Introduction:

For our project, we have explored 4 research questions relating to the online hacker community. In addition to the formulation of these questions, we have generated a list of hypotheses. By collecting data from Shodan and HackerWeb, we were able to analyze data and product information relevant to topic questions. Furthermore we have concluded our findings and performed additional research of similar topics through various websites.

Research Design:

Analytic Methods:

Shodan Maps were used to map various devices geographically. We used Hadoop for text mining and found interesting patterns among text. For visualization we used IBM’s BigInsights and Hadoop to organize data in structured format and display this information in the form of graphs and pie charts. Furthermore, Microsoft Excel was used in various areas of the project for graph creation.

Collection Methods:

Using SQL queries for HeidiSQL and MySQL, we were able to extract data from HackerWeb. We sifted through threads from various forums such as the Arhack forum and Anon forum. After identifying relevant threads, we dug deeper into content by extracting relevant posts and exported these tables to Excel in order to perform a statistical analysis of data.

Shodan was used to find devices connected to internet. We used Shodan API to find results. The programs were written using Python. We wrote programs to find results like: kinds of exploits, unprotected devices, which ports are exposed etc.

Research Question 1: What vulnerabilities are hackers taking advantage of in social networking? What techniques are being used in regards to hacking social networking sites? What incentives are encouraging hackers to hack social networking sites?

Based on research question #1, we have developed the following hypotheses:

a. We think that by detecting keystrokes, hackers are able to figure out user’s password. We believe vulnerabilities may include security holes which allow hackers to identify these keystrokes.

b. We believe the techniques being used for hacking include social engineering. For example, in Facebook tagging friends with weak privacy settings may lead to an opportunity for hacking. Other social engineering methods may include phishing emails.

c. Incentives may be financial, political, or to prove a point.

Literature Review:

The subject of social networking sites’ vulnerabilities and techniques have been explored by various websites and articles. Some examples of vulnerabilities include cross-site scripting. With cross-site scripting, attackers write dynamic HTML that sends cookies to a web server.
Many websites use a cookie flag called HTTP only, a feature introduced in web browsers. This feature is limited, as the attacker can inject Java script code on the website, instead of having to rely on stealing the cookie. *(Acunetix, 2014)*

According to the FBI website, a combination of computer savvy hackers and social engineers work together to compromise social networking users’ accounts. The FBI website goes on to describe how humans tend to be the “weak link in cyber security.” Some techniques discussed on the website include:

- Baiting
- Click-jacking
- Elicitation
- Pharming
- Doxing
- Phishing
- Spoofing

*(FBI Website, 2014)*

**Findings & discussions:**

In respects to this question, focus was given to the Arhack forum. This forum is primarily in Arabic, and posts and access was made primarily in the areas of the Middle East. Out of the 243 threads and 124,340 views the top three social networking sites of interest includes Facebook, Twitter, and Skype. *(Refer to Appendix 1.0 Social Networking Distribution)*

Of the social networking sites discussed in the forum, not all posts are hacker related. Some of these topics included general news found on these sites, while others threads, such as Skype related threads, focused on selling accounts. The percentage of hack related posts per social networking site can be found in the Appendix. *(Refer to Appendix 1.1 Hacker Related Posts)& (Refer to Appendix 1.2 Hacker Related Posts per Site)*

Vulnerabilities in which hackers are taking advantage of include user unawareness and security holes. With regards to techniques being used to take advantage of these vulnerabilities, the main method is social engineering. By taking advantage of users’ poor privacy settings, hackers are able to find users’ email addresses. Hackers then proceed to send phishing emails with fake links that give users’ information to hackers on a silver platter.

In regards to incentives, it is difficult to pinpoint what hackers are interested in achieving by hacking into the various social networking sites. By taking a closer look at the dates of hacks, some assumptions can be made. For posts between the years 2011 and 2013, there is a huge possibility that these hacks could be for political reasons. Social networking played a key role in the Arab Spring, and a strong presence of cyber warfare existed and continues to exist. In addition, based on the dialect used, this theory is better backed, as it seems hackers were of Syrian and Egyptian nationality. These two countries heavily used and continue to use social networking.
Research Question 2: What types of gambling are taking place? Which types of gambling are more prevalent?

