SIFTING THROUGH THE NOISE

A proactive approach in securing your infrasctructure.

Introduction

- Dejvid Sherri Data Engineer

alphatechs

What does sifting through the noise mean?

* Understanding, categorizing and interpreting very noisy data specifically when it comes web service logs.

Timestamp	IP	Method	Path	Query Parameters	Headers
8:39:49 PM	192.168.1.1	PATCH	/orders	N/A	{"User-Agent":"Mozilla/5.0 (W...
8:39:47 PM	8.8.8.8	DELETE	/products	{"page":"1","limit":"10"}	{"User-Agent":"Mozilla/5.0 (W...
8:39:45 PM	10.0.0.1	PATCH	/api/data	N/A	{"User-Agent":"Mozilla/5.0 (W...
8:39:43 PM	8.8.8.8	DELETE	/orders	N/A	{"User-Agent":"Mozilla/5.0 (W...
8:39:41 PM	172.16.0.1	PATCH	/products	{"page":"1","limit":"10"}	{"User-Agent":"Mozilla/5.0 (W...
8:39:39 PM	172.16.0.1	DELETE	/orders	N/A	{"User-Agent":"Mozilla/5.0 (W...
8:39:37 PM	192.168.1.1	POST	/api/data	{"page":"1","limit":"10"}	{"User-Agent":"Mozilla/5.0 (W...
8:39:35 PM	10.0.0.1	PUT	/api/data	N/A	{"User-Agent":"Mozilla/5.0 (W...
8:39:33 PM	8.8.8.8	GET	/api/data	N/A	{"User-Agent":"Mozilla/5.0 (W...
8:39:31 PM	10.0.0.1	POST	/auth/login	N/A	{"User-Agent":"Mozilla/5.0 (W...

CVE-2012-1823 /hello.world
Argument Injection | PHP CGI
2024-06-14 03:15:43
Browser/122.0
%ADd+allow_url_include%3d1+%ADd+auto_prepend_file%3dphp://input

Unknown /.env
Information Disclosure | Configuration Scanner
2024-06-14 02:23:55
Browser/81.0

Unknown /.vscode/sftp.json
Configuration Scanner | Information Disclosure
2024-06-14 05:05:17
Go-http-client/1.1

CVE-2020-0688 /ecp/Current/exporttool/microsoft.exchange.ediscovery.exporttool.application
Exchange Control Panel | Remote Code Execution
2024-06-14 05:05:18
Go-http-client/1.1

CVE-2017-9841 /phpunit/src/Util/PHP/eval-stdin.php
PHPUnit | Remote Code Execution
2024-06-14 19:51:48
Custom-AsyncHttpClient
Body: <?php echo(md5("Hello PHPUnit"));

High level overview

Data Collection

Using Sensors to collect formatted logs for easier parsing.

Data Storage

Storing the logs in a database, depending on the purpose and scale, PostgreSQL , MariaDB/MySQL or NoSQL. Essentially any database will do.

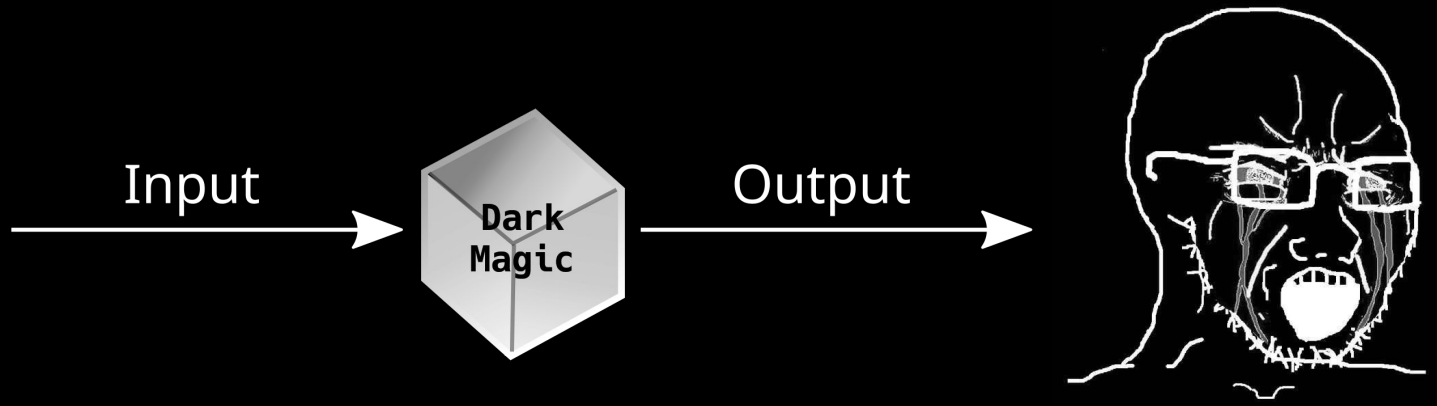
Data Processing

Processing using different methods to make the data interpretable. These methods can vary depending on the data you are working with.

