Trinity Cyber

Threat Mitigation and Prevention Case Study

Credential Harvesting Summary

Executive Summary

Credential Harvesting is a technique used by threat actors to gather usernames and passwords, usually through the use of a spoofed website. The goal of this technique is to stockpile as many username/password combinations as possible – to eventually sell or use to compromise organizations at a later time. Since password re-use across multiple sites and applications is common, the successful exfiltration of these credentials could allow unfettered access to corporate networks and sensitive information, as well as further breaches.

Trinity Cyber recently prevented a credential harvesting effort across several of our customer networks. Adversaries were attempting to gather valid Office 365 credentials by using deceptive hyperlinks, email phishing techniques, and the creation of fictitious domains. The unique prevention capabilities of our **TC:Edge** technology recognized the common methodologies leveraged by the adversary within JavaScript code and prevented any credential collection across our customer base by removing the adversary tradecraft from the customer's network traffic, in a fully automated fashion.

Use of Fake Domains

Through Trinity Cyber's investigative research, we identified two distinct domains being used across two similar efforts. The first was widespread and targeted more than 250 users at over 200 unique domains with a set of URLs hosted behind the ymcart[.]cloud domain.

The second attack was more specific, using the vivepipa[.]com domain to target the energy sector with adversary-controlled subdomains (ie. targetedcompany[.]vivepipa[.]com) to trick users into thinking the domain was legitimate. As of this writing, we've identified 25 energy companies and public utilities that the adversary attempted to phish using this technique. In these scenarios, the user was sent to URL that included uniquely named subdomains, as well as email address validation.

The process of generating these phishing URLS appear to follow these steps:

a. First, an email is selected from a list of valid emails the adversary has amassed.
b. The adversary then procedurally generated a domain, on infrastructure they own, that contains company names to mask their true intent.
   For example: targetedcompany2615[.]vivepipa[.]com
c. A parameter is appended to the URL, which is derived from the base64 encoded representation of the user's email address.
   For example: email.victim@targetedcompany.com becomes ZW1haWwudmljdGltQHRhcmdldGVkY29tcGFueS5jb20=
d. The adversary-owned server will now accept GET requests with that parameter and value.
   The URL would look like this:
   targetedcompanyr2615[.]vivepipa[.]com/?e= ZW1haWwudmljdGltQHRhcmdldGVkY29tcGFueS5jb20=

Both efforts utilized the same anti-analysis and Microsoft authentication JavaScript code as well as AtoB base64 encoding techniques, which we will describe below. This leads us to believe it is the same adversary or possibly the use of Phishing as a Service (PaaS).

