Kendrick Kee
7366814

Question 1:

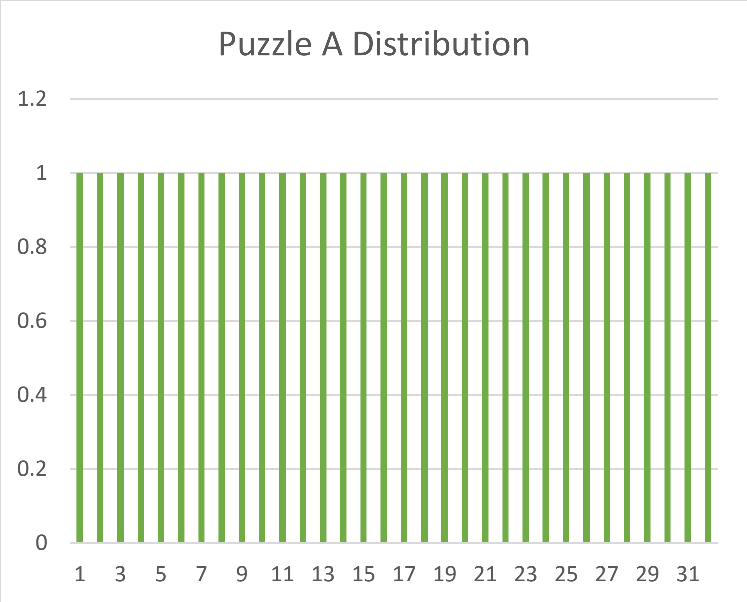
a)

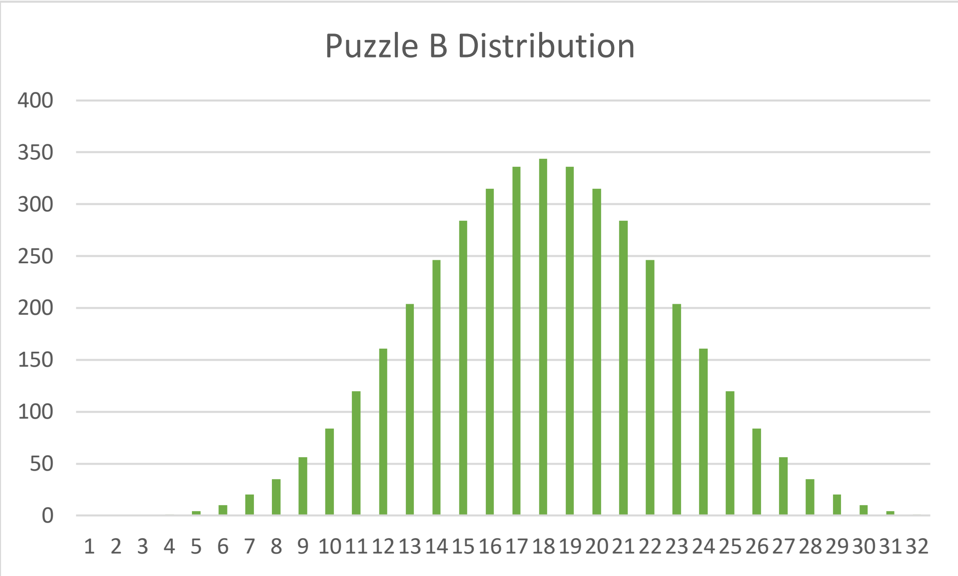
Total Expected Hash for Puzzle A = 32		
Expected Hash	**Frequency (A)**	**Frequency (B)**
1	1	0
2	1	0
3	1	0
4	1	1
5	1	4
6	1	10
7	1	20
8	1	35
9	1	56
10	1	84
11	1	120
12	1	161
13	1	204
14	1	246
15	1	284
16	1	315
17	1	336
18	1	344
19	1	336
20	1	315
21	1	284
22	1	246
23	1	204
24	1	161
25	1	120
26	1	84
27	1	56
28	1	35
29	1	20
30	1	10
31	1	4
32	1	1

b) For puzzle B, a python program was written to calculate the number of hashes required. It consists of multiple nested loops, the first looping for the total number of expect hash which is 32, an inner loop will execute from 1 to 8 for a total of 4 times . The last loop will proceed to then compute the total value and check if this sum is equal to a variable . If true, the frequency counter will be incremented.

