Episode Book

EPISODE #6391

How Forensic Tools Search Android for Spyware

PUBLISHED	31 October 2026 at 19:21 UTC
DURATION	25:07
LISTEN	myweirdprompts.com/episode/android-spyware-forensics-mvt-73057/
HOSTS	Corn, Herman, Neville
SCRIPT MODEL	anthropic/claude-sonnet-5.5
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	14 cited by the research (listed at the end)
BOOK GENERATED	1 November 2026 at 07:41 UTC

SYNOPSIS

Android forensics for detecting mercenary spyware like Pegasus has long relied on incidental logs that quickly overwrite themselves. Examining Amnesty International's Mobile Verification Toolkit and AndroidQF reveals why direct shell extraction was replaced with a custom pushed collector binary, why backups reveal almost nothing on modern Android, and how the introduction of Android Intrusion Logging in Advanced Protection Mode provides the first deliberate, timestamped event source for civil society digital investigators.

TRANSCRIPT



Corn

Plug the phone into a laptop, press a button, get a green tick or a red cross. That's the fantasy. Daniel wants to know how the real forensic tools check an Android phone for spyware, and the answer is a lot less tidy than that.



Corn

His question is about MVT, Amnesty's Mobile Verification Toolkit, and its companion AndroidQF. How useful is pulling data over a USB cable with debugging switched on, compared with bugreports, backups, or collecting on the phone itself? And then the consumer side: can a Play Store spyware scanner, sitting inside Android's sandbox, catch Pegasus or Predator at all?



Herman

Herman Poppleberry, reporting for duty. And the answer starts with a surprise. The authors of MVT deleted the method everyone assumes it's built on.



Corn

Before that, what are these tools, and who are they for?



Herman

MVT came out in July 2021, alongside the Pegasus Project. It's a collection of utilities for gathering forensic traces on Android and iOS, and the README is blunt that it is not intended for end-user self-assessment. You need command-line skills and some forensics knowledge.



Herman

AndroidQF is the acquisition half. It's a standalone Go binary for Linux, Windows and macOS, with a Docker image too. You connect a phone with USB debugging on, it pulls the data and writes a zip, and MVT analyses the zip. Amnesty says it's designed so non-technical people can run it.



Corn

So a person in the field runs one tool and an analyst somewhere else runs the other.



Herman

Yes. That split matters in practice. A human rights worker in one country can run the acquisition in ten minutes, send the zip to a lab in another country, and the analyst never has to touch the phone. And the reason the whole setup is awkward is Amnesty's own words: Android provides fewer complete forensically useful data sources than its iOS cousins.



Corn

Fewer. That's a polite word for it.



Herman

It's very Amnesty. Anyway, the ADB thing.



Corn

Go.



Herman

MVT's docs have a page titled "ADB Analysis Removed." Direct analysis of an Android phone over the Android Debug Bridge is gone. There's even a test file in the repo called test_check_android_adb_removed, which checks that the old command now refuses to work.



Corn

They wrote a test to confirm the feature is dead. That's commitment.



Herman

Four stated reasons. Inconsistent data collection across devices. Incomplete acquisition, where critical artifacts like full bug reports could be missing. Duplicated analysis paths across the ADB, AndroidQF and bugreport modules. And workflow consistency. The methodology page says direct extraction was error-prone and frequently gave different results from an AndroidQF acquisition of the same phone.



Corn

Hang on. AndroidQF needs USB debugging and a cable. That is ADB. So what actually got removed?



Herman

Good catch, and it's the whole answer to Daniel's question. ADB is still the transport. What changed is what runs on the phone. The old MVT approach ran shell commands through adb shell and scraped the output. But what you get depends on the shell utilities on that particular device, and those vary wildly between manufacturers and Android versions. A command exists on one phone, behaves differently on another, and is missing on a third.



Corn

So you're asking a stranger's kitchen to cook the same meal every time.



Herman

Roughly. Same recipe, but one kitchen has no oven and another labels the dials in a different language. AndroidQF instead pushes its own versioned collector binary onto the phone and runs that. It's like bringing your own portable stove. Same code, same output, whatever the vendor did to the shell. Amnesty lists the advantages as complete and raw data extraction, consistent and standardised output, future-proof analysis, and no dependencies on any platform. It even bundles its own copy of ADB.



Corn

Why does bundling ADB matter?



