Name : Benjamin Xavier Tay Cheng Lim

ID : 6740510

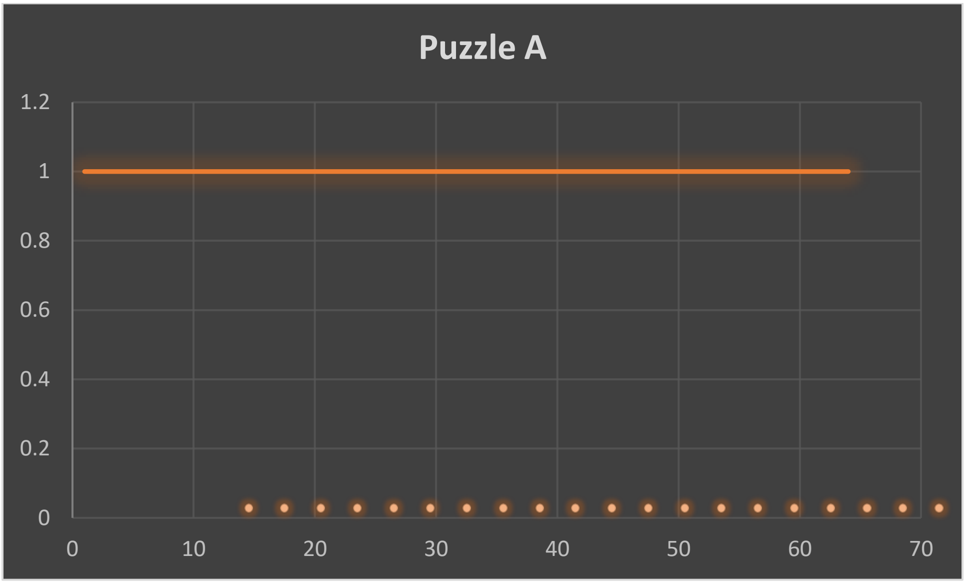
1.

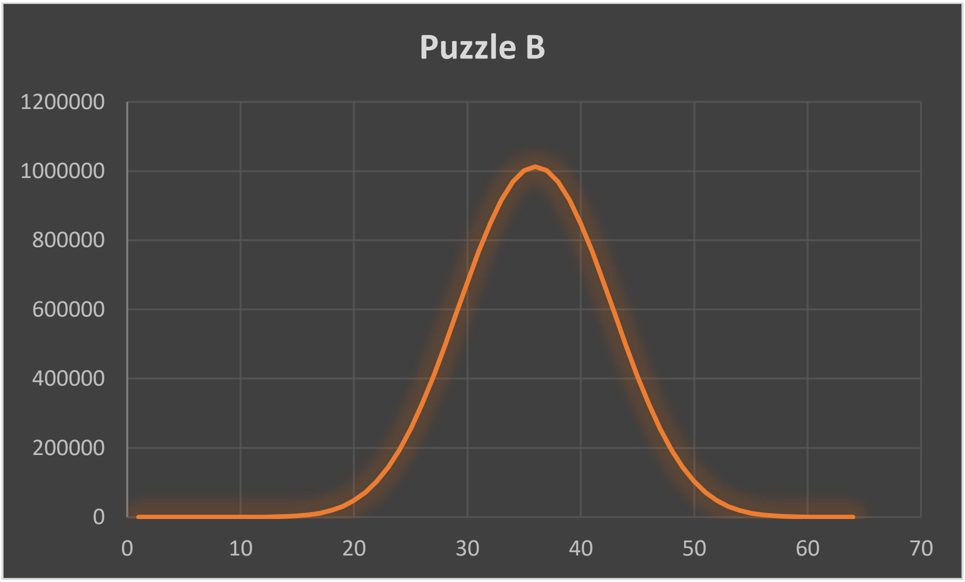
a)

