EXHIBIT A

FILED UNDER SEAL

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

WHATSAPP INC., et al,.

Plaintiffs,

v.

NSO GROUP TECHNOLOGIES LIMITED, et al.,

Defendants.

Case No.: 19-cv-07123-PJH

HIGHLY CONFIDENTIAL – ATTORNEYS EYES ONLY

Expert Report of David Youssef

August 30, 2024

August 30, 2024

EXPERT REPORT OF DAVID J. YOUSSEF

IN RE: WHATSAPP INC., ET AL., (PLAINTIFFS), V. NSO GROUP TECHNOLOGIES LIMITED, ET AL., (DEFENDANTS)

CASE NO.: 19-CV-07123-PJH

HIGHLY CONFIDENTIAL – ATTORNEYS EYES ONLY

made available to me, the "abs.log" file records that the exploit resulted in the "LEAKED ADDR" for "libwhatsapp_addr" (i.e., the address of libwhatsapp.so in the Target Device's memory) in each instance of a successful attack by "heaven" against a WhatsApp user.

[redacted]

Step 5: Conversion to Group Call; Delivery of Malicious Capabilities Buffer

89. In the next step of the exploit, Defendant utilized two phone numbers ("Phone A" and "Phone B") to convert the call to a group video call. After initiating the call with Phone A, Defendant invited Phone B to join the call, which provides the "capability_bit_mask" buffer. Ordinarily, this buffer describes all the VoIP capabilities it supports, but in the exploit, Defendants' Malware sends a modified bitmask buffer using the buffer memory layout and full pointer values gleaned from the Target Device.

90. The "abs.log" file indicates that the part of the attack involves adding another attacker account through a group call and the log contains evidence of the initial capabilities buffer sent to the second account.

[redacted]

Step 6: Delivery of Second Malicious Capabilities Buffer

91. As stated in Step 5, the voice call is converted into a group voice call including an additional participant controlled by Defendants. This secondary participant then sends a secondary capabilities buffer following inclusion into the group video call.

92. The "abs.log" also records the second malicious capabilities buffer sent by the second attacker account added in Step 5. Notably, the second message is returned less than one

second later, confirming that this process is automated and part of the exploit chain, and not the introduction of an independent second actor.

[redacted]

Step 7 & 8: Call Termination and Compromise of Target Device

93. As noted previously, the partial set of files from the AWS Server do not record the buffering messages over Plaintiffs' Relay Servers, such as those used in Step 7 to overwrite the pointers, but the "abs.log" file does record the identified indicators of voice call being terminated shortly after completing the steps above and before the call is answered.

[redacted]

94. In the figure above, after terminating the call, the log records that the "Call . . . was loud for 14.415701 seconds," which demonstrates Defendants' focus on stealth, and is consistent with the call termination marking the end of a covert zero-click exploit intended to be executed before the Target Device can accept the call.

DETAILED OPINIONS

Defendant's Exploitation of WhatsApp Infrastructure

Defendants' WhatsApp User Registration and Extraction of Region Token

95. For Defendants' "heaven" malware to successfully interact with WhatsApp's Signaling and Relay Servers and carry out their exploit chain, "heaven" must appear to these servers as a legitimate WhatsApp user or users operating the official WhatsApp Client. In my opinion, Defendants had to create WhatsApp user accounts and extract the Region Token to enable their Malware to gain access to WhatsApp's Servers and carry out their attack.

96. There are certain steps which must be completed before an individual can successfully register and subsequently utilize the WhatsApp services. This includes the user's acceptance of WhatsApp's Terms of Service, which must be done before completing the WhatsApp registration process. Only once the sign-up process has been completed can

the user access the WhatsApp services and the full functionality of the WhatsApp Client Application.