Herman

Because the version of ADB installed on a random laptop is another variable. Old versions misbehave with new phones. If the tool carries its own copy, that's one less thing that can go differently between two acquisitions.



Corn

And why does consistency matter so much? If I get slightly different output each time, can't an analyst just work around it?



Herman

Not easily. Say you're comparing two phones from the same newsroom, one Samsung and one Motorola, because you suspect the same attacker hit both. If the output formats differ because of the shell, you can't tell whether a difference is a real finding or just an artefact of the tool. Standardised output means a difference in the data is a difference in the phones.



Corn

Fine. That I buy. What does the collector actually grab?



Herman

Quite a lot, and it's worth being concrete. Build info from getprop. Every system setting. The process list and running services. Logcat and the logs folder. A big dumpsys dump. The list of installed packages, with optional copies of the APKs themselves. Android's APEX containers. A full file listing, optionally with hashes computed on the phone, in MD5, SHA-1, SHA-256 or SHA-512. A bugreport zip, plus any older bugreports sitting in the bugreports folder. If the phone is rooted, it also looks at Magisk module metadata and init scripts.



Corn

And you know it worked because...?



Herman

Every acquisition includes an acquisition file with per-module status, completed, partial or failed. A hashes file with the SHA-256 of every entry in the archive. And a command log. So an analyst can see what was attempted and what silently didn't happen.



Corn

Why is the silent part so important?



Herman

Because an empty folder is ambiguous. It could mean nothing suspicious was there, or it could mean the collection quietly failed. If the status file says that module failed, the analyst knows not to read the emptiness as good news.



Corn

And the hashes file? What does that add?



Herman

Integrity. If the zip is emailed across a border or sits on a shared drive, the analyst can verify nothing was altered or corrupted along the way. For anything that might end up as evidence, that chain matters.



Herman

Herman's clinician instinct likes that. A system that tells you what it couldn't see is more trustworthy than one that reports itself fine.



Corn

You just said your own name in the third person.



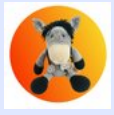
Herman

I did not.



Corn

You did. Carry on.



Herman

Moving on. The design detail I find striking: you can encrypt the acquisition with an age public key. The field user has the tool but not the private key, so they cannot decrypt their own data.



Corn

Which only matters if someone is going to make them.



Herman

Right. The tool assumes the person carrying the phone might be forced to hand over everything, including the archive. If they can't open it, there's nothing to give up. Picture a journalist stopped at a border crossing with the zip sitting on their laptop. That's a threat model most software never thinks about.



Corn

Okay, so ADB-with-a-collector is the good version. What about the methods that skip ADB analysis altogether? Backups first. I'd have guessed backups are the gold standard, since everything's in there.



Herman

That guess is wrong, and it's a very common one. The old adb backup with the all flag only requests backups from apps that explicitly allowed it. And from Android 12 onward, it extracts essentially nothing for almost all apps.



Corn

Nothing.



Herman

Almost nothing. There's an SMS-only option, which runs a backup of the telephony provider package, and that's effectively what MVT's backup module checks. Text messages and MMS. Useful if the infection vector was a malicious link in a text, but that's a narrow window.



Corn

And if the vector was a zero-click exploit through a messaging app?



Herman

Then the backup shows you the texts and none of the thing that mattered. You'd be checking the front door while the intruder came through a window.



Corn

So on iPhone a backup is a treasure chest and on Android it's a postcard.



Herman

Amnesty says much the same thing. Unlike iOS, the Android backup only provides a limited amount of relevant data.



Corn

What about bugreports? Those sound thorough.



Herman

They are rich, but inconsistent in format and structure across Android versions and vendors. Whatever parser you write has to cope with a dozen dialects. And the logs inside have a shelf life. Logcat relies on fixed buffers that fill up quickly. Crash dumps from debuggerd are stored as a limited number of tombstone files that rotate over time.



Corn

So the evidence overwrites itself.



Herman

Like a security camera that tapes over the oldest footage. And that's the real structural problem, bigger than any single method. Amnesty says defenders often contact them about suspected infections from months or even years earlier. By the time anyone looks, the historical evidence may already be gone. For five years, civil society forensics on Android relied, in their words, almost exclusively on incidental records and log files not designed for forensic or security use.



Corn

Incidental. So you're reconstructing a burglary from the receipts in the bin.



