ODIN

TLP:CLEAR

ODIN RESEARCH · MONOGRAPH 34 / 30 · DELIVERY INFRASTRUCTURE

Cloned GeoIP Proxies and APK Workbenches

Modelling Mobile Leads

Ten Nginx edge nodes cloned from one image, a four-architecture staged Android workbench, and seven feed-labeled mobile endpoints

Prepared November 2026
Full Report · Public edition · Version 1.1

10
Cloned Nginx reverse-proxy nodes on Microsoft cloud in Hong Kong (5,035 exposed files)

14.38 MB
Custom Nginx binary, byte-identical on all 10 nodes, with an 89,976-byte GeoIP module

5,399
Unpacked Apktool files on a US Android repackaging workbench

12 MiB
Size of a staged native library (name withheld), the same for all four Android CPU types

What Is in This Report. 11 chapters and two appendices.

Ten Nginx edge nodes cloned from one image, a four-architecture staged Android workbench, and seven feed-labeled mobile endpoints

01	Executive Summary	04
02	What Was Found: Three Groups, No Shared Links	07
03	The Cloned Ten-Node GeoIP Nginx Fleet	10
04	The Multi-ABI Android APK Repackaging Workbench	13
05	Feed-Labeled Mobile Leads: Seven Sampled Endpoints	16
06	Other Explanations and False-Positive Traps	19
07	What Defenders Should Do	22
08	Detection Rules	24
09	Indicators	27
10	Scope and Limits	29
11	Sources	31
A	Indicators of compromise (0)	33
B	Detection rules (4 files)	34

IN SHORT

Across evidence dated 3 February and 16–20 November 2026, Odin by Brandefense examined ten Nginx reverse proxies in Hong Kong cloned from one server image. It also examined an exposed US workbench holding staged Apktool files for four Android CPU types, and seven feed-labeled mobile endpoints. No shared address, certificate or hash links the three groups, and all 18 host addresses are withheld pending corroboration. Turn off unmanaged APK installs on Android fleets, alert on external APK downloads, and require multi-signal corroboration before blocking cloud hosts.

01

CHAPTER 01

Executive Summary

IN SHORT

Across evidence dated 3 February and 16–20 November 2026, Odin by Brandefense examined ten Nginx reverse proxies in Hong Kong cloned from one server image. It also examined an exposed US workbench holding staged Apktool files for four Android CPU types, and seven feed-labeled mobile endpoints. No shared address, certificate or hash links the three groups, and all 18 host addresses are withheld pending corroboration. Turn off unmanaged APK installs on Android fleets, alert on external APK downloads, and require multi-signal corroboration before blocking cloud hosts.

INSIDE THIS CHAPTER

Three Decisions for Leaders 04

The Study in Numbers 05

01 Executive Summary

Mobile trojan campaigns rarely use one server for everything. Regional traffic filtering, app repackaging and command-and-control (C2) often sit with different cloud providers, so a takedown at one layer leaves the others standing. This report looks at one example of each layer, examined from evidence dated 3 February and 16–20 November 2026, and is careful about what links them.

1. **Ten Hong Kong reverse proxies share byte-identical files from one server image.**
   Ten open web directories on Microsoft cloud space in Hong Kong exposed 5,035 files. Every node holds the same custom 14.38 MB Nginx binary, the same GeoIP module and the same 15.44 MB error log. They also share paired certificate bundles; one node also held a mobile download configuration file (path withheld).

2. **An exposed workbench holds staged IPTV app files for four Android CPU types.**
   A US open directory held 5,399 unpacked Apktool files consistent with a streaming lure. Two of the four build trees stage a 12 MiB native library for 32-bit and 64-bit ARM and x86, next to a commercial packer and inline hooking libraries. A staged 65.15 MB APK and two control-panel archives sit at the web root.

3. **Seven feed-labeled mobile endpoints remain withheld pending independent behavioral corroboration.**
   The source totals 138 global sightings across four feed labels: PixPirate, LightSpy, GobRAT and KTLVBackdoor. Seven feed-labeled endpoints appear in the sampled inventory. Two Huawei Cloud endpoints carry a PixPirate feed label and share a certificate name; their C2 role needs independent corroboration. All seven endpoint addresses are withheld.

Three Decisions for Leaders

1. **Stop browser and unmanaged APK installs on Android fleets.** Turn off installs from unknown sources on managed Android devices through your MDM. Keep Google Play Protect active and restrict Accessibility Service grants.
2. **Watch for external APK downloads and packed multi-ABI artifacts.** Alert on APK files fetched from outside your app store or MDM, and hunt the multi-ABI packer and hooking combinations in Chapter 8.
3. **Keep uncorroborated cloud addresses off block lists.** Cloud addresses change owners quickly. Pair the published JARM fingerprints and certificate leads with an independent certificate or endpoint process signal before blocking any host.

The Study in Numbers

Number	What it is
10	Cloned Nginx reverse-proxy nodes on Microsoft cloud in Hong Kong (5,035 exposed files)
14.38 MB	Custom Nginx binary, byte-identical on all 10 nodes, with an 89,976-byte GeoIP module
5,399	Unpacked Apktool files on a US Android repackaging workbench
12 MiB	Size of a staged native library (name withheld), the same for all four Android CPU types
65.15 MB	Staged multi-ABI IPTV APK on the workbench
2	Huawei Cloud endpoints with a PixPirate feed label and shared certificate name (addresses withheld)
5	Other suspected or candidate feed-labeled mobile endpoints (addresses withheld)

How to use this report.

Security leaders can read Chapters 1, 6 and 7. Mobile, network and cloud engineers should read Chapters 3 to 5. SOC analysts and detection engineers should start with Chapters 8 and 9.

02

CHAPTER 02

What Was Found: Three Groups, No Shared Links

IN SHORT

The study covers 18 hosts and 10,434 exposed files in three groups. All 18 host addresses are withheld pending ownership or behavioral corroboration; use structural and TLS hunts instead.

02 What Was Found: Three Groups, No Shared Links

The three groups are a cloned edge fleet, an APK workbench and seven feed-labeled mobile endpoints, which defenders can evaluate using cyber threat intelligence. Together they look like the layers of a mobile trojan delivery chain. But no shared IP address, TLS certificate or file hash links them. Figure 1 therefore shows them as an analytical model, not as one proven operation.



