	ZIP archive	`skill_uncodixify_v3.8.zip`
`bc95563880f17f4c3fc0fd8d3f7abc37b14ffb3daa627f92d5bd0b4f457d54e2`	SHA256	ZIP archive	`commerce-agent-enstatite.z`
`13690a008d375908399e7f0bf8d1b4733498f1145166c7788fb9966c3b551b2f`	SHA256	ZIP archive	`Server-Pirate-LL-ferfet.z`
`12a09f9425cd4058956214b237ec82577c7b9ae15f323c28d3b4ad846d0d2f6b`	SHA256	ZIP archive	`ts-symphony-chloromethane`
`b599a00d1226f6e0d433bf9be89958d6d4600a365c8e16bd86b4603e2552bf37`	SHA256	ZIP archive	`Software_v2.2.zip`

IoC	Type	Category	Filename
`998e14a100d1f541f9fd59f4e58dd86e76fe7a105b02646d3487f18583d46c42`	SHA256	ZIP archive	`calendar-dashboard-task-2`
`ebe63bd7715e7b2ff0b25c9bb6540a904f7195fb9fb2d405bd0cb5c0c4d34476`	SHA256	ZIP archive	`Software-3.2.zip`
`3217ae928395e00d873cccdf2adfa7828fe319fc84501453e702af91a0af2596`	SHA256	ZIP archive	`Token-Stream-2.2.zip`
`e12e7d4d7c5ccf825c9c0ba3af32a9d575d1624ad6e3998e7a71c3e0939c0d61`	SHA256	ZIP archive	`Download-Ninja-Full-Ripper`
`012b49c7f60bbe0501e61fc62c9b7dd69be9bbf15cb36a840a293b3cb066b865`	SHA256	ZIP archive	`agent_workspace_v1.5.zip`
`275275b5099a63724b6f525c1e9de082829e710078e8605c0286998b5a02e75d`	SHA256	ZIP archive	`autotrader-openclaw-v3.1-b`
`7e107cc2db66be4c9a90c2ef81f21ae2893962e1040531dff0305d9283f27387`	SHA256	ZIP archive	`Stealer_Hades_1.4-beta.3.z`
`768c28bff5e2ccd991a4a5cbfe3331015e3262cbc007829631483b46aa582cbc`	SHA256	ZIP archive	`Impacket_Reference_1.5.zip`
`6b518855404b7281246aab93a46288b25f0cb0f09cdeb820e677ef615bf3fda4`	SHA256	ZIP archive	`Open-RL-Claw-3.9.zip`
`572acf0d7a3801b9bc41f626bac781d75ea1f99770b176079ae5f9a347c09b78`	SHA256	ZIP archive	`Amazon_Analysis_Product_Ex`
`4b3231da6ba13aa1e1eb8dd371e287bc18505273bca5bda80065c60b024549b8`	SHA256	ZIP archive	`MCP_EVOKOR_3.4.zip`
`1aacc8cb9694293dc152891fc26de64a2061b31a066d297373cdc87da54b6fd8`	SHA256	ZIP archive	`Audio_Auditor_v1.8.zip`
`d142be1fc9a7eec7ec26aeea75e5f7a175c4ab9b2ee36b958280873bc3861b2e`	SHA256	ZIP archive	`smart_miner_money_2.0.zip`
`a50c4c26597cb4dc3ce340e1de0ec929b4a7ab0954a6ba214a32f158f01d6a8a`	SHA256	ZIP archive	`Gov_Tracker_Blocklist_madn`
`9de5dc4192a9dea43d9ff6289bb276bb3f2c244c15821b6d31fab90258b23149`	SHA256	ZIP archive	`Glider-UI-v2.7.zip`
`2149a0c948d87f6a80ebb4abddf742c2383f59c7558a313caa0c0fd3bd3cdeef`	SHA256	ZIP archive	`Enhance-Prompt-1.6.zip`
`a6fce76371d8b950b22bbea5a94d5688c19368979d06b2ef3c41f18ce6ada4c3`	SHA256	ZIP archive	`waf-mango-1.2.zip`
`a90898926236de8b574b50ef8c6c0411b193383d6db2214d73ead27c65867fd5`	SHA256	ZIP archive	`Ultra-Clopix-Delta-v2.7.zi`
`44d5de84ee0c31517d114640ba9b9b307ea9e1ec4e591de42cbb4d07ceb5e6c3`	SHA256	ZIP archive	`App-Vesper-AI-2.9.zip`
`f5bdf3d6c1376476b0d9eb0e74aa5aa8ccc7378068531c8346d76fbef04c6a6e`	SHA256	ZIP archive	`hub_premium_access_marketn`
`5b09803d2acbec734d9c88496f9590bb7cbaf5392607ef0f20f79fe177f7fd83`	SHA256	ZIP archive	`Dumper_Zygisk_Il_Cpp_v3.4`
`d8469b109bb22ad367c19971e1065074527af144d4c1e7d7a4cfb0f2d6e12767`	SHA256	ZIP archive	`dashboard-pump-fun-v2.6.zi`
`9f6368bccedd005075fe991719719b0df5af22df697cab76aa6b4392d38394b1`	SHA256	ZIP archive	`bread_simulator_run_toolki`