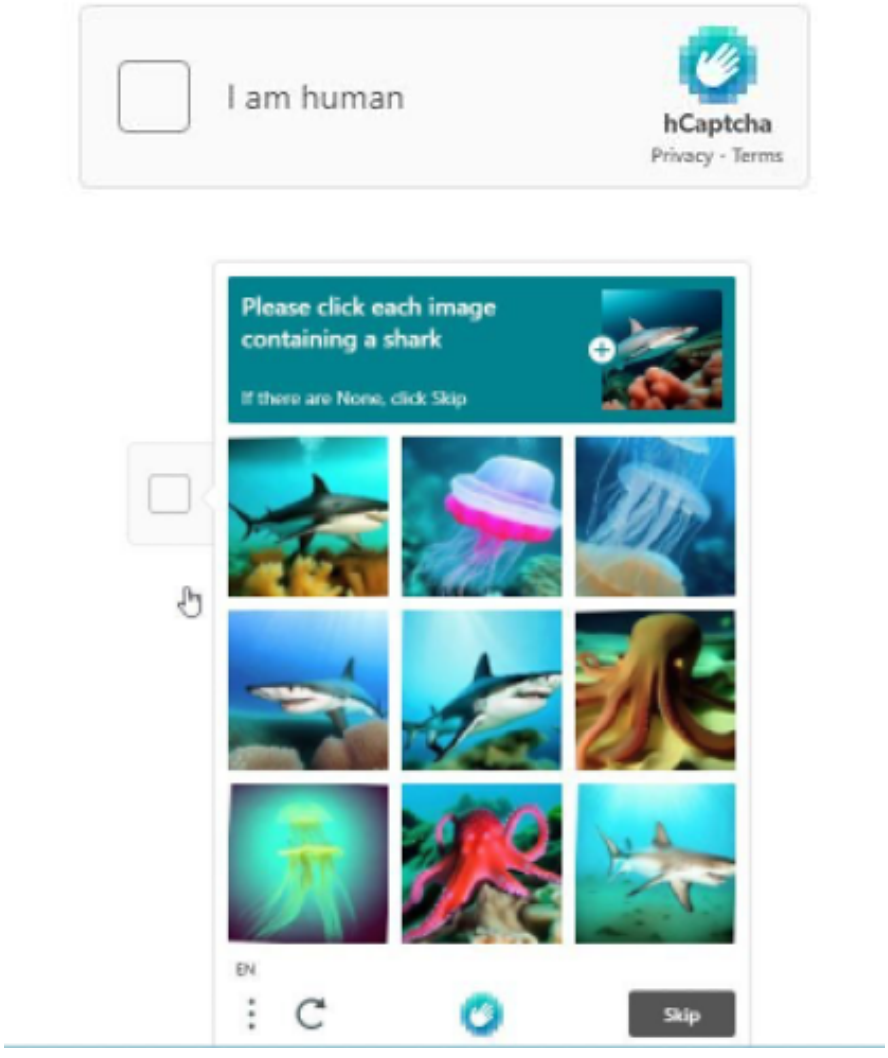
Gartner COOL VENDOR 2020

Harvesting Process Breakdown

1. First, a user would click on a link, most likely contained inside a phishing email, or redirected from a link on another webpage.

2. The browser issues a GET request to the suspect domain using a unique, base64 encoded email address as mentioned above.
   Example:
   GET /?e= ZW1haWwudmljdGltQHRhcmdldGVkY29tcGFueS5jb20=
   HOST: targetedcompany2615[.]vivepipa[.]com

3. User is presented with a seemingly normal captcha:



4. When captcha is completed, the browser is redirected to a spoofed Office365 splash page that contained the following base64 encoded, anti-analysis JavaScript:

```
document.write(atob("PHNjcmlwdD4KICAgIC8vIGRpc2FibGUgcmlnaHQgY2xpY2sKICAgIGRvY
3VtZW50LmFkZEV2ZW50TGlzdGVuZXIoJ2NvbnRleHRtZW51JywgZXZlbnQgPT4gZXZlbnQucHJldmV
udERlZmF1bHQoKSk7CiAKICAgIGRvY3VtZW50Lm9ua2V5ZG93biA9IGZ1bmN0aW9uIChlKSB7CiAKI
CAgICAgICAvLyBkaXNhYmxlIEYxMiBrZXkKICAgICAgICBpZihlLmtleUNvZGUgPT0gMTIzKSB7CiA
gICAgICAgICAgIHJldHVybiBmYWxzZTsKICAgICAgICB9CiAKICAgICAgICAvLyBkaXNhYmxlIEkga
2V5CiAgICAgICAgaWYoZS5jdHJsS2V5ICYmIGUuc2hpZnRLZXkgJiYgZS5rZXlDb2RlID09IDczKXs
KICAgICAgICAgICAgcmV0dXJuIGZhbHNlOwogICAgICAgIH0KIAogICAgICAgIC8vIGRpc2FibGUgS
iBrZXkKICAgICAgICBpZihlLmN0cmxLZXkgJiYgZS5zaGlmdEtleSAmJiBlLmtleUNvZGUgPT0gNzQ
pIHsKICAgICAgICAgICAgcmV0dXJuIGZhbHNlOwogICAgICAgIH0KCiAgICAgICAgLy8gUHJldmVud
CBDdHJsK3MgPSBkaXNhYmxlIHNhdmUKICAgICAgICBpZiAoZXZlbnQuY3RybEtleSAmJiAoZXZlbnQ
ua2V5Q29kZSA9PT0gODUgfHwgZXZlbnQua2V5Q29kZSA9PT0gODMgfHwgZXZlbnQua2V5Q29kZSA9P
T02NSApKSB7CiAgICAgICAgICAgIHJldHVybiBmYWxzZTsKICAgICAgICB9CgogICAgICAgIC8vIGR
pc2FibGUgVSBrZXkKICAgICAgICBpZihlLmN0cmxLZXkgJiYgZS5rZXlDb2RlID09IDg1KSB7CiAgI
CAgICAgICAgIHJldHVybiBmYWxzZTsKICAgICAgICB9CiAgICB9CiAKPC9zY3JpcHQ+"));
```

