the adversary playbook for the long-standing espionage activity of a Chinese nation-state adversary

Alex Hinchliffe

Unit 42, Palo Alto Networks, UK

Table of contents

Abstract

The discovery of two malware families – HenBox for *Android* and, recently, Farseer for *Windows* – with significant, mostly infrastructure-based overlaps with previously seen malware, such as 9002, PlugX, Poison Ivy and FHAPPI, has led us towards what appears to be an undocumented nation-state group, or groups, in China that we refer to as PKPLUG. The malware families, infrastructure, and campaign delivery used by PKPLUG highlights broad targeting of multiple sectors and victims in and around the Southeast Asia region and beyond. This research will detail some of the PKPLUG campaigns, describing the tooling used and, with MITRE's ATT&CK framework and other models that underpin *Unit 42*'s adversary playbooks, highlight PKPLUG's behaviour with some overlapping TTPs.

Introduction

PKPLUG

Unit 42 uses the moniker 'PKPLUG' in reference to a threat actor group, or groups, that we have been tracking for a few years. The name comes from the adversary's use of PlugX malware, which we noted in their early campaigns, and from the use of ZIP archive files to deliver the malware; the ZIP file format contains the ASCII magic bytes 'PK'.

Over the years, *Unit 42* has investigated PKPLUG and has discovered further malware families being used, including other custom malware for *Android* and *Windows* that will be described later in this report. Other malware families that have been seen relating to PKPLUG include 'usual suspects' Poison Ivy, Zupdax and 9002.

Based on targeting, the content in some of the malware, and ties to infrastructure documented publicly as being linked to Chinese nation-state adversaries, *Unit 42* believes with high confidence that PKPLUG has similar origins.

Targeting

Based on our observations of PKPLUG's campaigns and what we've learned from sharing with industry, we believe that its victims lie mainly in and around the Southeast Asia region. This report will provide further details, but to be more specific, considering the methods used for malware delivery, the social engineering topics of decoy applications

and documents used, and the command-and-control (C2) infrastructure themes, target countries include (with higher confidence): Xinjiang, Mongolia, Myanmar and Taiwan; and (with lower confidence): Tibet, Vietnam and Indonesia.

Three of these countries are ASEAN members [1], contributing towards intergovernmental cooperation, and another three are autonomous regions (AR) [2] of China that tend to be classified by China's ethnic minorities, granted the ability to govern themselves but ultimately answering to the People's Republic of China (PRC). Of the five autonomous regions, Tibet and Xinjiang are the only ones in which the ethnic group maintains a majority over other populations.

Most, if not all seven target countries are involved in some way with Beijing's Belt and Road Initiative (BRI) [3], designed to connect 71 countries across Southeast Asia to Eastern Europe and Africa. The path through Xinjiang is especially important [4] to the BRI's success but is more often heard about due to conflicts [5] between the Government and the ethnic Uyghur population [6]. News of the BRI is peppered with stories of success and failure; of countries opposed to it, or buying into or pulling out of BRI projects.

Further tensions in the region are attributed to disputes over ownership of the South China Sea, including disputes over fishing quotas and the yet unproven oil and gas reserves [7]. At least three of the target countries mentioned above have laid claim to parts of these waters, and some use the area for the vast majority of their trade; foreign militaries are also involved, attempting to keep the area open.

Taiwan (a.k.a. Republic of China), which isn't an AR, and which doesn't appear to be actively involved with the BRI, has its own long-standing history with the PRC; a recent $2BN+ arms sale with the US [8] may exacerbate matters.

The ultimate objective of PKPLUG is not entirely clear, but the backdoors and espionage malware used indicate that tracking victims and gathering information is key.

Xiaomi

HenBox malware, described later in this report, references *Xiaomi*, hence this section.

Xiaomi is a firm that designs, develops and sells smartphones, mobile apps, laptops and related consumer electronics. The firm released its first smartphone in August 2011 and rapidly gained market share in China where it became the largest smartphone company in 2014. In 2017, *Xiaomi* became the world's fifth largest smartphone company and overtook *Samsung* to become the number one smartphone brand in India. Now in fourth place in worldwide smartphone manufacturers behind *Apple*, *Huawei* and *Samsung*, the firm has yet to enter the US smartphone market space.

Figure 1 shows *Xiaomi*'s performance against other manufacturers. According to the International Data Comparison (IDC), Asia Pacific (excluding Japan) remains *Xiaomi*'s most important region, with China, India and Indonesia accounting for the majority volume [9].

Worldwide Quarterly Smartphone Top 5 Company Shipments, 2019Q1 and 2018Q1 (Shipments in millions)

Company	1Q19 Shipment Volumes	1Q19 Market Share	1Q18 Shipment Volumes	1Q18 Market Share	Year-Over-Year Change
1. Samsung	71.9	23.1%	78.2	23.5%	-8.1%
2. Huawei	59.1	19.0%	39.3	11.8%	50.3%
3. Apple	36.4	11.7%	52.2	15.7%	-30.2%
4. Xiaomi	25.0	8.0%	27.8	8.4%	-10.2%
5. vivo*	23.2	7.5%	18.7	5.6%	24.0%
5. OPPO*	23.1	7.4%	24.6	7.4%	-6.0%
Others	72.1	23.2%	91.9	27.6%	-21.5%
Total	**310.8**	**100.0%**	**332.7**	**100.0%**	**-6.6%**

Source: IDC Quarterly Mobile Phone Tracker, April 30, 2019

Figure 1: Xiaomi's performance against other manufacturers.

Over the years *Xiaomi* has also expanded into the smart home and IoT device ecosystem, producing many devices for the smart home, managed by the *MiHome* app for smartphones.

Adversary playbooks

Unit 42 [10] is the threat intelligence team at *Palo Alto Networks* that analyses available data to identify adversaries, their motivations, resources and tactics in order to better understand the threats our customers face. Adversary playbooks provide a threat intelligence package in STIX 2.0 for ingestion by machines for research or protection purposes. These packages also include structured details about attack campaigns and adversary behaviours – their tools, techniques, and procedures (TTPs) – as well as the expected indicators of compromise (IOCs). *Unit 42* aims to release adversary playbooks alongside research published.

The concept of adversary playbooks is straightforward: just as sports teams create offensive and defensive playbooks to win matches, adversaries also have offensive playbooks they employ during cyber attacks in an attempt to compromise organizations.

Network defenders, threat researchers and others can create adversary playbooks through observation of live or past attacks; by sharing data; and through intelligence analysis. Those playbooks can then be used to better defend networks and describe threat actor groups. Combining multiple playbooks, and thus others' visibility and data sets for the same attack or adversary, will ultimately provide a much better picture of the opposition we face.

In order to be successful and useful for many different use-cases, adversary playbooks must use a structured format that can be shared. We decided not to develop a proprietary format that would potentially make it exclusive to *Palo Alto Networks*, and instead we make use of Mitre ATT&CK [11], Attack Lifecycle or Cyber Kill Chain(™) [12] and STIX [13].

Malware used by the PKPLUG adversary

We know that the attacks carried out by the PKPLUG actor used multiple malware families, all of which provide backdoor, remote access and spying capabilities. The following sections describe only the newly discovered malware – HenBox and Farseer – in more detail.

HenBox for Android

In early 2018, *Unit 42* discovered [14, 15] a new *Android* malware family that we named 'HenBox' based on metadata, such as app package names and developer signer information, found in most of the malicious apps analysed. At the time of writing, *Unit 42* is tracking over 400 HenBox samples dating back as far as late 2015, and continuing to the present day.

HenBox often masquerades as legitimate *Android* apps, such as virtual private network (VPN) apps, *Android* system apps and so on. Occasionally, HenBox will install legitimate versions of these apps as well as itself, tricking users into thinking they have installed the desired app. Whilst some of the legitimate apps HenBox uses for such decoys can be found on the official *Google Play* app store, HenBox apps themselves have only been found on third-party (non-*Google Play*) app stores.

HenBox appears primarily to target the Uyghurs – a minority Turkic ethnic group that is primarily Muslim and lives mainly in the Xinjiang Uyghur autonomous region in Northwest China. It also targets devices made by Chinese manufacturer *Xiaomi* and those running *MIUI*, an operating system based on *Google Android* made by *Xiaomi*. Smartphones are the dominant form of Internet access in the region [16], and Xinjiang was recently found to have a higher number of Internet users than the national average in China [17]. The result is a large online population that has been the subject of numerous cyber-attacks in the past [18, 19, 20, 21].

Once installed, HenBox steals information from the device from a myriad of sources, including many mainstream chat, communication and social media apps. The stolen information includes personal and device information. Of note, in addition to tracking the location of the compromised device, HenBox also harvests all outgoing phone numbers with a '+86' prefix, which is the country code for the People's Republic of China (PRC). It can also access the phone's microphone and cameras.

Delivery via third-party app store

Of the 400+ samples *Unit 42* has seen, the vast majority, if used in attacks, have no associated delivery method. It is believed that such apps, as with many other malicious *Android* apps, would be delivered to victims via websites or file-sharing forums, possibly from links shared in phishing emails or SMS messages. Social media platforms and messaging – which support the larger file sizes often needed for *Android* package (APK) files – could also be used. The large file size is the reason phishing emails with HenBox attachments are unlikely to be the delivery mechanism.

In May 2016, a HenBox app – an APK file – was downloaded from the uyghurapps[.]net website. The domain name, language of the site and app content hosted on the site suggest that this is a third-party app store for which the intended users are the Uyghurs. Third-party app stores are so called because they are not officially supported by *Android*, and they are not provided by *Google*, unlike the *Play Store*. Third-party app stores are ubiquitous in China for a number of reasons, including increasingly powerful Chinese original equipment manufacturers (OEMs), a lack of an official Chinese *Google Play* app store, and a growing smartphone market.

