SPECTEROPS

Hunting SMB Shares

With Data, Graphs, Charts, and LLMS

Scott Sutherland

NetSPI™

Scott Sutherland

(nullbind)

GitHub: nullbind
X: @_nullbind
Bsky: @nullbind.bsky.social

VP of Research at NetSPI

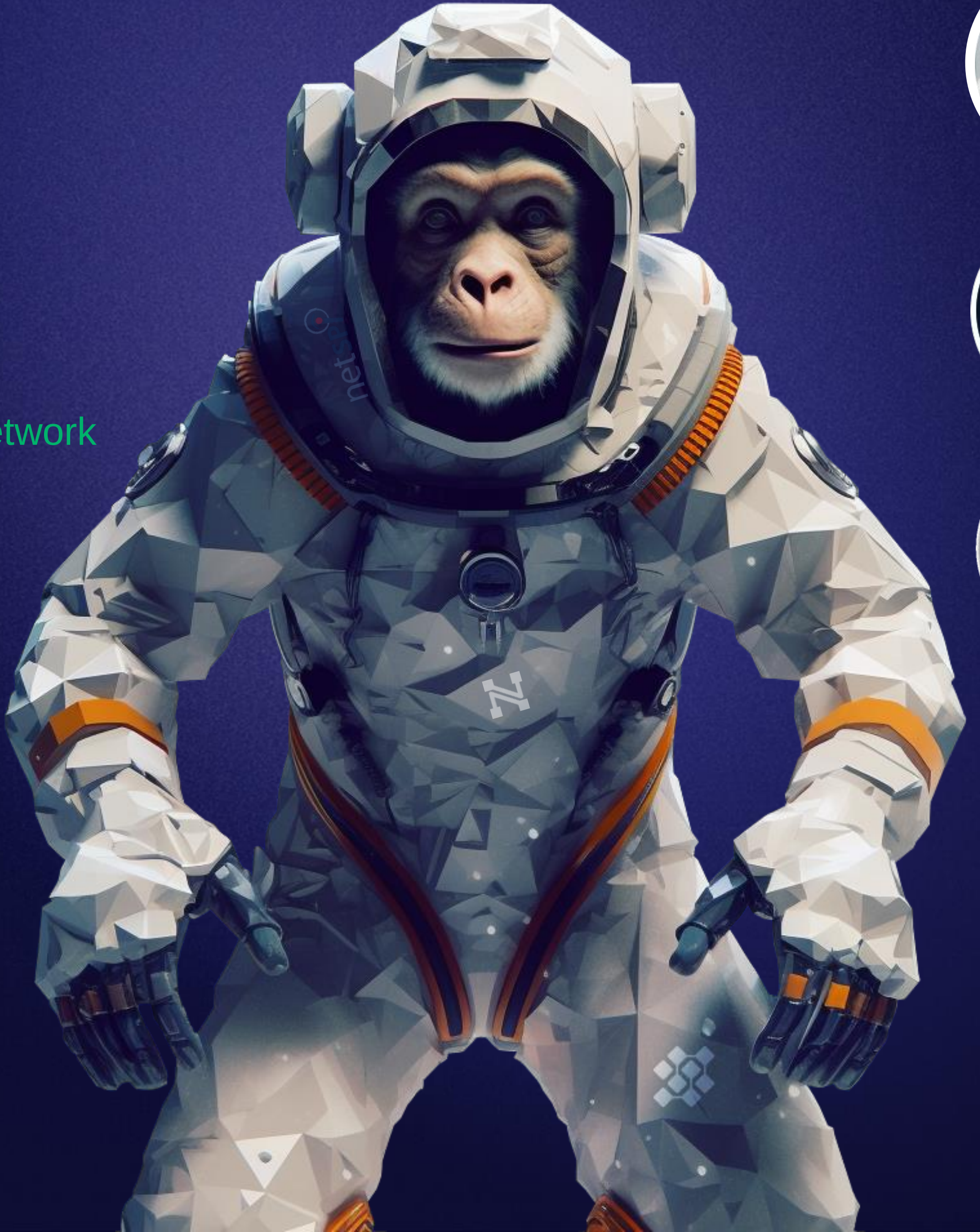
Service & Product Development
Find, exploit, and detect things that go boom on your network

GitHub Projects

github.com/netspi/PowerHuntShares
/PowerUpSQL
/DAFT
/SQLC2
/PowerHunt
/PowerShell/Crypt-It

Blogs

https://www.netspi.com/authors/scott-sutherland/



Two Parts
One Story

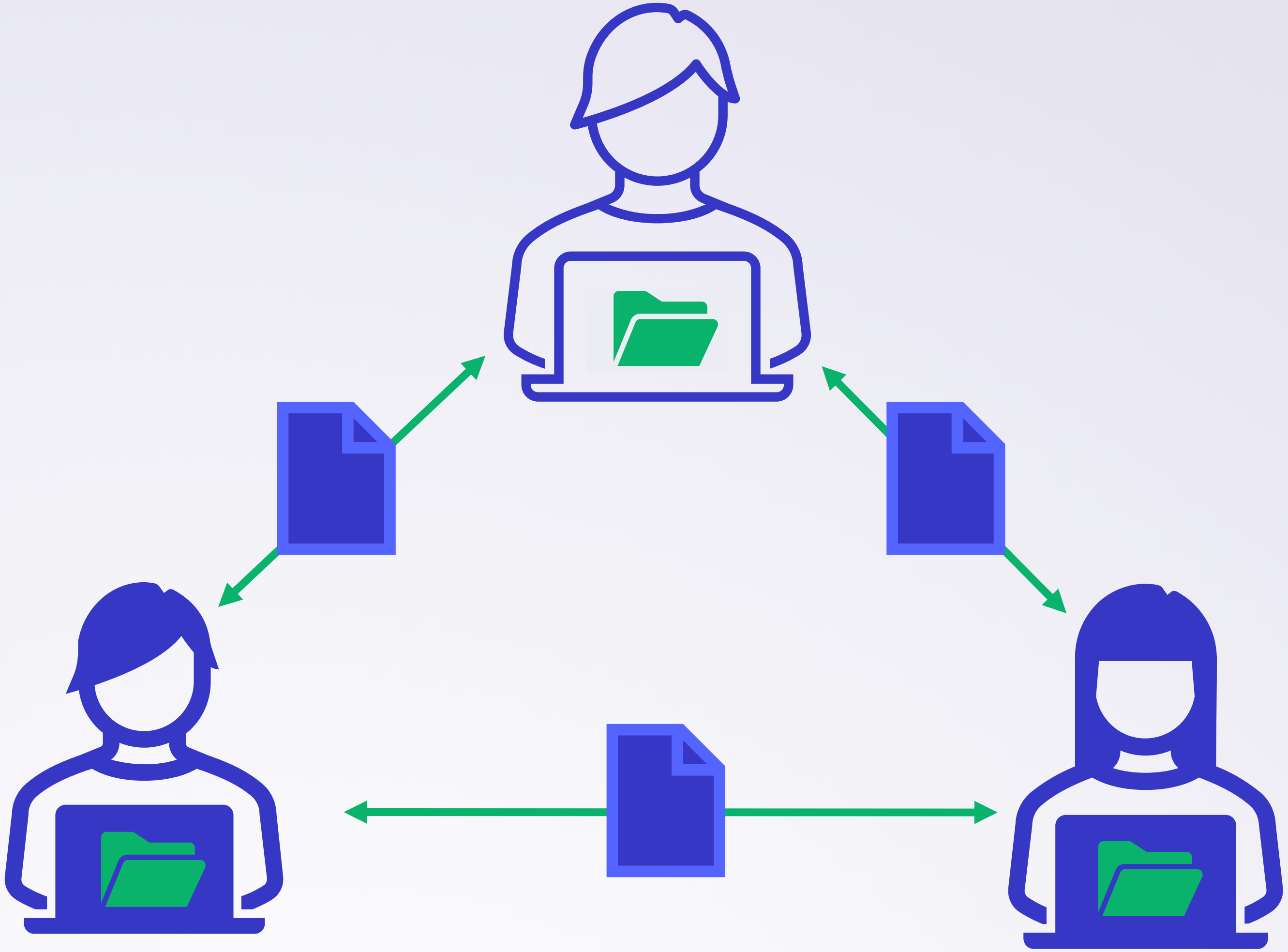
1. A legacy of excessive privileges.
2. Hunting for context in a sea of share data.

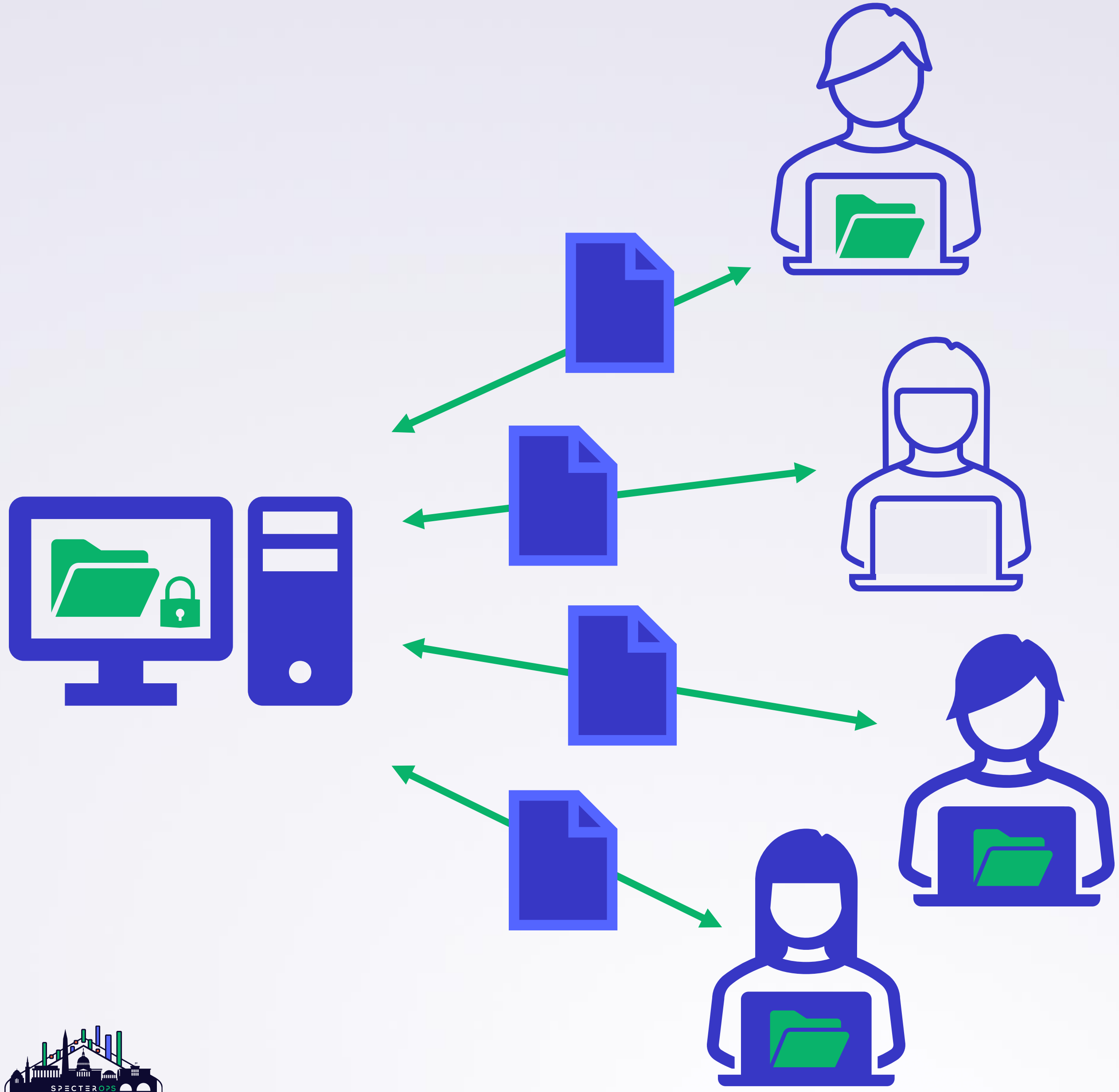
Story Time

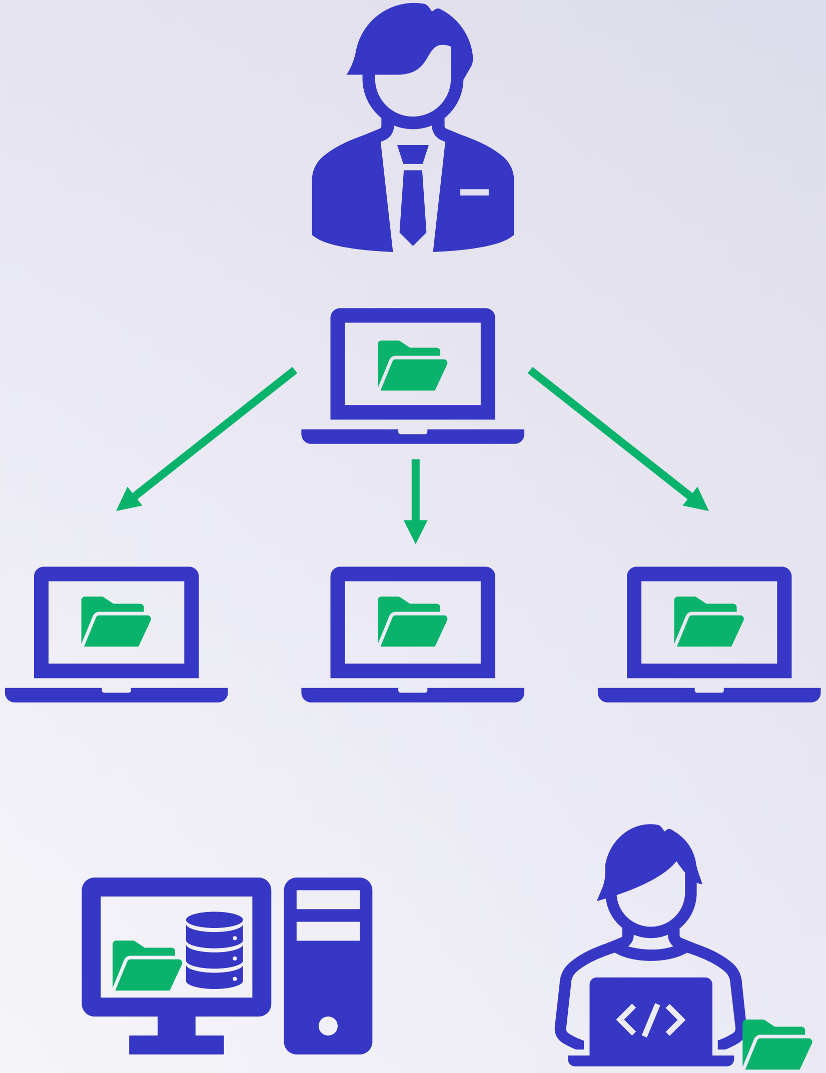
A legacy of excessive privileges.