Figure 1: Cloned GeoIP proxies alongside staged APK trees and feed-labeled leads

Group	Sampled hosts	Published addresses	Exposed files or global feed sightings	Hosting	Status
Cloned GeoIP Nginx edge fleet	10	0	5,035 exposed files	Microsoft Corp, Hong Kong	Owner not established, withheld
Multi-ABI Android APK workbench	1	0	5,399 exposed files	Unattributed US hosting	Owner not established, address and hashes withheld
PixPirate feed-labeled endpoints	2	0	18 global PixPirate sightings	Huawei Cloud, China	Needs independent corroboration, addresses withheld
Other feed-labeled mobile endpoints	5	0	120 global sightings (GobRAT, LightSpy, KTLVBackdoor)	AWS Germany, Alibaba, Huawei, UCloud	Suspected or candidate, addresses withheld

Open directories like these can belong to badly configured staging servers or to cloud machines that change tenants. Publishing their addresses, distinctive paths or file hashes could identify an unconfirmed third party, so this report withholds them and focuses on structural detection rules.

03

CHAPTER 03

The Cloned Ten-Node GeoIP Nginx Fleet

IN SHORT

All ten Hong Kong nodes share the same binary, module, error-log size and paired certificate bundles. That level of sameness points to one disk image copied ten times; disable directory listing and clear logs before cloning server images.

INSIDE THIS CHAPTER

One image, three tell-tale files 10

Paired certificates and configuration files 11

03 The Cloned Ten-Node GeoIP Nginx Fleet

Many open directories run ordinary distribution Nginx packages and look alike for that reason. These ten are different. They share non-standard file sizes and identical SHA-256 digests across executables, logs, TLS bundles and configuration includes. In total they share 14 file hashes.

The nodes exposed between 99 and 694 files each, 5,035 in all; defenders can use external attack surface management to find similar open directories. The mean is 503.5 files and the median 553.5.

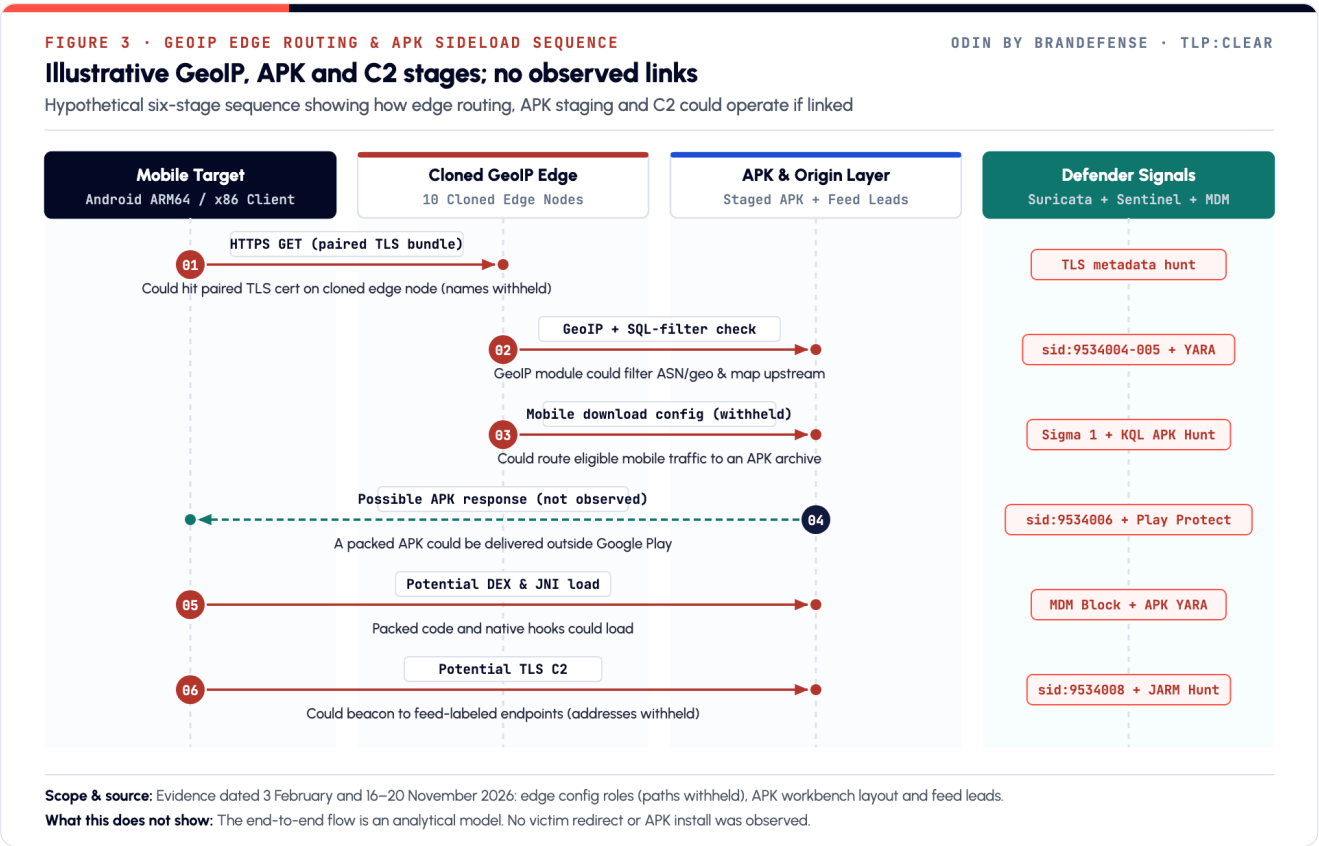


Figure 3: Illustrative GeoIP, APK and C2 stages; no observed links

Figure 3 is a model. It shows how GeoIP checks, SQL-injection filtering, paired certificates and a download configuration could work together in a mobile delivery chain. We did not observe a victim being redirected or an APK being installed.

One image, three tell-tale files

File (path withheld)	Nodes	Size (bytes)	What it is
Custom Nginx binary (path withheld)	10 of 10	14,379,936	Custom Nginx binary, about ten times larger than a stripped distribution build
Dynamic GeoIP module (path withheld)	10 of 10	89,976	Dynamic GeoIP module (`ngx_http_geoip_module`) that reads country and network variables such as `$geoip_country_code` and `$geoip_org`
Pre-filled Nginx error log (path withheld)	10 of 10	15,440,912	Pre-filled Nginx error log, identical on the nine nodes where it was hashed

The error log is the strongest clue. Live proxies write different log lines from the first request. The same 15.44 MB log, hash-identical on the nine nodes where it was hashed, indicates that the machines came from one captured disk snapshot that already held old logs. For defenders, the open directory puts that history on the owner's attack surface.

Paired certificates and configuration files

Every node holds 11 certificate bundles in an SSL folder (path and filenames withheld). Five are portal certificates. The other six form three paired bundles: a base bundle (7,120 to 8,847 bytes) and a companion bundle (7,137 to 9,471 bytes). The real certificate domains and filenames are withheld.

The paired certificate filenames could support domain rotation; no certificate switch or domain suspension was observed. This is a common pattern for grey-market app portals and for malware delivery alike.