Decoded the script looks like this:

```
<script>
    // disable right click
    document.addEventListener('contextmenu', event => event.preventDefault());

    document.onkeydown = function (e) {

        // disable F12 key
        if(e.keyCode == 123) {
            return false;
        }

        // disable I key
        if(e.ctrlKey && e.shiftKey && e.keyCode == 73){
            return false;
        }

        // disable J key
        if(e.ctrlKey && e.shiftKey && e.keyCode == 74) {
            return false;
        }

        // Prevent Ctrl+s = disable save
        if (event.ctrlKey && (event.keyCode === 85 || event.keyCode === 83 || event.keyCode ===65 )) {
            return false;
        }

        // disable U key
        if(e.ctrlKey && e.keyCode == 85) {
            return false;
        }
    }

</script>
```

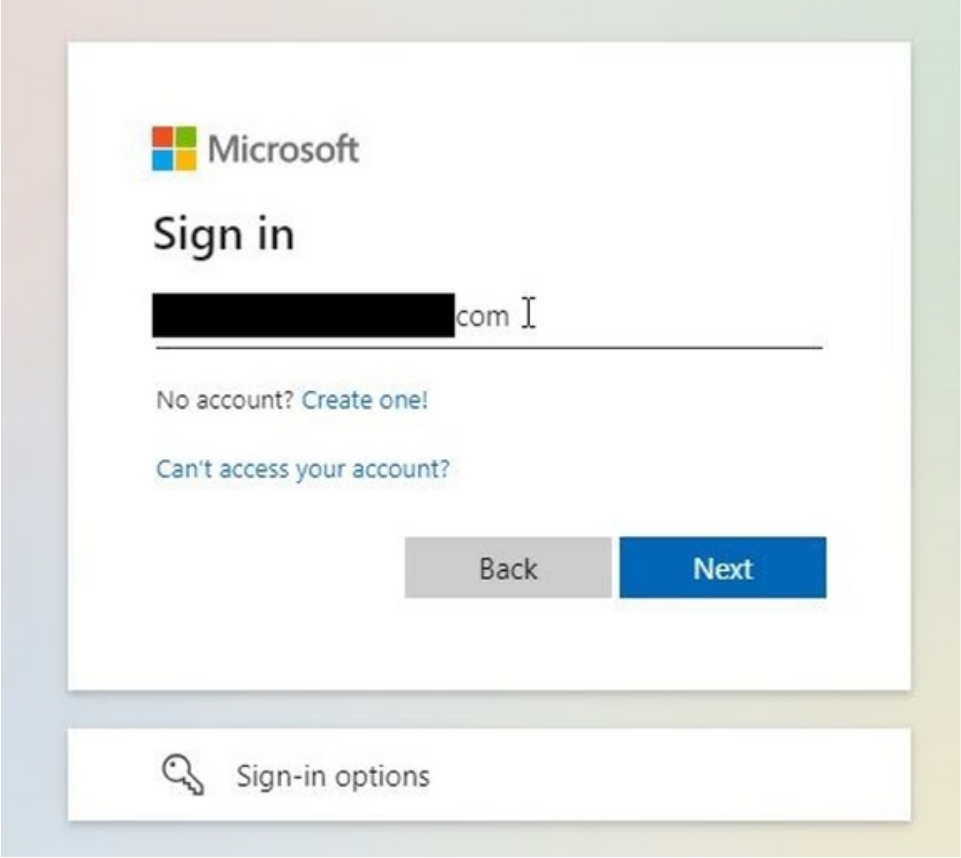
While this script is not overtly malicious, it exists to prevent someone from looking at the page source to determine if the captcha is legitimate. At any point, the user would be able to close their browser without any adverse consequences.

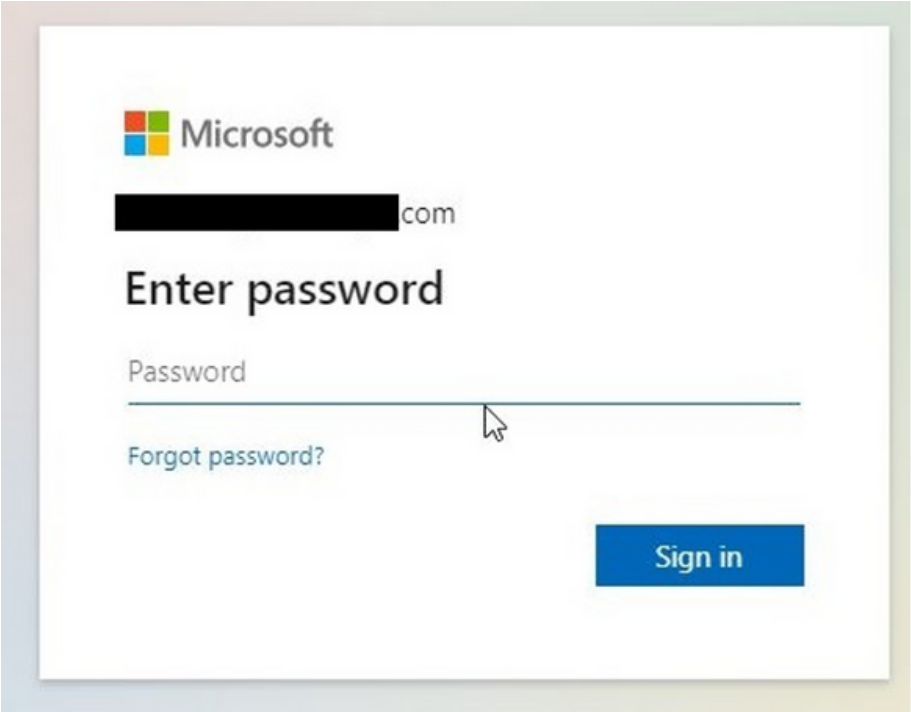
Once the captcha is completed, the user is then redirected to a phony splash page.