97. According to the deposition transcript of Claudiu Gheorghe, the registration process results in the creation of a Region Token, which informs the process through which the WhatsApp Servers attempt to assign requests from the account to the applicable geographical region. It is my understanding that neither the Signaling Server nor the Relay Server will begin communication without being provided with a valid Region Token.[56]

98. The programs found on Defendants' AWS Server do not contain any functionality to "forge" a WhatsApp Region Token. Therefore, it is my opinion Defendants must have created at least two WhatsApp user accounts and extracted the Region Tokens assigned to those accounts in order to use them with their Malware. Defendants then extracted the WhatsApp Client Region Tokens assigned to those accounts in order to use them as part of their "heaven" application.

99. In my opinion, by extracting the Region Tokens, "heaven" was configured to emulate the official WhatsApp Client from the perspective of the WhatsApp Signaling and Relay Servers. As a result, Defendants were able to make their Malware appear to the WhatsApp Servers as a legitimate WhatsApp user or users operating the WhatsApp Client.

Defendants' Engineered Malware to Access and Exploit WhatsApp Servers and Target Devices

100. In my opinion, Defendants' "heaven" malware is designed to interface with WhatsApp's Client Applications and Servers to successfully achieve remote code execution on the Target Device by camouflaging manipulated voice call messages as legitimate messages from a user running the WhatsApp Client Application.

101. In fact, Defendants were not using the official WhatsApp Client Application to carry out the exploit. The official WhatsApp Client does not enable users to create the types of malformed messages intrinsic to the exploit. In place of a genuine WhatsApp Client Application, Defendants designed and used their own system, a component of "heaven" referred to internally by Defendants as the "WhatsApp Installation Service" or "WIS,"[57] to emulate the official WhatsApp Client and send messages to the Target Device via the WhatsApp Signaling and Relay Servers, using authentication credentials taken from a registered WhatsApp Client to gain access to the servers.

[56] Gheorghe Dep. at 94:6-95:12.

[57] NSO_WHATSAPP_00008957.

102. As acknowledged by Defendants, their illegitimate client does not behave like a typical WhatsApp user; Defendants' illegitimate client instead "only sends very specific messages (call messages \ text messages)."[58]

103. Defendants' Malware was also designed to enable XOR encryption and insert a malicious bash script within the RTCP packet's "connecting_tone_desc" field, resulting in a malformed call offer message that the WhatsApp Client Application cannot create. Because Defendants' extraction of the registration token caused the Malware's messages to appear as if they came from the WhatsApp Client Application, this malformed message, which contains a malicious bash script, was then delivered to the Target Device.

104. Defendants' "heaven" malware also used multiple methods to carefully manipulate WhatsApp Servers and WhatsApp Client Applications running on Target Devices in order to achieve unimpeded access to and control over those devices. Defendants' Malware leveraged undocumented features of the WhatsApp call offer message to inject the malicious bash script into the Target Device's memory, as well as to enable the XOR encryption. Defendants' Malware, via the WhatsApp Servers, performed "fingerprinting" of the Target Device, which collected the user's online status, make, model, and operating systems. The fingerprinting allowed for Defendants' Malware to determine the manner in which to proceed with exploitation of the Target Device, based on extensive knowledge of the WhatsApp Client Application's internal workings which Defendants had apparently amassed. If the Target Device was determined to be vulnerable through fingerprinting, Defendants' Malware continued its actions by leveraging RTCP packets to overflow buffers on the Target Device, inserting further malicious data into the memory space used by the WhatsApp Client Application and manipulating this data, along with data natively present within the Application, to execute malicious code on the Target Device.

Defendants' Malware Operated in a Manner Which Exceeded the Capabilities Afforded by WhatsApp to an Ordinary User

105. It is my understanding that an ordinary user will download WhatsApp, create an account, and begin communications. This communication can include voice calls and multimedia messages, between two users or a group of users. The messages can include text or media.

106. It is my understanding that an ordinary user will typically have little to no knowledge of the WhatsApp infrastructure, let alone the intricacies of the function of a Signaling Server, Relay Server, RTP, and RTCP packets. The ordinary person will likely understand the

[58] NSO_WHATSAPP_00008957.

concept of encrypted communication but will unlikely understand the complexity of how to exploit end-to-end communications.

