DECODED
avast.io

Mobile   Network   PC   IoT   Careers

Type here to search...

Hitching a ride with Mustang Panda

by **Threat Intelligence Team** — December 2, 2022 — 31 min read

Avast discovered a distribution point where a malware toolset is hosted, but also serves as temporary storage for the gigabytes of data being exfiltrated on a daily basis, including documents, recordings, and webmail dumps including scans of passports from Asian, American and European citizens and diplomats applying for Burmese visas, from Burmese human rights activists and Burmese government institutions.

We recently came across a peculiar sample – a stager we believe is being used by Mustang Panda. The stager led us to the group's distribution point, where we found malicious toolsets. We have analyzed the malware and were able to see relations between various campaigns that have been described by other cybersecurity firms over the course of the last years. Based on publicly published research and our own investigations, we can say with high confidence that the modus operandi and the malicious toolset show a strong link to a group related to Mustang Panda, which has previously been reported as a Chinese APT group. The group has been known for gathering intelligence on Mongolia, and further Asian countries, and most recently was suspected of targeting European entities.

The distribution point, an FTP server, is also used as a transition point for exfiltrated victim data, before the data is moved to an unknown location. We continue to observe new data being uploaded and moved from the point, meaning the campaign is still active, and has been going on for some time. Gigabytes of data are moved around, and the amount of data indicates severe compromise of many high-profile targets in Myanmar. The data types include various office documents and PDFs, stolen browser profiles, webmail dumps and even sound recordings. Browsing profiles were also extracted which can provide access to other infrastructures, services, and private data of the victims. Most of the stolen data seems to be in Burmese making it challenging to analyze. The sensitive data is mainly being collected from devices used by the Myanmar government, state administration, police, army, significant public organizations, or companies, and includes data related to diplomatic meetings, court hearings, military information, contracts and more.

Disclaimer: We have only seen partial snapshots of the exfiltrated data as they are deleted shortly after being moved from the distribution point, so it should be noted that the information we have on victims may be inaccurate or incomplete. Most of the documents are in Burmese, therefore, a language barrier has also to be taken into account. Finally, due to the limited scope and the sheer volume of the data, some assumptions had to be made during the research process. We have reached out to local CERTs, informing them of our findings.

Victimology

Nearly all of the victims have close ties to Myanmar and it seems that both the Burmese government and opposition groups are being targeted. We have seen data originating from various departments of several Burmese ministries. Even the Office of the State Administrative Council has been targeted. The government breach is not isolated to Myanmar – we have also seen Myanmar embassies among targets, for example, the embassy in Serbia. The data also contained dumps of a mailbox used to communicate in 2016 and 2017, as well as in 2020 with visa applicants from all around the world. These messages contained scans of passports from citizens and diplomats from various countries, such as China, Australia, Czech Republic, France, Israel, Netherlands, UK, and USA.

After such an extensive list of targets, it ought to be little surprise that Myanmar Police Force is also among the targets. Even some higher profile departments, such as the Office of the Information Police Chief or the Department of Special Investigation, seem to have been breached. Tatmadaw (Myanmar Armed Forces) is also not an exception – we have seen victims from the Bureau of Air Defense, Myanmar Army Engineering, and the United Wa State Army.

Political NGOs and the government's opposition are also on the list of victims. It is possible that the list is even more extensive as we may not be able to find a straightforward association to said organizations as we would expect more common usage of personal computers or computers that are not centrally maintained by an IT department. We have seen data from devices belonging to the Karen National Union, Center for Diversity and National Harmony, National Reconciliation and Peace Centre, Ethnic Nationalities Affairs Center, and even the Union Civil Service Board.

Exfiltrated Data

The most common file types being exfiltrated by the group are Microsoft Office documents (.docx, .xlsx, .pptx, etc.), PDF documents, and plain text files. Other file types exfiltrated include audiovisual data in various forms, including sound recordings (.mp3), and pictures (.jpg, .png, etc.) or drawings. Emails, including entire conversations are also exfiltrated.

It appears that the attackers are also looking for and collecting data from browser profiles from various web browsers, e.g. Chrome, Firefox, Opera, and more, a serious threat to victims' privacy. The stolen browser profiles can provide access to other infrastructures, services, and the victims' private data of the victims. The attackers are extracting information about browsing history, stored credentials (personal and work), credit cards, used tokens, and valid cookie sessions. Consequently, poorly secured services, such as services without two-factor authentication or without a safe cookie policy, can be easily abused by attackers. Attackers can steal the identity of victims and can use their email, Facebook, Telegram, or other accounts to collect additional information about the victim and their family, friends, and activities.

Highly sensitive data is being collected from victims' computers, and, in most cases, these are computers used by the Myanmar government, state administration, police, army, significant public organizations, or companies. This in some cases included sensitive data and information belonging to international citizens and diplomats who have interacted with targeted departments.

The documents, and audiovisual data being exfiltrated by the group is massive. The files include everything from:

• Email dumps including visa applications and scans of passports belonging to citizens and diplomats from various countries, such as China, Australia, Czech Republic, France, Israel, Netherlands, UK, and USA
• A seating plan for the meeting between former US Ambassador to the United Nations Bill Richardson and Myanmar's leader, Senior Gen. Min Aung Hlaing
• Myanmar's constitution with proposed changes
• Invitations for diplomatic meetings, meeting programs, calls, and talking points
• Reports, maps, and screenshots from the Signal messaging app related to the UWSA (United Wa State Army)

- Data from the Office of the Chief of Myanmar Air Defense Force, including meeting minutes, full staff/rank lists, photo IDs (some with fingerprints), salaries, personal details of employees' families
- Peace treaty documents
- Interrogation reports
- Contracts
- Court hearings
- Town plans
- Contact information for police officers, including their names, addresses, telephone numbers, and salaries
- Transcripts of meetings around politics, and elections
- Meeting minutes and audio recordings of meetings between Myanmar senior officials (Prime minister, Chairman of State Administrative Council) and the President of Tatarstan
- Military buildings drawings, including munitions storage, oil storage and aerial photos of proposed sites
- International banking records and records and transfers from supporters to a refugee group

Ties to known campaigns

Since getting our hands on the distribution point, we have established links between known campaigns already publicly reported and what we have discovered. This gives us clues as to how resourceful the group may be and will also help us assess its *modus operandi*.

We have found files strongly resembling (or even matching) samples and their relations described in a [blogpost by ESET](#) around the Korplug variant dubbed Hodur. The campaign they described was targeting various government organizations in Mongolia, Vietnam, and Myanmar, along with politically-oriented NGOs. This is in alignment with the victimology of the stolen data we have seen on the distribution point. Hodur was attributed to the Mustang Panda group. The related part of the uncovered toolset we analyzed also contained a USB launcher written in Delphi, similar to the one seen accompanying the Hodur variant of Korplug analyzed by ESET. This installer is responsible for firing up the infection chain leading to a variant of Korplug RAT.

Similarly, we've found similarities to operations attributed to LuminousMoth both in structure and purpose. For instance, we have seen a very similar structure as the one described in [Bitdefender's research](#) on the LuminousMoth group. Namely, the usage of the same binaries for sideloading, same pattern for exfiltration – using RAR for collection and a sideloaded library for exfiltration via Google Drive. Perhaps the most common pattern was the usage of a USB launcher written in Delphi that was attributed to Mustang Panda, which was also described in Bitdefender's research.

In some cases, we have seen some unreliable links to older campaigns such as [Operation NightScout](#), a rather old [KMPlayer supply-chain attack](#), or [Operation Harvest](#). Namely, binaries used for sideloading or names of encrypted payloads matched the ones used in these old campaigns. Nevertheless, the specific payloads differ significantly, so while some of these were attributed to Mustang Panda, the similarity could also be coincidental.

Toolset overview

The storage we have discovered contains many archives with various tools to be downloaded by infected victims. We will use names of these archives to impose basic structure on the data we have found. It is worth noting that these names are partially consistent in successive versions. For instance, we have found an archive *KKL* which was later on accompanied by another version with a slightly different configuration called *KKL1*.

Some archives contained complete toolsets, whereas others only had single purpose tools in them that were meant to be used in connection with other tools; for instance, one contained a keylogger that obviously lacked any exfiltration functionality. This provides a strong indication that the tools are intended to be used modularly. We will build upon that and first talk about the usual Mustang Panda theme – Korplug. Then we'll get to the more specific tools and end with the single purpose tools. Notably, nearly all the tools, aside from Korplug, its loaders,

and Delphi installers, haven't been described before. The RAT written in Go (*JSX*) or the modular backdoor (*US_2*) deserve an extra mention due to their complexity

From the data we have seen, we conjecture that the main exfiltration tools are variants of tools contained in archives named *GDU*, which use Google Drive for the exfiltration. Since we haven't seen any exfiltration tool that uses the distribution point directly and the path on the server of exfiltrated files contains *gd*, we presume that the responsible group uses some other tools to move files from various Google Drives to the distribution point we saw.

A brief look at the toolsets brings up another interesting fact: almost all the files show approximately (up to a few cases within seconds) a seven or eight hour offset between the compilation timestamp and the "last modified" timestamp of the file itself. Since the compilation timestamp is usually in UTC and the archives use the local time for the contents' last modification date, this places us at UTC-8 and UTC-7. Therefore, we presume that the build setup operated in a time resembling Pacific Standard Time (PST) and Pacific Daylight Time (PDT), used on the West Coast of the United States. There are a few caveats – *SE3* and *SE4* contain files that were compiled on November 1, 2021 and still have an eight hour offset even though none of the countries using PST/PDT transitions to PST that early (both USA and Canada transition to PST a few days later).

There is also a file with an obviously spoofed compilation timestamp. *HT3* contains a DLL *Vender.dll* whose compilation timestamp dates more than a month after the last modification date. This further weakens hypotheses that build upon timestamp offsets. Unfortunately, we have no further leads explaining this outlier. The latest version of the uploader (multiUpload.exe), whose usage was spotted at the beginning of June, has a compilation timestamp of January 3, 2020. This is also very likely spoofed as analyses of the previous versions of this tool show clear evolution and, according to their respective timestamps, they were all compiled in April 2021. Not to mention that the corresponding infrastructure was only created at the end of May 2022.

The folder */pub/god*, which contained the toolset archives, was removed on June 25, 2022. On the same day, a new folder */pub/god1* was created with two files to which we didn't have read access. Two days later, the new folder was gone and */pub/god* appeared again with a subset of the original tools.

Variations on Korplug

The first group of tools that we'll introduce are various versions of Korplug. The binaries used for side-loading were already seen before. Even though the loaders were mostly new, they were rather uninteresting. A common theme was a Delphi binary that served as a launcher to be executed from an infected USB drive. As we've already mentioned, a similar installer was previously seen in campaigns attributed to Mustang Panda. It just executes the Korplug loader from a folder named "*Kaspersky*" that is on the very same USB drive. See the diagram below (based on a toolset from an archive named *BMD*) for more details.

Note the usage of the folder name *"Kaspersky"* and usage of *"Symantec"* in the names of the executables; since the launcher relies on social engineering tricks, it depends on a common strategy using seemingly legitimate file names to dispel any doubts concerning the content.
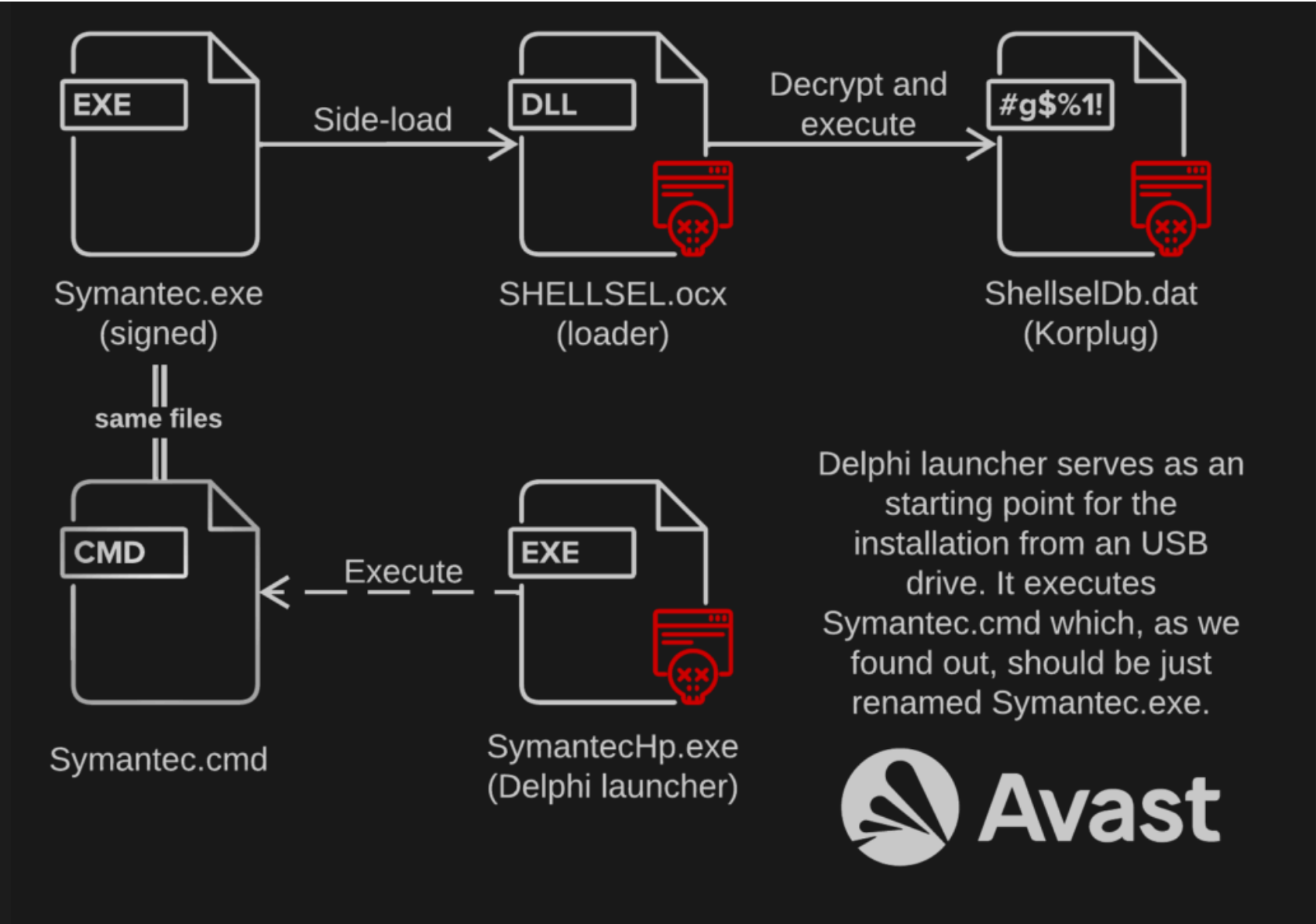
Contents of an archive called BMD. Archive YK41 follows the same structure, with ShellselDb.dat being replaced with hp_ui.xslbcdsj and without the Delphi launcher.

