

Ampire 360°



ПЕРСПЕКТИВНЫЙ
МОНИТОРИНГ

Ampire 360°

ПОЛИГОН КИБЕРБЕЗОПАСНОСТИ



Threat Intelligence

SOC & DFIR

Offensive Security

Киберучения

ИИ и киберзащита

Киберполигон Ampire

📅 14 октября



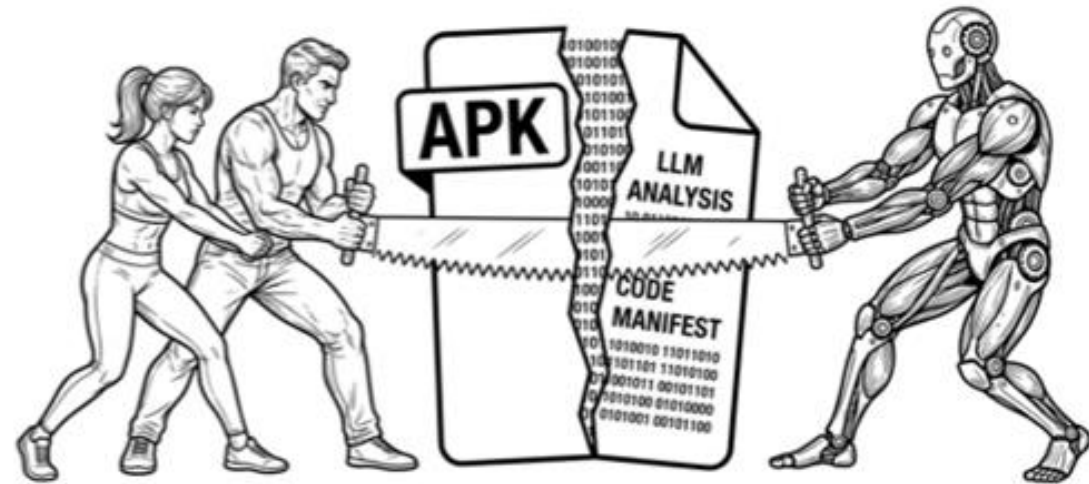
Анализ APK-файлов с помощью LLM

Кристина Стрельцова

Ведущий специалист
Перспективный мониторинг

Игорь Васильев

Ведущий специалист
Перспективный мониторинг



О чём поговорим?

Ampire 360°

Тренды

Актуальные мобильные угрозы

Кейс

На реальном примере покажем, как происходит процесс реверса APK с помощью LLM

Общая статистика по мобильным угрозам за 2025 год

14 млн

мобильных атак предотвращено за 2025 год
~1,17 млн атак в месяц

815 тыс.