At the time of analysis, the uyghurapps[.]net website hosted a number of secure communication, VPN and social media apps. Given what we know from the media about the region, it's clear that such apps are critical for the population to protect themselves and communicate with others.

The HenBox app downloaded from uyghurapps[.]net was masquerading as an another app, *DroidVPN*. At the time of analysis, the content served on uyghurapps[.]net at the URL from which HenBox was downloaded was a legitimate version of *DroidVPN*. The app page, where users can download the app and learn more about it, is shown in Figure 2. It's highly likely that the page looked the same during the time HenBox was available, and that the APK file for *DroidVPN* was simply replaced with a copy of HenBox.

Figure 2: The uyghurapps[.]net app store showing the current DroidVPN app.

VPNs allow connections to remote private networks, increasing the security and privacy of the user's communications. According to the *DroidVPN* app description, it 'helps bypass regional Internet restrictions, web

filtering and firewalls, by tunnelling traffic over ICMP'. Some features may require devices to be rooted in order to function and, according to some third-party app stores, unconditional rooting is required, which has additional security implications for the target device.

Unit 42 has not been able to ascertain how the malicious HenBox app, referenced in Table 1, got onto the app store. However, some open-source intelligence indicates that the server was running an outdated version of *Apache Web Server* on a *Windows* 32-bit operating system. In light of this, we believe an attack against unpatched vulnerabilities, or a brute-force login attack, are reasonable conjectures as to how the server was compromised, ultimately leading to the *DroidVPN* APK file being overwritten with the malicious HenBox APK.

APK SHA256	**Size (bytes)**	**First seen**	**App package name**	**App name**
0589bed1e3b3d623 4c30061be3be1cc66 85d786ab3a892a8d4 dae8e2d7ed92f7	2,740,860	May 2016	com.android.henbox	DroidVPN

Table 1: Details of the HenBox DroidVPN app on the uyghurapps[.]net app store.

As can be seen in Table 1 and Figure 3, despite the unique 'com.android.henbox' package name, the HenBox malware copied the legitimate app's name, 'DroidVPN', as well as its icon to further trick victims into believing they had installed *DroidVPN* and not something totally different.

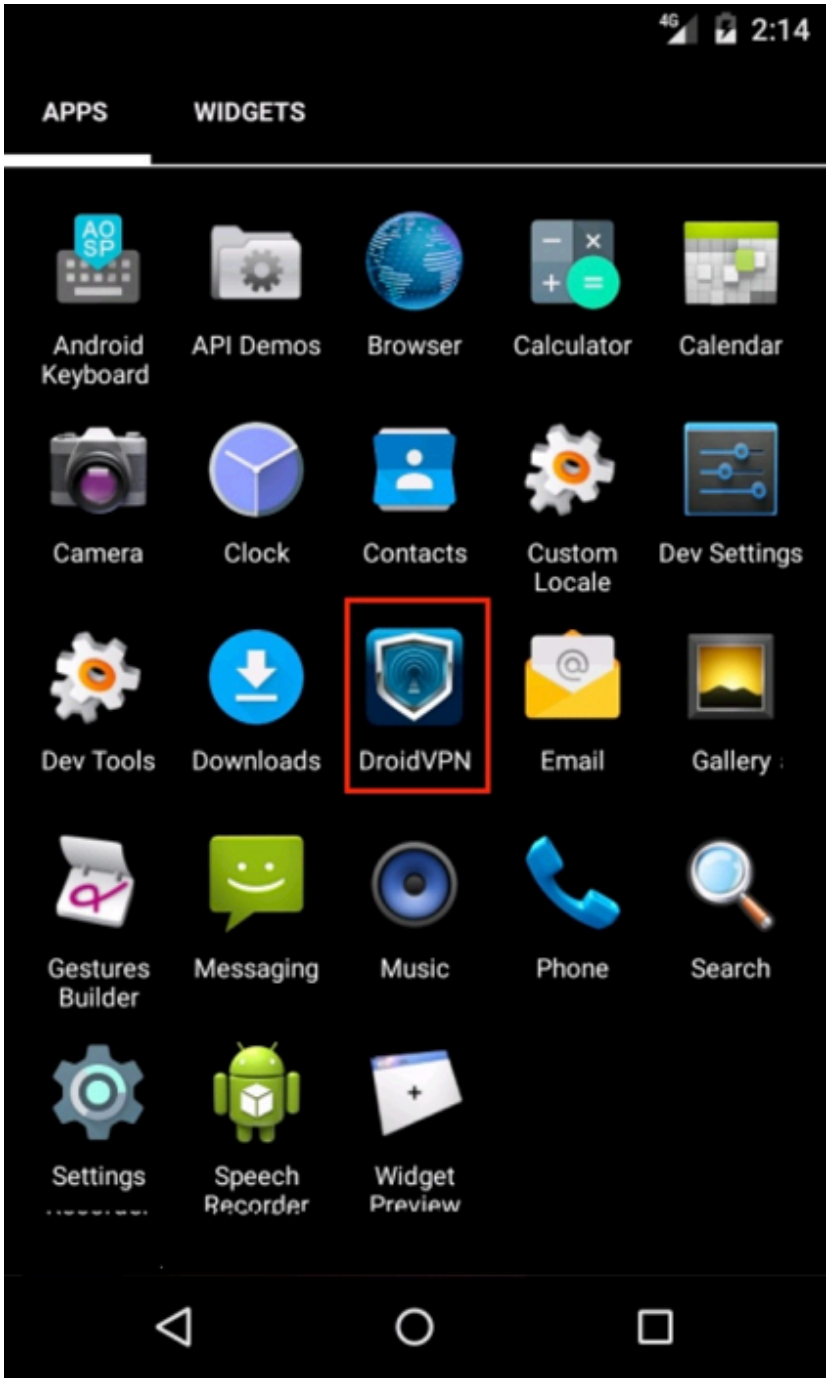

Figure 3: HenBox app installed, purporting to be DroidVPN.

In addition to the look and feel of *DroidVPN*, the HenBox variant also contained a copy of the original, legitimate *DroidVPN* app as an asset within its APK package. Assets can be compared to resource items within a *Windows* Portable Executable (PE) file. Once the HenBox app is installed and run, it executes code that causes the *Android* operating system (OS) to launch the install process for the embedded app. HenBox probably does this for two reasons. First, to act as a decoy to detract from other malicious behaviours occurring in the background, and secondly, to satisfy the victim that they are installing the app they wanted. Whether or not the user is suspicious of the first app installation is unknown to us, but based on the names used in other HenBox variants, such as ‘Backup’ and ‘Settings’, it’s highly likely that the app could be passed off in some instances as a benign activity to backup data pre-install, or to change settings prior to install.

At the time of our research, the version of *DroidVPN* available for download from uyghurapps[.]net matched that of the embedded *DroidVPN* app inside HenBox. It's worth noting that newer versions of the *DroidVPN* app were available on *Google Play* at the time, as well as in some other third-party app stores, which could indicate that uyghurapps[.]net is not very well maintained or up to date with the latest app versions available.

The right app at the right time

The HenBox-with-embedded-DroidVPN app combination is one example of the attackers choosing to mimic a legitimate app in order to compromise their victims. Further combinations included apps that, in their standalone form, were available on *Google Play*, as well as many third-party app stores. Table 2 lists just three further example apps together with their and HenBox's respective metadata.

#	**Parent APK SHA256**	**First seen**	**Package names** (HenBox parent APK) [embedded APK]	**APK app names** (HenBox parent APK) [embedded APK]
1	fa5a76e86abb26e48a f0b312f056d24000bc 969835c40b3f98e5ca 7e301b5bee	April 2016	(com.android.henbox) [com.ziipin.software]	(Uyghurche Kirguzguch) [Emojicon]
2	1749df47cf37c09a92 b6a56b64b136f15ec 59c4f55ec835b1e569 c88e1c6e684	May 2017	(cn.android.setting) [com.apps.amaq]	(设置 (Backup)) [Amaq Agency]
3	4d437d1ac29b1762c c47f8094a05ab73141 d03f9ce0256d200fc6 91c41d1b6e7	June 2017	(cn.android.setting) [com.example.ourplayer]	(islamawazi) [islamawazi]

Table 2: Three example apps with their and HenBox's respective metadata.

The app icons that would be seen and used to launch the app on an *Android* device are shown in Table 3.

#	**Icon**	**App description**
1		First HenBox sample seen with a legitimate app embedded within. The app was a Uyghur language keyboard app targeted at native speakers.
2		Masquerades as Android's Settings app, and has a similar package name. App used the green Bugdroid image for its logo; app name 设置 ('Backup'). Interestingly, the embedded app was 'Amaq Agency', which reports on ISIS-related news.
3		The names for both the parent HenBox and the embedded (media player for news) app were identical - Islamawazi. Islamawazi (a.k.a. Turkestan Islamic Party or 'TIP' [22]) is an

		organization formerly known as the East Turkestan Islamic Party, purported to be an Islamic extremist separatist organization founded by Uyghur jihadists.

Table 3: The app icons that would be seen and used to launch the app on an Android device.

These examples, together with the HenBox app placed on a very specific third-party app store, point clearly to at least some of the intended targets of these malicious apps being Uyghurs, specifically those with a potential interest in, or association with, terrorist groups. The threat actors behind HenBox appear to be choosing the right apps (those that could be popular with locals in the region) at the right time (while tensions grow in this region of China) to ensure a high probability of installing their malware.