IoC	Type	Category	Filename
`0a4bce0f0461335585550598ff33c40a389465f7d0094212bee40b7f525de123`	SHA256	ZIP archive	`hoshan-vehicles-2.3.zip`
`f6d10e879324c36914002ebd989e1a6fdc50e29257078a95e975f18b42f69836`	SHA256	ZIP archive	`ecommerce_theme_gatsby_v3`
`fad3d429172932b72e50f52af169a80439464e3538d97810509090e2e6cdf32a`	SHA256	SmartLoader LuaJIT interpreter	`loader.exe`
`bff0904456e3151221d29ed1d7c88fc31587efbdfb28817cdcb7ec7f20cade21`	SHA256	SmartLoader LuaJIT interpreter	`selector.exe`
`bbd438d3d7a59152f1dd5e45bb8d22ee1c07f95cfe42cebbe756aaf4feadc875`	SHA256	SmartLoader LuaJIT interpreter	`unit.exe`
`d1557bc3f5d8542f9b7f8e80b02283397d2e437386a6662251c4fc7342167cda`	SHA256	SmartLoader LuaJIT interpreter	`unit.exe`
`167b166e26dd44f580a00f2c879089c5362eff5120ac88e0701b11b1eb320ca9`	SHA256	SmartLoader LuaJIT interpreter	`load.exe`
`8b42ca9d05badf0e7327d816a56e5516431ae34627da68e12ae9347f365b2668`	SHA256	SmartLoader LuaJIT interpreter	`compiler.exe`
`d56213d08fb10c880f17e1a262bf1176cf234d1fc591188171e7be9cd856eb12`	SHA256	SmartLoader LuaJIT interpreter	`util.exe`
`3595a6b226ce4daa0a28edea152b3a887c01f6323db1d082f6568c995cdefb55`	SHA256	SmartLoader LuaJIT interpreter	`luajit.exe`
`e69873a3ef03b289aba8a0ec7130247dc5f2a3ce8c3b647b44518a899f39f789`	SHA256	SmartLoader LuaJIT interpreter	`luajit.exe`
`f3e34c9e36f3be065d80d456281d31dd1cc85eb4980db7fa8c1b0eb6f29c25d8`	SHA256	SmartLoader LuaJIT interpreter	`luajit.exe`
`09e0f7616dfd2f7eb2876f6ef7331d6dbc78775acd594a94b0397a56717d1fcc`	SHA256	SmartLoader Lua runtime DLL	`lua51.dll`
`440ceb0dc5911faca54ed9a4dd186dad3d006ae4f52d0bb7d1e4b4edd8c3693a`	SHA256	SmartLoader Lua runtime DLL	`lua51.dll`
`e450152d8dd9f7d2d92dbd53461a38ee8f154b69b2558ed43b5d3f603a43240a`	SHA256	SmartLoader obfuscated Lua payload	`proto.txt`
`76afe60e675e68906a2de61d45c46aa6502fe7f9c298260c226a4382744f4212`	SHA256	SmartLoader obfuscated Lua payload	`func.txt`
`25aff351f5b4195f33e2fb862f71e3668e699f2311e7844e277b8256a6cb47c0`	SHA256	SmartLoader obfuscated Lua payload	`package.txt`
`54bbd79ed1ee26d3e7aa079963ba26c36aa683c01cc8b05b6d255da8634df006`	SHA256	SmartLoader obfuscated Lua payload	`package.txt`
`830ec7352972fd1eb24fcaf72349ef9a27dd9f26f24552d6b68b87ffeada1212`	SHA256	SmartLoader obfuscated Lua payload	`package.txt`
`f9436ccb986760ca379d6cd2f00726e032a1d9c250a9bd261d40d98b914e7ef9`	SHA256	SmartLoader obfuscated Lua payload	`dynasm.txt`
`3989cdf958d258244f3a72bac594214112ffe1008d4d81233a5911482dd302ca`	SHA256	SmartLoader obfuscated Lua payload	`buff.log`

