Technical Research Report

Advanced iPhone Zero-Click Exploits and Spyware (2019–2025)

Author: Anurag Roy

Date: June 14, 2025

Audience: Law Enforcement Agencies & Cybersecurity Professionals

Table of Contents

Introduction 1
Evolving Zero-Click Attack Vectors 2
Pegasus (NSO Group) 3
Reign (QuaDream) 6
Predator (Cytrox/Intellexa) 8
Operation Triangulation 9
Exploit Techniques and Vulnerabilities 11
Mitigations and Defensive Measures 13
Conclusion 16

Technical Research Report: Advanced iPhone Zero-Click Exploits and Spyware (2019 - 2025)

Introduction

Over the past 34 years, Apple's iOS platform has been repeatedly targeted by **zero-click** exploits sophisticated attack chains requiring no user interaction to deploy spyware onto iPhones. Threat actors ranging from mercenary spyware vendors (e.g. NSO Group's **Pegasus** and QuaDream's **Reign**) to nation-state APT groups have leveraged vulnerabilities in iMessage, Safari/WebKit, the cellular **baseband**, and other components to silently compromise fully-patched iPhones. This report provides a comprehensive technical analysis of these exploits on recent iOS versions (roughly iOS 13 through iOS 16/17), including breakdowns of notable spyware frameworks (Pegasus, Reign, **Predator** by Cytrox/Intellexa, etc.), major attack vectors (iMessage, web browser, baseband, **Mobile Device Management (MDM)** abuse), and the underlying vulnerabilities (with **CVE references** and pseudocode). We also examine how attackers combined multiple zero-day bugs into exploit chains (often **remote code execution + kernel privilege escalation**) and how they bypassed iOS defenses like sandboxes, **Pointer Authentication Codes (PAC)**, and code signing. Diagrams of exploit flows and code snippets are included to illustrate the mechanics. Finally, we discuss mitigation measures from Apple's platform security enhancements (e.g. **BlastDoor** sandbox and **Lockdown Mode**) to forensic detection techniques relevant to cybersecurity researchers, incident responders, reverse engineers, and digital forensics experts in law enforcement.

Evolving Zero-Click Attack Vectors on iOS

Modern iPhones have a hardened security architecture, yet attackers have continually found novel entry points that require no user action. The main zero-click vectors exploited since 2019 include:

- **iMessage Attachments** The most notorious channel, wherein simply receiving a malicious text (often with a specially crafted attachment) can trigger code execution within Apple's messaging service. For example, NSO's Pegasus repeatedly used iMessage zero-clicks (20192021) to infect targets. iMessage's reach (default on all iPhones) and historically large **attack surface** (complex parsing of images, GIFs, videos, etc.) made it a prime target. Apple introduced the **"BlastDoor"** sandbox in iOS 14 (2021) to isolate iMessage parsing, but Pegasus found ways to bypass it. We detail these exploits (e.g. *KISMET* and

FORCEDENTRY) in subsequent sections.

- **WebKit/Safari** Attacks through the browser engine via malicious websites or web content. While not strictly "zero-click" if a user is lured to a link, some attacks achieve drive-by exploitation. Notably, a 2020 campaign dubbed **LightSpy** used a watering-hole site to auto-install spyware on iOS devices via a full WebKit RCE and kernel exploit chain. More recently, mercenary spyware vendor Intellexa (Cytrox) employed a **man-in-the-middle (MITM)** injection technique to silently redirect targets' web traffic to an exploit server, thereby not relying on any explicit click. In late 2023, this Intellexa campaign used a Safari/WebKit 0-day (CVE-2023-41993) as the initial trigger in an iOS 16 exploit chain.

- **Wireless/Cellular Baseband** The iPhone's baseband (modem firmware) handles cellular radio communications and has its own processing stack. A vulnerability here could be exploited by a rogue cellular station or malicious SMS/OTA message to compromise the device at a low level, *potentially without any user knowledge*. While public details of in-the-wild baseband exploits are scarce (they are extremely advanced and tightly held), researchers have long warned of this vector. Apple has patched baseband vulnerabilities in recent years (e.g. a baseband memory corruption in the new C1 modem, CVE-2025-31214, was fixed in 2025). There is no confirmed case of Pegasus or similar spyware using a baseband zero-click, but the possibility remains, especially for attackers with resources to mimic cell towers. One related attack was **SIMJacker** (2019), where an OTA SMS exploited SIM toolkit (not the baseband OS itself) to run code on SIM cards potentially affecting iPhones irrespective of iOS protections. Although not known to be used by Pegasus, it underscores the broader cellular attack surface in iPhones.

- **Apple Services & Other Vectors** Attackers have explored other zero-interaction vectors such as Apple's FaceTime and Push Notification systems. For instance, NSO's early exploits in 2016 used Apple's Push Notification Service (APNs) to invisibly deliver spyware (the "**KARMA**" exploit). In 2019, Pegasus famously exploited a zero-click **WhatsApp VoIP** bug (CVE-2019-3568) by simply calling the target via WhatsApp; Pegasus could infect the phone without answer. This wasn't an iOS flaw per se, but it affected iPhones via a third-party app vulnerability. It highlighted that popular messaging apps themselves (WhatsApp, or theoretically FaceTime) can be vectors. Apple's own services have had logic flaws (e.g. the 2019 FaceTime group-call bug that allowed eavesdropping), but those didn't enable code execution. Still, any service that auto-receives data from images in SMS to email content is a potential attack surface.