Based on research question #2, we have developed the following hypotheses:

a. We anticipate that illegal gambling is the most prominent form.
b. We think Facebook Poker will be the most popular.

Literature Review:

People gambling online are always interested in making a little extra money and so in order to do that they surf the web to get any cheat codes they can. Hackers take advantage of such people by asking them to download a “cool” program designed to cheat online poker rooms or hack online casinos which actually contain malwares and viruses intended to send your personal financial data to the author of the program.

According to a famous news website, www.dailymail.co.uk an IT expert siphoned off huge volumes of gaming chips and then transferred the electronic chips in fake Facebook accounts before selling them online at reduced prices.

An article by McMullan and Perrier D throws light at some of the techniques used by hackers. The report says that hackers use a computer virus to install ‘back door’ programs (aka ‘zombies’) on networks of personal computers (PCs) that they can trigger without the PC owner’s knowledge. Networks of zombies are subsequently used to take over targeted internet sites by denying access to real consumers.

Findings & discussions:

HackerWeb:

After extracting conversations from the Arhack forum, we found that Zynga and Texas Holdem Poker seems to be the most discussed gambling site. In fact, the only additional gambling site that was discussed aside from Facebook Poker was the www.sportgamblingpicks.com website.

With 3485 views and 19 posts thread 19869 had the most views and posts out of all other gambling related threads. This thread was focused on increasing poker chips by downloading Cheat Engine 5.6. The increase in poker chips is something that would not show up in the database. (Refer to Appendix 2.0 Thread 19869)

Other discussions were based on the sale of poker chips. For example thread 77957 with 734 views, was focused on selling Zynga Poker Chips. (Refer to Appendix 2.1 Thread 77957)

Shodan:

To further perform a detailed analysis of this research question, we used the Shodan tool. We extracted server with port 1025. Port 1025 is usually associated with gambling sites. In order to perform this procedure, we ran a script that can be found in the appendix. (Refer to Appendix 2.2 Script Port 1025). There were 39 results that appeared after running the script. The number of port 1025 increased 2-3 months. Most of them were in US.
There are many gambling servers around the world like: online-gambling-city.com, online-
gambling-poker.net but most of them are protected which can be found out by running the script
in appendix (Refer to Appendix 2.3 Script Gambling Servers). So, we can say that gambling servers
are generally protected to a good level by multi-level authentication. The gambling server were
of different types in method of file exchange. They were both FTP and HTTP servers.

**Research Question 3: How are online games being hacked? What are the incentives for hacking games?**

Based on research question #3, we have developed the following hypotheses:

- a. We think people are sharing high level C++/C codes to run the scripts on the different
games and win the game.
- b. We think that online gaming that involves prize money are more prone to hacking. We
think that these hacks are happening to avoid paying fees for the games.

**Literature Review:**

Millions of transactions take place over the internet each day and criminal organizations are
taking advantage of this fact to launder illegally acquired funds through covert, anonymous
online transactions. According to one of the news reports on www.wired.co.uk some of these
games use credits that players can exchange for real money. Through this system criminals can
send virtual money to any part of the globe which could then be converted to real money.

The report went on to cite that vulnerabilities through sites like PayPal make it easier for hackers
to succeed in Micro-laundering. This is a technique which involves moving a large amount of
money in small amounts through thousands of electronic transactions using mobile banking
systems.

Furthermore, a Yahoo news article back in May 28th, 2013 reported that 7 hackers were charged
in a $6 billion money laundering case.

MIT Technology Review reports a large number of cases concerning criminals transferring large
amounts of money into accounts and then asking unaware receivers to forward it. Interestingly
enough, since stolen funds are being laundered, unaware senders and receivers are held
accountable for the crime. Social media and other networking sites are examples of such
criminal activities using which hackers take advantage of individuals with little to no knowledge
of online security. People receiving money tend not to inform the police or investigating
agencies as feel they too could make money out of it, not realizing that they have been roped into
a serious crime.

