BOLETIM DE SEGURANÇA

RansomHub adota nova ferramenta para desativar EDR em ataques recentes

heimdall
security research
A DIVISION OF ISH

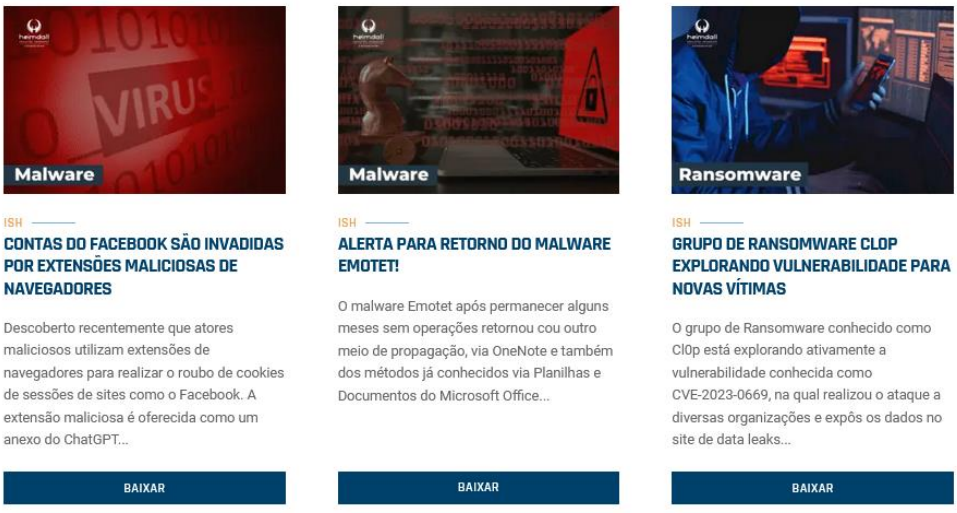
TLP: CLEAR

Receba alertas e informações sobre segurança cibernética e ameaças rapidamente, por meio do nosso **X**.

[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]



SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

1 SUMÁRIO EXECUTIVO

Um grupo de criminosos cibernéticos associado ao ransomware RansomHub foi identificado utilizando uma nova ferramenta destinada a desativar o software de detecção e resposta de endpoint (EDR) em sistemas comprometidos. Esta ferramenta se junta a outras similares, como AuKill (também conhecido como AvNeutralizer) e Terminator, ampliando o arsenal de técnicas para evadir a segurança dos endpoints.

2 INFORMAÇÕES SOBRE A AMEAÇA

A Sophos identificou um novo utilitário de desativação de EDR sendo utilizado por um grupo criminoso que tentou atacar uma organização com o ransomware RansomHub. Embora o ataque não tenha sido bem-sucedido, a análise pós-incidente revelou a presença de uma nova ferramenta projetada para desativar o software de proteção de endpoint, denominada **EDRKIllShifter**.

Desde 2022, observou-se um aumento na sofisticação do malware destinado a desabilitar sistemas EDR em sistemas infectados, à medida que mais clientes adotam essas ferramentas para proteger seus endpoints. Durante o incidente em maio, os agentes da ameaça, que se acredita com confiança moderada estarem utilizando essa ferramenta — tentaram usar o EDRKIllShifter para desativar a proteção da Sophos no computador alvo, mas a ferramenta falhou. Em seguida, tentaram executar o ransomware na máquina controlada, mas também falharam quando o recurso CryptoGuard do agente de endpoint foi ativado.

A ferramenta EDRKillShifter funciona como um executável "**loader**", servindo como um mecanismo de entrega para um driver legítimo que pode ser explorado ("*bring your own vulnerability driver*" ou BYOVD). Dependendo das necessidades do agente malicioso, ela pode entregar diversos payloads de driver. O processo de execução deste carregador ocorre em três etapas.

Primeiro, o invasor deve executar o EDRKillShifter com uma linha de comando que inclui uma string de senha. Com a senha correta, o executável descriptografa um recurso embutido chamado BIN e o executa na memória. O código BIN, então, descompacta e executa o payload final. Este payload, escrito em Go, descarta e explora um dos vários drivers legítimos vulneráveis para obter privilégios suficientes e desativar a proteção de uma ferramenta EDR.

