Cyber Threat Intelligence Report

TERACON

“To provide Cyber Warfare awareness, guidance, and security solutions through collaboration, education, volunteer work, and training to assist the Republic of Korea Government, Military, Commercial Interests, and International Partners."

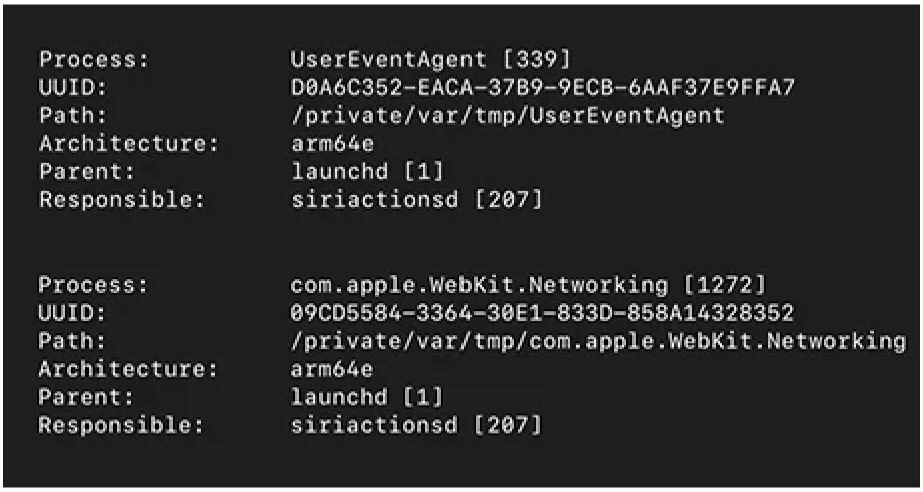
代表理事

Korea CYBER R&D

CYTROX PREDATOR vs DigitalBank Vault Knox Samsung Device (DBV)

Founded in 2017, Cytrox’s business activity is blandly described in Crunchbase as providing governments with an “operational cyber solution” that includes gathering information from devices and cloud services. In Pitchbook, their technology is defined as “cyber intelligence systems designed to offer security” to governments and assist with “designing, managing and implementing cyber intelligence gathering in the network, enabling businesses to gather intelligence from both end devices as well as from cloud services.”Cytrox is part of the so-called “Intellexa alliance,” a marketing label for a range of mercenary surveillance vendors that emerged in 2019. The consortium of companies includes Nexa Technologies (formerly Amesys), WiSpear/Passitora Ltd., Cytrox, and Senpai, along with other unnamed entities, purportedly seeking to compete against other players in the cyber surveillance market such as NSO Group and Verint. We obtained Android and iOS payloads from distedc[.]com and found them to be copies of a loader for a spyware product called Predator. We believe that these payloads are invoked by a previous exploit phase that we do not have.The iPhone executable is a 64-bit Mach-O binary which, like its Android counterpart, expects two arguments when the binary’s main function is called, which appear to be a kernel process task port and a pid value. The main function then calls kmem_init with these values, which proceeds to enable Predator stage 1 for continued execution. The Android sample passes its arguments to shared constants SHMEMFD_VSS and SHMEMFD_VSS.

Both the iOS and Android samples then call a startPy function to load a bundled Python 2.7 runtime. In the iOS sample, two additional built-in objects are added to the runtime: predutils and predconfig. The Android sample contains further additional built-in objects: injector, pc2, recorder, and voip_recorder. Upon initialization, startPy loads a frozen Python module named loader which begins by importing the Predator config from the interpreter's predconfig module. The iOS and Android configurations are slightly different. The complete configurations are available in Appendix 1. Once Predator iOS loads its configuration, it loads another frozen Python module named km_ios, a utility module that provides kernel memory management helper functions enabling additional Predator module capabilities. The iOS payload also contains a _check function, which queries the phone number and the phone's current locale country code. If the locale country code is equal to "IL" (the country code for Israel), or the phone number begins with "+972" (the telephone country code for Israel) then the spyware terminates. However, the method that Predator uses to query the phone number, CTSettingCopyMyPhoneNumber, may not work in recent versions of iOS. We could not determine how (or if) the _check function is called.

