Continuous Assessment CS4615 – Ines Roman Gracia, 123123969

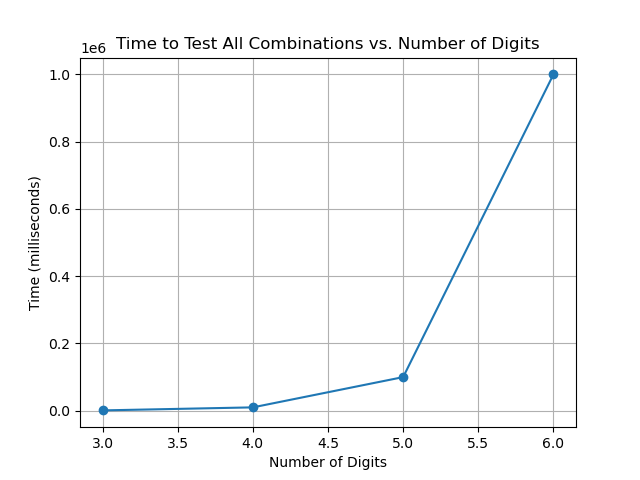
Question P1 [2 MARKS]: PIN Guessing

Assume the test of one combination takes 1ms. How long does it take to test all possible combinations for a 3 digit PIN, a 4 digit PIN, 5 digit PIN and 6 digit PIN. Plot a graph to show the relationship between time and number of digits. Can you give a formula describing the time to test all combinations in dependence of the number of digits N.

There are 10 possible digits:

* In a 3-digit PIN: 10³ possible combinations.
* In a 4-digit PIN: 10⁴ possible combinations.
* In a 5-digit PIN: 10⁵ possible combinations.
* In a 6-digit PIN: 10⁶ possible combinations.

The formula describing the time to test all combinations for a N-digit pin is 10^N.



Question P2 [2 MARKS]: Challange Collision Probability

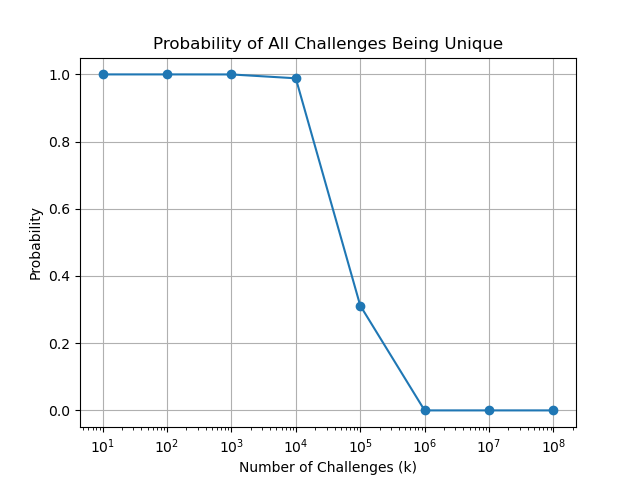
Assume in a CHAP authentication a 32bit integer is used as challenge. Assume that k authentication procedures are carried out. What is the probability that all k challenges are unique? Plot the result as a graph (probability in dependence of k).

For 32-bit integer challenges, there are 2³² possible combinations.

* After the first challenge, there are 2³² – 1 challenges that are different from the previous one. The probability that it will be different is (2³² – 1)/2³².
* After the second challenge, there are 2³² – 2 possible different challenges. The probability that the third will be different from the previous two is (2³² – 2)/2³².
* …
* After the k-1 challenge, there are 2³² – (k-1) possible different challenges. The probability that the k-th will be different from the previous k-1 challenges is (2³² – (k-1))/2³².

To find the probability that all k challenges are unique, we multiply these probabilities together:

P(all k challenges are unique) = (2³² – 1)/2³² × (2³² – 2)/2³² × … × (2³² – (k-1))/2³²



Question P3 [2 MARKS]: Balance Manipulation - Maximum Balance?

What is the maximum amount that the balance can be set to using a buffer overflow if you can only use lower and upper case letters? Explain in detail how you arrived at your answer.

To determine the maximum amount that the balance (a 4-byte integer) can be set to using a buffer overflow with only lower and upper case letters, we need to find the highest value that can be created using the 4 letters (each one is a byte).

Looking at an ASCII table, we observe that, of all the lower and upper case letters, the letter which corresponds to a higher value is ‘z’. The hexadecimal number corresponding to ‘z’ is 0x7a, so the highest number we can set using 4 letters is 0x7a7a7a7a, which translated into decimal is 2054847098.

Therefore, the maximum amount that the balance can be set to using a buffer overflow with only lower and upper case letters is 2054847098.

Question P4 [2 MARKS]: Buffer overflow protection methods

The virtual machine used in the practical has address space randomisation (in /proc/sys/kernel/randomize\_va\_space) turned off. Explain what address space randomisation is and how it is used to protect against buffer overflows.

Adress space randomisation is a computer security technique which prevents attackers from exploiting memory vulnerabilities by assigning random memory start addresses at a program start for the stack, heap and data. This makes harder carrying an overflow attack, where attackers attempt to overwrite data beyond the bounds of a buffer to hijack control flow or inject malicious code, because each time a program runs, the memory layout differs and attackers cannot predict the exact memory addresses they need to target.

Question P5 [2 MARKS]: Firewall Rules

Explain what a shadowing anomaly in the context of packet filters is. Give an example of a packet filter firewall policy that contains a shadowing anomaly. Explain your answer.

A firewall is a list of rules that determines which packets will be denied or passed in and out of a network. Shadow anomaly occurs when a rule unintentionally masks or shadows another rule listed below, this can lead to unexpected behavior where certain packets that should be permitted are instead denied due to a rule higher up in the list.

Here’s an example of two rules where shadowing anomaly occurs using the syntax specified in the lab script:

> {’PROTO’=’TCP’, ’SRC IP’=’ANY’, ’SRC PORT’=’ANY’, ’DST IP’=’ANY’, ’DST PORT’= ’ANY’ , ’ACTION’= ’DENY’}

> {’PROTO’=’TCP’, ’SRC IP’=’192.168.0.0/24’, ’SRC PORT’=’>1024’, ’DST IP’=’ANY’, ’DST PORT’= ’80’, ’ACTION’=’PASS’}

The second rule is masked by the first one. Packets with a source IP of 192.168.0.0/24, source port greater than 1024 and destination port 80 are aimed by the second rule to be allowed. However, these packets are also matched by the first rule, which will deny them.