General CCDC Notes – JMU CDC

By: Josh

**Turning off ssh public key authentication:**

1) open to ssh config file, /etc/ssh/sshd\_config

2) find line: “Pubkey Authentication yes” and change to “Pubkey Authentication no”

2b) or, just add the “no” line

3) restart ssh: on RHL: sudo service sshd restart

on Ubuntu: sudo service ssh restart

**Setting a batch script to run as the client when they log in via ssh:**

1) In the ssh config file: “ForceCommand <path to script>”

This script will run as the user – so make sure that the path to the script is writable by all users. This will likely need to be correlated with the firewall logs, as my guess is that the IP that we grab will likely be NAT’ed with a source IP of the firewall. But, we can grab the timestamp, so that will help.

Here is an example of what the script should look like. It will log all of the ssh logins to atuh.log (an intentional typo, or find another name for it that is more likely to not be detected):

echo $SSH\_CLIENT >> /var/log/atuh.log

echo “Logged in at” >> /var/log/atuh.log

echo date >> /var/log/atuh.log

/bin/bash

Then, you need to restart the ssh server. Commands below:

sudo service ssh restart (on Ubuntu)

sudo service sshd restart (on RH)

**Turning on firewall on Windows Server 2003**

Control Panel -> Network Connections (right click) -> Open

This will show your interfaces. Right click on one of them, select properties

From properties menu, click “Internet connection firewall” checkbox, then click “settings” in the bottom right. This will take you to your firewall config tool. This is where you can set which ports you want open and which ones you don’t.

**Searching the filesystem for VBS files (or another filetype)**

The command: dir /b /s C:\\*.vbs

When I ran this on the VM, it had vbs files in: system32, system32/dllcache, and VMware tools. We’re going to want to run this command soon upon logging in to see what we find initially as a baseline for later in the competition.

**Turning off LM Hash**

HKLM\SYSTEM\CurrentControlSet\Control\Lsa “NoLMHash” = 1 (a DWORD)

**Closing common Windows ports (by ending services):**

Port 135 (RPC) – HKLM\Software\Microsoft\OLE “EnableDCOM” = “N”

Then, HKLM\Software\Microsoft\RPC, delete valuesi n “DCOM Protocols” field

Port 445 (SMB) – HKLM\System\CurrentControlset\Services\NetBT\Parameters

Add a DWORD named “SMBDeviceEnabled” with a value of 0

Port 3389 (RDP) – Control Panel \ System – unclick “Allow users to connect remotely” in the “remote” tab.

Then reboot, and the ports will be closed.

**Downloading and using inotify to watch files**

First, download the inotify files - <https://launchpad.net/ubuntu/+source/inotify-tools/3.13-3>

Next, untar the tarball: tar –xvf <name-of-file>

cd into the directory it makes

then ./configure

then make

then sudo make install

Now it works – use the code below to monitor one file

#!/bin/bash

while true

do

inotifywait –e modify </absolutefilepath/tofile>

notify-send “Message” –t <milliseconds for it to be on screen>

echo “Message” >> <outfile>

date >> <outfile>

**IRIS Stuff**

Main files are in /etc/iris

/etc/iris/iris-server.preferences

to get the IRIS client, go to localhost/iris-client

read the README on the Desktop

**SQL Commands**

**Database**

A database consists of one or more tables. A table is identified by its name. A table is made up of columns and rows. Columns contain the column name and data type. Rows contain the records or data for the columns.

**Basic SQL**

Each record has a unique identifier or primary key. SQL, which stands for Structured Query Language, is used to communicate with a database. Through SQL one can create and delete tables. Here are some commands:

* CREATE TABLE - creates a new database table
* ALTER TABLE - alters a database table
* DROP TABLE - deletes a database table
* CREATE INDEX - creates an index (search key)
* DROP INDEX - deletes an index

SQL also has syntax to update, insert, and delete records.

* SELECT - get data from a database table
* UPDATE - change data in a database table
* DELETE - remove data from a database table
* INSERT INTO - insert new data in a database table

**SELECT**

The SELECT is used to query the database and retrieve selected data that match the specific criteria that you specify:  
  
SELECT *column1* [, *column2*, ...]   
FROM *tablename*   
WHERE *condition*   
  
The conditional clause can include these operators

* = Equal
* > Greater than
* < Less than
* >= Greater than or equal
* <= Less than or equal
* <> Not equal to
* LIKE pattern matching operator

SELECT \* FROM *tablename*   
  
returns all the data from the table.

Use single quotes around text values (most database systems will also accept double quotes). Numerical values should not be enclosed in quotes.

LIKE matches a pattern. The wildcard *%* is used to denote 0 or more characters.

* 'A%' : matches all strings that start with *A*
* '%a' : matches all strings that end with *a*
* '%a%' : matches all strings that contain an *a*

**CREATE TABLE**

The CREATE TABLE statement is used to create a new table. The format is:   
  
CREATE TABLE *tablename*   
(*column1* *data type*,   
*column2* *data type*,   
*column3* *data type*);

* char(size): Fixed length character string.
* varchar(size): Variable-length character string. Max size is specified in parenthesis.
* number(size): Number value with a max number of columns specified in parenthesis
* date: Date value
* number(size,d): A number with a maximum number of digits of "size" and a maximum number of "d" digits to the right of the decimal

**INSERT VALUES**

Once a table has been created data can be inserted using INSERT INTO command.  
  
INSERT INTO *tablename*   
(col1, ... , coln)   
VALUES (val1, ... , valn)

**UPDATE**

To change the data values in a pre existing table, the UPDATE command can be used.   
  