IoC	Type	Category	Filename
`c7b71a992c6ca1467164b643136d986c0eec28548f30533456a3ea0f442c85a8`	SHA256	SmartLoader obfuscated Lua payload	`buff.log`
`59c2115caf3104184de6cbc64c4029886b7302e1fa58acc910a2c567222e8616`	SHA256	SmartLoader obfuscated Lua payload	`static.txt`
`8cede35b80b1deaf732c2b178d908f91b3e7a0c114d06dfae9075b8a9bf78b8f`	SHA256	SmartLoader obfuscated Lua payload	`uix.txt`
`d067cacea4ec623dc715c27ff7568d14988af0be1f3db32d332f27744114f9ba`	SHA256	SmartLoader obfuscated Lua payload	`x64.txt`
`a93ac4fb3f9dadc22f7b7f1877bc99b84a77fe3bbe560bdd20bbd7c4b6f9c1d6`	SHA256	SmartLoader obfuscated Lua payload	`tree.txt`
`e1e6e28bc665b242fd4b496caf2542042d5720e87ea74551735664c202c486c7`	SHA256	SmartLoader launcher script	`Launcher.bat`
`3658fc38c10867e30e3c5c98a7a392e452a4ba497c8a674ff26554bc09f032b0`	SHA256	SmartLoader launcher script	`Application.cmd`
`b0f0b6e38f77c518ebfaf691d729636d82cc59dc2a329d7454e11f74a2cb2d3f`	SHA256	SmartLoader launcher script	`Application.cmd`
`cd4d2b6dc9c764c3f2b2b003bce035053a8ce81420c7ea886c76611219cae4ae`	SHA256	SmartLoader launcher script	`Application.cmd`
`7e8bd9ba64fcbd1cb753baa2f7bc8d5d7f3e91552bcdc9ec1ec04edd4916ff33`	SHA256	SmartLoader launcher script	`Application.cmd`
`af6f59bd3caee5daa2d6765dd8c1bc167060a9681617ee1e2aff32f1eda3477c`	SHA256	SmartLoader launcher script	`Application.cmd`
`a91b3308a7e9aa9fa660c72d27f226d8f50bfac2629f79a828fbecff323c0fe0`	SHA256	SmartLoader launcher script	`Application.bat / App.bat`
`c3b56d68c80c4a6a9879c45a7761a538e3546644623af1ee469d3b70130fa0cd`	SHA256	SmartLoader launcher script	`Application.bat`
`ce1e33483d353200a266b3bc383ccf500e5a760c6dcd8218747260f5bbe39509`	SHA256	SmartLoader launcher script	`Launcher.cmd`
`592ec6f529721acbe07100c5386c58ca20fddfee7ac90280943fc2a61661e2be`	SHA256	SmartLoader launcher script	`Launch.bat`
`212C76DAF355EDE116EB04D4F9D08A112D07940A14DC248BC568FF1BA0A64E18`	SHA256	Encrypted second-stage on GitHub (`message1.txt`)	2026-04-12
`87de3e5a8ef669589c421220cd392ae8027a8f8d3cd97d35ac339f87dcff12c8`	SHA256	StealC packed binary decrypted from `message2.txt`	2026-04-12