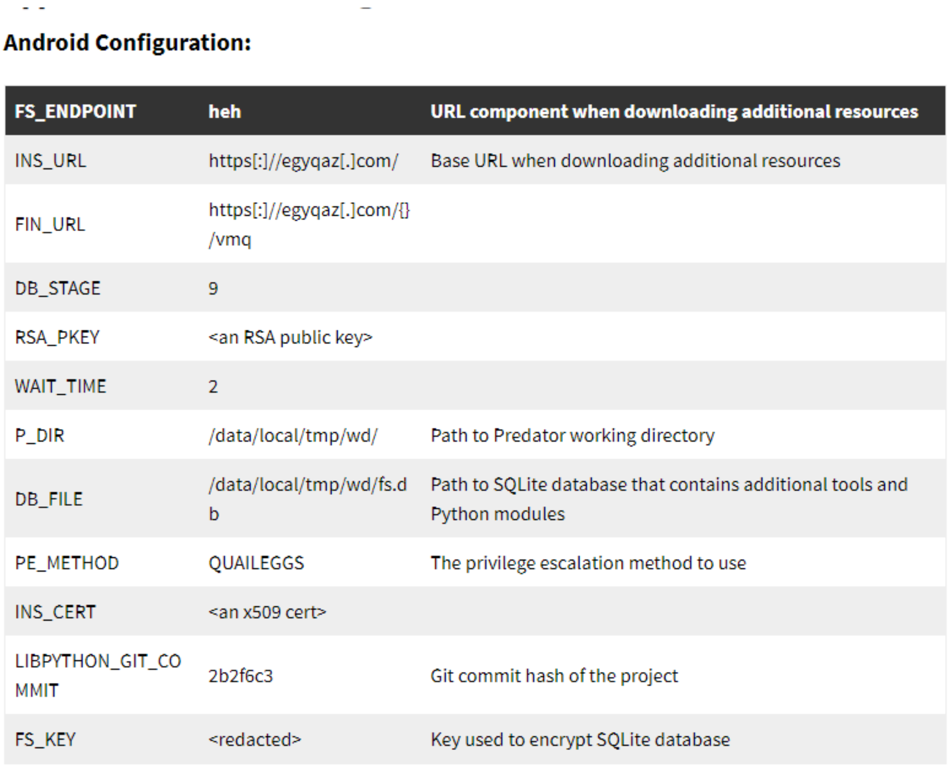


```
Process:          UserEventAgent [339]
UUID:             D0A6C352-EACA-37B9-9ECB-6AAF37E9FFA7
Path:             /private/var/tmp/UserEventAgent
Architecture:     arm64e
Parent:           launchd [1]
Responsible:      siriactionsd [207]


Process:          com.apple.WebKit.Networking [1272]
UUID:             09CD5584-3364-30E1-833D-858A14328352
Path:             /private/var/tmp/com.apple.WebKit.Networking
Architecture:     arm64e
Parent:           launchd [1]
Responsible:      siriactionsd [207]
```

In addition to the frozen loader module, "src/loader.py" ("frozenpyc/src/loader.py" in the Android sample), we also found copies of what appear to be older versions of the module that do not appear to be invoked by Predator: "src/loader2.py", "src/loader_real.py" and "src/loaderBackup03". All of the loader versions contain multiple references to "Predator." On Android, the loader module also downloads additional files from the server (specified by predconfig's INS_URL parameter, which is set to https://egyqaz[.]com).

After Infection process with Predator. while examining the DBV device logs we determined that, on December 30, 2021, two commands "/Payload2" were running on the phone (PIDs 339 and 1272), and that these commands had been launched with a single argument, a URL on distedc[.]com. The commands were not able to continue running as root. DBV device logs indicated that the process names of the commands were UserEventAgent and com.samsung.We did not find a mechanism for persistence on DBV devices, nor values in the Android configuration file that indicate persistence support. We also did not find some additional code in the Android sample, including code to disable SELinux and code for an audio recording component.

