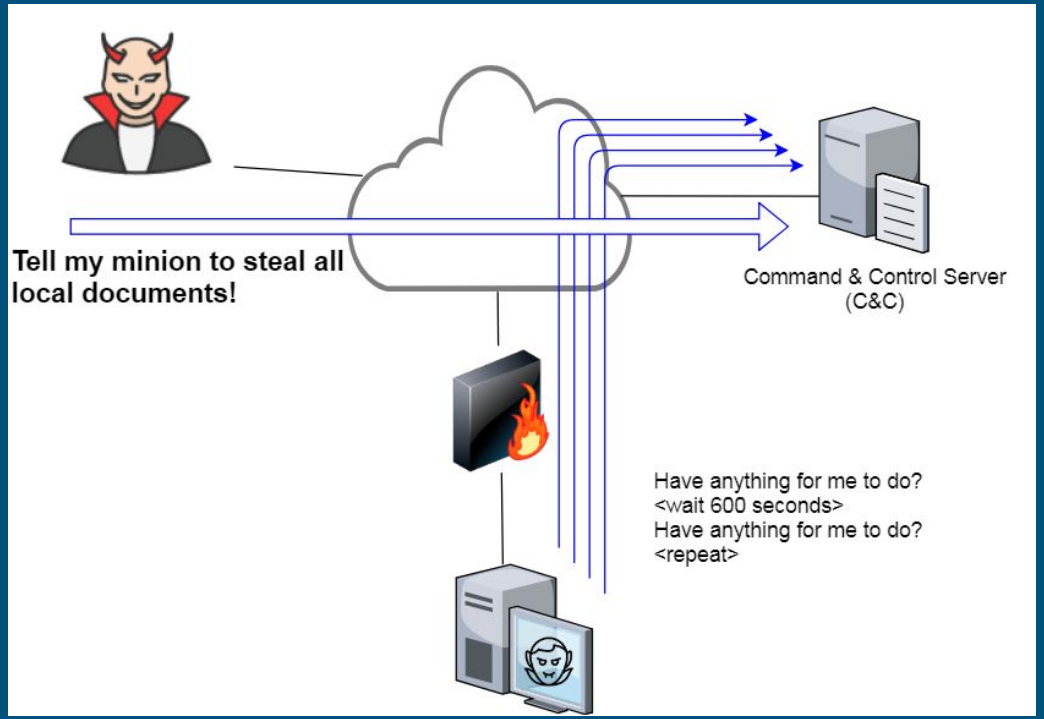
Detecting Periodic Events for Cyber Security

Unimodal case using SQL

What is Command & Control

A Command and Control attack is a type of attack that involves tools to communicate with and control an infected machine or network. To profit for as long as possible from a malware attack, a hacker needs a covert channel or backdoor between their server and the compromised network or machine.

What is Command & Control



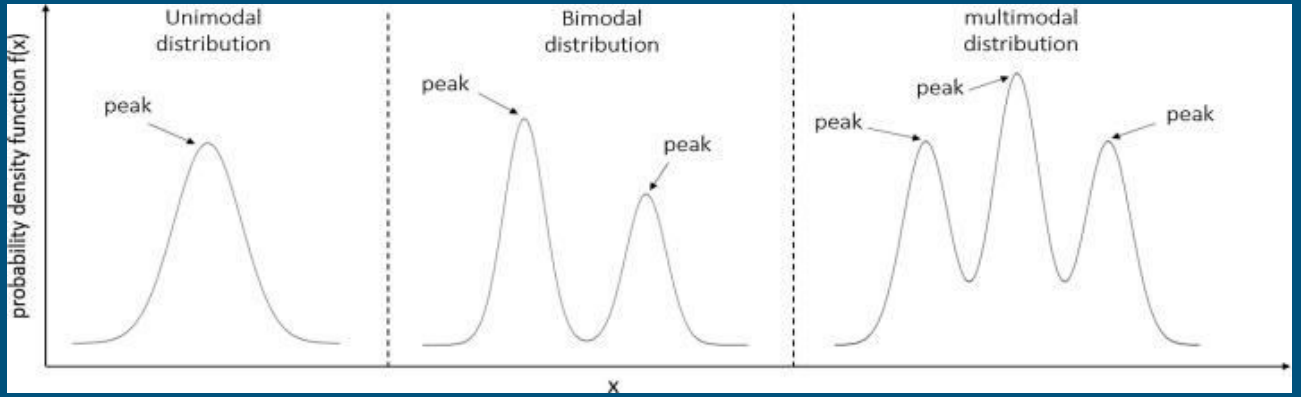
Affected hosts check for commands periodically.