- **Mobile Device Management (MDM) Abuse** Although not a zero-*bug* exploit, attackers have abused enterprise features to surreptitiously install spyware. In a **highly targeted 20152019 campaign**, hackers tricked at least 13 iPhone users in India into installing a malicious MDM profile. By enrolling the device in a fake enterprise MDM, the attackers gained the ability to remotely push apps and configuration. They then installed trojanized apps (e.g. a tampered WhatsApp and Telegram) to surveil communications. Enrolling in MDM requires user action (installing a configuration profile), but social engineering or physical access was used to accomplish this. Similarly, spyware like **Hermit** (RCS Labs, 2022) was deployed by tricking users into installing an enterprise-signed app outside the App Store (often under the guise of a mobile carrier assistance app). Once the malicious profile or app is installed, the spyware has extensive device control. These cases underscore how **enterprise provisioning** can be abused as an attack vector when users (or their employers) are coerced or deceived. Apple responded by tightening enterprise certificate policies and, in iOS 16's Lockdown Mode, blocking configuration profile installation and MDM enrollment by default for at-risk users.

Each of the above vectors has been exploited by different threat actors through chains of vulnerabilities. Next, we delve into specific exploit frameworks that combine these vectors with multiple zero-day bugs to fully compromise iPhones.

Pegasus (NSO Group): Premier Zero-Click Spyware

Pegasus by Israel's NSO Group is the most infamous iPhone spyware, wielding a suite of evolving zero-click exploits. Active since at least 2016, Pegasus has continually adapted to new iOS versions and defenses. In the 2016 "Trident" incident, Pegasus used a one-click Safari/WebKit chain on iOS 9, but more recently it has favored zero-click iMessage exploits to silently penetrate devices of journalists, activists, and officials worldwide. Key Pegasus exploit chains in the last 4 years include:

- **KISMET (2020)** An invisible iMessage exploit used in iOS 13.5.1 (mid-2020) to hack dozens of Al Jazeera journalists. The payload arrived as a message that **required no tap**. Citizen Lab observed Kismet attacks from July to October 2020, and believed it was a zero-day at the time. While the exact bug was never publicly revealed, it appears to have been blocked by Apple's iOS 14 update and the introduction of BlastDoor in early 2021. Pegasus operators likely shifted to a new exploit after iOS 14 rendered Kismet ineffective.

- **FORCEDENTRY (2021)** Perhaps the most technically sophisticated mobile exploit ever seen, FORCEDENTRY was a zero-click iMessage attack discovered in use by Pegasus in February 2021. It targeted Apple's **CoreGraphics** image rendering library via a malicious PDF passed off as a GIF. Citizen Lab recovered the exploit from an infected phone and worked with Google Project Zero to analyze it. The core vulnerability (CVE-2021-30860) was an **integer overflow** in how **JBIG2**-encoded data in PDFs was handled by CoreGraphics. By sending a specially crafted iMessage with a fake GIF (actually a PDF containing JBIG2 streams), NSO could trigger this overflow in the IMTranscoderAgent process (which handled incoming iMessage images outside BlastDoor). The overflow led to a heap buffer under-allocation and overflow in the JBIG2Stream::readTextRegionSeg function, allowing memory corruption and initial code execution. Below is a simplified pseudocode of the vulnerable logic (reconstructed from analyses):

```
// Pseudocode illustrating the CVE-2021-30860 vulnerability
unsigned int numSyms = seg->getSize();   // could overflow to a small value
uint64_t alloc_size = numSyms * 8;
uint64_t *syms = malloc(alloc_size);     // allocate too small if overflow occurred
for (unsigned int i = 0; i < seg->getSize(); i++) {
    syms[i] = bitmap->getSymbol(i);      // writes beyond alloc_size for large seg size
}
```

NSO's exploit cleverly evaded iOS defenses: it ran in an unsandboxed service (IMTranscoderAgent) and even **disabled ASLR** by exploiting a flaw in how the dyld shared cache was mapped, so that memory addresses were predictable. It also **bypassed PAC** (Pointer Authentication Codes); Project Zero found that the exploit subverted a JBIG2Segment object's vtable pointer, something PAC is meant to protect indicating a separate logic flaw was used to bypass pointer signing. In fact, FORCEDENTRY unfolded in multiple stages: the JBIG2 overflow built a **"weird machine"** (a custom bytecode interpreter hidden in the exploit data) allowing the loading of a second-stage payload. That second stage was a **sandbox escape** implemented via only logic bugs in iOS (no memory corruption). Google's analysis showed it abused **NSPredicate**/NSExpression serialization to execute Objective-C methods and break out of the iMessage BlastDoor sandbox. Essentially, the exploit's JBIG2 stage delivered a blob that, when deserialized by the system (as an NSKeyedArchive), invoked forbidden APIs (using the FUNCTION syntax in NSPredicate to call selectors). This allowed actions like writing files or spawning processes outside the sandbox. Once on the device, Pegasus would download its full spyware payload. Apple patched CVE-2021-30860 in iOS 14.8 (Sept 13,

2021) and fixed the NSPredicate loopholes by iOS 15. Project Zero deemed FORCEDENTRY "one of the most technically sophisticated exploits we've ever seen" a testament to the level of Pegasus's capabilities.