Kendrick Kee
7366814

Question 1:
c)





Kendrick Kee
7366814

d)

Puzzle A

worst no. of hashes

where puzzle A
$m = 1$
$k = 5$

$m \times 2^k$

$\approx 1 \times 2^5$

Avg no. of hashes

$$\frac{\left(\frac{n(n+1)}{2}\right)}{n}$$

$$\frac{\left(\frac{32(33)}{2}\right)}{32} = \frac{528}{32} = 16\tfrac{1}{2} \approx 16.5 \text{ Avg hashes}$$

Puzzle B

worst no. of hashes

$1 \times 2^3 = 8$

Avg no of hash

$$\frac{\left(\frac{8(8+1)}{2}\right)}{8} = \frac{\left(\frac{8(9)}{2}\right)}{8} = \frac{36}{8} = 4.5$$

As the are 4 sub puzzles, the avg number of hashes is $4 \times 4.5 = 18$ hashes

e)

Puzzle A

$$\sigma \text{ variance} = \frac{(16.5-1)^2 + (16.5-2)^2 + (16.5-3)^2 \ldots (16.5-32)^2}{32}$$

$= 85.25$

$\therefore$ SD of puzzle A is $\sqrt{85.25} \approx 9.2331$

Puzzle B

$$\sigma \text{ variance} = \frac{(18-1)^2 + (18-2)^2 + (18-3)^2 \ldots (18-8)^2}{8}$$

$= 5.25$

As there are 4 sub puzzles

Variance $= 4 \times 5.25 = 21$

$\therefore$ SD of puzzle B is $\sqrt{21} \approx 4.5826$

Kendrick Kee
7366814

Question 2:

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

Of the two basic philosophies in computer security related to access control, default allow and default deny. “Default Deny” is regarded as the industry’s best practise as it will permit only the bare minimum required access to the supposed system.

Hence the pseudo code should be:

```
permit = checkAccess()
if (permit == Access_Granted):
    print("Access Granted")
    Run function
else:
    Print "Access Denied"
```

Question 3:

Bayes Theorem:

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

P(A) = 799/800
P(B|A) = 5% = 0.05
P($\bar{A}$) = 1/800
P(B|$\bar{A}$) = 95% = 0.95

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

Therefore, P = 80.6 ~ the chance of the message being clean is 80.6%

Kendrick Kee
7366814

Question 4:

With the advent of open source collaboration and projects dominating the current software scene, two security researches and open source software contributors from the University of Minnesota decided to push and contribute known malicious vulnerabilities to the Linux kernel maintained by the Linux Foundation. The two contributors claims were that they were doing that as part of their paper "On the Feasibility of Stealthily Introducing Vulnerabilities in Open-Source Software via
Hypocrite Commits" debating that the security by making software open source is ineffective and decided to prove their point by sending malicious commits into the code base using their authority as inducted contributes to the Linux Kernel project.

Majority of their malicious commits were Use-After-Free() vulnerabilities (CVE-2019-15922) which is a common Heap Bug in the OS Kernel. This bug has the potential to allow an attacker to gain access to a system without owning the correct credentials in the presence of a infected system. A summary of how an attacker would proceed would be to,
first: Allocate a chunk on the machine to store the username
second: Allocate another chunk on the machine to store the password
third: Free the chunks using said vulnerability
fourth: by using the Free()d chunks, the chunk will return the last 2 pointers in the head that was previously storing the username and password.
Thereby, allowing the attacker to further comprise the system with his stolen access.

