Computer Systems Security Practicals

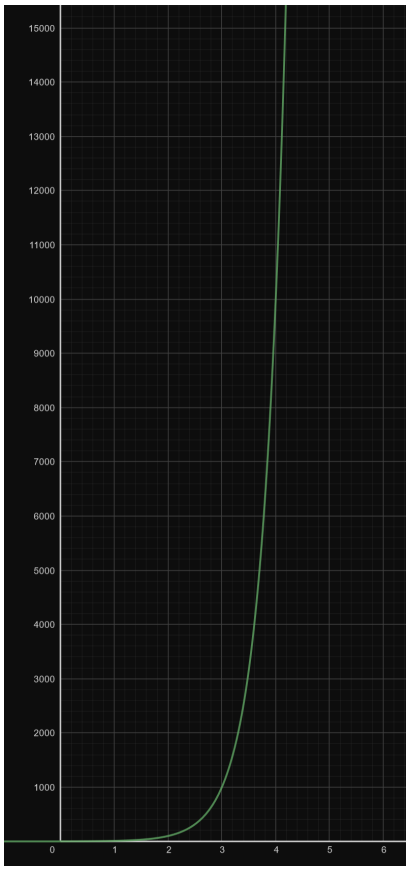
Alexander Stradnic - 119377263

Practical 1 - Authentication

One combination test time = 1ms
Number of distinct digits that can be used = 10 in base-10

- 3 digits = 000 - 999 -> 10^3 = 1,000ms
- 4 digits = 0000 - 9999 -> 10^4 = 10,000ms
- 5 digits = 00000 - 99999 -> 10^5 = 100,000ms
- 6 digits = 00000 - 99999 -> 10^6 = 1,000,000ms

Amount of time taken to test all combinations for a PIN comprised of digits D and length N:
T = D^N



Geogebra.org - T = 10^N

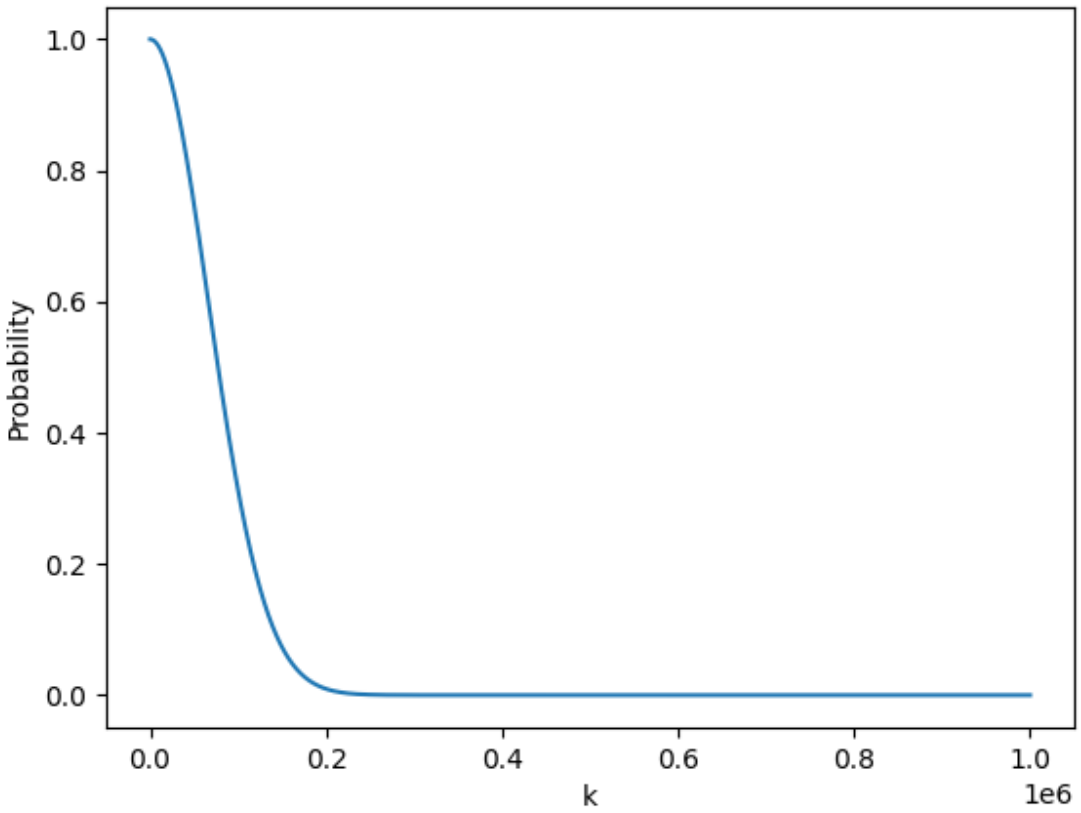
Practical 2 - Network Authentication

- 32-bit challenge int for CHAP = one of 2^32 numbers
- k authentication procedures

Given a procedure i, by virtue of being one procedure it is unique. P(1) = 1
Probability for a collision of 2 procs -> P(2) = [((2^32) - (2-1)) / (2^32)] * P(1), thus just ((2^32) - (2-1)) / (2^32)

Defining for list of unique procs higher than 2 -> P(3) = [((2^32) - (3-1)) / (2^32)] * P(2)

Generic function -> P(k procs) = [P(k)]*[P(k-1)]*...*[P(2)]
Subbing each k iteration i into the generic formula ((2^32) - (i-1)) / (2^32)



P(k): {k | 1 <= k <= 1,000,000}

From the graph, the first ~100,000 authentication procedures combined have a likelihood of about 50% to be unique from one another, with the probability falling further until it becomes infinitesimally after about ~200,000 statistically random procedures.

Practical 3 - Buffer Overflows

The maximum size of an integer in C depends on the platform and compiler used. Officially specified as a minimum of 2 bytes, most modern implementations, including on my machine, use 4 bytes.
Of all alphabet characters, ‘z’ has the highest bit value of 01111010/7A/122. Since each character takes a byte, 4 characters can be entered to replace the existing 4-byte integer balance.

To achieve the highest balance figure I entered a character string with the same length as l_name, so as to overwrite the null terminator, and then entered 4 ‘z’ characters.
However, fgets() appends a null terminator to the end of the input string, replacing the beginning character of the next first name.

The maximum value assigned to Balance in this method is 2,054,847,098.

Practical 4 - Buffer Overflow Protection

Address Space Layout Randomisation is a method used to protect against buffer overflow attacks. This is because it randomises the addresses of stack pointers, data, and code making it very difficult for a potential attacker to guess the locations of objects in memory for return-to-libc and shellcode attacks, where they would overflow the buffer, overwriting ret with their own code.

Practical 5 - Firewalls

Shadowing anomalies in the context of packet filters refers to when a specific filter rule is never accessed, due to being lower in priority than another rule, which is more general, thus always allowing or blocking a packet in set conditions before ever getting to the specific lower filter.

An example of a shadowing policy would be if a rule dropped TCP packets from a specific IP being placed above a rule that drops TCP packets from the specific IP but also specifying a port.

Incoming TCP packets from that IP are filtered by the first rule before ever reaching the second, port-checking rule.