There were also simpler infection chains, containing just a signed clean binary, a loader to be side-loaded, and an encrypted Korplug. These were contained in archives *WD*, *127C*, and *1260M*. The latter interestingly used the *OleView.exe* binary which then side-loaded the *ACLUI.dll* that decrypted and executed *ACLUI.DLL.UI*. The same signed binary, which surprisingly also has the same name of the encrypted payload, was used in the KMPlayer supply-chain attack in 2013. The accompanying research was only published in Chinese, likely due to the attack being limited to a few devices.

Exfiltration toolset

The exfiltrated data on the server in */pub/gd* folder showed perfect correlation to the data produced by *GDU* toolsets (*GDU_OLD*, *GDU*, *GDU1*, *GDU2*, *GDU1_NEW*, *GDU3,* GDUPlZ). These tools collect the files on the victim's disk, pack them into an archive whose name is prefixed with the victim's ID and upload that archive onto a Google Drive. We presume that the name *GDU* is an acronym for *Google Drive Uploader*. While the tools themselves were technically rather simple, the exfiltration process and their evolution piqued our interest.

The analysis of the exfiltration process brought up several interesting observations. A few days after May 24, 2022, the day we started systematically monitoring Google Drives used for the exfiltration, we started to see more frequent token changes and new features being implemented. These features mitigated possible downtime caused by the migration to a new token. Since Google Drive has extensive logging functionality and the tokens have to be present on the infected devices, it is only a reasonable expectation that access to these drives is monitored to some extent.

On the contrary, we have not seen such behavior with the distribution point. This could be attributed to the fact that the distribution point is never exposed by the toolsets, which brings us to the assessment that the group presumes the distribution point to be secret or not worth monitoring.
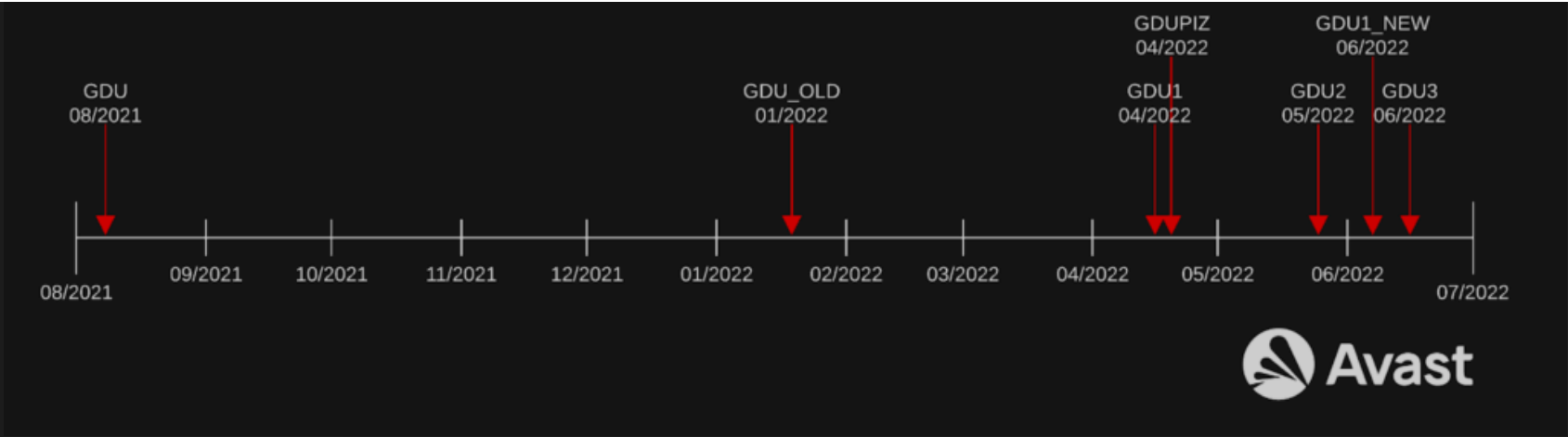
Timeline of GDU exfiltration toolsets

Version overview

The oldest version *GDU* uses RAR executable to collect the data and an encrypted *rar.dat* to store parameters for the RAR binary. Starting from *GDU_OLD*, they migrated to their own collector *piz.exe* (this functionality was later moved to a DLL to be side-loaded) and retained an encrypted *rar.dat* where they stored their configuration. These toolsets also rely on two optional configuration files that are backed up by hard-coded values: *token.dat* containing an encrypted token for Google Drive and *time.ini* that contains last execution date along with the victim's ID. The setup and the choice of binaries for side-loading bears resemblance to the LuminousMoth campaigns and tooling.
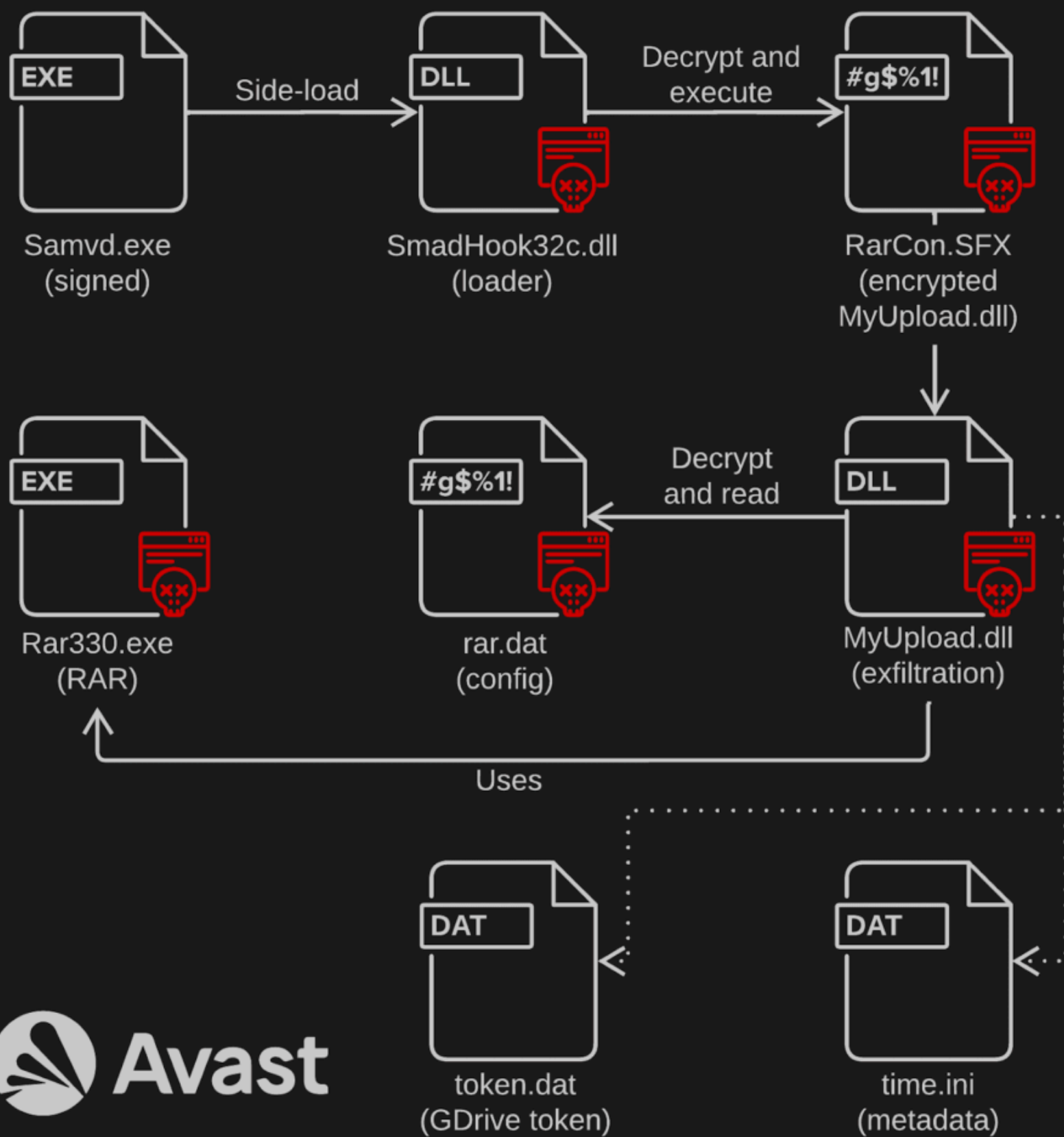
An archive GDU that contains a version relying on RAR instead of piz.exe for data collection.

The exfiltration process is usually handled by a variant of *MyUpload.dll* that is supplemented by the aforementioned configuration files. Quite recently, a new version of GDU1 has appeared on the distribution point (which we'll call *GDU1_NEW*) which came with its successor *multiUpload.exe. multiupload.exe* eschews hard-coded tokens and makes the exfiltration process more resilient to disruptions.
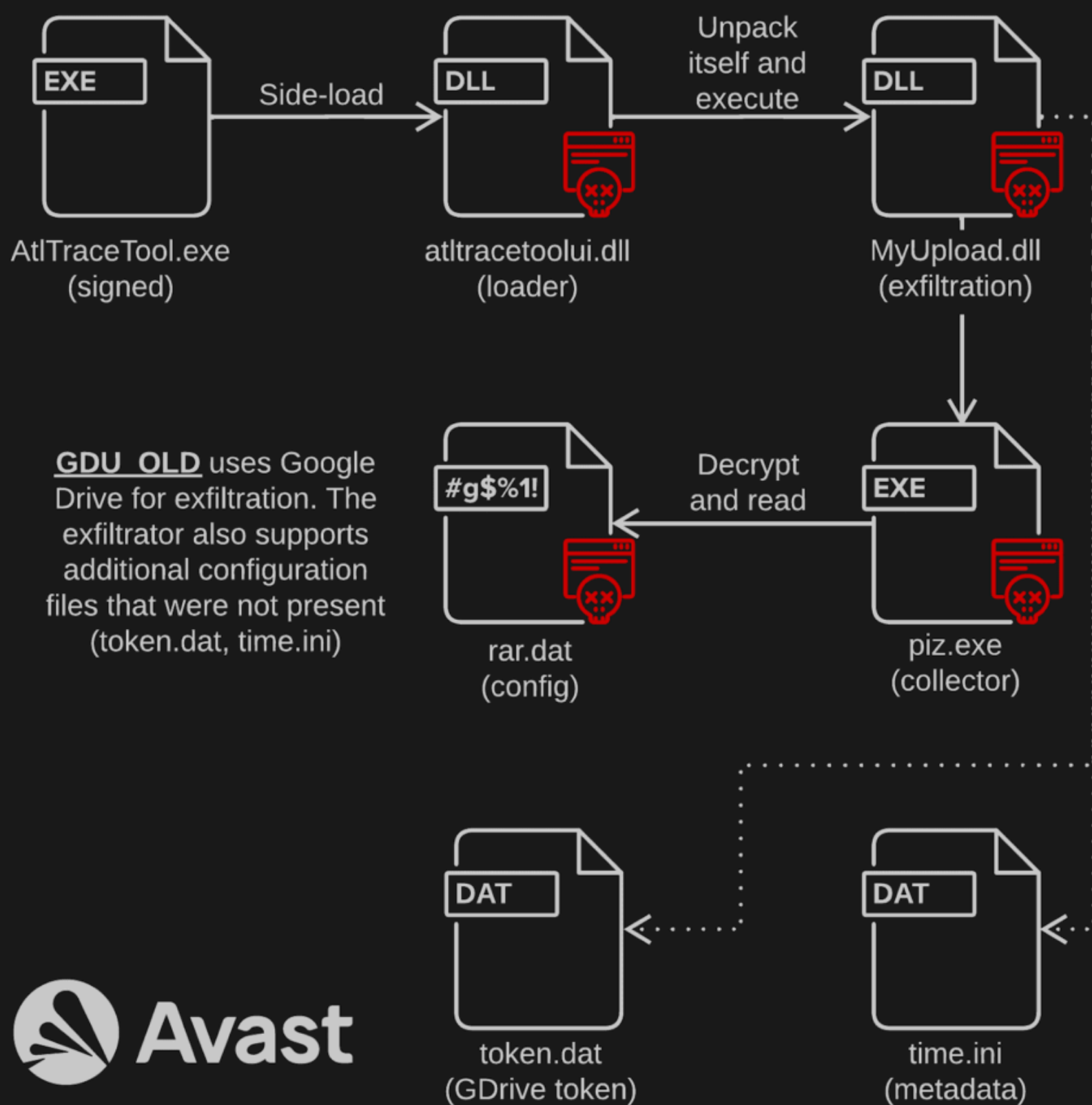
Archive GDU_OLD that uses its own collector piz.exe to collect potentially interesting files before the exfiltration.

