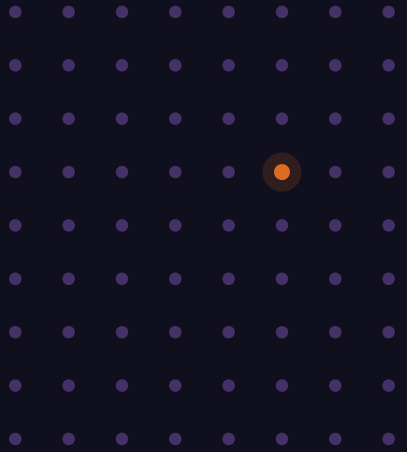
Boring Binaries, Behaving Badly

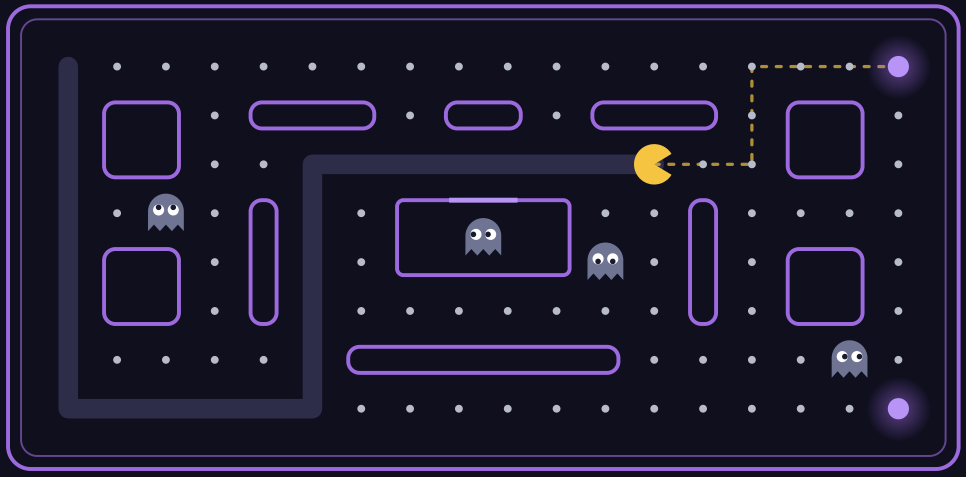
w/

Zane Gittins

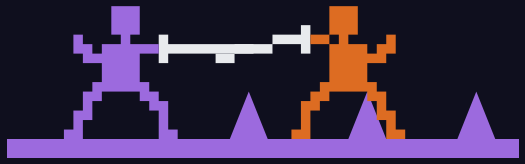


Defenders have to be right every time

Attackers only have to be right once.



Defenders have the home turf advantage



1. Deception

2. Know your environment

Baseline your environment

“Jen uses Quick Assist a lot”

Tribal knowledge



`Jen, quickassist: 30 of 31 days`

Baseline

Everything stands on the data

Detections, baselines, and AI all depend on it.



Start with what you want to catch

Malicious remote management tools.

Sysmon process creation Event ID 1

image

`C:\Users\jen\Downloads\support.exe`

Signed

original_file_name

`ScreenConnect.WindowsBackstageShell.exe`

signer Velociraptor

`ConnectWise, LLC`

Cheap to change

Costly to change

The data you need

Baseline recipe

Choose **Count** **Compare**



Choose what to count

Which RMM tools does each user run?

value group

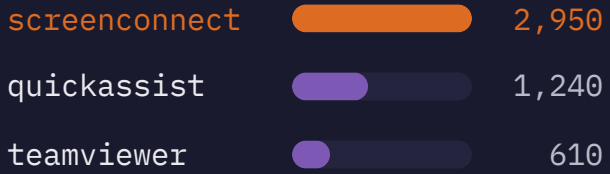
```
ScreenConnect.ClientSetup.exe
ScreenConnect.WindowsBackstageShell.exe
ScreenConnect.WindowsFileManager.exe
```

→ `screenconnect`

Count days, not events

jen's RMM tools: 30 days, plus today.

Count executions



`screenconnect` runs the most

Count days seen



`screenconnect` first seen today

Compare

Is it new?

First seen

or

Is it rare?