HenBox capabilities

HenBox has certainly evolved over the past four years but the structure of the over 400 samples has largely stayed the same. This structure includes multiple component files and native libraries used to achieve the goal of data collection and spying on the victim. Most components are obfuscated in some way, whether it be by simple XOR with a single-byte key, compressing using ZIP or Zlib compression, or encryption using RC4. These components are responsible for a myriad of functions and features including handling decryption, network communications, gaining super-user privileges, monitoring system logs, loading additional Dalvik code files, tracking the device location and more.

The remainder of this section describes at a high level what HenBox is capable of, and how it operates. The description is based on analysis of the sample described in the table below, which was of interest given that its C2 domain, mefound[.]com, overlapped with the PlugX, Zupdax and Poison Ivy malware families discussed in more detail later.

SHA256	Package name	App name
a6c7351b09a733a1b3ff8a0901c5bde fdc3b566bfcedcdf5a338c3a97c9f249b	com.android.henbox	备份 (Backup)

Table 4: HenBox variant used in analysis.

Execution flow

Once this variant of HenBox is installed on the victim's device, the app can be executed in two different ways.

The first method, as depicted in Figure 4, is automatic based on the operating system generating one of a handful of event broadcasts that HenBox registered its intent to process during the app installation process. Examples include events like device reboots, when an app is newly installed, or when a network connection is changed.

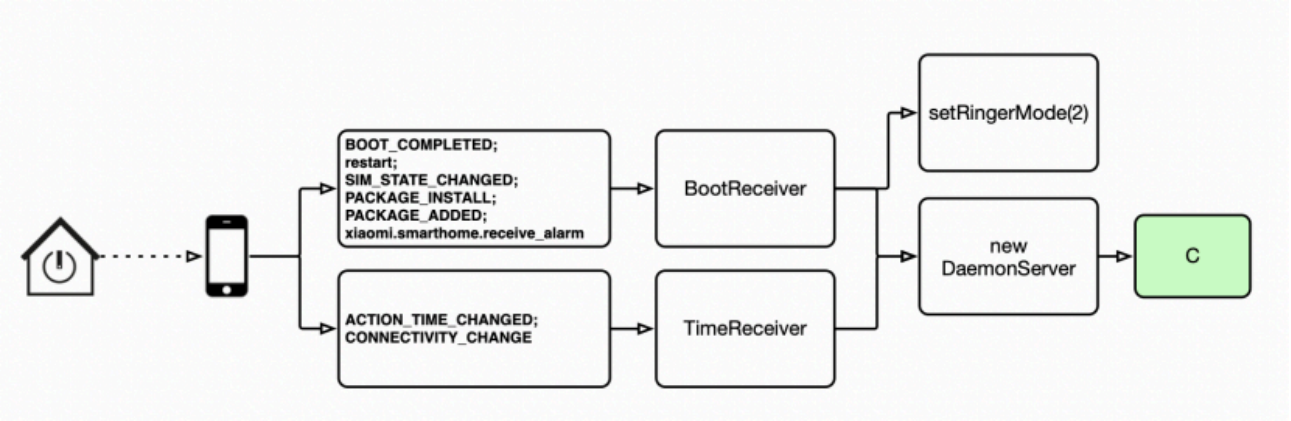

Figure 4: Automatic HenBox execution.

All the intents registered statically via this HenBox variant's AndroidManifest.xml file are listed and described in Table 5; HenBox also registers further intents at runtime.

Receiver	Intent name	Description
BootReceiver	android.intent.action.BOOT_COMPLETED	System notification that the device has finished booting.
	android.intent.action.restart	A legacy intent used to indicate a system restart
	android.intent.action.SIM_STATE_CHANGED	System notification that the SIM card has changed or been removed.
	android.intent.action.PACKAGE_INSTALL	System notification that the download and eventual installation of an app package is happening (this is deprecated).
	android.intent.action.PACKAGE_ADDED	System notification that a new app package has been installed on the device, including the name of said package.
	com.xiaomi.smarthome.receive_alarm	Received notifications from *Xiaomi*'s smart home IoT devices.
TimeReceiver	android.intent.action.ACTION_TIME_CHANGED	System notification that the time was set.
	android.intent.action.CONNECTIVITY_CHANGE	System notification that a change in network connectivity has occurred (has either been lost or established). Since *Android* version 7 (Nougat) this information has been gathered using

		other means – perhaps suggesting that the devices used by potential victims run older versions of *Android*.

Table 5: HenBox variant's intents and receivers defined statically.

Most of the intents listed in Table 5 and shown in Figure 4 are commonly found in malicious *Android* apps and are the equivalent of setting registry run keys in *Windows* to autostart applications at reboot. One intent stands out and is much less common: com.xiaomi.smarthome.receive_alarm.

Given the nature of connected devices in smart homes, it's highly likely they will communicate via alerts and notifications with controller apps, such as *Xiaomi*'s *MiHome*. Because HenBox registers the same intent, it too can process alerts destined for *MiHome* and use them as a trigger to execute code. Essentially, this allows for external IoT devices to act as a trigger to execute the malicious HenBox app's code.

Triggered intents result in execution of code that is present in either the BootReceiver class or the TimeReceiver class, both of which ultimately lead to a new instance of the DaemonServer service being created and started (this service is discussed in more detail later). In addition, BootReceiver changes the device ringer mode to a value of 2, which results in ringtones being audible and the vibrate mode being switched on. This may have been done in an attempt to get nearby people to interact with the (now noisy) device such that the information stolen may be richer in content.

The alternative method for executing HenBox is for the user to launch the malicious app (named 'Backup' in this instance) from the launcher view on their device, as shown in Figure 5.

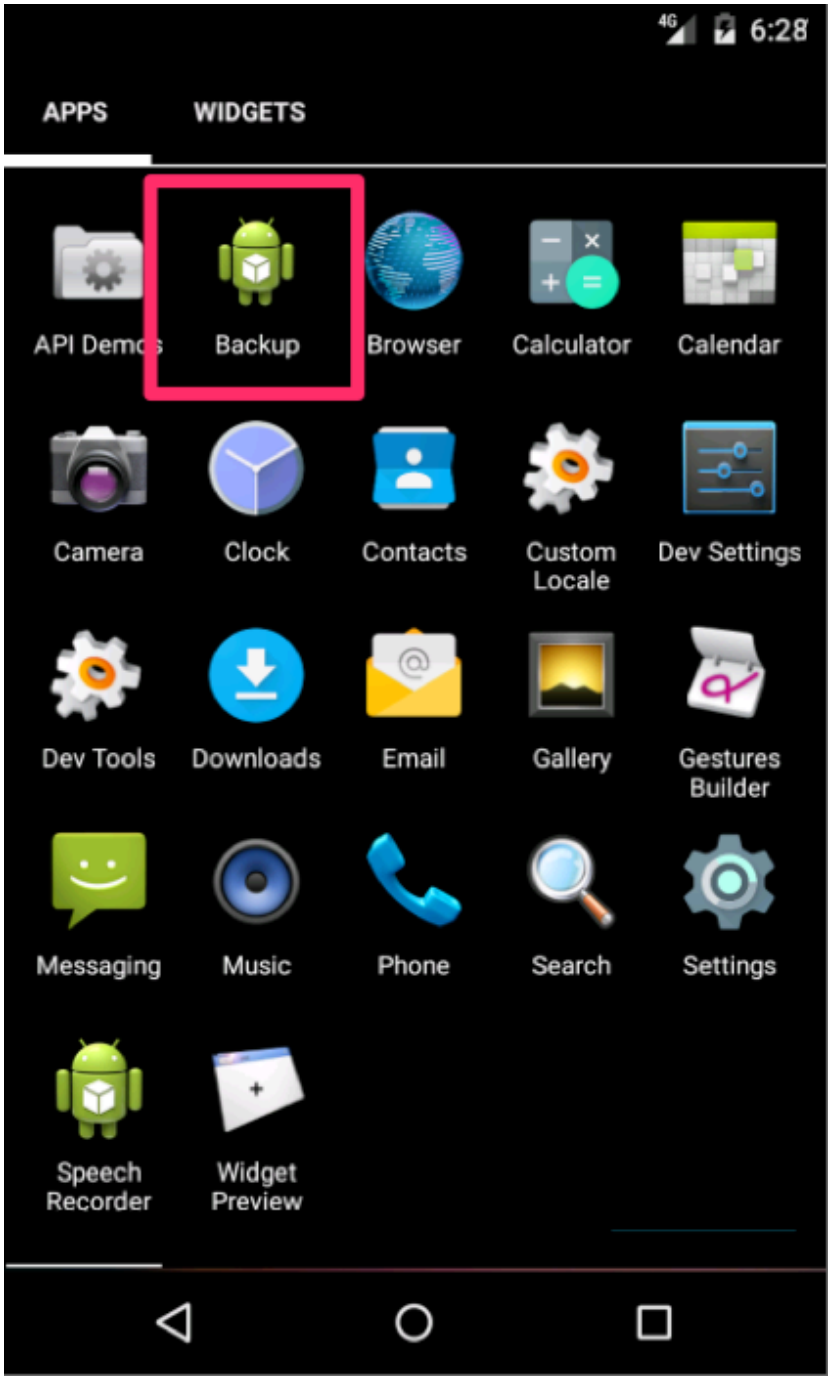

Figure 5: HenBox app installed and visible on Android's launcher view.

Behaviour

Upon manual launch, the HenBox code executes and performs the steps highlighted in Figure 6.

