

Intellexa's Prolific Zero-Day Exploits Continue

 cloud.google.com/blog/topics/threat-intelligence/intellexa-zero-day-exploits-continue

Google Threat Intelligence Group

December 3, 2025



Google Cloud

Threat Intelligence

Google Threat Intelligence

Visibility and context on the threats that matter most.

[Contact Us & Get a Demo](#)

Introduction

Despite extensive scrutiny and [public reporting](#), commercial surveillance vendors continue to operate unimpeded. A prominent name continues to surface in the world of mercenary spyware, Intellexa. Known for its “Predator” spyware, the company was [sanctioned by the US Government](#). New Google Threat Intelligence Group (GTIG) analysis shows that Intellexa is evading restrictions and thriving.

Intellexa has adapted, evaded restrictions, and continues selling digital weapons to the highest bidders. Alongside research published by our colleagues from [Recorded Future](#) and [Amnesty](#), this blog post will shed light on Intellexa’s recent activities, unveil the real-world impact of their surveillance tools, and detail the actions we are taking against this industry.

Continued Prolific Exploitation of Zero-Day Vulnerabilities

Over the past several years, Intellexa has solidified its position as one of, if not the most, prolific spyware vendors exploiting zero-day vulnerabilities against mobile browsers. Despite the consistent efforts of security researchers and platform vendors to identify and patch these flaws, Intellexa repeatedly demonstrates an ability to procure or develop new zero-day exploits, quickly adapting and continuing operations for their customers.

Intellexa is responsible for a substantial number of the zero-day vulnerabilities identified over the years by Google's Threat Analysis Group (TAG), now part of GTIG. As an example, out of approximately 70 zero-day vulnerabilities discovered and documented by TAG since 2021, Intellexa accounts for 15 unique zero-days, including Remote Code Execution (RCE), Sandbox Escape (SBX), and Local Privilege Escalation (LPE) vulnerabilities. All of these zero-days have been patched by the respective vendors. In addition to developing exploitation of zero-days, we increasingly see evidence that Intellexa is purchasing steps of exploit chains from external entities.

CVE	Role	Vendor	Product	Type	Description
CVE-2025-48543	SBX+LPE	Google	Android	Memory corruption	Use-After-Free in Android Runtime
CVE-2025-6554	RCE	Google	Chrome	Memory corruption	Type confusion in V8
CVE-2023-41993	RCE	Apple	iOS	Memory Corruption	WebKit JIT RCE
CVE-2023-41992	SBX+LPE	Apple	iOS	Memory Corruption	Kernel IPC Use-After-Free
CVE-2023-41991	LPE	Apple	iOS	Code Signing Bypass	Code Signing Bypass
CVE-2024-4610	LPE	ARM	Mali	Memory Corruption	Improper GPU memory processing operations

CVE-2023-4762	RCE	Google	Chrome	Memory corruption	Type confusion in V8
CVE-2023-3079	RCE	Google	Chrome	Memory Corruption	Type Confusion in V8
CVE-2023-2136	SBX	Google	Skia	Memory Corruption	Integer overflow in Skia SKSL
CVE-2023-2033	RCE	Google	Chrome	Memory Corruption	Use-After-Free in V8
CVE-2021-38003	RCE	Google	Chrome	Memory Corruption	Inappropriate implementation in V8
CVE-2021-38000	RCE	Google	Chrome	Logic/Design Flaw	Insufficient validation of untrusted input in Intents
CVE-2021-37976	SBX	Google	Chrome	Memory Corruption	Information leak in memory_instrumentation
CVE-2021-37973	SBX	Google	Chrome	Memory Corruption	Use-after-free in Portals
CVE-2021-1048	SBX+LPE	Google	Android	Memory Corruption	Use-After-Free in ep_loop_check_proc

Table 1: Zero-days associated with Intellexa since 2021

Exploit Chain

[Partnering](#) with our colleagues at CitizenLab in 2023, we [captured](#) a full iOS zero-day exploit chain used in the wild against targets in Egypt. Developed by Intellexa, this exploit chain was used to install spyware publicly known as Predator surreptitiously onto a device. According to metadata,

Intellexa referred to this exploit chain internally as “smack.”