Interpretation

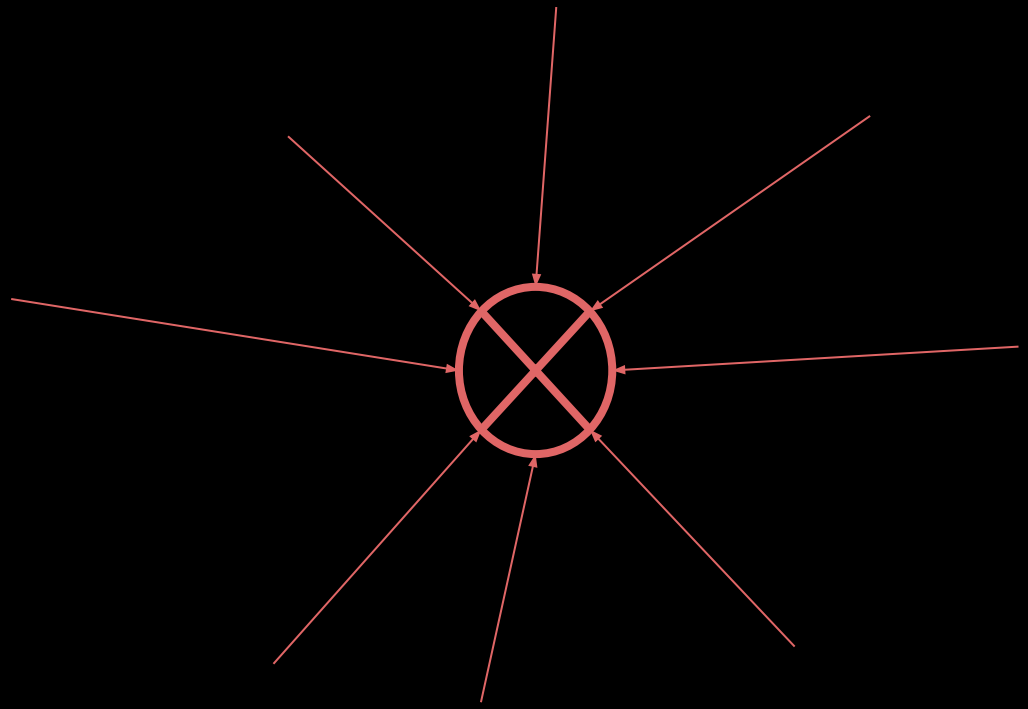
Using different visualization methods we can make this data easily consumable and understandable by a wider range of audience.

How most people imagine data engineering works.



Replicas, what are they?

This is our data collection “step”



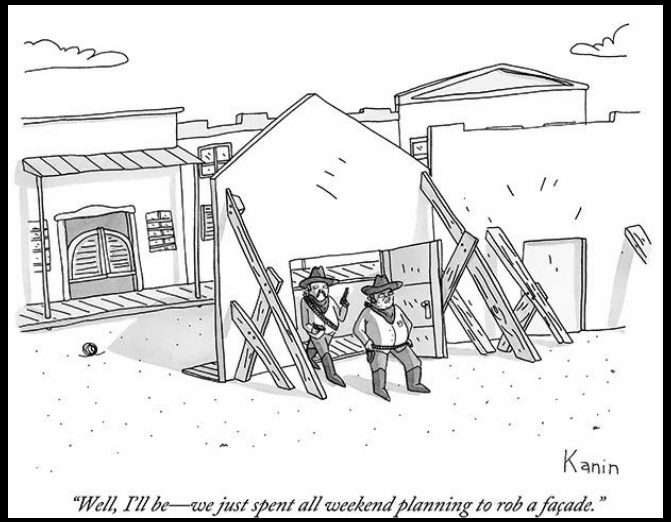
Replica

A decoy system or service that monitors and logs incoming web and network traffic.

Not just an open port!

- Replica’s can be as advanced or as simple as you want them to be depending on the task.

- My rule of thumb is that a Replica should try to be as “realistic” as it needs to be depending on the data you are trying to collect and the amount of time you are willing to waste for bad actors.



A login screen.

This can be a way to entice threat actors into thinking they are dealing with a legitimate system or device.

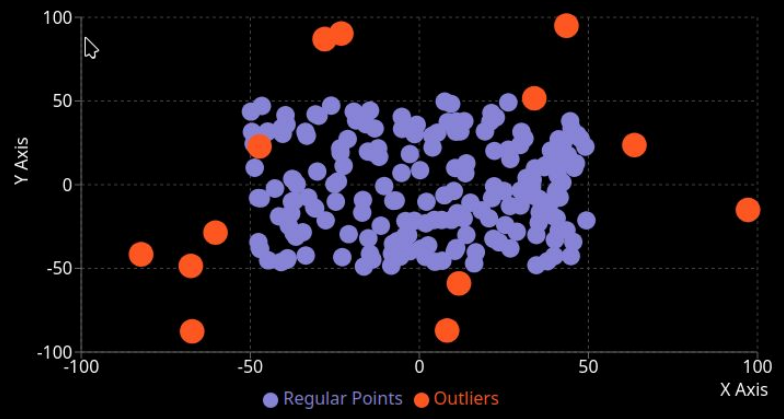
Not everyone will fall prey to this of course.