Herman

Pretty much. Now, the one more thing on collection methods: on-device collection. In September 2023, Amnesty fellow Paul Aguilar, working with SocialTIC in Mexico, published an experiment called APKQF. The idea was to do AndroidQF-style collection on the phone itself, no computer at all, in the same format, so MVT could analyse it. The output could be shared over WhatsApp or Google Drive.



Corn

Why Mexico?



Herman

Because computers are the bottleneck. Android is about three quarters of phones in Mexico and around eighty-five percent in South America. A journalist there may not have a laptop they trust, or a safe place to plug in.



Corn

And plugging into a borrowed machine is its own risk, I suppose.



Herman

You'd be connecting a possibly compromised phone to a computer you don't control, in a café or an office. Skipping the computer removes that step.



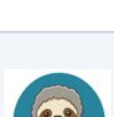
Corn

But doesn't running a collector on the phone itself have a problem? You're asking possibly compromised software to report on itself.



Herman

It's a fair worry, and it's a trade-off. A collector running on the device is subject to whatever the device is hiding, in a way a pushed binary under an external operator is only somewhat less so. But for people with no trusted computer at all, an imperfect check beats no check.



Corn

Is it a real product I can install?



Herman

No, and I want to be careful. It was a fellowship experiment. I haven't seen any sign of a Play Store release or ongoing maintenance. It's a proof that the idea works, not something you can go download.



Corn

So the no-computer option is a good idea nobody shipped. Typical.



Herman

That's the acquisition side. The bigger story is what changed in May, when Google shipped a log source built for this on purpose.



Corn

Go on.



Herman

On the twelfth of May, Google launched Android Intrusion Logging as part of Advanced Protection Mode, developed with Amnesty and Reporters Without Borders. Amnesty released AndroidQF and MVT updates the same day to collect and analyse the logs. It's the first time Android has had a log designed for forensics, rather than scraps that happened to be lying around.



Corn

What's in it?



Herman

Newline-delimited JSON, derived from Android's SecurityLog API. Process starts, DNS queries, outbound connections, ADB activity, keyguard events, app installs and uninstalls, certificate events. Generated daily. MVT has three modules for it: DnsEvent, ConnectEvent and SecurityEvent. If an indicator matches, you get a critical alert with the matched indicator, the offending event and a timestamp.



Corn

A timestamp is the thing nobody had before.



Herman

Yes. Imagine you learn a domain was used by a spyware operator. Now you can ask whether the phone looked that domain up, and on which day. That turns a vague suspicion into a timeline. And here's the clever bit. The logs are encrypted on the device, with a key protected by the user's PIN, and stored in the user's Google account. Google says the intrusion log data is not accessible to Google.



Corn

So the spyware can own the phone and still can't delete the evidence, because it's no longer on the phone.



Herman

That's the point. It directly answers the non-persistent malware problem, where a reboot wipes the traces. And under Advanced Protection a locked phone automatically reboots after three days, which is its own defence, since a lot of exploits don't survive a reboot.



Corn

I have a theory. This is going to have conditions.



Herman

It has a lot of them. You need Advanced Protection Mode, plus the optional Intrusion Logging toggle. The Android 16 December update or newer. A linked Google account. And right now it's Pixel-only.



Corn

Pixel-only. So the journalist on a Samsung gets nothing.



Herman

Or on a Xiaomi. And the people Amnesty built these tools for are very often not carrying Pixels. Also, the logs include browser navigation history, which some people won't want sitting in a cloud account, however encrypted.



Corn

So the fix came from the vendor, not the researchers, and the vendor decided who gets it.



Herman

Donncha Ó Cearbhaill, who heads Amnesty's Security Lab, said it promises to shift the balance toward defenders. He also said earlier that these technical limits meant they were unable to reliably detect known attacks on Android. That's a strong admission.



Corn

Known attacks. Not even the secret ones.



Herman

Which leads to indicators of compromise. MVT matches against STIX2 files, and supports a subset of types: domain names, process names, file hashes, app IDs, Android property names, URLs. You can download the public ones with a single command.



Corn

So I scan, nothing matches, I'm clean.



Herman

That is exactly the belief Amnesty warns against. Verbatim, public indicators of compromise are insufficient to determine that a device is clean, and reliance on public indicators alone can miss recent forensic traces and give a false sense of security.



Corn

Why would public ones miss things?



