1)

a) Puzzle A:

Expected hash (E)	Frequency (F)	(E x F)	Mean - E	((Mean - E)^2)*F
1	0	0	-17	0
2	0	0	-16	0
3	0	0	-15	0
4	1	4	-14	196
5	4	20	-13	676
6	10	60	-12	1440
7	20	140	-11	2420
8	35	280	-10	3500
9	56	504	-9	4536
10	84	840	-8	5376
11	120	1320	-7	5880
12	161	1932	-6	5796
13	204	2652	-5	5100
14	246	3444	-4	3936
15	284	4260	-3	2556
16	315	5040	-2	1260
17	336	5712	-1	336
18	344	6192	0	0
19	336	6384	1	336
20	315	6300	2	1260
21	284	5964	3	2556
22	246	5412	4	3936
23	204	4692	5	5100
24	161	3864	6	5796
25	120	3000	7	5880
26	84	2184	8	5376
27	56	1512	9	4536
28	35	980	10	3500
29	20	580	11	2420
30	10	300	12	1440
31	4	124	13	676
32	1	32	14	196
	4096		-18	86016

Puzzle B:

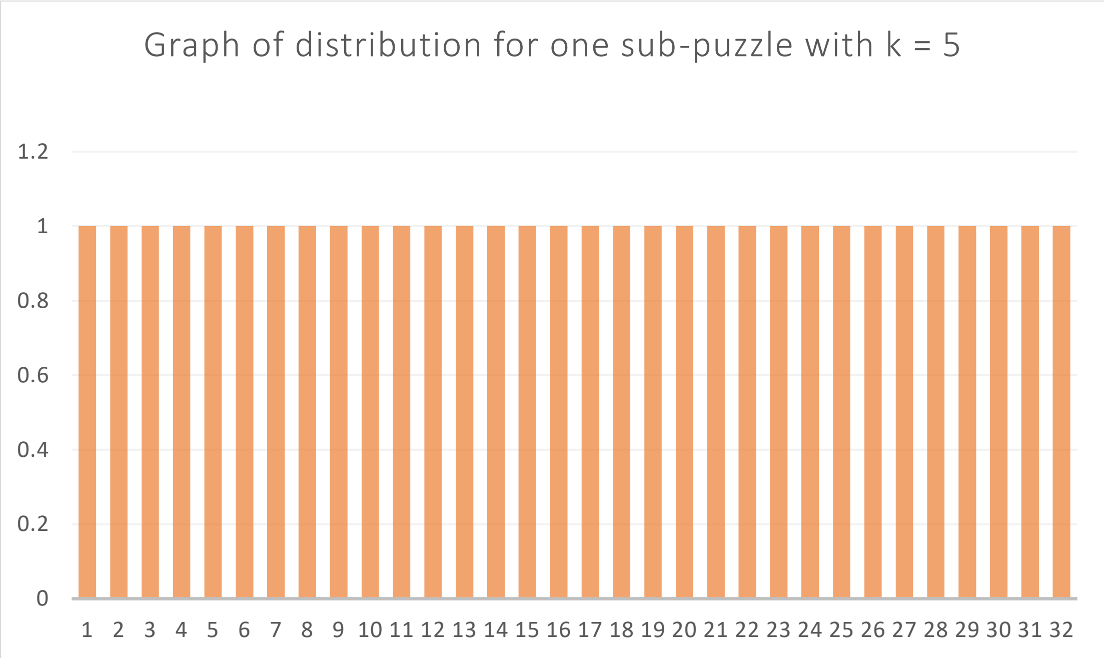
Expected hash (E)	Frequency (F)	(E x F)	Mean - E	((Mean - E)^2)*F
1	0	0	-17	0
2	0	0	-16	0
3	0	0	-15	0
4	1	4	-14	196
5	4	20	-13	676
6	10	60	-12	1440
7	20	140	-11	2420
8	35	280	-10	3500
9	56	504	-9	4536
10	84	840	-8	5376
11	120	1320	-7	5880
12	161	1932	-6	5796
13	204	2652	-5	5100
14	246	3444	-4	3936
15	284	4260	-3	2556
16	315	5040	-2	1260
17	336	5712	-1	336
18	344	6192	0	0
19	336	6384	1	336
20	315	6300	2	1260
21	284	5964	3	2556
22	246	5412	4	3936
23	204	4692	5	5100
24	161	3864	6	5796
25	120	3000	7	5880
26	84	2184	8	5376
27	56	1512	9	4536
28	35	980	10	3500
29	20	580	11	2420
30	10	300	12	1440
31	4	124	13	676
32	1	32	14	196
	4096		-18	86016

b) Puzzle A: I made use of the Microsoft Excel's in-built formulas and functions to calculate the distributions. For puzzle A, there is only one sub-puzzle, hence frequency is 1. And since puzzle A has only 1 sub-puzzle with k = 5, the worst case expected hashes with k = 5 is 32. Thus, each hash will only appear once, which explains the even distribution of one sub-puzzle.