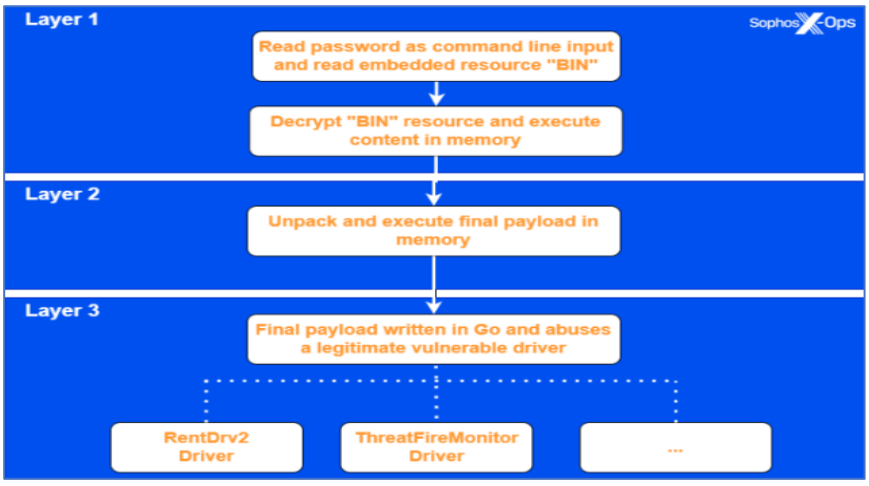


Figura 1 – Visão geral de alto nível do processo de execução do carregador.

Uma análise inicial mostrou que todas as amostras possuem os mesmos dados de versão. O arquivo original é chamado Loader.exe e seu produto é ARK-Game. Pesquisadores sugerem que o atacante tenta disfarçar a carga útil final como o jogo ARK: Survival Evolved. O binário está configurado em russo, indicando que o malware foi compilado em um computador com configurações de idioma russo. Todas as amostras requerem uma senha única de 64 caracteres na linha de comando. Sem a senha correta, o executável não funciona.

Ao ser executado, o EDRKillShifter carrega um recurso criptografado chamado BIN na memória. Ele também cria um arquivo Config.ini com esses dados e o salva no mesmo diretório do binário. O carregador aloca uma nova página de memória com VirtualAlloc e grava o conteúdo criptografado nela. Em seguida, o malware exclui o arquivo config.ini e descriptografa os próximos payloads: um driver vulnerável e um binário Go. A chave de descriptografia dos payloads de segunda camada é um hash SHA256 da senha fornecida.

```
CloseHandle(hFile);
DeleteFileW(L"Config.ini");
pdwDataLen = FileSize.LowPart;
if ( CryptAcquireContextW(&phProv, 0i64, 0i64, 0x18u, 0xF0000000)
  // 0x800C -> CALG_SHA256
  && CryptCreateHash(phProv, 0x800Cu, 0i64, 0, &phHash)
  && CryptHashData(phHash, password, dwDataLen, 0)
  // 0x6610 -> CALG_AES_256
  && CryptDeriveKey(phProv, 0x6610u, phHash, 0, &phKey)
  && CryptDecrypt(phKey, 0i64, 1, 0, (BYTE *)secondLayer, &pdwDataLen) )
{
  CryptReleaseContext(phProv, 0);
  CryptDestroyHash(phHash);
  CryptDestroyKey(phKey);
  return secondLayer;
}
else
{
  free(secondLayer);
```

SophosX-Ops

Figura 2 – Rotina de descriptografia de segunda camada do malware EDRKillShifter.

Caso o malware consiga descriptografar as cargas da segunda camada, ele gera uma nova thread e começa a execução nela.

O segundo estágio utiliza uma técnica de código automodificável para ofuscação. Durante a execução, a segunda camada modifica suas próprias instruções. Como as instruções reais só são reveladas durante a execução, é necessário o uso de ferramentas adicionais ou emulação para análise.

A primeira seção mostra o início do código automodificável. Após a primeira chamada na desmontagem, todas as instruções subsequentes parecem sem sentido. Se revisitarmos o mesmo bloco de instruções após a execução da primeira chamada, veremos um conjunto diferente de instruções. A primeira chamada altera o próximo conjunto de instruções, que por sua vez modifica o conjunto seguinte, e assim sucessivamente.