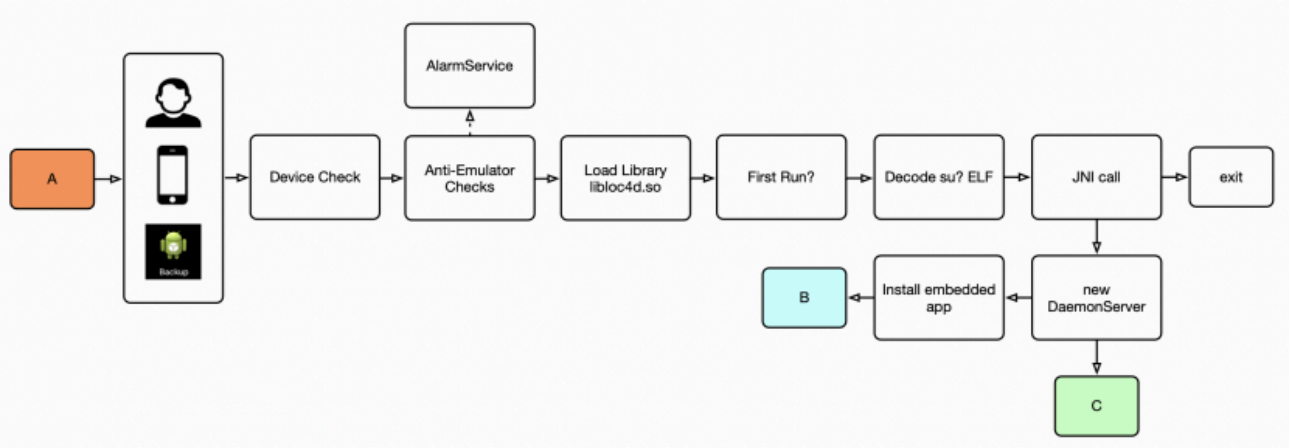

Figure 6: Manual HenBox execution.

First, checks are made to determine whether the device manufacturer is *Xiaomi*, or whether the firmware is *MIUI* (*Xiaomi*'s fork of *Android*). The intention here seems to be one of targeting *Xiaomi* and exiting prematurely if the checks fail. However, poorly written code results in the code being executed in perhaps more environments than the adversary intended. Anti-emulation and anti-debug checks try to ascertain whether HenBox is being analysed. Interestingly, the adversaries concealed their code for these additional checks inside a class called AlarmService, which appears to be a direct copy from online developer tutorials for creating alarm apps. If these checks pass, HenBox continues to execute by next loading the ELF library libloc4d.so.

Using *Android*'s shared preferences feature to persist XML key value pair data, HenBox checks whether this execution is its first. If it is, and if the app's path does not contain '/system/app' (i.e. HenBox is not running as a system app, which provides elevated privileges), one of two embedded 'su?' ELF libraries is XOR-decoded. A Java Native Interface (JNI) call is then issued to libloc4d.so to execute the 'su?' (henceforth sux) binary.

The two files, 'suy' and 'sux', are essentially the same: 'sux' is used if the *Android* version on the victim's device is 4.1 (a.k.a. 'Jelly Bean') or newer; 'suy' will be used for older versions.

Finally, an instance of the DaemonServer service starts and, if a decoy app is embedded inside HenBox, as per the *DroidVPN* example, the installation process for the decoy also starts.

Figure 7 illustrates the typical behaviour of the DaemonServer service, starting with hiding the HenBox app from the launcher view and from the app drawer/tray. This behaviour is common amongst *Android* malware and, while the app remains installed with its services running, it is harder for the victim to discover it.

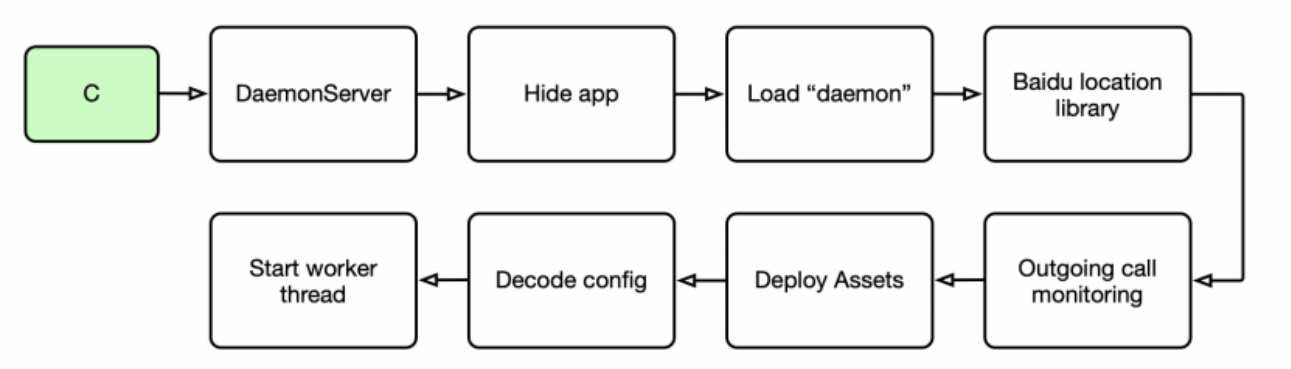

Figure 7: DaemonServer service behaviour.

The non-obfuscated ELF file 'daemon' is loaded next, to gather environmental information about the device by accessing system and radio log files, and by querying running processes.

A *Baidu* library is loaded next, and used to gather device geo-location information.

The DaemonServer class then registers a runtime intent to intercept outgoing phone calls, allowing the checking of numbers dialled. In particular, HenBox filters numbers based on prefixes matching '+86' – the country code for the People's Republic of China.

Further assets are then deployed and decoded, including a.zip and setting.txt – the config file for HenBox. Code is also present in this variant to deploy additional assets named 'plugin' and 'AppVoice' – they are not present in this particular sample, but are a likely indication of evolving development and the use of yet further components.

HenBox's config file, setting.txt, is decoded using XOR with a single-byte key, 0x88; filenames and XOR keys differ occasionally between variants. The config file is shown in the Farseer section later.

Finally, DaemonServer launches a worker thread to perform further execution tasks. One of the key components used is the ELF file named b.dat, which in turn interacts with a.zip. The archive a.zip contains two further files: libkernel.so (another ELF file) and lib.dat, which is actually a Dalvik DEX file containing further Java code and malicious functionality beyond the app's default (and mandatory) classes.dex file. Some of the key data-harvesting behaviour of HenBox stems from these files – b.dat and the contents of a.zip, all four of which are RC4-encrypted, forming the most heavily obfuscated components within HenBox.

Once unpacked and available for use, the new DEX file is executed from within the DaemonServer class to enumerate all running applications and kill those that have the permission to receive SMS messages, before registering its own runtime intent to process them instead, thus intercepting the victim's messages.

The method continues by loading the libkernel.so library file, also unpacked from the a.zip archive. This ELF file has numerous capabilities, many of which come from BusyBox – a package containing various stripped-down Unix tools that are useful for system administration. This executable interacts with the aforementioned sux executable and, amongst other things, temporarily disables the noise made by the device when photos are taken. This behaviour is achieved by moving the audio file '/system/media/audio/ui/camera_click.ogg' elsewhere, then moving it back again once picture-taking is complete.

The variant of HenBox analysed and described in the previous section specifically checked the compromised device for two apps listed in Table 6 below. If the apps are present, HenBox harvests information from them about contacts, numbers and conversations.

Package name	**App name**
com.rebelvox.voxer	Voxer Walkie Talkie Messenger
com.tencent.mm	Tencent's WeChat

Table 6: Targeted messaging apps in August 2017.

These types of apps tend to use databases to store their data, which for *Voxer* is located in '/data/data/com.rebelvox.voxer/databases/rv.db' on the device. HenBox runs SQL queries against the database to

gather their stored information.

A little over four months after this variant of HenBox was seen, newer versions were available with significant changes to the number of targeted apps, as shown in Table 7.

Package name	**App name**
com.whatsapp	WhatsApp Messenger
com.pugna.magiccall	n/a
org.telegram.messenger	Telegram
com.facebook.katana	Facebook
com.twitter.android	Twitter
jp.naver.line.android	LINE: Free Calls & Messages
com.instanza.cocovoice	Coco
com.beetalk	BeeTalk
com.gtomato.talkbox	TalkBox Voice Messenger – PTT
com.viber.voip	Viber Messenger
com.immomo.momo	MOMO陌陌
com.facebook.orca	Messenger – Text and Video Chat for Free
com.skype.rover	Skype; 3rd party stores only

Table 7: Targeted messaging apps in January 2018.

Most of these apps are well established and available on *Google Play*, however, com.skype.rover and com.pugna.magiccall appear to be available only on third-party app stores.

It's clear to see that the capabilities of HenBox are very comprehensive, not only in terms of a complex and pretty sophisticated *Android* app, but also as a very effective spying tool.

Infrastructure and related overlaps

While investigating HenBox, *Unit 42* discovered infrastructure ties to other malware families associated with targeted attacks against *Windows* users, with notable overlaps including PlugX, Zupdax, 9002 and Poison Ivy. Figure 8 paints a picture of an adversary with at least five malware families in its toolbox, dating back to at least 2015.