Rarity + coverage

Good-Turing, step by step

quickassist	30
teamviewer	24
screenconnect (today)	1
sightings	55

$$\text{coverage} = 1 - N_1 \div N$$

1. Count every sighting — $N = 30 + 24 + 1 = 55$
2. Count things seen once — $N_1 = 1$
3. Chance the next is new — $1 \div 55 \approx 2\%$
4. Coverage — $1 - 2\% = 0.98$

Do it in the SIEM you have

Two scheduled searches. Pair rarity with first seen.

1 **Nightly: count and save** pseudocode

```
process executions, last 30 days
| tool = LOLRMM match on signer,
|   original_file_name, or image
| active_days = count days by user
| days = count days by user, tool
| N = sum(days) by user
| N1 = count(days = 1) by user
| coverage = 1 - N1 / N
| save to lookup baseline
```

2 **Hourly: compare** pseudocode

```
process executions, last hour
| tool = LOLRMM match on signer,
|   original_file_name, or image
| look up user, tool in baseline
| days = baseline days + 1
| rare = days / (active_days + 1)
| where days = 1
|    or (rare < 10%
|        and coverage > 0.9)
```

Make normal boring

Allowlist one RMM tool, and anything else stands out.

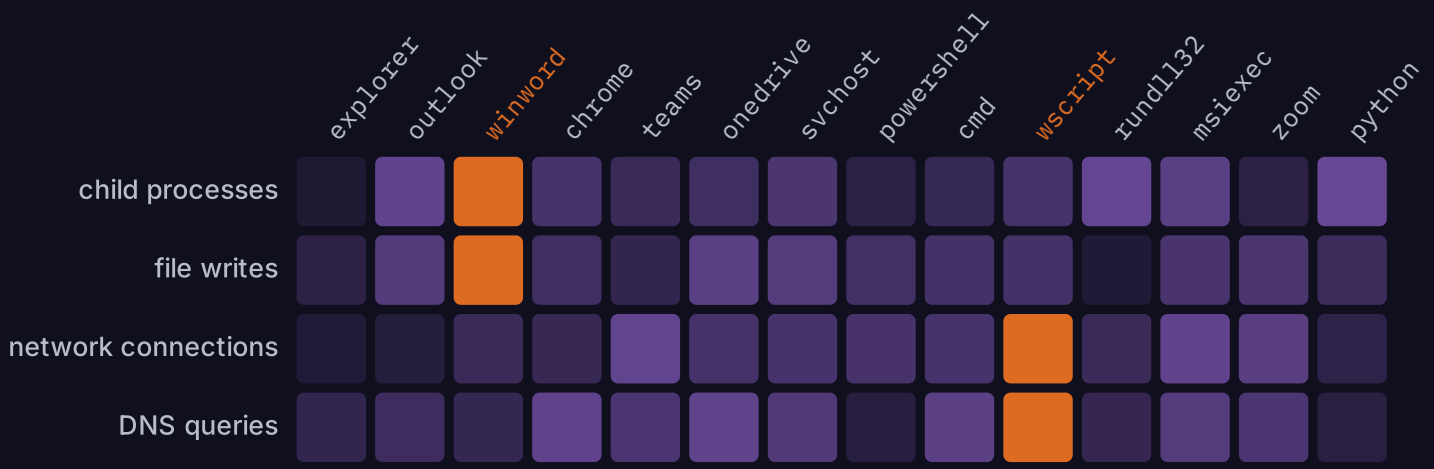
Many RMM tools allowed

- teamviewer
- anydesk
- screenconnect
- splashtop
- quickassist
- rustdesk
- vnc
- atera

One RMM tool allowed

- quickassist
- screenconnect

What if every program had a baseline?



Bifract

An open source SIEM.



- Self-hosted, built on ClickHouse
- Baselines, kept as logs arrive
- Provenance graphs

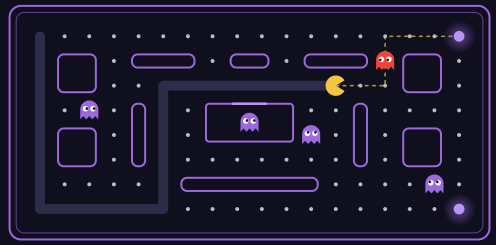
`github.com/zaneGittins/bifract`

Demo

Live in Bifract

TL;DR

1. Know your home turf
2. Everything stands on the data
3. Choose, count, compare



Try the labs yourself **labs.bifract.io**