**Findings & discussions:**

Based on the research done by our team, we analyzed the anonposts, hackhound and icodeposts
which generally talked about the different online games and the ways to find the cheat-code for
the particular game. Here is our analysis from these various threads:
Anonposts: There is one post with threadid ‘77’ and postID 136 which discuss the Cheat-Engine v2.0 and how we can use this cheat engine to produce cheats on the online games. The author has made the Cheat-Engine himself and provided clear steps on how to use the cheat-engine to hack some of the online games website (Refer to Appendix 3.0 PostId:136). Other threads included details of how many sites of online games exists and their links.

HackHoundPosts: In the hackhound posts, people mainly discussed the ways in which one can download online games (paid) for free. Cyberghost has a free and paid service. So, one can easily use this service and download games. (Refer to Appendix 3.1 PostId: 11735)

Icodeposts: In the Icodeposts, people have shared some C++/C codes to hack the online games. For instance, postid 54066 discussed about how to use Delphi and Opcode for hacking purpose. Apart from Delphi and Opcode, Combat Arms (hacksfield) found some of the DirectX function suitable for avoiding displaying the in-game hack menu. The other posts just described how you can use C++ codes to hack the games. (Refer to Appendix 3.2 PostId: 54066)

According to the data being collected, we have analyzed that the maximum number of hacks/cheats is being conducted through writing C/C++ codes. The icodeposts, anonposts and hackhound posts has a number of posts related to hacking the online games so that people can play the online games without paying for them.

Research Question 4: What types of attacks are being used the most? For what purposes are these attacks? Where are the source of attacks showing up the most?

Based on research question #4, we have developed the following hypotheses:

a. We think DoS is the most common form of attack. We anticipate that SQL Injection will be another common form.

b. We believe incentives may include financial, as well as publicity purposes.

c. We predict that most of these attacks occur in the USA.

Literature Review:

While we found many articles relating to various attacks, we found that many of them focused on DoS/DDoS attacks. For example, the CIS journal published in October 2011 gives detailed information regarding DOS and DDOS attacks. The journal also talks about the architecture the hackers follow to execute such attacks. Additionally, the journal gives information regarding the background of botnet based DDOS attacks and how they hamper webservers.

The journal cites that the motive of such attacks is to degrade the performance of the server at the victim site. These attacks could be performed by a single user or an organization, such as competitors.

China Realtime, a Chinese online news, reported a similar attack on a part of the Chinese Internet when it went down on August 26, 2013. The attack, which was aimed at the registry that allows users to access sites with the extension “.cn”, intensified after a while as reported by China Internet Network Information Center. CloudFlare Chief Executive, Matthew Prince. Mr.
Prince said the company observed a 32% drop in traffic for the thousands of Chinese domains on the company’s network during the attack compared with the same time 24 hours earlier.

Findings & discussions:

Using Shodan we pulled up information in regards to the four following types of attacks:

1- SQL Injection
2- DoS
3- DDoS
4- Spoofing
5- Spam
6- Brute Force
7- Zero Day

As seen in the bar graph in the Appendix, the top three attacks with the most inquiries include SQL Injection (12,236), DoS (4,307), and Spoofing (638). (Refer to Appendix 4.0 Attacks Bar Graph)

Using the script found in the appendix (Refer to Appendix 4.1 Attacks Script) information in regards to the specified attacks are pulled up. The script pulls out information such as the source of the attack, the description, the date and in some cases the author. (Refer to Appendix 4.2 Output of Script)

Taking a look at the appendix (Refer to Appendix 4.3 Source Map) a map is used to show where the source of these types of attacks are. As seen in the map many of these attacks come from various parts of the world. Mainly the source of these attacks come from the US, parts of Europe and Southeast Asia.

Most of the DDoS and DoS attacks were carried out by exploiting port 0 and the sources of the most of the attacks were ExploitDB.