The version we are calling *GDU_OLD* is basically the same as *GDU1* and *GDUPIZ*, the most significant difference being a different PE being used for side-loading. *GDU1* and *GDUPIZ* rely on CefSub.exe and subsequently on *CefBrowser.dll* instead of *AtlTracetool.exe*. *GDUPIZ* also uses a slightly different approach to execute the file collection tool *piz.exe* – the version included in this folder is in fact a renamed clean binary spoololk.exe which in turn side-loads *vntfxf32.dll*. This malicious binary implements the file-collecting functionality formerly contained in *piz.exe*. *GDU2* is basically the same as *GDUPIZ*.

On June 8, 2022, we saw a new GDU1 toolset on the distribution point. This time *MyUpload.dll* has been upgraded to provide redundancy in the exfiltration process. It no longer uses *token.dat* but rather uses a Github repository as its source for the token. If this fails, there are two backups – one using HTTP PUT to *www.watercaltropinfo[.]com* with Basic HTTP Authorization (*123:123*). The other sends the data via HTTPS POST to *m.watercaltropinfo[.]com*. The collector is the same as in *GDUPIZ*. *GDU3* uses basically the same process but uses different PE for side-loading (*FwcMgmt.exe*).

Tokens

A special chapter is devoted to Google Account tokens that are used in these tools, partially because our research may have forced the group's hand to refresh the tokens once they discovered that we knew of their

Google Drives. The fact that after each token decommission, every client had to have the token updated and that *GDU* toolsets do not have any remote update functionality suggests that these toolsets have to be accompanied by other tools that provide this update functionality. We have noticed a longer delay between the decommissioning of the token from May 29, 2022 and before a replacement token was being distributed. Its distribution coincides with the time a new version of *GDU1_NEW* was released. It's exactly this version that has introduced new functionality in the exfiltration tool, namely smoother token swapping and failsaves for cases when Google Drive exfiltration fails. Therefore, we presume that this delay was caused by the development of this new functionality.

| Source	Refresh Token	Metad
GDU/*GDU_OLD*	1//030YFi1XWWVY2CgYIARAAGAMSNwF-L9Ir_I6Aqhg4gn3UWyi3I5J_6q4VxHPYNkit3x9RtlpVkFCEOD4KU-E2u7NghS0hx5nPU8Y	First se 22.4.20 accessi
Distribution point	1//0clbh_CVKereiCgYIARAAGAwSNwF-L9IrtPxC9zt-onaeeq2gTUwZTVa8f8ILvDYM8dyVVjmmKmNr5Pnsi27iMjLll2dxbDRxUQs	First se 22.4.20 accessi
GDU1/*GDU2*/*GDUPIZ*	1//0cZLNy7GH0LknCgYIARAAGAwSNwF-L9IrV7aXKTZb-WQM1aqXkjX0ph25dTZngOYAUlAXHB1NvCorgw62XI_TXsWNrK98KBYfbpA	First se 17.3.20 first acc 24.5.20 until: 26
Distribution point	1//0e29aixufumhCCgYIARAAGA4SNwF-L9IrX2w0GnjBQe4g5hQWKJlXjIQF7XZBnQ9VKArYhJVwOayZq9Ad-G8YHbsptTN5DhfXI2E	First se 29.5.20 first acc 29.5.20 until: 3.
Github	1//0ewRnXWCf2AunCgYIARAAGA4SNwF-L9IradANDPAvIqPOaIfReqT1fQ0GO5A9FnUhdpdI0Q0V1IRye5RbTi-hcGQbbHKgDc02xfM	First se 8.6.202 access: 8.6.202 usable

Temporal analysis of exfiltrated data

We can also have a look at the metadata of the exfiltrated archives. What is rather unsurprising are the upload times which closely coincide with Burmese business hours – a smaller peak in the morning and a huge peak in the afternoon. Note that Myanmar is in UTC+6:30 time zone and China, presumed land of origin of Mustang Panda, is in UTC+8.

GDrive upload times in MM time (UTC+6:30)
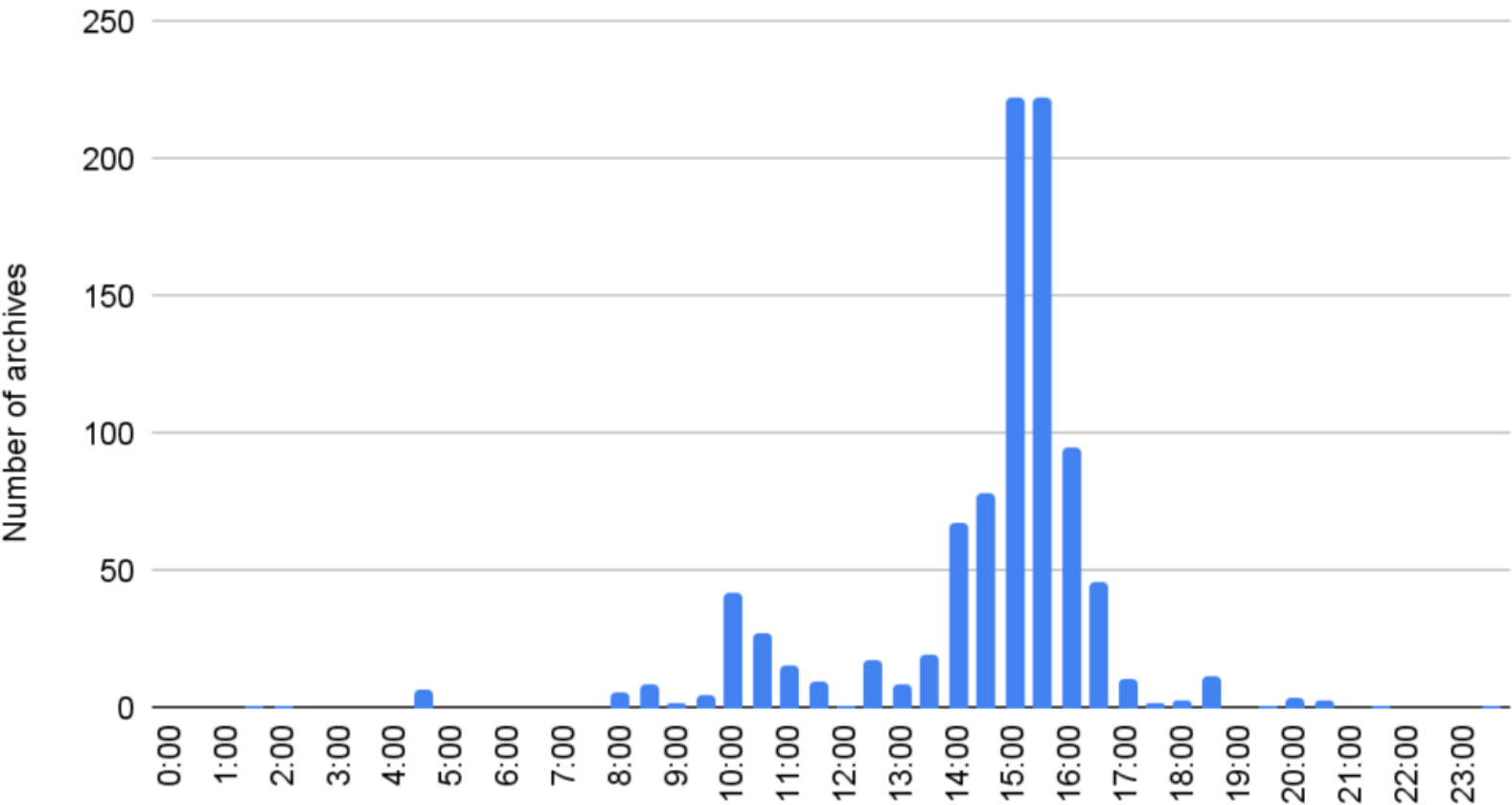
What is more interesting are events produced by the group itself – transfers from Google Drive onto the distribution point and deletion of files from the distribution point. The huge peak is around 18:00 MM time which coincides with the end of the work day in Myanmar. The spread of starts of upload windows is negligible, leading us to the presumption that the transfer is automated. We have seen a few archives being placed in the wrong directories which could indicate that the tooling is still under development or there's still some manual work involved… Usually, the files have accumulated during the day on Google Drive and were transferred to the distribution point in the evening MM time.

Stolen data transfer times in MM time (UTC+6:30)
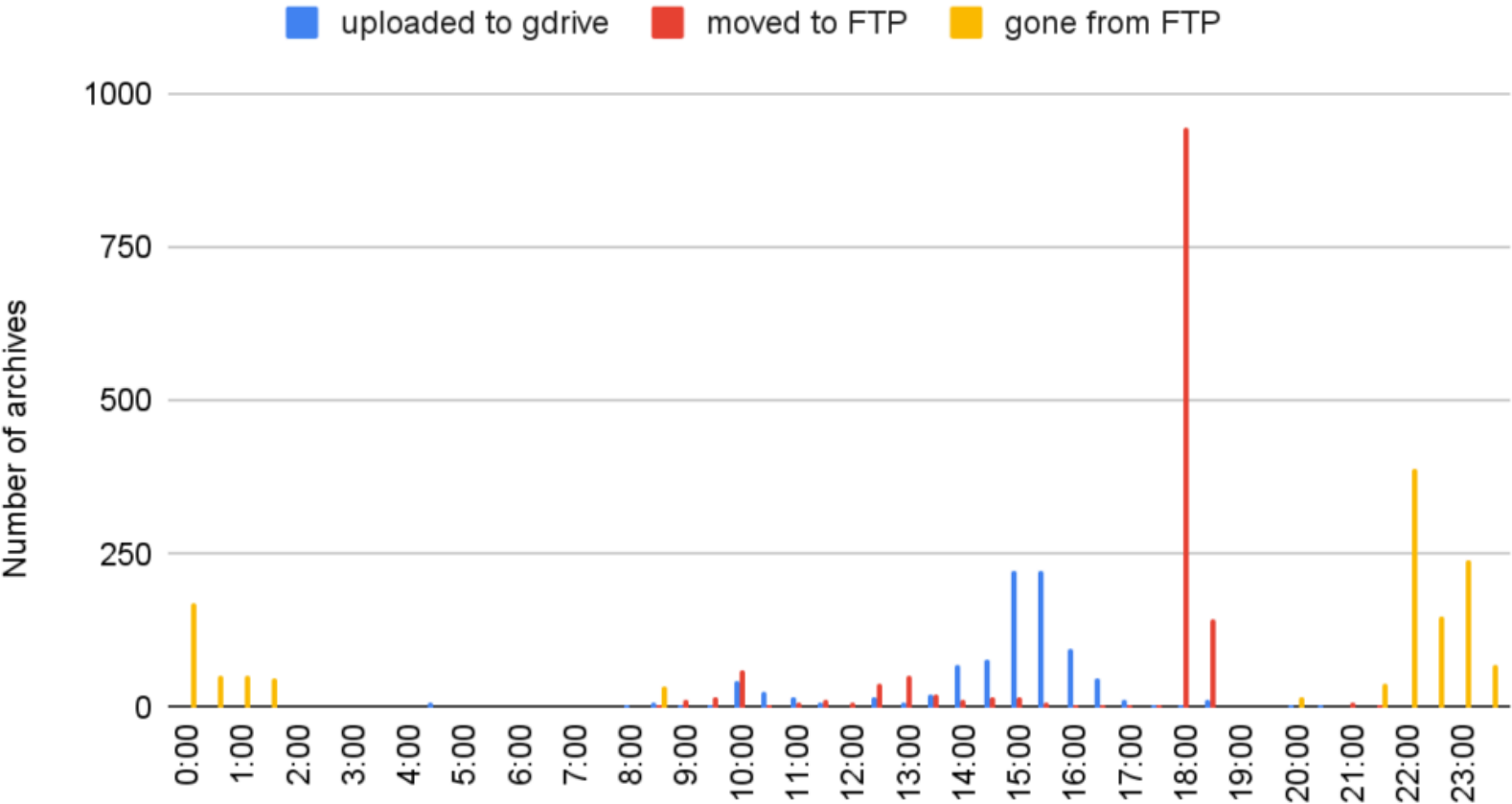
As we already mentioned in the introduction of uncovered tools, we have found quite a lot of files that were "missing" something. By that we mean that on their own, they were either lacking communication functionality or implemented some techniques that were useless without being accompanied by another payload. Interestingly, while these were also using side-loading, they were not relying on external encrypted files, making their execution flow straightforward with 2-point graphs. For the sake of brevity, we will list these in a table:

Archive	Clean executable	SideLoaded DLL	Purpose (high-level)
MG/MG44	dabs.ex/44.ex	SensorAware.dll	fingerprinting/remote shell
AUD	mcsync.ex	mcaltlib.dll	audio recording
CHR	browser.ex	browser_elf.dll	Cookie dumping
T3YK	ygfdt.exe	corecrl.dll	remote shell
UC	melt_64.exe	libmlt-6.dll	UAC bypass
KKL/KKL1	mscorsvw.exe	mscorsvc.dll (contains KBE.dll)	clipboard stealer, keylogger
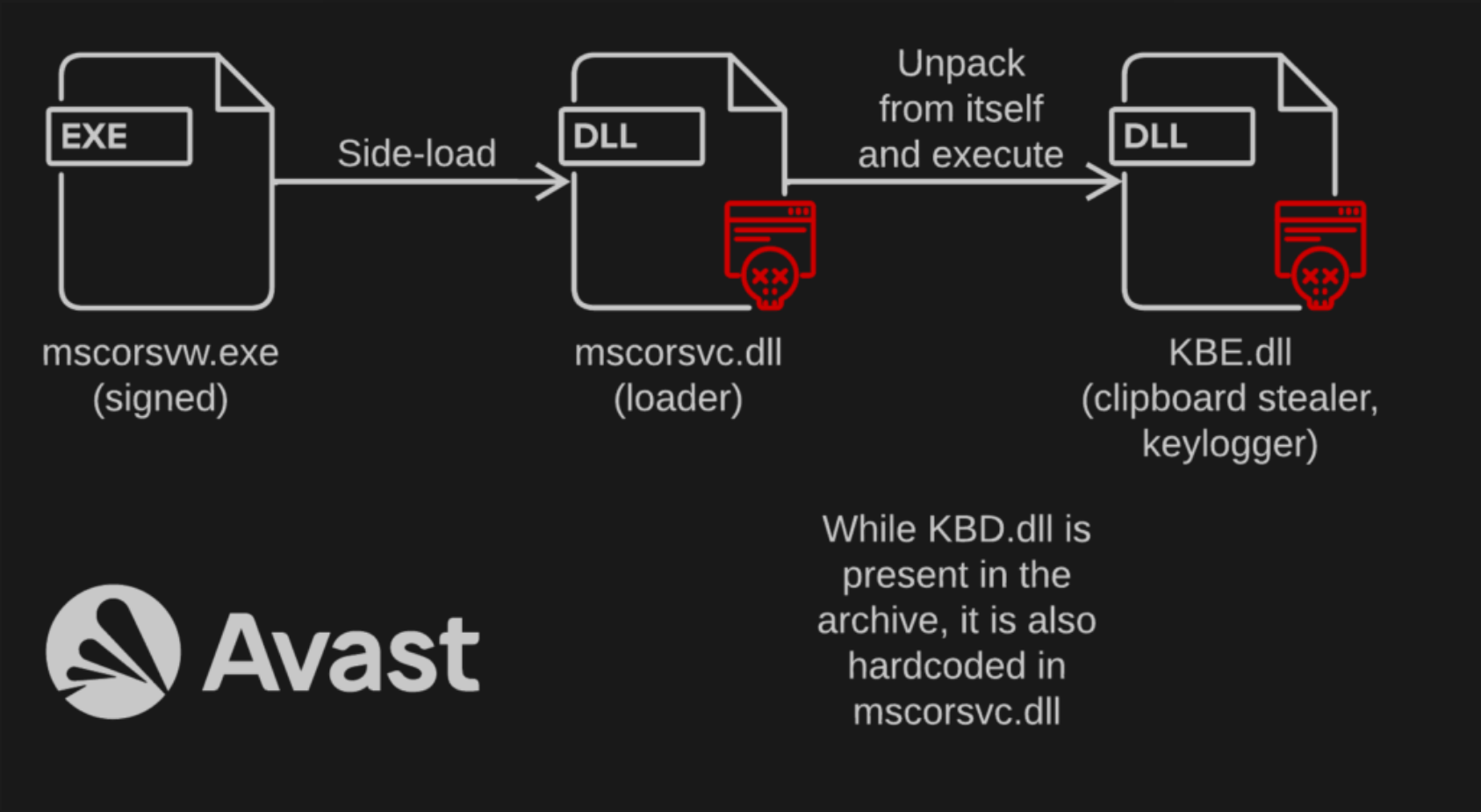
Archive KKL contained a straightforward stealer in KBE.dll that was also hard-coded (in an encrypted form) in mscorsvc.dll from which it was unpacked after side-loading. The newer KKL1 was practically the same.

There are also a few standalone files such as *x.ex* in archive *X* which composes logs of registry entries (Run), services, and scheduled tasks along with checking all signatures of executable files in *%WINDIR%*. Archive *NB* contains *nb.dat* which is just the *nbstat* utility executable. There was also a coinminer in the archive *INFO* along with its encrypted configuration file.

An interesting utility is DISK2 (and its variant DISKM); it is responsible for monitoring the system drive for any changes. Both are accompanied by a configuration file that defines which files are of interest and where such files should be copied to. It also notifies its C&C server about these files via an encrypted message over HTTP. There is also a version of this tool named *MF20211228* that does not contain any configuration file and does not copy files anywhere, it just sends messages to the C&C server.

Contents of an archive DISK2. This toolset is responsible for monitoring the system drive for any changes in files defined by its configuration.
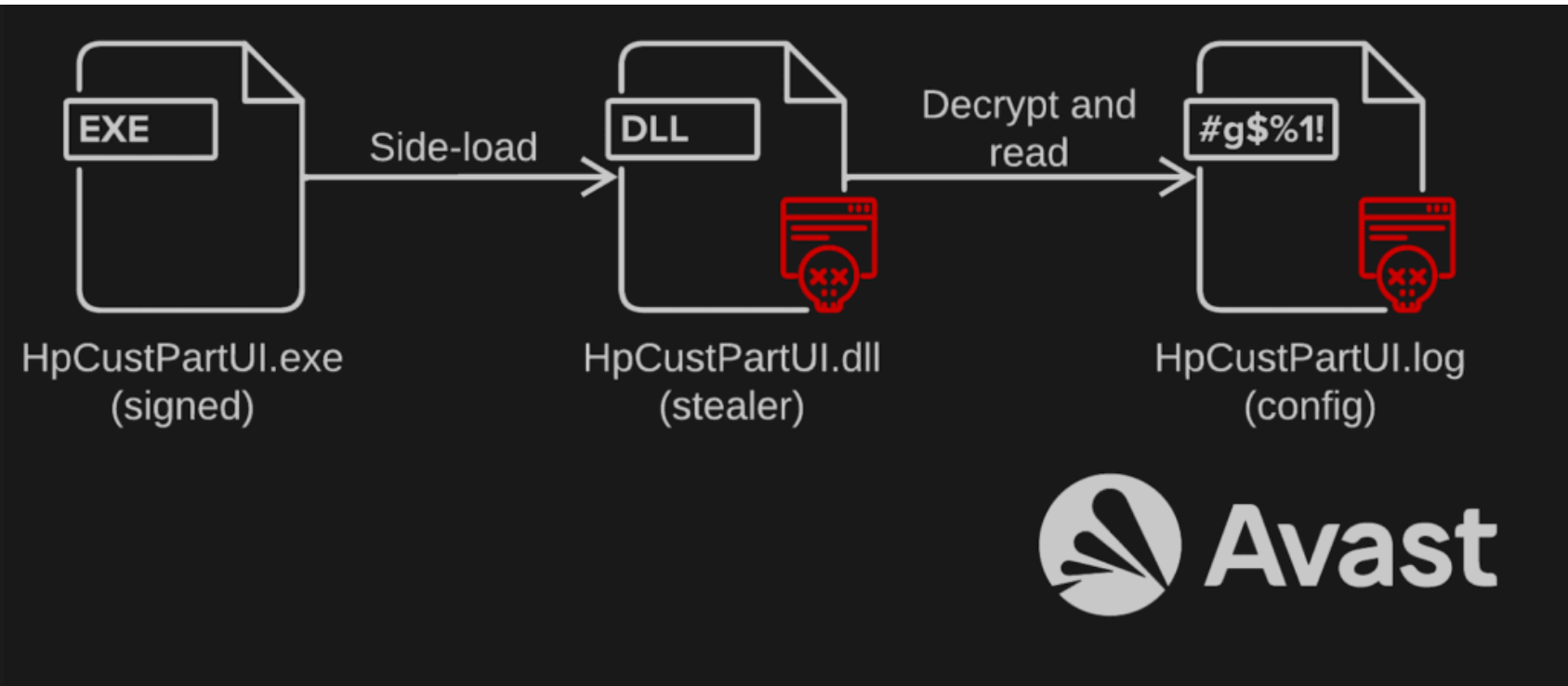
Oddballs in the collection

JSX

JSX archives (*JSX86* and newer *JSX861* for 32 bit version and *JSX64* and newer *JSX641* for 64 bit version) deserve a separate mention as they use a rather uncommon setup; a JavaScript file is at the beginning of the chain and instead of side-loading, the respective DLL is launched as a service. *mozload.dll* is a RAT written in Go that uses HTTPS and websockets for its communication. Interestingly, the RAT uses TLS Client Authentication; see Appendix A.1 and Appendix A.2 for the private key and the certificate.
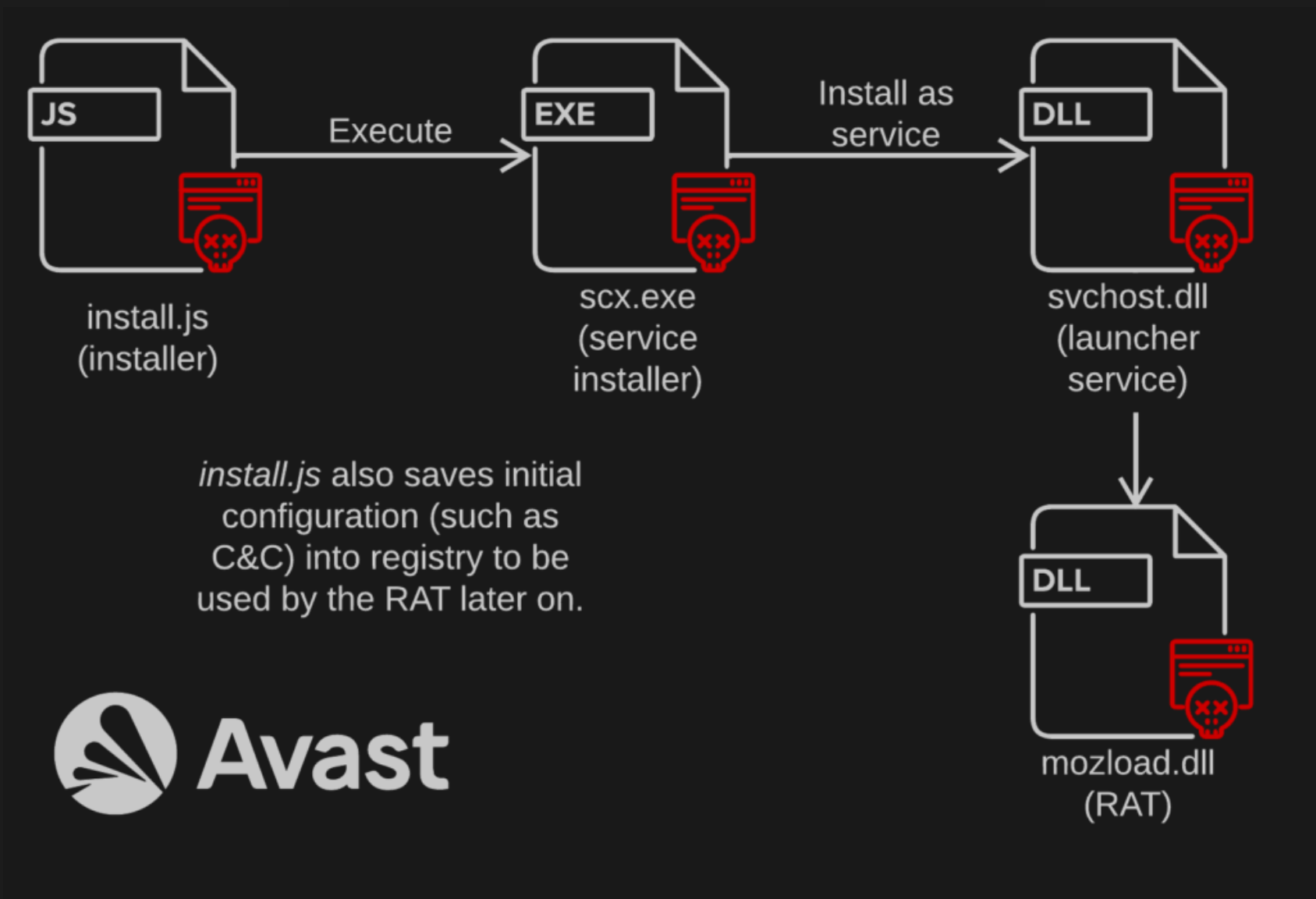
The execution flow of the packages from JSX archives.

HT3

HT3 simply does not fall into any of the previous categories – it is a backdoor with external configuration accompanied with a shellcode loader and UAC bypass.
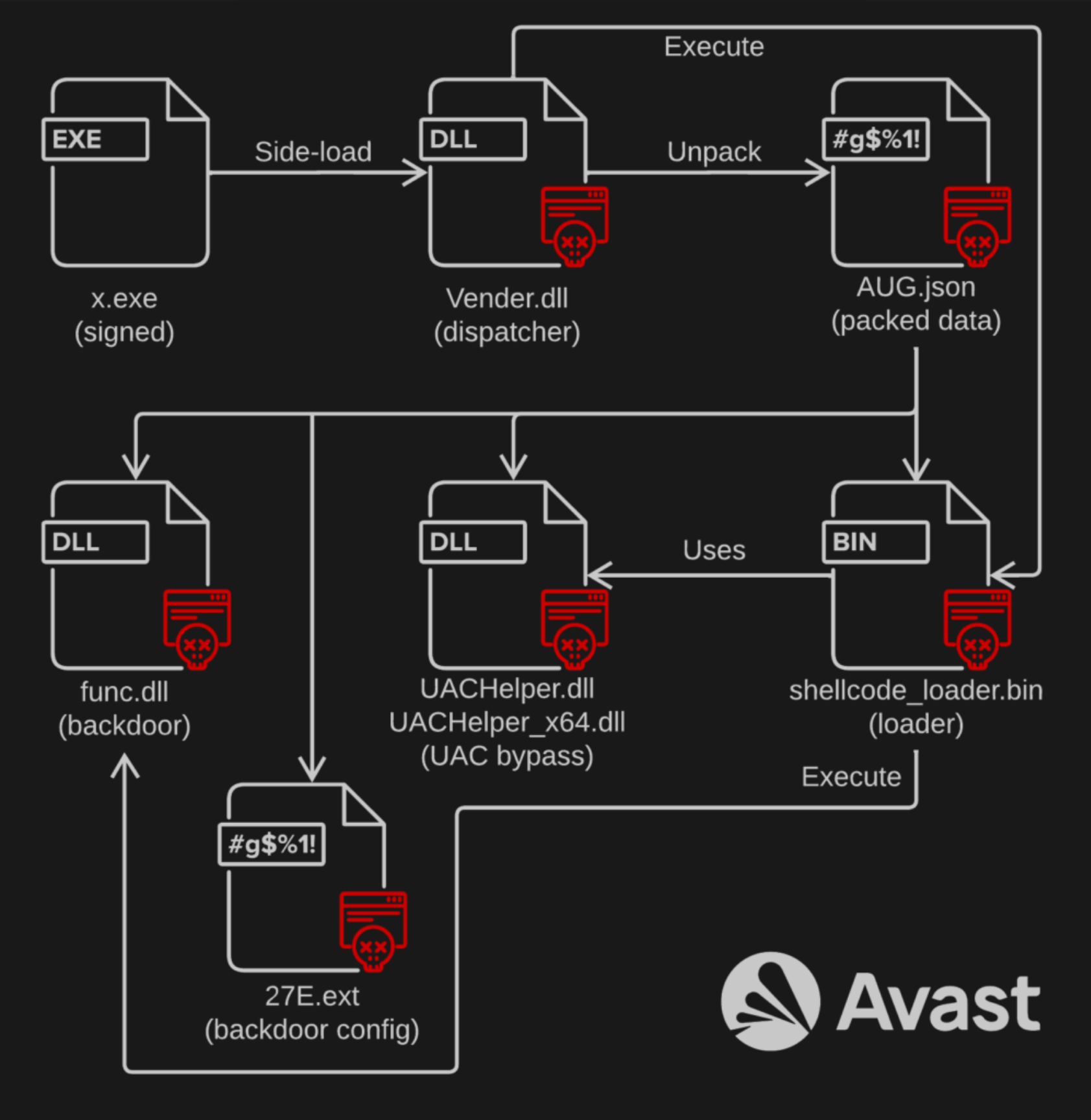
Execution flow of HT3. Note that it contains both 32 bit and 64 bit versions of an UAC bypass tool.