Shared file (name and path withheld)	Nodes	Size (bytes)	Role
Upstream proxy configuration (path withheld)	1 of 10	19,834	File name suggests upstream proxy mapping; contents and behavior were not verified
Mobile download configuration (path withheld)	1 of 10	19,123	File name suggests app-download routing; contents and behavior were not verified
Companion TLS bundle B (name and path withheld)	10 of 10	9,471	Companion TLS bundle; rotation not verified
Companion TLS bundle A (name and path withheld)	10 of 10	7,142	Companion TLS bundle; rotation not verified
SQL-injection filter include (path withheld)	7 of 10	6,701	Query-string SQL-injection filter include; effect on scanners was not tested
Generated traffic-control config (path withheld)	4 of 10	5,730	Upstream routing settings; behavior not verified
Web routing script template (path withheld)	1 of 10	5,487	Edge routing template; behavior not verified

The two 19 KB `.conf` files surfaced on one node, where they were among the 15 largest files. On the other nodes, larger certificate chains pushed smaller `.conf` files out of the largest-file listing. So "1 of 10" means seen on one node, not missing from nine.

Together, GeoIP checks, an anti-SQL-injection include and a mobile download configuration suggest an edge layer built for traffic filtering and routing. Its effect on automated scanners was not tested. Whether this configuration steered mobile browsers toward a download was not tested.

04

CHAPTER 04

The Multi-ABI Android APK Repackaging Workbench

IN SHORT

A US open directory exposes staged Apktool files consistent with IPTV app repackaging across four Android CPU types, alongside a commercial packer and inline hooking libraries. Block unmanaged APK installs on Android fleets.

INSIDE THIS CHAPTER

Four Apktool trees 13

Staged archives and installer at the web root 14

Four CPU types, one packer, two hooking engines 14

04 The Multi-ABI Android APK Repackaging Workbench

Sideloaded Android trojans often pose as free streaming or IPTV players, an app impersonation problem that brand protection teams track. Users then install them outside Google Play and grant extra permissions. This workbench shows the server side of that kind of lure: 5,399 files in total. Our review covers the 90 largest files.

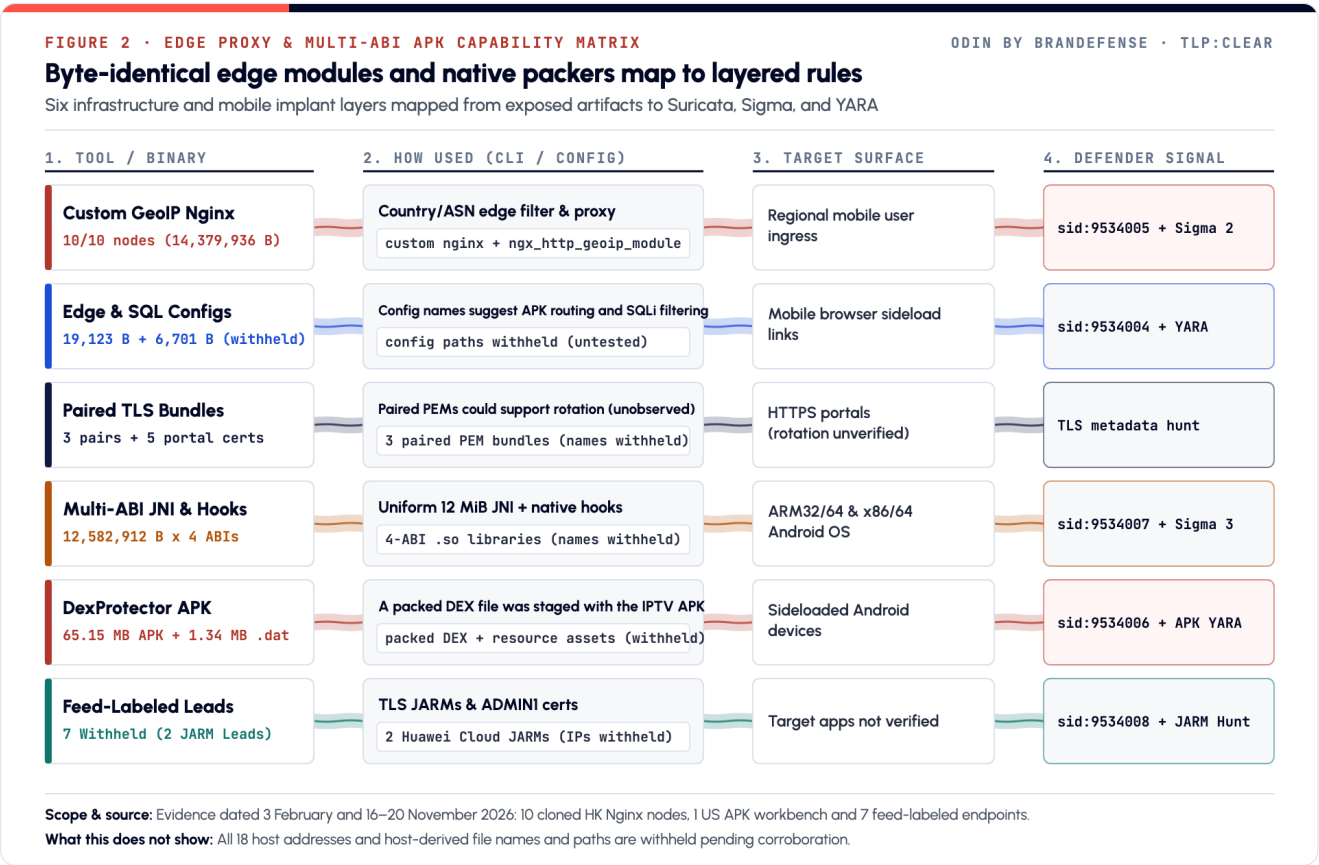


Figure 2: Byte-identical edge modules and native packers map to layered rules

Four Apktool trees

The workbench holds four parallel Apktool build trees; their paths are withheld because the host owner is unconfirmed. Four Apktool trees contain files consistent with app repackaging, including branding assets, native libraries and a staged APK; no build was observed.

Three file types point to this unpacking and editing layout. There is an original RSA signing certificate (1,050 bytes), a saved backup of the Android manifest (20,480 bytes), and Apktool duplicate-resource stubs (names and paths withheld).

Staged archives and installer at the web root

Artifact (path withheld)	Size (bytes)	What it is
Packaged IPTV archive A (`.zip`)	76,546,048	Packaged IPTV distribution bundle
Packaged IPTV archive B (`.zip`)	76,546,048	Second bundle of the same size
Signed multi-ABI installer (`.apk`)	65,149,417	Staged multi-ABI Android installer (65.15 MB)
Control-panel deployment archive (`.zip`)	2,228,822	Back-end web control panel archive
Panel rebranding patch archive (`.zip`)	819,308	Panel rebranding patch archive

Four CPU types, one packer, two hooking engines

Of the 90 largest files, 63 (70.0%) are native `.so` libraries. The staged folders cover all four Android NDK ABIs: `arm64-v8a`, `armeabi-v7a`, `x86` and `x86_64`. That spans phones, TV boxes and emulators.

