6. WYNIKI ANALIZY TELEFONU SAMSUNG S20 FE

6.1 Ustalenia analizy offline

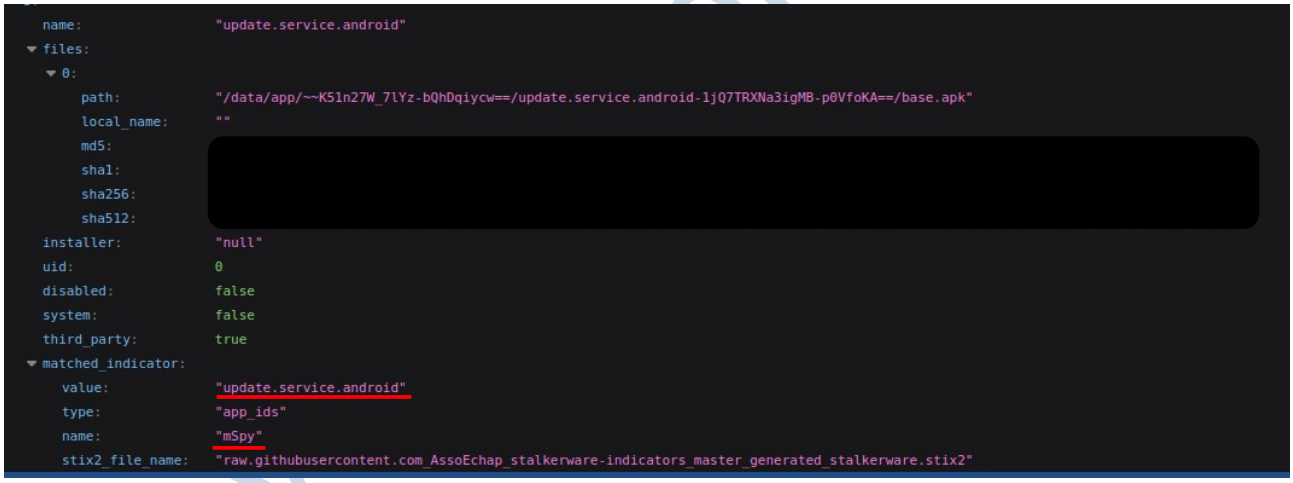
Ujawniono aplikację służącą do kontroli telefonów z systemem Android – '***mSpy'***. Aplikacja ta widnieje w systemie badanego telefonu pod zmienioną nazwą '***update.service.android***' / '***Update Service'*** – co jest znaną strategią maskującą jej obecność oraz w zamyśle mającą utrudnić wykrycie.

Na podstawie pobranych danych oraz analizy statycznej aplikacji ustalono podstawowe dane:

package_name (nazwa pakietu aplikacji)	**app_name** (nazwa aplikacji)	**file_name** (nazwa pliku instalacyjnego)	**UID** (nr systemowy)
update.service.android	Update Service	update.service.android_update.service.android-1jQ7TRXNa3igMB-p0VfoKA.apk	10340

A. Potwierdzenie faktu, że ujawniona w systemie aplikacja '*Update Service'*, to w rzeczywistości aplikacja szpiegująca '*mSpy'*:

1) dopasowanie wskaźników IOCS przez oprogramowanie śledcze MVT (obraz 1)



Obraz 1 – raport oprogramowania MVT z widocznym dopasowaniem wskaźników zainfekowania 'mSpy'

Końcowy wynik skanowania pobranych danych (obraz 2)

Obraz 2 – końcowy wynik skanowania MVT

2) Zdefiniowana nazwa stałego typu ustawień: „*Enable mSpy monitoring!*” – po dekompilacji pakietu aplikacji ‘*Update Service*’ (update.service.android)

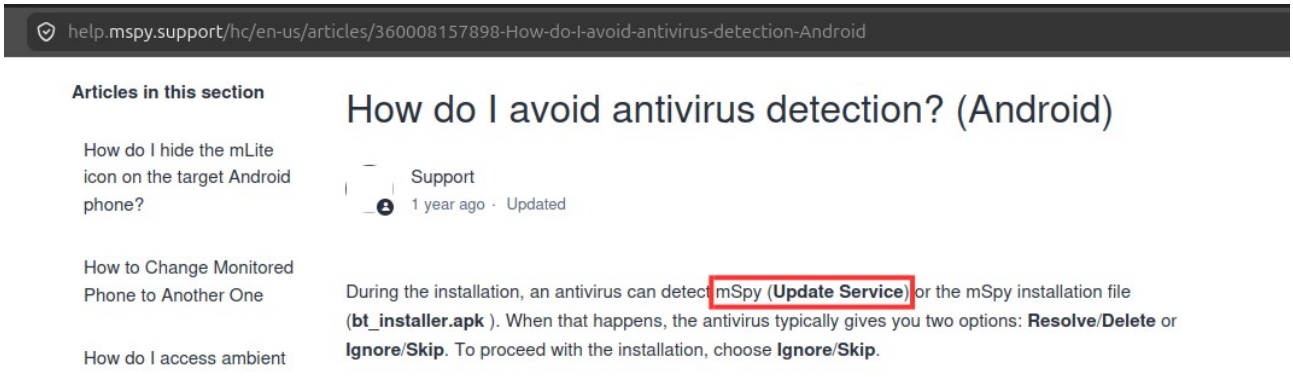
```
/* JADX WARN: Failed to restore enum class, 'enum' modifier and super class removed */
/* JADX WARN: Unknown enum class pattern. Please report as an issue! */
/* compiled from: SettingType.kt */
@Metadata(d1 = {"\u0000\u0012\n\u0002\u0018\u0002\n\u0002\u0010\u0010\n\u0000\n\u0002\u0010\u000e\n\u0002\b\u0012\b\u0086\u0081\u0002\u0018\u00002\b\u
/* loaded from: classes6.dex */
public final class SettingType {
    private static final /* synthetic */ EnumEntries $ENTRIES;
    private static final /* synthetic */ SettingType[] $VALUES;
    private final String jsonName;
    public static final SettingType Additional = new SettingType("Additional", 0, "Make the app always active");
    public static final SettingType Autostart = new SettingType("Autostart", 1, "Enable Update Service");
    public static final SettingType Accessibility = new SettingType("Accessibility", 2, "Enable messages tracker");
    public static final SettingType Location = new SettingType(HttpHeaders.LOCATION, 3, "Location access");
    public static final SettingType RunTimePermissions = new SettingType("RunTimePermissions", 4, "RunTimePermissions");
    public static final SettingType DrawLayout = new SettingType("DrawLayout", 5, "Enable mSpy monitoring!");
    public static final SettingType UsageAccess = new SettingType("UsageAccess", 6, "Enable call tracking");
    public static final SettingType BackgroundDataUsage = new SettingType("BackgroundDataUsage", 7, "Allow background data usage");
    public static final SettingType BatteryOptimization = new SettingType("BatteryOptimization", 8, "Make the app undetectable");
    public static final SettingType Admin = new SettingType("Admin", 9, "Prevent detection by antivirus");
    public static final SettingType Record = new SettingType("Record", 10, "Enable social media tracking");
    public static final SettingType Notification = new SettingType("Notification", 11, "Prevent uninstallation");
    public static final SettingType NotificationListener = new SettingType("NotificationListener", 12, "Enable notification listener");
```

Obraz 3 – nazwy stałego typu ustawień widoczne w Jadx

Widoczne powyżej (obraz 3) zdefiniowane ustawienia, są charakterystyczne dla aplikacji szpiegujących, m.in.:

`"Enable mSpy monitoring!"`	włącz monitoring mSpy
`"Enable social media tracking"`	włącz śledzenie mediów społecznościowych
`"Enable call tracking"`	włącz śledzenie połączeń
`"Make the app undetectable"`	uczyń aplikacje niewykrywalą
`"Enable messages tracker"`	włącz śledzenie wiadomości
`"Prevent detection by antivirus"`	zapobiegaj wykryciu przez antywirus
`"Prevent uninstallation"`	zapobiegaj odinstalowaniu
`"Allow background data usage"`	pozwól na użycie danych w tle