We also analyzed the result on the basis of top 5 unprotected devices i.e. the devices using default password. Results indicate that Print Servers top the list followed by routers. It is kind of surprising because we though print servers to be protected. The script takes input “Default Password” to execute. (Refer to Appendix 4.2 Output of Script)
References:


Appendix:

(Appendix 1.0 Social Networking Distribution)

<table>
<thead>
<tr>
<th>Social Networking Topics</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td>66.89%</td>
</tr>
<tr>
<td>Twitter</td>
<td>4.30%</td>
</tr>
<tr>
<td>Skype</td>
<td>28.72%</td>
</tr>
<tr>
<td>LinkedIn</td>
<td>0.05%</td>
</tr>
<tr>
<td>Flickr</td>
<td>0.03%</td>
</tr>
</tbody>
</table>

(Appendix 1.1 Overall Hacker Related Posts)
(Appendix 1.2 Hacker Related Posts per Site)

<table>
<thead>
<tr>
<th>Hacker Related Posts</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td>49.49%</td>
</tr>
<tr>
<td>Twitter</td>
<td>10.74%</td>
</tr>
<tr>
<td>Skype</td>
<td>14.08%</td>
</tr>
<tr>
<td>LinkedIn</td>
<td>100.00%</td>
</tr>
<tr>
<td>Flickr</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

(Appendix 2.0 Thread 19869)

<table>
<thead>
<tr>
<th>threadID</th>
<th>title</th>
<th>numOfView</th>
<th>numOfPosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>19869</td>
<td>تعال عدل على الالتوس (Texas Hold Em Poker)</td>
<td>3485</td>
<td>19</td>
</tr>
<tr>
<td>21805</td>
<td>[مروره والتباع على ZYNGA POKER] اعمال شركه</td>
<td>1130</td>
<td>12</td>
</tr>
<tr>
<td>77456</td>
<td>مطلوب رقمي بوكر للقتات Poker Texas Hold'em poker</td>
<td>129</td>
<td>3</td>
</tr>
<tr>
<td>77527</td>
<td>مطلوب رقمي بوكر للقتات Poker Texas Hold'em poker</td>
<td>108</td>
<td>1</td>
</tr>
<tr>
<td>77531</td>
<td>مطلوب رقمي بوكر للقتات Poker Texas Hold'em poker</td>
<td>217</td>
<td>6</td>
</tr>
<tr>
<td>77957</td>
<td>Selling Zynga Poker Chips</td>
<td>724</td>
<td>17</td>
</tr>
<tr>
<td>79957</td>
<td>مطلوب شراء في Poker Texas HoldEm Poker</td>
<td>210</td>
<td>0</td>
</tr>
</tbody>
</table>
(Appendix 2.1 Thread 77957)

<table>
<thead>
<tr>
<th>threadID</th>
<th>title</th>
<th>numOfView</th>
<th>numOfPosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>19869</td>
<td>عالم عدّل على الفلوس بالموزّر (Texas Hold Em Poker)</td>
<td>3485</td>
<td>19</td>
</tr>
<tr>
<td>21805</td>
<td>موزّر والبلد على 2010 متجمّع [ZYNGA POKER</td>
<td>1130</td>
<td>12</td>
</tr>
<tr>
<td>77456</td>
<td>مطورات رصيد بوكر Texas Hold'em poker</td>
<td>129</td>
<td>3</td>
</tr>
<tr>
<td>77527</td>
<td>مطورات رصيد بوكر Texas Hold'em poker</td>
<td>108</td>
<td>1</td>
</tr>
<tr>
<td>77531</td>
<td>مطورات رصيد بوكر Texas Hold'em poker</td>
<td>217</td>
<td>6</td>
</tr>
<tr>
<td>77957</td>
<td>Selling Zynga Poker Chips</td>
<td>734</td>
<td>17</td>
</tr>
<tr>
<td>79957</td>
<td>مطورات الفيديو في Texas HoldEm Poker</td>
<td>210</td>
<td>0</td>
</tr>
</tbody>
</table>

(Appendix 2.2 Script Port 1025)
Code:
## to find servers with port 1025
from shodan import WebAPI

SHODAN_API_KEY = "72WhTJ2f0PIhtepGEAdaF5Qvevlv8znxB"

api = WebAPI(SHODAN_API_KEY)