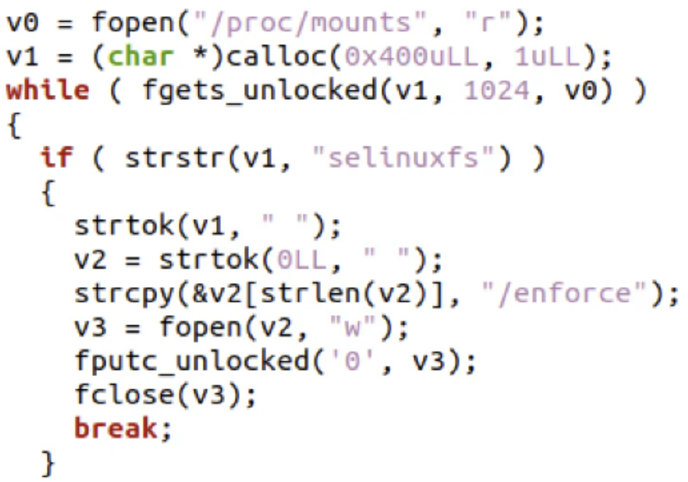
Predator cannot store additional Python modules and native ELF binaries in the fs.db SQLite file which is located at the path set in DB_FILE. The Python interpreter has a frozen module called sqlimper which is responsible for interacting with this database. The database contains a table called files which has a column called file_hash and a column called file_data. The injector module declares one function, inject, which can inject a shared object into a running process. Interestingly, there is a function called prior to injection which attempts to disable SELinux enforcement via the SELinuxFS.It should be noted that this approach likely will not succeed on devices that have additional checks and protections around SELinux enforcement—for example, Samsung RKP. However, there are artifacts associated with Predator that suggest approaches like RKP can be defeated by stomping on the SELinux access vector cache entries to grant the needed permissions. The pc2 module contains a single function, pc2_send_command, that is used as an IPC mechanism to send commands to Predator's audio recording component. The supported commands are START_VOIP, STOP_VOIP, START_MICRORECORDER, STOP_MICRORECORDER, and POLL_VOIP. This module works in conjunction with the recorder and voip_recorder modules. Each of the recorder modules have a start and stop function which are used to start/stop Predator's hot mic (recorder) and call recording (voip_recorder) capabilities. Recordings are stored in /data/local/tmp/wd/r/ in MP3 format.



Android Configuration:

FS_ENDPOINT	heh	URL component when downloading additional resources
INS_URL	https[:]//egyqaz[.]com/	Base URL when downloading additional resources
FIN_URL	https[:]//egyqaz[.]com/{}/vmq	
DB_STAGE	9	
RSA_PKEY	<an RSA public key>	
WAIT_TIME	2	
P_DIR	/data/local/tmp/wd/	Path to Predator working directory
DB_FILE	/data/local/tmp/wd/fs.db	Path to SQLite database that contains additional tools and Python modules
PE_METHOD	QUAILEGGS	The privilege escalation method to use
INS_CERT	<an x509 cert>	
LIBPYTHON_GIT_COMMIT	2b2f6c3	Git commit hash of the project
FS_KEY	<redacted>	Key used to encrypt SQLite database

This report is the first investigation to discover Cytrox's mercenary spyware being abused to target civil society and how it can be blocked. Remarkably, we tried to simultaneously infect DBV devices with NSO Group's Pegasus spyware without results.. NSO Group has received outsized publicity in recent years, thanks to a growing customer list, spiraling abuse problems, and groundbreaking investigative work by civil society. Cytrox and its Predator spyware, meanwhile, are relatively unknown.

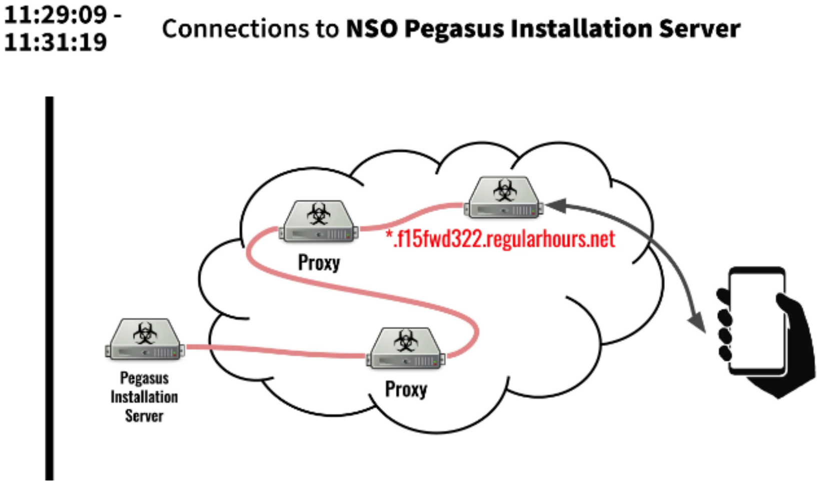


```
v0 = fopen("/proc/mounts", "r");
v1 = (char *)calloc(0x400uLL, 1uLL);
while ( fgets_unlocked(v1, 1024, v0) )
{
  if ( strstr(v1, "selinuxfs") )
  {
    strtok(v1, " ");
    v2 = strtok(0LL, " ");
    strcpy(&v2[strlen(v2)], "/enforce");
    v3 = fopen(v2, "w");
    fputc_unlocked('0', v3);
    fclose(v3);
    break;
  }
```