Story Time

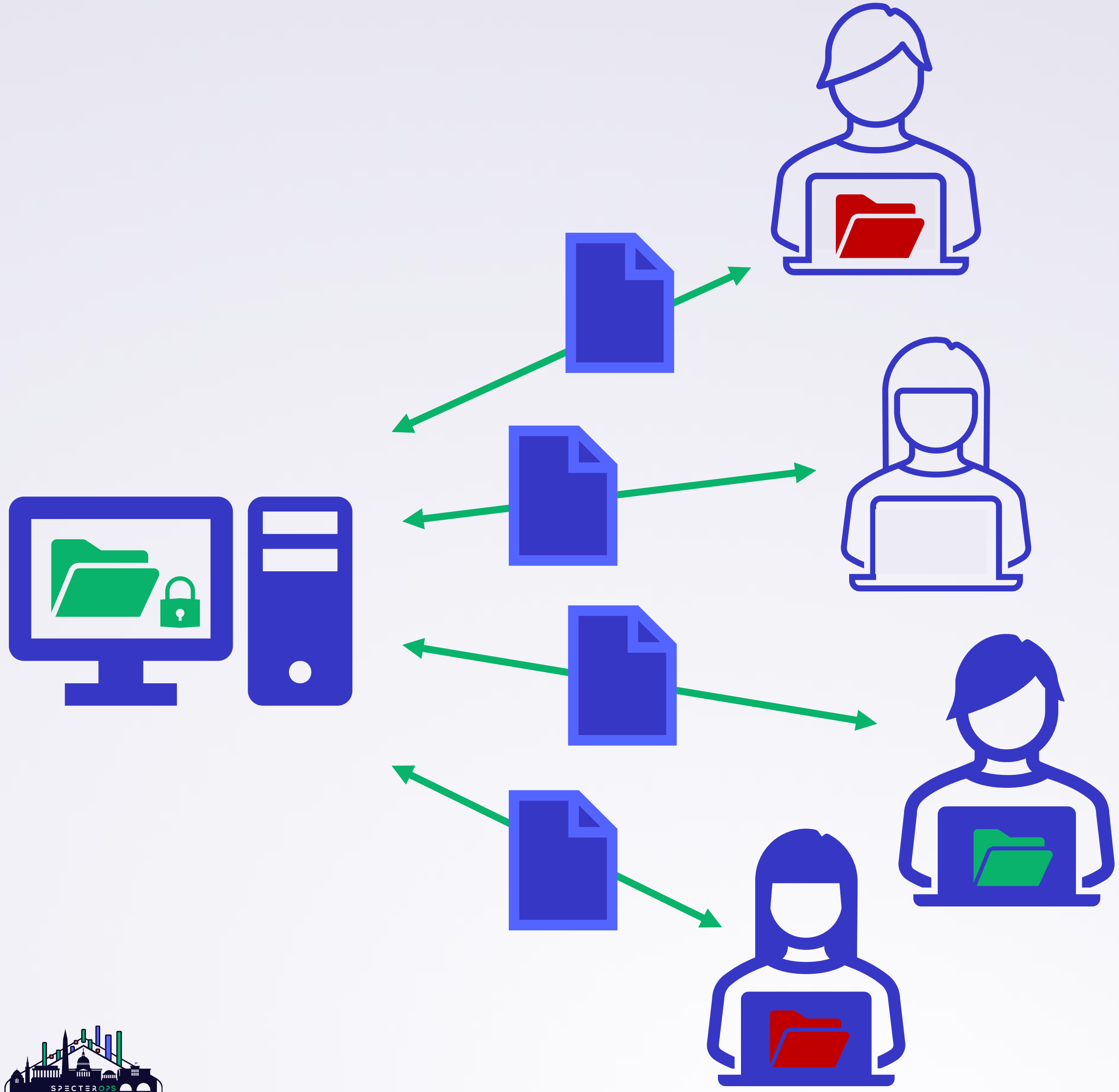
Story Time

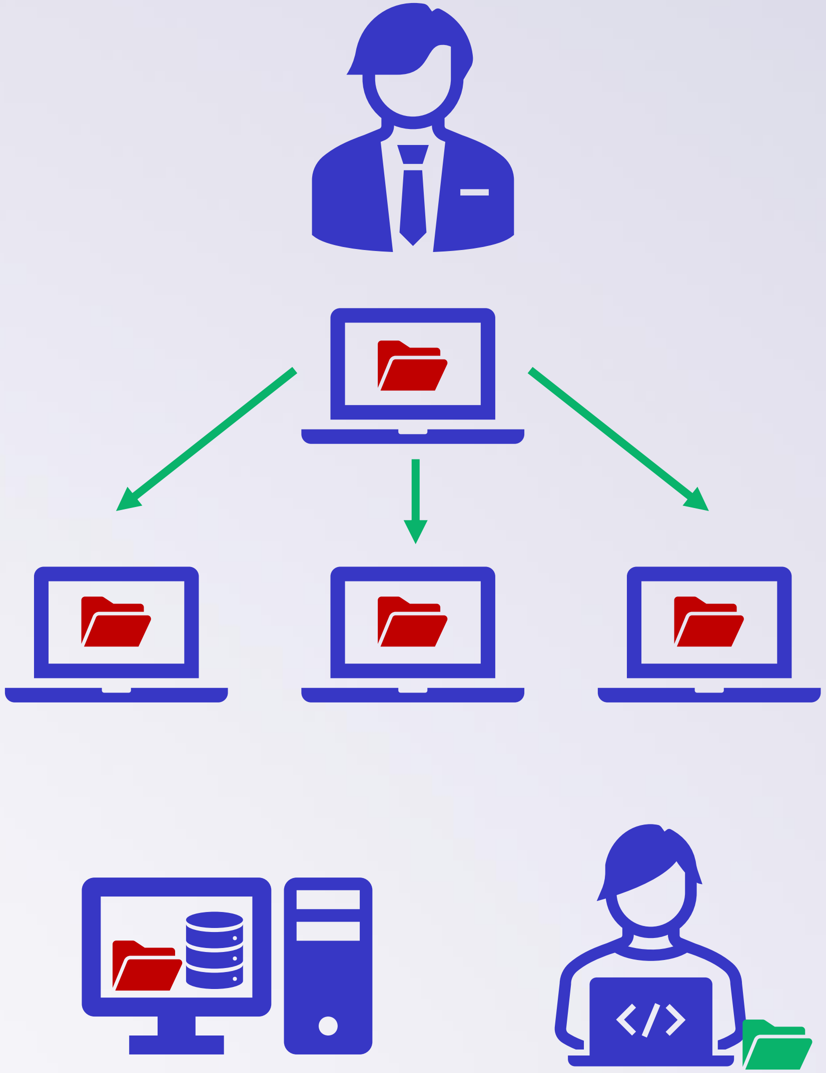




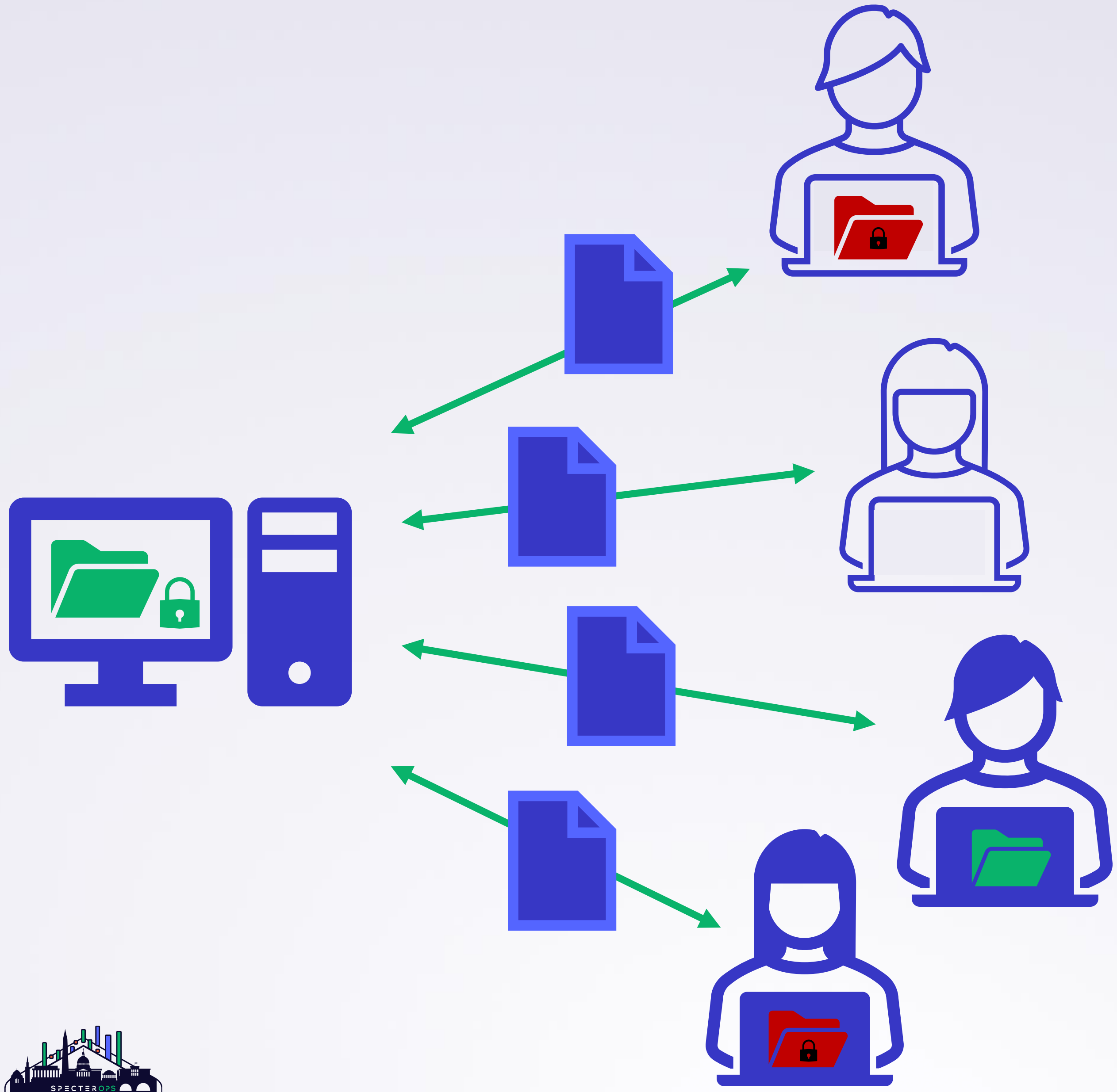


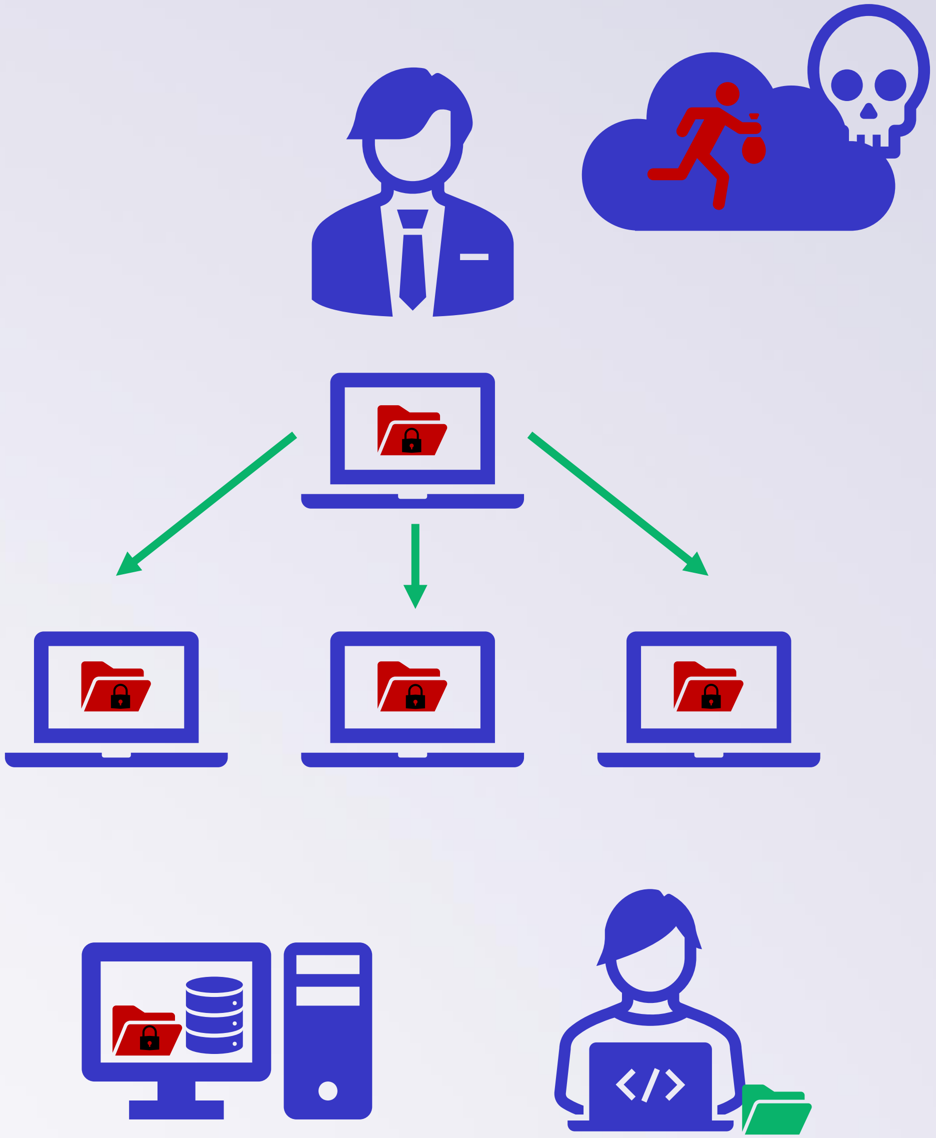
Story Time



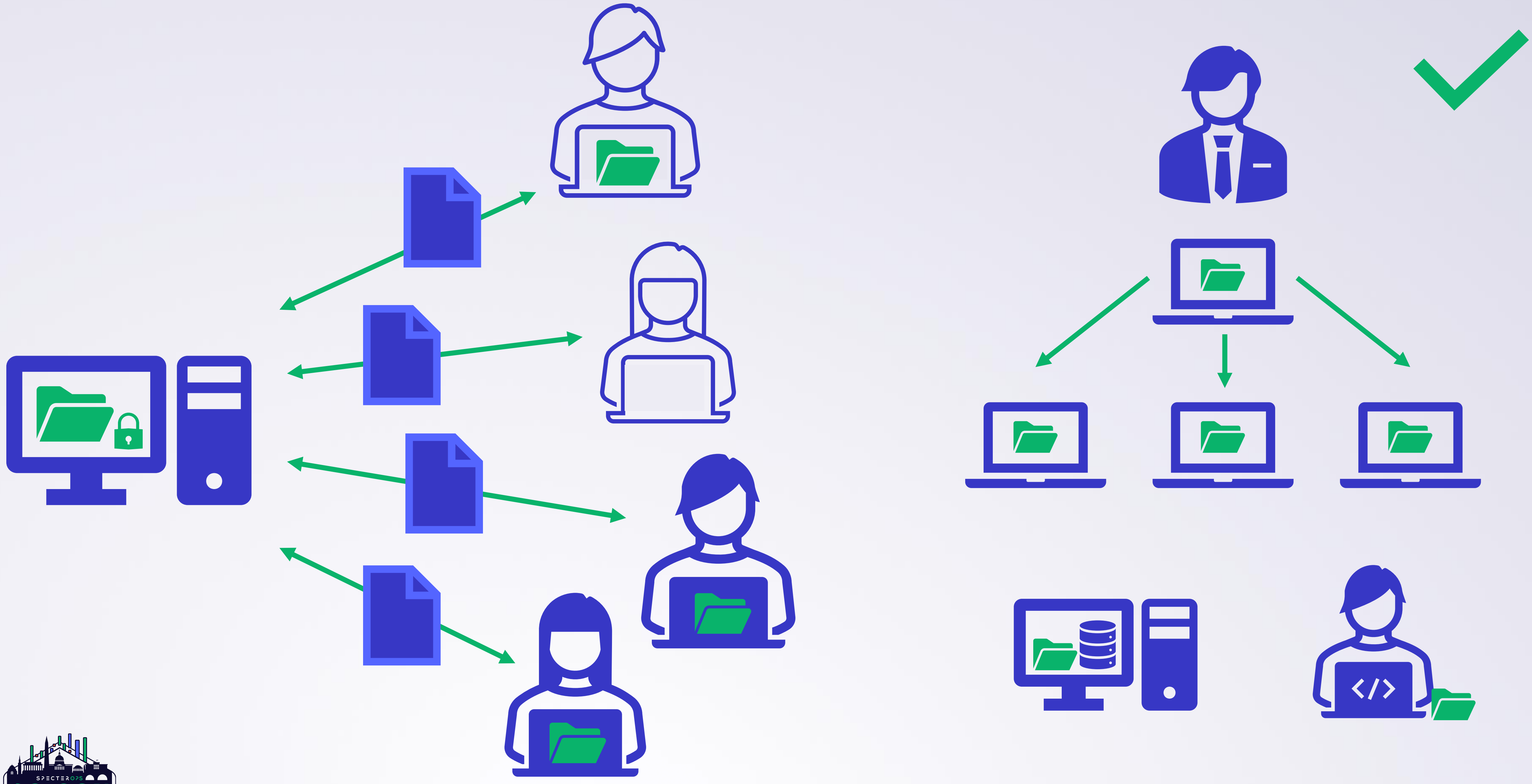


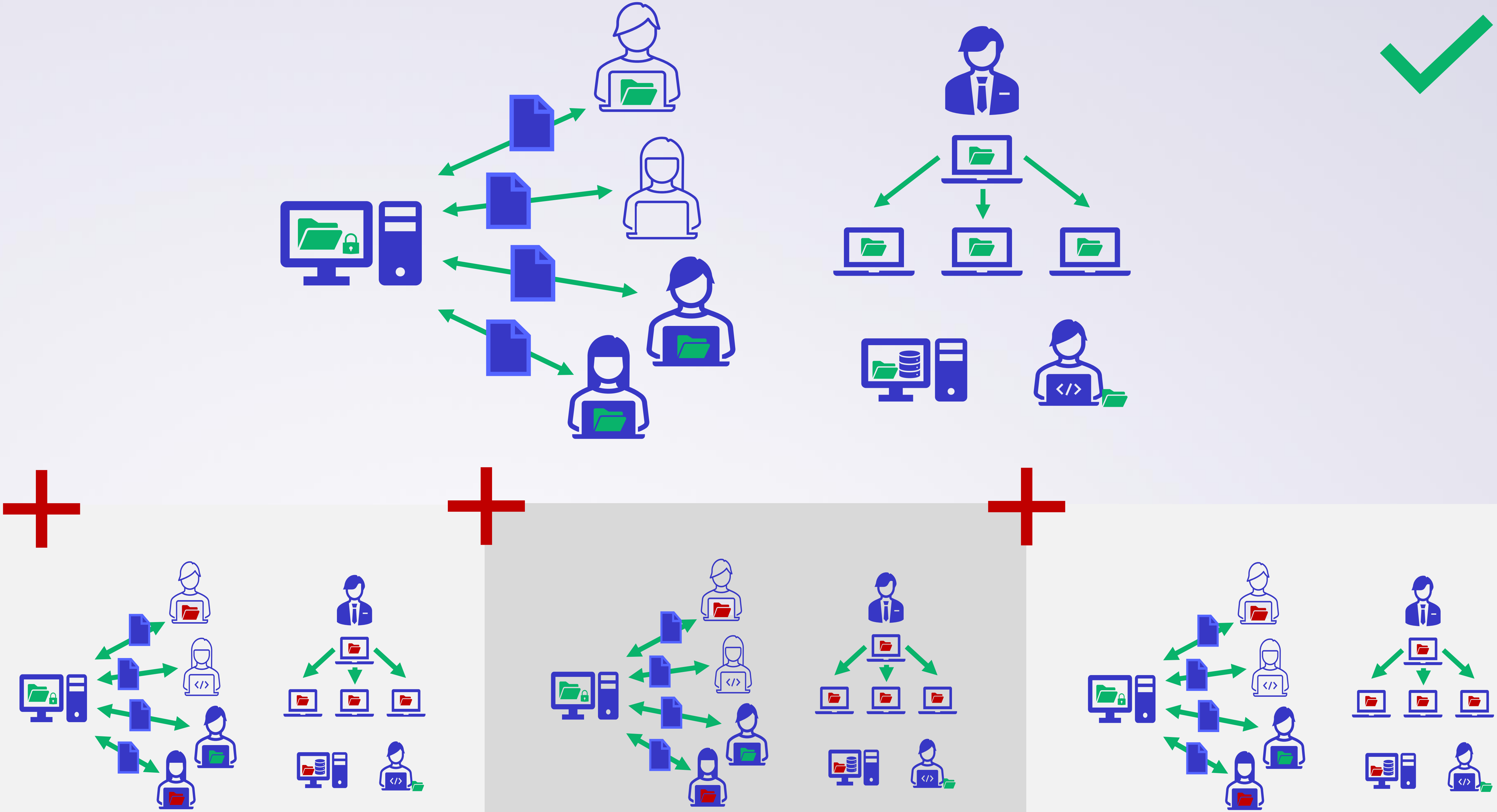
Story Time





Story Time





This is a reality that a lot of businesses are trying to manage. Still.

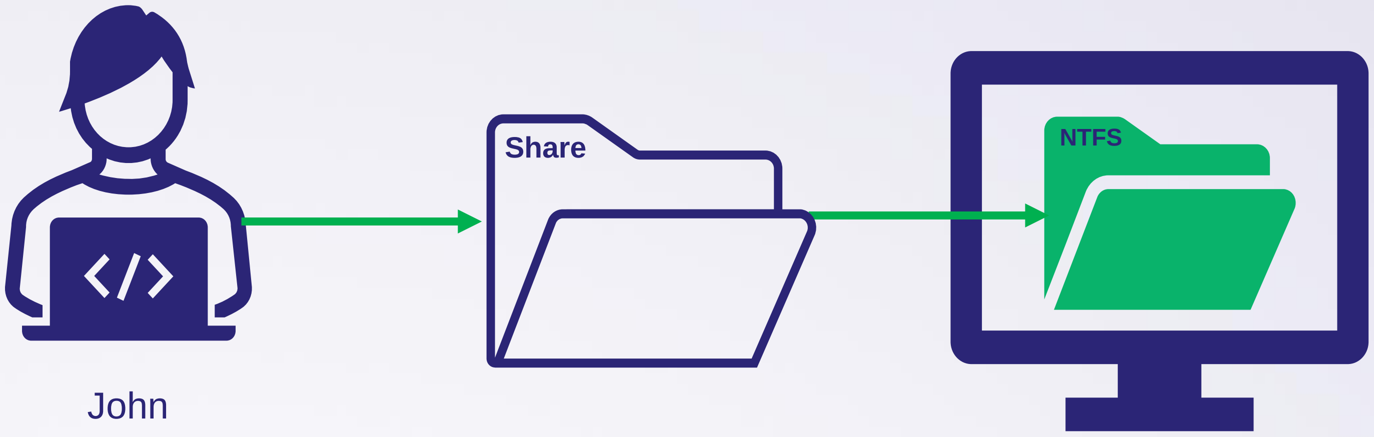
What’s broken, why are we missing so much now?

What’s broken, why are we missing so much now?

- Incomplete inventory
- Insufficient vulnerability scanning
- Privilege inheritance and nested groups
- Generally understanding share context