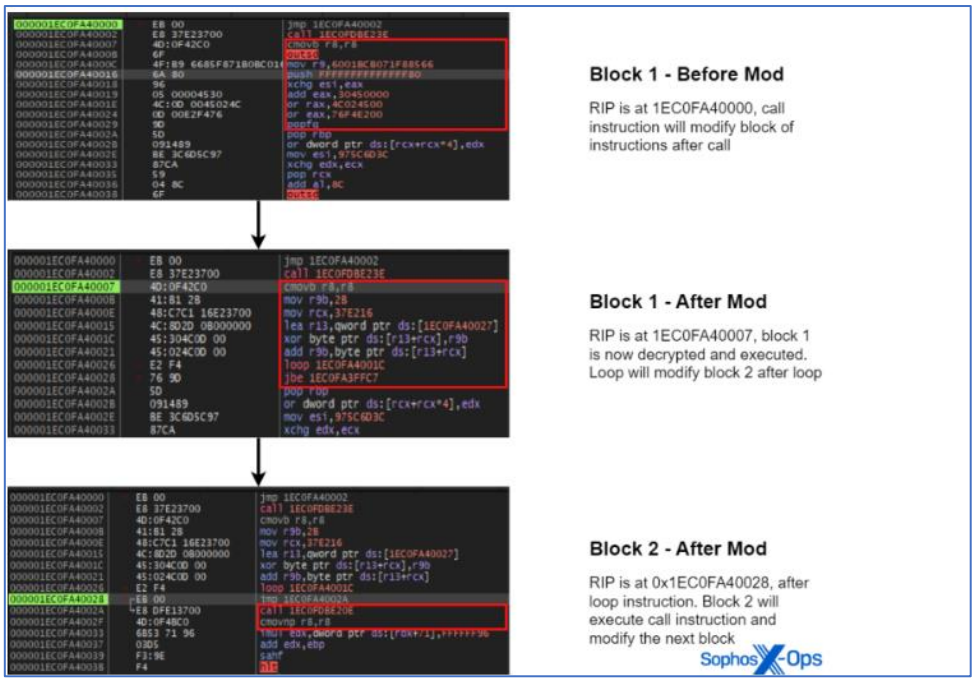


Figura 3 – O código automodificável do EDRKillShifter.

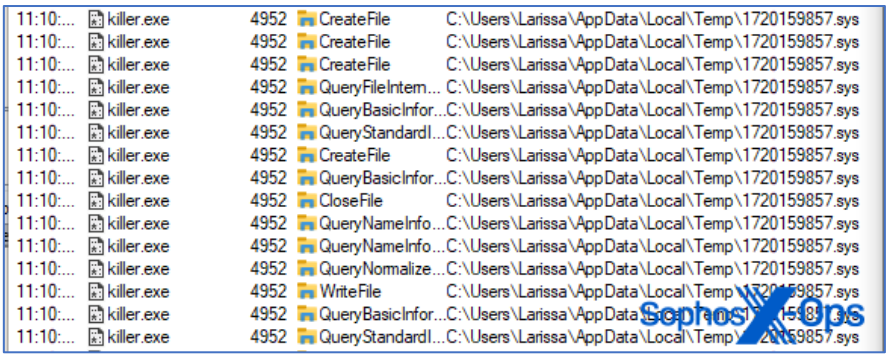
A camada final decodificada tem como único objetivo carregar dinamicamente a carga final na memória e executá-la.

Todas as amostras analisadas executaram diferentes variantes do EDR killer na memória. Escrito em Go e ofuscado (provavelmente com a ferramenta de código aberto gobfuscate), o objetivo dos ofuscadores é dificultar a engenharia reversa. Embora engenheiros de software possam usar ofuscadores para proteger propriedade intelectual, autores de malware os utilizam para complicar a análise de segurança.

Engenheiros reversos geralmente dependem de dados ofuscados ao analisar malware em Go, mas, neste caso, esses dados estão ocultos no código compilado. Exemplos incluem:

- *Strings criptografadas, descriptografadas durante a execução.*
- *Informações da versão Go ausentes, essenciais para ferramentas de engenharia reversa.*
- *Dados úteis de pacotes criptografados ou removidos.*
- *Foi utilizada a ferramenta GoReSym da Mandiant para extrair informações valiosas.*

Todos os EDR killers descompactados incorporam um driver vulnerável na seção .data. Seu comportamento é direto, semelhante a outros EDR killers analisados. A principal diferença entre as variantes é o driver vulnerável carregado e explorado. Após a execução, ambas as variantes adquirem privilégios para carregar um driver e soltar o arquivo sys explorável na pasta \AppData\Local\Temp, gerando um nome de arquivo aleatório a cada execução.