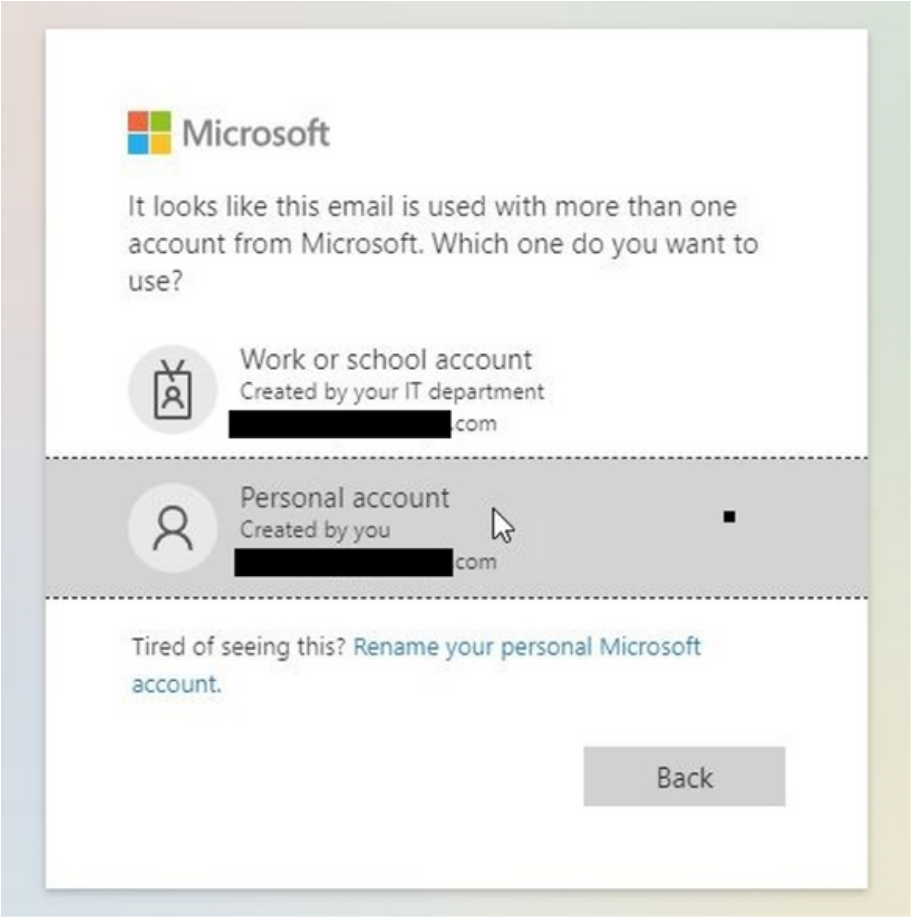
The Trinity Cyber **TC:Edge** technology detects the anti-analysis JavaScript, and not just the fake captcha itself. Our unique, ground-breaking technology also completely removes (automatedly outside our customers' network edge) all the offending JavaScript from the page, editing the network session with precision and preventing any redirection to the fake credential harvesting site. In this way, our service combines a unique and enduring detection capability with an automated threat mitigation action that fire simultaneously, increasing customer security and reducing the incident response workload. No other network security technology operates in this way.