обнаружено уникальных пакетов

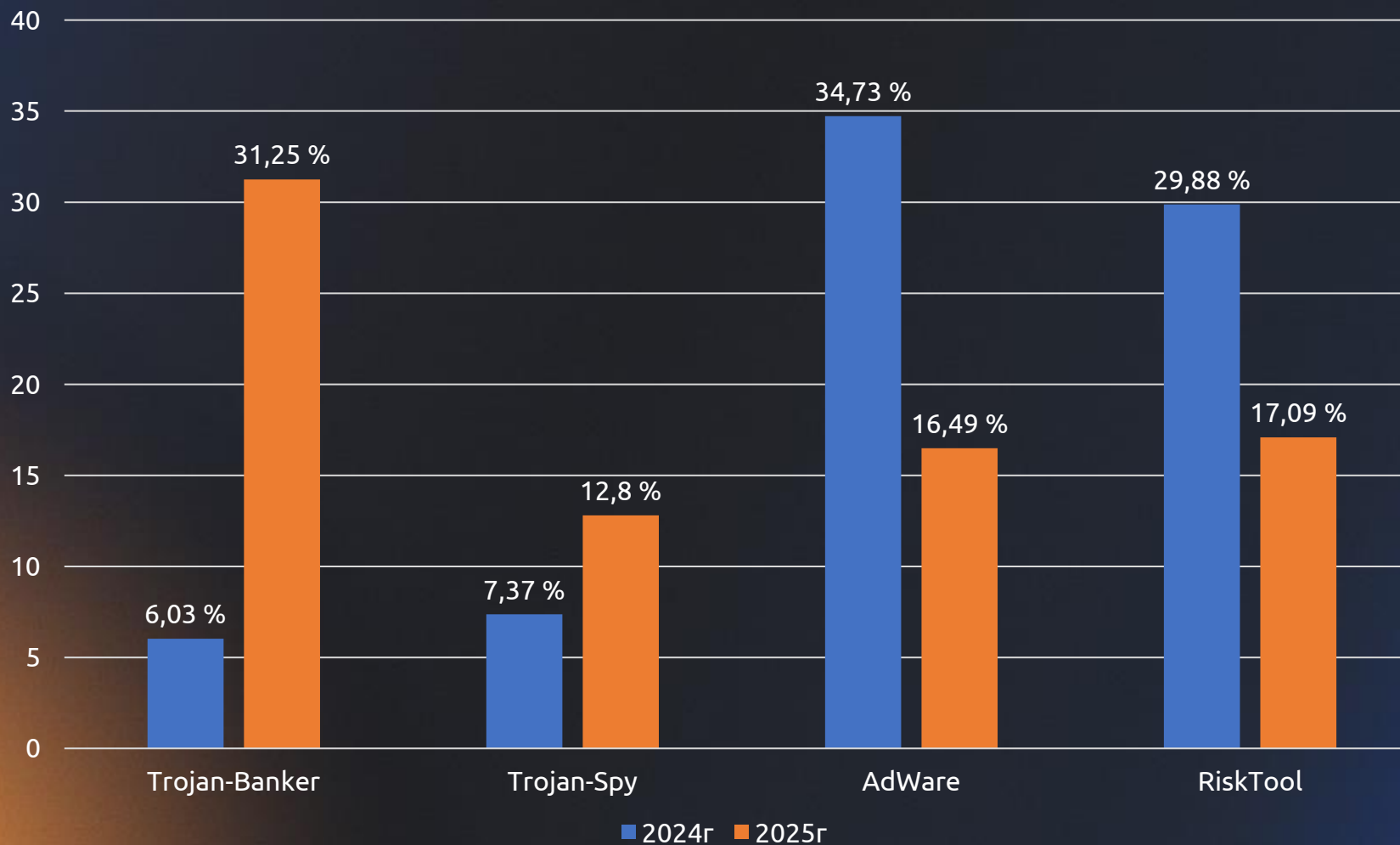
62 %

рекламное ПО

+271 %

установочных пакетов троянских программ
для Android

Распределение обнаруженного мобильного ПО по типам (2024–2025 гг)



Популярные фишинговые кампании

Empire 360°



Три человека погибли и восемь пострадали в ДТП. Авария произошла сегодня около 11:55 по местному времени, сообщили в региональном управлении СКР.

Ведомством организована проверка, по итогам которой будет решен вопрос о возбуждении уголовного дела по ст. 264 УК РФ (нарушение ПДД).

От полученных травм в карте скорой помощи скончался водитель, три его пассажира погибли на месте. Один пострадавший доставлен в больницу, 2 человека, в том числе двое детей, эвакуированы, еще двоим пострадавшим помощь была оказана на месте происшествия.

Видео с места происшествия публикуем ниже

71 16:34

Переслано из

ФОТО_ВИДЕО_С_МЕСТА_ПРОИСШЕСТВИЯ_АРХИВ.apk
991.6 KB 47 16:34



UnlimVPN.apk
9,8 МБ

Привет, дали 5 бесплатных активаций на лучший VPN на андроид, на днях релиз - 6 месяцев бесплатно, полный безлимит и обход всех БС, малое кол-во людей будет его использовать.

Не забудь выдать приложению нужные разрешения для обхода белых списков, спец возможности - разрешить и так далее, нереально крутой продукт будет!