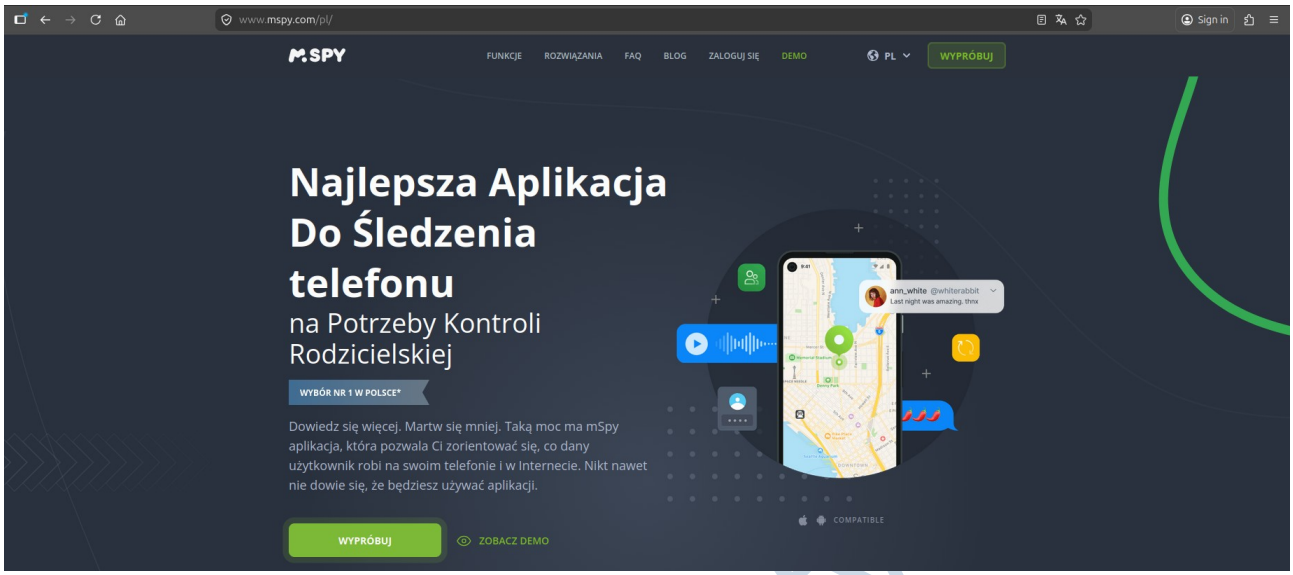
3) Użycie nazw 'mSpy' oraz 'Update Service' jako wymienne, na oficjalnej stronie mSpy Help Center (*https://help.mspy.support/hc/en-us*): "*During the installation, an antivirus can detect mSpy (Update Service) or the mSpy installation file (bt_installer.apk).*"



Obraz 4 – strona internetowa mSpy Help Center z widocznym określeniem aplikacji ‘mSpy’ jako ‘Update Service’

B. Wnioski po analizie offline

1) W systemie analizowanego telefonu komórkowego jest zainstalowana aplikacja ‘*Update Service*’ – która w rzeczywistości jest aplikacją ‘*mSpy*’ występującą pod zmienią nazwą.



Obraz 5 – strona internetowa dostawcy aplikacji mSpy: www.mspy.com/pl/

2) Wskaźniki zainfekowania (IOCS) wskazują na oprogramowanie typu „**stalkerware**”.

Definicja oprogramowania „stalkerware” ze strony *https://stopstalkerware.org/* (organizacji Coalition Against Stalkerware – międzynarodowej grupy roboczej ds. cyber-stalkingu):

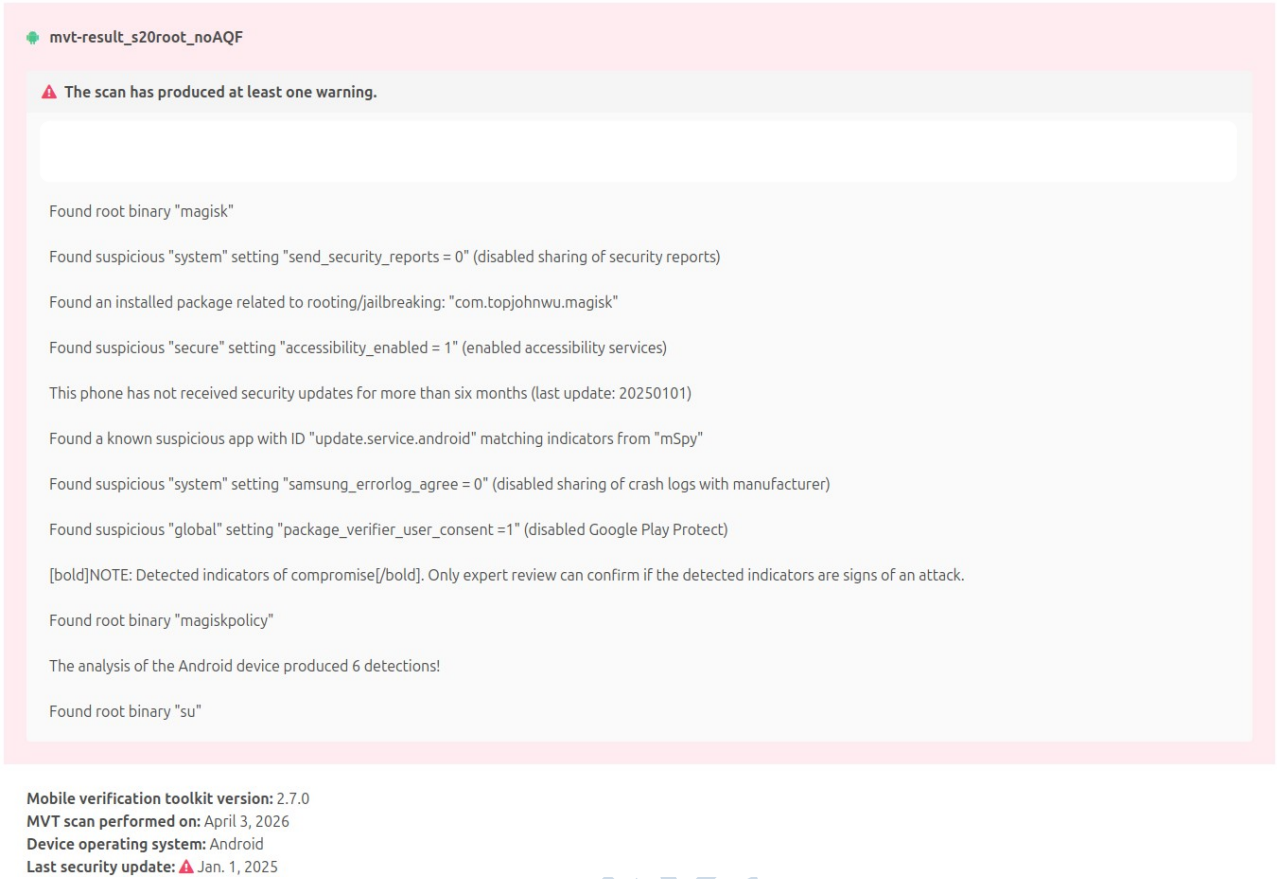
„Termin „stalkerware” odnosi się do narzędzi – programów, aplikacji i urządzeń – które umożliwiają ukryte szpiegowanie życia prywatnego innej osoby za pośrednictwem jej urządzenia mobilnego. Sprawca może zdalnie monitorować całe urządzenie, w tym wyszukiwanie w internecie, geolokalizację, wiadomości tekstowe, zdjęcia, rozmowy głosowe i wiele innych. Takie programy są łatwe w zakupie i instalacji. Działają one w tle, bez wiedzy i zgody ofiary. Niezależnie od dostępności oprogramowania stalkerware, sprawca ponosi odpowiedzialność za jego użycie jako narzędzia, a tym samym za popełnienie przestępstwa.”

3) Zidentyfikowane pozostałe **naruszenia bezpieczeństwa** analizowanego telefonu:

- wyłączona funkcja *Google Play Protect*
- zainstalowana funkcja dostępności (accessibility)
- wykryta ingerencja w system – root binary (magisk)
- brak aktualizacji systemowej powyżej 6 miesięcy,

są charakterystyczne dla aplikacji monitorujących, które ukrywają swoją obecność przed użytkownikiem oraz wymuszają wyłączenie podstawowych funkcji zabezpieczeń systemu.