- **Managing permissions at scale is hard!**

So How do SMB Share Permissions Work?

NTFS & Share Permission
Most Restrictive Wins



Share Permissions

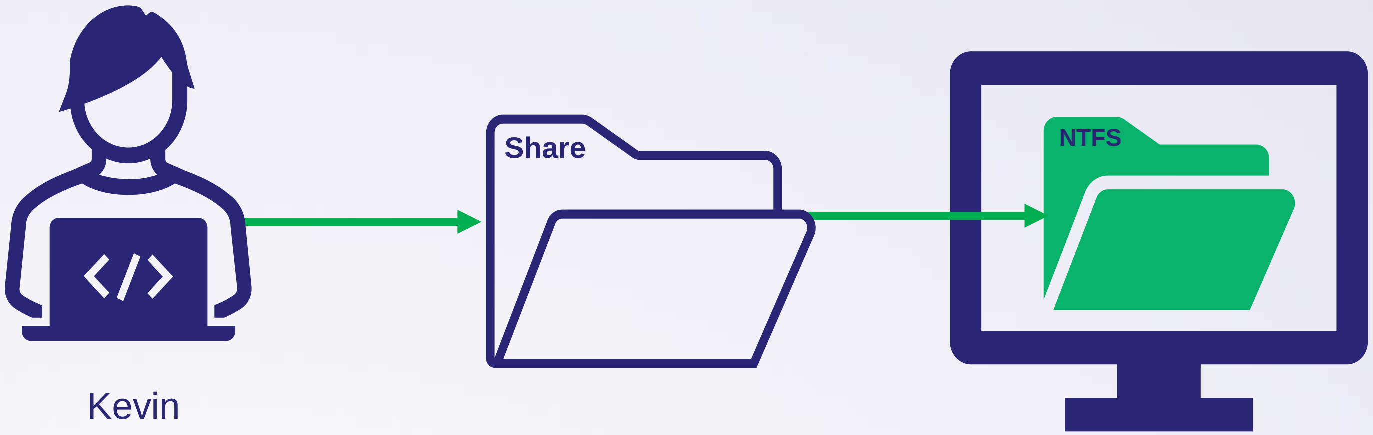
John:	**Read**
Sue:	Write
Kevin:	Full Control

NTFS Permissions

John:	**Full Control**
Sue:	Change
Kevin:	Read

NTFS & Share Permission

Most Restrictive Wins



Kevin

Share

NTFS

Share Permissions

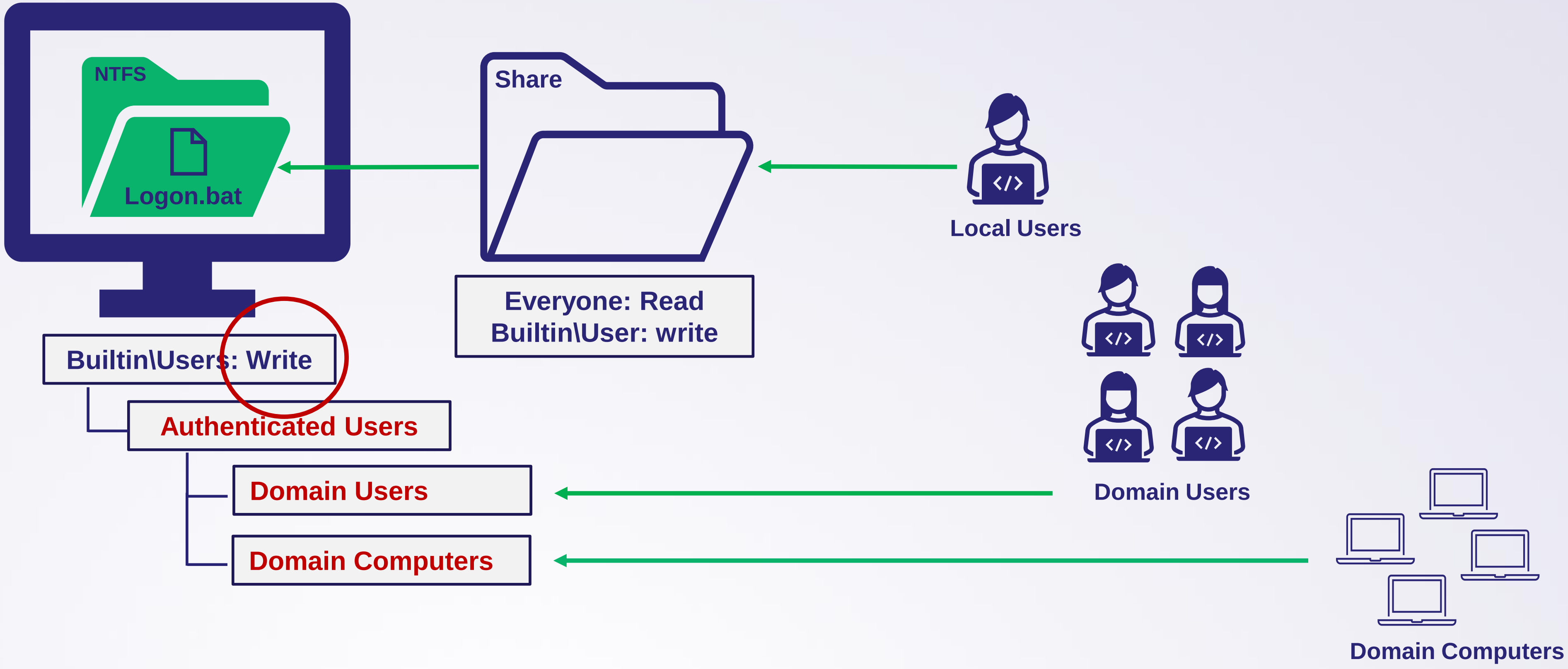
John:	Read
Sue:	Write
Kevin:	**Full Control**

NTFS Permissions

John:	Full Control
Sue:	Change
Kevin:	**Read**

Default Inherited Permissions

Are. The. Worst. …Best?

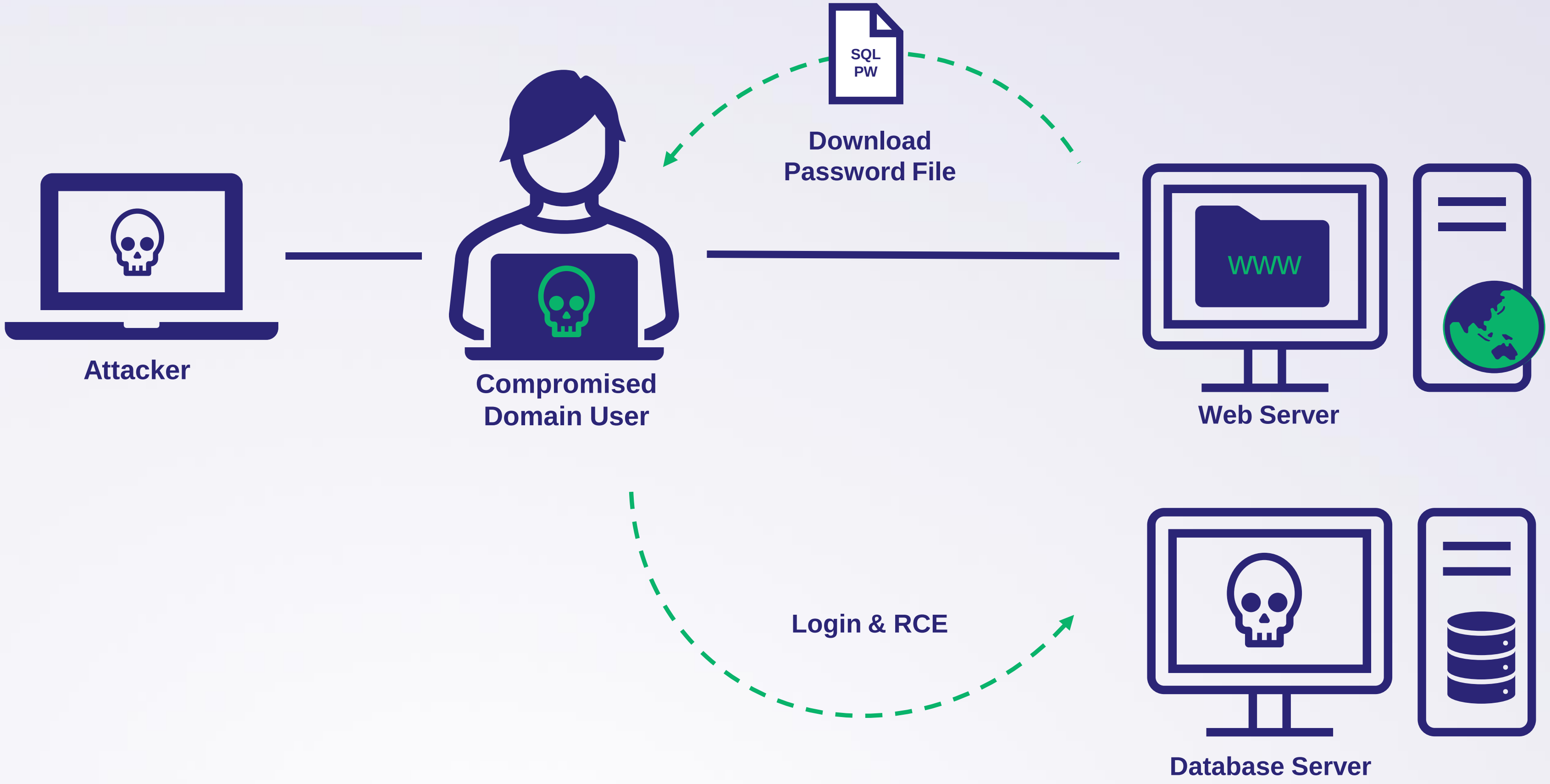


What’s the impact, what can attackers do?