- **A fixed 12 MiB native container.** In two of the four build trees, one native library (name and path withheld) is exactly 12,582,912 bytes for all four ABIs. The same size on ARM and x86 suggests fixed padding or an appended encrypted container.
- **A DexProtector loader.** Each ABI folder stages a DexProtector native loader alongside an encrypted secondary DEX asset (1,339,955 bytes) and an encrypted resource asset (68,720 bytes) in all four trees (names and paths withheld). DexProtector is a commercial app-protection packer.
- **Inline hooking.** Next to FFmpeg streaming libraries sit four native hooking libraries, including an older backup copy and a 79,872-byte hook library (names and paths withheld). These libraries are designed to intercept native functions at runtime inside the Android app sandbox.

Workbench file (tree paths withheld)	Copies	Size (bytes)	Role
Multi-ABI native container library (name and path withheld)	8	12,582,912	Native core container, same size for all four ABIs
Primary Dalvik bytecode file (name and path withheld)	2	6,252,104	Main compiled Dalvik bytecode
Encrypted secondary DEX asset (name and path withheld)	4	1,339,955	Encrypted secondary DEX for DexProtector
Older x86 native hook library (name and path withheld)	2	1,121,444	Older x86 native hook binary
Packer unpacking stub, 64-bit ARM (name and path withheld)	1	349,632	Packer unpacking stub, 64-bit ARM
Packer unpacking stub, 32-bit ARM (name and path withheld)	1	312,296	Packer unpacking stub, 32-bit ARM
Native app library (name and path withheld)	1	282,120	64-bit ARM library; behavior unverified
Native inline hook library, 64-bit ARM (name and path withheld)	1	228,176	Native inline hook engine, 64-bit ARM

The file names, file hashes and build-tree paths for these artifacts are withheld because the host owner is unconfirmed. We read the directory listing and file metadata. We did not install or run the APK.

05

CHAPTER 05

Feed-Labeled Mobile Leads: Seven Sampled Endpoints

IN SHORT

Seven sampled feed-labeled mobile endpoints appear alongside 138 global feed sightings across four family labels. All seven endpoint addresses are withheld pending independent behavioral corroboration; use the JARM and certificate leads to hunt.

INSIDE THIS CHAPTER

Two Huawei Cloud endpoints with a PixPirate feed label 16

Five other withheld endpoints 17

05 Feed-Labeled Mobile Leads: Seven Sampled Endpoints

The ten clones show clear internal links. Nothing of that kind ties them to the workbench or to the feed-labeled endpoints below. Figure 4 lays the three groups side by side as a model of how such layers fit together.



Figure 4: Golden-image cloning and Apktool trees expose deterministic hashes

The source totals 138 global sightings across four feed labels; seven endpoints appear in the sampled inventory. The family names are the feeds' own labels, not our attribution to a named actor.

Two Huawei Cloud endpoints with a PixPirate feed label

Two Huawei Cloud endpoints carry a PixPirate feed label and share a certificate name; their C2 role needs independent corroboration. PixPirate is an Android banking trojan. Three traits appear on the pair.

1. **Same provider.** Both sit in a Huawei Cloud data center in China; the two addresses are withheld pending a second qualifying signal.
2. **Same certificate name.** The two listeners present the same TLS certificate common name. The name itself is withheld.
3. **Two TLS fingerprints.** Each listener has its own JARM fingerprint, shown in the table below. Pair the JARM value with certificate or mobile process signals before blocking.

Endpoint	Hosting	JARM fingerprint (hunt only)
Huawei Cloud PixPirate-labeled endpoint A (withheld)	Huawei Cloud, China	`29d29d00029d29d22c29d29d29d29dc2ddcfd203d071c45b4b0ffe3d7b4b89`
Huawei Cloud PixPirate-labeled endpoint B (withheld)	Huawei Cloud, China	`21d19d00021d21d21c21d19d21d21d3b0d229d76f2fd7cb8e23bb87da38a20`

Five other withheld endpoints

The other three feed labels total 120 global sightings: 55 under GobRAT, 40 under LightSpy and 25 under KTLVBackdoor, alongside 18 under PixPirate. Five endpoints under those three labels appear in the sampled inventory. Each is suspected or candidate only, so its address is withheld.

Feed label	Endpoints	Hosting	Status	Shared trait
LightSpy	2	AWS, Germany	Suspected	Self-signed certificate `CN=ADMIN1` and one shared JARM (withheld)
LightSpy	1	UCloud, Hong Kong	Candidate	Single signal only
GobRAT	1	Alibaba Cloud, China	Suspected	Single host on shared cloud space
KTLVBackdoor	1	Huawei Cloud, China	Suspected	Single host on shared cloud space

`CN=ADMIN1` is a default-looking certificate name and turns up on harmless servers too. Treat it as a hunting lead only.

CHAPTER 06

Other Explanations and False-Positive Traps

IN SHORT

10,434 exposed files and 138 global feed sightings yield hunting leads rather than immediate IP block lists. Pair TLS and file leads with endpoint context before blocking.

06 Other Explanations and False-Positive Traps

Large open directories and single-signal TLS matches are easy to over-read. Figure 5 compares the size of each surface with what this report publishes from it.