A bit of Math

If a series of events, happening at times $\{t_n\}_{n=0}^{N}$

is periodic, then the time gap between consecutive events $\Delta t_n = t_{n+1} - t_n, n = \overline{0..N-1}$

should have sharp histogram. Either unimodal or multimodal …



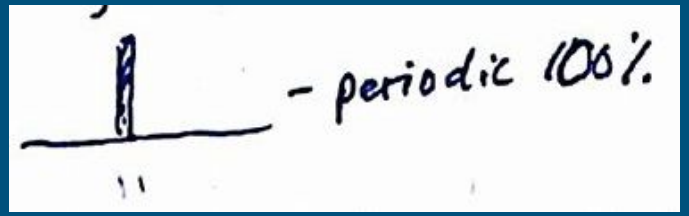
A bit of Math - KISS

Let’s stick to the unimodal version, then statistically this means variance should be very close to zero

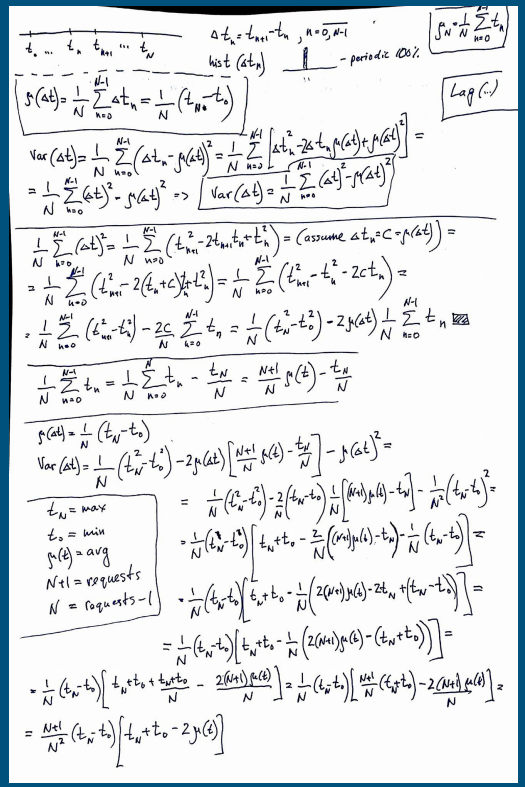
$$\mathrm{Var}(\Delta t) = 0$$

and of course the average is

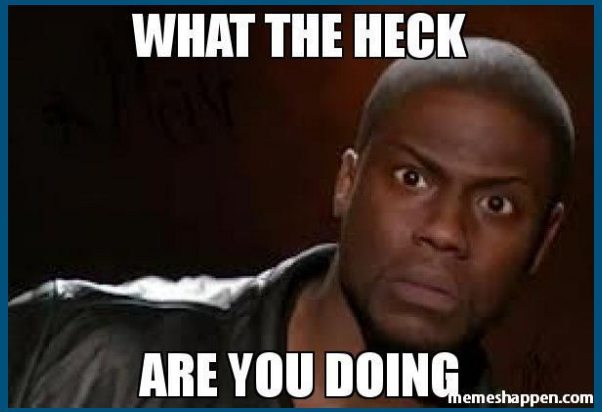
$$\Delta t_n = \mu(\Delta t), \forall n$$



A bit of Math - brain storming



Maths dudling in action …



A bit of Math - clean version

The average is

$$\mu(\Delta t) = \frac{1}{N}\sum_{n=0}^{N-1}\Delta t_n = \frac{t_N - t_0}{N}$$

The variance is

$$Var(\Delta t) = \frac{N+1}{N^2}\left(t_N - t_0\right)\left(t_N + t_0 - 2\mu(t)\right)$$

And the point is?

We can map the calculations above with SQL aggregate functions, like

$$
\begin{aligned}
t_N &= \max \\
t_0 &= \min \\
\mu(t) &= \text{avg} \\
N + 1 &= \text{count}(*)
\end{aligned}
$$

What Database was used?

In this case, events were stored in Amazon Redshift, capable of storing Petabytes of data.

Amazon Redshift is a clone of PostgreSQL. It's a columnar database, offering good compression.

Amazon Redshift competes with Snowflake.