22:04

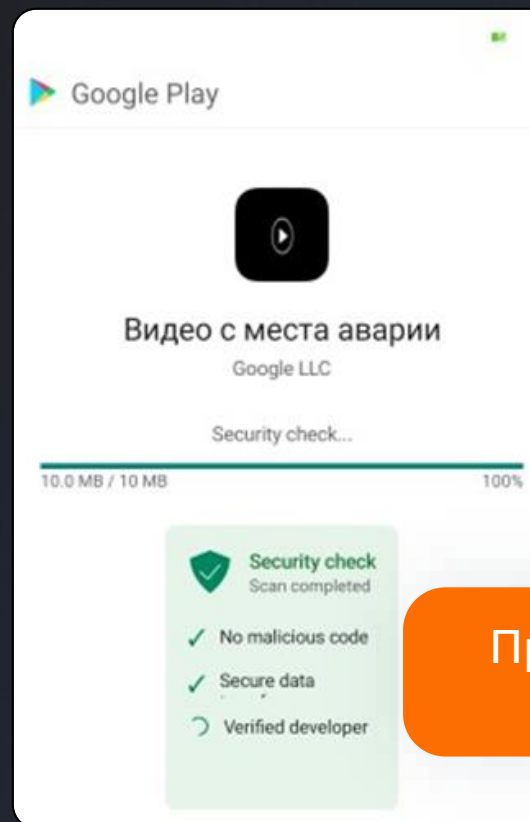
Сегодня

Привет, это ты на видео? 14:38



Видео.apk
44 МБ

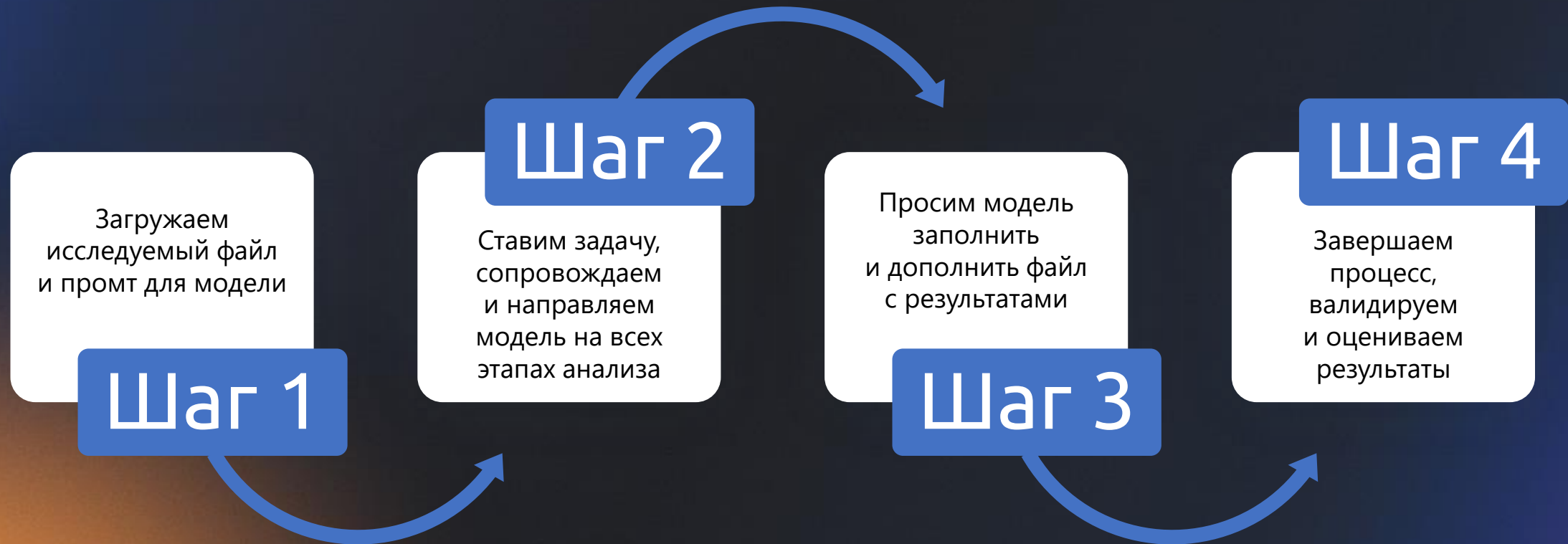
14:39



Прочитать статью
на Хабре



Основные этапы реверса APK с помощью LLM



Кейс



Подготовительный этап

Ampire 360°

Дай инфу по 2152b98a832ce3e442b486cееаа8e41a8a99657168986757913e2d916c7b1c21

Поиск информации по sha256. Это свежий APK, найденный на MalwareBazaar от 2026-09-24 07:03:52 UTC

Ответ модели:

Вердикт: ВРЕДНОСНОЕ ПО (Malware) — Android-ботнет CecBot / Mirai

- **MalwareBazaar:** сигнатура `null`, теги `apk`, `apk.cecbot`, `cecbot`, `ddos`, `signed`
- **Triage:** `malware_family: cecbot`, score 10, теги `family:cecbot`, `botnet`
- **Kaspersky:** `Malware`, `HEUR:Backdoor.AndroidOS.Agent.il`
- **VirusTotal:** 25/75 детектов (25 malicious, 42 undetected, 0 harmless), reputation -11

2152b98a... — Android-APT DDoS-бот семейства CecBot (вариант Mirai под Android), маскирующийся под пакет `com.imjustafriendlyguy.firmware` ("firmware"). Агрессивная персистентность через 17 системных сервисов + device-admin + boot-receiver, нативные ELF-модули `libattack.so` / `libwatchdog.so`, DoH-резолвинг C2 через Cloudflare/Google/Quad9, анти-VM и обфускация. Детектируется 25 вендорами; уверенно атрибутируется к mirai/usblid26.