107. With the actions I discussed above, regarding the access and use of WhatsApp Servers and Target Devices, I will note for all activity, it is unreasonable for an ordinary person to have awareness of the concepts or have the technical, networking, scripting, and malicious software development skills to execute the activity. An ordinary person will unlikely understand the following technical concepts, such as the definition of a "buffer overflow," fingerprinting a device in order to inform how an exploitation is to be deployed, the concept of a malformed call offer message with manipulated settings, let alone how to create and transmit one, how to write in bash script, how to create and send modified packets to determine a device's architecture, the awareness of buffering messages and how to manipulate them to execute a buffer overflow, awareness of ASLR to determine where a WhatsApp library is held in a Target Device, and further how to perform remote code execution. Herein, it is my opinion Defendants accessed and used WhatsApp Servers and Target Devices in separate ways than an ordinary person, which includes overcoming technical limitations in the WhatsApp Client Application and WhatsApp's Servers that are beyond an expected user's activity.

Function and Impact of Defendants' Spyware

Defendants' Malware, Most Notably Pegasus, Operates as Spyware

108. It is my understanding that spyware involves secretly installing malicious software on a device without the knowledge of its user, which then monitors, captures, and extracts the user's activity on the device.

109. The exploit discussed in this report served as a mechanism to deliver Defendant's "zero-click" mechanism to a Target Device, which then secretly executed a malicious script onto the Target Device's memory.

110. Defendants knowingly designed their exploit to be executed without the target user's awareness. This is seen in both the fingerprinting and attack stages of the exploit, where Defendants explicitly define steps to minimize the target's awareness. In the fingerprinting stage, this is achieved by the following:[59]

- Using the same group chat name in all fingerprinting attempts of a target;
- Using the same mobile number in all fingerprint attempts of a target; and
- Avoiding using the same mobile number repeatedly when adding targets to various groups chats.

[59] NSO_WHATSAPP_00008960.

exploit, and the use of the WhatsApp Signaling and Relay Server, the victims in this instance would have likely believed they were receiving a legitimate incoming communication via WhatsApp.

Defendants' Malware Obtained Information from WhatsApp Servers and Target Devices

121. Regarding the steps mentioned above, it is my opinion that Defendants would have had to develop and test their exploit using WhatsApp Servers and the WhatsApp Client Application. Further, WhatsApp network infrastructure was identified in enough detail to learn the locations of specific Signal and Relay Servers utilized by WhatsApp. This knowledge allowed Defendants' Malware to be deployed onto specific Target Devices for the purposes of extracting data from the Target Device and surveilling its user without the user's awareness.

122. Through the fingerprinting and architecture identification mechanisms of "heaven," Defendants were able to collect information about the Target Devices, including hardware details, installed versions of the WhatsApp Client Application, and the targeted users' activity status within WhatsApp. I note Defendants obtained information about Target Devices through the WhatsApp Servers which was not intended by Plaintiffs to be disclosed by the servers to other users.

123. Defendants' "heaven" malware accessed the Target Device's memory and used information gathered in that manner to identify the position of memory space used by libwhatsapp.so on the Target Device. This position is not intended to be disclosed outside of the WhatsApp Client Application, to any other user, or even to the legitimate user of the Client Application due to its sensitivity; indeed, this information was instrumental to "heaven" in carrying out the subsequent steps of the exploit chain.

124. As previously discussed, Defendants' Pegasus Spyware is designed to covertly install on a Target Device for the purpose of monitoring and collecting sensitive information from the device. I was not provided with copies of Pegasus or other variants of spyware developed by Defendants, which limited my ability to assess the full impact of Defendants' Malware on Target Devices and the WhatsApp Client Applications. However, my opinion is that the exploit and associated malware discussed by this report served as a delivery mechanism for Pegasus and/or other variants of Defendants' Malware, given that the evidence I have reviewed demonstrates that the WhatsApp Servers and Target Devices were specifically targeted by the exploit as the vector for delivery of a final-stage malicious payload.