SE

Now we are finally getting to a more complex setup. These archives include several versions with very similar structures and sometimes with varying payloads. Functional changes are presented below; note that these do not include changes in side-loading which will be discussed later on. All versions feature a few evasion tricks that use registry tricks to hide files and file extensions.

Version	Version changes
SE1	Uses volume name for USB installer executable
SE3/SE4/SSE	Uses Delphi launcher (the one attributed to Mustang Panda), persistence integrated into *LPVDPOCX.OCX* (equivalent of *facesdk.dll* from *SE1*)

SE5	Uses volume name for USB installer executable, rollbacks to old USB installer
SE6	No significant functional changes
SE7	No significant functional changes

SE3 and therefore also *SE4* and *SSE* (which are mostly the same) use *vivaldi.exe* and *vivaldi_elf.dll* for its evasion module. Also, *FacialFeatureDemo.exe* and *facesdk.dll* are replaced by *Symantecs.exe* and *LDVPOCX.OCX*; the latter integrating persistence into itself instead of having it in a separate module. Interestingly, the USB installer has been replaced with one similar to the one in the archive *BMK*; a Delphi launcher that executes *Symantec.cmd* which is actually *Symantecs.exe*. These versions are also the only ones that have renamed 3 of 4 payload bundles (using *csdkset.dat* for backdoor, EdrEpmpCStorages.dat for USB installer, and *PchEpmpCStorages.dat* with *WTSAPI32.dll*). Confusingly, *WTSAPI32.dll* does not seem to be used anywhere and will be used for side-loading by later versions when the USB installer replacement is roll-backed.

SE6 and *SE7* abuse old Avast's proxy executable *wsc_proxy.exe* to side-load *wsc.dll* which serves as the dispatcher. Aside from this change, there are no other significant changes to the functionality.
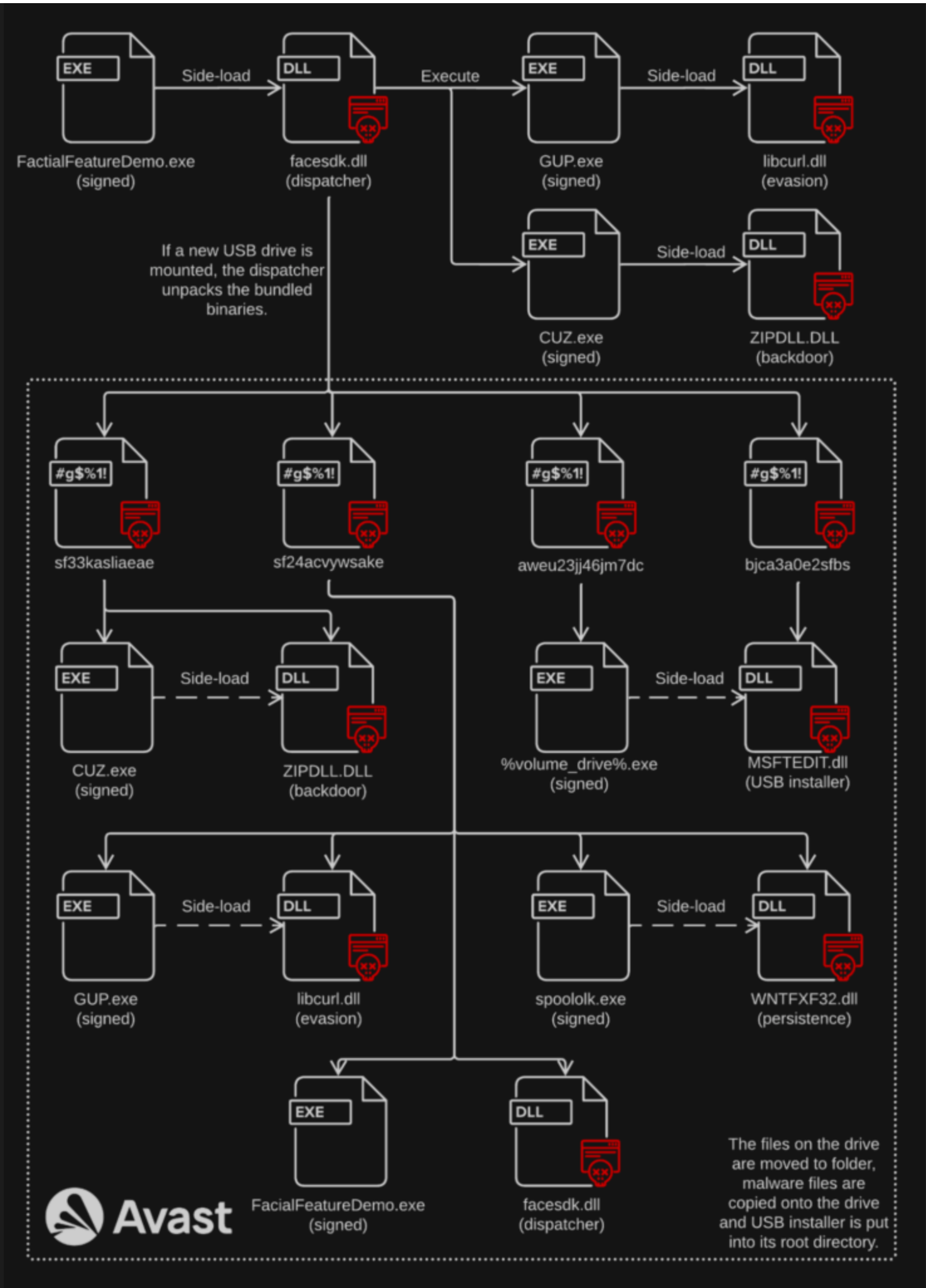
The schema of contents and the control flow of SE1. In newer versions, the persistence module is integrated in the dispatcher.

U5_2

The last complex toolset that we will present is from the archive *U5_2*. Most of the functional code is encrypted and bundled in *AtiVir.csc*. With the exception of a chain to a removable drive watcher, all the parts share similar XOR keys: *user_panda_%section_name%*; a rather interesting choice if the toolset really belongs to Mustang Panda.

An interesting part of the toolset is a file *install_.exe* that reads a file from a given path, takes the serial number of the volume where the file is located, computes MD5 hash from the serial number and changes the first bytes of the file to the hex-encoded computed hash value.
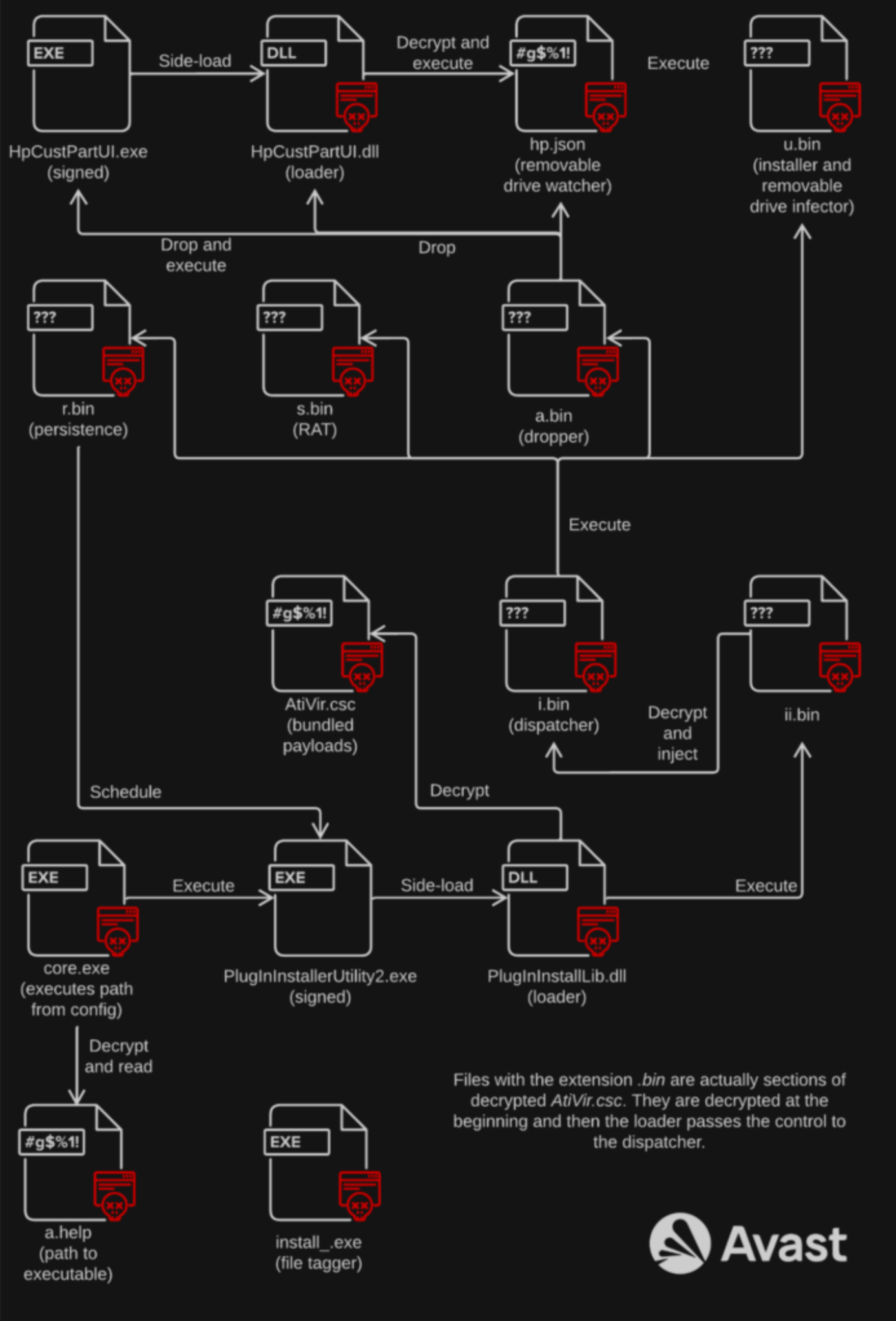
The schema of contents and the control flow of U5_2.

Server infrastructure

The distribution point is an FTP server located in Malaysia that is accessible without a password (username *anonymous*, password is empty). We have also encountered a different FTP server in late 2020 containing very similar archives to the ones we now know contain exfiltrated data. Unfortunately, we did not have enough information to process the archives it contained. We presume that these two FTP servers were closely related or that even the current FTP server may be successor of the one we found previously.

We have noticed that the FTP server has stopped responding in October. Fortunately, the server itself was still alive and the distribution has migrated to using HTTP instead of FTP. They also started using HTTP Authorization; nevertheless, they have reused a weak username:password combination (123:123). This has caused a downtime of a few days in our tracking, but with a quick fix, we've managed to get back on track. Presumably, this might have been another attempt to foil our tracking attempts.

Our telemetry data also revealed another server in Russia. A client from Myanmar tried to download an archive XYZ from it via HTTP. Upon further inspection, the archive was found to be identical to XYZ from the aforementioned FTP server. We have tried to crawl the server for archives and files we have already seen on the FTP server and found the following toolsets:

Archive on the HTTP server	Matching archive on the FTP	Note
gdupiz.rar	GDU	Retrieved from our telemetry
xcrs.exe	X	Discovered by crawling
jsx861.rar	JSX861	Discovered by crawling

The fact that the affected client is from Myanmar, and the fact that the server contains some parts of the described toolset strongly indicates that it is part of the same campaign. Nevertheless, since at least one archive was renamed, we were unable to fully enumerate its contents. Similarly, we were not able to verify whether the server contains exfiltrated data.

C&C infrastructure

JSX RAT

The JSX RAT attempts to communicate with 103.169.90[.]132 using TLS client authentication. The certificate (see Appendix A.2) is hard-coded and we can try to use it to confirm related infrastructure. Looking at the server certificate data, we see that it imitates a real hosting company.

```
Common name:        blue.net
City, country:      San Francisco, CA, US
Fingerprint (sha1): e0adf667e287b0051988dda2b85e7541d7532703
Self-signed
```

Interestingly, the C&C's certificate had the exact same subject as the client certificate. Searching for other servers that use the same certificate yielded a couple more servers. Furthermore, we were able to confirm that the majority of these servers are running the same C&C software because they accept the RAT's hard-coded client certificate.

A server at 118.31.166[.]5 seems to be an outlier among these servers with port 4433 being exposed. Since it is the oldest one using the same certificate and the server exhibits the same communication traits on that port, we suspect that it might be a development server.