Нужен дальнейший разбор (скачать APK, декомпиляция DEX/ELF через jadx/apktool, извлечение C2/IoC) — скажи, разверну.

Предложение модели для дальнейшего анализа

Создай папку Android-CecBot

Туда запиши эту информацию в cecbot.md

Создаём папку проекта по данному APK и просим записать найденную информацию

Ответ модели:

Bash: `mkdir -p /workspace/Android-CecBot && ls -la /workspace/Android-CecBot`

Write: `/workspace/Android-CecBot/cecbot.md`

Готово.

- Папка: `/workspace/Android-CecBot/`
- Файл: `/workspace/Android-CecBot/cecbot.md`

Записан полный отчёт: идентификация (все хеши), вердикт CecBot/Mirai, таблица детектов по вендорам, состав APK (4x ELF), список персистент-сервисов/ресиверов, DoH C2-эндпоинты, кодовая подпись, сработавшие YARA-правила, vendor intel (Triage/MWDB/Kaspersky/FileScan), хеши подобию и сводка.

Download link:

[https://\[redacted\]/files/eb2824c734b046b8be776b4dcb5c2a9b](https://[redacted]/files/eb2824c734b046b8be776b4dcb5c2a9b)

Copy

Upload another file

Так как файл у нас уже был скачан, мы просто загрузили на специализированную share-платформу

Вредоносный APK:

[https://\[redacted\]/files/eb2824c734b046b8be776b4dcb5c2a9b.apk](https://[redacted]/files/eb2824c734b046b8be776b4dcb5c2a9b.apk)

Только декомпилируй и деобфусцируй в существующую папку - /workspace/Android-CecBot
Далее сделай структуру(пути) распакованного APK, укажи отдельно .so если будут.
После сделай /workspace/Android-CecBot/struct.md

Отправляем запрос в модель для создания структуры распакованного APK

Read: /workspace/skills/reverse_playbook.md

Read: /workspace/skills/apk-reverse.md

Модель читает наши playbooks для работы с такими файлами

```
Bash: cd /workspace/Android-CecBot && jadx --show-bad-code --deobf -d jadx cecbot.apk 2>&1 | tail -30
Bash: cd /workspace/Android-CecBot && echo "=== jadx top ===" && ls -la jadx/ && echo "=== jadx sources ===" && find jadx/sources -maxdepth 3 -type d 2>/dev/null | head -50 && echo "=== jadx class count ===" && find jadx/sources -name '*.java' 2>/dev/null | wc -l
Bash: cd /workspace/Android-CecBot && echo "=== apktool top ===" && ls -la apktool/ && echo "=== smali dirs ===" && ls -d apktool/smali* 2>/dev/null && echo "=== lib ===" && find apktool/lib -type f 2>/dev/null && echo "=== assets ===" && find apktool/assets -type f 2>/dev/null
```

Часть процесса распаковки

Ответ модели:

Нативные `.so` (4 шт., отдельно в `struct.md`):

ПУТЬ	ARCH	ELF	SIZE
<code>lib/arm64-v8a/libattack.so</code>	aarch64	shared object, stripped	101 344 B
<code>lib/arm64-v8a/libwatchdog.so</code>	aarch64	PIE exec (linker64), stripped	7 896 B
<code>lib/armeabi-v7a/libattack.so</code>	ARM EABI5	shared object, stripped	104 656 B
<code>lib/armeabi-v7a/libwatchdog.so</code>	ARM EABI5	PIE exec (linker), stripped	5 808 B

Ключевые наблюдения структуры:

- `libwatchdog.so` — не JNI-библиотека, а standalone PIE-исполняемый файл с interpreter (`/system/bin/linker*`), т.е. запускается отдельным процессом.
- Обфускация: default-пакет (80 smali) с короткими именами → после `jadx --deobf` перенесён в `p000` и переименован (`Cxxxx` , `AbstractCxxxx` , `RunnableCxxxx`).
- Именованные классы (39): `BotService` , `AttackEngine` , `GuardService` , `CommandBlocker` , `DeviceLockdown` , `AntiAnalysis` , `SelfDestruct` , 17× `Persist*` , крипто `crypto/{Ed25519, HKDF, X25519}` .
- JADX завершился с 2 ошибками распаковки (типично для обфусцированных ссылок/типов).