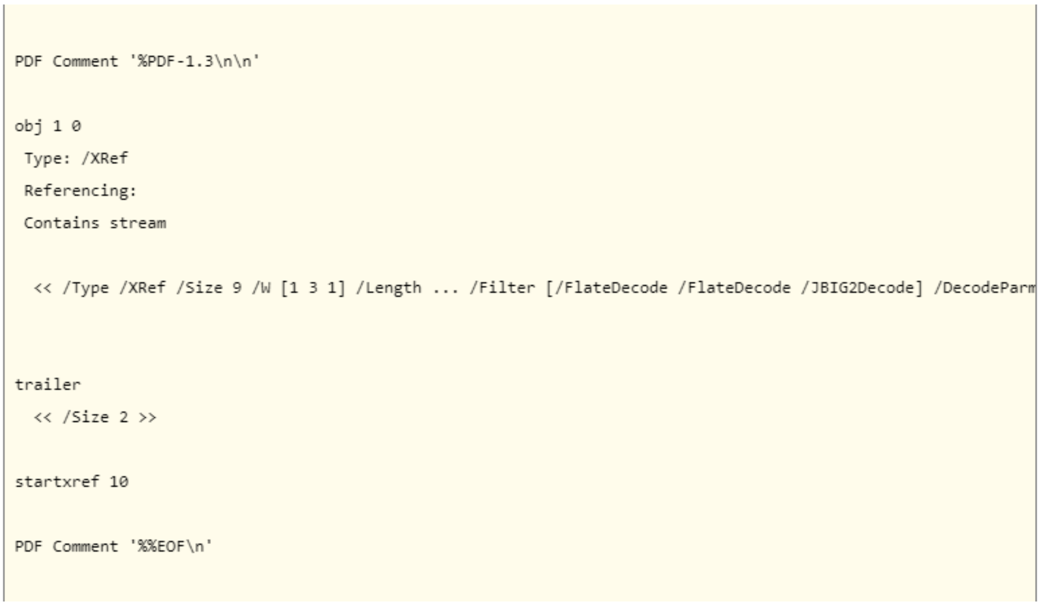
The targeting of a single individual DBV device with both Pegasus and Predator underscores that the practice of hacking civil society transcends any specific mercenary spyware company. Instead, it is a pattern that we expect will persist as long as autocratic governments are able to obtain sophisticated hacking technology. Absent international and domestic regulations and safeguards, journalists, human rights defenders, and opposition groups will continue to be hacked into the foreseeable future.

NSO GROUP SPYWARE PEGASUS vs. DigitalBank Vault Knox Samsung Device (DBV)

NSO Group’s Pegasus spyware is a mobile phone surveillance solution that enables customers to remotely exploit and monitor devices. The company is a prolific seller of surveillance technology to governments around the world, and its products have been regularly linked to surveillance abuses.Pegasus became known for the telltale malicious links sent to targets via SMS for many years. This method was used by NSO Group customers to target Ahmed Mansoor, dozens of members of civil society in Mexico, and political dissidents targeted by Saudi Arabia, among others. The use of malicious links in SMSes made it possible for investigators and targets to quickly identify evidence of past targeting. Targets could not only notice these suspicious messages, but they could also search their message history to detect evidence of hacking attempts.