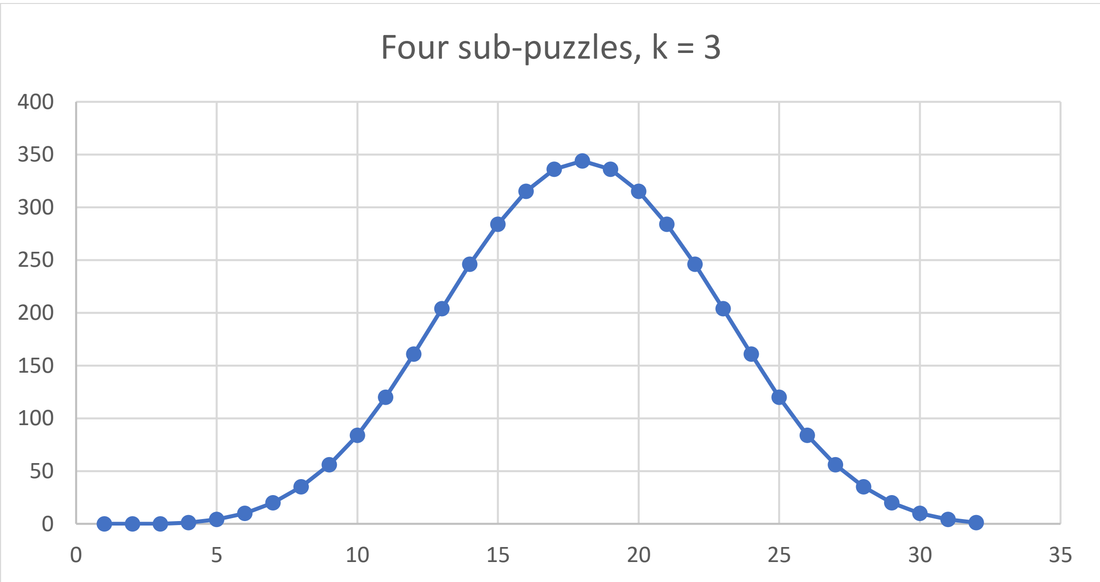
Puzzle B: I wrote a python code, making us of the combinations library to calculate the possible combinations. For puzzle B, there are four sub-puzzles, hence to avoid confusion, I calculated based on one sub-puzzle first then multiplying by 4. For instance, 1 sub-puzzle with k = 3, the worst case expected hashes with k = 3 is 8. Thus, with 4 sub-puzzles, the worst case expected hashes is 32. To get the distribution, I used the itertools.product library from python to check and loop through from hash value 1 to 32, check that if the sum of the 4 hash value in the sub puzzle matches the expected hash, it would count+=1 and I would repeat and display the total possible combinations.

c)

Puzzle A:



Puzzle B:



d) Puzzle A: The average number of hashes required is 16.5 Since Puzzle A is 1 sub puzzle with k = 5, 1 x 2^5= 32, which is the worst case expected hashes. Mean is as calculated.

$$\frac{\sum_{n=1}^{32} n}{32} = \frac{\frac{32(32+1)}{2}}{32} = \frac{528}{32} = 32.5$$

Puzzle B: The average number of hashes required is 18. Since Puzzle B has 4 sub-puzzles with k = 3, we calculate the worst case expected hashes of 1 sub puzzle first which is 4.5. 4.5x4 sub-puzzles which is 18 in total.

$$\frac{\sum_{n=1}^{8} n}{8} = \frac{\frac{8(8+1)}{2}}{8} = \frac{36}{8} = 4.5\ (1\ sub\ puzzle)$$

e) Puzzle A: The standard deviation for the distribution of hashes required is 9.233093.
Puzzle B: The standard deviation for the distribution of hashes required is 4.58258.