Файл отчёта: `/workspace/Android-CecBot/struct.md` (267 строк).

Модель дала краткую информацию и создала наш `struct.md`

Часть отчёта struct.md:

```
## 2. Рабочая раскладка ~/workspace/Android-CecBot/`
```

```

Android-CecBot/
├── cecbot.apk          # исходный APK
├── cecbot.md           # прежний аналитический отчёт (идентификация/вердикт)
├── struct.md          # настоящий файл
├── original/
│   └── cecbot.apk      # копия исходного APK (провенанс)
├── unzip/             # сырые байты из ZIP (без декодирования)
├── apktool/           # декодированный манифест + smali + res + lib
└── jadx/              # деобфусцированный псевдокод (Java) + ресурсы

```

```
## 3. Структура распакованного APK (сырой вид, unzip/`)
```

```

unzip/
├── AndroidManifest.xml
├── classes.dex
├── resources.arsc
├── META-INF/
│   ├── DEBUG.RSA
│   ├── DEBUG.SF
│   └── MANIFEST.MF
├── lib/
│   ├── arm64-v8a/
│   │   ├── libattack.so
│   │   └── libwatchdog.so
│   ├── armeabi-v7a/
│   │   ├── libattack.so
│   │   └── libwatchdog.so
└── res/

```

Теперь можем перейти в новый чат, чтобы очистить контекст от лишнего и уже начать полноценную работу, так как документацию по APK мы уже создали

Процесс реверса

/workspace/Android-CecBot/struct.md - структура проекта

/workspace/Android-CecBot/cecbot.md - краткая информацию по APK

Проведи подробный реверс данного файла.

Опиши подробно, что делает данный APK, так же попробуй раскрыть C2.

Для динамики можешь использовать adb у себя в среде, тебе доступен emulator.

Если потребуется патчить проверки анти-root и.т.п, то все инструменты есть.

Ставим задачу модели

Read: /workspace/Android-CecBot/jadx/sources/com/imjustafriendlyguy/firmware/AttackEngine.java

Read: /workspace/Android-CecBot/jadx/sources/com/imjustafriendlyguy/firmware/Watchdog.java

Bash:

```
bash
1 cd /workspace/Android-CecBot && cat > /tmp/decrypt_strings.py << 'PYEOF'
2 import re, json
3
4 src = open('jadx/sources/com/imjustafriendlyguy/firmware/C0034S.java').read()
5
6 # key
7 keym = re.search(r'f186a = \{([^\}]+)\}', src)
8 key = [int(x.strip()) for x in keym.group(1).split(',')]
9 ...
```

Процесс работы модели

Grep: dns-query|1\.\.1\.\.1|8\.\.8\.\.8|9\.\.9\.\.9|onion|\.tor|SocksPort|DataDirectory|DisableNetwork|Bootstrapped|tor/

Grep: m171g\(|C0034S|S\.m171g|\.\.m171g

Взаимодействие с эмулятором

Bash: adb -s emulator-5554 shell am start -n com.imjustafriendlyguy.firmware/.MainActivity 2>&1; sleep 4; echo "=== process check ==="; adb -s emulator-5554 shell "ps -A | grep -i friendly" 2>&1; echo "=== self-destruct markers ==="; adb -s emulator-5554 shell "ls -la /data/local/tmp/.sd_* 2>/dev/null; ls..."

Bash: adb -s emulator-5554 logcat -d 2>&1 | grep -iE "imjustafriendly|firmware|BotService|AntiAnalysis|MainActivity|AndroidRuntime|FATAL|Process.*died|killProcess" | tail -60

Ответ модели:

Вердикт

Android DDoS-ботнет CecBot (вариант Mirai), пакет `com.imjustafriendlyguy.firmware`, маскируется под `System Update`. Нацелен на Android TV/IoT (`LEANBACK_LAUNCHER`, CEC/HDMI-управление, отсутствие UI).