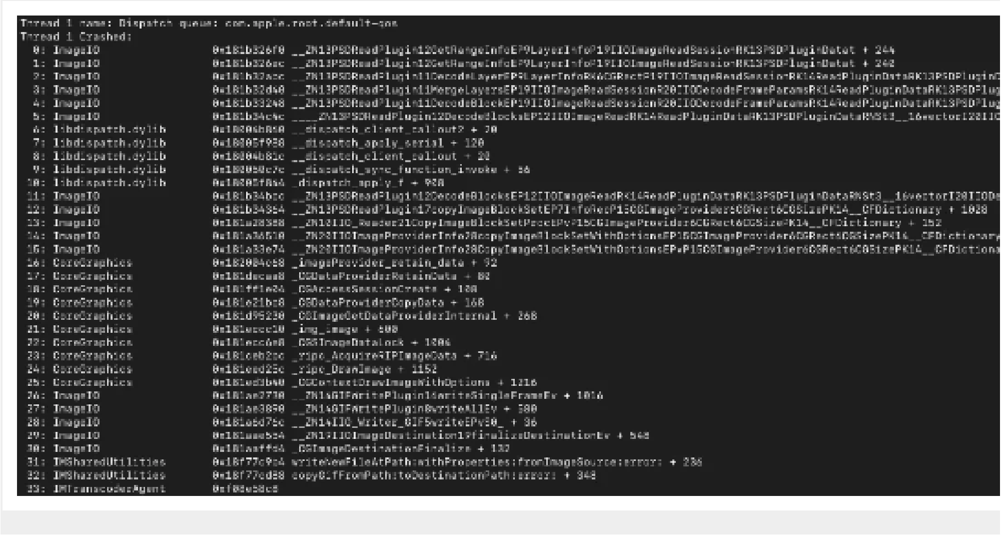

11:29:09 -
11:31:19
Connections to NSO Pegasus Installation Server
Proxy
*.f15fwd322.regularhours.net
Pegasus Installation Server
Proxy

In November 2021, we examined a DBV phone given by an activist , and determined that they had not been hacked with NSO Group’s Pegasus spyware. We observed multiple distinctive elements that allowed us to make a high-confidence attribution to NSO Group: The spyware installed by the FORCEDENTRY exploit exhibited a forensic artifact that we call CASCADEFAIL, which is a bug whereby evidence is incompletely deleted from the phone’s DataUsage.sqlite file. In CASCADEFAIL, an entry from the file’s ZPROCESS table is deleted, but not entries in the ZLIVEUSAGE table that refer to the deleted ZPROCESS entry. We have only ever seen this type of incomplete deletion associated with NSO Group’s Pegasus spyware, and we believe that the bug is distinctive enough to point back to the fact that DBV phone blocked the infection. The specific CASCADEFAIL artifact can be detected by SELECT "CASCADEFAIL" FROM ZLIVEUSAGE WHERE ZLIVEUSAGE.ZHASPROCESS NOT IN (SELECT Z_PK FROM ZPROCESS);



```
PDF Comment '%PDF-1.3\n\n'

obj 1 0
 Type: /XRef
 Referencing:
 Contains stream

  << /Type /XRef /Size 9 /W [1 3 1] /Length ... /Filter [/FlateDecode /FlateDecode /JBIG2Decode] /DecodeParm


trailer
  << /Size 2 >>

startxref 10

PDF Comment '%%EOF\n'
```

The spyware that tried to be installed by the FORCEDENTRY exploit used multiple process names, including the name “setframed”. FORCEDENTRY is the latest in a string of zero-click exploits linked to NSO Group. In 2019, WhatsApp fixed CVE-2019-3568, a zero-click vulnerability in WhatsApp calling that NSO Group used against more than 1400 phones in a two-week period during which it was observed, and in 2020, NSO Group employed the KISMET zero-click exploit. To our knowledge, the KISMET vulnerability was never publicly identified, though we suspect that the underlying vulnerability (if it still exists) can no longer be exploited via SMS due to DBV introduction of the KNOX Door mitigation in DBV phone.