Herman

Because once an indicator is published, the operator rotates the domain and changes the process name. Public lists describe last year's campaign. It's like a wanted poster for a burglar who has since changed his coat and moved town. Also, MVT can optionally check non-system APK hashes against VirusTotal. It's off by default, because it sends your hashes to a third party, and it waits sixteen seconds between requests so it doesn't burn the quota.



Corn

Sixteen seconds per app. I'd go and make tea.



Herman

Corn, you'd go and make tea if it were one second.



Corn

Fair.



Herman

Corn, let me ask you something. If you were handed a phone and told it might have Predator on it, would you trust a clean result from a scan?



Corn

No. I'd trust a dirty result and nothing else. A clean one just means the tool didn't know what to look for.



Herman

That's pretty much Amnesty's position.



Corn

Now let's hear from a listener. Percy left us a voicemail.
Percy: Hi Corn, hi Herman, this is Percy calling from the back of my brother's taxi in Port of Spain, waiting on a fare outside the fruit market, and I have been listening to your show about the Android phone spyware checking, and I wanted to ask about the thing where you plug the phone into the computer with the USB debugging turned on, because I tried that once with my nephew's phone and the computer made a sound like it was thinking very hard and then nothing happened at all, so my question is whether the cable itself matters, because I have a drawer full of cables and some of them are from a camera that broke in two thousand and one, and I keep thinking one of them might be the good one.



Herman

Percy, the cable does matter. Some are charge-only and carry no data, and that gives you exactly the silence you describe. Use one you know carries data.



Corn

The camera cable might be fine. Just try the phone's own one first.



Herman

Also worth checking that the phone actually asked you to authorise the computer. If that prompt never appeared on the screen, the connection isn't really established, whatever the computer's sound effects suggest.



Corn

And if the prompt doesn't show up even with a good cable?



Herman

Then try a different port on the computer, and make sure the phone's USB mode isn't set to charging only. Those two fix a surprising share of these cases. Debugging has to be switched on in developer options too, which hides behind tapping the build number several times.



Herman

Back to the consumer side. iVerify Basic landed on Google Play in May 2024 at ninety-nine cents, offering on-demand malware detection and secure browsing. Its enterprise sibling, Mobile Threat Hunting, uses signatures, heuristics and machine learning to look for anomalies in device activity.



Corn

And does it find anything?



Herman

In a 2024 test, roughly twenty-five hundred users ran Mobile Threat Hunting, and six, possibly seven, turned up Pegasus infections. That's a real signal. I'm not sneering at it.



Corn

Six out of twenty-five hundred. Though those aren't random users, are they?



Herman

No, people who run it are self-selected, often journalists and activists. And it came from a threat-hunting service backed by a security company, not from a sandboxed scanner reading its own folder.



Corn

Spell out the sandbox for me. What can an app see?



Herman

Its own data, and whatever the OS exposes through APIs. It cannot read other apps' private directories, cannot pull system logs the way a pushed collector can, cannot generate a bugreport, and it has no access to non-public indicators. Amnesty says reliable triage requires non-public indicators, research and threat intelligence, which you get through their Security Lab or Access Now's Digital Security Helpline.



Corn

So a scanner is a person standing outside the house checking the mailbox, and the investigator has the keys.



Herman

Close enough. And the sandbox is a good thing in general. It's why a random flashlight app can't read your messages. It's just the same wall that stops a defensive app from looking at what an attacker may have touched.



Corn

So the protection and the blindness are the same feature.



Herman

And the Play Store listing for iVerify Basic sits at two point nine stars from a hundred and nine reviews. People expect a magic shield for a dollar.



Corn

Two point nine. At least it's honest about not being magic.



Herman

And I haven't seen any public evidence that an ordinary Play Store antivirus or spyware scanner has ever detected Pegasus or Predator on Android. The one documented detection record is that paid iVerify tier.



Corn

Does that trend get better?



Herman

Worse. Amnesty says surveillance actors increasingly know about these forensic efforts and are more diligent about hiding their actions on a targeted device.



Corn

Meaning every public technique becomes a checklist for the other side.



Herman

Yes. Anything a researcher publishes about how they caught an implant is also a lesson in what to clean up next time.

N

Neville

Hello! Yes. So, I've been brought in specially as a consultant on this one.



Corn

Neville, you've been at the desk since nine making the tea.

N

Neville

Well, yes. That's also true. Actually, I'm the intern. Uncle Hilbert is away, and he's technically my great uncle twice removed, then sideways. He keeps everything in boxes, labelled and dated, boxes inside boxes.