hashes needed	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Puzzle A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Puzzle B	0	0	0	0	0	0	0	1	8	36	120	330	792	1716	3432	6427
hashes needed	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Puzzle A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Puzzle B	11376	19160	30864	47748	71184	102552	143088	193705	254808	326124	406568	494166	586056	678588	767544	848443
hashes needed	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
Puzzle A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Puzzle B	916896	968976	1001568	1012664	1001568	968976	916896	848443	767544	678588	586056	494166	406568	326124	254808	193705
hashes needed	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
Puzzle A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Puzzle B	143088	102552	71184	47748	30864	19160	11376	6427	3432	1716	792	330	120	36	8	1

b) I wrote a java program to calculate the number of frequencies based on the hashes needed. It has multiple loops that loops between number 1 to 8 each and a variable named called "hash" so the loop will loop and calculate the total number and if it is the same value as "hash" the frequency count will increase. If there is duplicate or the number exceeds the "hash" value, the code will break and loop again and at the end I will output the count value based on the hash.

c)





d) The average number of hashes needed:

Puzzle A:

Worst expected of hashes = m * 2^k

= 1 * 2^6

= 64

Average number of hashes = $\frac{(\frac{n(n+1)}{2})}{n}$

= $\frac{(\frac{64(64+1)}{2})}{64}$

= $\frac{2080}{64}$

= 32.5 hashes

Puzzle B:

Worst expected of hashes = m * 2^k

= 1 * 2^3

= 8

Average number of hashes = $\frac{(\frac{n(n+1)}{2})}{n}$

= $\frac{(\frac{8(8+1)}{2})}{8}$

= $\frac{36}{8}$

= 4.5 hashes

Since there are 8 sub puzzles.

4.5 * 8 = 36 hashes

e) The standard deviation:

Puzzle A:

Variance = $\frac{(32.5-1)2 + (32.5-2)2 + (32.5-3)2 + \dots + (32.5-64)2}{64}$

= 341.25

Standard Deviation = $\sqrt{341.25}$ = 18.47

Puzzle B:

Variance = $\frac{(4.5-1)2 + (4.5-2)2 + (4.5-3)2 + \dots + (4.5-8)2}{8}$

= 5.25

Since there is 8 sub puzzle 5.25 * 8 = 42

Standard Deviation = $\sqrt{42}$ = 6.48

2. The original pseudo-code violated the default deny, not default allow principle.

   Fix:

```
        permit = CheckAccess()
        IF (permit == Access_Granted)
                Print "Access Granted"
                Run Function()
        ELSE
                Print "Access Denied"
```

3.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|\bar{A})P(\bar{A})+P(B|A)P(A)}$$

$$= \frac{0.05\ x\ \frac{799}{800}}{\left(0.05\ x\ \frac{799}{800}\right)+(0.95\ x\ \frac{1}{800})}$$

$= 0.97677$

$\approx 97.68\%$

Therefore the probability that the message is actually okay is 97.68%

4. One of these instances would be the infamous Stuxnet cyber attack on Iran's nuclear program this was an attack done by an insider which was recruited by a Dutch intelligence agency. The Stuxnet code was written to sabotage the nuclear facility only in certain operational scenarios.

   The insider deployed the virus through USB to jump the airgaps. Stuxnet is a computer worm type of malware which targets Supervisory Control and Data Acquisition (SCADA) systems. It can use multiple zero-day attacks. The worm uses other exploits like peer-to-peer remote procedure call to infect and update all the other hardware which were within the private network that weren’t connected directly to the internet.

Outcomes of the damage done by Stuxnet has been mostly unknown. Some estimates believe Stuxnet was able to destroy 1000-2000 centrifuges at the Natanz facility.

5.

a.