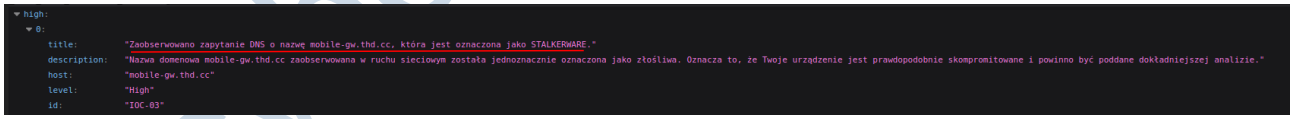
Obrazuje to poniższa (obraz 6) wizualizacja wyników (MVT-companion):



Obraz 6 – raport analizy MVT z widocznymi naruszeniami bezpieczeństwa badanego telefonu

6.2 Ustalenia analizy live

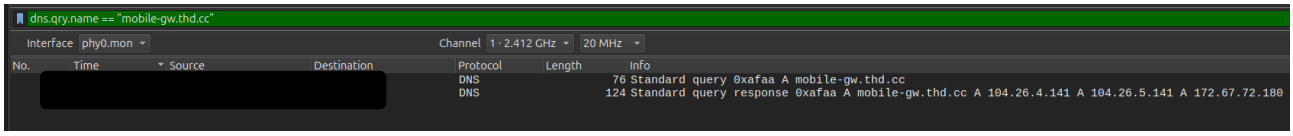
Ujawniono komunikację sieciową – zapytanie DNS – anlizowanego telefonu z domeną ***mobile-gw.thd.cc***



Obraz 7 – raport analizy ruchu sieciowego z widocznym oznaczeniem domeny mobile-gw.thd.cc jako stalkerware

- wskazana domena została oznaczona przez wskaźniki zainfekowania (IOCS) jako złośliwa – "stalkerware" – podobnie jak wykryta aplikacja '*Update Service*'.

Na podstawie szczegółowej analizy wyników ustalono, że podczas komunikacji sieciowej wychodzącej z badanego telefonu, serwer DNS zwrócił 3 adresy IP: *104.26.4.141 / 104.26.5.141 / 172.67.72.180* przypisane dla domeny *mobile-gw.thd.cc*, co jest widoczne na poniższej grafice (obraz 8).

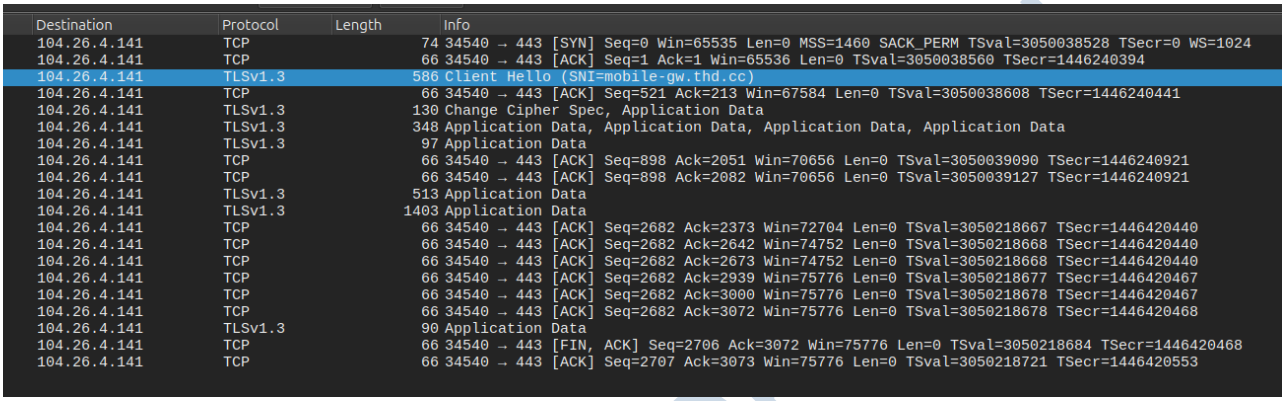


Obraz 8 – szczegółowa analiza pakietów z widoczną komunikacją pomiędzy badanym telefonem a domeną mobile-gw.thd.cc

Wnioski po analizie live

Stwierdzono **przesył danych** z analizowanego urządzenia na **zewnętrzy serwer** powiązany z domeną ***mobile-gw.thd.cc***

Dalsze szczegółowe sprawdzenie ujawnionej komunikacji sieciowej wykazało, że badany telefon (host [redacted] – adres prywatny IPv4, urządzenie w sieci lokalnej) nawiązał szyfrowane połączenie HTTPS z zewnętrznym serwerem (IP *104.26.4.141*, domena *mobile-gw.thd.cc).* Po ustanowieniu szyfrowanego kanału nastąpił transfer danych w postaci zaszyfrowanych rekordów *Application Data*. Sesja została następnie poprawnie zakończona poprzez zamknięcie połączenia TCP. Całość widoczna poniżej (obraz 9):



Destination	Protocol	Length	Info
104.26.4.141	TCP	74	34540 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=3050038528 TSecr=0 WS=1024
104.26.4.141	TCP	66	34540 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3050038560 TSecr=1446240394
104.26.4.141	TLSv1.3	586	Client Hello (SNI=mobile-gw.thd.cc)
104.26.4.141	TCP	66	34540 → 443 [ACK] Seq=521 Ack=213 Win=67584 Len=0 TSval=3050038608 TSecr=1446240441
104.26.4.141	TLSv1.3	130	Change Cipher Spec, Application Data
104.26.4.141	TLSv1.3	348	Application Data, Application Data, Application Data, Application Data
104.26.4.141	TLSv1.3	97	Application Data
104.26.4.141	TCP	66	34540 → 443 [ACK] Seq=898 Ack=2051 Win=70656 Len=0 TSval=3050039090 TSecr=1446240921
104.26.4.141	TCP	66	34540 → 443 [ACK] Seq=898 Ack=2082 Win=70656 Len=0 TSval=3050039127 TSecr=1446240921
104.26.4.141	TLSv1.3	513	Application Data
104.26.4.141	TLSv1.3	1403	Application Data
104.26.4.141	TCP	66	34540 → 443 [ACK] Seq=2682 Ack=2373 Win=72704 Len=0 TSval=3050218667 TSecr=1446420440
104.26.4.141	TCP	66	34540 → 443 [ACK] Seq=2682 Ack=2642 Win=74752 Len=0 TSval=3050218668 TSecr=1446420440
104.26.4.141	TCP	66	34540 → 443 [ACK] Seq=2682 Ack=2673 Win=74752 Len=0 TSval=3050218668 TSecr=1446420440
104.26.4.141	TCP	66	34540 → 443 [ACK] Seq=2682 Ack=2939 Win=75776 Len=0 TSval=3050218677 TSecr=1446420467
104.26.4.141	TCP	66	34540 → 443 [ACK] Seq=2682 Ack=3000 Win=75776 Len=0 TSval=3050218678 TSecr=1446420467
104.26.4.141	TCP	66	34540 → 443 [ACK] Seq=2682 Ack=3072 Win=75776 Len=0 TSval=3050218678 TSecr=1446420468
104.26.4.141	TLSv1.3	90	Application Data
104.26.4.141	TCP	66	34540 → 443 [FIN, ACK] Seq=2706 Ack=3072 Win=75776 Len=0 TSval=3050218684 TSecr=1446420468
104.26.4.141	TCP	66	34540 → 443 [ACK] Seq=2707 Ack=3073 Win=75776 Len=0 TSval=3050218721 TSecr=1446420553

Obraz 9 – analiza pakietów z widoczną komunikacją pomiędzy badanym telefonem a domeną mobile-gw.thd.cc (IP 104.26.4.141)

Analizowane urządzenie wyeksportowało dane na serwer zewnętrzny, przy użyciu szyfrowanego kanału komunikacji sieciowej. Komunikacja odbyła się z domeną (mobile-gw.thd.cc) zidentyfikowaną jako ‘stalkerware’.

6.3 Korelacja wyników analizy offline + live

Stwierdzono zgodność pomiędzy artefaktami ujawnionymi w analizie offline a aktywnością sieciową zarejestrowaną podczas analizy live. Na urządzeniu jest zainstalowana aplikacja mogąca inwigilować użytkownika oraz aktywnie przesyła dane (przez internet) poza urządzenie.

Poniżej wykazano związek pomiędzy zainstalowaną aplikacją ***Update Service*** (mSpy) a domeną ***mobile-gw.thd.cc***, z którą połączył się telefon.

Offline	**Live**	**Korelacja**
Ujawniono pakiet aplikacji o nazwie ***update.service.android*** - sklasyfikowany jako "**mSpy**"	Zaobserwowano zapytanie DNS o nazwę ***mobile-gw.thd.cc***, która jest oznaczona jako **STALKERWARE**	Domena, z którą łączył się telefon, jest zdefiniowana w plikach aplikacji '*Update Service*' (obraz 10)
• Na podstawie analizy statycznej w postaci dekompilacji pakietu aplikacji '*Update Service*' (update.service.android) wykazano bezpośredni związek między aplikacją a domeną '*https://mobile-gw.thd.cc/*'. • Widoczna na poniższej grafice (obraz 10) linia '*public static final String BASE_URL = "https://mobile-gw.thd.cc/";*' informuje o głównym adresie serwera, z którym komunikuje się aplikacja.		