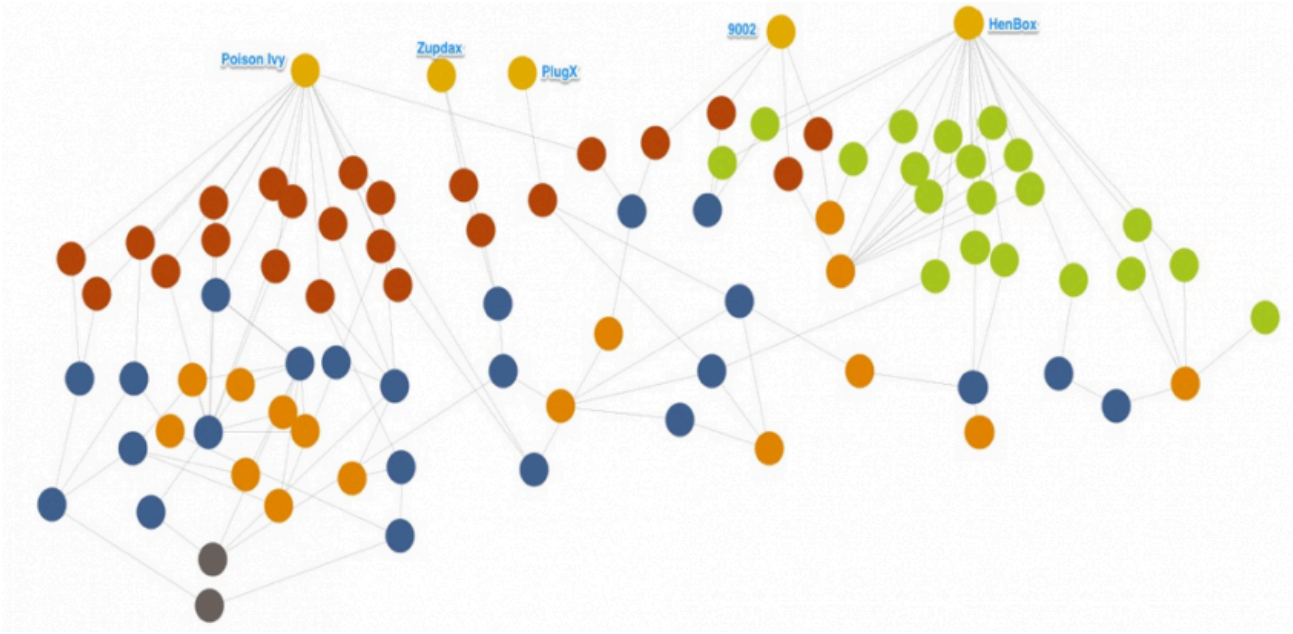

Figure 8: HenBox and related malware.

The overlap between the HenBox and 9002 malware families involves three shared C2s between several samples:

- 47.90.81[.]23
- 222.139.212[.]16
- lala513.gicp[.]net

The overlaps between the HenBox, PlugX, Zupdax and Poison Ivy malware families involves a web of shared C2s and IP resolutions centred around the following:

- 59.188.196[.]172
- cdncool[.]com (and third levels of this domain)
- www3.mefound[.]com
- www5.zyns[.]com
- W3.changeip[.]org

Ties to previous activity

The registrant of cdncool[.]com also registered six other domains. To date, we have seen four of the seven (the first three in the list below, along with cdncool[.]com) used in malicious activity, and it is reasonable to assume that the remaining three are, or were, intended to serve the same purpose.

- tcpdo[.]net
- adminsysteminfo[.]com
- md5c[.]net
- linkdatax[.]com
- csip6[.]biz
- adminloader[.]com

Unit 42 published a blog [23] in July 2016 about 9002 malware being delivered using a combination of shortened links and a file hosted on *Google Drive*. The spear-phishing emails had Myanmar political-themed lures and, if the 9002 C2 server responded, the trojan sent system-specific information along with the string 'jackhex'. 'Jackhex' has also been part of a C2 for what is probably related Poison Ivy activity (detailed below), along with additional infrastructure ties.

The C2 for the aforementioned 9002 sample was logitechwkgame[.]com, which resolved to the IP address 222.239.91[.]30. At the same time, the domain admin.nslookupdns[.]com also resolved to the same IP address, suggesting that these two domains are associated with the same threat actor. In addition, admin.nslookupdns[.]com was a C2 for Poison Ivy samples associated with attacks on Myanmar and other Asian countries and discussed in a blog post [24] published by *Arbor Networks* in April 2016. Another tie between the activities is the C2 jackhex.md5c[.]net, which was also used as a Poison Ivy C2 by the samples discussed in the *Arbor Networks* blog. Finally, since publishing the 9002 blog, *Unit 42* has also seen the aforementioned 9002 C2 being used as a Poison Ivy C2 with a Myanmar political-themed lure.

In our 9002 blog we noted some additional infrastructure used either as C2s for related Poison Ivy samples, or as domain registrant overlap with those C2 domains. When we published that blog we hadn't seen any of the three registrants overlap domains used in malicious activity. Since then, we have seen Poison Ivy samples using third levels of querlyurl[.]com, lending further credence to the idea that the remaining two domains, gooledriveservice[.]com and appupdatemoremagic[.]com, are, or were, intended for malicious use. While we do not have complete targeting information associated with these Poison Ivy samples, several of the decoy files were in Chinese and appear to be part of a 2016 campaign targeting organizations in Taiwan with political-themed lures.

Farseer for Windows

Through further investigations into infrastructure used by the HenBox malware, *Unit 42* discovered [25] another, previously unknown, malware family designed to run on *Windows*.

Farseer – named due to a string found in the PDB path embedded within the executable files (see example below) – is a backdoor trojan that we can trace back in our data to 2016 and that we continue to see in 2019, albeit in small numbers.

```
e:\WorkSpace\A1\coding\Farseer\RemoteShellsRemote\Release\RemoteShellsRemote.pdb.
```

Ties to HenBox

The infrastructure used by the combination of malware families discussed so far is vast, with numerous overlaps, however the remainder of this report will focus only on some core ties between the Farseer and HenBox, PlugX, Zupdax, 9002 and Poison Ivy malware families.

Figure 9 shows a high-level representation of file hashes, IP addresses and domain names used by some of the malware families mentioned, together with their overlaps highlighted by the green rectangle.

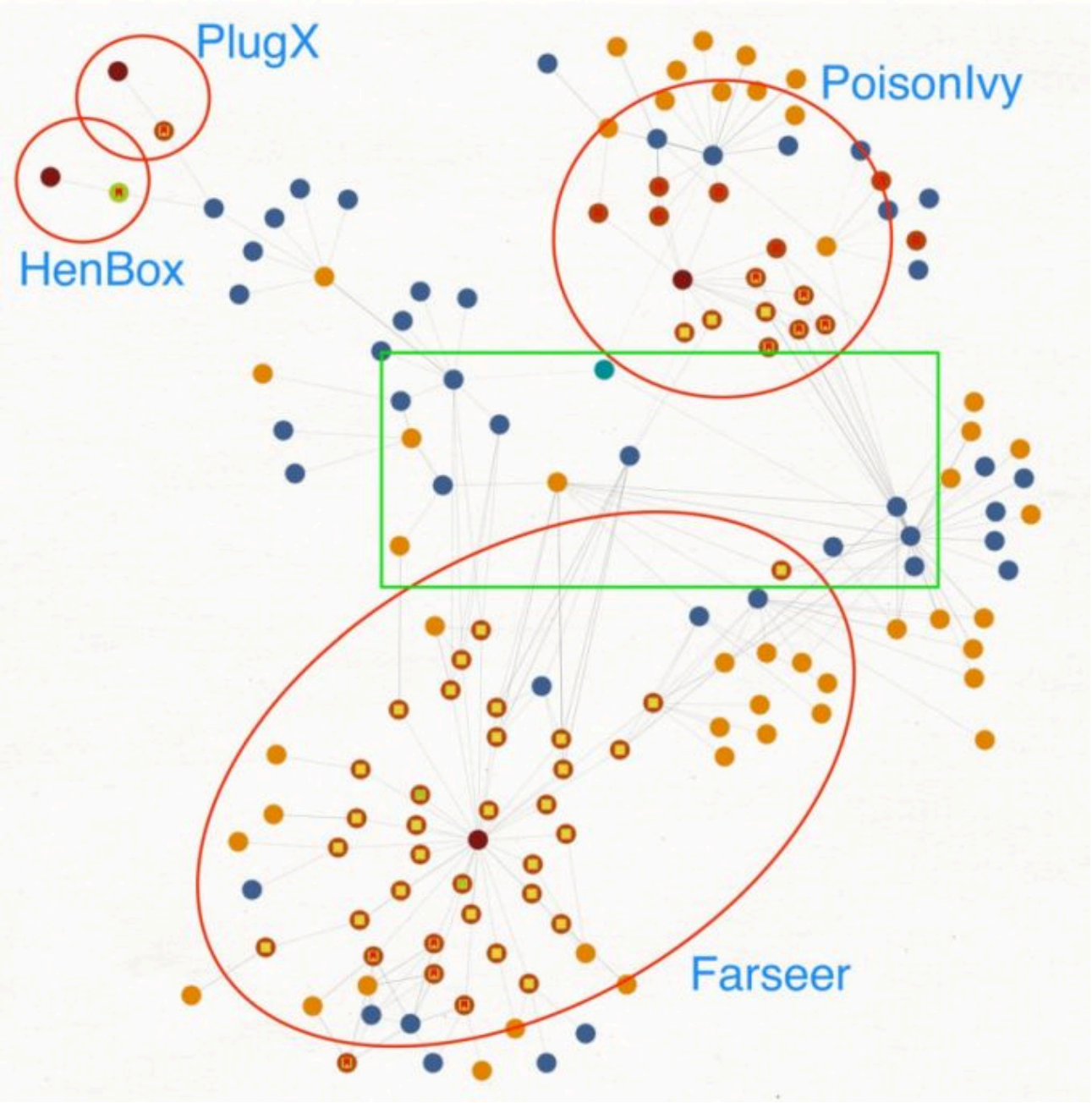

Figure 9: Maltego chart showing overlaps between Farseer and related threats.