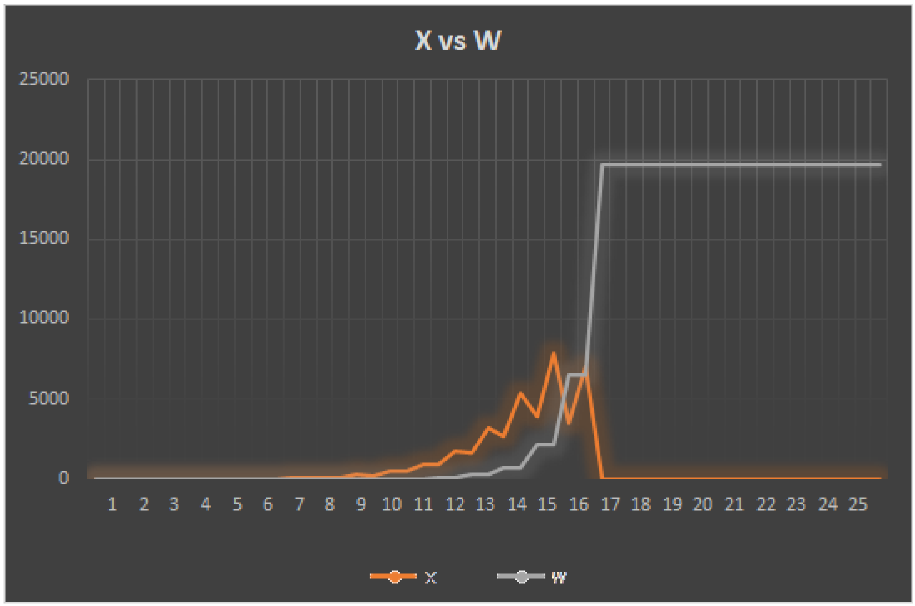
T	X
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768
16	65536
17	131072
18	262144
19	524288
20	1048576
21	2097152
22	4194304
23	8388608
24	16777216

b.

T	X	W
0	1	0
0.5	1	0
1	2	0
1.5	2	0
2	4	0
2.5	4	0
3	8	0
3.5	8	0
4	16	0
4.5	16	0
5	32	0
5.5	32	0
6	64	0
6.5	63	1
7	126	1
7.5	124	3
8	248	3
8.5	242	9
9	484	9
9.5	466	27
10	932	27
10.5	878	81
11	1756	81
11.5	1594	243
12	3188	243

T	W	X
12.5	2702	729
13	5404	729
13.5	3946	2187
14	7892	2187
14.5	3518	6561
15	7036	6561
15.5	0	19683
16	0	19683
16.5	0	19683
17	0	19683
17.5	0	19683
18	0	19683
18.5	0	19683
19	0	19683
19.5	0	19683
20	0	19683
20.5	0	19683
21	0	19683
21.5	0	19683
22	0	19683
22.5	0	19683
23	0	19683
23.5	0	19683
24	0	19683

c.



d. At T = 9, X already infected 484 computers whereas W has only infected 9 computers. If X now spreads faster than W, then there is no stopping X which will continue to infect much more computer than the rate of W.

6.

a. XML bomb tries to overload an XML parser which is normally a HTTP server. It does so by using messages to the server. When the XML parser processes an XML bomb, the data will feed on itself and grows exponentially. It will be able to shutdown a website or even possibly an internet service provider. Generally used to carry out DDoS attack.

b. BlueSmack is a Bluetooth attack that enables it to take out some Bluetooth enabled devices immediately. It is done by using l2ping which comes in Linux Bluex utils package. With this, Bluetooth enabled devices will be swamped with malicious request from the hacker which causes the device to crash or not usable. The performance might of the victim's device might be degraded as an effect of the attack.

c. Mydoom is a computer worm which affects the Windows OS it spreads itself through infect email attachment and peer-to-peer network. The worm will install a backdoor on TCP port and launch scheduled DoS attack.

d. Torpig is a type of botnet that spread through systems that were compromised by Mebroot rootkit through a variety of trojan. Torpig usually targets computers which uses Window OS and by doing so it will recruit a network of zombies for the botnet. It is able to modify data on the computer and able to perform man-in-the browser attacks