C2 — раскрыто

Seed-домены (обфусцированы XOR+Base64 в `C0029a0.f89g`, порт 6969, fallback 443/80/8080/8443):

ДОМЕН	ДЕКОД
<code>HMS1iyTk...X3rA==</code>	<code>okiloveyoupleasedonttouchme.net</code>
<code>B815lDvz...3MaA==</code>	<code>thespacemachines.st</code>
<code>H8pqgi3g...3EaA==</code>	<code>lovefromallareas.at</code>

Все три резолвятся в один пул из 6 живых IP:

⌵ Свернуть Сохранить Копировать

1	36.255.97.37	64.89.160.15	62.60.226.173
2	130.12.181.199	94.154.43.235	217.60.103.96

Tor onion C2 (bootstrap конфига): `c2kxpjr7cux7fqrfmimsz7rtq527xauw627xrjojimt66nwxqvrqbuyd.onion`, путь `/c`, через SOCKS5. В этом APK `libtor.so` отсутствует (урезанный вариант) — путь через Tor неактивен.

Крипто-ключи C2 (собственная реализация X25519+Ed25519+HKDF+ChaCha20-Poly1305):

⌵ Свернуть Сохранить Копировать

1	Ed25519 pubkey:	<code>38b4b69cdcd338e81bae5c09af0aa35055674e9277c347eadc74fca3a95a9b5a</code>
2	salt:	<code>d41cafbae4000c3fdb911cd91b232acb7f4152b00a26bc9550ae0a4acfd66d65</code>
3	config key:	<code>9fe8224b5c23c7faed9028b71b544a46e28307c2d1cfc30702dc065a41792e4e</code>

Протокол

Handshake (Noise-NK гибрид): бот шлёт `BEEF00 + X25519_pub + SHA256(salt)` (67 байт), сервер отвечает `BEEF00 + pub + Ed25519_sig` (99 байт) → ECDH → HKDF → ChaCha20-Poly1305. Фрейминг `BEEF 02 flags seq len`. Команды: 1/2/3 (атаки), 16 (self-destruct), 17/19 (shell), 18 (ротация ключа), 48/49 (C2-управление), 32/34 (сбор инфо), 33-44 (ADB/CEC), 255 (ping).

Динамика (эмулятор API 35)

- Без патча: `AntiAnalysis` детектит эмулятор → самоликвидация → цикл «watchdog → старт → гибель» (подтверждено в logcat).
- С патчем (`check()` → `return true`): бот зарезолвил домены, протупал все 6 IP, подключился к `36.255.97.37:6969`, выполнил хендшейк и установил живую зашифрованную сессию — полный обмен зафиксирован в rсар.

Потребовался патч APK, чтобы обойти защиту самоликвидации

3.2 Живые C2 IP (резолв всех трёх доменов → общий пул)

Все три домена резолвятся в ****один и тот же набор из 6 IP**** (round-robin / fast-flux):

...

```
36.255.97.37      64.89.160.15      62.60.226.173
130.12.181.199    94.154.43.235      217.60.103.96
...
```

****Подтверждено динамически****: бот параллельно прощупывает все 6 IP по 6969, выбирает самый ёмкий C2-сессию (см. п.6).

3.3 Tor onion C2 (bootstrap-конфиг)

- Onion (v3): `c2kxpjr7cux7fqrfmimsz7rtq527xauw627xrjojimt66nwxqvrqbuyd.onion`
- Путь: `GET /c` через SOCKS5 `127.0.0.1:9050`.
- Запуск Tor: `libtor.so --SocksPort 9050 --DataDirectory <files>/td --DisableNetwork 0 --L`
- Ответ – зашифрованный список C2-серверов (nonce12 + ChaCha20-Poly1305). Формат: `[0x01][ti`
`str}...]`.
- ****Примечание****: `libtor.so` ****отсутствует**** в данном APK (только `libattack.so` + `libwat`
через Tor в нём не работает, используется только seed-список.