Two of these servers were open to RDP connections with a certificate that had *o9c[.]jpg* as its CN (Common name). We tried to go further using these certificates to uncover further candidates for C&Cs. The timeline of uncovered servers is below:
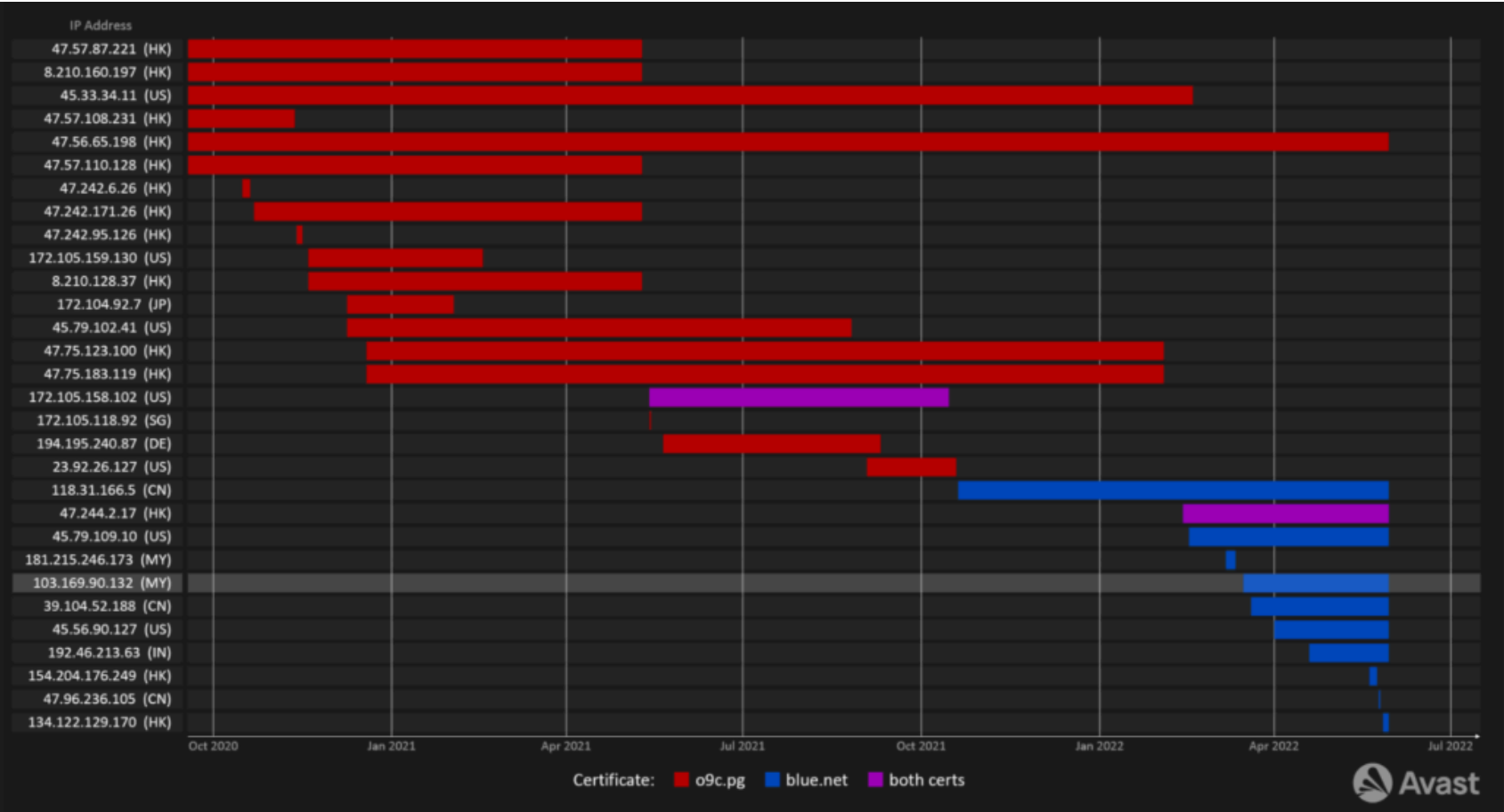
Timeline of servers using the discovered certificates. Blue ones use a certificate with blue[.]net CN, red ones use o9c[.]pg as CN, purple ones have both. The highlighted line corresponds to JSX RAT C&C.

These are servers using the same certificate:

IP	Geo	Notes
103.169.90[.]132	MY	Original C&C Ports: 443, 22, 53, 3389
45.79.409[.]10	US	Suspected RAT C&C on port 443
118.31.166[.]5	CN	Suspected RAT C&C on port 443, different response codes
181.215.246[.]173	MY	
39.104.52[.]188	CN	Suspected RAT C&C on port 443
45.56.90[.]127	US	Suspected RAT C&C on port 443
154.204.176[.]249	HK	
47.244.2[.]17	HK	Suspected RAT C&C on port 443, RDP
47.96.236[.]105	CN	
134.122.129[.]170	HK	Suspected RAT C&C on port 443
172.105.158[.]102	US	Suspected RAT C&C on port 443, RDP
192.46.213[.]63	IN	Suspected RAT C&C on port 443
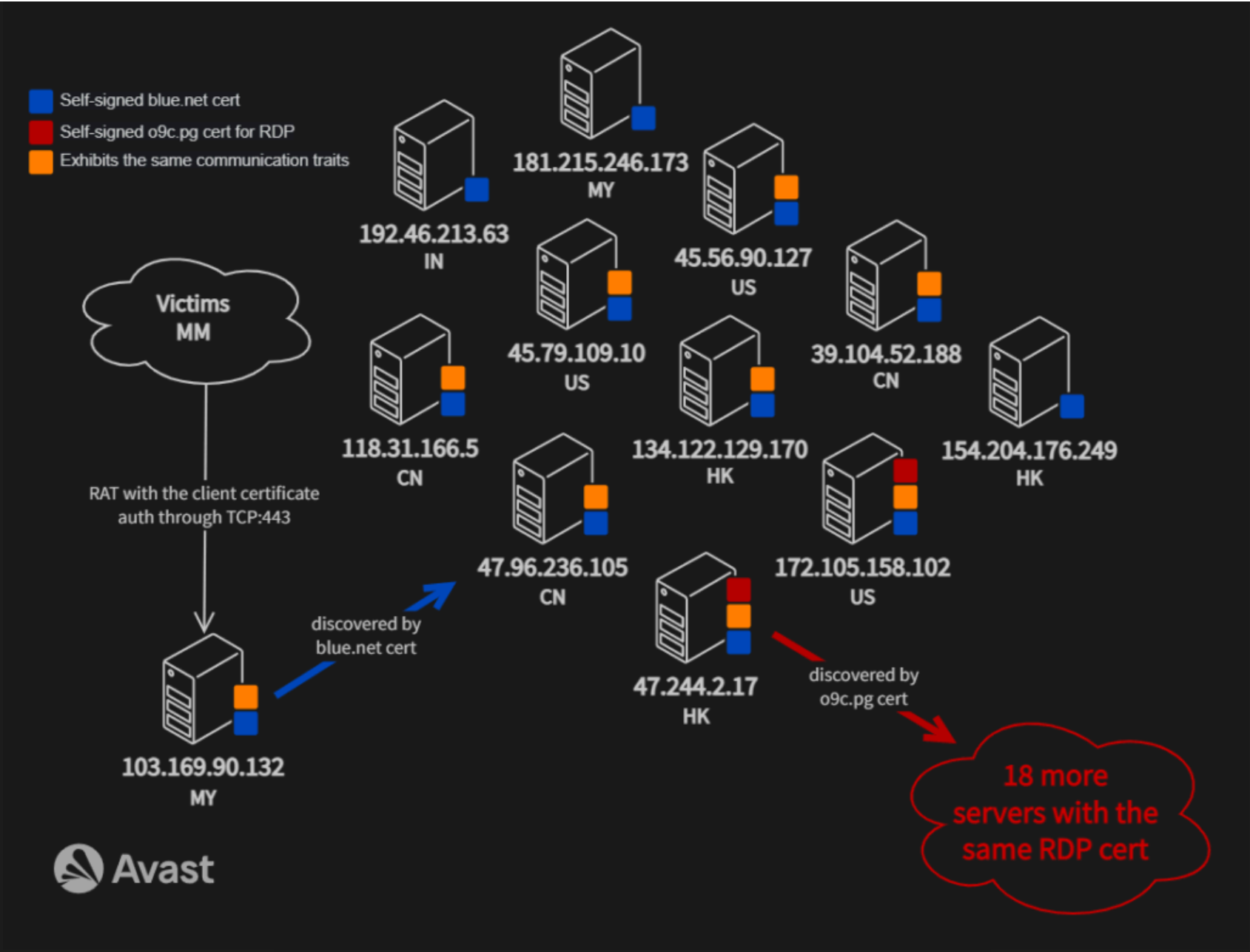
Overview on the servers with ties to JSX RAT C&C.

These are RDP servers sharing the same certificate as the two C&Cs with open RDP port:

IP	Geo	Notes
47.75.123[.]100	HK	
47.242.171[.]26	HK	
47.57.87[.]221	HK	
172.105.118[.]92	SG	
172.105.159[.]130	US	
47.244.2[.]17	HK	Is also on the previous list.
47.242.95[.]126	HK	
8.210.16[.]197	HK	
172.104.92[.]7	JP	
45.33.34[.]11	US	
45.79.102[.]41	US	
47.57.108[.]231	HK	
194.195.240[.]87	DE	
47.56.65[.]198	HK	
8.210.128[.]37	HK	
172.105.158[.]102	US	Is also on the previous list.

47.57.110[.]128	HK	
47.242.6[.]26	HK	
23.92.26[.]127	US	
47.75.183[.]119	HK	

Other C&C servers:

Folder	Geo	C&C	Notes
DISK2/DISKM/MF20211228	RU	188.127.249[.]169	Stealer
HT3	MY	45.121.147[.]172	Korplug
JSX	MY	103.169.90[.]132	RAT
MG/MG44	SG	23.106.122[.]81	Remote shell
SE4/SE5/SE6/SE7	SG	91.245.253[.]72	Backdoor (ZIPDLL.dll)
SEE	MY	103.91.66[.]116	Backdoor (ZIPDLL.dll)
T3YK	MY	111.90.148[.]95	Remote shell
U5_2	MY	103.117.141[.]202	RAT
YK51LOW	MY N/A	mod.mmgpms[.]com txt.mm-film[.]com	Backdoor
1260M	US	45.134.83[.]4	Korplug
GDU1_NEW/GDU3	–	https://github.com/YanNaingOo0072022/View2015	Encrypted Google Drive token

Conclusion

It is not very often we stumble upon such a stash of samples that is used to distribute malware to infected devices; especially when we are talking about tools that are strongly correlated with a notorious APT group. We have shown links to multiple previously published research around campaigns both using tools and TTPs, providing us with high confidence that the threat actor in question is Mustang Panda.

The exfiltrated data indicates that the toolsets that we have found were actively used around Myanmar. For instance, we have found audio recordings that corresponded to the audio recording tools we have identified in the archive named *AUD*. Although many tools were simplistic in their nature and sometimes also in chosen obfuscation methods, some archives contained tools which seem to deserve further analysis; be it due to their complexity or technical implementation.

What was really surprising was the sheer scale of the compromation. We have identified many high-profile government targets, some opposition entities along with a few NGOs. It is worth noting that given the sheer volume of data and the language barrier, we have only been able to associate some of the victims with a specific organization. This means that the list of targets is likely incomplete and should be considered as approximate.

Nevertheless, the daily rate of gigabytes of exfiltrated data should be enough to give a strong hint on the scale of the operation.

Appendix

C&Cs

Folder	C&C	Notes
DISK2	188.127.249[.]169	Stealer
HT3	45.121.147[.]172	Korplug
JSX	103.169.90[.]132	RAT
MG/MG44	23.106.122[.]81	Remote shell
SE4/SE5/SE6/SE7	91.245.253[.]72 193.42.36[.]214	Backdoor (ZIPDLL.dll)
SEE	103.91.66[.]116	Backdoor (ZIPDLL.dll)
T3YK	111.90.148[.]95	Remote shell
U5_2	103.117.141[.]202	RAT
YK51LOW	mod.mmgpms[.]com txt.mm-film[.]com	Backdoor
1260M	45.134.83[.]4	Korplug
GDU1_NEW/GDU3	https://github.com/YanNaingOo0072022/View2015	Encrypted Google Drive token

Certificates and keys

A.1 JSX private key

```
-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIPq1gFM9BHY6lKw+F09iQ4rY5ZDpZhpVuLbLEgKpN1EFoAoGCCqGSM49
AwEHoUQDQgAE8ReYJNz1RlchdTIxo0/4GqPVsJ2m6QFMW0vVMLKYWeINX4Ih9vPV
OgzHq6+qeNxzvAbS4D9jTETTMKssSssr0Q==
-----END EC PRIVATE KEY-----
```

A.1 JSX certificate

```
-----BEGIN CERTIFICATE-----
MIICBDCCAamgAwIBAgIUPAoKZshUkyHcTvej+gio/kTTd/AwCgYIKoZIzj0EAwIw
RTELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAkNBMRYwFAYDVQQHEw1TYW4gRnJhbmNp
c2NvMREwDwYDVQQDEwhibHVlLm5ldDAeFw0yMTEwMTIwMzQ0MDBaFw0zMTEwMTAw
MzQ0MDBaMEUxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJDQTEWMBQGA1UEBxMNU2Fu
```

```
IEZyYW5jaXNjbzERMA8GA1UEAxMIYmx1ZS5uZXQwWTATBgcqhkjOPQIBBggqhkjO
PQMBBwNCAATxF5gk3PVGVyF1MjGjT/gao9WwnabpAUxbS9UwsphZ4g1fgiH289U6
DMerr6p43HO8BtLgP2NMRNMwqyxKyyvRo3cwdTAOBgNVHQ8BAf8EBAMCBaAwEwYD
VR0lBAwwCgYIKwYBBQUHAwIwDAYDVR0TAQH/BAIwADAdBgNVHQ4EFgQULCt+JQ1h
n+CNR27Lm0giuJbAHGAwIQYDVR0RBBowGIIIYmx1ZS5uZXXSCDHd3dy5ibHVlLm5l
dDAKBggqhkjOPQQDAgNJADBGAiEA9c8UxcF/xYGVThbl3vfKpmJQKQLi8LP+2cui
o9Z3iZICIQCxJOXs+4ScVpyIkw8HYCCv3x0CDVv4xfiGHEEW+ZLZYA==
-----END CERTIFICATE-----
```