BuildConfig

```
1  package update.service.core;
2
3  /* loaded from: classes6.dex */
4  public final class BuildConfig {
5      public static final String BASE_URL = "https://mobile-gw.thd.cc/";
6      public static final String BUILD_TYPE = "release";
7      public static final boolean DEBUG = false;
8      public static final String LIBRARY_PACKAGE_NAME = "update.service.core";
9      public static final String PATCH_CERTIFICATE = "/sdcard/charles-ssl-proxying-certificate.pem";
10 }
```

Obraz 10 – dekompilacja pakietu aplikacji 'update.service.android' z widocznym zdefiniowanym adresem url (BASE_URL = "https://mobile-gw.thd.cc/"

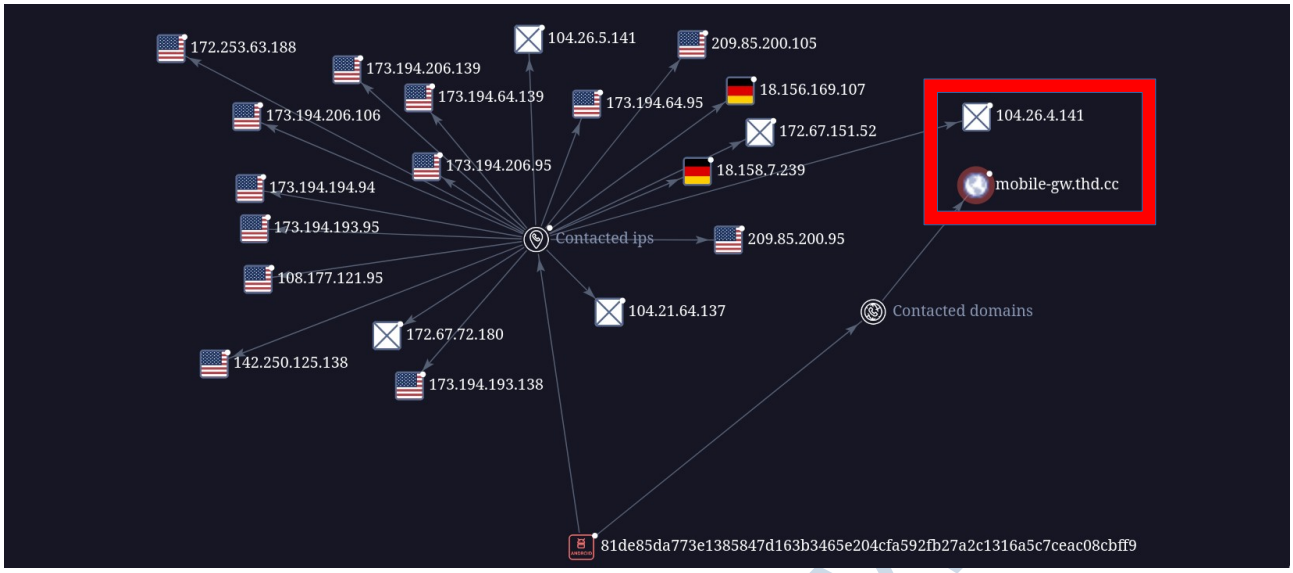
Na tej podstawie można stwierdzić, że wskazana domena jest integralną częścią działania aplikacji i bez niej aplikacja niem może działać poprawnie. Adres ten został odnaleziony w klasie konfiguracyjnej *BuildConfig*, co wskazuje, że stanowi on główny punkt komunikacji aplikacji z infrastrukturą serwerową.

Przeszukanie sieci pod kątem nazwy domeny *mobile-gw.thd.cc* oraz adresu *IP 104.26.4.141* potwierdza powiązania wymienionych danych domeny oraz adresu IP z aplikacją mSpy. Poniżej opracowanie opublikowane przez *The Centre for Internet and Society (CIS)* pod tytułem "*India's parental control directive and the need to improve stalkerware detection*"

Name	Domain	IP Address[26]	Country	ASN Name and Number
SafeNet	safenet.family	103.10.24.124	India	Amrita Vishwa Vidyapeetham, AS58703
OneMonitar	onemonitar.com	3.15.113.141	United States	Amazon.com, Inc., AS16509
OneMonitar	api.cp.onemonitar.com	3.23.25.254	United States	Amazon.com, Inc., AS16509
Hoverwatch	hoverwatch.com	104.236.73.120	United States	DigitalOcean, LLC, AS14061
Hoverwatch	a.syncvch.com	158.69.24.236	Canada	OVH SAS, AS16276
TheTruthSpy	thetruthspy.com	172.67.174.162	United States	Cloudflare, Inc., AS13335
TheTruthSpy	protocol-a946.thetruthspy.com	176.123.5.22	Moldova	ALEXHOST SRL, AS200019
Cerberus	cerberusapp.com	104.26.9.137	United States	Cloudflare, Inc., AS13335
mSpy	mspy.com	104.22.76.136	United States	Cloudflare, Inc., AS13335
mSpy	mobile-gw.thd.cc	104.26.4.141	United States	Cloudflare, Inc., AS13335
FlexiSPY	flexispy.com	104.26.9.173	United States	Cloudflare, Inc., AS13335
FlexiSPY	djp.bz	119.8.35.235	Hong Kong	HUAWEI CLOUDS, AS136907

Obraz 11 – https://cis-india.org/internet-governance/blog/india2019s-parental-control-directive-and-the-need-to-improve-stalkerware-detection

Korelacja na podstawie wyników analizy (obraz 12) pakietu aplikacji 'update.service.android_update.service.android-1jQ7TRXNa3igMB-p0VfoKA.apk' w serwisie Virus Total (https://www.virustotal.com/)



Obraz 12 – Virus Total Graph

7. USTALENIE ZAKRESU INWIGILACJI

Uwaga! Konieczny dostęp 'root'

7.1 Analiza offline – przeglądanie zawartości bazy danych SQLite aplikacji *Update service*

SQLite to relacyjna baza danych używana jako główny magazyn danych dla instalowanych w systemie aplikacji. Składa się z tabel, a dane w nich przechowywane nie są (z reguły) szyfrowane. Uzyskując dostęp do powłoki telefonu (shell) przy użyciu narzędzia 'adb', uzyskano dostęp do bazy danych SQLite pakietu aplikacji update.serivce.android (obraz 13)

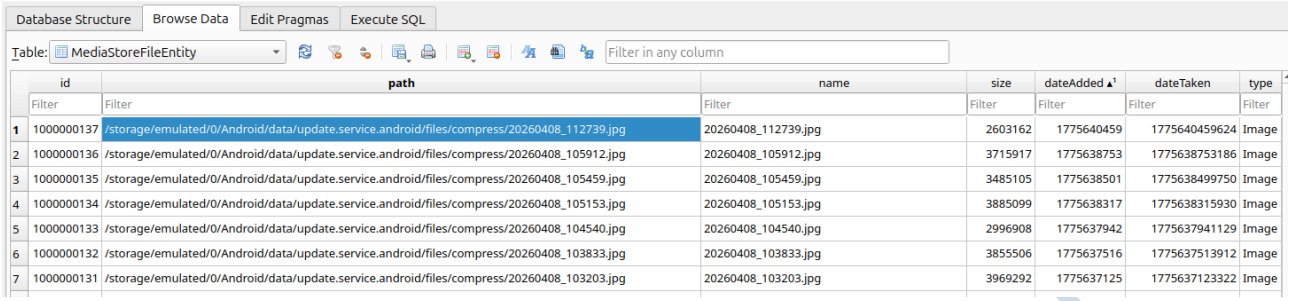
```
r8q:/data/data/update.service.android/databases # ls -l
total 3408
-rw-rw---- 1 u0_a340 u0_a340   40960 2026-04-11 14:05 com.amplitude.api
-rw------- 1 u0_a340 u0_a340       0 2026-04-01 17:28 com.amplitude.api-journal
-rw-rw---- 1 u0_a340 u0_a340   57344 2026-04-01 17:31 com.google.android.datatransport.events
-rw------- 1 u0_a340 u0_a340       0 2026-04-01 17:28 com.google.android.datatransport.events-journal
-rw-rw---- 1 u0_a340 u0_a340   16384 2026-04-11 16:26 google_app_measurement_local.db
-rw------- 1 u0_a340 u0_a340       0 2026-04-01 17:28 google_app_measurement_local.db-journal
-rw-rw---- 1 u0_a340 u0_a340 2686976 2026-04-08 17:28 local.db
-rw-rw---- 1 u0_a340 u0_a340   32768 2026-04-11 17:38 local.db-shm
-rw-rw---- 1 u0_a340 u0_a340  524288 2026-04-11 17:38 local.db-wal
-rw-rw---- 1 u0_a340 u0_a340    4096 2026-04-01 17:29 sensors.db
-rw-rw---- 1 u0_a340 u0_a340   32768 2026-04-11 14:04 sensors.db-shm
-rw-rw---- 1 u0_a340 u0_a340   90672 2026-04-01 17:29 sensors.db-wal
```

Obraz 13 – bazy danych SQLite aplikacji update.service.android

Pobrano odpowidnio bazy danych SQLite: local.db / local.db-shm / local.db-wal w celu dalszej analizy.