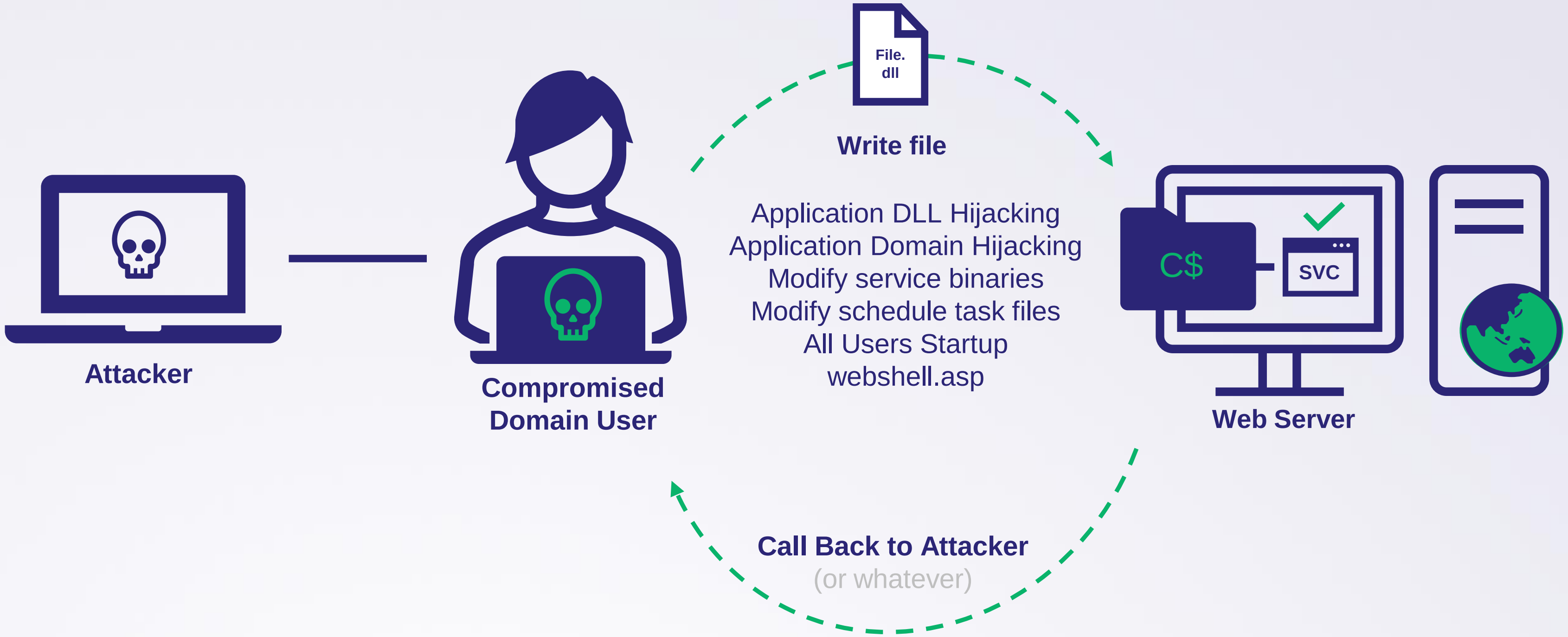
- **Read** data they shouldn’t be able to
- **Write, Modify, Delete** data they shouldn’t be able to
- **Execute Code Remotely**…

Attacking Shares

Read Access.



Attacking Shares
Write Access.



Write file

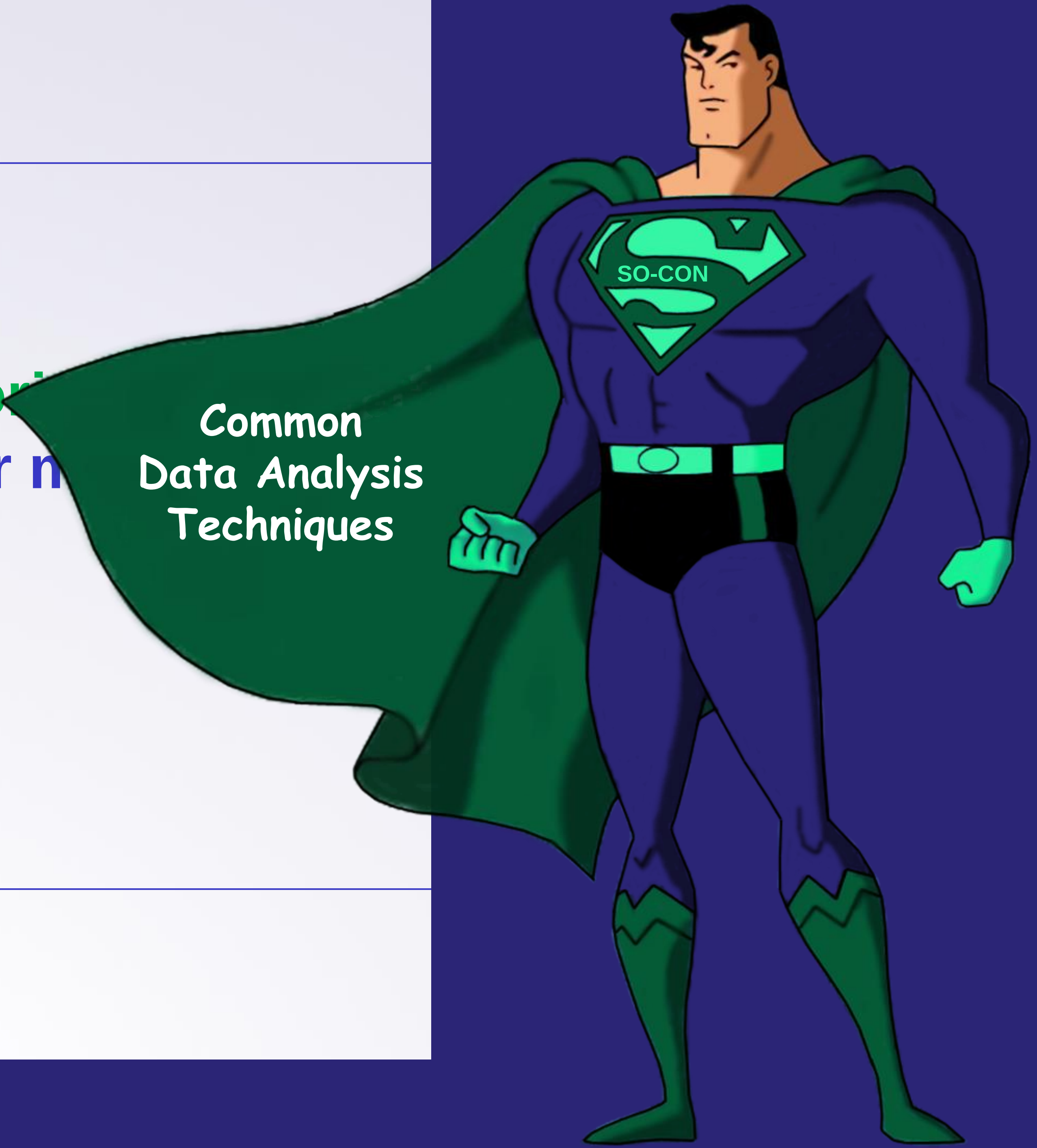
Application DLL Hijacking
Application Domain Hijacking
Modify service binaries
Modify schedule task files
All Users Startup
webshell.asp

Call Back to Attacker
(or whatever)

Shares are one of the
MOST abused attack surfaces but require the
LEAST amount of knowledge to attack

How do we determine which share exposures represent actual risk?

How do we priori
100,000 or n



Hunting for context in a sea of share data

…while building PowerHuntShares v2

What is PowerHuntShares?

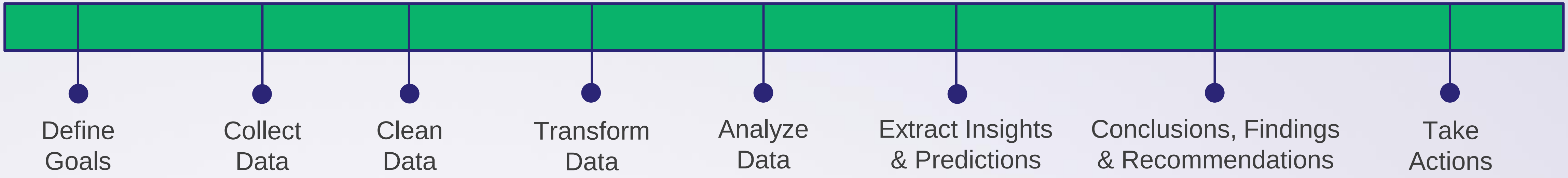
https://github.com/NetSPI/PowerHuntShares

*“**PowerHuntShares** is PowerShell tool designed to help cybersecurity teams and penetration testers better identify, understand, attack, and remediate SMB shares in the Active Directory environments they protect.”*

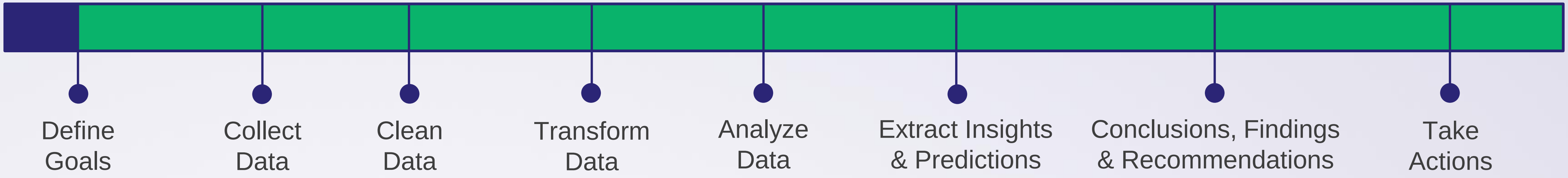
Key Features

- Find Shares with Excessive Privileges
- Find RCE
- Find Data Exposures
- Find & Extract Secrets
- Add context through data enrichment
- Gain insights to prioritize and drive action!

PowerHuntShares Process



PowerHuntShares Process



Define Goals

Collect Data

Clean Data

Transform Data

Analyze Data

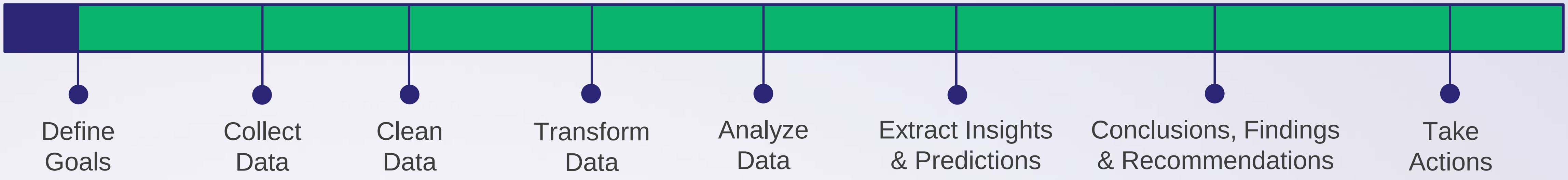
Extract Insights & Predictions

Conclusions, Findings & Recommendations

Take Actions

"Alice created the 'MyApp$' share on 200 systems to support the SuperPOS3k application on 4/1/2025. The shares were configured excessive read/write privileges which exposed sensitive data and provided a means to execute remote code."

PowerHuntShares Process



Define Goals

Collect Data

Clean Data

Transform Data

Analyze Data

Extract Insights & Predictions

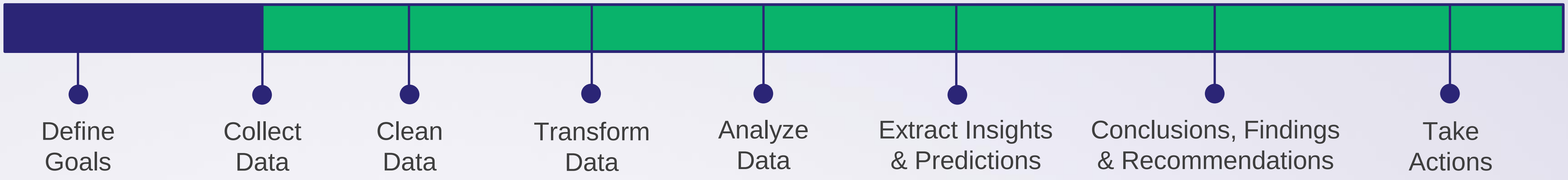
Conclusions, Findings & Recommendations

Take Actions

Goals: **Who, What, When, Where, Why, How**

- **What Happened?** ✓
  Descriptive Analysis
- **Why did it happen?** ✓
  Diagnostic Analysis
- **What will happen?** ✓
  Predictive Analysis
- **What should I do?** ✓
  Prescriptive Analysis

PowerHuntShares Process



Data Collection

- **Asset Coverage**
  Active directory query + port connectivity tests + optional ping test

- **Data Visibility**
  Names, dates creation, last modified, and last accessed dates
  Directory listings, hashes of directory listings, file counts

Bypass. Download. Run.

https://github.com/NetSPI/powerhuntshares

```
# Bypass execution policy restrictions
Set-ExecutionPolicy -Scope Process Bypass

# Import module that exists in the current directory
Import-Module .\PowerHuntShares.psm1

or

# Reduce SSL operating level to support connection to github
[System.Net.ServicePointManager]::ServerCertificateValidationCallback = {$true}
[Net.ServicePointManager]::SecurityProtocol =[Net.SecurityProtocolType]::Tls12

# Download and load PowerHuntShares.psm1 into memory
IEX(New-Object
System.Net.WebClient).DownloadString("https://raw.githubusercontent.com/NetSPI/PowerHuntShares/main/PowerHuntShares.psm1")
```

Discovery Output

https://github.com/NetSPI/powerhuntshares

```
------------------------------------------------------------
SHARE DISCOVERY
------------------------------------------------------------
[*][03/01/2021 09:35] Scan Start
[*][03/01/2021 09:35] Output Directory: c:\temp\smbshares\SmbShareHunt-03012021093504
[*][03/01/2021 09:35] Successful connection to domain controller: dc1.demo.local
[*][03/01/2021 09:35] Performing LDAP query for computers associated with the demo.local domain
[*][03/01/2021 09:35] - 245 computers found
[*][03/01/2021 09:35] Pinging 245 computers
[*][03/01/2021 09:35] - 55 computers responded to ping requests.
[*][03/01/2021 09:35] Checking if TCP Port 445 is open on 55 computers
[*][03/01/2021 09:36] - 49 computers have TCP port 445 open.
[*][03/01/2021 09:36] Getting a list of SMB shares from 49 computers
[*][03/01/2021 09:36] - 217 SMB shares were found.
[*][03/01/2021 09:36] Getting share permissions from 217 SMB shares
[*][03/01/2021 09:37] - 374 share permissions were enumerated.
[*][03/01/2021 09:37] Getting directory listings from 33 SMB shares
[*][03/01/2021 09:37] - Targeting up to 3 nested directory levels
[*][03/01/2021 09:37] - 563 files and folders were enumerated.
[*][03/01/2021 09:37] Identifying potentially excessive share permissions
[*][03/01/2021 09:37] - 33 potentially excessive privileges were found across 12 systems..
[*][03/01/2021 09:37] Scan Complete
```

Analysis Output

https://github.com/NetSPI/powerhuntshares

```
---------------------------------------------------------------
SHARE ANALYSIS
---------------------------------------------------------------
[*][03/01/2021 09:37] Analysis Start
[*][03/01/2021 09:37] - 14 shares can be read across 12 systems.
[*][03/01/2021 09:37] - 1 shares can be written to across 1 systems.
[*][03/01/2021 09:37] - 46 shares are considered non-default across 32 systems.
[*][03/01/2021 09:37] - 0 shares are considered high risk across 0 systems
[*][03/01/2021 09:37] - Identified top 5 owners of excessive shares.
[*][03/01/2021 09:37] - Identified top 5 share groups.
[*][03/01/2021 09:37] - Identified top 5 share names.
[*][03/01/2021 09:37] - Identified shares created in last 90 days.
[*][03/01/2021 09:37] - Identified shares accessed in last 90 days.
[*][03/01/2021 09:37] - Identified shares modified in last 90 days.
[*][03/01/2021 09:37] Analysis Complete
```

Share Report Output

https://github.com/NetSPI/powerhuntshares

```
--------------------------------------------------------------
SHARE REPORT SUMMARY
--------------------------------------------------------------
[*][03/01/2021 09:37] Domain: demo.local
[*][03/01/2021 09:37] Start time: 03/01/2021 09:35:04
[*][03/01/2021 09:37] End time: 03/01/2021 09:37:27
[*][03/01/2021 09:37] Run time: 00:02:23.2759086
….
[*][03/01/2021 09:37] SHARE ACL SUMMARY
[*][03/01/2021 09:37] - 374 ACLs were found.
[*][03/01/2021 09:37] - 374 (100.00%) ACLs were associated with non-default shares.
[*][03/01/2021 09:37] - 33 (8.82%) ACLs were found to be potentially excessive.
[*][03/01/2021 09:37] - 32 (8.56%) ACLs were found that allowed READ access.
[*][03/01/2021 09:37] - 1 (0.27%) ACLs were found that allowed WRITE access.
[*][03/01/2021 09:37] - 1 (0.27%) ACLs were found that are associated with HIGH-RISK share names
```

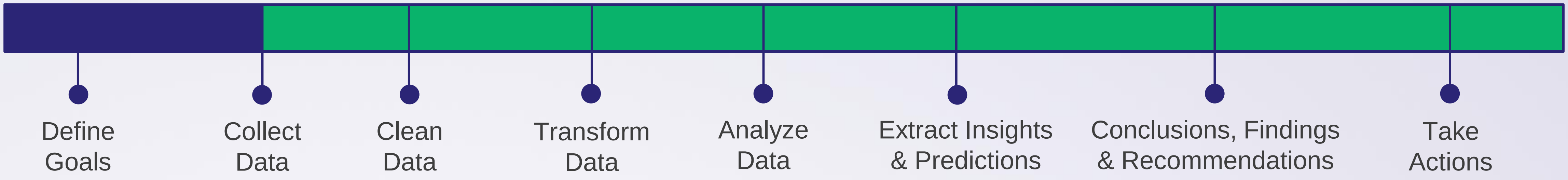
Share Report Output

https://github.com/NetSPI/powerhuntshares

```
-------------------------------------------------------------
SHARE REPORT SUMMARY
-------------------------------------------------------------
[*][03/01/2021 09:37] Domain: demo.local
[*][03/01/2021 09:37] Start time: 03/01/2021 09:35:04
[*][03/01/2021 09:37] End time: 03/01/2021 09:37:27
[*][03/01/2021 09:37] Run time: 00:02:23.27
….
[*][03/01/2021 0
[*][                                    d.
                     (100.00%) ACLs were associated with non-default shares.
                  :37] - 33 (8.82%) ACLs were found to be potentially excessive.
[*][03/01/2021 09:37] - 32 (8.56%) ACLs were found that allowed READ access.
[*][03/01/2021 09:37] - 1 (0.27%) ACLs were found that allowed WRITE access.
[*][03/01/2021 09:37] - 1 (0.27%) ACLs were found that are associated with HIGH-RISK share names
```

BORING! What’s happening under the hood?