125. Given Defendants' acknowledged business in the development and marketing of malware, I assess that, as a result of the observed exploitation activity, a final-stage payload such as Pegasus would have enabled Defendants and/or their customers to

monitor and collect information from Target Devices.[65] Defendants' business model entails the delivery and installation of malware they have engineered for the purposes of remotely monitoring their specified targets. As mentioned above, Defendants go to great lengths to reliably deploy their exploit payloads and to obfuscate or destroy evidence of this intrusion. Thus, it is my opinion that Defendants and/or their customers did leverage this exploit to target specific WhatsApp end users with the end goal of surreptitiously collecting information from these users' devices, including their WhatsApp communications.

Evading Detection

Concealment of Defendants' Malware

126. Defendants' "heaven" malware was designed to interact with WhatsApp Servers and targeted WhatsApp Client Applications by creating messages which appeared to be from a legitimate WhatsApp Client. During this process, "heaven" imitated characteristics of transmissions generated by the genuine WhatsApp Client, and used credentials and keys derived from authentic WhatsApp user accounts. Defendants' "heaven" malware took steps to conceal the malformed and malicious nature of its communications from Plaintiff's official WhatsApp Servers and targeted WhatsApp Clients.

127. Defendants engineered their Malware to conceal its access to and use of WhatsApp Servers and Target Devices. Defendants' Malware interacted with WhatsApp Servers as well as with the WhatsApp Client installed on Target Devices by creating messages that appeared to be from a legitimate WhatsApp Client Application. During this process, Defendants' engineered malware, which in my opinion should be considered spyware, reproduced transmission properties of the WhatsApp Client Application, and used credentials and keys retrieved from authentic WhatsApp user applications. This activity was used to disguise Defendants' Malware as a valid WhatsApp user application in order to access and use both WhatsApp's Servers and the Target Devices. As described in this process, Defendants' Malware deceived WhatsApp Servers and Target Devices' WhatsApp Client Applications about the type of and the source of the transmission between Defendants' Malware and those devices.

128. Defendants at various points leveraged XOR encryption to obfuscate the data transmitted by their Malware to targeted devices by way of the WhatsApp Relay Servers. This technique served to mask the malicious nature of certain RTCP packets crafted and transmitted by Defendants. Defendants' Malware used the RTCP packet header, which normally contains transportation information for the packet, as a key with which to

[65] Compl. Ex. 10, Dkt. No. 1.

enable encryption. The resultant packets resembled RTCP_REMB packets, which are used by the legitimate WhatsApp systems to control media congestion by indicating the rate at which media can be sent. The XOR encryption ultimately served to "camouflage" the appearance of Defendants' packets among normal network traffic, limiting the probability of detection by WhatsApp engineers and/or security mechanisms.

129. Upon exploiting the buffer overflow vulnerability, Defendants' Malware took discernible and deliberate actions to "clean up" evidence of exploitation which may have remained on the Targeted Devices. The first callback pointer overwritten by "heaven" pointed to a regular function of the WhatsApp application which organized and resets certain parts of the process' memory. This function removed information from the memory of the Target Device as well as data related to Defendants' Malware activity, which may have been indicative of Defendants' Malware's previous behavior. After resetting these portions of memory space, the affected system would have appeared closer to the baseline of a running WhatsApp application.

130. Each of the executable files which were written to the disk by Defendants' various exploit payloads were erased after completing their functions. In addition, based upon my study of the full exploit chain, it appears that Defendants' final-stage payload was intended to be loaded directly into executable memory rather than written to the device's storage. This is a common technique used by cyber threat actors to evade detection and inhibit efforts by incident responders or forensic analysts to identify and analyze their malicious software.

Defendants' Malware Altered, Damaged, or Deleted Data from WhatsApp Servers and Target Devices

131. While executing the buffer overflow exploits, Defendants were inherently altering or damaging targeted WhatsApp Client Applications. As demonstrated in previous sections, the buffer overflow altered and corrupted memory on Target Devices, causing them to behave in ways not designed or intended by Plaintiffs when developing the Applications.

132. Further, when Defendants' malicious code was executed against Target Devices, this triggered other actions which reset the Target Device's memory, deleted temporary files, and downloaded a second-stage payload to complete the compromise of the Target Device.