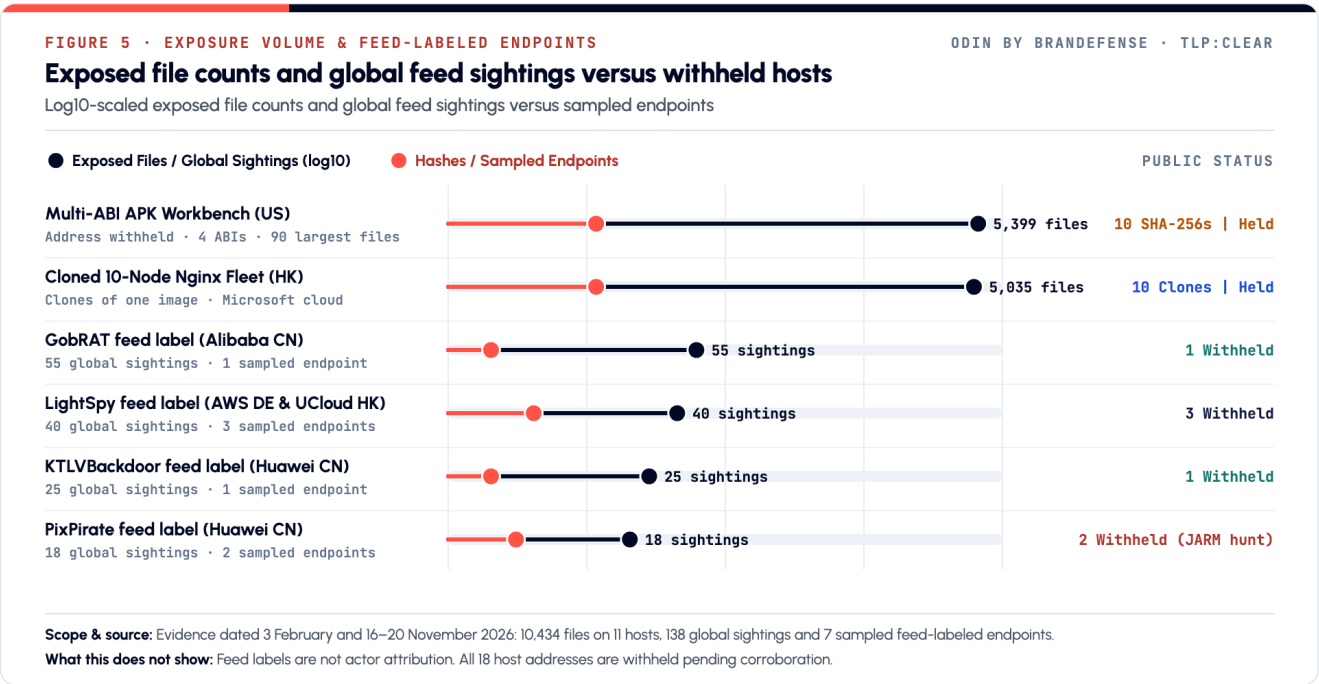


Figure 5: Exposed file counts and global feed sightings versus withheld hosts

Four other readings of the evidence remain open.

1. **A grey-market app-portal CDN.** Cloned proxies with GeoIP modules, anti-SQL includes, paired certificates and mobile download configurations also front offshore gaming and streaming portals. These portals hand out apps outside the stores. No APK behind the download configuration was retrieved, so the ten Hong Kong addresses and their configuration paths stay withheld.
2. **A commercial IPTV white-label builder.** The four build trees and panel archives could belong to an illicit IPTV rebranding service that uses DexProtector to protect pirated streams. Whatever the motive, unmanaged APKs with multi-ABI inline hooks bypass Google Play protection. The workbench artifacts need behavioral confirmation before any hash-based block.
3. **Cloud address churn.** All 18 hosts sit with large cloud providers: Microsoft, AWS, Huawei, Alibaba and UCloud. Addresses there move between customers quickly. Require an independent behavioral or certificate signal before blocking any cloud address, and review temporary blocks after 30 days.
4. **Default certificates and stock TLS stacks.** `CN=ADMIN1` and common JARM values recur on benign servers. Suricata SID 9534008 treats `CN=ADMIN1` as a hunting signal only.

Surface	Easy mistake	Safer approach
10 cloned Hong Kong Nginx nodes	Blocking cloud addresses or any `ngx_http_geoip_module` build	Keep the addresses off block lists; hunt the GeoIP module paired with multi-PEM TLS configs
APK workbench	Flagging legitimate CI `apktool` or FFmpeg builds	Require multi-ABI native libraries, packer assets and hooking libraries together
`CN=ADMIN1` certificate and JARM	Matching default appliance certificates or stock TLS stacks	Hunt only, and confirm with mobile endpoint process data
PixPirate-labeled Huawei Cloud pair	Blocking on a feed label and JARM without a second signal	Withhold from block lists; pair JARM with certificate or mobile process telemetry

07

CHAPTER 07

What Defenders Should Do

IN SHORT

This chapter lists the actions defenders should take, in priority order.

07 What Defenders Should Do

Mobile and endpoint

- [ ] Turn on the Android Enterprise policies that block installs from unknown sources through your MDM. Keep Google Play Protect on, and limit Accessibility Service rights for apps from outside the store.
- [ ] Test the three unvalidated YARA structural hunts from Chapter 8 on configuration text, certificate manifests and multi-ABI build specifications before adding them to sandbox pipelines.

Network security

- [ ] Use the two PixPirate-labeled JARM fingerprints and the `CN=ADMIN1` certificate subject as TLS hunting leads on firewalls and mobile ZTNA gateways. Require certificate or endpoint process corroboration before blocking.
- [ ] Alert on APK files (`application/vnd.android.package-archive`) delivered from outside approved stores and MDM hosts.

Cloud and platform

- [ ] Check your own Nginx reverse proxies. Turn directory listing off (`autoindex off`) and block public access to binary, module, certificate, log and configuration directories, as well as `*.conf` and `*.inc` files.
- [ ] Clear old logs from golden VM images before you clone them.

Detection engineering

- [ ] Deploy the Suricata, Sigma and KQL rules in Chapter 8. Fill the MDM and Android CI allowlists with exact hostnames first, so authorized builds do not flood the queue.

08

CHAPTER 08

Detection Rules

IN SHORT

This chapter describes each published rule and where to deploy it.

INSIDE THIS CHAPTER

How the Suricata signatures work 24

How the other rules work 24

08 Detection Rules

The full rules are in Appendix B. The Suricata file passes a syntax lint. The YARA, Sigma and KQL files were checked by reading, not loaded in live engines for this edition. None of the rules was replayed against live traffic, so test them in your own environment first.

Rule file	What it catches
`odin_cloned_geoip_nginx_apk_delivery_and_pixpirate.rules`	Suricata 7: five signatures for exposed SQL-filter regexes, GeoIP module downloads, external APK deliveries, unpacked multi-ABI `.so` requests and `CN=ADMIN1` certificates
`sentinel_geoip_apk_delivery_and_mobile_c2_hunt.kql`	Microsoft Sentinel and Defender XDR: explicit `.apk` URL downloads outside approved MDM hosts, plus `nginx`, `apktool`, `java` or `adb` commands that name the GeoIP module or multi-ABI repackaging markers
`sigma_exposed_nginx_conf_geoip_and_apk_sideload.yml`	Sigma (3 rules): proxy requests for external APKs or unpacked multi-ABI `.so` paths; unmanaged `nginx` launches with the GeoIP module; and Linux repackaging tool command lines referencing Android NDK ABI names
`odin_cloned_nginx_apk_conf_and_jni_library.yar`	YARA (3 unvalidated structural hunts): Nginx configuration text, certificate manifests and multi-ABI build specifications.