Can we see the SQL now?

```
select
        sourceip, useragent, proxyname, port, site, requests,
        TRUNC(DATEADD(ms, last_seen, '1970-01-01')) as last_date_seen,
        (max_time - min_time)/(requests*1.0 - 1.0) as average,
        abs(((requests*1.0)/((requests - 1.0)*(requests - 1.0)))*(max_time - min_time)*(max_time + min_time - 2*avg_time)) as variance
from (
        select sourceip, useragent, proxyname, port, site,
                count(sourceip) as requests,
                max(startutcinmsraw/(60*1000)) as max_time,
                min(startutcinmsraw/(60*1000)) as min_time,
                avg(startutcinmsraw/(60*1000)) as avg_time,
                max(startutcinmsraw) as last_seen
        from proxy_logs
        where (startutcinms BETWEEN datediff(ms,'1970-1-1',getdate() = interval '10 day') AND datediff(ms,'1970-1-1',getdate()))
                and useragent not like '%Darwin%'
                and useragent not like '%Android%'
                and useragent not like '%iPhone%'
                and useragent not like '%iPad%'
                and useragent not like 'okhttp%'
                and useragent not like 'CloudKit%'
                and useragent not like 'AppleNews%'
                and useragent not like '%iOS%'
                and useragent not like 'Microsoft Office%'
                and useragent not like 'Outlook%'
                and useragent not like 'iTunes%'
                and useragent not like 'MobileAsset%'
                and useragent not like 'Dropbox%'
                and useragent not like 'GmsCore%'
                and useragent not like 'GoogleAuth%'
                and useragent not like 'GoogleMobile%'
                and useragent not like 'WhatsApp%'
                and useragent not like 'Answers%'
        group by sourceip, useragent, proxyname, port, site
)
where requests >= 5
and (max_time - min_time) >= 0.1*(requests*1.0 - 1.0)
and abs((requests*1.0)*(max_time - min_time)*(max_time + min_time - 2*avg_time)) <= 0.01*abs(((requests - 1.0)*(requests - 1.0)))
order by variance asc, last_date_seen desc, average desc, requests desc, sourceip
```

Anything interesting to report?

*	sourceip	useragent	proxyname	port	site	requests	average	variance
1	95.154.244.106	-	c4610a9e58f14bfb.proxy.wandera.com	3128	icanhazip	10	1440.2222222222222222	0E-15
2	95.154.244.106	-	d9a8766eb2644b95.proxy.wandera.com	3129	icanhazip	10	1440.2222222222222222	0E-15
3	95.154.244.106	-	2cb360529cff4292.proxy.wandera.com	3128	icanhazip	10	1440.2222222222222222	0E-15
4	182.92.8.72	Go 1.1 package http	d9a8766eb2644b95.proxy.wandera.com	3129	teaduoduo	10	1439.777777777777777777	0E-15
5	46.36.37.20	Go-http-client/1.1	1508728ab51b4887.proxy.wandera.c...	3128	muni	14	729.384615384615384615	0E-15
6	186.37.202.87	-	c7811861da2744ad.proxy.wandera.com	5611	bloomberg	14	714.9230769230769230769	0E-15
7	87.214.219.246	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.11) Gecko/2009060215 Firefox/3.0.11	4bdfd6604ab444de.proxy.wandera.com	4444	master-iptv	14	693.846153846153846153	0E-15
8	85.255.233.17	-	f519d3603d8641b1.proxy.wandera.com	4157	maps	11	488.200000000000000000	0E-15
9	31.64.231.185	-	f519d3603d8641b1.proxy.wandera.com	3630	fbcdn	12	460.000000000000000000	0E-15
10	166.172.185.229	MobilePhone	d9a8766eb2644b95.proxy.wandera.com	2531	apple	12	371.818181818181818181	0E-15
11	166.172.185.19	Mozilla/5.0	8b6203e042654224.proxy.wandera.c...	2281	waze	10	330.888888888888888888	0E-15
12	93.145.206.201	CYQRJUG9MR.com.paybay.qui	c24837fef834446d.proxy.wandera.com	2910	apple	16	296.933333333333333333	0E-15
13	31.110.158.186	navd	4b5bc33abd26424b.proxy.wandera.com	4677	apple	16	291.866666666666666666	0E-15
14	23.253.105.202	-	18139450fcd44a30.proxy.wandera.com	4444	samair	18	286.352941176470588235	0E-15
15	85.255.233.151	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.102 ...	7f530f61400240b5.proxy.wandera.com	5387	google	16	282.933333333333333333	0E-15
16	174.217.12.51	SafetyNet/11975436 (j3ltevzw NMF26X); gzip	d9a8766eb2644b95.proxy.wandera.com	3440	www	12	273.636363636363636363	0E-15

The team also won a champagne bet, trying to tackle multimodal version with Fourier analysis.