Defendants' Malware Development and Circumvention

Defendants' Reverse Engineered and Tested the Design of its Software to Work Against WhatsApp Infrastructure

133. I have not received or reviewed Defendants' financials or any other materials directly related to the organization of Defendants' business. However, based on the evidence at hand, including the deposition of the NSO's CEO, my expert opinion is that while the stated core of Defendants' business is in developing and marketing the Pegasus Spyware, Defendants' business is in large part dedicated to the discovery and weaponization of zero-day software exploits with which to deliver their Malware. I assess that Defendants maintain and invest massively in an in-house exploit research and development apparatus, and that the "heaven" exploit chain and associated software discussed here is a product of that apparatus.

134. The legitimate WhatsApp Client Application does not provide users with the functionality necessary to activate the XOR cipher setting, nor to set the value of the "connecting_tone_desc" field within the call offer message, nor to add freeform text to that field. The manipulated offer message prepared during step two of the exploit chain therefore had to have been prepared and sent using a modified or fake WhatsApp client, i.e., a program intended to emulate a WhatsApp Client Application from the perspective of the WhatsApp Servers, in order to circumvent the technological limitations of the legitimate WhatsApp Client Application. My understanding is that Defendants' "heaven" malware and/or its associated "WIS" component did in fact serve in part to emulate a genuine WhatsApp Client Application in order to carry out the attacks.

135. The buffer overflow vulnerability at the center of the exploit chain would have required a significant amount of time and effort dedicated to testing in order to be discovered and leveraged to achieve arbitrary code execution against the targeted device. Achieving arbitrary code execution through a buffer overflow attack against modern systems and applications requires a great deal of precision, as corrupting the wrong pieces of data in the targeted system's memory is likely to result in an application or system crash.

136. The exploit chain demonstrates that Defendants possessed a detailed understanding of the memory layout of the WhatsApp Client Application for Android, and particularly of its library component libwhatsapp.so. Defendants maintained knowledge of the memory offsets for various pieces of data and functions within libwhatsapp.so applicable to each of multiple different versions of the WhatsApp Client Application and two underlying device architectures (32-bit and 64-bit). These memory offsets were essential to Defendants in their carrying out of the exploitation against WhatsApp users, as without an intimate understanding of the memory layout, the buffer overflow vulnerability could not feasibly have been made to execute malicious code. This type of information is not publicly documented by Plaintiffs, and in my opinion could only have been derived from Defendants' methodical reverse engineering of each relevant version of the WhatsApp Client Application.

CONCLUSION

156. Based on the evidence available to me that I have had a reasonable opportunity to analyze as of the date of this report, including evidence provided by Plaintiffs, evidence provided by Defendants, and evidence provided to Counsel by DOJ, I have reached the conclusion that Defendants accessed, modified, and exploited, without authorization from Plaintiffs, Plaintiffs' software and infrastructure in May 2019. I have further concluded that Defendants continuously worked to circumvent any technical measures implemented by Plaintiffs in order to maintain access to Plaintiffs' servers and Target Devices for the purpose of deploying spyware.

157. I concluded that Defendants' Malware exploited and accessed Plaintiffs' architecture and subsequently the Target Devices.

158. I concluded that Defendants' Malware is spyware, in that it was implemented to collect information from Target Devices without the Target Devices' users' awareness, and was used to deliver an additional malicious payload which I assess was likely Defendants' Pegasus Spyware product.

159. I concluded that Defendants' Malware was designed to evade detection by Plaintiffs and was designed over time to adapt to changes and updates in Plaintiffs' infrastructure, including technical measures specifically intended to prevent such activity. I determined that Defendants' exploitation was, in fact, successful due to an optimization, or update, in Android Target Devices, which allowed Defendants to manipulate communications and deliver the exploitation.

160. I concluded that Defendants conducted extensive reverse engineering of Plaintiffs' software and infrastructure in order to develop more effective malware.