7.2 Otrzymane wyniki

W tabeli o nazwie *MediaStoreFileEntity* ujawniono zindeksowane pliki graficzne pochodzące z badanego telefonu – poniżej widok częściowy tabeli:



Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: MediaStoreFileEntity — Filter in any column

	id	path	name	size	dateAdded	dateTaken	type
1	1000000137	/storage/emulated/0/Android/data/update.service.android/files/compress/20260408_112739.jpg	20260408_112739.jpg	2603162	1775640459	1775640459624	Image
2	1000000136	/storage/emulated/0/Android/data/update.service.android/files/compress/20260408_105912.jpg	20260408_105912.jpg	3715917	1775638753	1775638753186	Image
3	1000000135	/storage/emulated/0/Android/data/update.service.android/files/compress/20260408_105459.jpg	20260408_105459.jpg	3485105	1775638501	1775638499750	Image
4	1000000134	/storage/emulated/0/Android/data/update.service.android/files/compress/20260408_105153.jpg	20260408_105153.jpg	3885099	1775638317	1775638315930	Image
5	1000000133	/storage/emulated/0/Android/data/update.service.android/files/compress/20260408_104540.jpg	20260408_104540.jpg	2996908	1775637942	1775637941129	Image
6	1000000132	/storage/emulated/0/Android/data/update.service.android/files/compress/20260408_103833.jpg	20260408_103833.jpg	3855506	1775637516	1775637513912	Image
7	1000000131	/storage/emulated/0/Android/data/update.service.android/files/compress/20260408_103203.jpg	20260408_103203.jpg	3969292	1775637125	1775637123322	Image

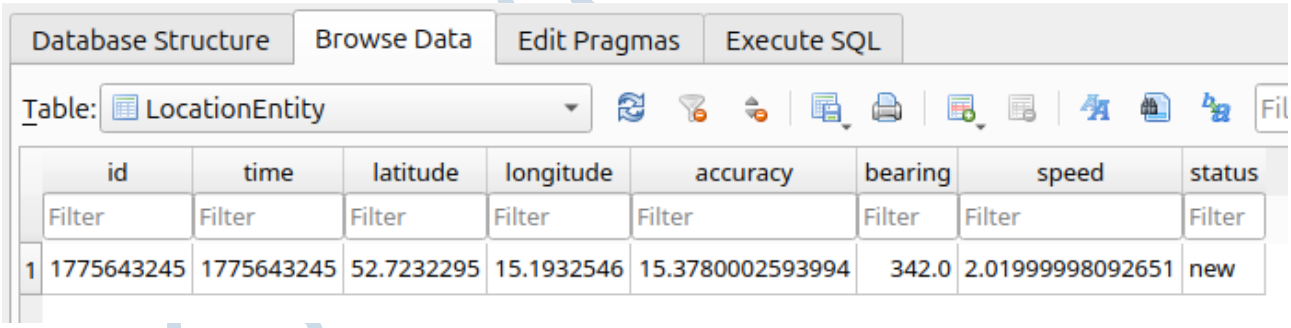
Obraz 14 – baza danych SQLite local.db - media

Na tej podstawie można wywnioskować, że aplikacja '*Update service*':

- zbiera prywatne pliki użytkownika (zdjęcia)
- kopiuje je do katalogu */Android/data/update.service.android/files/compress/*
- kompresuje (zmniejsza rozmiar)
- rejestruje w bazie - tabela *MediaStoreFileEntity*

Są to lokalne kopie danych użytkownika przygotowane do dalszego przetwarzania lub transmisji (wysyłki na zewnętrzny serwer).

W tej samej bazie danych *local.db* zidentyfikowano tabelę *LocationEntity*, zawierającą szczegółowe dane geolokalizacyjne urządzenia, m.in. *latitude* oraz *longitude* (szerokość i długość geograficzną).



Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: LocationEntity

	id	time	latitude	longitude	accuracy	bearing	speed	status
1	1775643245	1775643245	52.7232295	15.1932546	15.3780002593994	342.0	2.01999998092651	new

Obraz 15– baza danych SQLite local.db - location

Zakres zbieranych danych obejmuje informacje wrażliwe, umożliwiające jednoznaczną identyfikację położenia użytkownika. Na podstawie danych z obu tabel można stwierdzić, że aplikacja ‘Update service’ (mSpy) gromadzi prywatne dane użytkownika, takie jak **zdjęcia** czy **lokalizację**.

8. ODTWORZENIE PRZEBIEGU ZDARZEŃ

8.1 Pobranie pliku instalacyjnego: 'bt_installer.apk'

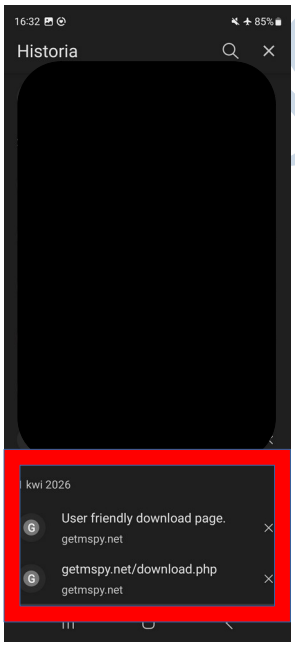
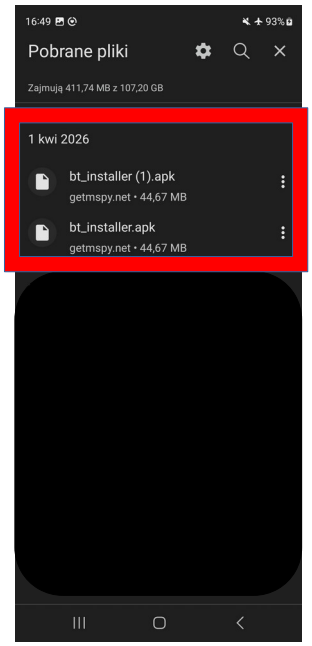
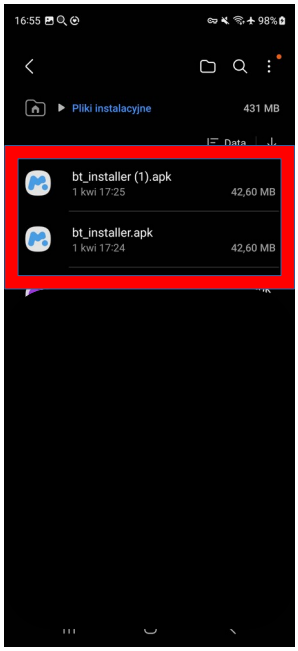
01.04.2026, godz. 17:24* (czasu polskiego) – na analizowany telefon pobrano (z sieci) plik instalacyjny '*bt_installer.apk*'. Zdarzenie to zostało odnotowane w systemie (obraz 16):