UPDATE *tablename*   
SET *colX* = *valX* [, *colY* = *valY*, ...]   
WHERE *condition*

**DELETE**

The DELETE command can be used to remove a record(s) from a table.   
  
DELETE FROM *tablename*   
WHERE *condition*   
  
To delete all the records from a table without deleting the table do   
  
DELETE \* FROM *tablename*

**DROP**

To remove an entire table from the database use the DROP command.  
  
DROP TABLE *tablename*

**ORDER BY**

ORDER BY clause can order column name in either ascending (ASC) or descending (DESC) order.  
  
ORDER BY *col\_name* ASC

**AND / OR**

AND and OR can join two or more conditions in a WHERE clause. AND will return data when all the conditions are true. OR will return data when any one of the conditions is true.

**IN**

IN operator is used when you know the exact value you want to return for at least one of the columns   
  
SELECT \* FROM *table\_name* WHERE *col\_name* IN (*val1*, *val2*, ...)

**BETWEEN / AND**

The BETWEEN ... AND operator selects a range of data between two values. These values can be numbers, text, or dates.   
  
SELECT \* FROM *table\_name* WHERE *col\_name* BETWEEN *val1* AND *val2*

**JOIN**

There are times when we need to collate data from two or more tables. That is called a **join**. Tables in a database are related to each other through their keys. We can associate data in various tables without repeating them. For example we could have a table called *Customers* which could have information about customers like their name, address, phone numbers. We could have another table called *Products* that has information regarding the products like part number, product name, manufacturer, number in stock, unit price. A third table called *Orders* could have information regarding what product was ordered, by whom, the date the order was placed, and quantity. Here are the tables:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Customers | | | | |
| **Cust\_ID** | **FirstName** | **LastName** | **Address** | **Phone** |
| 01 | Mickey | Mouse | 123 Gouda St. | 456-7890 |
| 02 | Donald | Duck | 325 Eider Ln. | 786-2365 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Products | | | | |
| **Part\_No** | **Name** | **Manufacturer** | **In\_Stock** | **Price** |
| 20-45 | Hammer | Stanley | 57 | 3.50 |
| 21-68 | ScrewDriver | DeVries | 84 | 2.75 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Orders | | | | |
| **Order\_No** | **Part\_No** | **Cust\_ID** | **Date** | **Quantity** |
| 2005-27 | 21-68 | 02 | 31 Oct 2005 | 2 |
| 2005-34 | 20-45 | 01 | 02 Nov 2005 | 3 |

We can obtain information on who has ordered what:   
  
SELECT Customers.FirstName, Customers.LastName, Products.Name   
FROM Customers, Products, Orders   
WHERE Customers.Cust\_ID = Orders.Cust\_ID AND Products.Part\_No = Orders.Part\_No

We can select data from two tables with INNER JOIN. The INNER JOIN returns all rows from both tables where there is a match. If there are rows in *Customers* that do not have matches in *Orders*, those rows will not be listed.   
SELECT Customers.FirstName, Customers.LastName, Orders.Date   
FROM Customers   
INNER JOIN Orders   
ON Customers.Cust\_ID = Orders.Cust\_ID

The LEFT JOIN returns all the rows from the first table (Customers), even if there are no matches in the second table (Orders). If there are rows in *Customers* that do not have matches in *Orders*, those rows also will be listed.   
SELECT Customers.FirstName, Customers.LastName, Orders.Date   
FROM Customers   
LEFT JOIN Orders   
ON Customers.Cust\_ID = Orders.Cust\_ID

The RIGHT JOIN returns all the rows from the second table (Orders), even if there are no matches in the first table (Customers). If there had been any rows in *Orders* that did not have matches *Customers*, those rows also would have been listed.   
SELECT Customers.FirstName, Customers.LastName, Orders.Date   
FROM Customers   
RIGHT JOIN Orders   
ON Customers.Cust\_ID = Orders.Cust\_ID

**ALTER TABLE**

With ALTER TABLE you can add or delete columns in an existing table. When you add a column you must specify a data type.   
ALTER TABLE table\_name   
ADD col\_name datatype   
  
ALTER TABLE table\_name   
DROP COLUMN col\_name

**UNION**

The UNION command is used to select data from two tables very similar to the JOIN command. But the UNION command can be used only with columns having the same datatype. With UNION only distinct values are selected, i.e. if there are common data in the two tables only one instance of that data is returned.   
  
SELECT Name FROM Customers\_USA   
UNION   
SELECT Name FROM Customers\_Asia   
  
This will select all the customers from USA and Asia but if there is a name that occurs in both the tables it will return only one such name. To get all the names use UNION ALL instead.

**SQL Functions**

There are several built-in functins in SQL. The basic function types are:

* Aggregate Functions: These are functions that operate against a collection of values, but return a single value.
* Scalar Functions: These functions operate against a single value, and return a single value.

To use a built-in function the syntax is:   
  
SELECT function (col\_name) FROM table\_name 

**GROUP BY**

The GROUP BY was added to SQL so that aggregate functions could return a result grouped by column values.   
  
SELECT col\_name, function (col\_name) FROM table\_name GROUP BY col\_name

HAVING keyword was introduced because the WHERE keyword could not be used. HAVING states a condition.   
  
SELECT clo\_name, function (col\_name) FROM table\_name   
GROUP BY col\_name   
HAVING function (col\_name) condition value

**CREATE VIEW**

A view is a virtual table that is a result of SQL SELECT statement. A view contains fields from one or more real tables in the database. This virtual table can then be queried as if it were a real table.   
  
CREATE VIEW view\_name AS   
SELECT col\_name(s)   
FROM table\_name   
WHERE condition   
  
A view could be used from inside a query, a stored procedure, or from inside another view. You can add functions and joins to a view and present the data you want to the user.