Forensic artifacts: Despite Pegasus's stealth, investigators found traces. Citizen Lab identified a distinctive Pegasus footprint called **CASCADEFAIL** in device logs, a peculiar partial deletion in iOS data usage databases caused by Pegasus failing to clean up properly. They also noted Pegasus processes with names like "setframed" on victim devices. Amnesty International developed the Mobile Verification Toolkit (MVT) to scan iPhone backups for such indicators (e.g. unusual crash logs, process names, or domain names associated with Pegasus infrastructure). In late 2021, Apple itself sent out notifications to victims of Pegasus, based on unknown detection methods (likely Apple's cloud-side logs of exploit messages).

- **NSO's Continued Exploits (20222023)** Pegasus did not stop with FORCEDENTRY. In 2022 and 2023, NSO reportedly deployed new zero-click chains as Apple locked down iMessage. One such chain, dubbed **"PWNYOURHOME"** by researchers, may have involved vulnerabilities in HomeKit or iCloud Photos (unconfirmed publicly). However, a confirmed case came in 2023 with the **BLASTPASS** exploit discovered by Citizen Lab. BLASTPASS was an NSO zero-click chain that exploited a flaw in how iOS handled PassKit (Wallet) attachments containing images. It consisted of **CVE-2023-41064**, an ImageIO buffer overflow when processing a malicious image, and **CVE-2023-41061**, a vulnerability in Wallet that allowed code execution with elevated privileges. By sending an iMessage with a malicious image (possibly disguised as a Wallet pass invitation), Pegasus could run code on iOS 16 without user action. Apple rushed out iOS 16.6.1 in September 2023 to patch these, noting they were actively exploited to deliver Pegasus spyware. As of September 2023, reports indicated Pegasus operators could compromise fully updated iPhones up to iOS 16.6 using zero-click exploits. In short, NSO has continually found or bought new exploits to keep Pegasus effective against even the latest iOS versions.

Pegasus Payload Capabilities: Once Pegasus spyware is installed, it thoroughly compromises the device. It can access messages, call logs, contacts, microphone, camera, GPS location, and even encrypted chats (by reading the screen or keyboard inputs). It operates with kernel or root privileges, allowing it to evade detection and communicate stealthily with its servers. Pegasus can also **persist** in various ways: on some iOS versions it reportedly used the **"ReSpring"** trick (restarting userland to stay running), and in one case Cytrox's Predator (akin to Pegasus) abused the iOS **Shortcuts/Automation** feature to relaunch on reboot. However, Pegasus often avoids persistence; it resides mainly in memory to reduce forensic footprint, re-infecting as needed via zero-click. This

makes live memory analysis or network monitoring crucial for detection, as filesystem traces may be sparse.

Reign (QuaDream): The ENDOFDAYS Exploit and "KingsPawn" Spyware

QuaDream is an Israeli spyware vendor, often viewed as NSO's quieter counterpart. QuaDream's iOS spyware, named **Reign**, came to light in 2023 along with its exploit **ENDOFDAYS**. Active around the same period as NSO's FORCEDENTRY, ENDOFDAYS was a **zero-click** exploit against iOS 14 (specifically 14.4 and 14.4.2 in early 2021). Instead of iMessage, it leveraged **invisible iCloud calendar invitations** to infect devices. The attack flow was as follows:

1. The operator quietly added a calendar event *invite* for the target, using the target's Apple ID/email. These invites were crafted maliciously, containing exploit data in the event fields. Because they were set in the future and marked invisible, the target user saw no notification.

2. Apple's iCloud Calendar service would send a push notification (APNs with topic com.apple.cal) to the target device about the new event. The device's dataaccessd process would automatically fetch the event details via CalDAV (a web calendar protocol). The server's response embedded the event data inside an XML <calendar-data> field wrapped in <![CDATA[...]]>.

3. QuaDream's exploit took advantage of this by inserting *malicious XML content after a fake CDATA close tag* inside the event. Citizen Lab found that the .ics invite files contained atypical sequences like]]>:x and extra CDATA tags in fields. This suggests an **XML injection**: the attacker prematurely closed the CDATA, then injected additional XML that would be parsed by the iPhone's calendar processing logic. In theory, this could trigger a vulnerability in how iOS parses Calendar data (for instance, an XML parser bug or logic flaw in event handling).

4. The end result was code execution on the device, loading the QuaDream spyware (which Microsoft calls **KingsPawn**). Apple later notified some victims in November 2021 that they had been targeted by *both* NSO and QuaDream exploits, confirming ENDOFDAYS was active alongside FORCEDENTRY.