With enough effort someone experienced will unmask even the most well built Replica.

* Another fun trick is to see what kind of response an exploit will have when successfully exploited on a server and emulating that response and watch as you waste a threat actor's time & compute while their C2 keeps retrying to ping the beacon!

How to make sense of the noise?

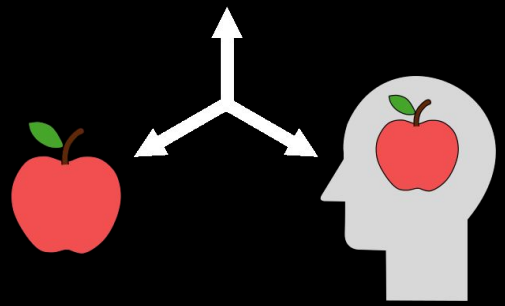
Anomaly Detection (?)



* Doesn't work very well for systems which don't have a baseline for what is a regular and outlier request.

We need semantic context!

apple



Identifying the payload locations.

All the highlighted points can contain some sort of exploit and sometimes the exploit is a combination of two or more.

- Timestamp
- Client IP
- Method
- Path
- Query Parameters
- Headers
- Cookies
- Form Data
- Files
- Body
- Json

CVE-2024-4577 - Endpoint: /hello.world

Argument Injection | PHP CGI

DATETIME	PATH	USER AGENT	REQUEST QUERYSTRING
2024-09-17 11:38:23.650456	/ hello.world	Custom-AsyncHttpClient	%ADd+allow_url_include%3d1+ %ADd+auto_pre...

Form Data


```
{
  "<?php
shell_exec(base64_decode(\"b2sgbm93IHUgZGVjb2RlZCBtZSBidXQgd2h5Pw==\")); echo(md5(\"Hello CVE-2024-4577\")); ?>": "\"")); echo(md5(\"Hello CVE-2024-4577\")); ?>"
}
```


* In this example you can see that the exploit itself is being done on the Request Querystring and Form Data is being used to hold the shell.

Researching some more…

DBSCAN

CLUSTERING

Isolation Forest

K-Means

Dimensionality Reduction

Statistical Analysis

Embeddings

NLP

TF-IDF

Regex Patterns

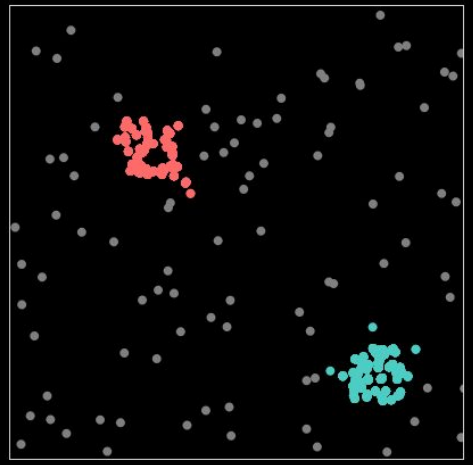
Time Series Analysis

LINK TO ARTICLE

Clustering is the answer, right?

- After considering many different methods i decided that clustering this data is one of the most important parts of actually being able to analyze it.

- This clustering could essentially reduce the amount of logs that need to be manually checked and labeled from thousands to tens. Instead of having to check each log manually, you get a high level overview which makes it easier to extract semantic context from it.



Try all of the methods!

- Needing to try many different methods I realized that simply saving my results in a table and querying to see if the methods worked wasn't going to be enough.
- So I put together this simple GUI visualization built with PyQt and some PyOpenGL!
- I also really wanted a method to easily tweak values such as the eps and max samples through a slider instead of changing the code

Nothing was working, yet!

After extensively changing from different methods, of feature extraction and clustering and tweaking the eps manually and picking specific embedding models, it seemed like nothing was working.

More research, was needed. I started looking into methods that had already implemented some way of clustering logs through DBSCAN and I discovered this specific paper which had developed a method that would essentially optimize the eps automatically and use a specific embeddings model that was all-mpnet-base-v2.

LINK TO PAPER

Find your path!

This was an extremely short overview of how this process of creating threat intelligence systems can work.

As with many issues when it comes to interpreting data some things have been already solved which leaves you to pick and choose what you think works best for your use case.

Some promising concepts to look into would be :

- Fine-tuning of a base embeddings model.
- Regex to detect common vulnerability exploits
- Automatic tagging through Language Models

Thank you!