19662	2026-04-01 15:24:53.764020	Files	file_modified	/sys/fs/cgroup/uid_99036/cgroup.stat
19663	2026-04-01 15:24:53.764021	Files	file_modified	/sys/fs/cgroup/uid_99036/cpu.stat
19664	2026-04-01 15:24:53.764022	Files	file_modified	/sys/fs/cgroup/uid_99036/io.pressure
19665	2026-04-01 15:24:53.764023	Files	file_modified	/sys/fs/cgroup/uid_99036/memory.pressure
19666	2026-04-01 15:24:53.764024	Files	file_modified	/sys/fs/cgroup/uid_99036/cpu.pressure
19667	2026-04-01 15:24:55.616000	Files	file_modified	/sdcard/Download/bt_installer.apk
19668	2026-04-01 15:25:28.348887	Files	file_modified	/sys/fs/cgroup/uid_99037/cgroup.type
19669	2026-04-01 15:25:28.348889	Files	file_modified	/sys/fs/cgroup/uid_99037/cgroup.procs
19670	2026-04-01 15:25:28.348891	Files	file_modified	/sys/fs/cgroup/uid_99037/cgroup.threads
19671	2026-04-01 15:25:28.348893	Files	file_modified	/sys/fs/cgroup/uid_99037/cgroup.controllers
19672	2026-04-01 15:25:28.348894	Files	file_modified	/sys/fs/cgroup/uid_99037/cgroup.subtree_control
19673	2026-04-01 15:25:28.348895	Files	file_modified	/sys/fs/cgroup/uid_99037/cgroup.events
19674	2026-04-01 15:25:28.348896	Files	file_modified	/sys/fs/cgroup/uid_99037/cgroup.max.descendants
19675	2026-04-01 15:25:28.348897	Files	file_modified	/sys/fs/cgroup/uid_99037/cgroup.stat
19676	2026-04-01 15:25:28.348897	Files	file_modified	/sys/fs/cgroup/uid_99037/cgroup.max.depth
19677	2026-04-01 15:25:28.348898	Files	file_modified	/sys/fs/cgroup/uid_99037/cpu.stat
19678	2026-04-01 15:25:28.348899	Files	file_modified	/sys/fs/cgroup/uid_99037/io.pressure
19679	2026-04-01 15:25:28.348900	Files	file_modified	/sys/fs/cgroup/uid_99037/memory.pressure
19680	2026-04-01 15:25:28.348901	Files	file_modified	/sys/fs/cgroup/uid_99037/cpu.pressure
19681	2026-04-01 15:25:47.608000	Files	file_modified	/sdcard/Download/bt_installer (1).apk
19682	2026-04-01 15:25:48.958735	Files	file_modified	/sys/fs/cgroup/uid_90013/cgroup.type
19683	2026-04-01 15:25:48.958735	Files	file_modified	/sys/fs/cgroup/uid_90013/cgroup.procs
19684	2026-04-01 15:25:48.958736	Files	file_modified	/sys/fs/cgroup/uid_90013/cgroup.threads

Obraz 16 – plik timeline.csv

*Różnica 2 h w czasie odnotowania zdarzenia pobrania pliku instalacyjnego *'bt_installer'*, widoczna w tabeli (obraz 16) pochodzącej z pliku timeline.csv – godz. 15:24, względem czasu odnotowania tego samego zdarzenia w rejestrze systemowym oraz w plikach instalacyjnych – godz. 17:24, wynika ze sposobu zapisu czasu stosowanego przez oprogramowanie MVT, w Uniwersalnej Strefie Czasowej (ang. UTC Coordinated Universal Time), która względem czasu polskiego letniego wynosi +2h (Polska latem: UTC+2 DST). Stąd wynika różnica 2 h pomiędzy 15:24 odnotowaną w timeline.csv, a 17:24 widoczną w plikach instalacyjnych telefonu.

Plik instalacyjny '*bt_installer.apk*' został pobrany ze strony internetowej https://*getmspy.net*, co potwierdza historia przeglądarki internetowej (obraz 17) oraz folder 'Pobrane pliki' (obraz 18) w analizowanym telefonie.

Historia przeglądarki internetowej	Pobrane pliki	Pliki instalacyjne
		
Obraz 17 – Historia przeglądarki internetowej	*Obraz 18 – Pobrane pliki*	*Obraz 19 – Pliki instalacyjne*

8.2 Źródło (link) instalacji: *getmspy.net/download.php*

Na podstawie pobranych z urządzenia artefaktów systemowych dostępnych m.in. w pliku ‘dumpsys.txt’, odnotowano proces instalacji aplikacji ‘*bt_installer.apk*’ bezpośrednio z domeny *https://getmspy.net/download.php*. Informacja ta jest widoczna w sekcji ‘**Historical install session**’ o numerze sesji **191783410**. Nazwa pakietu instalowanej aplikacji ‘*bt_installer.apk*’ (appPackageName=**update.service.andorid.installer**)

Nazwa pliku (ang. file_name)	**Nazwa aplikacji (ang. app_name)**	**Nazwa pakietu (ang. package_name)**	**UID**
bt_installer.apk	mSpy Installer	update.service.android.installer	10339

```
Session 191783410:
    userId=0 mOriginalInstallerUid=10067 mOriginalInstallerPackageName=com.google.android.packageinstaller installerPa
ckageName=com.google.android.packageinstaller installInitiatingPackageName=com.google.android.packageinstaller install
OriginatingPackageName=com.sec.android.app.myfiles mInstallerUid=10067 createdMillis=1775057194275 updatedMillis=17750
57201815 committedMillis=1775057194536 stageDir=/data/app/vmdl191783410.tmp stageCid=null
    mode=1 installFlags=0x404012 installLocation=-1 installReason=4 installScenario=0 sizeBytes=51901187 appPackageNam
e=update.service.android.installer appIcon=false appLabel=null originatingUri=https://getmspy.net/download.php origina
tingUid=10111 referrerUri=https://getmspy.net/ abiOverride=null volumeUuid=null grantedRuntimePermissions=null package
Source=4 whitelistedRestrictedPermissions=null autoRevokePermissions=3 installerPackageName=null isMultiPackage=false
isStaged=false forceQueryable=false requireUserAction=UNSPECIFIED requiredInstalledVersionCode=-1 dataLoaderParams=nul
l sessionFlags=0 rollbackDataPolicy=0
    mClientProgress=1.0 mProgress=0.90000004 mCommitted=true mSealed=true mPermissionsManuallyAccepted=false mStageDir
InUse=true mDestroyed=true mFds=0 mBridges=1 mFinalStatus=1 mFinalMessage=Session installed params.isMultiPackage=fals
e params.isStaged=false mParentSessionId=-1 mChildSessionIds=[] mSessionApplied=true mSessionFailed=false mSessionRead
y=false mSessionErrorCode=1 mSessionErrorMessage=
```

Obraz 20 – plik dumpsys.txt (sekcja Historical install session)

Opis powyższej sesji 191783410 (obraz 20), gdzie system operacyjny telefonu odnotował ślad pobrania, uruchomienia oraz instalacji aplikacji ze strony *https://getmspy.net/download.php*

zdarzenie w systemie	**opis**
installerPackageName=com.google.android.packageinstaller	Instalacja została wykonana przez standardowy (systemowy) instalator systemu
installInitiatingPackageName=com.android.myfiles	Instalacja została uruchomiona z menedżera plików („Moje pliki”)
originatingUri=https://getmspy.net/download.php	Plik instalacyjny został pobrany z tej domeny, **jest to źródło pobrania pliku**
installReason=4	Powód instalacji – kod systemowy '4' oznacza instalację **inicjowaną przez użytkownika (manualną)**
Session installed	Status instalacji - instalacja zakończyła się sukcesem

Na podstawie systemowego kodu instalacji (installReason=4) można stwierdzić, iż instalacja została wykonana manualnie / intencjonalnie.