Side-loads

Archive	Binary	DLL
BMD,YK41	Symantec.exe	SHELLSEL.ocx
YK41LOW	GoogleUpdate.exe	SbieDll.dll
YK41LOW	atkexComSvcRes.exe	CefBrowser.dll
HT3	x.exe	Vender.dll
GDU	Samvd.exe	SmadHook32.dll
GDUPIZ	vsgraphics.exe	vsgResources.dll
GDU1/GDU2	CefSub.exe	CefBrowser.dll
GDU2/GDUPIZ/GDU1_NEW/GDU3	piz.exe	vntfxf32.dll
GDU_OLD	AltTraceTool.exe	atltracetoolui.dll
GDU1_NEW/GDU3	NitroPro.exe	CefBrowser.dll
GDU3	piz.exe	FwcWsp.dll
WD	HP.exe	HPD.dll
DISK2/DISKM/MF2021188	HPCustPartUI.exe	HPCustPartUI.dll
127C	OleView.exe	ACLUI.dll
KKL/KKL1	mscorsvw.exe	mscorsvc.dll
MG/MG44	dabs.ex/44.ex	SensorAware.dll
AUD	msync.ex	mcaltib.dll
CHR	browser.ex	browser_elf.dll
T3YK	ygfdt.exe	corecrl.dll
UC	melt_64.exe	libmlt-6.dll
1260M	upservice.exe	breakpad.dll
U5_2	HPCustPartUI.exe	HPCustPartUI.dll
U5_2	PlugInInstallerUtility2.exe	PlugInInstallLib.exe
SE1	FacialFeatureDemo.exe	facesdk.dll
SE1/SE3/SE4/SE5/SE6/SE7/SEE	CUZ.exe	ZIPDLL.dll

SE1	%volume_drive%.exe	MSFTEDIT.dll
SE1/SE7	GUP.exe	libcurl.dll
SE1	spoololk.exe	WNTFXF32.dll
SE3/SE4/SE5/SE6/SEE	vivaldi.exe	vivaldi_elf.dll
SE6/SE7	wsc_proxy.exe	wsc.dll

File hashes

\1260M

```
1ded7b4cab302bc7229c92723056d07d5bd9563e88fe082da0a396942fba5958 breaklog.dat (Korplug)
2895fdac192a4b0ffd70b6b207d49cd7c8f68945eb5c09e3d51e2fded6c6c32f breakpad.dll (Loader)
ce13248fa2da5b27773f855c2dd0c6ce276b4a10b020e4da57bc47ab0fe07eae upservice.exe
```

\127C

```
1769c7778cbcd937ae317f4982f404b0d7ae7ee5e2b2af4efb160c5233a8f476 ACLUI.dll (Korplug)
8ff84f79455b84bd73e7c0641532a60e8132599c29d3f85fb54f3d7da53e1817 ACLUI.DLL.UI (Loader)
91f6547bceddfb2f241570ac82c00de700e311e4a38dea60d8619638f1ed3520 OleView.Exe
```

\AUD

```
cd6bcf240de87fe3f1b5a6a24db1b2728acad5f7bcfe124e5bc2d7bdac2f64a9      mcaltlib.dll      (Audio
recording)
075f9dfb6ab3379f69165c03991abf1a969ca0c21e04564543564dc536ea95dd mcsync.ex
```

\BMD

```
55eacabb7c054355d2e8c3a82c075338c9ac642d86ee5d3fa1fca3f621e43cb2 SHELLSEL.ocx (Loader)
d139940023fa2c602e2a31faa807b9df074f34747511bd61db961d20155b8c84 ShellselDb.dat (Korplug)
61d1943f0b702f4c16bb37228ade1d8f0ef4675b480921950d026c82e4a65fde Symantec.exe
491d9f6f4e754a430a29ac6842ee12c43615e33b0e720c61e3f06636559813f7 SymantecHp.exe (Launcher)
```

\CHR

```
e3e2164c54a5c8ab063695bc41b6c0c0ddc390c790de8ad24d6169dba46f7734 22.1.1.1543.manifest
4063cf4ba2d4e12c277479399d4489e45a48b1013d8d54b5a589784fe7158978    browser_elf.dll    (cookie
dumping)
12b15f31f295416417f1e028904a4e10a0c8ec39dd00bac7df4887c194f2865b browser.ex
```

\DISK2

```
8857232077b4b0f0e4a2c3bb5717fd65079209784f41694f8e1b469e34754cf6           HPCustParticUI.exe
(Sideload)
7ea21215968c43f9fe28f94926e3547f2f7a0e35cdf40392b0b6aea80fe11314 HPCustPartUI.dll (Stealer)
c9121c7874d2fd88ff7af35eb3f3cd18ab7162390db008043037383cdad6ff56  HPCustPartUI.log  (Ecnrypted
stealer config)
86e39b5995af0e042fcdaa85fe2aefd7c9ddc7ad65e6327bd5e7058bc3ab615f MSVCR120.dll
```

\DISKM

```
8857232077b4b0f0e4a2c3bb5717fd65079209784f41694f8e1b469e34754cf6           HPCustParticUI.exe
(Sideload)
788fe764f6f5e0fff31b06fd8b94ee0cf51a3082c1321d8db96708c2a6abc3ee HPCustPartUI.dll (Stealer)
62d770f240cecebd6cf725df05ab1e863d83895abc9396664a6090dbcc983d6f  HPCustPartUI.log  (Ecnrypted
stealer config)
86e39b5995af0e042fcdaa85fe2aefd7c9ddc7ad65e6327bd5e7058bc3ab615f MSVCR120.dll
```

\GDU

```
d12a1750980ef3943c6d7e152948059261944b8afe06b8a280b7fbe61aba3c9b rar.dat (Config)
e64b533d60a21ca8ddbfcf8a1b154ed351383b0196d534bf229101a6cc4d1931 rar330.exe (RAR)
def8fdb95bb53514698b9df1c64e329adfca59adf2e898c3daab16f1e4760bc6 RarCon.SFX (Exfiltrator)
4f54a6555a7a3bec84e8193d2ff9ae75eb7f06110505e78337fa2f515790a562 Samvd.exe
375e88d1f38604c901f2b9fd5b9ada4c44c1f4c172f7cd58cd67e9248ff966ab SmadHook32c.dll (Loader)
```

\GDU-OLD

```
197d0ad8e3f6591e4493daaee9e52e53ecf192e32f9d167c67f2ffb408c76f2c AtlTraceTool8.exe
33f631c0b561199b5feb9020faa99e50efa9f421d7484ffa640c5561494726da mfc110u.dll
45a61f4b7e5798f1389a7d6abc8a924c37db6f51552b4cafc901e7e4a50dabc6 msvcr110.dll
db75b25b69b7b6f3206226461d8bde7c05049922dc463e8932d11710fad74833 piz.exe (Collector)
46811fc41623677637aaadcfbe89811d187b390bfd7e4f3e8efd2dd1d078a631 rar.dat (config)
0dcaf08b7b1f8de3999af567144b13f36bea3a68f46f81f8443a81a50a86a09c atltracetoolui.dll (Loader)
```

\GDU1

```
2c17b68040dda192939e4b7f65b2935cb6c467b8a4b2c3d512bde6cc5a60adaf cache (Exfiltrator)
e412569c23722c469ee533efb62bbded53d1909b58c8cf7bff9897c466c9df9a CefBrowser.dll (Loader)
cb8a83b590893daa9b02b8e1a1c9afb68d6f2a82c9e0d2d2c63a36a510f6fda3 CefSub.exe
beb44eadd141b7ae46e40e1bf888c302cb7096826e772f0b20ce6f213c69058d piz.exe (Collector)
46811fc41623677637aaadcfbe89811d187b390bfd7e4f3e8efd2dd1d078a631 rar.dat (config)
```

\GDU2

```
2c17b68040dda192939e4b7f65b2935cb6c467b8a4b2c3d512bde6cc5a60adaf cache (Exfiltrator)
e412569c23722c469ee533efb62bbded53d1909b58c8cf7bff9897c466c9df9a CefBrowser.dll (Loader)
cb8a83b590893daa9b02b8e1a1c9afb68d6f2a82c9e0d2d2c63a36a510f6fda3 CefSub.exe
390d75e6c7fc1cf258145dc712c1fac1eb183efccee1b03c058cec1d790e46b1 piz.exe (Collector)
46811fc41623677637aaadcfbe89811d187b390bfd7e4f3e8efd2dd1d078a631 rar.dat (config)
869b8dd87e402049eae435de3de1e15a021d9fcbf79a20be3b030d3782599903 vntfxf32.dll
```

\HT3

```
59cf961f7316656e73b269a86b04836a7a7254f021a8a3132a927b02373225d6  AUG.json  (Encrypted  and
compressed data)
091408cdd56267bc4fb4cb54f2d91701aa8cdcede334a648566eea89f1682925 Vender.dll (Loader)
00bfbbe6e9d0c54312de906be79cc1e9f18b2957856a1215eaff1ac7bb20e66f x.exe
```

\JSX64

```
c617016fb8809655f9189648b9b41a727c0b49cdb79a28f13f710d23f3527a64 install.js (Executer)
21bf4631775b6c17f9e94c0901ffbb7718a0e6094582bcb1683b934aca24e18f mozload.dll (RAT)
f4a31d15cd5aa3441e5e31c1add6e0c3551a1aad5abb75f0abd76990f2824acc scx.exe (Installer)
645ee3601aea4c1af8b938f64698bf6c5978b1151aef53e183bb768791c927e2 svchost.dll
```

\JSX641

```
73903c2c46b5055380fc2a238c96f7f2ca2a5acf1cd1e568b2d2be0638c68fd1 install.js (Executer)
50bee35c965a99b3f8f722296e4ed6474ca62d96ea5fc4897e7d1563ed173d5e mozload.dll (RAT)
e27bfbe87c78945b1d79fc027c3f0a27a07d0dddc742783bf686c1a8133a2f48 scx.exe (Installer)
8cebfe33cd69747cc1333fe598d9b0331103e0869d6f1b1f75e28b3b8f11243d svchost.dll
```

\JSX86

```
c617016fb8809655f9189648b9b41a727c0b49cdb79a28f13f710d23f3527a64 install.js (Executer)
fd1ec183124d2d82dae1dd228de88440bc142cf6430c9c93518e25f1dde052fd mozload.dll (RAT)
9e3788cacb3d38e4e15da7e4887650efa6a3b17a65a314fcb4e059d9f88481a8 scx.exe (Installer)
5e8311c26091839a292e2d12f88378f8093fc739ced86aa1e9ba1b707ad516d8 svchost.dll
```

\JSX861

```
73903c2c46b5055380fc2a238c96f7f2ca2a5acf1cd1e568b2d2be0638c68fd1 install.js (Executer)
f9d94c1dcdbcefddb4f1d47291422c6198fd11052aea761acf8b5755802ca922 mozload.dll (RAT)
49a81878ec282c3c9d4dd72920d9283e2c86d0bb96b468e010901b3f4f9c75ed scx.exe (Installer)
79440abf29d1b56cb1c95a12f554fe052e21a865fea56a025e216f342ffbccbd svchost.dll
```

\KKL

fae5b61723106d44de46b3ec49e80067f63f82f09501142186984a658bc99c38 KBE.dll (Clipboard stealer, keylogger)
ed6b3af0edcd3b57c0616e1b7819b5e1c1e72327300172ff2664b158f65861b2 mscorsvc.dll (Loader)
0809e3b71709f1343086eeb6c820543c1a7119e74eef8ac1aee1f81093abec66 mscorsvw.exe

\KKL1

4afa4582975d31144b3af692f123f87b6400a45475e41fa1822c7acdb17590f0 KBE.dll (Clipboard stealer, keylogger)
9af8336050c40105864bf9314355471494dc631fd88a0b444291b63b941b7822 mscorsvc.dll (Loader)
0809e3b71709f1343086eeb6c820543c1a7119e74eef8ac1aee1f81093abec66 mscorsvw.ex

\MF20211228

8857232077b4b0f0e4a2c3bb5717fd65079209784f41694f8e1b469e34754cf6 HPCustParticUI.exe
9f1d1a94026c54396a4c0b6327d317836dc9dc67178810428302efcbf5225a42 HPCustPartUI.dll (Stealer)
86e39b5995af0e042fcdaa85fe2aefd7c9ddc7ad65e6327bd5e7058bc3ab615f MSVCR120.dll

\MG

473b4f8b8640a68d1092f6b54b521c6b0ccb1c567eca4a18a2c2da3481bc027a dabs.ex
cfe1447e7515ad831fcfedb9a5c1a721885b0542b775e4028a277a27e724ec73 SensorAware.dll (Figerpring, remote-shell)

\MG44

473b4f8b8640a68d1092f6b54b521c6b0ccb1c567eca4a18a2c2da3481bc027a 44.ex
cfe1447e7515ad831fcfedb9a5c1a721885b0542b775e4028a277a27e724ec73 SensorAware.dll (Figerpring, remote-shell)

\NB

c9d5dc956841e000bfd8762e2f0b48b66c79b79500e894b4efa7fb9ba17e4e9e nb.dat (Nbtscan)