The repercussions faced by the two researchers were mild for this case as compared to other insider attacks. Firstly, they were banned from contributing the code repository for life, all their previous commits has been pulled out from the code base and is being "re-reviewed" by the rest of the open source community citing that the Linux Foundation does not appreciate "being experimented on". Furthermore, as the researchers were from University of Minnesota, the Linux foundation has decided to ban all contributors from the Institution as well despite unpopular sentiment.

References:

Clark, M. 2021. Available at : https://www.theverge.com/2021/4/22/22398156/university-minnesota-linux-kernal-ban-research

Wu QuiShi, Lu Kangjie. 2021.
Available at:
https://raw.githubusercontent.com/QiushiWu/qiushiwu.github.io/main/papers/OpenSourceInsecurity.pdf

MITRE. 2019 **https://nvd.nist.gov/vuln/detail/CVE-2019-15922**

Kendrick Kee
7366814

Question 5:

a) Given at t = 0, the first worm has infected the first machine. After an hour passes, the host machine will infect another machine giving a total number of infected machines X = 2 at t= 1. Subsequently, at t = 3, the 2 infected machines will each infect a uninfected computer giving a grand total of 4 infected machines (X = 4), this goes on for the next 24 hours leading to a total of 2^{24} (r^t) given that r = rate of infection. A table below represents the number of infected computers over time:

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

Kendrick Kee
7366814

Question 5:

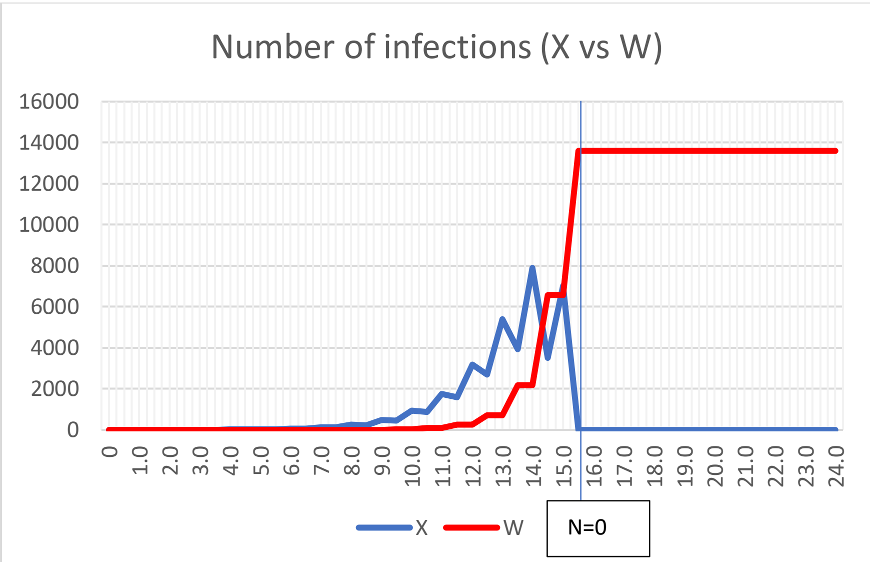
b) For t = 0 till t = 6.5, the amount of infected computers X will remain the same as the table shown in figure 5a, that is where W a counter worm is deployed with a growth rate of 2. This would mean that on t = 6.5, there would be 63 infected computers as compared to 64 in the previous non-counter worm example. As the growth rate for W is faster than X over time we would see X increasing and decreasing over time as shown in the table below. And at t= 15.5, the amount of X is 0 as the counterworm has completed 'countered' all infected X machines having the total count of W plateau at 13597 indefinitely.