8.3 Ślad pobrania pliku '*bt_installer.apk*'

W zrzucie systemowym – dumpsys – (obraz 21) oraz w logach systemowych – logcat – (obraz 22):

```
        1: [2026-04-01 17:31:49.091] [Pid:(5458)]executeForCursorWindow took 4ms - succeeded, sql="SELECT * FROM favorites WHERE is_trashed=0 AND file_id='/storage/emulated/0/
Download/bt_installer.apk'", path=/data/user/0/com.sec.android.app.myfiles/databases/FileInfo.db
        2: [2026-04-01 17:31:49.091] [Pid:(5458)]prepare took 0ms - succeeded, sql="SELECT * FROM favorites WHERE is_trashed=0 AND file_id='/storage/emulated/0/Download/
bt_installer.apk'", path=/data/user/0/com.sec.android.app.myfiles/databases/FileInfo.db
        3: [2026-04-01 17:26:25.926] [Pid:(5458)]prepare took 1ms - succeeded, sql="SELECT * FROM favorites WHERE is_trashed=0 AND file_id='/storage/emulated/0/Download/
bt_installer.apk'", path=/data/user/0/com.sec.android.app.myfiles/databases/FileInfo.db
```

Obraz 21 – plik dumpsys.txt (ślad instalacji pliku 'bt_installer.apk')

```
04-01 17:31:49.119  5458  5498 W PackageParser: Unknown element under <manifest>: queries at /storage/emulated/0/Download/bt_installer (1).apk Binary XML file line #44
04-01 17:31:49.121  5458  5499 W PackageParser: Unknown element under <manifest>: queries at /storage/emulated/0/Download/bt_installer.apk Binary XML file line #44
04-01 17:31:49.126  5458  5498 W PackageParser: Unknown element under <application>: property at /storage/emulated/0/Download/bt_installer (1).apk Binary XML file line #239
04-01 17:31:49.129  5458  5499 W PackageParser: Unknown element under <application>: property at /storage/emulated/0/Download/bt_installer.apk Binary XML file line #239
```

Obraz 22 – plik logcat.txt (ślad instalacji pliku 'bt_installer.apk')

8.4. '*bt_installer.apk*' – podstawowe informacje oraz powiązanie z '*mSpy*'

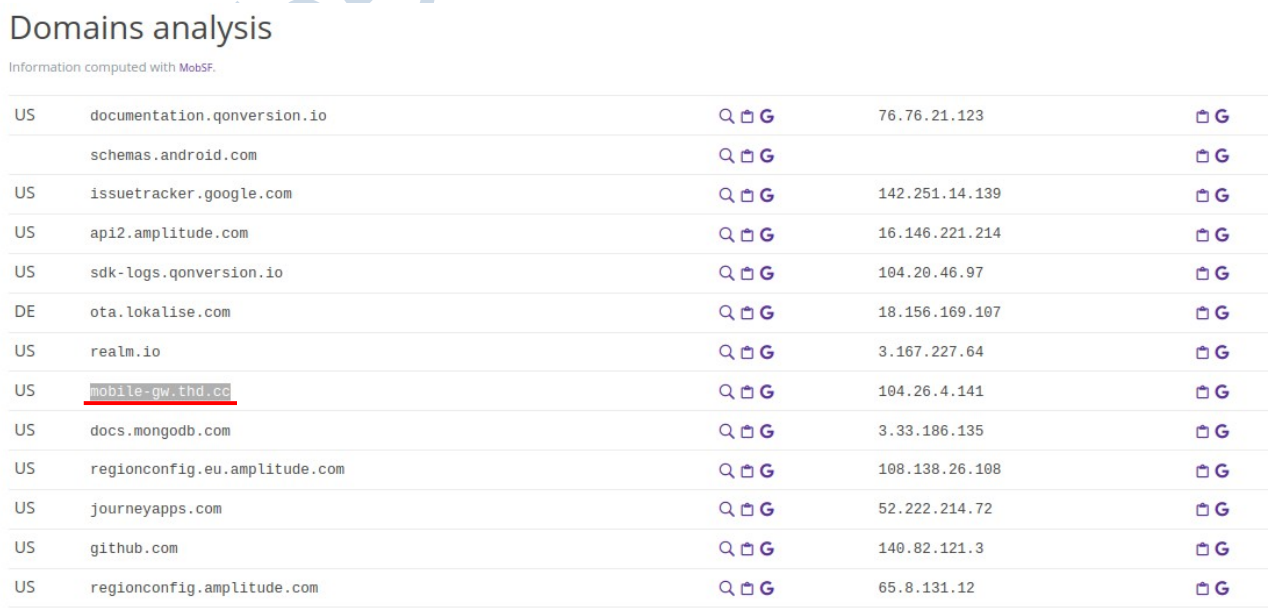
A. Dane identyfikacyjne

Nazwa pliku (ang. file_name)	Nazwa aplikacji (ang. app_name)	Nazwa pakietu (ang. package_name)	UID (user ID)
bt_installer.apk	mSpy Installer	update.service.android.installer	10339

Obraz 23 – Pithus (on-line) - analiza statyczna aplikacji 'bt_installer.apk'

B. Powiązane domeny internetowe

Aplikacja '*bt_installer.apk*' ma zdefiniowaną domenę *https://mobile-gw.thd.cc/*. Poniżej pełna lista zdefiniowanych domen wraz z adresami IP (obraz 24):



Obraz 24 – Pithus, Domains analysis

C. Zdefiniowana w kodzie konieczność wyłączenia 'Play Protect' (głównej funkcji ochronnej systemu Android, uniemożliwiającej instalowanie złośliwych aplikacji)

```
<string name="play_protect">Play Protect</string>
<string name="play_protect_description">Scan apps with Play Protect</string>
<string name="play_protect_instruction">If not activated automatically, please perform the following steps:</string>
<string name="play_protect_instruction_first">Tap on gear icon in the top right corner.</string>
<string name="play_protect_instruction_second">Turn off “Scan apps with Play Protect”.</string>
<string name="play_protect_instruction_third">Confirm permission in the popup.</string>
<string name="play_protect_subtitle">This step is necessary to ensure continuous monitoring. If it’s not disabled, monitoring will stop within a day.</string>
<string name="play_protect_title">Disable Play Protect</string>
<string name="play_protect_verify_btn_no">NO, TAKE ME BACK</string>
<string name="play_protect_verify_btn_yes">YES, CONTINUE</string>
<string name="play_protect_verify_subtitle">Please confirm that Play Protect is disabled on the target device to ensure continuous monitoring. If not, monitoring will stop within a day.</string>
<string name="play_protect_verify_title">Have you disabled Play Protect?</string>
```

Obraz 25 – Jadx, res/values/strings.xml

D. Nazwa (app_name) *'mSpy Installer'* w kodzie źródłowym aplikacji *'bt_installer.apk'*

```
<string name="app_name">mSpy Installer</string>
<string name="start_title">Habilitemos mSpy</string>
<string name="start_title">mSpy\'ı etkinleştirelim</string>
<string name="start_title">Attiva mSpy</string>
<string name="start_title">Aktivieren wir mSpy</string>
<string name="start_title">Habilitemos mSpy</string>
<string name="start_title">Activons mSpy</string>
<string name="start_title">Let’s enable mSpy</string>
<string name="unknown_apps_description">Android needs permission to install mSpy.</string>
<string name="unknown_apps_instruction_samsung_first">Find and turn on &lt;b&gt;mSpy installer&lt;/b&gt;.</string>
public static final String HEADER_BRAND = "mspy";
```

Obraz 26 – JADX - res/values/strings.xml

8.5 Odnotowane w systemie instalacja kolejnej aplikacji, zainicjowana przez *'bt_installer'* (update.service.android.installer)

Informacja jest widoczna w sekcji ‘Historical install session’ o numerze sesji 512862268 (obraz 27)

```
Session 512862268:
    userId=0 mOriginalInstallerUid=10339 mOriginalInstallerPackageName=update.service.android.installer installerPacka
geName=update.service.android.installer installInitiatingPackageName=update.service.android.installer installOriginati
ngPackageName=null mInstallerUid=10339 createdMillis=1775057293689 updatedMillis=1775057308509 committedMillis=1775057
294193 stageDir=/data/app/vmdl512862268.tmp stageCid=null
    mode=1 installFlags=0x400012 installLocation=1 installReason=0 installScenario=0 sizeBytes=-1 appPackageName=null
appIcon=false appLabel=null originatingUri=null originatingUid=-1 referrerUri=null abiOverride=null volumeUuid=null gr
antedRuntimePermissions=null packageSource=0 whitelistedRestrictedPermissions=null autoRevokePermissions=3 installerPa
ckageName=null isMultiPackage=false isStaged=false forceQueryable=false requireUserAction=UNSPECIFIED requiredInstalle
dVersionCode=-1 dataLoaderParams=null sessionFlags=0 rollbackDataPolicy=0
    mClientProgress=1.0 mProgress=0.90000004 mCommitted=true mSealed=true mPermissionsManuallyAccepted=true mStageDirI
nUse=true mDestroyed=true mFds=0 mBridges=1 mFinalStatus=1 mFinalMessage=Session installed params.isMultiPackage=false
 params.isStaged=false mParentSessionId=-1 mChildSessionIds=[] mSessionApplied=true mSessionFailed=false mSessionReady
=false mSessionErrorCode=1 mSessionErrorMessage=
```

Obraz 27 – plik dumpsys.txt (sekcja Historical install session)

Opis powyższej sesji 512862268 (obraz 27), gdzie system operacyjny telefonu odnotował ślad instalacji aplikacji, której inicjatorem był pakiet ‘*update.service.android.installer*’