Reign's Payload: The KingsPawn malware delivered by ENDOFDAYS is a full-featured implant. Citizen Lab and Microsoft obtained samples and noted advanced capabilities and anti-forensic measures. It runs with kernel-level privileges (having likely exploited the kernel as part of the chain) and can perform tasks like: recording audio, taking pictures, exfiltrating messages and files, and tracking location. Interesting modules in KingsPawn were specifically designed to *delete evidence* of the zero-click exploit from the device. For example, **Calendar Cleanup Functions** in the malware would delete any calendar events associated with the attacker's email address (used in the invite) and even remove that email from the device's cache of recent iCloud IDs. We see the malware executing SQL deletion queries on the calendar database to purge records of the malicious event and then vacuuming the DB. It also deleted entries from a plist that stored iCloud identity usage, to cover tracks. These "anti-forensic" routines indicate a strong concern for stealth.

Despite these efforts, Reign infections left some clues. Citizen Lab reported an "*Ectoplasm Factor*" , a trace that sometimes remains even after the spyware tries to remove itself. They chose not to disclose details, as it could help future detection. Additionally, the malware created a unique directory (/private/var/db/com.apple.xpc.roleaccountd.staging/PlugIns/fud.appex/) when spawning processes. This unusual path (abusing Apple's roleaccountd service) is a forensic indicator of QuaDream activity. Apple patched iOS 14 vulnerabilities used by ENDOFDAYS (exact CVEs not explicitly named by Apple, likely fixed by iOS 14.5/14.6). Once publicly exposed, QuaDream reportedly shut down operations in 2023, but the case underscored that Pegasus was not the only game in town; multiple firms were selling comparable iPhone zero-click exploits.

Predator (Cytrox/Intellexa): Multi-Platform Spyware and Hybrid Exploits

Predator is a spyware tool developed by North Macedonian company **Cytrox**, now part of the Intellexa alliance (an EU-based spyware consortium). Predator came onto the scene around 2021, used by government clients in Egypt, Greece, Armenia, Madagascar and others. Predator's iPhone exploits have involved both one-click and zero-click approaches:

- **Single-Click Links (2021):** In June 2021, Predator was used to infect an iPhone running iOS 14.6 by sending a WhatsApp message with a malicious link. The target (Egyptian politician Ayman Nour) clicked the link, which likely triggered a WebKit exploit or a profile download. Predator was able to jailbreak iOS 14.6 (then the latest) using this exploit chain, indicating it was a zero-day. The Citizen Lab investigation found that the same phone was *simultaneously* infected with Pegasus and Predator by different operators, a rare case of dueling spyware. Once Predator got in, it established persistence: it installed an iOS *Automation* (via the

Shortcuts app) to run itself after reboot. Using a user automation for persistence is clever, as it leverages an iOS feature rather than a kernel exploit for reboot persistence. Citizen Lab was able to obtain Predator's loader binary and noted it drops additional stages and uses elaborate techniques to hide its processes and data.

- **Zero-Click MITM Injection (2023):** In a more sophisticated turn, Google's Threat Analysis Group (TAG) reported in 2023 that Intellexa (Predator's parent entity) employed a **network injection** method to deliver Predator in Egypt. Attackers with ISP-level or local network access injected an HTML/JavaScript exploit when the target attempted to visit any HTTP website (unencrypted). This means the victim did not have to click a specific malicious link any web traffic could be hijacked opportunistically. The injected exploit was a chain of **three** iOS 0-days that were unknown at the time, all fixed later in iOS 17.0.1. These were:

    - *CVE-2023-41993:* a Safari/WebKit remote code execution (likely a JS engine bug). This gave initial code execution in the Safari context.

    - *CVE-2023-41991:* a vulnerability in certificate validation that allowed bypassing code signing or PAC (Pointer Authentication) constraints. This was an unusual bug Apple described as a "validation issue" in X.509 certificates, which TAG noted was used to bypass PAC, perhaps by tricking the system's code signature checks. By exploiting this, the attackers could likely execute unsigned or modified code in a privileged context.

    - *CVE-2023-41992:* a kernel privilege escalation flaw. With this, the attackers escaped the Safari sandbox and achieved kernel code execution.

- After the chain, a small stub checked if the device was indeed the intended target (Predator attacks were highly targeted) and only then downloaded the full Predator spyware. This elaborate infection chain from web injection to multi-vuln exploit shows the level of sophistication Predator had achieved, rivaling Pegasus. Apple credited this discovery to Google and Amnesty and patched all three CVEs in September 2023. TAG pointed out that this approach (MITM injection) is "similar to zero-click" in effect the victim does nothing, and the exploit is injected invisibly highlighting that surveillance vendors are diversifying their delivery methods beyond messaging apps.

Predator's Features: Predator provides comparable capabilities to Pegasus: recording calls, activating camera/mic, and stealing messages from apps like Signal (likely by screen reading or keylogging). It also has modules to **evade detection**. For instance, Predator knows to stop or remove itself if certain conditions are met (possibly if on an unsupported iOS version or if a forensic tool is detected). Cisco Talos reported Predator's Android version can survive device reboots and even detect SIM changes, implying similar resiliency on iOS. Intellexa, Predator's vendor, aggressively markets "zero-click infection" and "GPS tracker" capabilities, though it faced legal troubles and export bans recently. In late 2023, following these disclosures, the EU and US scrutinized Intellexa.