PowerHuntShares Process

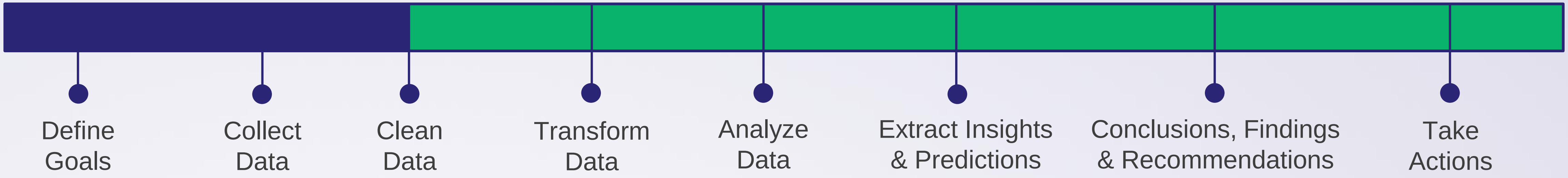


Data Collection

- **Asset Coverage**
  Active directory query + port connectivity tests + optional ping test

- **Data Visibility**
  Names, dates creation, last modified, and last accessed dates
  Directory listings, hashes of directory listings, file counts

PowerHuntShares Process



Define Goals

Collect Data

Clean Data

Transform Data

Analyze Data

Extract Insights & Predictions

Conclusions, Findings & Recommendations

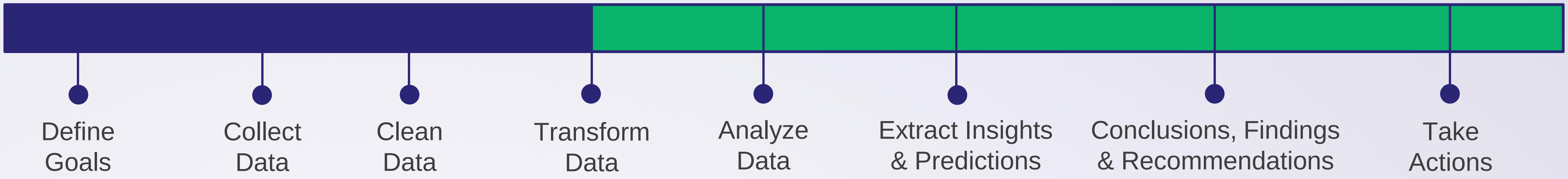
Take Actions

Data Cleaning

Data

- Parse data
- Normalize data structures
- Fix data type errors
- Remove records with errors
- Filter out unneeded data

PowerHuntShares Process



Define Goals

Collect Data

Clean Data

Transform Data

Analyze Data

Extract Insights & Predictions

Conclusions, Findings & Recommendations

Take Actions

Transform Data

Data + Context

Static Data Labeling

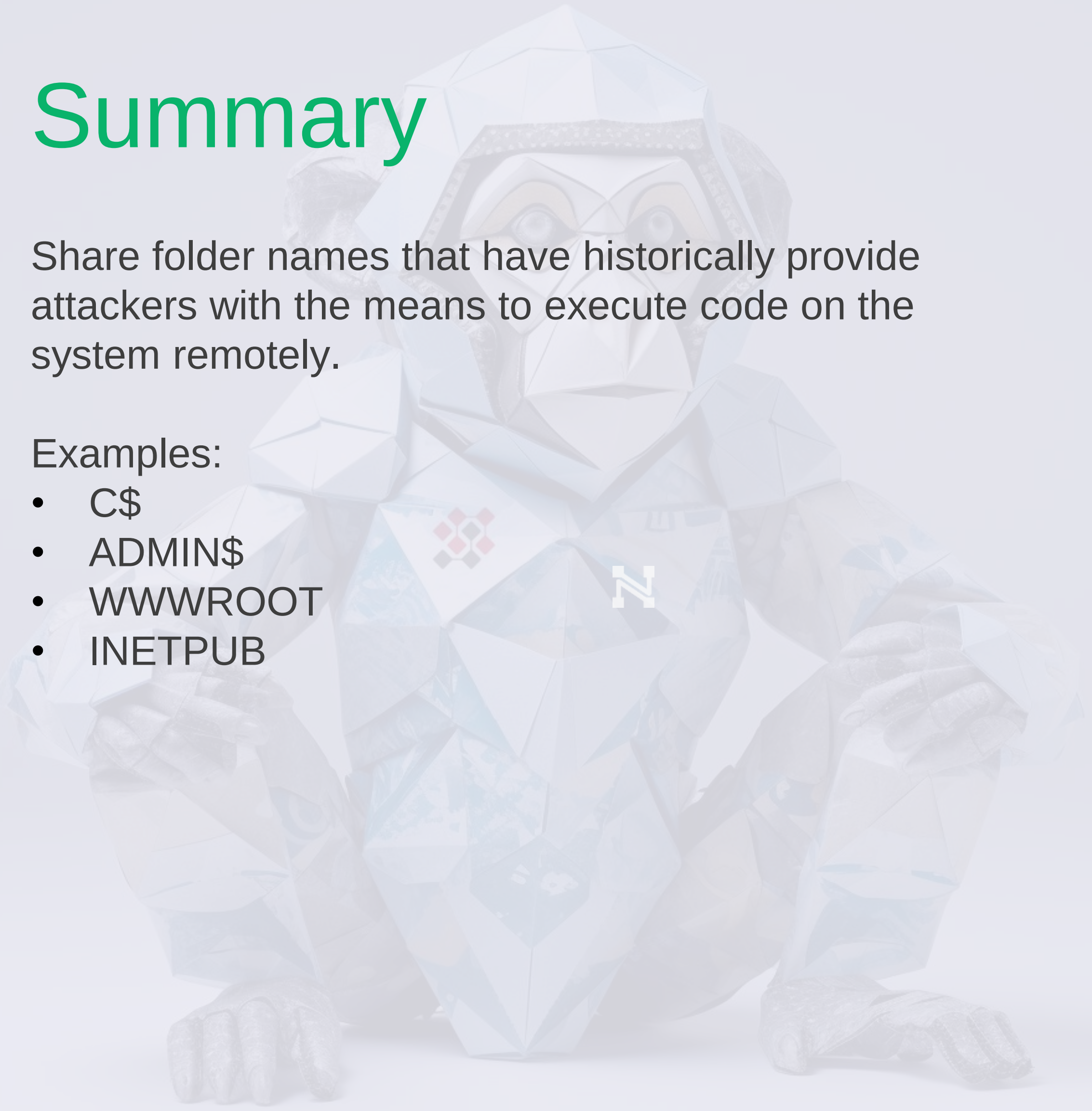
- Highly Exploitable, Interesting Files, Secrets Extraction, Stale, Empty

https://github.com/NetSPI/PowerHuntShares

Static Labeling

… mostly :)

- **Highly Exploitable**
- Interesting Files (data and secrets)
- Extracting Secrets
- Stale (last modified date > 1yr)
- Empty (no files)



Summary

Share folder names that have historically provide attackers with the means to execute code on the system remotely.

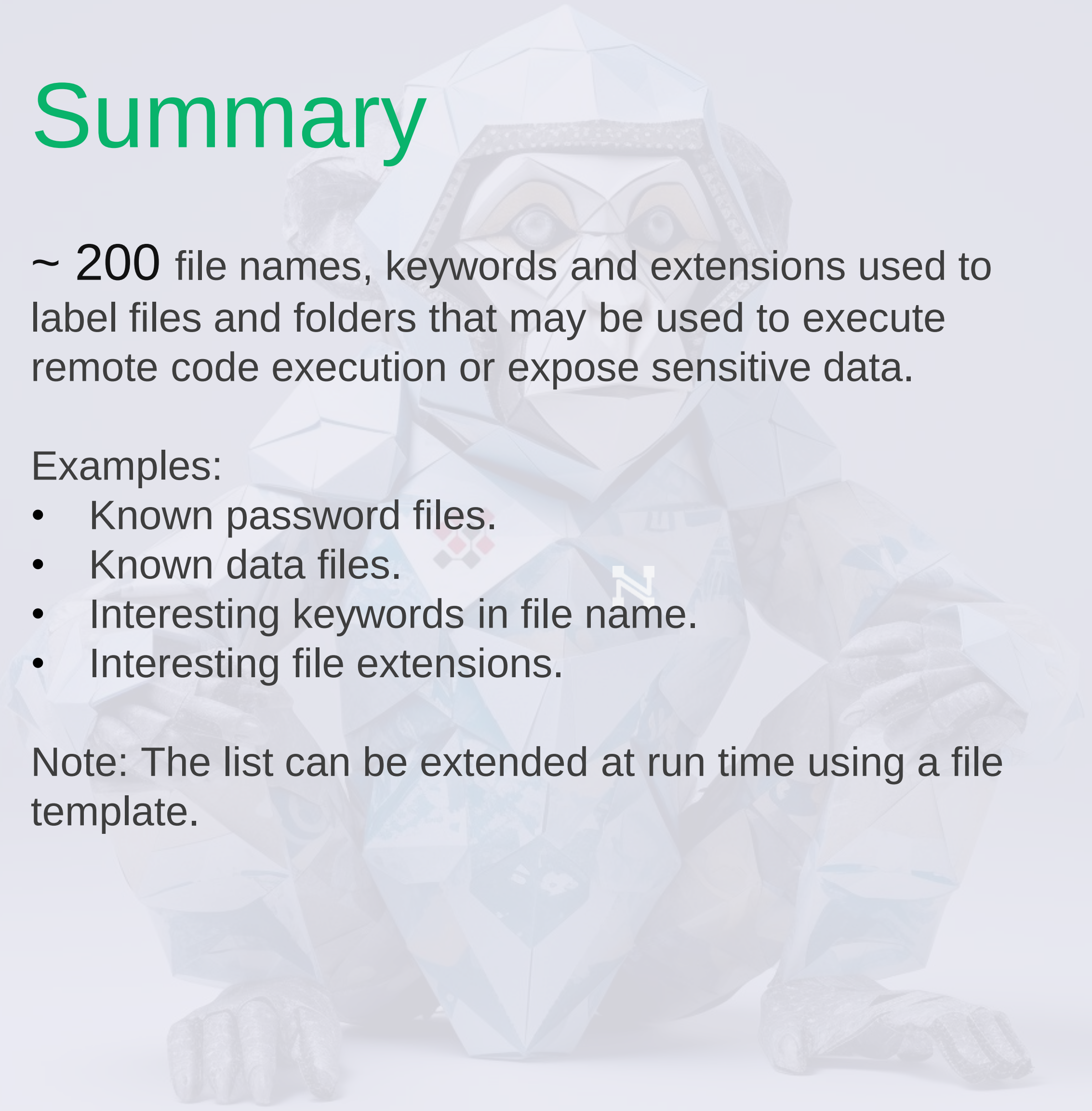
Examples:

- C$
- ADMIN$
- WWWROOT
- INETPUB

Static Labeling

… mostly :)

- High Risk Shares
- **Interesting Files** (data and secrets)
- Extracting Secrets
- Stale (last modified date > 1yr)
- Empty (no files)



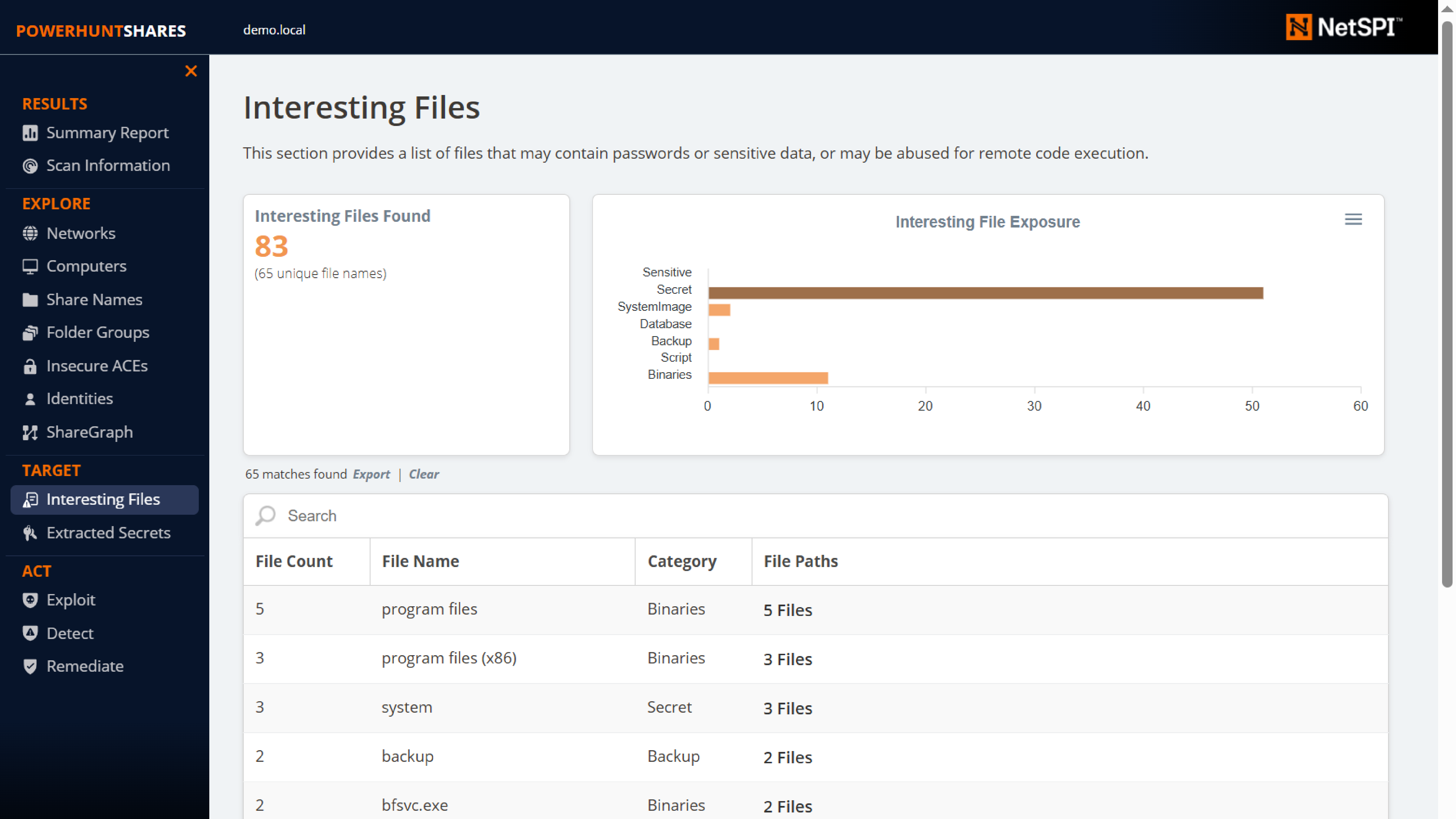
Summary

~ 200 file names, keywords and extensions used to label files and folders that may be used to execute remote code execution or expose sensitive data.