Network Indicators

IoC	Type	Category	L S
`144.31.57.67`	IPv4	SmartLoader C2, SERVHOST-AS, United Kingdom	2(0
`144.31.57.65`	IPv4	SmartLoader C2, SERVHOST-AS, United Kingdom	2(0
`213.176.73.149`	IPv4	StealC C2	2(0
`https://raw.githubusercontent.com/deepanshugoel99/long/refs/heads/main/long/long/message1.txt`	URL	Second-stage Lua payload hosted on GitHub	2(0
`https://raw.githubusercontent.com/deepanshugoel99/long/refs/heads/main/long/long/message2.txt`	URL	StealC payload hosted on GitHub	2(0
`0x1823A9a0Ec8e0C25dD957D0841e3D41a4474bAdc`	Polygon contract	Blockchain dead drop resolver returning C2 IP	2(0
`0x3bc5de30`	Function selector	Smart contract method called to retrieve C2 address	2(0
`polygon.drpc.org`	Domain	Polygon RPC endpoint used by SmartLoader	2(0
`POST /api/<base64_victim_id>`	URL pattern	Exfiltration endpoint (multipart/form-data)	2(0
`POST /task/<base64_victim_id>`	URL pattern	Task completion callback endpoint	2(0

Repositories

IoC	Type	Category	Last Seen
`https://github.com/stcitlab1/PyrsistenceSniper/`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Shonpersus/founders-kit`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/therajeshpatil/home-assistant-global-health-score`	URL	Malicious GitHub repository	2026-04-12

IoC	Type	Category	Last Seen
`https://github.com/Cherishpolyploid691/One-Player`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/eltayep2/india-district-nightlights-viirs`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/jayed50/cpp-dumper`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Arlinablind800/qclaw-wechat-client`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/amosshadowy76/ai-product-skills`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/anubhavsingh-0218/uncodixify-skill`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/AlexSilgidzhiyan/agent-commerce`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/somya-droid/Pirate-LLM-Server`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/ashiskumarnanda/symphony-ts`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/rakibul3790/mdexplore`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Cobras1934/task-calendar-dashboard`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/mohadesehfllh/whispr`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/FILDA007/TokenStream`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/halim2023/Ninja-Ripper-2.13-Full-Download`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/pandu1992/agent-workspace`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/freefire2chyko-a11y/openclaw-autotrader`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/silent-whisper/Hades-Stealer`	URL	Malicious GitHub repository	2026-04-12

IoC	Type	Category	Last Seen
`https://github.com/Lyrothanak20/Impacket_Reference`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/sabalearning01/OpenClaw-RL`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/minullaksen/Amazon_Sales_Product_-_Revenue_Analysis_Excel`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/shripadk1999/EVOKORE-MCP`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/viktor820/AudioAuditor`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/arnautoff1/smart-money-miner`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/CobraZero969/EU-Gov-Tracker-Blocklist-by-madnesscc`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/hayate001/GliderUI`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/wtfhanin/Enhance-Prompt`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/jakariyaox-dot/mango-waf`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/AdebSamra/Delta-Clopix-Ultra`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/shmilymaria/VesperAIApp`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/AdebSamra/marketmuse-premium-access-hub`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/hayate001/Zygisk-Il2CppDumper`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/arnautoff1/pump-fun-dashboard`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/sabalearning01/bread-run-simulator-toolkit`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/eltayep2/hoshan-vehicles`	URL	Malicious GitHub repository	2026-04-12