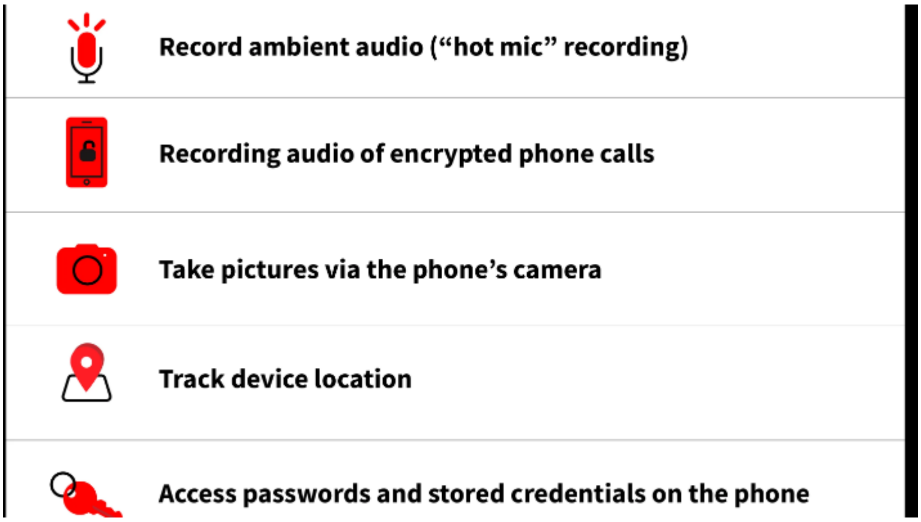


```
Thread 1 name: Dispatch queue: com.apple.root.default-qos
Thread 1 Crashed:
  0: ImageIO                0x181b326f0 __ZN13PSDReadPlugin12GetRangeInfoEP9LayerInfoP19IIOImageReadSessionRK13PSDPluginDatat + 244
  1: ImageIO                0x181b326ec __ZN13PSDReadPlugin12GetRangeInfoEP9LayerInfoP19IIOImageReadSessionRK13PSDPluginDatat + 240
  2: ImageIO                0x181b32acc __ZN13PSDReadPlugin11DecodeLayerEP9LayerInfoRK6CGRectP19IIOImageReadSessionRK14ReadPluginDataRK13PSDPluginD
  3: ImageIO                0x181b32d40 __ZN13PSDReadPlugin11MergeLayersEP19IIOImageReadSessionR20IIODecodeFrameParamsRK14ReadPluginDataRK13PSDPlug
  4: ImageIO                0x181b33248 __ZN13PSDReadPlugin11DecodeBlockEP19IIOImageReadSessionR20IIODecodeFrameParamsRK14ReadPluginDataRK13PSDPlug
  5: ImageIO                0x181b34c4c ____ZN13PSDReadPlugin12DecodeBlocksEP12IIOImageReadRK14ReadPluginDataRK13PSDPluginDataRNSt3__16vectorI20IIO
  6: libdispatch.dylib      0x18006b850 __dispatch_client_callout2 + 20
  7: libdispatch.dylib      0x18005f938 __dispatch_apply_serial + 120
  8: libdispatch.dylib      0x18004b81c __dispatch_client_callout + 20
  9: libdispatch.dylib      0x180058c7c __dispatch_sync_function_invoke + 56
 10: libdispatch.dylib      0x18005f8a4 _dispatch_apply_f + 928
 11: ImageIO                0x181b34bcc __ZN13PSDReadPlugin12DecodeBlocksEP12IIOImageReadRK14ReadPluginDataRK13PSDPluginDataRNSt3__16vectorI20IIODe
 12: ImageIO                0x181b34354 __ZN13PSDReadPlugin17copyImageBlockSetEP7InfoRecP15CGImageProvider6CGRect6CGSizePK14__CFDictionary + 1028
 13: ImageIO                0x181a28388 __ZN12IIO_Reader21CopyImageBlockSetProcEPvP15CGImageProvider6CGRect6CGSizePK14__CFDictionary + 152
 14: ImageIO                0x181a36510 __ZN23IIOImageProviderInfo28copyImageBlockSetWithOptionsEP15CGImageProvider6CGRect6CGSizePK14__CFDictionary
 15: ImageIO                0x181a33e74 __ZN23IIOImageProviderInfo28CopyImageBlockSetWithOptionsEPvP15CGImageProvider6CGRect6CGSizePK14__CFDictiona
 16: CoreGraphics           0x182084c68 _imageProvider_retain_data + 92
 17: CoreGraphics           0x181decaa8 _CGDataProviderRetainData + 80
 18: CoreGraphics           0x181ff1a24 _CGAccessSessionCreate + 100
 19: CoreGraphics           0x181e21bc8 _CGDataProviderCopyData + 168
 20: CoreGraphics           0x181d95230 _CGImageGetDataProviderInternal + 268
 21: CoreGraphics           0x181eccc10 _img_image + 600
 22: CoreGraphics           0x181ecc6a8 _CGSImageDataLock + 1096
 23: CoreGraphics           0x181ceb2cc _ripc_AcquireRIPImageData + 716
 24: CoreGraphics           0x181eed25c _ripc_DrawImage + 1152
 25: CoreGraphics           0x181ed3b40 _CGContextDrawImageWithOptions + 1216
 26: ImageIO                0x181ae2730 __ZN14GIFWritePlugin16writeSingleFrameEv + 1016
 27: ImageIO                0x181ae3850 __ZN14GIFWritePlugin8writeAllEv + 580
 28: ImageIO                0x181a6d75c __ZN14IIO_Writer_GIF5writeEPvS0_ + 36
 29: ImageIO                0x181aae554 __ZN17IIOImageDestination19finalizeDestinationEv + 548
 30: ImageIO                0x181aaff64 _CGImageDestinationFinalize + 132
 31: IMSharedUtilities      0x18f77c9c4 writeNewFileAtPath:withProperties:fromImageSource:error: + 236
 32: IMSharedUtilities      0x18f77cd88 copyGifFromPath:toDestinationPath:error: + 348
 33: IMTranscoderAgent      0xf08e58c8
```