Herman

Sounds like a man who'd love a forensic archive.

N

Neville

Oh, he would. Anyway, I looked this up on the bus this morning, on the number forty-two down Jaffa Road. I think a hedgehog is not a hog at all. The name comes from it rooting through hedges and grunting like a pig. Sorry, that's not about phones. Where was I? My flatmate Dov downloaded a spyware scanner for one pound ninety-nine and it said his phone was fine, so he was very pleased, and then he found a sticker on his camera that wasn't his.



Corn

That tells you the scanner didn't see the sticker.



Herman

Which is fair, a sticker isn't in any log. But it's a good reminder that a green tick only covers what the tool can look at.

N

Neville

Right! And I thought Herman said the logs get saved on the phone, which is where the spyware lives. Is that right?



Herman

Other way round, Neville. The logs are encrypted and stored in your Google account precisely so spyware on the phone can't delete them.

N

Neville

Oh, that's much better, actually. Much cleverer.



Herman

It is. And it only works if you're on the right Pixel with the right toggle on.



Corn

Which is the catch with most good ideas on Android.



Herman

Let me add one thing about logs. The three-day auto reboot interacts with logging nicely. If something non-persistent gets in, the reboot clears it, but the log of what it did has already left the building.



Corn

So the evidence outlives the attack.



Herman

That's the first time that has been true on Android for civil society researchers.



Corn

One common wrong belief out there: that you plug your phone in over ADB, run MVT, and that's the standard reliable check.



Herman

Amnesty removed that path because it was error-prone and inconsistent. The reliable route is AndroidQF with its pushed collector, and even then a clean result from public indicators proves very little.



Corn

Here's what sticks with me. If the best forensic log source on Android is Pixel-only and opt-in, what happens to the defenders Amnesty built these tools for, the ones carrying other phones?



Herman

It's a vendor decision, not a researcher one, so it can be scoped or withdrawn. It's a genuine shift, but a conditional one. And the gap between a consumer scanner and an investigator isn't a product gap. It's an access-to-intelligence gap.



Corn

Thanks to producer Hilbert Flumingtop, and to Neville for the tea and the cameo.



Herman

If this was your kind of episode, go back for episode sixteen, On Deepfakes, SynthID, And AI Watermarking; episode thirty-five, The Privacy Gap; and episode two, Local STT For AMD GPU Owners. This has been My Weird Prompts. You can find every episode at my weird prompts dot com.



Corn

See you tomorrow.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **MVT README, LICENSE, Dockerfiles , current main branch (commit c9190e0) [primary]**
https://github.com/mvt-project/mvt

2. **AndroidQF README, collection table, encryption docs , current main (commit 961fc17) [primary]**
https://github.com/mvt-project/androidqf

3. **Methodology for Android forensic , current [primary]**
https://docs.mvt.re/en/latest/android/methodology/

4. **ADB Analysis Removed , current [primary]**
https://docs.mvt.re/en/latest/android/adb/

5. **Check Android Intrusion Logs , current [primary]**
https://docs.mvt.re/en/latest/android/intrusion_logs/

6. **Indicators of Compromise / STIX2 support , current [primary]**
https://docs.mvt.re/en/latest/iocs/

7. **Amnesty Security Lab, 2026-05-12 [primary]**
https://securitylab.amnesty.org/latest/2026/05/android-intrusion-logging-as-a-new-source-of-data-for-consensual-forensic-analysis/

8. **TechCrunch, 2026-05-12**
https://techcrunch.com/2026/05/12/google-launches-new-android-security-feature-to-help-uncover-spyware-attacks/

9. **Amnesty Security Lab / Paul Aguilar, 2023-09-21**
https://securitylab.amnesty.org/latest/2023/09/fellowship-apkqf-simplifying-android-forensics/

10. **iVerify, 2024-05-22**
https://www.iverify.com/blog/iverify-basic-is-now-on-android

11. **Help Net Security, 2024-12-04**
https://www.helpnetsecurity.com/2024/12/04/detect-mercenary-spyware/

12. **WIRED**
https://www.wired.com/story/iverify-spyware-detection-tool-nso-group-pegasus/

13. **iVerify Basic Play Store listing**
https://play.google.com/store/apps/details?id=com.trailofbits.iverify

14. **Google Security Blog (Intrusion Logging developed with Amnesty and RSF)**
https://blog.google/security/whats-new-in-android-security-privacy-2026/