Note: Another spyware, **Hermit** (by Italy's RCS Labs), targeted iPhones in 202122 by getting users to install a malicious configuration (often via a carrier-branded SMS asking to install a new settings app). Apple blocked the certificates used by Hermit. While Hermit didn't use zero-click exploits (it was more social engineering), it underscores that multiple commercial spywares exist in the ecosystem.

Operation "Triangulation": APT-Level Zero-Click Attack Chain

In 2023, Kaspersky Labs revealed an extraordinarily complex zero-click iMessage exploit *not* attributed to a commercial vendor but likely a state-backed APT. Termed **"Operation Triangulation,"** it targeted iOS 15 and earlier (with some components up to iOS 16.2) and was used against Kaspersky's own researchers and others (notably including Russian entities). This attack stands out for its use of *four* 0-days and deep exploitation of iOS internals. Here is a summary of the Triangulation attack chain (largely confirmed by Kaspersky's public report and 37C3 presentation):

- **Initial Vector:** A malicious iMessage was sent to targets, with an attachment that automatically triggered on receipt. The user sees nothing; the message may even be deleted automatically post-exploit.

- **Stage 1 Font Vulnerability (CVE-2023-41990):** The attachment exploited a vulnerability in an **undocumented** Apple TrueType font instruction called "ADJUST". This legacy feature (present since the 90s) had an RCE flaw that allowed the attacker's font data to execute code in the context of iMessage. Essentially, the font would overflow or manipulate memory during text rendering. Apple removed this instruction entirely once patched.

- **Stage 2 NSExpression and JIT Logic:** The shellcode from Stage 1 set up a *multi-phase loader* in memory. It used a two-pronged strategy: first, employing **Return-Oriented**

Programming (ROP) and **Jump-Oriented Programming** to bootstrap execution in a limited environment, then leveraging **NSPredicate/NSExpression** tricks (similar to FORCEDENTRY's sandbox escape) to execute higher-level payloads. The attackers actually embedded a **JIT-compiled JavaScript** engine exploit *written in JavaScript itself*. They modified the JavaScriptCore environment via Objective-C runtime calls (enabled by NSPredicate code injection) to run a privilege escalation exploit *in JS*. This JavaScript payload was huge (~11,000 lines), heavily obfuscated, and dedicated to parsing and manipulating memory structures (both JavaScriptCore and kernel). It exploited the $vm internal debugging interface of JavaScriptCore to get low-level memory access from JavaScript. Through that, it achieved *kernel read/write*.

- **Stage 3 Kernel Exploits (CVE-2023-32434 & CVE-2023-38606):** Armed with arbitrary memory access, the attackers next exploited an *integer overflow in XNU's kernel memory mapping (mach_vm) calls* CVE-2023-32434 to gain access to physical memory from user space. This essentially broke the user-kernel boundary. They also discovered a secret hardware register interface to bypass the **Page Protection Layer (PPL)** , a security feature that protects kernel memory which was fixed as CVE-2023-38606. By writing to certain memory-mapped I/O registers (likely meant for Apple's hypervisor or debugging), their code disabled PPL, undermining even the CPU-level protections. This is extraordinary: it was an **undocumented hardware backdoor** which the attackers found and weaponized to make kernel memory fully writable despite Apple's strongest defenses. With PPL bypassed and kernel memory RW, the attackers could insert their own code and defeat PAC protections as well.

- **Stage 4 Implant Delivery:** After kernel takeover, the malware launched additional processes. It spawned an IMAgent instance (iMessage agent) to **erase evidence** of the attack likely deleting the exploit attachment and associated logs. It then started an **invisible Safari** instance (no GUI) which connected to a hardcoded URL for the next stage. The purpose was twofold: (a) performing an additional victim verification (the server would only respond if the device was of interest), and (b) delivering a **Safari/WebKit exploit (CVE-2023-32435)** to run a **Mach-O shellcode** in the context of Safari. This Mach-O shellcode, in turn, executed a *second* kernel exploit using the same CVE-2023-32434 and CVE-2023-38606 primitives (perhaps to re-establish kernel control in the Safari process, or to bypass new mitigations). It was modular and contained various post-exploitation tools (such as file search, network comms), some of which were unused.

- **Stage 5 Spyware Installation:** With root/kernel privileges obtained again, the final stage was to deploy the actual spyware implant (Kaspersky dubbed it *TriangleDB*). This is a fully-featured backdoor running as root, with commands to collect data, execute arbitrary code, and communicate over TLS to the C2. Interestingly, the implant in Triangulation did **not persist** on disk, it ran in memory only. Kaspersky noted it lacked persistence and infected devices would reinfect after reboot via the same initial iMessage (the attackers would presumably resend the message). The implant even *blocked iOS updates* by modifying system settings, likely to keep the device vulnerable (some victims saw "Software Update Failed" repeatedly).