We do not believe that KISMET works against DBV KNOX devices, which includes new security protections. Our finding also highlights the paramount importance of securing popular messaging apps. Ubiquitous chat apps have become a major target for the most sophisticated threat actors, including nation state espionage operations and the mercenary spyware companies that service them. As presently engineered, many chat apps have become an irresistible soft target. Without intense engineering focus, we believe that they will continue to be heavily targeted, and successfully exploited.

We obtained logs from the DBV phone device while it was in the trial process of getting infected. Our analysis indicates that the current Pegasus implant could not function and a number of capabilities has been completely blocked including: recording audio from the microphone including both ambient “hot mic” recording and audio of encrypted phone calls, and taking pictures. In addition, we believe the implant cannot track device location, and access passwords and stored credentials.

The DBV system blocked the following:



Record ambient audio (“hot mic” recording)

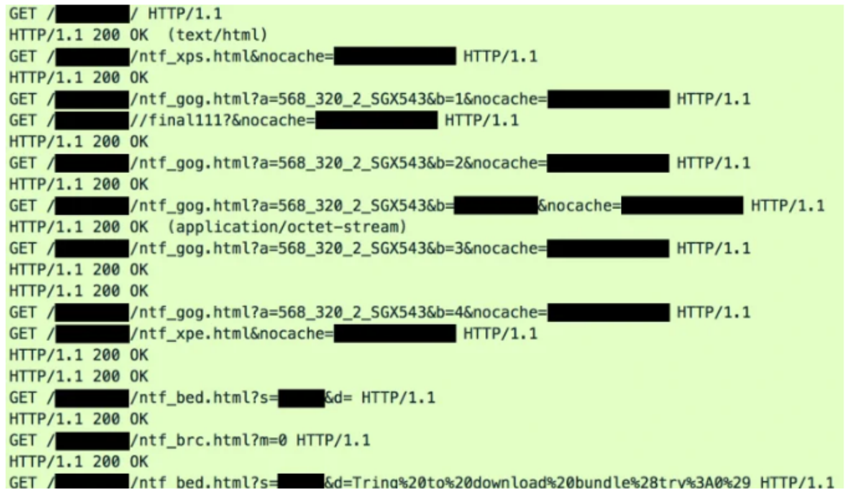
Recording audio of encrypted phone calls

Take pictures via the phone’s camera

Track device location

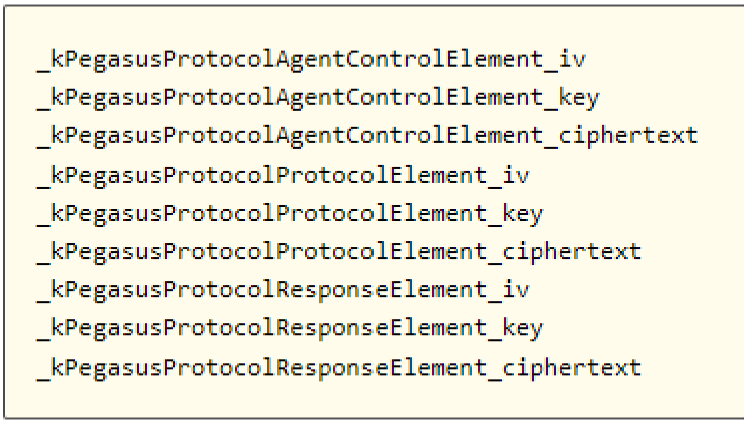
Access passwords and stored credentials on the phone