# This example search a specific keyword in Shodan database, and print the results.
try:
    # Search Shodan
    results = api.search('port=1025')
    # Show the results
    print 'Results found: %s' % results['total']
    for result in results['matches']:
        print 'IP: %s' % result['ip']
        print result['data']
        print ''
except Exception, e:
    print 'Error: %s' % e

(Appendix 2.3 Script Gambling Servers)

### gambling
# servers in gambling
from shodan import WebAPI

SHODAN_API_KEY = "72WhTJ2f0PIhtepGEAdaF5Qvevlv8znxB"

api = WebAPI(SHODAN_API_KEY)

# This example search a specific keyword in Shodan database, and print the results.
try:
    # Search Shodan
    results = api.search('gambling')
    # Show the results
    print 'Results found: %s' % results['total']
    for result in results['matches']:
        print 'IP: %s' % result['ip']
        print result['data']
        print ''
except Exception, e:
    print 'Error: %s' % e

(Appendix 3.0 Thread Anonthread)
Appendix 3.1 Thread Hackhoundthread

Cyberghost has a free and paid service. The free service is good... I have paid the service if you can afford it, I would suggest you purchase it. With CyberGhost VPN you re-establish your online privacy and internet freedom with just a few clicks. No exhausting searches for new, overloaded and in the end slow web proxies anymore and no crazy and complicated browser settings needed. And the best thing: you're not limited to your browser to use the web anonymously. You can also use any other program on your PC with access to the internet, like your messenger, Skype, WhatsApp, your downloads clients, and games (Dota, CS:GO, etc.), homepages: http://cyberghost.com...ur- anonymity.html

Appendix 3.2 Postid: 54066

<table>
<thead>
<tr>
<th>opeplc is correct, while C++ is commonly used for game hacking when dealing with Direct3D games, Delphi is just as good. To avoid hardcoding, here are a few &quot;hacks&quot; that can be used depending on...</th>
</tr>
</thead>
</table>
| Originally Potted by Pauleigames: some call assembly not really getting what you mean here (but I see simple apps like open process you find hard to master maybe you should learn more Delphi) and then | Hi, I'd be happy to be here. I've just discovered the forum today (I come from Hacktitude) in programming C++, but I'm not sure if I like it. I was coding with Win32 API and I'm actually coding a c++/c++ | I'm not sure if I'm just trying to get a better understanding of the Direct3D function for avoiding detection and displaying the in-game hack menu. I'm not sure, but I think you can't do this for looking for something to play on your server (you don't need 200kG in exchange for a diamond bot). The moment I don't have any bots for the first trade ask you for your trustname y= (C++: 20000kmp 8:68k 2:TN, you get your trustname y= C++: 20000kmp 8:68k 2:TN, you get your...)

Appendix 4.0 Attacks Bar Graph

If you ever played simple 3D flash games or similar and struggled with them, you can now make them easier by &quot;hacking&quot; and the way you do that, is by using this tool called Cheat Engine (v6.6). This is one of the easiest ways to &quot;hacks&quot; those kind of games. Cheat Engine works like this: Select the website of the flash game you are trying to hack in the top left corner (computer icon). Find the value of something e.g. health, money, experience etc. Let say you have 300 coins on a game. 3. Buy something in the game if possible or gain more coins. 4. Input the new value in the cheat engine so that less answers come up for you are looking for the one you are using. 5. Restart step 2 and in 4 hours no more than 5 answers, preferably 1 to 3 answers. Right click the value and change it (see the tutorial below) to the value you want, e.g. you have 300 coins, so type in 1000000 and then the box on the left of the value to freeze it so that the value never changes even if you buy something. There are lots of ways to use this and the one I told you was just the basic one. You can find the tutorial on Youtube, this one is a simple one (not made by S lamp): http://www.youtube.com/watch?v=7FrB0PVt772

3. If you ever played simple 3D flash games or similar and struggled with them, you can now make them easier by &quot;hacking&quot; and the way you do that, is by using this tool called Cheat Engine (v6.6)
(Appendix 4.1 Attacks Script)
import shodan
import sys

API_KEY = '72WhTJ2fOPhtepGEdaF5Qvev1v8znxB'