11:10:...	killer.exe	4952	CreateFile	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	CreateFile	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	CreateFile	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	QueryFileIntern...	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	QueryBasicInfor...	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	QueryStandardI...	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	CreateFile	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	QueryBasicInfor...	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	CloseFile	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	QueryNameInfo...	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	QueryNameInfo...	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	QueryNormalize...	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	WriteFile	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	QueryBasicInfor...	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys
11:10:...	killer.exe	4952	QueryStandardI...	C:\Users\Larissa\AppData\Local\Temp\1720159857.sys

Figura 4 – Log do Process Monitor.

Após criar um novo serviço para o driver, o malware inicia o serviço e carrega o driver, entrando em um loop infinito que enumera continuamente os processos em execução. Ele encerra processos cujos nomes estão em uma lista codificada de alvos, comportamento observado em ambas as variantes. Ambas as variantes exploram drivers legítimos, mas vulneráveis, utilizando exploits de prova de conceito disponíveis no Github. Acreditamos que os agentes de ameaças copiaram, modificaram e portaram partes dessas provas de conceito para Go, uma prática comum em outros EDR killers, como o Terminator.

A carga útil final do carregador varia de incidente para incidente e, possivelmente, de criador para criador. No cenário maior de ameaças, é plausível que o carregador e as cargas úteis finais sejam desenvolvidos por diferentes atores de ameaças. A venda de carregadores ou ofuscadores é lucrativa na dark net. A Sophos suspeita que o carregador tem como único propósito implantar a carga útil final do BYOVD, possivelmente adquirida na dark net. As cargas úteis finais do EDR killer são entregues pelo carregador, que consiste nas camadas 1 e 2 descritas.

Figura 5 – Anúncio de ferramenta ofuscadora à venda em um fórum criminoso da dark net.

3 MITRE ATT&CK - TTPs

Tática	Técnica	Detalhes
Execution	T1204.002 User Execution: Malicious File	Execução do EDRKillShifter com senha específica.
Defense Evasion	T1562.001 Impair Defenses: Disable or Modify Tools	EDRKillShifter desativa o EDR carregando um driver vulnerável.
Privilege Escalation	T1068 Exploitation for Privilege Escalation	A ferramenta explora um driver vulnerável para aumentar privilégios e encerrar processos de segurança.

Tabela 1 – Tabela MITRE ATT&CK.

4 RECOMENDAÇÕES

- **Ativar a proteção contra alterações**: Garanta que seu software de segurança de endpoint tenha proteção contra alterações ativada para evitar mudanças não autorizadas.
- **Praticar a boa higiene de segurança**: Limite os privilégios administrativos para reduzir o risco de exploração de drivers.
- **Manter todos os sistemas atualizados**: Aplique atualizações que revoguem certificados de drivers vulneráveis para mitigar o abuso potencial.

Além das recomendações básicas, para se proteger contra ferramentas como o "EDRKillShifter", as organizações devem considerar as seguintes medidas adicionais:

- **Monitoramento contínuo**: Implementar soluções de monitoramento contínuo para detectar comportamentos anômalos que possam indicar uma tentativa de desativação do EDR.
- **Educação e treinamento**: Capacitar funcionários sobre as ameaças modernas e a importância de seguir as melhores práticas de segurança.
- **Resposta rápida a incidentes**: Estabelecer um plano de resposta a incidentes que inclua procedimentos específicos para lidar com tentativas de desativação de proteções.

5 INDICADORES DE COMPROMISSOS

Segue abaixo os indicadores repassados pela Sophos em sua análise.

Indicadores de compromisso do artefato	
sha256:	d0f9eae1776a98c77a6c6d66a3fd32cee7ee6148a7276bc899c1a1376865d9b0
sha256:	451f5aa55eb207e73c5ca53d249b95911d3fad6fe32eee78c58947761336cc60

Tabela 2 – Indicadores de Compromissos de artefatos

6 REFERÊNCIAS

- **Heimdall *by* ISH Tecnologia**
- Sophos
- MITRE ATT&CK
- Thehackernews

7 AUTORES

- **Leonardo Oliveira Silva**
- **Ismael Pereira Rocha**

heimdall
security research
A DIVISION OF ISH