2) The original code given reflects that if the user is not granted access before the function is called/action is being performed. This breaks “default deny rule” whereby we should enforce positive validation rather than negative. As such, the following pseudocode should be applied as follows:

```
permit = CheckAccess()
IF (permit == Access_Granted)
        Print “Access Granted”
        Run Function ()

ELSE

        Print "Access Denied"
```

3)

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

$$P = \frac{0.05 \times \frac{799}{800}}{(0.05 \times \frac{799}{800}) + (0.95 \times \frac{1}{800})}$$

$$\therefore\ P = 80.6$$

There is 80.6% chance of the message being clean.

4) There has been a rise in bribing employees to install malware from malicious conspirator(s). An example of such case would employees from A&T. Many were bribed by Muhammad Fahd, to install malware – in this case, a keylogger on AT&T’s network at the bothell call center. The malware collected data such as the how the infrastructure worked as well as functioning of the internal protected computers and applications. This happened over a span of a few years, namely from 2012 to September of 2017.

A keylogger, often defined as a software, which is programmed to secretly monitor and log all keystrokes. This means that all information typed into a website or application is recorded, and sent back to a third party, which leverages on algorithms via techniques such as pattern recognition.

The attackers, improved on their malware and created a second malware strain that leveraged on the information acquire. By using AT&T employee credentials. The malware could perform automated actions on AT&T's internal application to unlock phone's without needing to interact with AT&T employees every time. Rogue wireless access points inside AT&T's Bothell call center were also installed, gaining access to AT&T internal apps and network to continue the rogue phone unlocking scheme. The motive behind plotting the attack, and to recuperate loss, was selling phone unlocking services through the now-defunct SwiftUnlocks.com website.

As a result, phones do not need to be “locked” in AT&T subscription. Roughly 2 million of smartphones were unlocked, accounting to more than $200million lost in terms of subscription fees, as phones could be used on another carrier’s network. Muhammad Fahd was arrested in Hong Kong in 2018 and extradited to the US in August 2019. He pleaded guilty in September 2020 and was sentenced 12 years in prison for the 7 years fraud.

References

Cimpanu, C., 2019. *AT&T employees took bribes to plant malware on the company's network.* [Online]
Available at: https://www.zdnet.com/article/at-t-employees-took-bribes-to-plant-malware-on-the-companys-network/
[Accessed 25 October 2021].

Swinhoe, D., 2018. *What is a keylogger? How attackers can monitor everything you type.* [Online]
Available at: https://www.csoonline.com/article/3326304/what-is-a-keylogger-how-attackers-can-monitor-everything-you-type.html
[Accessed 25 October 2021].

Techopedia, 2019. *Mydoom.* [Online]
Available at: https://www.techopedia.com/definition/27497/mydoom
[Accessed 21 October 2021].

Techopedia, n.d. *XML Bomb.* [Online]
Available at: https://www.techopedia.com/definition/13716/xml-bomb
[Accessed 23 October 2021].

Wadhwani, S., n.d. *Fraudster Gets 12 Years in Prison for Conning AT&T Out of $200M.* [Online]
Available at: https://www.toolbox.com/tech/security/news/att-fraudster-sentenced-12-years/
[Accessed 25 October 2021].

5)

a) Given the statement that when t=0, the number of infected computers is 1, when t=1 it will affect another computer which means x = 2. Each infected computer can spread to 1 uninfected computer, thus when x = 3, 4 of the infected computers will spread to 1 uninfected computer each, which results at x = 8. And this pattern continues till the 24^{th} hour mark. I make use of the formula of 2^t which will give me the total.