How the Suricata signatures work

SID	Looks for	Why
9534004	An HTTP 200 response body holding `select|union` and `$query_string`	Catches an exposed SQL-injection filter include file
9534005	A `GET` for `ngx_http_geoip_module.so`	Catches retrieval of the GeoIP module from outside
9534006	An external response with content type `application/vnd.android.package-archive` and a ZIP header	Hunt only: flags APK deliveries outside approved channels
9534007	A `GET` for any `.so` library under `/lib/<abi>/` across the four Android NDK ABIs	Hunt only: flags retrieval of unpacked multi-ABI Android native libraries
9534008	A TLS server certificate with `CN=ADMIN1`	Hunt only: default-looking certificate seen on hosts with a LightSpy feed label

Keep SIDs 9534004 through 9534008 in alert mode, with `pass` rules for approved MDM and build hosts.

How the other rules work

The KQL query looks back 7 days. Because the unconfirmed hosts' paths and uncorroborated C2 addresses are withheld, the network branch matches the generic `.apk` URL pattern explicitly (`(?i)\.apk(?:\?|$)`). The query skips remote hosts in an exact MDM hostname list and devices in an exact build-host list.

The process side looks for `nginx`, `apktool`, `java` or `adb` with the GeoIP module or Android NDK ABI and repackaging markers on the command line. Each match gets one hunting label: hunt a multi-ABI APK workbench, hunt a cloned GeoIP edge config, or hunt an unmanaged APK transfer.

The first Sigma rule reads web proxy logs and flags requests for `.apk` URLs or unpacked `.so` libraries under `/lib/<abi>/`. Three exact MDM hostnames and one internal scanner address are excluded. The second flags `nginx` launches whose command line names `ngx_http_geoip_module.so`. The third pairs an unvalidated `selection_repack_tool` executable list with the Android NDK ABI names `arm64-v8a`, `armeabi-v7a` or `x86_64`. The third rule flags Linux command lines mentioning these ABI names under repackaging tools; analysts must verify that a repackaging process ran.

YARA rule	Fires when
`ODIN_Cloned_Nginx_GeoIP_APK_Delivery_Config`	`ngx_http_geoip_module.so` and `proxy_pass` appear with `$geoip_country_code`, `$geoip_org` or the `select|union` and `$query_string` filter terms
`ODIN_Cloned_Nginx_Multi_PEM_TLS_Manifest`	Multiple `ssl_certificate` and `ssl_certificate_key` directives appear with `.pem`
`ODIN_MultiABI_APK_Repack_Build_Spec`	`apktool` or `AndroidManifest.xml` appears with all four Android NDK ABI names (`arm64-v8a`, `armeabi-v7a`, `x86` and `x86_64`)

The two MITRE ATT&CK for Mobile tactics in play are Initial Access (TA0027), through sideloaded and repackaged apps, and Command and Control (TA0037), through encrypted channels.

09

CHAPTER 09

Indicators

IN SHORT

This chapter lists the published indicators and explains what stays withheld.

09 Indicators

Appendix A holds 0 published block indicators: all 18 host addresses and all 18 file hashes are withheld pending independent behavioral corroboration or host ownership checks. Three hunting indicators appear in this report only: the two PixPirate-labeled JARM fingerprints and the `CN=ADMIN1` certificate subject.

Type	Count	Where to use it
Network addresses	0 (18 withheld)	Withheld pending independent behavioral corroboration and host ownership checks
File hashes (SHA-256)	0 (18 withheld)	Withheld pending host ownership and behavioral confirmation
JARM fingerprints	2 (1 withheld)	TLS hunting only, pair with certificate or mobile process signals (Chapter 5)
Nginx configuration paths	0 (withheld)	Owner-specific configuration paths withheld pending host ownership confirmation
Certificate subject	1	TLS hunting only: `CN=ADMIN1`

Withheld: all 18 host addresses, covering the ten Hong Kong nodes and the US workbench. This also includes the two Huawei Cloud PixPirate-labeled endpoints and the five other suspected or candidate feed-labeled hosts. Also withheld are all 18 file hashes from the unconfirmed open-directory hosts (eight from the Hong Kong fleet and ten from the US workbench). The owner-specific Nginx configuration paths, the distinctive workbench paths and file names, the real certificate domains and filenames, the shared PixPirate certificate name and the LightSpy JARM are withheld as well.

10

CHAPTER 10

Scope and Limits

IN SHORT

This chapter sets out what the evidence can show and what it cannot.

10 Scope and Limits

- **Evidence dates.** The findings draw on evidence dated 3 February and 16–20 November 2026, with a research cutoff of 20 November 2026.
- **Files, not outcomes.** We read directory listings and file metadata. We did not run any binary or install the APK, and we saw no victim device.
- **No proven chain.** No shared address, certificate or hash links the Hong Kong fleet, the US workbench and the feed-labeled endpoints. The end-to-end delivery picture is an analytical model.
- **Owners not established.** The Hong Kong fleet could serve a grey-market app portal, and the workbench could serve an IPTV piracy business. Their addresses, configuration paths and host-derived file names are withheld for that reason.
- **Feed labels are not attribution.** PixPirate, LightSpy, GobRAT and KTLVBackdoor are the feeds' own family labels.
- **Partial view.** The workbench figures describe its 90 largest files out of 5,399. The Hong Kong figures rely on each node's largest files, so a file missing from a node's list may still exist there.
- **Rule testing.** The Suricata file passes a syntax lint. The other rule files were not loaded in live engines for this edition, and no rule was replayed against live traffic. Test and tune them before production use.

11

CHAPTER 11

Sources

IN SHORT

This chapter lists the public references used in the report.

11 Sources

1. NGINX documentation. Module ngx_http_geoip_module. https://nginx.org/en/docs/http/ngx_http_geoip_module.html (verified 20 November 2026)
2. NGINX documentation. Module ngx_http_proxy_module. https://nginx.org/en/docs/http/ngx_http_proxy_module.html (verified 20 November 2026)
3. Android Developers. Android ABIs. https://developer.android.com/ndk/guides/abis (verified 20 November 2026)
4. Android Open Source Project. Application Sandbox. https://source.android.com/docs/security/app-sandbox (verified 20 November 2026)
5. OWASP Mobile Application Security. Mobile Application Security Testing Guide (MASTG). https://mas.owasp.org/MASTG/ (verified 20 November 2026)
6. MITRE ATT&CK for Mobile. TA0027 Initial Access. https://attack.mitre.org/tactics/TA0027/ (verified 20 November 2026)
7. MITRE ATT&CK for Mobile. TA0037 Command and Control. https://attack.mitre.org/tactics/TA0037/ (verified 20 November 2026)
8. Suricata documentation. HTTP Keywords. https://docs.suricata.io/en/latest/rules/http-keywords.html (verified 20 November 2026)

APPENDICES

Indicators and Detection Rules

FOR DEFENDERS

No individual indicators are published for this study; Appendix A explains why. Appendix B carries every published detection rule in full.

INSIDE

A. Indicators of compromise (0) 33

B. Detection rules (4 files) 34

A Indicators of Compromise

No individual addresses or file hashes are published for this study. The hosts in this report are exposed or unattributed systems rather than confirmed adversary infrastructure, so they stay anonymous. Use the detection rules in Appendix B.