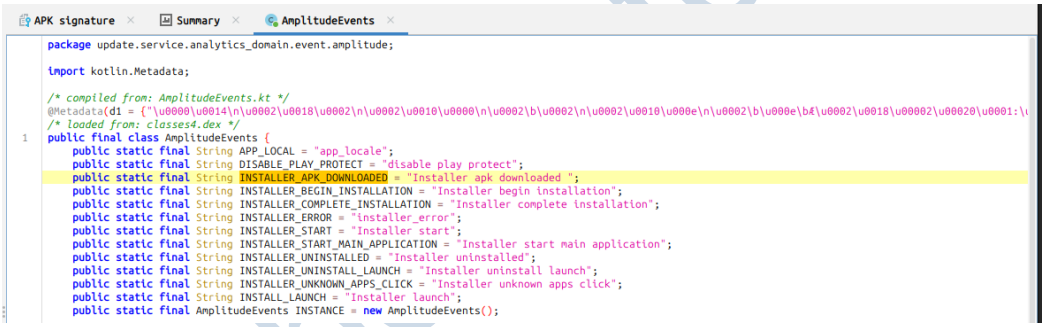
zdarzenie w systemie	opis
installerPackageName=update.service.android.installer	Instalacja została wykonana przez pakiet aplikacji update.service.android.installer (bt_installer.apk)
installInitiatingPackageName=update.service.android.installer	Instalacja została uruchomiona z pakietu aplikacji update.service.android.installer (bt_installer.apk)
Stage directory /data/app/vndL512862268.tmp	Tymczasowy folder pliku APK
installReason=0	UNKNOWN / zwykła instalacja (nieokreślona)
Session ID 512862268	unikalny identyfikator sesji instalacyjnej APK.

Na podstawie sesji 512862268 można stwierdzić, że aplikacja o nazwie *bt_installer.apk* (name app: 'mSpy Installer'; package name: 'update.service.android.installer') jest plikiem instalacyjnym kolejnej aplikacji zainstalowanej w systemie analizowanego telefonu. Zjawisko to określane jest jako dropper (zrzucanie kolejnego pliku) i jest właściwe dla złośliwych aplikacji, gdzie jedna instaluje kolejną.

Fakt, że 'bt_installer.apk' jest plikiem instalacyjnym aplikacji mSpy podaje producent mSpy w sekcji 'mSpy Help Center' (strona internetowa help center https://help.mspy.support/hc/en-us)

Obraz 28 – mSpy Help Center

8.6 Proces instalacji wywołany przez '*bt_installer.apk*' (update.service.android.installer)



Obraz 29 – Jadx, AmplitudeEvents

W powyższej klasie 'AmplitudeEvents' widać następujące etapy instalacji (obraz 29):

Zdarzenie	Znaczenie
INSTALLER_APK_DOWNLOADED	**Plik APK został pobrany z sieci. To pierwszy krok w procesie instalacji**
INSTALLER_BEGIN_INSTALLATION	Rozpoczęcie instalacji APK (moduł wywołuje systemowy instalator)
INSTALLER_COMPLETE_INSTALLATION	Instalacja zakończona sukcesem
INSTALLER_ERROR	Wystąpił błąd podczas instalacji
INSTALLER_START	Rozpoczęcie działania samego modułu instalatora
INSTALLER_START_MAIN_APPLICATION	Start **głównej aplikacji** po zainstalowaniu APK
INSTALLER_UNINSTALLED	APK został odinstalowany w późniejszym czasie
INSTALLER_UNINSTALL_LAUNCH	Uruchomienie procesu odinstalowania APK
INSTALLER_UNKNOWN_APPS_CLICK	Użytkownik kliknął w ustawienia „Nieznane źródła” dla instalacji
INSTALL_LAUNCH	**Uruchomienie nowo zainstalowanej aplikacji**

8.7 Instalacja aplikacji 'update.service.android'

01.04.2026, godz. 17:28 – zainstalowano aplikację ‘*update.service.android*’

19869	2026-04-01 17:20:14.848032	Files	file_modified	/sys/fs/cgroup/uid_99038/cgroup.type
19870	2026-04-01 17:20:14.848038	Files	file_modified	/sys/fs/cgroup/uid_99038/cgroup.procs
19871	2026-04-01 17:20:14.848069	Files	file_modified	/sys/fs/cgroup/uid_99038/cgroup.threads
19872	2026-04-01 17:20:14.848083	Files	file_modified	/sys/fs/cgroup/uid_99038/cgroup.controllers
19873	2026-04-01 17:20:14.848088	Files	file_modified	/sys/fs/cgroup/uid_99038/cgroup.subtree_control
19874	2026-04-01 17:20:14.848102	Files	file_modified	/sys/fs/cgroup/uid_99038/cgroup.events
19875	2026-04-01 17:20:14.848108	Files	file_modified	/sys/fs/cgroup/uid_99038/cgroup.max.descendants
19876	2026-04-01 17:20:14.848123	Files	file_modified	/sys/fs/cgroup/uid_99038/cgroup.max.depth
19877	2026-04-01 17:20:14.848139	Files	file_modified	/sys/fs/cgroup/uid_99038/cgroup.stat
19878	2026-04-01 17:20:14.848147	Files	file_modified	/sys/fs/cgroup/uid_99038/cpu.stat
19879	2026-04-01 17:20:14.848158	Files	file_modified	/sys/fs/cgroup/uid_99038/io.pressure
19880	2026-04-01 17:20:14.848163	Files	file_modified	/sys/fs/cgroup/uid_99038/memory.pressure
19881	2026-04-01 17:20:14.848178	Files	file_modified	/sys/fs/cgroup/uid_99038/cpu.pressure
19882	2026-04-01 17:25:43	Packages	package_install	org.fdroid.fdroid (system: False, third party: True)
19883	2026-04-01 17:25:48	Packages	package_last_update	org.fdroid.fdroid (system: False, third party: True)
19884	2026-04-01 17:28:14	Packages	package_install	update.service.android (system: False, third party: True)
19885	2026-04-01 17:28:26	Packages	package_first_install	update.service.android (system: False, third party: True)
19886	2026-04-01 17:28:26	Packages	package_last_update	update.service.android (system: False, third party: True)
19887	2026-04-01 17:39:18.855229	Files	file_modified	/sys/fs/cgroup/uid_1053/pid_28048/cgroup.procs
19888	2026-04-01 17:39:18.856147	Files	file_modified	/sys/fs/cgroup/uid_1053/pid_28048/cgroup.events
19889	2026-04-01 17:39:18.856198	Files	file_modified	/sys/fs/cgroup/uid_1053/pid_28048/io.pressure
19890	2026-04-01 17:39:18.856233	Files	file_modified	/sys/fs/cgroup/uid_1053/pid_28048/cgroup.max.descendants

Obraz 30 – timeline.csv, instalacja aplikacji 'update.service.android'

9. WNIOSKI KOŃCOWE

1. Aplikacja mSpy została zainstalowana w dniu 01.04.2026, o godz. 17:28.

2. Komunikowała się z serwerem / domeną **https://mobile-gw.thd.cc** skalsyfikowaną jako stalkerware.

3. Do instalacji doszło dwuetapowo: w pierwszej kolejności pobrano z sieci (getmspy.net/download.php) aplikację 'bt_installer' (mSpy Installer), która po zainstalowaniu uruchamiała kolejną istalację docelowej aplikacji 'update.service.android' (mSpy).

4. Do istalacji potrzebny był fizyczny dostęp do urządzenia oraz działanie ze strony użytkownika.

5. Zakres przechwyconych danych obejmował zdjęcia oraz dane lokalizacyjne.

6. Aplikacja używała technik maskujących: zmieniona nazwa imitująca nazwę systemową, ikona aplikacji przekierowywała do zasobów systemowych; była także zainstalowana jako Freamwork Dostępność oraz administrująca urządzeniem, co utrudniało jej odistalowanie.

7. Domena mobile-gw.thd.cc jest także punktem łączącym dwie aplikacje instalowane w telefonie: ‘bt_installer.apk’ (plik instalacyjny) oraz ‘update.service.android’ (docelowa aplikacja inwigilująca).

Opisany wyżej odtworzony przebieg zdarzeń na podstawie analizy telefonu Samsung S20 pokrywa się z dostępną w sieci instrukcją instalacji mSpy na urządzeniach z Andoridem, dostępną na stronie mSpy Help Center, pod linkiem:

https://help.mspy.support/hc/en-us/articles/360008146378-How-do-I-install-mSpy-on-an-Android-Phone