Despite the image indicating that Farseer has the largest number of malware samples (red dots), this is not the case when considering the entire set of malware samples and merely appears this way due to the focus of this section of the report.

One of the most recent Farseer samples (SHA256: 271e29fe8e23901184377ab5d0d12b40d485f8c404aef0bdcc4a4148ccbb1a1a) introduced a new C2 domain – tcpdo[.]net – into the Farseer set, as shown in Figure 10.

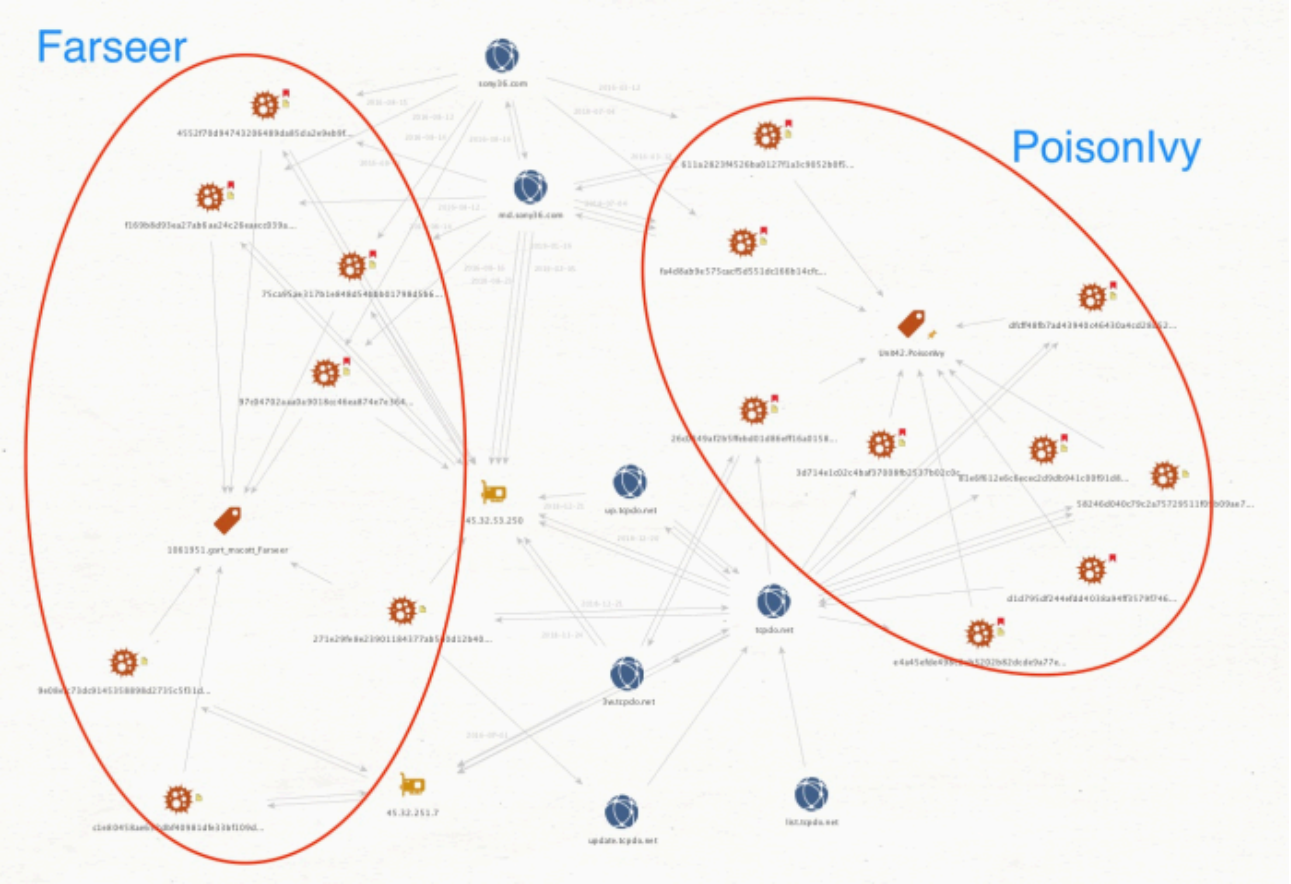

Figure 10: Tcpdo[.]net ties between Farseer and Poison Ivy samples.

This sample communicates directly with tcpdo[.]net for its C2 whereas other Farseer samples communicate indirectly, through third-level domains and IP addresses. A handful of Poison Ivy samples have also used this domain as their C2, most of them prior to this Farseer sample (as early as mid-2015) but also more recently, on 17 December 2018, indicating a fairly active domain.

The overlaps between Farseer and Poison Ivy don't end with tcpdo[.]net. Much as with HenBox, other infrastructure ties exist: directly through sony36[.]com and md.son36[.]com; indirectly through third-level domains of tcpdo[.]net and IP addresses 45.32.251[.]7 and 45.32.53[.]250.

Farseer also overlaps with HenBox and PlugX samples through multiple C2 domains and IP address resolutions:

- outhmail[.]com (and third levels of this domain)
- cdncool[.]com (and third levels of this domain)
- www3.mefound[.]com
- w3.changeip[.]org
- www5.zyns[.]com
- 45.32.53[.]250
- 45.32.44[.]52
- 45.32.45[.]77
- 59.188.196[.]162
- 59.188.196[.]172

C2 server structure

As previously mentioned, a common registrant registered seven known domains related to the malware discussed. Interestingly, all of the domains have at least one third-level domain in common, perhaps indicating a template being used for the infrastructure setup. Table 8 lists the commonalities, aside from other more common sub-domains such as www, mail and dns.

Domain / third-level domain	info.	re.	update.	up.
tcpdo[.]net	•		•	•
adminsysteminfo[.]com	•	•	•	
md5c[.]net				
linkdatax[.]com	•	•	•	
csip6[.]biz	•	•	•	
adminloader[.]com		•	•	
cdncool[.]com	•	•	•	•
newfacebk[.]com			•	

Table 8: Common third-level domain names set up on C2 servers.

Malware execution flow

This section aims to provide a description of the general behaviour of the Farseer malware. Figure 11 describes at a high level the post-installation execution flow of a typical sample.

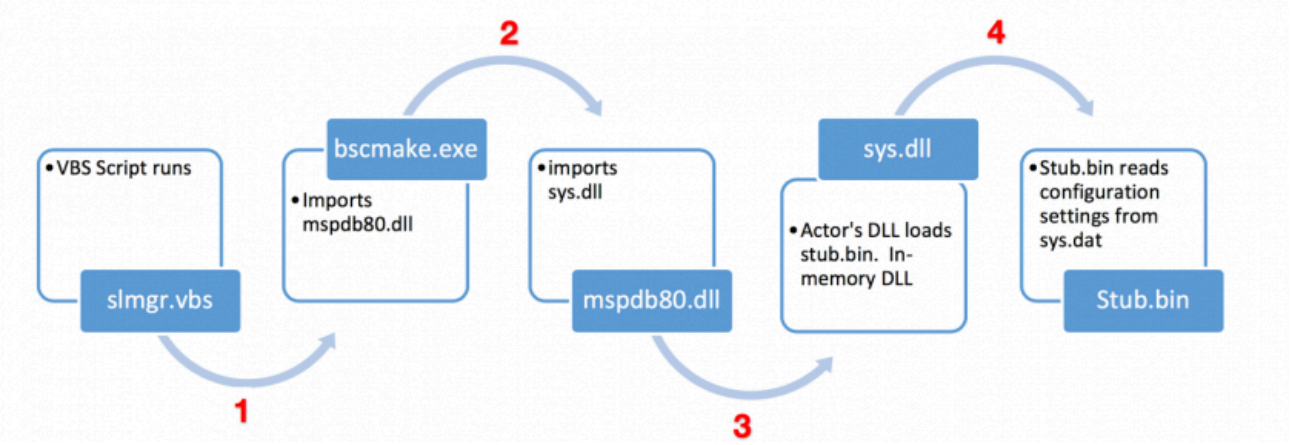

Figure 11: Execution flow of Farseer malware.

For persistence on the host, Farseer creates a registry entry named sys under:

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
```

This runs the VBS script slmgr.vbs that contains:

```
createobject("wscript.shell").run "C:\Users\[username]\AppData\Roaming\windows\bscmake.exe"
```

This is step 1 (in Figure 11), which starts the Farseer execution when a user logs onto their PC.

Steps 2 and 3 involve a DLL-sideloading technique using a signed *Microsoft* file, bscmake.exe, which is part of *VisualStudio*. This executable in turn imports several DLL files, including mspdb80.dll, which in turn imports sys.dll – the malicious payload.

The payload, stub.bin, is encrypted and compressed on disk but is decrypted as it's loaded into memory by sys.dll. Farseer's config file, sys.dat, is also loaded during this fourth step in the flow. Much like the HenBox config file, sys.dat is obfuscated simply using ASCII encoding. Once decoded, the config is structured as per the example in the left column of Table 9.

Farseer config	**HenBox config**
p1=up.outhmail[.]com p2=80 p4=test-04-11 p5=C:\Users\[username]\AppData\Local\Temp\main.exe	a1=wd.w3.ezua[.]com a2=80 a3=crash_report@21cn[.]com a4=smtp.21cn[.]com a5=crash_report a6=lxy.cn@163[.]com a7= a8=0914D1D428914B09A5372866B39524B9 a9= b1=0 b2=0 b3=1 b4=http://www3.mefound[.]com/aa.txt

Table 9: Similarities between the Farseer and HenBox config files.

In the Farseer config file:

- p1 is the C2 FQDN
- p2 is the TCP port used (many variants use non-standard ports)
- p3 is missing
- p4 is a version string sent in the C2, perhaps a campaign identifier of some sort
- p5 is the full file path from which the malware was launched.

At present, we do not know what all the HenBox options refer to.

The two malware config files have some similarities, which strengthens the idea of them being related to a common adversary. Both are text files, read and parsed at runtime; more often than not, the data is encoded simply. Perhaps the most notable similarities in notation are as follows:

- Key value pairs are '=' delimited
- Each line uses a single character followed by a single digit starting at 1
- Both have the C2 host on line 1
- Both have the TCP port on line 2.

Targeting

One of the earliest Farseer samples we analysed also contained a decoy PDF document that was opened during execution. The PDF content included a copy of an article from a Myanmar news website that reports on the Southeast Asia region. The PDF file properties indicate that it was created on a Chinese-language system, and the creation date was eight days prior to the Farseer sample using said PDF.

After publishing information on Farseer, an industry partner told us that their product telemetry showed a Farseer sample running on a *Windows* system located in, or communicating through an ISP in Ulaanbaatar, Mongolia. This additional context, along with the decoy document used, helps to confirm our suspected target countries.

Constructing an adversary playbook

This section introduces the public frameworks and tools underpinning adversary playbooks, and describes the PKPLUG edition.

ATT&CK

MITRE's Adversarial Tactics, Techniques and Common Knowledge (ATT&CK™) is a curated knowledgebase and model for cyber adversary behaviour, reflecting the various phases of an adversary's lifecycle and the platforms they are known to target.

STIX

Structured Threat Information Expression (STIX™) is a language and serialization format used to exchange cyber threat intelligence (CTI).

Specifically, the latest iteration of the STIX format, version 2.0, simplifies the creation of documents and uses JSON, rather than XML. This version also provides a list of objects to represent types of information typically generated for CTI. For instance, STIX includes objects for intrusion sets, malware and indicators, amongst others. The information and attributes stored within STIX objects, and the relationship between the various object types, adhere to standards, which allows this intelligence to be shared and consumed without the need for complex parsing tools.

Attack lifecycle

An adversary must complete a linear, phase-based process to successfully execute an attack. Humans can better comprehend an attack by breaking it down into smaller, phased-based pieces, and work to break the lifecycle at various points in order to prevent successful attacks.

The attack lifecycle is a customized Cyber Kill Chain(™) from *Lockheed Martin*, and is described in Figure 12.

To meld these three frameworks together, we looked at how ATT&CK data mapped to STIX 2.0 and then chose appropriate objects for additional adversary playbook components, as Table 10 describes.

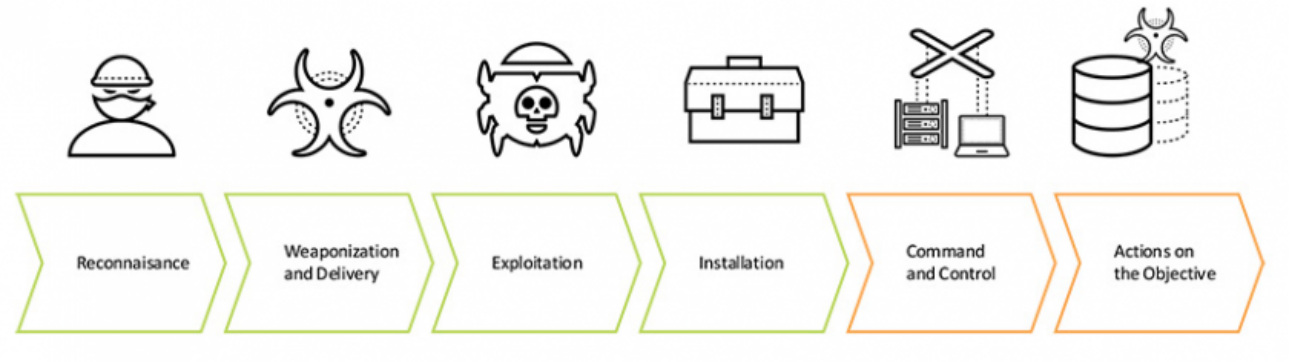

Figure 12: Attack lifecycle.

STIX 2.0 object	**Adversary playbook component**
Intrusion Set	Adversary
Report	Playbook
Report	Play
Campaign	Campaign
Kill-Chain-Phase	ATT&CK Tactic
Attack-Pattern	ATT&CK Techniqu
Indicator	Indicator
Malware	Adversary Malware
Tool	Adversary Tool

Table 10: STIX 2.0 to adversary playbook object mapping.

With these definitions complete, we began mapping the activities of particular adversaries to the ATT&CK framework, and stored the respective data and related IOCs as STIX in JSON format.

Playbook Viewer

As previously mentioned, adversary playbooks are JSON-formatted STIX CTI packages describing threat actors, their campaigns (each one an instance of the attack lifecycle), their behaviours (using ATT&CK) and, finally, the IOCs for each campaign. Consumers can ingest the STIX as they always have done, however, many systems (at the time of launch) did not handle STIX 2.0 content, and certainly none existed that would display an entire adversary playbook for humans to better understand and visualize the information.

Unit 42 released a simple tool to enable the playbook to be viewed through a web interface. A screenshot of the Playbook Viewer [26] is shown in Figure 13; a live version can be accessed at *GitHub*.

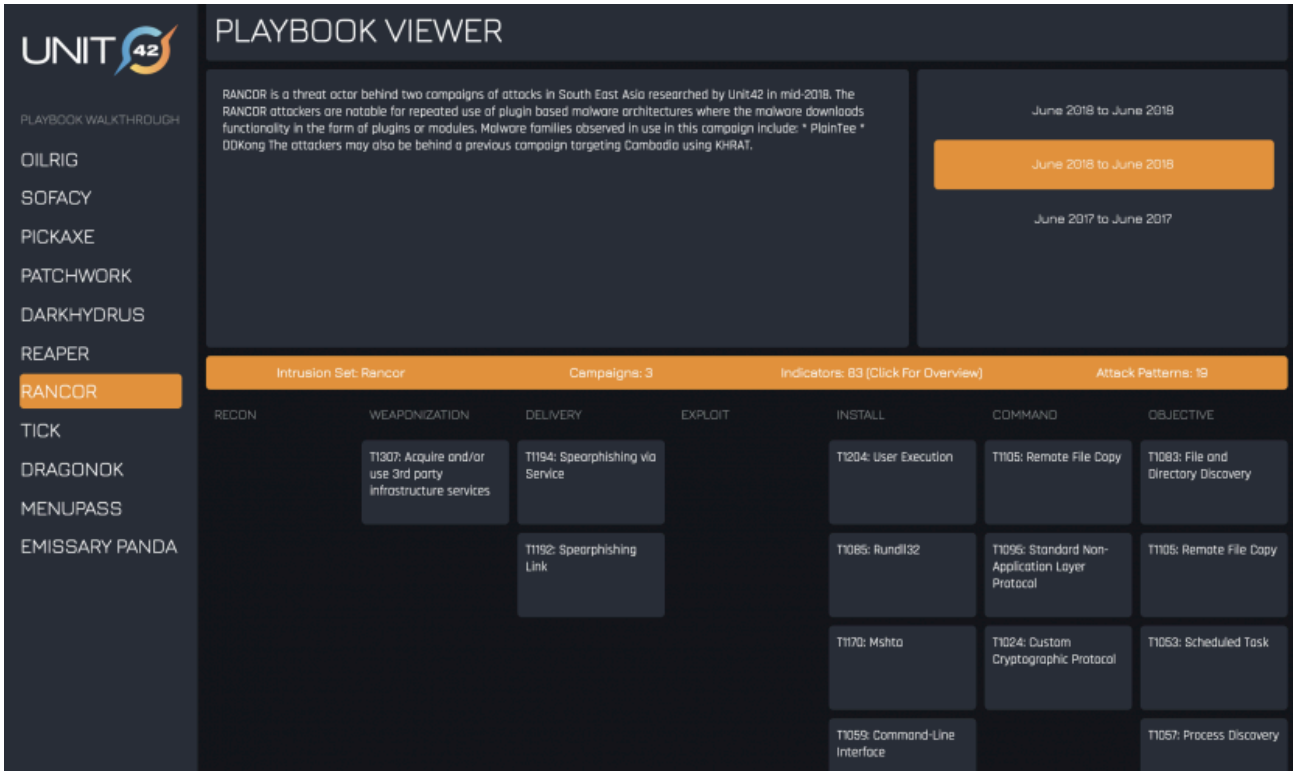

Figure 13: Example Rancor adversary playbook viewed through Playbook Viewer.

Playbook Viewer allows a user to choose an adversary from the list shown. This provides a description and another list of their campaigns (at least the ones we know about, and those we've converted to STIX). Selecting a campaign shows lists of the adversary's TTPs (using ATT&CK notation) laid out in columns as per the attack lifecycle phases. Of course, the level of detail described in each playbook is limited to the visibility one has of a given campaign, thus the sharing, merging and enriching of these is critical to build a more holistic view of a given adversary.

Defence analysis

Beyond ingesting IOCs and visualizing adversary playbooks, another use-case exists around improving defences. Understanding the common TTPs used by malware and adversaries that persistently attack your organization should help to prioritize defence efforts. These don't just have to be deploying security solutions but also designing policies and processes to reduce the risk for the organization and enforcing them, wherever possible, using technology.

PKPLUG adversary playbook

Some of the malware families used by the PKPLUG adversary have been described in detail in this report. It is those (HenBox and Farseer) that are described here, in playbook form.

HenBox

Table 11 describes the single 'play' (a.k.a. campaign) related to the variant of HenBox discovered on the Uyghur app store. Other plays, each an instance of an attack lifecycle, exist for many more samples and may have some TTPs that differ. A full list of plays, together with all the IOCs, is available on the Playbook Viewer.

Phase	TTP	Description / STIX & IOCs

Reconnaissance	T1249: Conduct social engineering	Creation of decoy documents; spoofing legitimate mobile apps; setting up domains with copycat names using relevant and interesting themes.
	T1264: Identify technology usage patterns	Targeting of *Xiaomi* devices, *Android* users and Uyghur app store infers understanding of the victims' MO.
	T1265: Identify supply chains	Using the Uyghur app store to deliver HenBox would first require identification of the delivery mechanism.
	T1295: Analyse social and business relationships, interests, and affiliations	Knowledge of the Uyghur ethnicity and religious beliefs to use in social engineering lures (e.g. Islam-related apps); knowledge, or suspected use, of various social network, secure messaging and communications apps by the victims.
Weaponization	T1307: Acquire and/or use third-party infrastructure services	Use of Uyghur app store to deliver HenBox.
