Telling The Time

chris.anley@nccgroup.trust

The Bug

Server generates a time-based:

-Password reset token

-Session id

-Random password

-REST API Key

...

For example: PHP

uniqid()

“Gets a prefixed unique identifier based on the current time in microseconds.”

CAUTION NOT SECURE

WARNING NOT UNIQUE (?!)

blah blah SECURE blah UNIQUE blah ...

Check Github

```
$token = uniqid(); // 57eb8c5bbf47b; time

$token = md5(uniqid()); // 41eced92fef729c756...  time

$pwd = substr(md5(uniqid()),0,8);// 41eced92; time

srand((double) microtime() * 1000000);

$token = md5(uniqid(rand())); // time,time

$password = md5(uniqid($session, true));//time,known,time

$password = md5(uniqid(time(), true));// time,time,time
```

Let’s Take a Moment

A microsecond is a *really* short period of time

“Lightning fast” - a lightning flash takes ~200,000 microseconds.

“In the blink of an eye” ~100,000 microseconds

“In a flash” ~1000 microseconds

British Army L115A3 rifle muzzle velocity: 938 m/s

= ~1mm per 1 µs

The Target - Reset

```
<?php // resetPwd.php

date_default_timezone_set("GMT");

...

$pwd = uniqid();

file_put_contents('/tmp/pwd', $pwd );

...
```

The Target - Login

```
<?php // login.php

$pwd = $_GET['password'];

$target = file_get_contents('/tmp/pwd');

if( strcmp( $pwd, $target ) == 0 )

{

        print("Access Granted<br>");

        print("target: $target\\n<BR>");

        print(phpinfo());

}
```

Methodology

Could use - ntp, icmp timestamp, snmp, web app...

RFC 2616: Origin servers MUST include a Date header field in all responses except: 100,101,500,503 or no clock.

If no clock, MUST NOT use expires or last-modified (ie. uncacheable).

But date has a resolution of 1,000,000 µs... (!)

Known Unknowns

Find a script with similar timing to the password reset script.

Request this many times to find the clock diff.

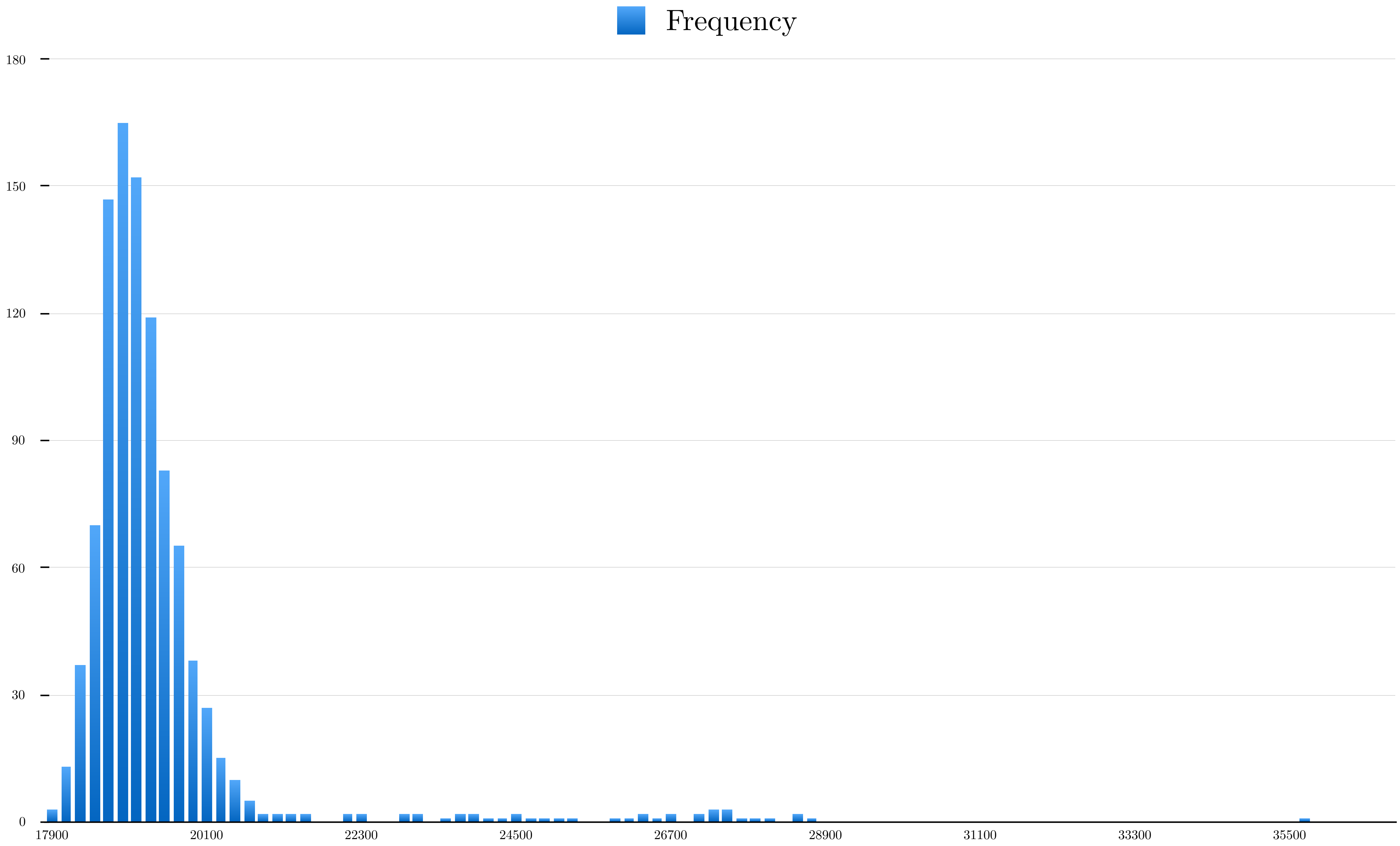
Date resolution is 1,000,000 µs, but there's an edge.