First Stage: JSKit Framework Déjà Vu

The initial stage of the exploit chain was a Safari RCE zero-day that Apple fixed as CVE-2023-41993. The exploit leveraged a framework internally called “JSKit.” Once arbitrary memory read and write primitives have been achieved thanks to a vulnerability in the renderer, in this case CVE-2023-41993, the framework provides all the requisite components to perform native code execution on modern Apple devices.

We believe that Intellexa acquired their iOS RCE exploits from an external entity, as we have seen this exact same JSKit framework used by other surveillance vendors and government-backed attackers since 2021. In 2024, we [reported publicly](#) on a campaign by Russian government-backed attackers using this exact same iOS exploit and JSKit framework in a watering hole attack against Mongolian government websites. We have also seen it used in other campaigns by surveillance vendors, including another surveillance vendor using the same framework when exploiting CVE-2022-42856 in 2022.

The JSKit framework is well maintained, supports a wide range of iOS versions, and is modular enough to support different Pointer Authentication Code (PAC) bypasses and code execution techniques. The framework can parse in-memory Mach-O binaries to resolve custom symbols and can ultimately manually map and execute Mach-O binaries directly from memory. In addition, the JSKit framework is fairly robust and well engineered, with each step of the exploitation process tested carefully. To date, we haven't seen a similar framework exist for Android.

Figure 1: Example of testing and validating shellcode execution

The exploit Intellexa used was apparently tracked internally as “exploit number 7,” according to debug strings at the entry point of the RCE exploit. This suggests that the external entity supplying exploits likely possesses a substantial number of iOS exploits targeting a wide range of versions.

Figure 2: Debug string suggesting multiple iOS exploits

Regarding Chrome exploitation, Intellexa has used a custom framework with all the features needed to gain code execution from any arbitrary vulnerability capable of leaking [TheHole magic object in V8](#). They first used this framework with CVE-2021-38003, then with CVE-2023-4762, CVE-2023-3079, CVE-2023-2033, and more recently in June 2025 with CVE-2025-6554, observed in Saudi Arabia. This most recent, CVE-2025-6554, was a type confusion error in Chrome’s v8 engine. Chrome quickly mitigated the issue for all Chrome users with a configuration change and then [fixed](#) the bug as CVE-2025-6554 in version 138.0.7204.96. All these CVEs are vulnerabilities in V8 that all can be used to leak TheHole object.

Following Stages: Watching the Helper

The second stage is the most technical part of the chain and would require an entire separate blog post to describe all of its functionality. Essentially, this stage is in charge of breaking out of the Safari sandbox and executing an untrusted third stage payload as system by abusing the kernel vulnerabilities CVE-2023-41991 and CVE-2023-41992. This second stage communicates with the first stage to re-use some of the primitives (e.g., PAC bypass) and offers kernel memory read/write capabilities to the third stage.

The third stage (tracked by GTIG as PREYHUNTER) is the last one we captured and is composed of two modules called "helper" and "watcher."

The watcher module primarily ensures that the infected device does not exhibit suspicious behavior; if such behavior is detected, a notification is generated, and the exploitation process is terminated. The module is also in charge of monitoring crashes.

The following behaviors are detected:

- Developer mode via `security.mac.amfi.developer_mode_status`
- Console attached via diagnosticd
- US or IL locale set on the phone
- Cydia installed
- Bash, tcpdump, frida, sshd, or checkrain process currently running on the phone
- McAfee, AvastMobileSecurity, or NortonMobileSecurity installed on the phone
- Custom HTTP proxy setup
- Custom root CA installed

The helper module is communicating with the other parts of the exploit via a Unix socket at `/tmp/helper.sock`. Similar to the [ALIEN malware for Android](#), the module has the ability to hook various places with custom frameworks called DMHooker and UMHooker. These hooks are allowing the module to perform basic spyware capabilities such as:

- Recording VOIP conversations (stored in `/private/var/tmp/1/voip_%lu_%u_PART.m4a`)
- Running a keylogger
- Capturing pictures from the camera

The module is also hooking into the SpringBoard in order to hide user notifications caused by the aforementioned actions. We believe these capabilities are provided to the operator to make sure the infected device is the correct one before deploying a more sophisticated spyware, such as Predator.

The binary left compilation artifacts such as the following build directory including the name of the exploit chain.