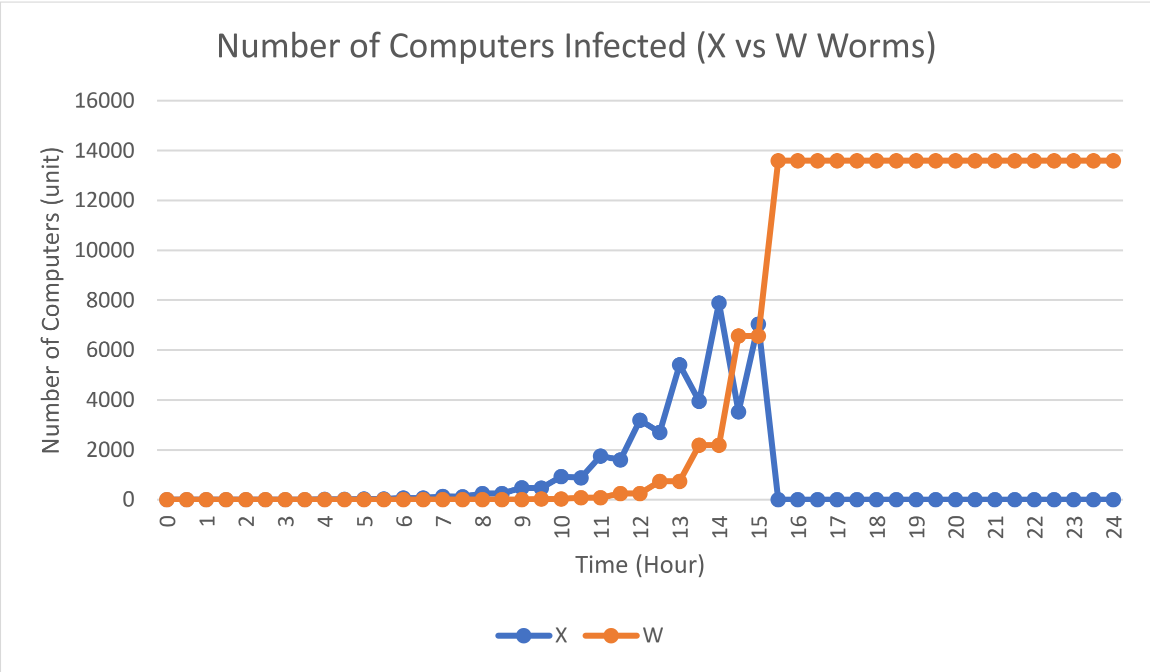
	t	**X**
0	0	1
1	0.5	2
2	1.0	4
3	1.5	8
4	2.0	16
5	2.5	32
6	3.0	64
7	3.5	128
8	4.0	256
9	4.5	512
10	5.0	1024
11	5.5	2048
12	6.0	4096
13	6.5	8192
14	7.0	16384
15	7.5	32768
16	8.0	65536
17	8.5	131072
18	9.0	262144
19	9.5	524288
20	10.0	1048576
21	10.5	2097152
22	11.0	4194304
23	11.5	8388608
24	12.0	16777216

b) Given the statement that the counterworm spreads to 2 infected computers at each 0.5 mark, whereas X infects at every whole hour mark. At 6.5 hour, x is 63 and w is 1 because counterworm just started. When 7.0, X is 2 multiply by 63 which turns to 126, and at 7.5 hour It turns out to be 124 because W is 3 now, whereby we need to minus off the recovered computer and this happens every hour.

	t	X	W
0	0	1	0
1	0.5	1	0
2	1.0	2	0
3	1.5	2	0
4	2.0	4	0
5	2.5	4	0
6	3.0	8	0
7	3.5	8	0
8	4.0	16	0
9	4.5	16	0
10	5.0	32	0
11	5.5	32	0
12	6.0	64	0
13	6.5	63	1
14	7.0	126	1
15	7.5	124	3
16	8.0	248	3
17	8.5	242	9
18	9.0	484	9
19	9.5	466	27
20	10.0	932	27
21	10.5	878	81
22	11.0	1756	81
23	11.5	1594	243
24	12.0	3188	243
25	12.5	2702	729
26	13.0	5404	729
27	13.5	3946	2187
28	14.0	7892	2187
29	14.5	3518	6561
30	15.0	7036	6561
31	15.5	0	13597
32	16.0	0	13597
33	16.5	0	13597
34	17.0	0	13597
35	17.5	0	13597
36	18.0	0	13597
37	18.5	0	13597
38	19.0	0	13597
39	19.5	0	13597
40	20.0	0	13597
41	20.5	0	13597
42	21.0	0	13597

43	21.5	0	13597
44	22.0	0	13597
45	22.5	0	13597
46	23.0	0	13597
47	23.5	0	13597
48	24.0	0	13597

c)



d) When t = 9, X has infected 484 computers and W has only infected 9 computers. Assuming the time if t = 9 and X evolves to spread to three uninfected computers, X will continue to manifest and there will be no stopping. To add on, as X has infected 484 computers, it will manifest at a much faster and steeper rate than W.