\SE1\Bin

8364bae4e2951957403cbe3a78362edb7d41c34f49c81f0336fcb28d1510d5e1 facesdk.dll (Dispatcher)
0d243cbcd1c3654ca318d2d6d08f4e9d293fc85a68d751a52c23b04314c67b99 FacialFeatureDemo.exe
e5bbbf34414426f63e6cd1354c306405e54bf31279829c7542dccfb7d85af0ec GUP.exe
92717951aae89e960b142cef3d273f104051896a3d527a78ca4a88c22b5216a5 gup.xml
7e1c49d45935fb5d20add5baf60400fb64fbf0299a3af3b0be764b2d265e368a libcurl.dll (hides malware files)
390d75e6c7fc1cf258145dc712c1fac1eb183efccee1b03c058cec1d790e46b1 spoololk.exe
abf7bb6eb92f2f358e8e57c1be03efe5a7f81e3d3eb4134257c3483e9e7782c0 VNTFXF32.dll (persistence)

\SE1\Data

```
1a4e92e09957578cc8d8c1fbdaba55e306e7bcbc6208ee00e33bb37e849156f9 aweu23jj46jm7dc
13cf1c57f1c143c592173b1e91ddb652d5dd1c2015289ac890a37253058b54be bjca3a0e2sfbs
12acd296a009d9e8fbd9511d3c0586f331d450b9c12f651e0554764e50cfb7e7 sf24acvywsake
4a6ed717a2d7f0953e4b25c2652c9a231146f60b35d9a5e3cf782c772727b1bc sf33kasliaeae
```

\SE3\Bin

```
61d1943f0b702f4c16bb37228ade1d8f0ef4675b480921950d026c82e4a65fde Symantecs.exe
a8f3bc45ac0dcf351c028ecabfd68e8e551cd97f8dc0fc6e62e135668cde9277 LDVPOCX.OCX (Dispatcher)
bb6cf240bdabeea90321cab7d48e268df2b5240d84aab0d5ae5ffe415a6943e4    vivaldi_elf.dll    (hides
malware files)
58e7af5eb1acb5c9bee821d59054c69263aed3dce1b95616255dea7114ad8494 vivaldi.exe
```

\SE3\Data

```
51c3d115e0173e3ba6eeaea3d53b86bce45367e50feed82d8efed2065d845d28 3.8.2259.42.manifest
6ded96d7609cc085db57764c40a38379cba50b965f959650ca8d1605ae0411e8 csdkset.dat
a8f0dff3c57621282a1262ddaa559f055f2f2cc717a7695d8bfbf7a6898b843c EdrEpmpCStorages.dat
7659be61fc1e16c4721b451225ec7c8f932e9e7357894ddea3a4ada9583996b5 PchEpmpCStorages.dat
9015378ed6d7537f07e61c78b3c35766d63465970b63d13c9b447dc8bb90e2d7 prodcltdef.dat
```

\SE4\Bin

```
61d1943f0b702f4c16bb37228ade1d8f0ef4675b480921950d026c82e4a65fde Symantecs.exe
01cd1530b0db54c834ef275e0cc617645a23e1f250bc35c248d546c28da220fb    vivaldi_elf.dll    (hides
malware files)
9fb4c9f1995b02ece99b62a4efc0df5c916a1858f57730225f3c419fce0de24c LDVPOCX.OCX (Dispatcher)
58e7af5eb1acb5c9bee821d59054c69263aed3dce1b95616255dea7114ad8494 vivaldi.exe
```

\SE4\Data

```
51c3d115e0173e3ba6eeaea3d53b86bce45367e50feed82d8efed2065d845d28 3.8.2259.42.manifest
01cd1530b0db54c834ef275e0cc617645a23e1f250bc35c248d546c28da220fb csdkset.dat
baaaffe80060fb89b06ff19dfb6c76835fc6639d81513e2d9e49716f1816ccc4 EdrEpmpCStorages.dat
9fb4c9f1995b02ece99b62a4efc0df5c916a1858f57730225f3c419fce0de24c LDVPOCX.OCX
f488e4e838fa447c9b08fc74d4180faeb465f9070c443625b7515aed7c282fa6 PchEpmpCStorages.dat
ab89d614923b92ce2eb7ed48357b2d1755b8a8f572ead3b32bb63a79e259186d prodcltdef.dat
```

\SE5\Bin

```
5828fd07716140e5fefec1b07751378d9b76952e66b2c0fb0a860313d4030b4d LDVPOCX.OCX (Dispatcher)
61d1943f0b702f4c16bb37228ade1d8f0ef4675b480921950d026c82e4a65fde Symantecsy.exe
bb6cf240bdabeea90321cab7d48e268df2b5240d84aab0d5ae5ffe415a6943e4    vivaldi_elf.dll    (hides
```

```
malware files)
58e7af5eb1acb5c9bee821d59054c69263aed3dce1b95616255dea7114ad8494 vivaldi.exe
```

\SE5\Data

```
e6fdd0d22abe3484d57715bd83143e5810b74f3f9dc8780344c66af2c0894d76 aweu23jj46jm7dc
50814a35a9d157405252c8ba52c12d1cf5adf137598173c6522cbe058e14b7ff bjca3a0e2sfbs
1d68f4afd0fd908d35db6d9710ab2fc92fb5ca739d6351e1bf513e068fbd00a0 sf24acvywsake
5427cd51f0120a27ed75d3ac27d6f8eac6f27c54d8658236a52a281d6433496b sf33kasliaeae
```

\SE6\Bin

```
a67094334ae2135e50bf2074f08d3a99075a53a174da6bdf22eca54293bb8e9b   vivaldi_elf.dll   (hides
malware files)
58e7af5eb1acb5c9bee821d59054c69263aed3dce1b95616255dea7114ad8494 vivaldi.exe
bd4635d582413f84ac83adbb4b449b18bac4fc87ca000d0c7be84ad0f9caf68e wsc_proxy.exe
e0c240f5776d158da7529d8c0e3d5be4d6f007e51e4be570e05b744d0452011d wsc.dll (Dispatcher)
```

\SE6\Data

```
51c3d115e0173e3ba6eeaea3d53b86bce45367e50feed82d8efed2065d845d28 3.8.2259.42.manifest
7620acb11f0471515079a69ee2cec0cd74485fb13c779d41c2b43b87718c63ff aweu23jj46jm7dc
3fc3fb81a43b9ac155e42367769eb5c0d6dd08c06a025ba93697c6b2667bf1e7 bjca3a0e2sfbs
f2c5004450a749bef14ee779e1c8e4c08702f089248d0a282e6a679d29b0996d sf24acvywsake
10d58013b8a34e10e8548b016154963097dcff15e5673bf24e8ed18513ad4a64 sf33kasliaeae
```

\SE7\Bin

```
e5bbbf34414426f63e6cd1354c306405e54bf31279829c7542dccfb7d85af0ec GUP.exe
92717951aae89e960b142cef3d273f104051896a3d527a78ca4a88c22b5216a5 gup.xml
7e1c49d45935fb5d20add5baf60400fb64fbf0299a3af3b0be764b2d265e368a  libcurl.dll  (hides  malware
files)
bd4635d582413f84ac83adbb4b449b18bac4fc87ca000d0c7be84ad0f9caf68e wsc_proxy.exe
e4ddf5af63fdfe85c5a4573d4768699ebdaa5b5b67b7cb6834840c696808a8e5 wsc.dll (Dispatcher)
```

\SE7\Data

```
b7a38292131c131d75413133f101114a1b72bd02e27cc6aea7a836ff964f961f sf24acvywsake
28aadf5b14ba0cb38a33ab53796dba12e7d59479744f0cca225b10be44730b9c sf33kasliaeae
ec56a6fa6804e47f331daee1460c3d07e01fe45edac5d6b1feb01fbbd8396f91 aweu23jj46jm7dc
e32447bd309a6941a1fff4fa559376d9c723afd1b9ce2a1c2dced4b9db6a6f6a bjca3a0e2sfbs
```

\SSE

```
51c3d115e0173e3ba6eeaea3d53b86bce45367e50feed82d8efed2065d845d28 3.8.2259.42.manifest
5dafacfa147f087dd0a706cf274e20cbb58f634ba14424d3433efc2e829aa7cd csdkset.dat
b9924c66506ccad566d6c26b8db499e498a9dc840acacb2d8d3bf9d73818814a EdrEpmpCStorages.dat
180a2f3eb004f93590e4fb18cdc3dd6e18815587637ac354ca99f7513aa63633 LDVPOCX.OCX (Dispatcher)
9add5663bc846b4b7cdefcd0e09b882e2f16f755e2e6540efc6ea2072c93f3f2 PchEpmpCStorages.dat
756d1cb0e74b309d53d4f16b043514da128c8b3b89c7d5e46897b61f74bad2d7 prodcltdef.dat
61d1943f0b702f4c16bb37228ade1d8f0ef4675b480921950d026c82e4a65fde Symantec.exe
bb6cf240bdabeea90321cab7d48e268df2b5240d84aab0d5ae5ffe415a6943e4    vivaldi_elf.dll    (hides
malware files)
58e7af5eb1acb5c9bee821d59054c69263aed3dce1b95616255dea7114ad8494 vivaldi.exe
```

\T3YK

```
3c0d3783a5ccdecf3786db0053b1352d6fb5a37d9081cc32ec6d5bb611064ce3 coreclr.dll (remote shell)
f11009988b813821857c8d2db0f88e1d45b20762f62a3cf432339f352b12cefe ygfdt.exe
```

\U5_2

```
2a971ba79f9f7378e11a47dcafa78e9fe4b1f0c659f7f310209d3e6671d5dc31    Ativir.csc    (bundled
payloads)
9cd8c5d34fd460dd0e240f5e54ade689d808469d6da5e0bd087cc71e6f851c6a core.exe (executes path from
config)
caeb48fd04a5fe8b0b4bd32b538ed5f1f303b0487037cf37864f0b5665ff093a install_.exe
2f2a5e5cdb262cd62b43b88bf1e9cfb40a26eac5897616b9eacec4e25d95cbb9 PlugInInstallerUtility2.exe
a90e048c74697775bba2e4c4bfa45d369e44e9a020a83956aa44a50ab8a9a249 PlugInInstallLib.dll
```

\UC

```
f349183462f1aeac8d3afb43c723af0252c157d376637f30fb7c87fdf80ee965 libmlt-6.dll (UAC bypass)
a23dbce5bcde8ce541b8f326a951d29f6241280d944a1e921ca8658d3d4b65ac melt_64.exe
```

\WD

```
8857232077b4b0f0e4a2c3bb5717fd65079209784f41694f8e1b469e34754cf6 HP
97efd0abf726acfc1a5b4a0b460a727724f43ef9f1e788bada4942d715d4ab87 HPD (Loader)
86e39b5995af0e042fcdaa85fe2aefd7c9ddc7ad65e6327bd5e7058bc3ab615f MSVCR1
5f31d558417528b4c635afd6c17347dc393c7dfcecfb79040fe97d9f1abf3776 S (korplug)
```

\X

```
28bed0d5bcfb2d5597af881a2be3098327f2d83f14948c6a46cde3cd0776eb1c x.ex (status checker)
```

\YK41

```
edab53d39734965a7cadc2a21662d6a16c9b04b2961dfe9eb76aeda040786e25   hp_ui.xslbcdsj   (encrypted
Korplug)
071558464f6d067f3044b7ee3819fcb3a049b8be3535043db41123c2fde5d451 SHELLSEL.ocx (Loader)
61d1943f0b702f4c16bb37228ade1d8f0ef4675b480921950d026c82e4a65fde Symantec.exe
```

\YK51LOW

```
cb8a83b590893daa9b02b8e1a1c9afb68d6f2a82c9e0d2d2c63a36a510f6fda3 atkexComSvcRes.exe
9bdccd5e4617dfbcaf85228c60703369a8848ad8bb86e00e75e504a26fbe932a CefBrowser.dll
b29b38217921a6b36113049bd9cb4fb2ec52816bff7cd731621ff2fa3dbc7b01 DP45126C.lfl (Decryptor)
90a29c688ce683fb2201145faac00cb44c3d5566697279b68960c6bc3208ae84 GoogleUpdate.exe
fa56ba25861f1b5040afd04bfbfd36353004cd6b2c457971fb01db26ff002f35 GoogleUpdateOnDemand.exe
c9ed69e7bf233ba1edd18a1f91671faee9b7756aa77fe517319098706e78cde5 SbieDll.dll (Loader)
```

Further reading

___
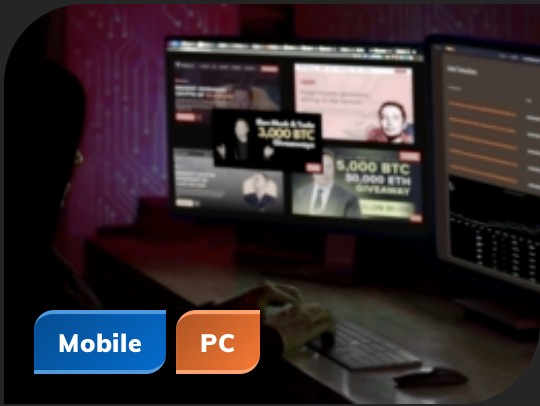
Mobile | PC

CryptoCore: Unmasking the Sophisticated Cryptocurrency Scam Operations

August 13, 2024

– by **Martin Chlumecký**

As digital currencies have grown, so have cryptocurrency scams, posing significant user risks. The rise of AI and deepfake technology has intensified scams exploiting famous personalities and events by creating realistic fake videos. Platforms like X and YouTube have been especially targeted, with...

PC

Decrypted: DoNex Ransomware and its Predecessors

July 8, 2024

– by **Threat Research Team**

Researchers from Avast have discovered a flaw in the cryptographic schema of the DoNex ransomware and its predecessors. In cooperation with law enforcement organizations, we have been silently providing the decryptor to DoNex ransomware victims since March 2024. The cryptographic weakness was...

PC

New Diamorphine rootkit variant seen undetected in the wild

June 18, 2024 – by **David Álvarez**

Introduction Code reuse is very frequent in malware, especially for those parts of the sample that are complex to develop or hard to write with an essentially different alternative code. By tracking both source code and object code, we efficiently detect new malware and track the evolution of...