NSO Group offers two remote installation vectors for spyware onto a target’s device: a zero-click vector, and a one-click vector. The one-click vector involves sending the target a normal SMS text message with a link to a malicious website. The malicious website contains an exploit for the web browser on the target’s device, and any other required exploits to implant the spyware. To use NSO Group’s zero-click vector, an operator instead sends the same link via a special type of SMS message, like a WAP Push Service Loading (SL) message. A WAP Push SI message causes a phone to automatically open a link in a web browser instance, eliminating the need for a user to click on the link to become infected. Many newer models of phones have started ignoring or restricting WAP Push messages. Mobile network providers may also decide to block these messages.



```
GET /[redacted]/ HTTP/1.1
HTTP/1.1 200 OK  (text/html)
GET /[redacted]/ntf_xps.html&nocache=[redacted] HTTP/1.1
HTTP/1.1 200 OK
GET /[redacted]/ntf_gog.html?a=568_320_2_SGX543&b=1&nocache=[redacted] HTTP/1.1
GET /[redacted]//final111?&nocache=[redacted] HTTP/1.1
HTTP/1.1 200 OK
GET /[redacted]/ntf_gog.html?a=568_320_2_SGX543&b=2&nocache=[redacted] HTTP/1.1
HTTP/1.1 200 OK
GET /[redacted]/ntf_gog.html?a=568_320_2_SGX543&b=[redacted]&nocache=[redacted] HTTP/1.1
HTTP/1.1 200 OK  (application/octet-stream)
GET /[redacted]/ntf_gog.html?a=568_320_2_SGX543&b=3&nocache=[redacted] HTTP/1.1
HTTP/1.1 200 OK
HTTP/1.1 200 OK
GET /[redacted]/ntf_gog.html?a=568_320_2_SGX543&b=4&nocache=[redacted] HTTP/1.1
GET /[redacted]/ntf_xpe.html&nocache=[redacted] HTTP/1.1
HTTP/1.1 200 OK
HTTP/1.1 200 OK
GET /[redacted]/ntf_bed.html?s=[redacted]&d= HTTP/1.1
HTTP/1.1 200 OK
GET /[redacted]/ntf_brc.html?m=0 HTTP/1.1
HTTP/1.1 200 OK
GET /[redacted]/ntf_bed.html?s=[redacted]&d=Tring%20to%20download%20bundle%28try%3A0%29 HTTP/1.1
```

The final payload that we identified, test111.tar, contained several files, including libaudio.dylib, which appeared to be the base library for call recording, libimo.dylib, which appeared to be the library for recording chat messages from apps, and two libraries for WhatsApp and Viber call recording: libvbcalls.dylib, and libwacalls.dylib. In each file, we found several hundred strings containing the text "_kPegasusProtocol," the name of NSO Group's solution.



```
_kPegasusProtocolAgentControlElement_iv
_kPegasusProtocolAgentControlElement_key
_kPegasusProtocolAgentControlElement_ciphertext
_kPegasusProtocolProtocolElement_iv
_kPegasusProtocolProtocolElement_key
_kPegasusProtocolProtocolElement_ciphertext
_kPegasusProtocolResponseElement_iv
_kPegasusProtocolResponseElement_key
_kPegasusProtocolResponseElement_ciphertext
```

In this report, we started a highly technically sophisticated attack involving a zero-day remote jailbreak which installs spyware on a phone whose user clicks just once on a malicious link. We use the attack related to NSO Group's Pegasus spyware suite, sold exclusively to government agencies by Israel-based NSO Group. We made the connection based on our previous work tracing a group of servers that appeared to be part of an infrastructure for attacking mobile phones.

TERACON
주소 : 301호, 백토로 166, 군산시, 전라북도, 대한민국
사업자등록번호 : 82633677033
e-mail : info@teracon.io
대표 번호 : 82)1031598222

TERACON
Humanity Oriented