161. Finally, I came to the conclusion that at least 1,500 Target Devices users appeared to have been targeted, exploited, and impacted by Defendants' Malware during the specific timeframe, making them victims to Defendants' Malware through Defendants' unauthorize access of and use of Plaintiffs' servers. Further, based on Defendants' business model, I have no reason to suspect Defendants stopped this activity after May 2019.

Dated: August 30, 2024

DocuSigned by:
David Youssef
75FB98DF156B485...

David Youssef

EXHIBIT B

FILED UNDER SEAL

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

WHATSAPP INC., et al,. Plaintiffs, v. NSO GROUP TECHNOLOGIES, et al., Defendants.	)	Case No.: 19-cv-07123-PJH

Expert Rebuttal Report of David Youssef

September 21, 2024

September 21, 2024

DAVID YOUSSEF
EXPERT REBUTTAL REPORT

IN RE: WHATSAPP INC., ET AL (PLAINTIFFS), V. NSO GROUP TECHNOLOGIES LIMITED, ET AL., (DEFENDANTS),
CASE NO.: 19-CV-07123-PJH

designed to overwrite two callback pointers[34] in the Target Device's memory to point towards the memory address values set by the two attacker phone numbers in steps five and six. By overwriting these pointers, Eden forced the WhatsApp Client Application to execute functions specified by the addresses in the malicious capabilities buffers, which at this point were present in the Target Device's memory.

48. In his deposition, when asked about the overwriting of callback pointers by Eden, Mr. Gazneli responded in the affirmative.[35]

49. As detailed within the "Conversion to a Group Call" section of his report, Mr. McGraw describes actions taken by the exploit to send an additional capabilities buffer from "LI Phone 2" to the Target Device. Mr. McGraw states that this capability buffer causes "the target device to set additional memory pointers to specific locations."[36] Mr. McGraw does not provide any additional detail covering the specific actions of altering the callback pointers described in this step of the exploit chain.

Step 8: Call Termination and Compromise of Target Device

50. At this point in the exploit chain, Eden terminated both calls by sending a message to the WhatsApp Signaling Server before the victim could answer or decline. This step caused the Target Device to initiate the hijacked instructions established in steps five through seven of the exploit. Due to the changes made by Eden to the Target Device's memory, the end of the call triggered the modified pointers, thus executing the malicious bash script inserted in step 2.

51. In his deposition, when asked about this, Mr. Gazneli responded in the affirmative. Mr. Gazneli also confirmed what is stated in the Youssef report regarding triggering of the modified pointers stating this was done "to be able to control the memory allocations base to be able to write specific code to be downloaded and executed."[37]

52. Mr. McGraw states that, following the main steps of the exploit chain, "LI Phone #1 and LI Phone #2" would end the call with the Target Device causing it to "run its call termination routines". Mr. McGraw confirms my understanding with his explanation of this step of the exploit chain, specifically when he states that the call termination routine would "cause the target device to execute shell code that was delivered in the initial stages of the

[34] A callback pointer is a pointer to a function which can be passed to another function, allowing that function to "call back" and execute the pointed-to function. By overwriting a callback pointer, an attacker can potentially modify the logic and execution flow of the program to induce unintended behavior.

[35] Gazneli Dep. Tr. at 309:21-310:8.

[36] McGraw Report ¶ 79, at 16.

[37] Gazneli Dep. Tr. at 311.

exploit."[38] I understand Mr. McGraw's use of the term "shell code" here to refer to the bash script which was inserted in step 2, which can also be described accurately as a "shell script;" he may instead be referring to the hexadecimal machine instructions contained within that script (normally written as "shellcode"), but in any case is referring to the malicious payload inserted within the "connecting_tone_desc" field of the call offer message.

2. The McGraw Report Misrepresents How Defendants' Exploit Worked

Mr. McGraw misrepresents how Defendants exploited WhatsApp to deliver spyware to Target Devices.