Модель создала отчёт в нашем
проекте, включая:
- Команды C2
- DDoS-движок (libattack.so)

5. Протокол команд C2 (C0018P.m57a)

Cmd	Назначение	Ответ
1	Запуск атаки (id, длительность 1..86400с, тип, цели IP+mask,	
2	Останов атаки (id)	130
3	Статус атак	131
16	Self-destruct	-
17	Shell exec (без ответа)	-
18	Ротация C2 Ed25519 ключа/salt	-
19	Shell exec (с ответом)	132
20-24	Управление модулем <code>C0064r0</code> (скриншот/файл)	135
32/34	Сбор информации об устройстве	133
33/35-44	ADB-over-TCP / TV CEC/HDMI управление	143
48	Добавить C2-серверы (<code>d_prefs.pd</code>)	-
49	Переподключение	-
64-70	CEC/HDMI команды (ответ 137)	137
80-83	Сеть / скан	138-142
98	Установить флаг (<code>f129y</code>)	-
255	PING/keepalive	129

HELLO (cmd 128) шлёт: пакет, версия, ABI, SDK, модель, производитель, fingerprint.

6. DDoS-движок (libattack.so)

Нативный движок с JNI-экспортом `Java_com_imjustafriendlyguy_firmware_AttackEngine_native*`
flags[], options[]`)`. Флуд-векторы, выявленные по строкам:

- ****HTTP GET/POST flood****: `GET /info.json HTTP/1.1`, `POST /client HTTP/1.1`, полные заголо
- ****HTTPS flood + referer-спуфинг****: встроенный список ~400 популярных доменов (google/amazo
- ****TLS ClientHello / JA3-спуфинг****: UA Chrome/Firefox/Edge/Safari/Brave/Android/iOS, GREASE
- ****SSH banner flood****: `SSH-2.0-PuTTY_Release_0.79..0.83`.
- ****SMTP/IRC flood****: `EHLO`, `HELO`, `NICK`, `USER`, `PASS guest@`, `LUSERS`, `PING :keepal`
- ****IMAP flood****: `CAPABILITY`, `NOOP`, `VERSION`, `getstatus`, `stat`.
- ****FTP flood****: `USER anonymous`, `USER ftp`; ****MySQL****: `mysql_native_password`.
- ****Спуфинг UA****: `User-Agent: curl/7.83.1-DEV`, `CitizenFX`.

Корректировка направления работы модели

Ampire 

Возможно ли получить доступ до 36.255.97.37:6969
Есть ли уязвимость в данном C2?

Попробуем исследовать
данный C2. Делаем запрос
модели

```
1 cat > /workspace/Android-CecBot/c2_client.py << 'PYEOF'
2 import socket, sys, hashlib, time
3 from cryptography.hazmat.primitives.asymmetric.x25519 import X25519PrivateKey, X25519PublicKey
4 from cryptography.hazmat.primitives.asymmetric.ed25519 import Ed25519PublicKey
5 ...

Bash: cd /workspace/Android-CecBot && python3 c2_client.py 36.255.97.37 6969 4 2>&1

Bash: cd /workspace/Android-CecBot && sed -i 's/shared = priv.exchange(srv_pub)/shared =
priv.exchange(X25519PublicKey.from_public_bytes(srv_pub))/' c2_client.py && python3 c2_client.py 36.255.97.37 6969 5 2>&1

Bash: cd /workspace/Android-CecBot && echo "=== nmap 36.255.97.37 (bot ports + common) ===" && timeout 90 nmap -Pn -
sT --open -p
21,22,23,25,53,80,110,143,443,465,587,993,995,2222,3306,3389,5432,6379,6969,8000,8080,8081,8443,8888,9000,9090
36.255.97.37 2>&1 | grep -vE "^Starting|^Nmap done|^Not shown|^Sta..."
```

Модель активно работает
над запросом

Страна:  Филиппины

IP диапазон: 36.255.97.0 - 36.255.97.255

CIDR: 36.255.97.0/24

Название провайдера: Cyber-Security-PH

Параллельно сами
посмотрим на провайдера
IP, название интересное

Ответ модели:

2. Главная уязвимость — боты не аутентифицируются

Handshake односторонний (Noise NK): сервер доказывает личность Ed25519, но сервер не проверяет бота. Любой, кто восстановил клиентскую часть (а я её восстановил), может:

- подключиться и «стать ботом»;
- получать командный поток C2;
- слить всю инфраструктуру и текущие цели.

Криптослой корректен (уязвимостей в шифровании нет) — дыра на уровне доверия.