# Configuration

# Input validation
if len(sys.argv) == 1:
    print 'Usage: %s <search query>' % sys.argv[0]
    sys.exit(1)

try:
    # Setup the api
    api = shodan.Shodan(API_KEY)
    # Generate a query string out of the command-line arguments
    query = ' '.join(sys.argv[1:])
    # Use the count() method because it doesn't return results and doesn't require a paid
    # API plan
    # And it also runs faster than doing a search().
    result = api.Exploits.search(api.exploits,query,page=6)
    print 'Shodan Summary Information'
    print 'Query: %s' % query
    print 'Total Results: %s' % result['total']

    # writes result to file
    results1 = open('myname13.txt', 'a')
    i= 0
    # Print the summary info from the facets
    for results in result['matches']:
        i = i +1
        if (sys.argv[1] == 'DoS'):
            print >>results1, '%d <code> %s <description> %s <author>
            %s' %(i, results['code'], results['description'],results['author'])
            # print out the result on console as well
            print '<code> %s ,' %results['code']
            print '<description> %s ,' %results['description']
            print '</source>'
            print 'port %s' %results['port']
            print 'type %s' %results['type']
            print ''
        # Print an empty line between summary info
        elif (sys.argv[1] == 'SQL Injection'):
            print >>results1, '%d <ID> %s <description> %s <source>
            %s' %(i, results['id'], results['description'],results['source'])
        # Print an empty line between summary info
        elif (sys.argv[1] == 'DDoS'):
            print >>results1, '%d <description> %s' %(i, results['description'])
        except Exception, e:
            print 'Error: %s' % e
            sys.exit(1)
    results1.close();

(Appendix 4.2 Outputs of script)
import socket
import sys
from time import sleep

buff = "" * 1000
print("[*] payload sent "+ str(len(buff)))
exp1 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
exp1.connect(('10.0.0.98', 8080))
ex1.send(f"HTTP/1.1\r\n\host: 192.168.1.10:20\r\n\user-agent: Mozilla/4.0 (Linux 2.6.21.3) java/1.5.0_02\r\n\r\n")
data = exp1.recv(1024)
print data
exp1.close()

# m1lworm.com [2009-04-02]
<description> Novel directory HTTP Denial of Service Exploit <author> muts

import socket
import sys
print "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
print "" "" "" "" "" "" "" "" "" "" "" ""
print "" "" "" "" "" "" "" "" "" "" "" ""
host = "127.0.0.1"
port = 80
try:
    buf = "/\"\" * 1000
    request = "GET / HTTP/1.0"
    connection = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    connection.connect((host, port))
    connection.send(request)
    raw_input("\nexploit completed, press "enter" to quit..."
) #sys.exit()
except:
    raw_input("\nunable to connect, press "enter" to quit...
"

# m1lworm.com [2009-07-01]
<description> A90-9908 DVR Card Security Camera (GET Request) Remote DoS Exploit <author> Stack

(Appendix 4.3 Source Map)

(Appendix 4.3 Source Map)

#default password webcams
import shodan
import sys

# Configuration
API_KEY = '72WhTJ2fOPIhtepGEdaF5QvevLv8zmxB'
# The list of properties we want summary information on
FACETS = [
    'device',
    # We only care about the top 5 countries, this is how we let Shodan know to return 5
    instead of the

(Appendix 4.3 Source Map)
# default 10 for a facet. If you want to see more than 10, you could do ('country', 1000) for example
# to see the top 1,000 countries for a search query.

FACET_TITLES = {
    'device':'top 5 devices'
}

# Input validation

try:
    # Setup the api
    api = shodan.Shodan(API_KEY)
    # Generate a query string out of the command-line arguments
    query = ' '.join(sys.argv[1:2])
    # Use the count() method because it doesn't return results and doesn't require a paid
    # API plan
    # And it also runs faster than doing a search().
    result = api.search(query, facets=FACETS)
    # Print the summary info from the facets
    for facet in result['facets']:
        print FACET_TITLES[facet]
        for term in result['facets'][facet]:
            print '%%s: %%s' % (term['value'], term['count'])
    # Print an empty line between summary info
    print ''
except Exception, e:
    print 'Error: %%s' % e
sys.exit(1)