Examples:

- Known password files.
- Known data files.
- Interesting keywords in file name.
- Interesting file extensions.

Note: The list can be extended at run time using a file template.



Interesting Files

This section provides a list of files that may contain passwords or sensitive data, or may be abused for remote code execution.

Interesting Files Found

83

(65 unique file names)

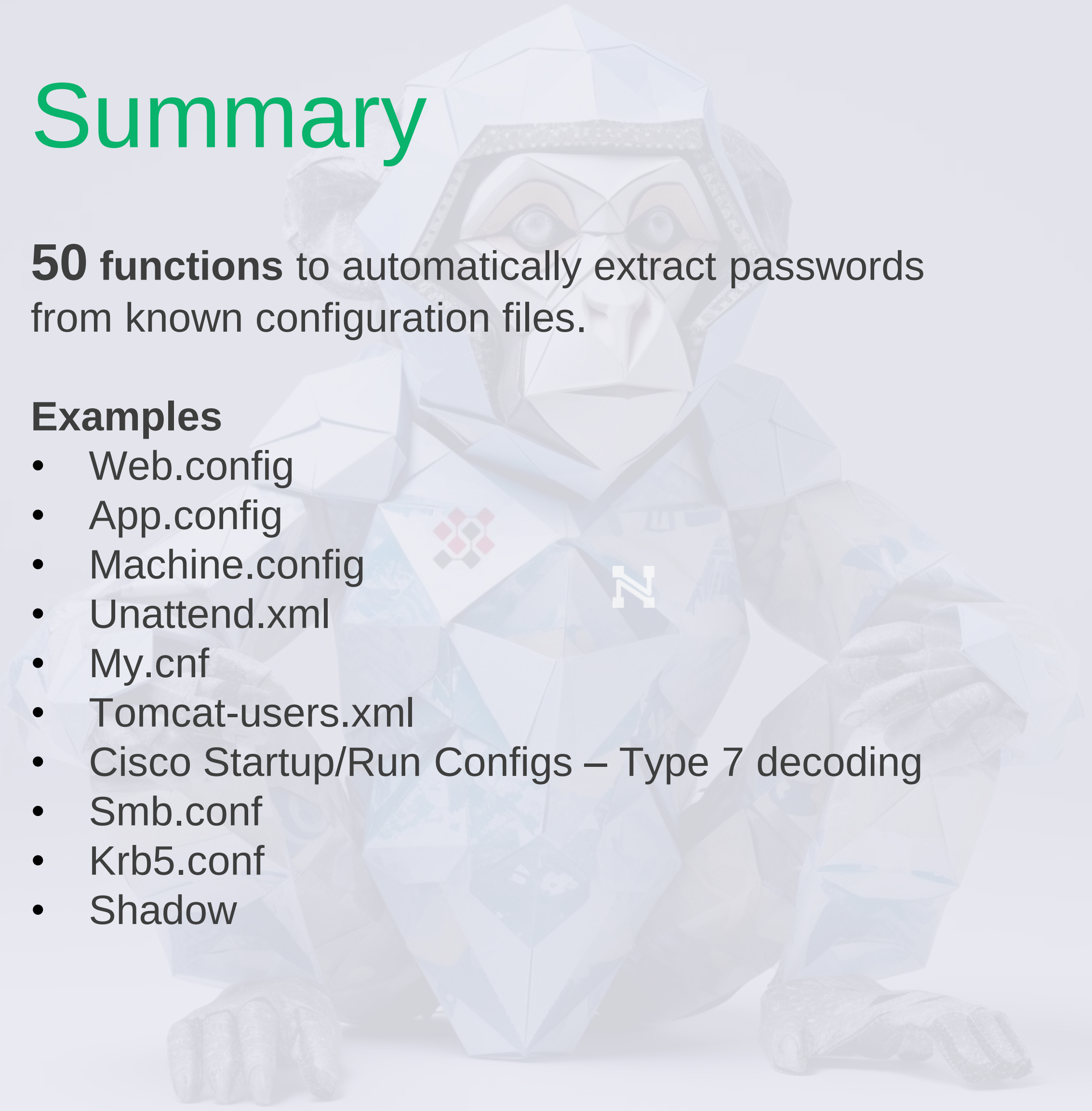
65 matches found *Export* | *Clear*

File Count	File Name	Category	File Paths
5	program files	Binaries	**5 Files**
3	program files (x86)	Binaries	**3 Files**
3	system	Secret	**3 Files**
2	backup	Backup	**2 Files**
2	bfsvc.exe	Binaries	**2 Files**

Static Labeling

… mostly :)

- High Risk Shares
- Interesting Files (data and secrets)
- **Extracting Secrets**
- Stale (last modified date > 1yr)
- Empty (no files)



Summary

50 functions to automatically extract passwords from known configuration files.

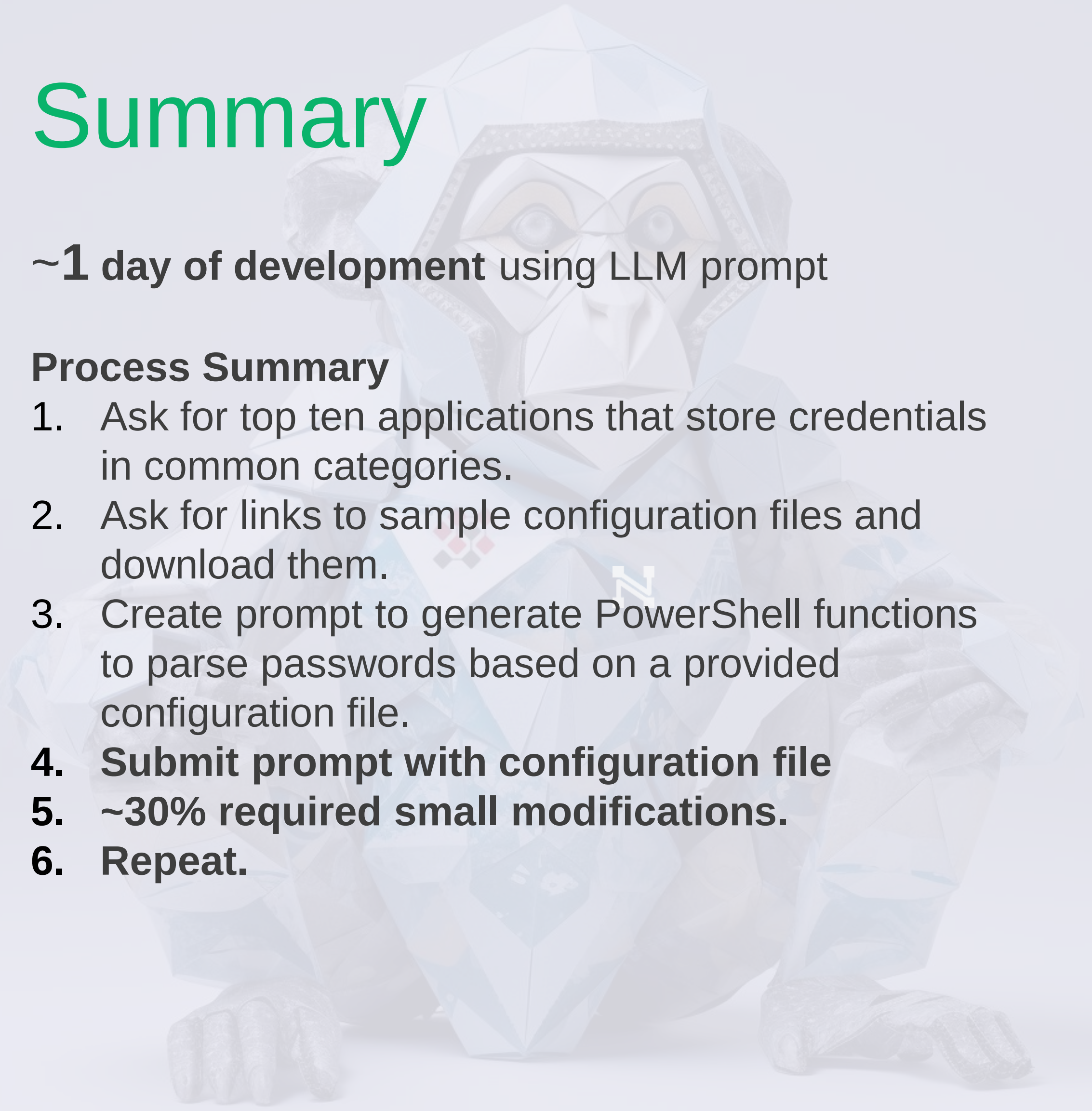
Examples

- Web.config
- App.config
- Machine.config
- Unattend.xml
- My.cnf
- Tomcat-users.xml
- Cisco Startup/Run Configs – Type 7 decoding
- Smb.conf
- Krb5.conf
- Shadow

Static Labeling

… mostly :)

- High Risk Shares
- Interesting Files (data and secrets)
- **Extracting Secrets**
- Stale (last modified date > 1yr)
- Empty (no files)



Summary

~**1 day of development** using LLM prompt

Process Summary

1. Ask for top ten applications that store credentials in common categories.
2. Ask for links to sample configuration files and download them.
3. Create prompt to generate PowerShell functions to parse passwords based on a provided configuration file.
4. **Submit prompt with configuration file**
5. **~30% required small modifications.**
6. **Repeat.**

Static Labeling

… mostly :)

- High Risk Shares
- Interesting Files (data and secrets)
- **Extracting Secrets**
- Stale (last modified date > 1yr)
- Empty (no files)

Sample Prompt

1. Create a PowerShell function that parses usernames and passwords from the provided example file.

2. Ensure the PowerShell function supports an input parameter named “FilePath” that accepts a path to the configuration file so it can be read and parsed.

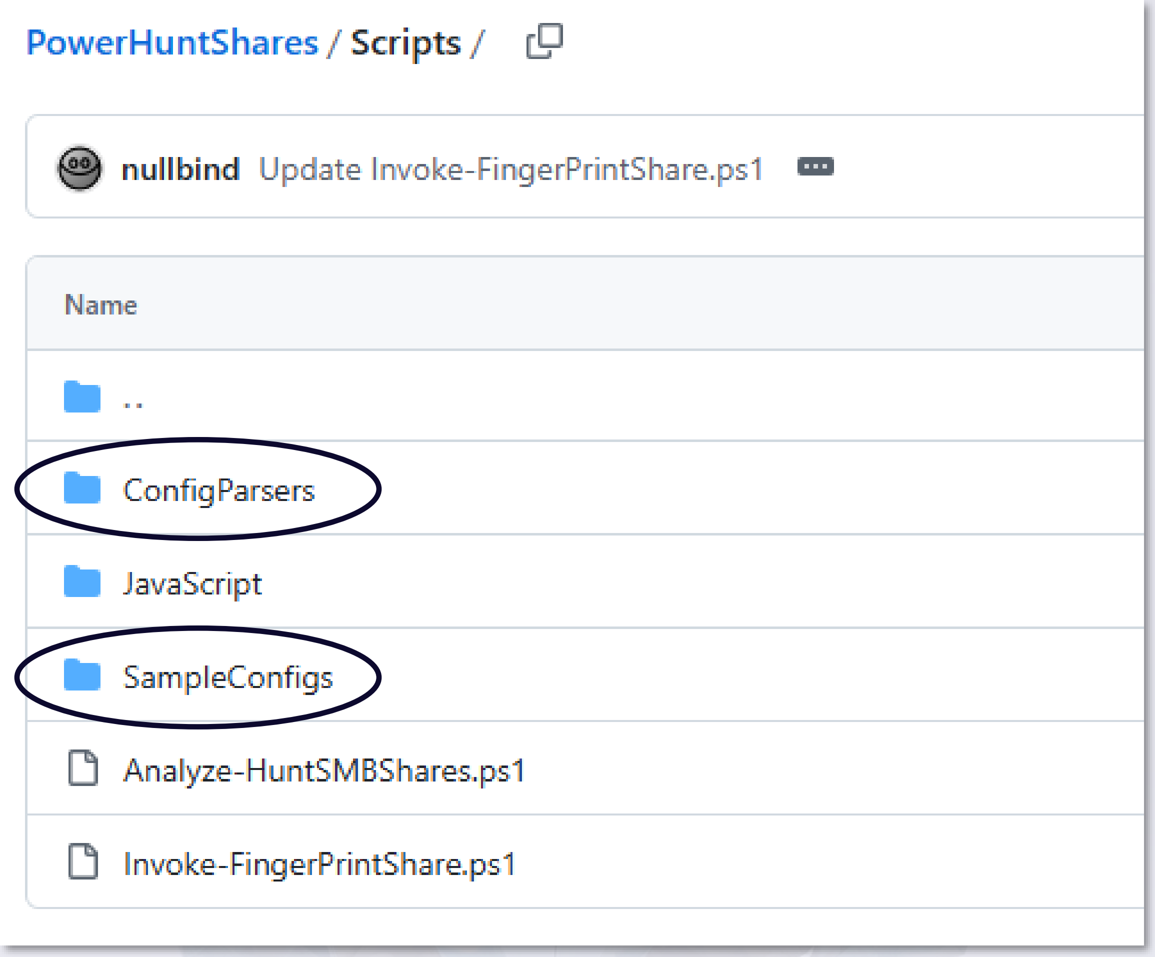
3. Ensure all output is provided as a PSObject. Ensure each parsed username and password pair is returned as a separate record. Output parameters should include “username” and “password”. If their values are empty in the file, then return “EMPTY” for their values in the PSObject.

Example Configuration File:
Content Here

Static Labeling

… mostly :)

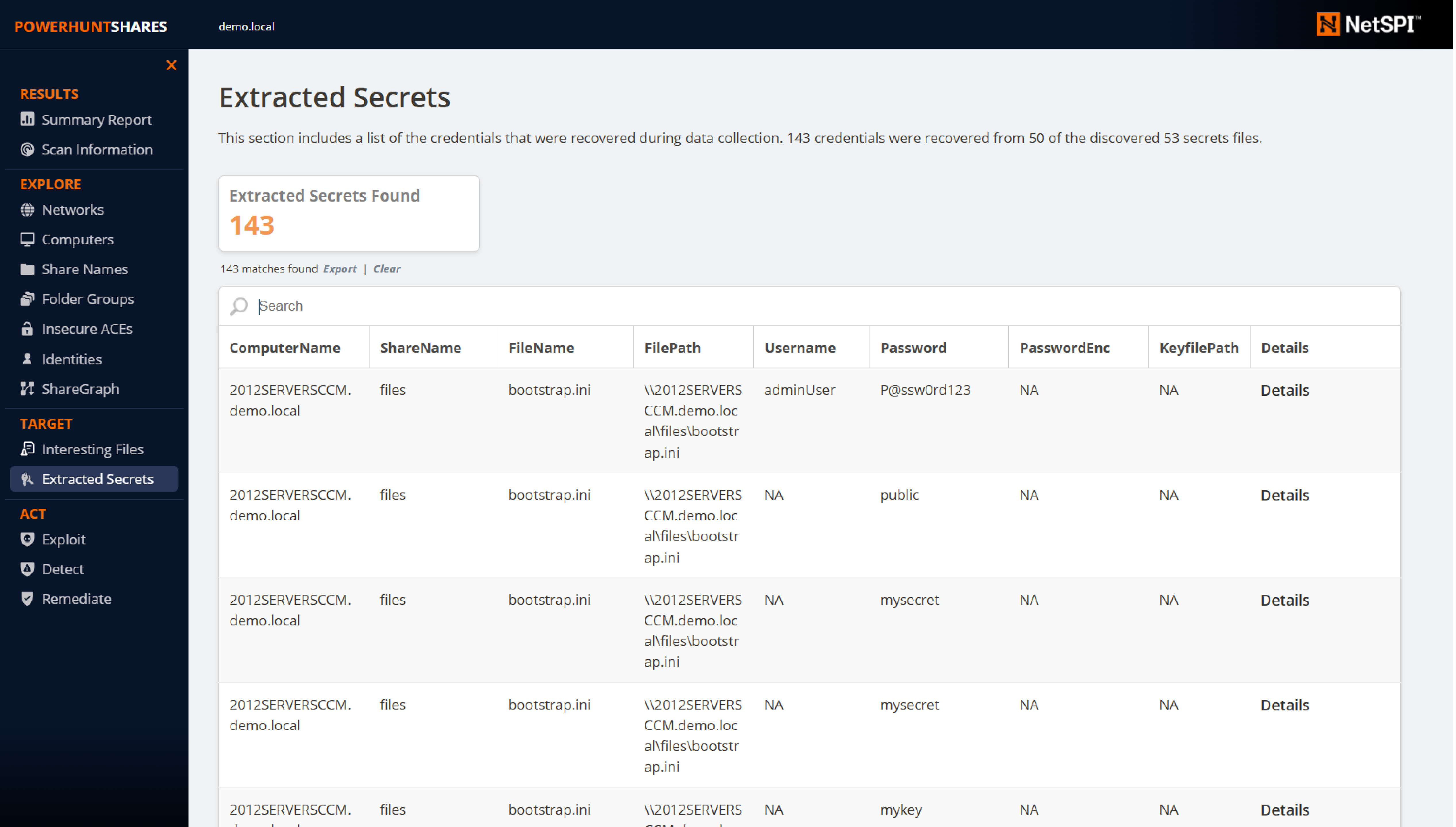
- High Risk Shares
- Interesting Files (data and secrets)
- **Extracting Secrets**
- Stale (last modified date > 1yr)
- Empty (no files)



PowerHuntShares / Scripts /

nullbind Update Invoke-FingerPrintShare.ps1

Name
..
ConfigParsers
JavaScript
SampleConfigs
Analyze-HuntSMBShares.ps1
Invoke-FingerPrintShare.ps1



Extracted Secrets

This section includes a list of the credentials that were recovered during data collection. 143 credentials were recovered from 50 of the discovered 53 secrets files.