Correct for distance from the edge.

Apply this difference.

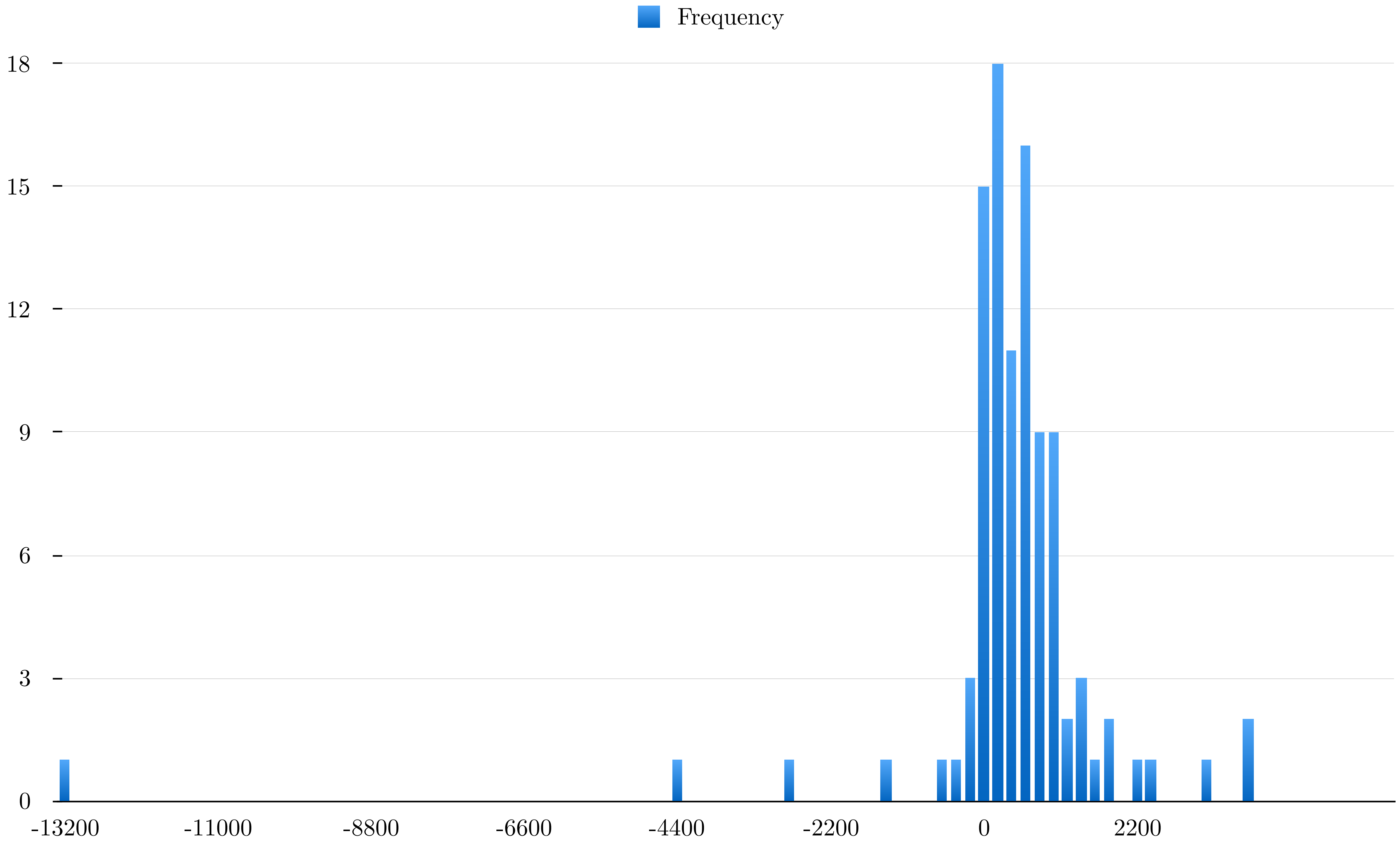
Brute force (0, 1, -1, 2, -2, 3, -3...)