3. Что утекло (реальные данные из живой сессии)

Команда 48 — ротация C2-доменов (8 доменов, 5 новых):

```
⌵ Свернуть  Сохранить  Копировать
1 dnsrecordsarepowerful.com      iloveuevenifudontlovemeback.is
2 iwonderifulovemeaswell.vg      sdkconnect121.st
3 sdkconnector.com
4 + lovefromallareas.at / thespacemachines.st / okiloveyoupleasedonttouchme.net

Все 8 резолвятся в тот же пул из 6 IP ( 36.255.97.37 64.89.160.15 62.60.226.173 130.12.181.199 94.154.43.235
217.60.103.96 ).
```

Модель может раскрыть все ротационные C2 через команду и их текущие цели

Команда 1 — живая DDoS-атака, которую C2 отправил моему фейковому боту:

```
⌵ Свернуть  Сохранить  Копировать
1 attack_id=151561 duration=60s type=3 target=131.196.31.0/24 opt=07:"0"

(в первом сеансе — 131.196.29.0/24 ). C2 прямо сейчас ведёт DDoS по 131.196.x.0/24 (LACNIC/Бразилия), тип 3 — один из флуд-векторов libattack.so .

Команда 255 — ping/keepalive.
```

Также видно текущую их цель Ddos
Они ведут атаки на подсеть Бразилии - 131.196.x.0/24

Backend-IP (fast-flux: все 8 доменов → один пул из 6 IP)

IP	ЛОКАЦИЯ	ХОСТИНГ
36.255.97.37	US, San Jose	AS205759 Ghosty Networks LLC
64.89.160.15	LU, Schieren	AS205759 Ghosty Networks LLC
62.60.226.173	DE, Frankfurt	AS214351 Femo IT Solutions
130.12.181.199	SI, Ljubljana	AS197769 VPS Dedicated LLC
94.154.43.235	NL, Amsterdam	AS219502 Storm Industries LLC
217.60.103.96	CH, Bern	AS209373 Swissnet LLC (pfcloud)

Все 6 — **bulletproof/офшорные VPS** в 6 юрисдикциях (Ghosty, Femo IT, VPS Dedicated, Storm Industries, Swissnet/pfcloud).

Крипто-материал C2

Свернуть
Сохранить
Копировать

```

1 Ed25519 pubkey: 38b4b69cdcd338e81bae5c09af0aa35055674e9277c347eadc74fca3a95a9b5a
2 salt:          d41cafb4e4000c3fdb911cd91b232acb7f4152b00a26bc9550ae0a4acfd66d65
3 config key:    9fe8224b5c23c7faed9028b71b544a46e28307c2d1cfc30702dc065a41792e4e

```

Продолжили исследование

Текущие цели, итоги

Текущие цели (живая DDoS-кампания)

C2 отдал команду атаки **тип 3, 60 с** на мой фейковый бот. Цели:

ЦЕЛЬ	ЖЕРТВА
131.196.29.0/24	AS271410 Smart Serviço de Internet Ltda (Бразилия, São Paulo)
131.196.31.0/24	AS271410 (тот же ISP)

C2 ведёт целенаправленный DDoS по конкретному бразильскому ISP (AS271410), перебирая /24-подсети диапазона 131.196.0.0/16 .

а получить базу данных из C2 реально или нету уязвимости там?

Ответ модели:

Почему база недостижима

1. **Протокол C2 (6969) не даёт доступа к БД.** Бот — чистый клиент: он *принимает* команды (1/2/3 атака, 17/19 shell, 48 ротация доменов, 255 ping...), но у него **нет ни одной команды-запроса к бэкенду C2**. Команд «отдай данные БД» в протоколе не существует.
2. **DB-порты — tarpit.** Реальный MySQL/PostgreSQL/Redis недоступен (вероятно, слушает только на internal-интерфейсе или разрешён только с IP оператора через firewall).
3. **Web-панель — tarpit.** Порт 80 и высокие порты (8000–9090) не отдают контент. Если у оператора и есть веб-админка ботнета, она не торчит наружу.
4. **SSH — без RCE.**

Есть ли уязвимости в БД?

Единственный теоретический вектор (не подтверждён)

Единственное, что *гипотетически* могло бы быть — SQL-инъекция в обработчике HELLO на стороне C2 (поля device id / SSID / model идут в бэкенд). Но:

- это не подтверждается чёрным ящиком (C2 шлёт ответы только через зашифрованный канал, SQL-ошибки наружу не видны);
- требует длительного слепого фаззинга без наблюдаемого эффекта;
- в Mirai-подобных C2 эти поля обычно пишутся prepared-запросами.

Теория модели, как можно было бы

Благодарим за внимание

Кристина Стрельцова

Ведущий специалист
Перспективный мониторинг

Игорь Васильев

Ведущий специалист
Перспективный мониторинг