	T1312: Compromise third-party infrastructure to support delivery	How the app store is compromised but an app was overwritten with HenBox.
	T1345: Create custom payloads	HenBox and Farseer are custom malware; others used, such as PlugX, were custom when discovered but are now believed to be used by many groups.
Delivery	T1474: Supply chain compromise (mobile)	Third-party app store APK URL: [url:value = 'uyghurapps[.]net/mobile/downAction.action?appId=40'] Hash of HenBox APK purporting to be *DroidVPN* app on third-party app store:

		[file.hashes.'SHA-256' = '0589bed1e3b3d6234c30061be3be1cc6685d786ab3a892a8d4dae8e2d7ed92f7']
	T1476: Deliver malicious app via other means (mobile)	Delivery of HenBox through compromised third-party app store; other methods assumed including phishing/smishing, file-sharing websites, forums, etc. These are common with *Android* malware delivery.
Exploitation	-	No exploits against vulnerabilities used, to our knowledge; requires user interaction.
Installation	T1027: Obfuscated files or information	Mixture of compression, obfuscation and encryption used for components of HenBox malware, including config files and further payloads.
	T1406: Obfuscated files or information (mobile)	
	T1204: User execution	HenBox requires installation by victim, through social engineering.
	T1402: App auto-start at device boot (Mobile)	HenBox monitors for system event broadcasts and executes accordingly. This includes device reboots, SIM card and network changes, new apps installed, and so on.
	T1418: Application discovery (mobile)	HenBox monitors installed apps to steal information from target apps.
Command & control	T1065: Uncommonly used port	This variant of HenBox used TCP port 888.
	T1071: Standard application layer protocol	HenBox used HTTP to communicate with the C2.

Actions on objectives	T1412: Capture SMS messages (mobile)	Intercepts SMS messages.
	T1413: Access sensitive data in device logs (mobile)	Gathers system and device logs.
	T1416: Android intent hijacking (mobile)	HenBox registered for *Xiaomi* events likely to originate from IoT devices.
	T1418: Application discovery (mobile)	Enumeration of existing and monitoring of newly installed apps.
	T1421: System network connections discovery (mobile)	Enumerates cellular and Wi-Fi networks; monitors for network changes (e.g. switching from one to another).
	T1422: System network configuration discovery (mobile)	HenBox gathers IMEI and similar device and system identifiers.
	T1426: System information discovery (mobile)	Gathers system version information.
	T1429: Microphone or camera	Records information using device sensors.

	recordings (mobile)	
	T1430: Location tracking (mobile)	Tracks device location.
	T1432: Access contact list (mobile)	Gathers information from device contacts database, as well as contacts stored on certain target messaging apps.
	T1433: Access call log (mobile)	Gathers call log information and sets a filter for calls to +86 country code (China) to steal the phone numbers involved.

Table 11: The single 'play' (a.k.a. campaign) related to the variant of HenBox discovered on the Uyghur app store.

Farseer

Table 12 describes the single 'play' (a.k.a. campaign) related to some of the latest variants of Farseer.

Phase	**TTP**	**Description / STIX & IOCs**
Pre-ATT&CK:		
Adversary opsec	T1319 Obfuscate or encrypt code	Mixture of compression, obfuscation and encryption used for components of Farseer malware, including config files and further payloads.
Establish & maintain infrastructure	T1328 Buy domain name	Buying and registering domains for command & control use.
ATT&CK:		
Persistence	T1060 Registry run keys / startup folder	Sets a registry run key to launch.
Defence evasion	T1140 Deobfuscate / decode files or information	Mixture of compression, obfuscation and encryption used for components of Farseer malware, including config files and further payloads.
	T1045 Software packing	
	T1073 DLL side-loading	
Command & control	T1071 Standard application layer protocol	Farseer used HTTP to communicate with the C2

	T1065 Uncommonly used port	Farseer has used TCP ports 158, 993 and others
	T1043 Commonly used port	Farseer has also used TCP port 80

Table 12: The single ‘play’ (a.k.a. campaign) related to some of the latest variants of Farseer.

Conclusions

PKPLUG is a fairly long-standing, active and formidable adversary operating against targets in the Southeast Asia region for what could be various reasons, but clearly interested in information-gathering, tracking and espionage.

Sharing threat intelligence data is very important if others are to learn about targeted cyber attacks and data breaches. Furthermore, sharing not only the IOCs of a given attack but also the TTPs of how the adversary breached and moved throughout the network to fulfil its goals is critical. Sharing TTPs is more difficult, but the use of adversary playbooks – building on solid foundational frameworks – is a great start in providing the necessary structure to do so.

Unit 42 continues to track PKPLUG and the tools used by this adversary; updates to research, IOCs and the PKPLUG adversary playbook will be released periodically.

References

[1] ASEAN Member States. https://asean.org/asean/asean-member-states/.

[2] What Are the Autonomous Regions of China? https://www.sporcle.com/blog/2019/04/what-are-the-autonomous-regions-of-china/.

[3] What is China’s Belt and Road Initiative? https://www.theguardian.com/cities/ng-interactive/2018/jul/30/what-china-belt-road-initiative-silk-road-explainer.

[4] China and Xinjiang: The Fate of BRI. https://thegeopolitics.com/china-and-xinjiang-the-fate-of-bri/.

[5] China’s Crackdown on Uighurs in Xinjiang. https://www.cfr.org/backgrounder/chinas-crackdown-uighurs-xinjiang.

[6] This map shows a trillion-dollar reason why China is oppressing more than a million Muslims. https://www.businessinsider.com/map-explains-china-crackdown-on-uighur-muslims-in-xinjiang-2019-2.

[7] The Battle for the South China Sea. https://edition.cnn.com/interactive/2018/08/asia/south-china-sea/.

[8] China Demands US Cancel Arms Sale to Taiwan. https://www.military.com/daily-news/2019/07/10/china-demands-us-cancel-arms-sale-taiwan.html.

[9] Smartphone Shipments Experience Deeper Decline in Q1 2019 with a Clear Shakeup Among the Market Leaders, According to IDC. https://www.idc.com/getdoc.jsp?containerId=prUS45042319.

[10] Unit 42. https://unit42.paloaltonetworks.com/.

[11] Mitre ATT&CK. https://attack.mitre.org/.

[12] The Cyber Kill Chain(™). https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html.

[13] Structured Threat Information Expression (STIX™). https://oasis-open.github.io/cti-documentation/stix/intro.

[14] HenBox: The Chickens Come Home to Roost. https://unit42.paloaltonetworks.com/unit42-henbox-chickens-come-home-roost/.

[15] HenBox: Inside the Coop. https://unit42.paloaltonetworks.com/unit42-henbox-inside-coop/.

[16] Welcome to the Uighur Web. https://foreignpolicy.com/2014/04/21/welcome-to-the-uighur-web/.

[17] Internet popularity in Xinjiang higher than China's national average. http://www.chinadaily.com.cn/business/tech/2017-07/08/content_30041010.htm.

[18] Hackers Target Uyghur Groups. https://www.rfa.org/english/news/uyghur/hackers-09062012153043.html.

[19] Study Finds Unrelenting Cyber Attacks Against China's Uyghurs. https://securityledger.com/2014/08/study-finds-unrelenting-cyber-attacks-against-chinas-uyghurs/.

[20] Cyber Attacks Against Uyghur Mac OS X Users Intensify. https://securelist.com/cyber-attacks-against-uyghur-mac-os-x-users-intensify/64259/.

[21] Scarlet Mimic: Years-Long Espionage Campaign Targets Minority Activists. https://unit42.paloaltonetworks.com/scarlet-mimic-years-long-espionage-targets-minority-activists/.

[22] Turkistan Islamic Party. https://en.wikipedia.org/wiki/Turkistan_Islamic_Party.

[23] Attack Delivers '9002' Trojan Through Google Drive. https://unit42.paloaltonetworks.com/unit-42-attack-delivers-9002-trojan-through-google-drive/.

[24] New Poison Ivy Activity Targeting Myanmar, Asian Countries. https://web.archive.org/web/20160618095613/https://www.arbornetworks.com/blog/asert/recent-poison-iv/.

[25] Farseer: Previously Unknown Malware Family bolsters the Chinese armoury. https://unit42.paloaltonetworks.com/farseer-previously-unknown-malware-family-bolsters-the-chinese-armoury/.

[26] Unit 42 Adversary Playbook Viewer. https://pan-unit42.github.io/playbook_viewer/.
