School of Computing & Information Technology

CSCI862 System Security
Spring 2016

Assignment 2 (16 marks, worth 8%)

Due 11:59pm Saturday 24th September 2016

Part One: Short Answer — 13 Marks

For questions where you are referencing material outside the lecture notes, you should provide appropriate referencing.

1. Consider that you have two puzzles

   `Puzzle A:` One sub–puzzles. $k = 8$.

   `Puzzle B:` Four sub–puzzles. $k = 6$.

   For each puzzle provide

   (a) A graph of the distribution of the number of hashes needed. **1 Mark**

   ```
   The first is just like the first one in the lecture notes, a rectangle.
   The second one is approximating the classic bell curve shape.
   The different coloured bands in the graphs provided are just an
   artifact of the drawing.
   ```

   See the posted file: `A2-Q1-Graph.pdf`

   (b) The average number of hashes needed. **0.75 Mark**

   ```
   Puzzle A: 128.5
   Puzzle B: 130
   ```

   (c) The standard deviation for the distribution of the number of hashes needed. **2 Marks**
   Approximately:

   ```
   Puzzle A: 74.045
   Puzzle B: 36.96
   ```

   (d) Describe the method you used to obtain your solutions to (c). Don't go into too many details or show working, it's more "I wrote a C++ program to ... and then using ... I ...". **0.25 Mark**

You should assume that if there are N possible solutions you check the N^{th} by hashing even if all others have failed and there has to be a solution.

2. Using a TCP SYN spoofing attack, the attacker aims to flood the table of TCP connection requests on a system so that it is unable to respond to legitimate connection requests. Consider a server system with a table for 1024 connection requests. This system will retry sending the SYN-ACK packet five times when it fails to receive an ACK packet in response, at 45 second intervals, before purging the request from its table. Assume that no additional countermeasures are used against this attack and that the attacker has filled this table with an initial flood of connection requests. At what rate (per minute) must the attacker continue to send TCP connection requests to this system in order to ensure that the table remains full? Assuming that the TCP SYN packet is 50 bytes in size (ignoring framing overhead). How much bandwidth does the attacker consume to continue this attack? **1 Mark**

   For a TCP SYN spoofing attack, on a system with a table for 1024 connection requests, that will retry 5 times at 45 second intervals, each connection request occupies a table entry for 6 X 45secs (initial + 5 repeats) = 4.5 min. (0.5 marks)

   In order to ensure that the table remains full, the attacker must continue to send 1024 requests per 4.5 minutes or about 228 TCP connection requests per minute. (0.25 marks)

   Assuming the TCP SYN packet is 50 bytes in size, this consumes about 50*228 = 11,400 bytes per minute (or 50*228*8/60 =1,520 bits per second). (0.25 marks)

3. What is a password mangler and why would we use one? **1 Mark**

   Password manglers are client side components, typically browser plug--ins, that take a user password and produce a domain specific, and strong, password that is a function of the user entered password.
   It can be a function of some other secret too, held by your computer, so a dictionary attack is not going to be likely to succeed.
   This has the advantage of the user having different passwords for different locations without the user needing to remember lots of different passwords.

   The mechanism can be something like the PwdHash briefly described in the lecture notes.

4. What is a Cinderella attack? In particular, describe the target, the vulnerability being exploited, and the likely effect. **1 Mark**

   This is a concurrency based attack attempting to stop something like a malware scanning from working by changing the system clock to after the expiry date of the malware scanner. Potential effect: scanner stops working.

5. Every hour the malware **X** spreads from each infected computer to two previously uninfected computers. In answering these questions you should explain how you determined your answers.

   (a) Give a table showing the number of infected computers at each hour across a 24 hour period. At time $t = 0$ the number of **X** infected computers is $N = 1$. **0.5 Mark**

   ```
   Since at each step each infected computer infects two previously uninfected computers,
   the number of infected computers is multiplied by 3 at each step. At time $t=24$ the
   number will be up to $3^{24}$
   ```

Time	# Infected computers
0	$1 = 3^0$
1	$3^1 = 3$
2	$3^2 = 9$
3	$3^3 = 27$
4	81
5	243
6	729
7	2,187
8	6,561
9	19,683
10	59,049
11	177,147
12	531,441
13	1,594,323
14	4,782,969
15	14,348,907
16	43,046,721
17	129,140,163
18	387,420,489
19	1,162,261,467
20	3,486,784,401
21	10,460,353,203
22	31,381,059,609
23	94,143,178,827
24	$3^{24} = 282,429,536,481$

   (b) By time $t = 10.5$ a counter worm **W** has been developed and it is deployed on one infected computer. **W** removes malware **X** from any host **W** is on. The counter worm **W** spreads slightly more quickly than **X**, with each **W** spreading to three **X** infected hosts each hour, provided such hosts are available.

   Provide another table showing the spread of **W** and the impact on **X** across a relevant time frame, starting from $t = 0$ again.

   Note the offset in time means that at $t = 10.5$ the number of **X** infected computers reduces by 1, so the spread of $t = 11$ will be slightly smaller than before. **1.5 Marks**

   The number of cleaned computers goes up in powers of 4.