B Detection Rules

Each rule file loads in its own engine (Suricata 7, YARA-X, Kusto, pySigma). Tune thresholds and allow-lists to your environment before alerting or blocking. The same files can be downloaded from the online edition.

odin_cloned_geoip_nginx_apk_delivery_and_pixpirate.rules

SURICATA

```
# Odin by Brandefense - Monograph 34 / 35 Suricata 7.x Detection Suite (public edition)
# Coverage: Cloned Nginx open-directory SQL-filter regex exposure, dynamic GeoIP module retrieval,
#           external Android APK archive delivery, unpacked multi-ABI Android native library requests, and
default CN=ADMIN1 TLS certs (hunt only)
# Classification: TLP:CLEAR

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R34] Cloned Nginx Open-Directory SQL-Filter Include
Regex Exposure"; flow:established,from_server; http.stat_code; content:"200"; http.response_body; bsize:>32;
content:"select|7C|union"; fast_pattern; nocase; content:"$query_string"; classtype:policy-violation; sid:
9534004; rev:2; metadata:created_at 2026_11_20, updated_at 2026_11_22, deployment perimeter, tactic
initial_access;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R34] Cloned Nginx Dynamic GeoIP Module Retrieval
(ngx_http_geoip_module.so)"; flow:established,to_server; http.method; content:"GET"; http.uri; bsize:>24;
content:"ngx_http_geoip_module.so"; fast_pattern; classtype:policy-violation; sid:9534005; rev:1;
metadata:created_at 2026_11_20, deployment perimeter, tactic command_and_control;)

alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R34] HUNT External Android APK Sideload Archive
Delivery"; flow:established,from_server; http.stat_code; content:"200"; http.content_type; bsize:39;
content:"application/vnd.android.package-archive"; fast_pattern; http.response_body; bsize:>16; content:"PK|03
04|"; startswith; classtype:policy-violation; sid:9534006; rev:2; metadata:created_at 2026_11_20, updated_at
2026_11_20, deployment mobile_egress, policy behavioral_hunt;)

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ODIN [R34] HUNT Unpacked Multi-ABI Android Native Library
Retrieval"; flow:established,to_server; http.method; content:"GET"; http.uri; bsize:>14; content:"/lib/";
fast_pattern; pcre:"/\/lib\/(arm64-v8a|armeabi-v7a|x86|x86_64)\/[A-Za-z0-9_.-]+\.so(\?|$)/"; classtype:policy-
violation; sid:9534007; rev:3; metadata:created_at 2026_11_20, updated_at 2026_11_22, deployment perimeter,
policy behavioral_hunt;)

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ODIN [R34] HUNT Default CN=ADMIN1 Server Certificate (seen on
hosts with a LightSpy feed label)"; flow:established,from_server; tls.cert_subject; bsize:>8;
content:"CN=ADMIN1"; fast_pattern; classtype:trojan-activity; sid:9534008; rev:2; metadata:created_at 2026_11_20,
updated_at 2026_11_20, deployment perimeter, policy behavioral_hunt;)
```

odin_cloned_nginx_apk_conf_and_jni_library.yar

YARA

```
/*
   Odin by Brandefense - Monograph 34 / 35 YARA Artifact Suite
   Scope: Three unvalidated structural hunts for Nginx GeoIP edge configuration text,
          multi-bundle PEM TLS certificate manifests, and multi-ABI Android
          Apktool build specifications.
   Classification: TLP:CLEAR
*/

rule ODIN_Cloned_Nginx_GeoIP_APK_Delivery_Config
{
    meta:
        author = "Odin by Brandefense"
        monograph = "34 / 35"
        classification = "TLP:CLEAR"
        disposition = "HUNT_CLONED_NGINX_EDGE_CONFIG"
        description = "Matches Nginx edge configuration text combining ngx_http_geoip_module.so and proxy_pass
with GeoIP variables or SQL-filter regexes"
    strings:
        $geo = "ngx_http_geoip_module.so" ascii
        $var1 = "$geoip_country_code" ascii
```

```
        $var2 = "$geoip_org" ascii
        $sql1 = "select|union" ascii nocase
        $sql2 = "$query_string" ascii
        $ngx = "proxy_pass" ascii
    condition:
        filesize < 128KB and $geo and $ngx and ($var1 or $var2 or ($sql1 and $sql2))
}

rule ODIN_Cloned_Nginx_Multi_PEM_TLS_Manifest
{
    meta:
        author = "Odin by Brandefense"
        monograph = "34 / 35"
        classification = "TLP:CLEAR"
        disposition = "HUNT_PAIRED_EDGE_TLS"
        description = "Matches Nginx SSL configuration blocks declaring multiple PEM certificate and key
directives"
    strings:
        $ssl1 = "ssl_certificate" ascii
        $ssl2 = "ssl_certificate_key" ascii
        $pem = ".pem" ascii
    condition:
        filesize < 64KB and #ssl1 >= 4 and #ssl2 >= 2 and $pem
}

rule ODIN_MultiABI_APK_Repack_Build_Spec
{
    meta:
        author = "Odin by Brandefense"
        monograph = "34 / 35"
        classification = "TLP:CLEAR"
        disposition = "HUNT_REPACKAGED_MULTI_ABI_APK_SPEC"
        description = "Matches Android build or Apktool specification files referencing all four NDK ABIs
alongside manifest or tool entries"
    strings:
        $tool1 = "apktool" ascii nocase
        $tool2 = "AndroidManifest.xml" ascii
        $abi1 = "arm64-v8a" ascii
        $abi2 = "armeabi-v7a" ascii
        $abi3 = "x86" ascii
        $abi4 = "x86_64" ascii
    condition:
        filesize < 512KB and ($tool1 or $tool2) and $abi1 and $abi2 and $abi3 and $abi4
}
```