53. In section V of his report, "Pegasus's Use of WhatsApp Servers," Mr. McGraw states that the exploit referred to by the Complaint "purportedly used vulnerabilities in the WhatsApp application"[39] and "merely used the WhatsApp servers to exchange data with the target device, and it did so using functions written by WhatsApp, functioning as WhatsApp had intended those functions to work."[40] This is incorrect and misleading. Based on the evidence available for my review, Defendants exploited WhatsApp's server-side infrastructure in addition to the "WhatsApp application." The vulnerability central to the operation of Defendants' Heaven and Eden exploits, CVE-2019-3568, is documented as a "buffer overflow vulnerability in WhatsApp VOIP stack [allowing] remote code execution via [a] specially crafted series of RTCP packets sent to a target phone number."[41] The term "WhatsApp VOIP stack" here refers to WhatsApp's VoIP architecture as a whole unit, including the server-side infrastructure that supported use of the WhatsApp Client Application during the relevant time period in 2019. Indeed, I concluded that Defendants developed automated malicious code to systematically exploit the WhatsApp Relay Servers, WhatsApp Signaling Servers, and WhatsApp Client Application, ultimately in order to surreptitiously deliver malicious spyware payloads to Target Devices.

54. As acknowledged by Defendants[42] and detailed in the Youssef Report, the malicious data delivered to Target Devices during Defendants' exploit process originated not from any legitimate WhatsApp Client Application, but from software designed and developed by Defendants for the express purpose of exploiting WhatsApp. During the deposition of Mr. Gazneli, he testified about the WIS, and when asked if "NSO created its own source client

[38] McGraw Report ¶ 80, at 16.

[39] McGraw Report ¶ 14, at 3.

[40] McGraw Report ¶ 81, at 16.

[41] https://nvd.nist.gov/vuln/detail/CVE-2019-3568.

[42] Gazneli Dep. Tr. at 160-162.

in WIS that had certain of those same capabilities" as the legitimate WhatsApp Client Application but was "not an actual WhatsApp client" and only "uses part of the protocol capabilities to be sent from one source to the target", Mr. Gazneli responded in the affirmative.[43] Mr. McGraw appears to acknowledge this fact, generally referring to the origin of the exploit data as "Pegasus," though in paragraph 76 of his report, he describes the malicious call offers as having originated from "a client application controlled by the investigator." He goes on to refer to this stage and further stages of the exploit chain as having originated from "LI Phone #1" and/or "LI Phone #2."[44] This is, in my opinion, misleading, and Mr. McGraw fails to explain that none of this data originated, or could have originated, from a legitimate WhatsApp Client Application or a typical mobile device.

55. While Mr. Gazneli testified that "WIS has a connection to a client which is activated through the normal procedures of client activation," and that "the message is created by the WIS but transmitted through that WhatsApp client,"[45] in my opinion, this testimony is misleading. I have seen no evidence that the WIS utilizes the legitimate WhatsApp Client Application, and the messages sent by Defendants' malware could not be sent by the legitimate WhatsApp Client Application without modifying it and bypassing its technological limitations. In my opinion, Mr. Gazneli more accurately described the WIS when he stated "[t]his is not an actual WhatsApp client. It uses part of the protocol capabilities to be sent from one source to the target."[46] Mr. Gazneli appears to be using "WhatsApp client" to refer not to the legitimate WhatsApp Client Application, but to the messages sent "through an active WhatsApp account,"[47] using actual but misappropriated WhatsApp authentication credentials as explained in the Youssef Report.

56. Mr. McGraw describes the first stage of Defendants' exploit process as follows:

> "Pegasus would obtain information about the [T]arget [D]evice that was normally exchanged during a WhatsApp VoIP call. This included determinations of whether the call recipient has a WhatsApp account, what operating system the recipient is running, whether the recipient is currently online, and whether the WhatsApp application was in the foreground."[48]

57. This description is misleading and factually incorrect. Information about the operating system used by the device on the receiving end of a WhatsApp VoIP call is not ordinarily

[43] Gazneli Dep. Tr. at 161-162.

[44] McGraw Report ¶ 76-80, at 15-16.

[45] Gazneli Dep. Tr. at 187-188.

[46] Gazneli Dep. Tr. at 161-162.