Extracted Secrets Found
143

143 matches found *Export* | *Clear*

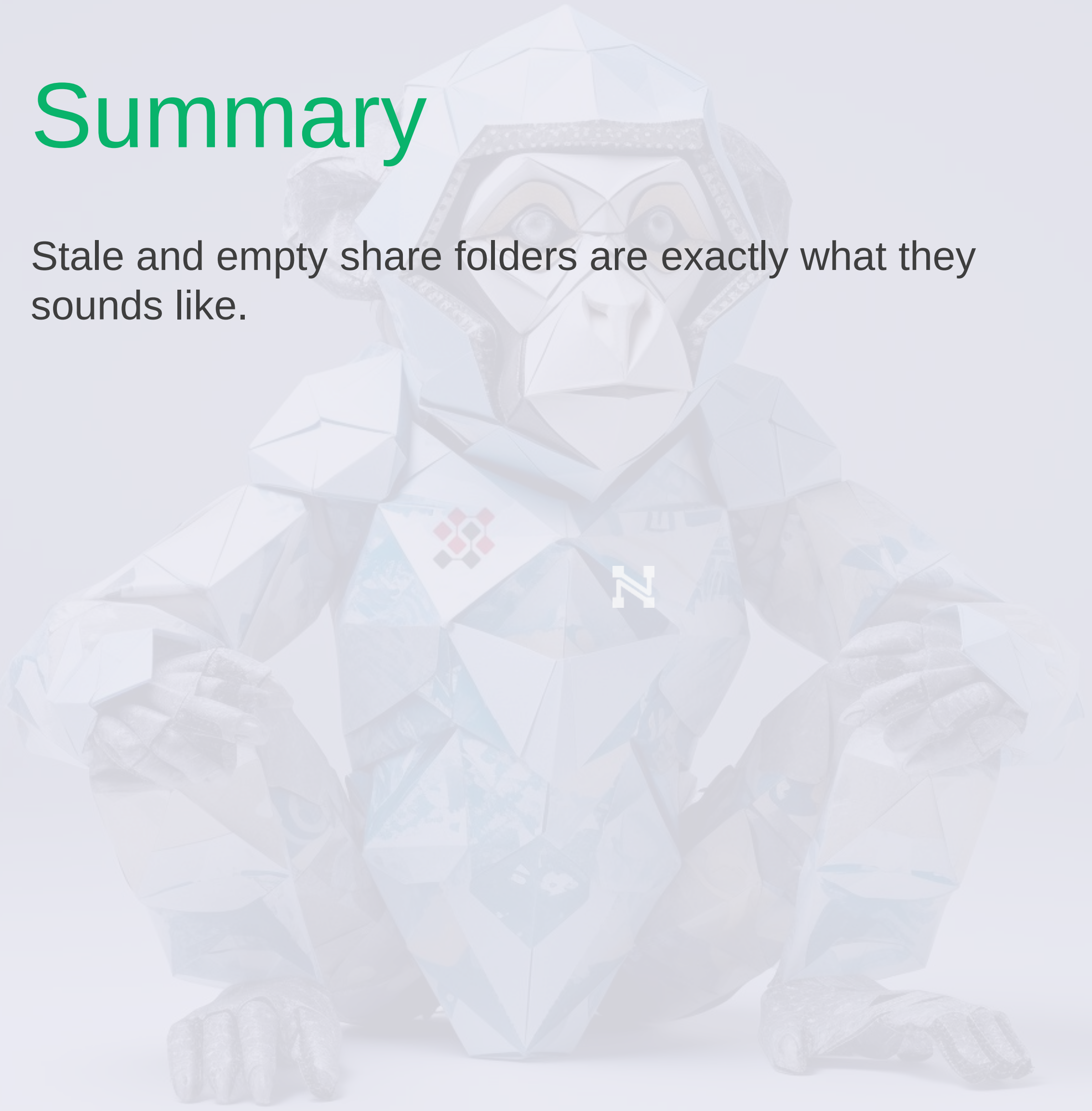
Search

ComputerName	ShareName	FileName	FilePath	Username	Password	PasswordEnc	KeyfilePath	Details
2012SERVERSCCM.demo.local	files	bootstrap.ini	\\2012SERVERSCCM.demo.local\files\bootstrap.ini	adminUser	P@ssw0rd123	NA	NA	Details
2012SERVERSCCM.demo.local	files	bootstrap.ini	\\2012SERVERSCCM.demo.local\files\bootstrap.ini	NA	public	NA	NA	Details
2012SERVERSCCM.demo.local	files	bootstrap.ini	\\2012SERVERSCCM.demo.local\files\bootstrap.ini	NA	mysecret	NA	NA	Details
2012SERVERSCCM.demo.local	files	bootstrap.ini	\\2012SERVERSCCM.demo.local\files\bootstrap.ini	NA	mysecret	NA	NA	Details
2012SERVERSCCM.	files	bootstrap.ini	\\2012SERVERS	NA	mykey	NA	NA	Details

Static Labeling

… mostly :)

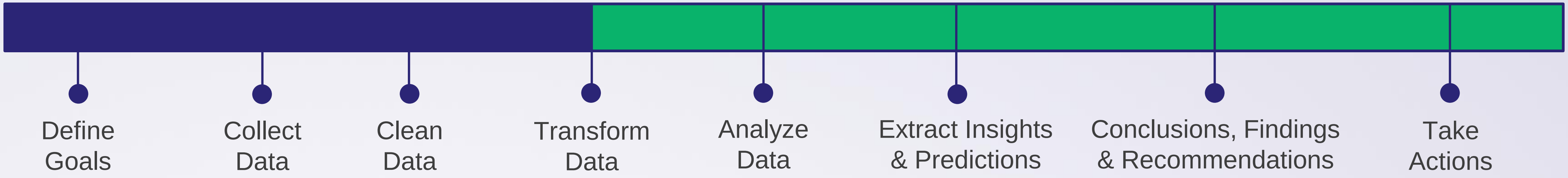
- High Risk Shares
- Interesting Files (data and secrets)
- Extracting Secrets
- **Stale (last modified date > 1yr)**
- **Empty (no files)**



Summary

Stale and empty share folders are exactly what they sounds like.

PowerHuntShares Process



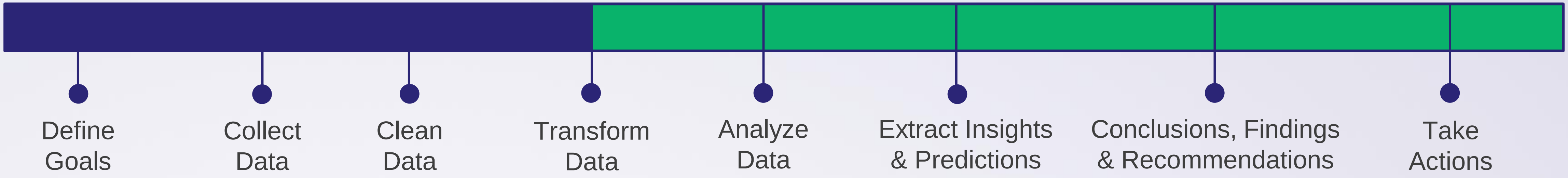
Transform Data

Data + Context

Static Data Labeling

- Highly Exploitable, Interesting Files, Secrets Extraction, Stale, Empty

PowerHuntShares Process



Define Goals

Collect Data

Clean Data

Transform Data

Analyze Data

Extract Insights & Predictions

Conclusions, Findings & Recommendations

Take Actions

Transform Data

Data + Context + Context

Static Data Labeling

- Highly Exploitable, Interesting Files, Secrets Extraction, Stale, Empty

Dynamic Data Enrichment

- **Fingerprinting**

Share Fingerprinting

"What is this share used for?"

Share Fingerprinting

“What is this share used for?”

Why Fingerprint Shares?

Improve Offensive Context

- Increase confidence that a share contains specific files with stored secrets, sensitive data or can be used for remote code execution.

Improve Defensive Context

- Better understand the impact of removing potentially excessive privilege.
- Increase confidence the share or group of shares are related to a specific application or process that can be remediated at the same time.

Share Fingerprinting

“What is this share used for?”

- **Static Hardcoded Application Fingerprint Library**

Summary

~ **100** environments manually analyzed
~ **80** share names mapped to common applications and operating systems

Pros

- Better than what I had, which was nothing. ☺
- Includes descriptions for the shares and related apps.

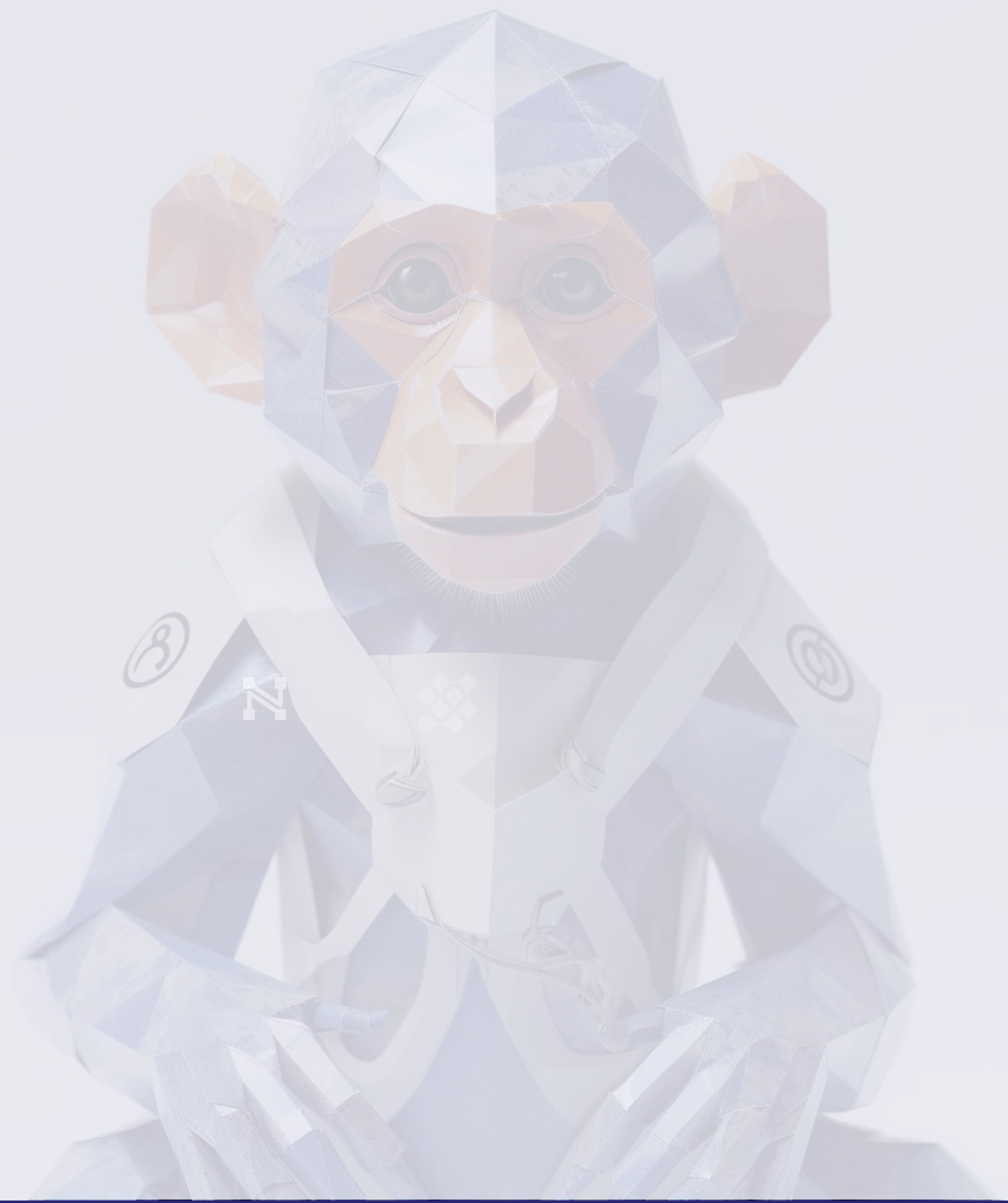
Cons

- Doesn’t consider file listings which can lead to false positives.
- Doesn’t include any fuzzy logic to account for share name variations which can lead to false negatives.
- Currently doesn’t output CPE.

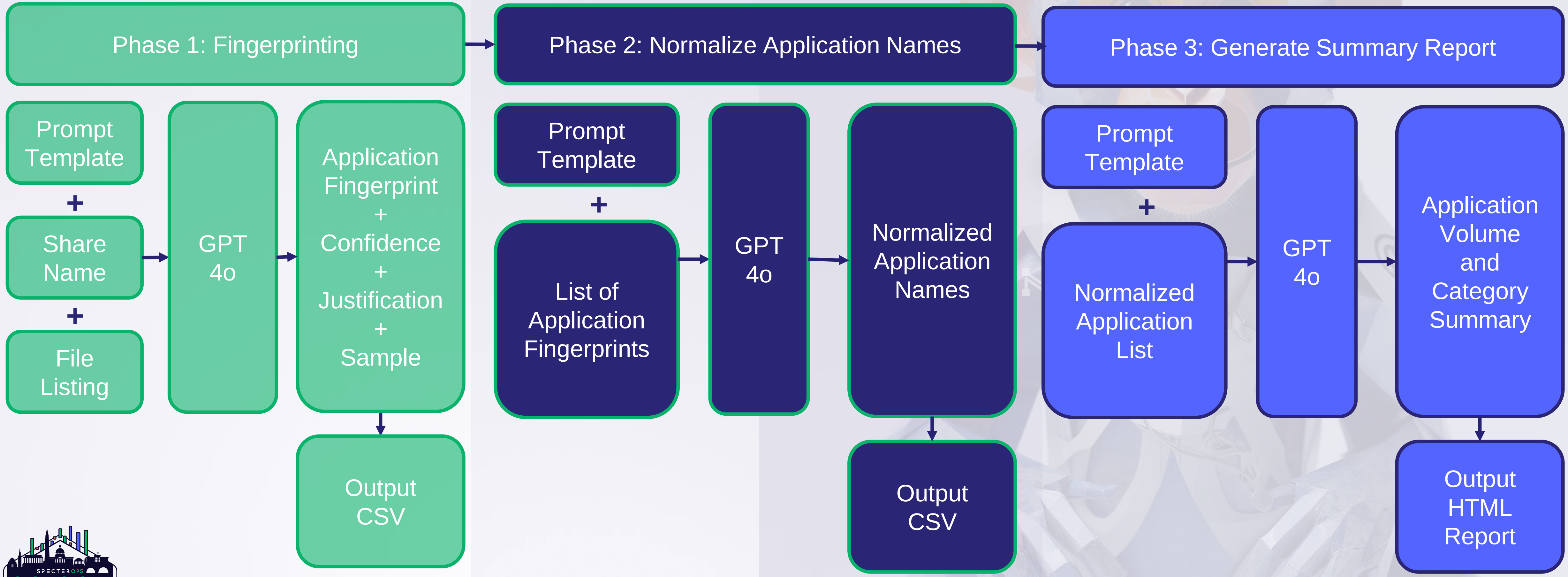
Share Fingerprinting

“What is this share used for?”