Operation Triangulation is arguably on par with Pegasus's exploits in complexity if not exceeding them. Kaspersky called it "the most sophisticated attack chain we have seen". It combined a soup-to-nuts approach: from a novel font bug to logic bomb payloads, JS JIT exploits, kernel 0-days, and hardware tricks. Apple patched the vulnerabilities (41990, 32434, 32435, 38606) in iOS 16.3 to 16.6 during 2023 once alerted. Detecting Triangulation was challenging, but Kaspersky's researchers found clues by analyzing device backups for example, the presence of a mysterious process BackupAgent using cellular data right after an IMTransferAgent event, and modified system plist files with timestamps corresponding to the attack. These artifacts, along with domains the implant communicated with (e.g. tagclick-cdn.com, snoweeanalytics.com, etc. found in DNS logs), allowed for retrospective detection. Kaspersky released a utility and indicators to help others uncover this malware.

In-Depth: Exploit Techniques and Vulnerabilities

Across the above cases, several recurring technical patterns emerge. Here we highlight some in-depth aspects:

Image Parsing Vulnerabilities and "Fake Gif" Trick

Many iOS zero-clicks exploit the complexity of image and multimedia formats. Pegasus's FORCEDENTRY is a prime example: NSO took advantage of an obscure image codec (**JBIG2**) embedded in PDF and the fact that iMessage would automatically process images. NSO also discovered they could send an attachment with a .gif extension that was **not actually a GIF** iMessage's pipeline would still attempt to render it by *guessing the format* (using ImageIO). This is the "fake GIF" trick: by naming the file .gif but embedding a PDF, they expanded the reachable code to include PDF parsers (and thus JBIG2). Apple later closed this hole as of iOS 15, GIFs in iMessage

are processed only within BlastDoor and strictly as GIFs. But prior to that, a single mislabeled image could force iMessage to wade through numerous complex codecs in unmanaged C/C++ code, ripe for memory corruption.

Similarly, Triangulation's first exploit was in a font (TrueType) essentially treating font data as code. Attackers love such bugs because image and font rendering often run by default when a message or webpage loads, with no clicks. Apple's large codebase for media (ImageIO, PDFKit, FontParser, etc.) has yielded multiple 0-days. Another example: **CVE-2023-32439** (patched in 2023) was a WebP image processing bug that NSO reportedly used on iOS and macOS (WebP is an image format common in browsers). The lesson: media formats are an expansive attack surface. Apple has hardened this by isolating many parsers in the BlastDoor service and introducing run-time mitigations (like disabling JIT for WebKit in Lockdown Mode). But as long as iPhones auto-render content, attackers will seek vulnerabilities in those decoders.

Kernel Exploits and PAC Bypasses

Gaining initial code execution (say in an IMTranscoderAgent or Safari process) is only half the chain; the spyware still needs to break out of the sandbox and gain **kernel (root) privileges** to fully control the device. Every one of the discussed chains included at least one **kernel vulnerability** for privilege escalation: e.g., Pegasus used a memory corruption in the iOS kernel (an **IOMobileFrameBuffer** heap overflow, CVE-2021-30807, in mid-2021) to escape the sandbox and install its agent. Predator's 2023 chain had CVE-2023-41992 (a kernel bug), and Triangulation had CVE-2023-32434 (kernel mapping bug) plus CVE-2023-38606 (PPL bypass).

Modern iPhone kernels are fortified with **Pointer Authentication Codes (PAC)** cryptographic signatures on pointers that make arbitrary code execution much harder, as an attacker can't easily forge valid function pointers. Yet, as we saw, attackers found ways around PAC. Project Zero noted that in FORCEDENTRY's JBIG2 exploit, the vtable pointer had been tampered with in memory and PAC did not stop it. How? The likely answer: the attackers first achieved a controlled **memory read/write** (e.g., via an info leak or overflow) and could locate a signed pointer, then reused that pointer (or its authentication context) in a way to "swap in" their own code. Additionally, some exploits like the Intellexa chain exploited a logic bug (CVE-2023-41991) to bypass code signing/PAC altogether possibly by tricking the system to treat malicious code as if it were signed by Apple (e.g., by exploiting certificate validation in dyld or AMFI). Also, Triangulation's method of disabling PPL essentially nullified PAC's added layer, because PPL normally prevents even a kernel bug from modifying core kernel code once PPL was gone, PAC protections on critical structures could likely be bypassed by brute force or reuse of existing signed pointers.

In summary, the exploit developers combined *memory corruption* with *logic flaws* to defeat iOS kernel defenses. They chain a memory bug to get a foothold, then a logic bug to escalate privileges in a stealthy way. It's a cat-and-mouse game: Apple introduced PAC in A12 chips; Pegasus responded with exploits that don't rely on forging pointers (like the NSPredicate approach, a logic bomb needs no fake pointer). Apple introduced BlastDoor; NSO created a JBIG2 "weird machine" to jump out of BlastDoor's sandbox. Apple added PPL in iOS 16; Triangulation found an internal switch to turn it off. This illustrates the high-end exploit development ethos: **find the weakest link**, whether in code or design, and exploit it to undermine stronger security features.