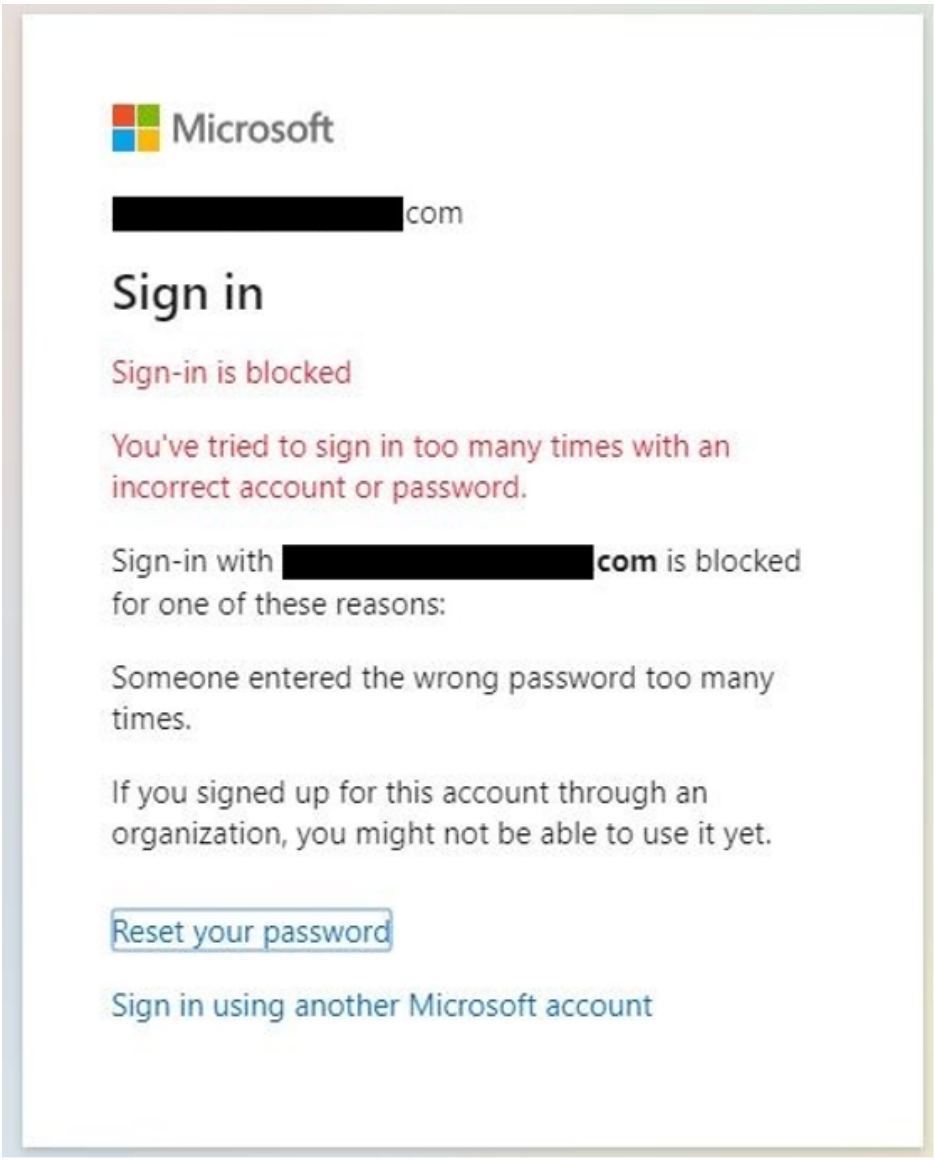
5. The splash page itself resides on a fake website. In the particular event we recently captured, the address was: hxxps://login-azuremicrosoftonline-com-s[.]wempo-peru[.]org

   This page did include JavaScript to submit the credentials to Microsoft Office365 to verify if they are indeed legitimate. Regardless of what the Microsoft-owned server response is, the credentials are still bundled up in a POST form and sent to a PHP file hosted by malicious actors.









6. Once the login request is submitted to Microsoft/Office365 and the credentials are exfiltrated, the user is redirected to the Microsoft owned office.com website. If the user had entered a valid username and password, they would be logged in to the site with no indication anything malicious or out of the ordinary had occurred.

Trinity Cyber has provided protection for attack techniques such as those outlined in this document for all our customers. No action is required on their part to be automatically protected from this and many other tactics that are commonly missed by traditional Intrusion Prevention Systems (IPS) and Secure Web Gateways (SWG). All of this occurs in-line, in real-time, with no latency or impact on the user experience. **TC:Edge** customers can view notifications of our detections and mitigation actions taken via their customer portal or ingest them into their SIEM. For more information, please feel free to reach out to our sales or customer success teams at **sales@trinitycyber.com**