sentinel_geoip_apk_delivery_and_mobile_c2_hunt.kql

KQL (SENTINEL / DEFENDER XDR)

```
// Odin by Brandefense - Monograph 34 / 35 Microsoft Sentinel & Defender XDR Hunt (public edition)
// Correlates unmanaged Android APK sideloading URLs,
// custom GeoIP Nginx module loads, and multi-ABI apktool repackaging commands
// The MDM allowlist compares exact hostnames only.
// Classification: TLP:CLEAR

let Lookback = 7d;
let ApprovedEdgeAndBuildHosts = dynamic(["EDGE-NGINX-PROXY-01", "ANDROID-CI-RUNNER-01", "SEC-VALIDATION-
LNX-01"]);
let ApprovedMdmHosts = dynamic(["play.googleapis.com", "dl.google.com", "mdm-packages.corp.internal"]);
let SuspiciousEdgeOrApkProcs = DeviceProcessEvents
    | where Timestamp >= ago(Lookback)
    | where not(DeviceName in~ (ApprovedEdgeAndBuildHosts))
    | where (FileName in~ ("nginx", "apktool", "java", "adb")
        and ProcessCommandLine has_any ("ngx_http_geoip_module.so", "apktool", "arm64-v8a", "armeabi-v7a",
"x86_64"))
    | project ProcTime = Timestamp, DeviceId, DeviceName, InitiatingProcessFileName, FileName,
ProcessCommandLine;
DeviceNetworkEvents
| where Timestamp >= ago(Lookback)
| extend RemoteHost = tolower(tostring(parse_url(iff(RemoteUrl contains "://", RemoteUrl, strcat("http://",
```

```
RemoteUrl))).Host))
| where not(DeviceName in~ (ApprovedEdgeAndBuildHosts))
    and RemoteUrl matches regex @"(?i)\.apk(?:\?|$)"
    and not(RemoteHost in (ApprovedMdmHosts))
| summarize
    NetFirstSeen = min(Timestamp),
    NetLastSeen = max(Timestamp),
    DistinctRemoteIps = dcount(RemoteIP),
    SampleRemoteIps = make_set(RemoteIP, 10),
    SampleRemoteUrls = make_set(RemoteUrl, 10),
    ConnectionCount = count()
    by DeviceId, DeviceName, InitiatingProcessFileName
| join kind=fullouter (SuspiciousEdgeOrApkProcs) on DeviceId, DeviceName
| extend ResolvedDevice = coalesce(DeviceName, DeviceName1)
| extend AlertDisposition = case(
    isnotempty(ProcessCommandLine) and ProcessCommandLine has_any ("apktool", "arm64-v8a", "armeabi-v7a",
"x86_64"), "HUNT_UNMANAGED_MULTI_ABI_APK_WORKBENCH",
    isnotempty(ProcessCommandLine) and ProcessCommandLine has "ngx_http_geoip_module.so",
"HUNT_CLONED_GEOIP_NGINX_EDGE_CONFIG",
    "HUNT_UNMANAGED_APK_SIDELOAD_TRANSFER"
)
| project NetFirstSeen, NetLastSeen, ProcTime, ResolvedDevice, AlertDisposition, InitiatingProcessFileName,
FileName, ProcessCommandLine, SampleRemoteIps, SampleRemoteUrls, ConnectionCount
| sort by coalesce(ProcTime, NetLastSeen) desc
```

sigma_exposed_nginx_conf_geoip_and_apk_sideload.yml

SIGMA

```
title: Unmanaged Android APK Sideload or Multi-ABI Native Library Request
id: 4b91e7a2-3d18-4c52-9f04-202611203401
status: experimental
description: Detects web proxy or perimeter HTTP requests retrieving unmanaged Android APK archives or unpacked
multi-ABI native libraries outside approved MDM channels.
author: Odin by Brandefense
date: 2026/11/20
tags:
  - attack.initial-access
  - attack.command-and-control
logsource:
  category: proxy
detection:
  selection_uri:
    c-uri|re: '(\.apk(\?|$)|/lib/(arm64-v8a|armeabi-v7a|x86|x86_64)/[A-Za-z0-9_.-]+\.so(\?|$))'
  filter_approved_mdm:
    cs-host:
      - 'play.googleapis.com'
      - 'dl.google.com'
      - 'mdm-packages.corp.internal'
  filter_security_scanner:
    c-ip:
      - '10.240.10.50'
  condition: selection_uri and not 1 of filter_*
falsepositives:
  - Internal mobile continuous-integration hosts downloading signed test APKs from non-registered staging servers
level: high
---
title: Unmanaged Nginx Execution Loading Dynamic GeoIP Module
id: 8d34c2f9-6a21-4e77-b108-202611203402
status: experimental
description: Identifies Linux hosts launching nginx and referencing ngx_http_geoip_module.so outside registered
reverse-proxy tiers.
author: Odin by Brandefense
date: 2026/11/20
tags:
  - attack.command-and-control
  - attack.defense-evasion
logsource:
  category: process_creation
  product: linux
```

```
  detection:
    selection_nginx:
      Image|endswith:
        - '/nginx'
    selection_conf_markers:
      CommandLine|contains:
        - 'ngx_http_geoip_module.so'
    filter_approved_edge_hosts:
      ComputerName:
        - 'EDGE-NGINX-PROXY-01'
        - 'EDGE-NGINX-PROXY-02'
    filter_lab_validation:
      ComputerName:
        - 'SEC-VALIDATION-LNX-01'
    condition: selection_nginx and selection_conf_markers and not 1 of filter_*
  falsepositives:
    - Newly provisioned regional reverse-proxy gateways not yet added to the approved edge inventory
  level: high
  ---
  title: Unmanaged Multi-ABI Android APK Repackaging Command Line
  id: 2f67a9e1-8b40-4d19-92c6-202611203403
  status: experimental
  description: Detects command-line references to Android NDK ABI names under repackaging or signing executables
  outside designated mobile CI runners.
  author: Odin by Brandefense
  date: 2026/11/20
  tags:
    - attack.defense-evasion
    - attack.initial-access
  logsource:
    category: process_creation
    product: linux
  detection:
    selection_repack_tool:
      Image|endswith:
        - '/apktool'
        - '/java'
        - '/aapt'
        - '/aapt2'
        - '/jarsigner'
        - '/apksigner'
    selection_repack_cli:
      CommandLine|contains:
        - 'arm64-v8a'
        - 'armeabi-v7a'
        - 'x86_64'
    filter_ci_build_runners:
      ComputerName:
        - 'ANDROID-CI-RUNNER-01'
        - 'ANDROID-CI-RUNNER-02'
    filter_security_lab:
      ComputerName:
        - 'SEC-VALIDATION-LNX-01'
    condition: selection_repack_cli and selection_repack_tool and not 1 of filter_*
  falsepositives:
    - Authorized mobile reverse-engineering sandboxes inspecting captured third-party APK archives
  level: critical
```

ODIN

About Odin Research

Odin Research is the public research series of Odin by Brandefense. It documents adversary infrastructure seen on the internet and turns each finding into indicators and detection rules that defenders can use. Victims and third parties stay anonymous, addresses appear only for confirmed adversary infrastructure, and no working attack code is published.

Odin is the agentic analyst inside the Brandefense threat intelligence platform, which unifies Cyber Threat Intelligence, Digital Risk Protection and External Attack Surface Management.

Online edition, indicators and rules · brandefense.io/odin-research/reports/cloned-geoip-proxies-and-apk-workbenches/
Short Read · brandefense.io/odin-research/blog/cloned-proxies-a-staged-apk-workbench-and-pixpirate-feed-leads/
Defense Kit · brandefense.io/odin-research/defense-kit/