[47] Gazneli Dep. Tr. at 186.

[48] McGraw Report ¶ 75, at 15.

disclosed to the caller or the caller's WhatsApp Client Application, nor is information about whether the callee's WhatsApp Client Application is in the foreground at the time of the call's initiation. Shortly after erroneously describing the fingerprinting stage in this manner, Mr. McGraw acknowledges that "information indicating the version of the operating system on the device" was returned to Pegasus "[d]ue to the buffer overflow vulnerability in the WhatsApp code."[49]

58. Defendants' internal documentation[50] describes a process, complex in its own right, which was designed by Defendants to circumvent WhatsApp's protections against the disclosure of detailed device information to other WhatsApp users. This process, referred to by Defendants as the [redacted]," involved [redacted] This [redacted], which Defendants' software induced WhatsApp's servers to send to the Target Device, would manipulate the target's WhatsApp Client Application to disclose more information than was necessary or intended to be sent to its peer (the other WhatsApp Client Application involved in the communication) by appearing to be a message originating from a WhatsApp server. Although the message is the type that the server could send, a WhatsApp user using the WhatsApp Client Application cannot send such a message, nor obtain the information returned to the server by the peer (i.e., the other WhatsApp Client Application involved).

59. The same documentation further describes a second process for obtaining the information, referred to as the "Old Fingerprint." This process also involved "send[ing] a request . . . mimicking the server's response to the target's profile pic request." Defendants explicitly state that "this request is not sent by a legitimate WhatsApp client, only by our [NSO's] client." Both of these fingerprinting methods were clearly designed by Defendants to obtain information from Target Devices which would not ordinarily be received in the course of a WhatsApp VoIP call.

60. In paragraph 79 of his report, Mr. McGraw states that the two WhatsApp accounts and corresponding devices simulated and controlled by Pegasus sent the Target Device "a large list of capabilities" and "a specific set of capabilities . . . [which] caused the [T]arget [D]evice to set additional memory pointers to specific locations."[51] Here Mr. McGraw again fails to adequately describe the process. In my assessment, the capabilities buffers sent by each of the accounts controlled by Pegasus were not true capabilities buffers whatsoever, in that they did not contain any of the usual data which would describe capabilities in a

[49] McGraw Report ¶ 78, at 15-16.

[50] NSO_WHATSAPP_00008957.

[51] McGraw Report ¶ 79, at 16.

IV. CONCLUSION

100. In the McGraw Report, Mr. McGraw details five (5) opinions regarding Defendants' use of WhatsApp servers in the delivery of its malware to Target Devices. After reviewing the McGraw Report, the cited materials, additional material provided by Counsel, and after conducting my own additional research, some of which is detailed in the Youssef Report, it is my opinion that Mr. McGraw incorrectly concludes that Defendants activity, notably its access and use of WhatsApp servers, falls within the boundaries of lawful intercept as defined in this report. Further, Mr. McGraw understates the degree to which Defendants sought to circumvent technical and procedural measures put in place by Plaintiffs to prevent such activity.

101. It is my conclusion that Defendants did not merely use WhatsApp servers or the WhatsApp Client Application as they were intended to be used, as Mr. McGraw opines, but deliberately researched and reverse-engineered WhatsApp servers to exploit them in unintended and impermissible ways to convert them into a delivery mechanism for Pegasus to Target Devices.

102. It is my conclusion that Mr. McGraw disregards the evidence that Defendants intentionally targeted WhatsApp servers and were willfully blind to the location of such servers, including those located in the United States and specifically in California.

103. It is my conclusion that Mr. McGraw mischaracterizes the manner in which Defendants gained access to WhatsApp servers and disregards the technological barriers that Defendants circumvented in the process of using WhatsApp servers to deliver Pegasus to Target Devices.

104. Further, it is my conclusion that Mr. McGraw improperly downplays the harm that Defendants caused to WhatsApp and the degree to which the activity was performed by Defendants rather than Defendants' customers.

Dated: September 21, 2024

DocuSigned by:
David Youssef
75FB98DF156B485...

David Youssef