`/Users/gitlab_ci_2/builds/jbSFKQv5/0/roe/ios16.5-smackjs8-production/.`

Overall, these exploits are high in sophistication, especially compared to the less sophisticated spyware stager, supporting our assessment that the exploits were likely acquired from another party.

Disrupting Novel Delivery Capabilities

The primary delivery mechanism for Intellexa's exploits remains one-time links sent to targets directly via end-to-end encrypted messaging applications. However, we have also observed another tactic with a few customers—the use of malicious advertisements on third-party platforms to fingerprint users and redirect targeted users to Intellexa's exploit delivery servers.

We believe this campaign is another example of commercial surveillance vendors abusing ads for exploit delivery, and Intellexa has gotten increasingly involved in this space since early 2025. Working with our partners, we identified the companies Intellexa created to infiltrate the advertising ecosystem, and those partners subsequently shut down the accounts from their platforms.

Addressing the Threat of Intellexa's Activities

Community efforts to raise awareness have built momentum toward an international policy response. Google has been a committed participant in the [Pall Mall Process](#), designed to build consensus and progress toward limiting the harms from the spyware industry. Together, we are focused on developing international norms and frameworks to limit the misuse of these powerful technologies and protect human rights around the world. These efforts are built on earlier governmental actions, including [steps taken](#) by the US Government to limit government use of spyware, and a [first-of-its-kind international](#) commitment to similar efforts.

Recognizing the severity and widespread nature of Intellexa's activities in particular, we have made the decision to simultaneously deliver our [government-backed attack warning](#) to all known targeted accounts associated with Intellexa's customers since 2023. This effort encompasses several hundred accounts across various countries, including Pakistan, Kazakhstan, Angola, Egypt, Uzbekistan, Saudi Arabia, and Tajikistan, ensuring that individuals at risk are made aware of these sophisticated threats.

Following our [disclosure policy](#), we are sharing our research to raise awareness and advance security across the ecosystem. We have also added all identified websites and domains to Safe Browsing to safeguard users from further exploitation. We urge users and organizations to apply patches quickly and keep software fully up-to-date for their protection. Google will remain focused on detecting, analyzing, and preventing zero-day exploitation as well as reporting vulnerabilities to vendors immediately upon discovery.

Indicators of Compromise (IOCs)

To assist the wider community in hunting and identifying activity outlined in this blog post, we have included IOCs in a [GTI Collection](#) for registered users.

File Indicators

- 85d8f504cadb55851a393a13a026f1833ed6db32cb07882415e029e709ae0750
- e3314bcd085bd547d9b977351ab72a8b83093c47a73eb5502db4b98e0db42cac

YARA Rule

This rule is intended to serve as a starting point for hunting efforts to identify PREYHUNTER malware; however, it may need adjustment over time.

```
rule G_Hunting_PREYHUNTER_IOSStrings_1 {
    meta:
        author = "Google Threat Intelligence Group (GTIG)"
    strings:
        $ = "/Users/gitlab_ci_2/builds/jb"
        $ = "/roe/ios1"
        $ = "-production/libs/Exploit" ascii wide
        $ = "/private/var/tmp/l/voip_%lu_%u_PART.m4a" ascii wide
        $ = "/private/var/tmp/ethereum.txt" ascii wide
        $ = "/private/var/tmp/kusama.txt" ascii wide
        $ = "_gadget_pacia" ascii wide
        $ = "ZN6Helper4Voip10setupHooksEvE3$_3" ascii wide
        $ = "Hook 1 triggered! location:" ascii wide
        $ = "KernelReaderI11CorelliumRWE" ascii wide
        $ = "NSTaskROP20WithoutDeveloperMode" ascii wide
        $ = "UMHookerI14RemoteTaskPort" ascii wide
        $ = "callFunc: building PAC cache for" ascii wide
        $ = "select tset FROM tsettings WHERE INSTR(tset, ?)" ascii wide
        $ = "select * from tsettings WHERE length(sha256) > ?" ascii wide
        $ = "isTrojanThreadERK" ascii wide
        $ = "getpid from victim returned:" ascii wide
        $ = "victim task kaddr:" ascii wide
    condition:
        1 of them
}
```

Acknowledgements

We would like to acknowledge and thank The Citizen Lab and Amnesty International for their collaboration and partnership.

Posted in

[Threat Intelligence](#)