6)

a) An XML Bomb, also referred commonly as a billion laughs attack, is a denial-of-service (DoS) attack in malware. DoS attack prevents or hinders the use of the system. XML bomb encompasses a message or a piece of dangerous code, intended to crash the server or program (typically HTTP server) which tries to read and decode it by overloading it, which can result in the shutting down of a server or ISP.

b) A Bluesmack refers to a cyber-attack targeted on Bluetooth enabled devices. It is a denial-of-service (DoS) attack in malware. An oversized packet is sent to the Bluetooth enabled devices by the attacker, by usage of tools such as L2CAP (Logic Link Control And Adaptation Protocol) layer. The attack overwhelms the device via malicious requests, making it inaccessible to the owner and drains the battery life, and affects the operations of the device.

c) Mydoom is a type of computer worm done that affects the Microsoft Windows Operating System and is a denial-of-service (DoS) attack in malware. The attack works when a user opens the attachment, and the worm starts to scrap email addresses of the infected window computers and spread to victim's contacts by duplicating itself and presenting a new version of itself as a malicious attachment, performing Distributed Denial of Service (DDoS) attacks. This kind of attack is aggressive because the malware is sufficient and potentially could last forever.

d) Torpig, also known as Sinowal or Mebroot, is a type of trojan horse in malware, that targets users on the Microsoft Windows Platform. It makes use of a keystroke logger, which works by recording key presses and opens window files, collecting information to a remote user via HTTP. It can be used to download and execute various file to infect the computer with other malware. Such functions can be used to steal credentials and use to make unauthorized transactions and purchases, and other activities such as identity theft.

References

Cimpanu, C., 2019. *AT&T employees took bribes to plant malware on the company's network.* [Online]
Available at: https://www.zdnet.com/article/at-t-employees-took-bribes-to-plant-malware-on-the-companys-network/
[Accessed 25 October 2021].

Government of Singapore, 2014. *The Bluetooth Security Issues.* [Online]
Available at: https://www.csa.gov.sg/gosafeonline/go-safe-for-me/homeinternetusers/the-bluetooth-security-issues
[Accessed 23 October 2021].

Ionots Technologies Pvt.Ltd, n.d. *BlueSmack Attack | What is Bluetooth Hacking?.* [Online]
Available at: https://www.cybervie.com/blog/bluesmack-attack/
[Accessed 23 October https://www.cybervie.com/blog/bluesmack-attack/].

Johanns, K., 2021. *Tech Time Warp: Torpig malware collects stolen financial data.* [Online]
Available at: https://smartermsp.com/tech-time-warp-torpig-malware-collects-stolen-financial-data/
[Accessed 23 October 2021].

J, S., n.d. *BlueSmack Attack.* [Online]
Available at: https://iq.opengenus.org/bluesmack-attack/
[Accessed 23 October 2021].

Naraine, R., 2009. *Botnet hijack: Inside the Torpig malware operation.* [Online]
Available at: https://www.zdnet.com/article/botnet-hijack-inside-the-torpig-malware-operation/
[Accessed 23 October 2021].

Palmer, D., 2019. *MyDoom: The 15-year-old malware that's still being used in phishing attacks in 2019.* [Online]
Available at: https://www.zdnet.com/article/mydoom-the-15-year-old-malware-thats-still-being-used-in-phishing-attacks-in-2019/
[Accessed 23 October 2021].

Seals, T., 2021. *AT&T Phone-Unlocking Malware Ring Costs Carrier $200M.* [Online]
Available at: https://threatpost.com/att-phone-unlocking-malware/174787/
[Accessed 25 October 2021].

SmartBear Software, 2021. *XML Bomb.* [Online]
Available at: https://www.soapui.org/docs/security-testing/security-scans/xml-bomb/
[Accessed 23 October 2021].

Sophos Ltd, n.d. *Troj/Torpig-A.* [Online]
Available at: https://www.sophos.com/en-us/threat-center/threat-analyses/viruses-and-spyware/Troj~Torpig-A/detailed-analysis.aspx
[Accessed 23 October 2021].

Swinhoe, D., 2018. *What is a keylogger? How attackers can monitor everything you type.* [Online]
Available at: https://www.csoonline.com/article/3326304/what-is-a-keylogger-how-attackers-can-monitor-everything-you-type.html
[Accessed 25 October 2021].

Techopedia, 2019. *Mydoom.* [Online]
Available at: https://www.techopedia.com/definition/27497/mydoom
[Accessed 21 October 2021].

Techopedia, n.d. *XML Bomb.* [Online]
Available at: https://www.techopedia.com/definition/13716/xml-bomb
[Accessed 23 October 2021].

Wadhwani, S., n.d. *Fraudster Gets 12 Years in Prison for Conning AT&T Out of $200M.* [Online]
Available at: https://www.toolbox.com/tech/security/news/att-fraudster-sentenced-12-years/
[Accessed 25 October 2021].