Req Duration - Metropolitan



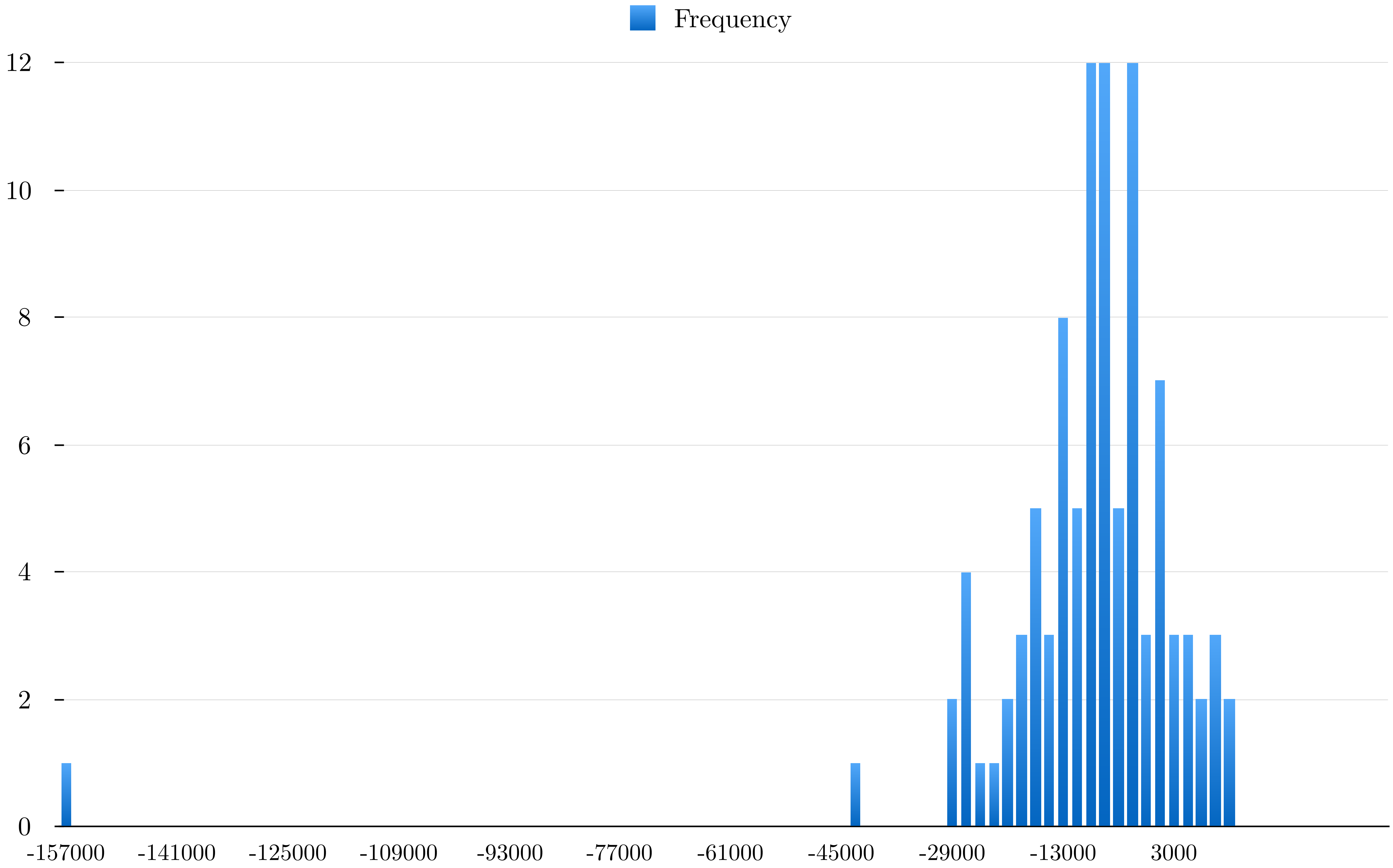
Microsecond Req Duration - Leatherhead to Telecity (SOV), Docklands (~30km) 200 µsec resolution.

Results - Metropolitan



Microsecond Error in Brute Force - Leatherhead to Telecity (SOV), Docklands (~30km)
200 µsec resolution.

Results - Antipodes



Microsecond Error in Brute Force - Leatherhead to Sydney (ec2), ~17000km, 2000 µsec resolution.

But what does it mean?

We can brute force the µs time at which a web script will generate a token in:

LAN: ~500 requests

Metropolitan Area: ~1000 requests (~30 seconds)

Antipodes, tiny server: ~40,000 requests (~1 hour)

...without trying very hard...

Questions?

Improvements:

- Frequency buckets.
- Faster client environment.
- Reliability testing; use a better network.

We haven't talked about:

- Local brute force.
- Millisecond brute force.
- Remote timing attacks in the literature.
- All the many situations in which this is useful...