MDM and Configuration Exploits

Although not as technically fancy as a kernel 0-day, the abuse of Apple's configuration and enterprise deployment system is worth a deeper note. The 2015-2018 MDM attack in India succeeded because once an attacker enrolls a device in *their* MDM, the iPhone will obediently install any app or setting the MDM server dictates (with almost no user prompts after enrollment). In that campaign, the attackers trojanized legit apps by adding dynamic libraries. They took WhatsApp and Telegram IPA files, added a dylib that could read messages and send them to C2, then pushed those apps as "updates" via MDM. The iPhones, thinking their enterprise MDM owner is pushing authorized apps, installed these spyware-laced versions. Similarly, **Hermit** in 2022 leveraged Apple's Ad Hoc distribution: victims received an SMS claiming to be from their telecom, with a link to "download a security app." The app was in fact signed with an enterprise certificate tied to an Apple Business account that the spyware company controlled. Because the victims approved the install (likely after some social engineering about a service outage or COVID tracking app), the app installed and then used exploits or excessive permissions to spy on the device.

Apple has since made such social engineering harder: it cracked down on rogue enterprise certificates and now requires explicit trust steps for profiles, with warnings. **Lockdown Mode** (introduced in iOS 16) outright blocks profile installations and enterprise app installs, closing this vector for high-risk users. MDM remains a powerful tool in enterprise settings it's necessary, but outside that, users should almost never enroll in an unknown MDM. Any unexpected prompt to install a configuration profile should raise alarms.

Mitigations and Defensive Measures

Apple's Platform Security Responses

Apple has steadily fortified iOS in response to these spyware threats:

- **BlastDoor Sandbox:** Debuted in iOS 14 (2021) for iMessage, BlastDoor is a sandboxed service that handles most message parsing (images, links) in isolation. It limited what zero-clicks like Kismet could do (indeed Kismet ceased to work on iOS 14). FORCEDENTRY had to specifically target a component *outside* BlastDoor (IMTranscoderAgent). Apple has since moved even that GIF handling inside BlastDoor or removed the functionality. By iOS FiforcedPotentially, this neuters entire classes of iMessage attacks but as shown, attackers pivoted to other services or found sandbox escapes.

- **Pointer Authentication (PAC) & Memory Tagging:** Hardware features in newer iPhones (A12+ for PAC, M2 for MTE) help prevent code reuse and heap exploits. PAC adds cryptographic signatures to function pointers and return addresses, foiling many buffer overflow techniques. Memory Tagging (MTE) in latest devices can detect use-after-free and overflow at runtime by tagging memory allocations. While these make exploitation *much* harder, the likes of Pegasus and Triangulation show they are not foolproof logic bugs or novel techniques (like JIT exploitation, or reusing valid pointers) can bypass them. Nonetheless, PAC likely stopped numerous lesser attackers; only top-tier exploits manage to bypass it, often by chaining multiple issues.

- **Lockdown Mode:** Apple's boldest mitigation for at-risk users, introduced in iOS 16 (2022), Lockdown Mode disables or restricts many features that spyware has abused. This includes blocking most message attachment types (no previews for images or links), disabling JIT in Safari (reducing web exploit reliability), removing custom fonts, and preventing configuration profiles or MDM enrollment. Essentially, it shrinks the attack surface dramatically. Citizen Lab confirmed that Lockdown would have mitigated FORCEDENTRY and other iMessage exploits (the BlastDoor bypass would be harder with Lockdown's stricter policies). However, Lockdown Mode does come at a usability cost (e.g., no iMessage link previews), so it's intended for those who suspect they are targeted by mercenary spyware or nation-state hackers. Apple also began offering **Rapid Security Responses** quick out-of-band patches to accelerate fix delivery for exploits like BLASTPASS.

- **Vulnerability Bounties & Lawsuits:** Apple launched a high-tier bug bounty program (with up to $1 million rewards for zero-click exploit chains) to incentivize researchers to report bugs to Apple rather than sell to companies like NSO. While take-up has been mixed, it has led to some discoveries. Separately, Apple took the unprecedented step of suing NSO Group in 2021 and obtaining injunctions to bar it from targeting Apple devices. While legal action doesn't stop exploits, it increases pressure and has led to NSO reportedly struggling with finances and zero-day supply. Apple's notifications to victims and public attribution of these

attacks also help by removing the cloak of total secrecy that offensive actors rely on.

Detection and Forensic Analysis

Detecting a zero-click spyware infection is extremely challenging by design, these tools leave minimal traces. However, **indicators of compromise (IoCs)** can be found with careful forensic methods:

- **Backup and Logs Analysis:** As demonstrated by Amnesty and Kaspersky, analyzing an iTunes or Finder backup of the iPhone can reveal traces in system databases and logs. For example, unusual entries in the DataUsage.sqlite (recording data transfers by process) have given spies away. The Pegasus CASCADEFAIL indicator is one leftover DB entries from a partially deleted process record. In Operation Triangulation, the presence of BackupAgent using cellular data and certain plist files being touched at odd times signaled the exploit's execution. Apple doesn't provide a full filesystem view (without jailbreak), but backup data includes many app sandboxes and system preference files that can hold clues.

- **Mobile Verification Toolkit (MVT):** Amnesty International's open-source tool **MVT** encodes much of the forensic knowledge about Pegasus/Predator. It scans backups for known indicators: suspicious SMS messages, domain names of known spyware servers in system caches, weird process names, etc. For instance, it will flag if it finds domains like those listed in Pegasus infrastructure (e.g. the service.ac domains identified in 2021) or remnants of crash logs pointing to ImageIO or IMTranscoderAgent crashes (which might indicate exploit attempts). MVT was crucial in the Pegasus Project revelations, finding dozens of infected phones by these artifacts.