IoC	Type	Category	Last Seen
`https://github.com/mohadesehfllh/gatsby-ecommerce-theme`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Casheu1/perplexity-2api-python`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/usernamedoxelghk/WindsurfSwitch`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/GamerX3560/Aria-V-7.1`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/oliverkanda254/medusa-mobile-react-native`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/haren2312/medusa-mobile-react-native`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Jonaskouame/Phone-Number-Tracker`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/h4vzz/awesome-ai-agent-skills`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Sriv4/insta-hack-termux`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/renny2020/Open-UI`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/YahiaGrdh/vibe-agents`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/abuferas1262/yandex-speedtest-cli`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/CuddlyPaws22/codeclaw`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/jessevanwyk1/claude-scholar`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/ejfhgo/hacker-Toolkit`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Ksalazar29/deepseek-claw`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Pr-E/openclaw-master-skills`	URL	Malicious GitHub repository	2026-04-12

IoC	Type	Category	Last Seen
`https://github.com/mreshuu/STForensicMacOS`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/phongdshh-debug/Ghost-MSG`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/TalangoJames/fractals`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/GH8ST007/llms_with_google_cloud`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/mohamedfaro7/Chuks-YT-Live_AI`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/mrizky214/task-runner-1771921051-1`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/sidiishan/soul.py`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Xhtira20/scraped`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Always15dppk/register`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/omkargundle/claude-usage-bar`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/fajarsm14/epic-games`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/JoOdSy/mini-apps`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/MPB0828/Greenhouse-Gas-Emissions-Forecasting-with-ARIMA-LSTM`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/twinklew9/notes2latex`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Sawyer60/Dataset_HealthHub`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/vlsienthusiast00x/Spodrue`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/IvannGonzzalez/hve-core`	URL	Malicious GitHub repository	2026-04-12

IoC	Type	Category	Last Seen
`https://github.com/Aditya923-c/xpoz-agent-skills`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Shavan889/minisforum-ms-s1-max-bios`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/DarkSliceYT/ai-infra-index`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Seragatia/DocGenie`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/whydixit/cursor-starter`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Tawhidhere/OneRec-Think`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/rushikeshjaware/DiffusionDriveV2`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Abisheak250402/cloakbrowser-human`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Loune3213/Wazuh-Openclaw-Autopilot`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Ragulrajtcestd/LSTM-Optuna`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Tod-weenieroast366/coding-plan-mask`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/KemalFasa/discord-adapter-meme`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Bhin4787/AI-Powered-Ticket-Routing-SLA-Breach-Prediction-in-JIRA`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Jacksonsmg/SoftwareTesting-Cunit`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/linhkat3057/Valthrun`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/DIMANANDEZ/refrag`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/MichaelQDLe/CodeHive`	URL	Malicious GitHub repository	2026-04-12

IoC	Type	Category	Last Seen
`https://github.com/VantageSolutions/ShadowTool`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Pataterustiche/tonconnect`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/cristiancctlv/recaptcha-botguard`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/marciunyielding712/openage`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Ali-Shady/claude-agent-desktop`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/2aryanZ/paper-submission-check`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Hosk9612/venutian-antfarm`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/MohamedSamiHdj/realtime-data-pipeline`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/alvfpinedo/go-prometheus-exporter`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/MrKillerq/Mini-o3`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/vickykumar11062/Replication-package-for-gender-and-regional-differences-in-scientific-mobility-and-immobility`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/nonunion-loasa895/codapter`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/WILLIAM86-CAPTAIN/gooey-search-tabs`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/hama1981/ROBLOX-MACRO-V3.0.0`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/syedabdullahuddin/n8n-workflow-sdk-mcp`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/wanderconnect01/ika-network-skill`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/okoid721/chloroDAG`	URL	Malicious GitHub repository	2026-04-12

IoC	Type	Category	Last Seen
`https://github.com/Valentin6595/WhatDreamsCost-ComfyUI`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/virginiadiom2000-ai/osv-ui`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/gage6903/son-of-claude`	URL	Malicious GitHub repository	2026-04-12
`https://github.com/Milan-sisodia-27/idl-pu3`	URL	Malicious GitHub repository	2026-04-12