Time	# Infected computers	# Newly cleaned	# Cleaned computers
0	$1 = 3^0$		
1	$3^1 = 3$		
2	$3^2 = 9$		
3	$3^3 = 27$		
4	81		
5	243		
6	729		
7	2,187		
8	6,561		
9	19,683		
10	59,049		
10.5	59,048	1	1
11	177,144		1
11.5	177,141	3	4
12	531,423		4
12.5	531,411	12	16
13	1,594,233		16
13.5	1,594,185	48	64
14	4,782,555		64
14.5	4,782,363	192	256
15	14,347,089		256
15.5	14,346,321	768	1,024
16	43,038,963		1,024
16.5	43,035,891	3,072	4,096
17	129,107,673		4,096
17.5	129,095,385	12,288	16,384
18	387,286,155		16,384
18.5	387,237,003	49,152	65,536
19	1,161,711,009		65,536
19.5	1,161,514,401	196,608	262,144
20	3,484,543,203		262,144
20.5	3,483,756,771	786,432	1,048,576
21	10,451,270,313		1,048,576
21.5	10,448,124,585	3,145,728	4,194,304
22	31,344,373,755		4,194,304
22.5	31,331,790,843	12,582,912	16,777,216
23	93,995,372,529		16,777,216
23.5	93,945,040,881	50,331,648	67,108,864
24	281,835,122,643		$4^{13} = 67,108,864$

So far we have $594,413,838$ less infected computers at $t = 24$ relative to the first scenario. This seems pretty decent by itself but it's a small proportion of the number of computers that are actually infected.

Time	# Infected computers	# Newly cleaned	# Cleaned computers
24	281835122643		67108864
24.5	281633796051	20,326592	268435456
25	844901388153		268435456
25.5	844096081785	805306368	1073741824
26	2532288245355		1073741824
26.5	2529067019883	3221225472	4294967296
27	7587201059649		4294967296
27.5	7574316157761	12884901888	17179869184
28	22722948473283		17179869184
28.5	22671408865731	51539607552	68719476736
29	68014226597193		68719476736
29.5	67808068166985	206158430208	274877906944
30	203424204500955		274877906944
30.5	202599570780123	824633720832	1099511627776
31	607798712340369		1099511627776
31.5	604500177457041	3298534883328	4398046511104
32	1813500532371123		4398046511104
32.5	1800306392837811	13194139533312	17592186044416
33	5400919178513433		17592186044416
33.5	5348142620380185	52776558133248	70368744177664
34	16044427861140555		70368744177664
34.5	15833321628607563	211106232532992	281474976710656
35	47499964885822689		281474976710656
35.5	46655539955690721	844424930131968	1125899906842624
36	139966619867072163		1125899906842624
36.5	136588920146544291	3377699720527872	4503599627370496
37	409766760439632873		4503599627370496
37.5	396255961557521385	13510798882111488	18014398509481984
38	1188767884672564155		18014398509481984
38.5	1134724689144118203	54043195528445952	72057594037927936
39	3404174067432354609		72057594037927936
39.5	3188001285318570801	216172782113783808	288230376151711744
40	9564003855955712403		288230376151711744
40.5	8699312727500577171	864691128455135232	1152921504606846976
41	26097938182501731513		1152921504606846976
41.5	22639173668681190585	3458764513820540928	4611686018427387904
42	67917521006043571755		4611686018427387904
42.5	54082462950761408043	13835058055282163712	18446744073709551616
43	162247388852284224129		18446744073709551616
43.5	106907156631155569281	55340232221128654848	73786976294838206464
44	320721469893466707843		73786976294838206464
44.5	99360541008952088451	221360928884514619392	295147905179352825856
45	298081623026856265353		295147905179352825856
45.5	0	298081623026856265353	593229528206209091209

In the last step all of the currently infected computers are cleaned. To get full marks here I'm not expecting exactly the same solution precisely, but an indication, where explict or otherwise, that you have some awareness of the precision problem needs to be made.

I wasn't intending for this to be quite such an exercise on precision and I should have set the counterworm spread to start earlier or spread slightly more quickly. Starting W at the same point but having it spread to four infected computers each step instead of three would have killed off X at t=31.5; and at t=26.5 for spreading to five infected computers each step.

(c) Graph the two cases against each other, clearly indicating on it where $N = 0$. **0.5 Mark**

(d) Assume that at time $t = 12$, **X** evolves to spread to three uninfected computers each hour. What subsequently happens? **0.5 Mark**

The malware is now going to win. We cannot catch up. We don't really have this many computers though ...

6. In the context of phishing, list 8 points that can be used in checking the legitimacy of an email. Justify why each is appropriate as an indicator. Note that some points could relate to characteristics of legitimate messages, and others could be indicators of a phishing message. **2 Marks**

```
There are a lot of different points that could be made here so I'm not going to
attempt to give a complete list, and  I would expect most people to get this
question mostly correct anyway.
```

7. What protection is provided at the memory level by using the private access specifier in declaring a class in C++? Is it possible to overflow into private variables? **1 Mark**

```
It provides protection at the level of the coding, not at the level of memory.
It is certainly possible to overflow into private variables.
```