- **Network Monitoring:** Although spyware tries to hide its traffic (e.g., using HTTPS with its own certificate or routing via legitimate cloud services), there are often patterns. In Pegasus infections, phones frequently contact certain uncommon domains or IPs at odd intervals. Triangulation's implant, for example, communicated with C2 domains that, once known, could be searched in DNS logs. A defender monitoring an enterprise network might spot a device beacons to an unusual domain like businessvideonews.com (one of the Triangulation IoCs). Law enforcement or researchers can also perform **memory forensics** if they can obtain a live memory dump (on jailbroken or exploited devices) in one case, Citizen Lab pulled Pegasus artifacts from memory that were not on disk.

- **Apple Threat Notifications:** Since late 2021, Apple has taken to notifying individuals suspected of being targeted by state-sponsored spyware. They haven't published how they detect this (likely a combination of detecting anomaly patterns in iCloud or APNs usage). For example, Apple notified Ugandan diplomats in 2021 about Pegasus, and Thai activists in 2022 about Pegasus/QuaDream. These notifications are a valuable source if you get one, it's a strong sign your device was targeted or compromised. Apple encourages those users to enable Lockdown Mode and contact Apple/security researchers for further analysis.

- **Endpoint Security Solutions:** Traditionally, iOS has been a black box to security software (antivirus can't run with privileges on an iPhone). This is still largely true, but specialized mobile threat defense (MTD) apps can detect certain behaviors e.g., a process running that shouldn't, or excessive battery/network usage. Kaspersky researchers noted that on macOS (which is more open) one can install EDR (endpoint detection & response) agents to catch these attacks in real-time, but on iOS the best bet is network and backup forensics. Rebooting iPhones daily is even suggested by some experts (Triangulation's lack of persistence means a reboot could evict it, though that's a band-aid, especially if the zero-click can just re-trigger). The consensus for now: high-risk users should use Lockdown Mode, keep iOS updated immediately, and treat unexpected messages or prompts with extreme suspicion. Organizations may also route device network traffic through VPNs with logging to detect anomalies outside the device.

Conclusion

In the cat-and-mouse game of iOS security, the last few years have demonstrated both the incredible sophistication of attackers and the significant strides in defense. We saw Pegasus and its peers wield chains of 0-days that string together image decoding bugs, logic flaws, and kernel exploits to achieve the impossible remote takeover of an iPhone without a single tap by the user. The technical breakdowns of **FORCEDENTRY** and **ENDOFDAYS**, the mercenary exploits used by Pegasus and Reign, reveal an astonishing depth of exploit craft, from hidden **JBIG2 bytecode interpreters** to calendar invite injection. Equally, the Operation Triangulation case shows that nation-states can parallel or even exceed mercenary toolkits, discovering hardware-level tricks and esoteric font bugs to penetrate targets.

Apple's responses BlastDoor, Lockdown Mode, and rapid patching have meaningfully raised the bar. Yet the cat adapts: NSO found non-BlastDoor pathways and Triangulation exploited parts of the system that security features didn't cover. The existence of multiple players (NSO, QuaDream,

Cytrox, Candiru, etc.) means defenders face an agile and resourceful adversary community. Each time Apple fixes a flaw, attackers pivot to a new vector or refine a different bug, underscoring the need for continual security research and patching.

For cybersecurity researchers and law enforcement forensics teams, the battle entails developing novel detection methods from scrutinizing backup artifacts to sharing threat intelligence on spyware infrastructure. Collaboration has been key: Citizen Lab, Amnesty, Google TAG/P0, and others have coordinated disclosure and shone light on these "zero-click" dark arts. As a result, many of these exploits, once clandestine, are now in security advisories and CVE databases (as cited throughout this report). This transparency helps demystify how iPhones have been infiltrated and arms defenders with the knowledge to harden their systems.

Moving forward, mitigations like broader sandboxing of parsing code, more frequent security updates (Apple's new bi-weekly Rapid Security Response mechanism), and even perhaps optional disabling of apps (e.g., some activists now use separate devices or disable iMessage entirely) are likely to continue. Meanwhile, law enforcement and incident responders must stay updated on the latest exploits. Today's iOS 17 or 18 could have new entry points and thus new Pegasus-like exploits. The overarching takeaway is that **no system is impervious**, but with rigorous technical analysis and swift defensive action, the window of exposure can be narrowed.

In summary, the period 20192025 has been a "golden age" of spyware from a technical standpoint, forcing unprecedented innovations in both offense and defense on iOS. By studying these exploit chains in depth as we have in this report the security community can better anticipate the next moves in this arms race and strive to protect users, especially those most at risk of these highly advanced threats.

Sources: This report drew on technical analyses and official advisories from Citizen Lab, Google Project Zero, Amnesty International, Microsoft, Kaspersky, Apple security bulletins, and others as cited inline. All code/CVE references correspond to publicly documented vulnerabilities and exploits, as indicated by the citations.