t	X	W
0	1	0
0.5	1	0
1.0	2	0
1.5	2	0
2.0	4	0
2.5	4	0
3.0	8	0
3.5	8	0
4.0	16	0
4.5	16	0
5.0	32	0
5.5	32	0
6.0	64	0
6.5	63	1
7.0	126	1
7.5	124	3
8.0	248	3
8.5	242	9
9.0	484	9
9.5	466	27
10.0	932	27
10.5	878	81
11.0	1756	81
11.5	1594	243
12.0	3188	243
12.5	2702	729
13.0	5404	729
13.5	3946	2187
14.0	7892	2187
14.5	3518	6561
15.0	7036	6561
15.5	0	13597
16.0	0	13597
16.5	0	13597

Kendrick Kee
7366814

17.0	0	13597
17.5	0	13597
18.0	0	13597
18.5	0	13597
19.0	0	13597
19.5	0	13597
20.0	0	13597
20.5	0	13597
21.0	0	13597
21.5	0	13597
22.0	0	13597
22.5	0	13597
23.0	0	13597
23.5	0	13597
24.0	0	13597

c)



c) Assuming t = 9, where X has 484 infected machines, and W has 9 counter infected machines, X will continue to spread indefinitely as the rate of spread of X is greater than W. In which case the rate of total W will never exceed the total amount of X in this example.

Kendrick Kee
7366814

Question 6:

a) An XML Bomb, is a type of denial-of-service (Dos) in which the main goal is to overload and eventually crash a server (usually a server that accepts HTTP request). It is also colloquially known as a 'billion laughs attack' due to it's implementation usually requiring the attacker to define an entity in XML denoting '&lol9;' which abbreviates to laughing out loud in the current internet culture. The server crashes upon parsing a XML Bomb due to the nested data entities sent to the server for parsing, as the nested entities grow exponentially creating a "data explosion" situation hence the name "bomb" in the attack and thereby crashing the server.

b) A Bluesmack is a cyber-attack done on Bluetooth-enable devices, it falls within the denial-of-service (Dos) domain as the main idea behind a Bluesmack attack is to overload the victim's Bluetooth device with massive packets of data over s L2CAP (Logic Link Control And Adaptation Protocol) layer. This attack is usually done with standard Bluetooth debug and development tools such as l2ping that is packaged with Linux Bluex utils package shipping in most Linux distros. This attack is possible as it exploits the fact that all Bluetooth 'echos' has to be received, parsed and responded to. In which case the attacker would customise the sent 'echo' packet to a extremely large size at a high frequency to a victims device usually resulting in a degradation in the existing Bluetooth connection of the victims device and usually causing a connection outage denying them of the service.

c) Mydoom is a computer worm affecting the Windows operating system by Microsoft. The worm propagates through a 'phishing' style email with usually cites "I'm just doing my job, nothing personal, sorry." Which incites curiosity in the victim causing them to open the attachment with the email which executes bits of code which is the MyDoom virus/worm. Upon execution, the code will then dig the user's contacts usually the Outlook address book located in the Windows machine and sends a copy of itself (propagation) to every contact found. Once completed, the worm stay dormant until a set date of which is will begin spamming requests to a larger organisation's network in 2004's case was SCO Group and Microsoft's web domains thereby performing distributed denial of service attack on these organisations.

d) Torpig is a Trojan horse that exclusively affects the Windows platform. It comprises of a keylogger which records every keystroke of the victims machine which in turn can lead to compromised credentials or confidential information being exposed to the wrong party. The malware is usually propagated using social engineering means for example getting the victim to run the file unknowingly or by hiding the executable in a known 'clean' file which will be concurrently executed hence infecting the victim's machine. The key logs generated on the infected machine will then be sent out via HTTP to a remote user/server set up by the attacker for him to review and sieve out any confidential information/credentials.

Kendrick Kee
7366814

References:

https://www.okta.com/identity-101/mydoom/
https://www.eweek.com/security/mebroot-the-stealthiest-rootkit-in-the-wild/
https://www.cybervie.com/blog/bluesmack-attack/
https://www.soapui.org/docs/security-testing/security-scans/xml-bomb/