- Static Hardcoded Application Fingerprint Library
- **Dynamic LLM-Based Application Fingerprinting**



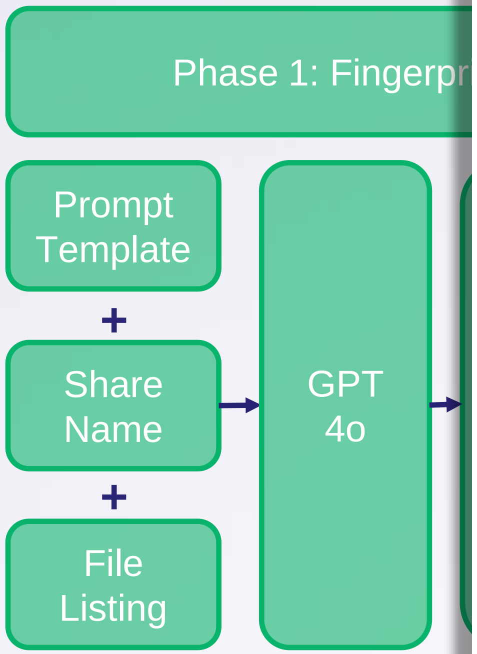
Share Fingerprinting
LLM-Based Process

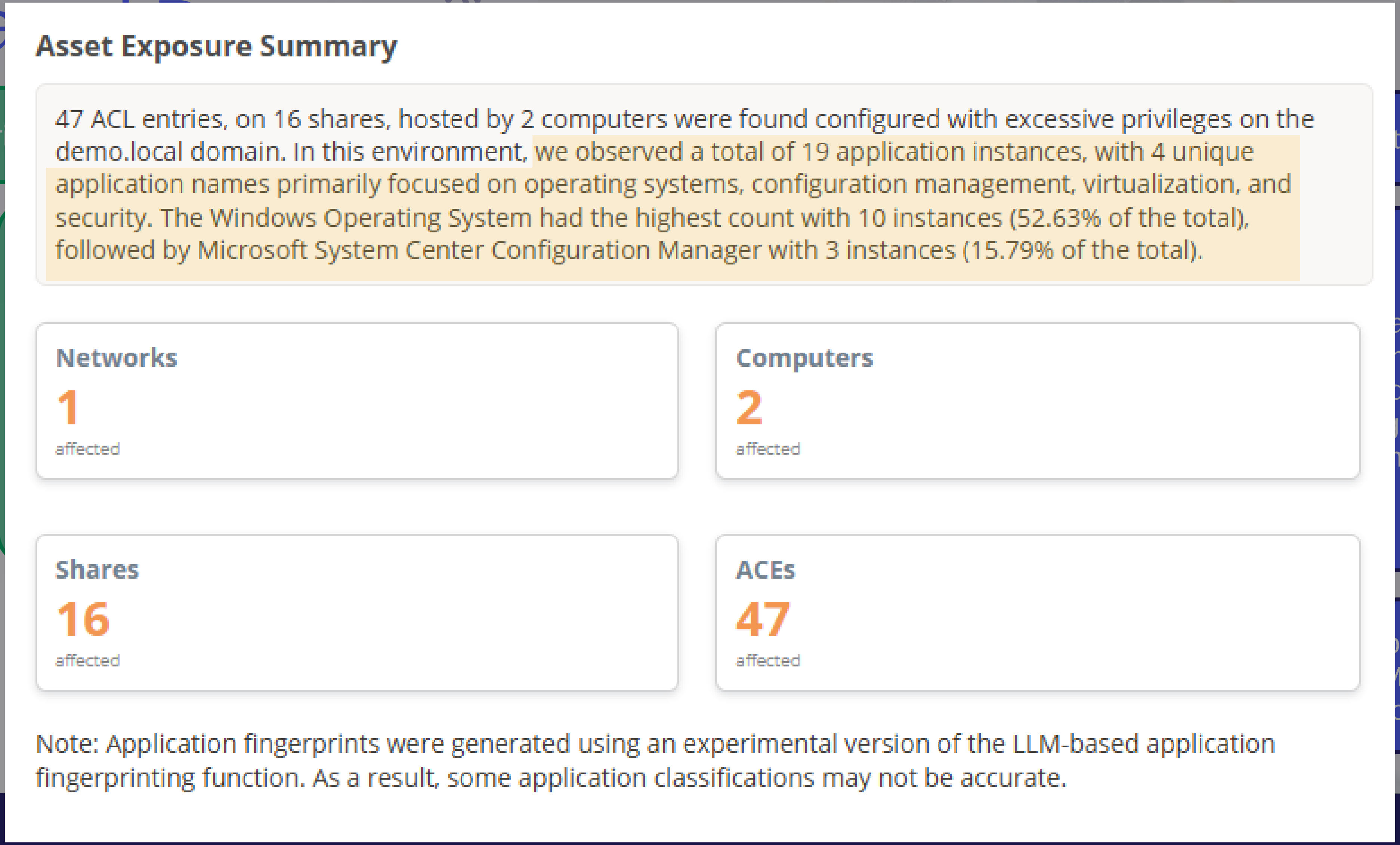


Phase 1: Fingerprinting
Prompt Template
+
Share Name
+
File Listing
GPT 4o
Application Fingerprint + Confidence + Justification + Sample
Output CSV
Phase 2: Normalize Application Names
Prompt Template
+
List of Application Fingerprints
GPT 4o
Normalized Application Names
Output CSV
Phase 3: Generate Summary Report
Prompt Template
+
Normalized Application List
GPT 4o
Application Volume and Category Summary
Output HTML Report

Share Fingerprinting

LLM-Ba





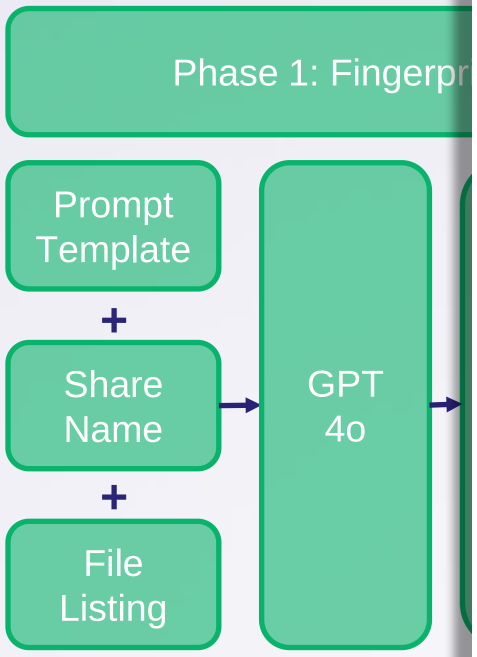
Asset Exposure Summary

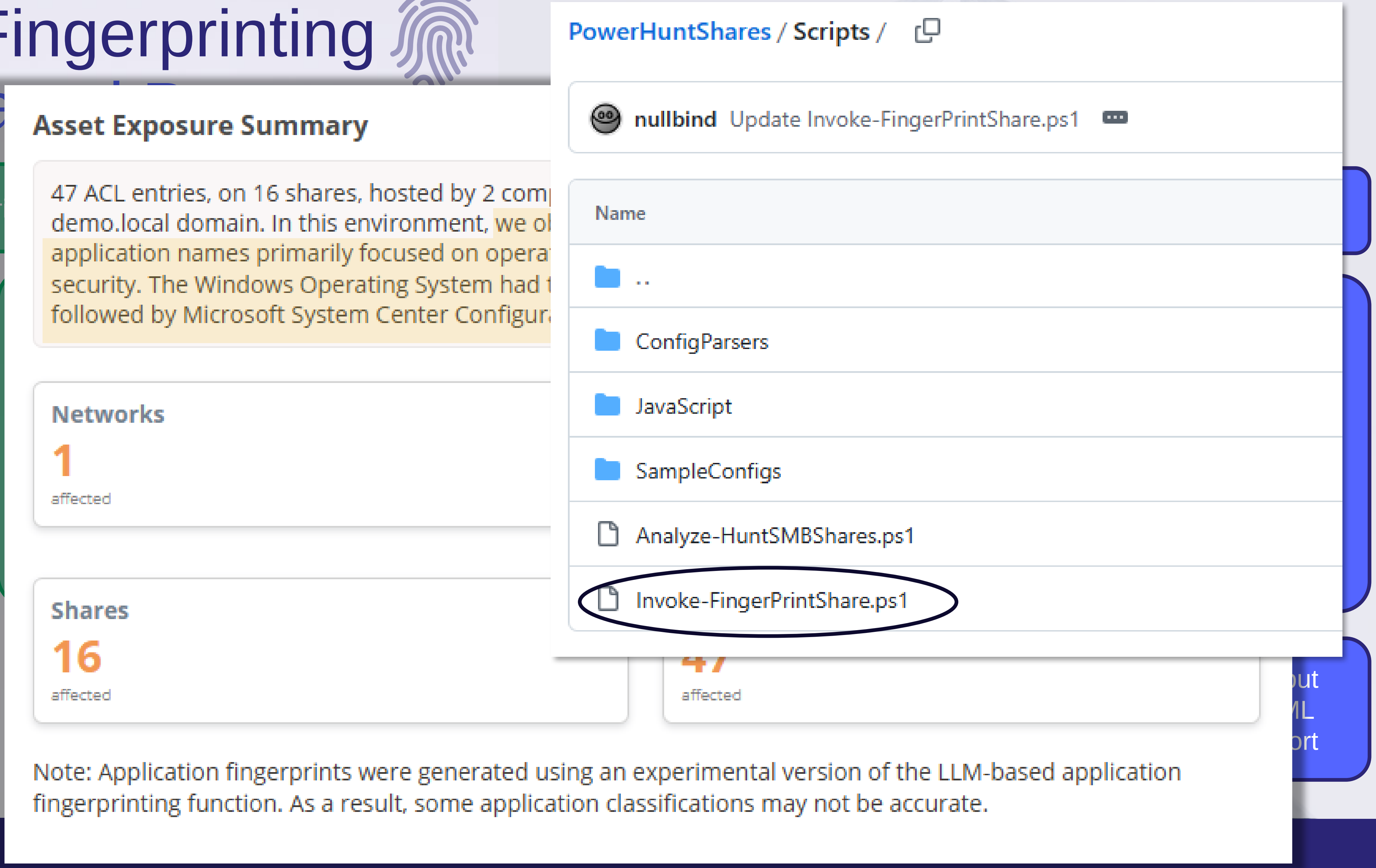
47 ACL entries, on 16 shares, hosted by 2 computers were found configured with excessive privileges on the demo.local domain. In this environment, we observed a total of 19 application instances, with 4 unique application names primarily focused on operating systems, configuration management, virtualization, and security. The Windows Operating System had the highest count with 10 instances (52.63% of the total), followed by Microsoft System Center Configuration Manager with 3 instances (15.79% of the total).

Networks 1 affected	Computers 2 affected
Shares 16 affected	ACEs 47 affected

Note: Application fingerprints were generated using an experimental version of the LLM-based application fingerprinting function. As a result, some application classifications may not be accurate.

Share Fingerprinting
LLM-Ba





Asset Exposure Summary

47 ACL entries, on 16 shares, hosted by 2 comp
demo.local domain. In this environment, we o
application names primarily focused on opera
security. The Windows Operating System had t
followed by Microsoft System Center Configur

Networks
1
affected

Shares
16
affected

47
affected

Note: Application fingerprints were generated using an experimental version of the LLM-based application fingerprinting function. As a result, some application classifications may not be accurate.

PowerHuntShares / **Scripts** /

nullbind Update Invoke-FingerPrintShare.ps1

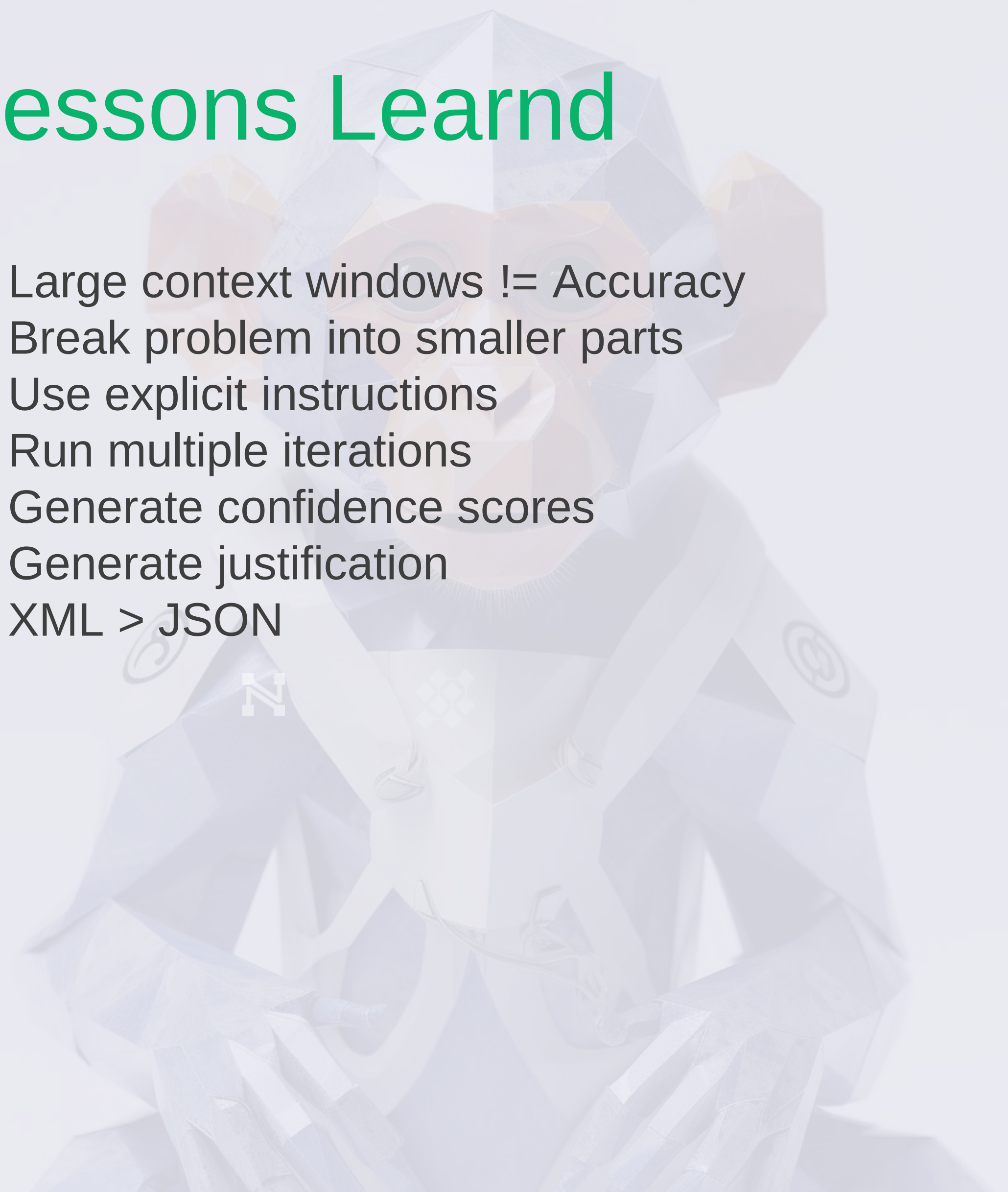
Name
..
ConfigParsers
JavaScript
SampleConfigs
Analyze-HuntSMBShares.ps1
Invoke-FingerPrintShare.ps1

Share Fingerprinting

“What is this share used for?”

- Static Hardcoded Application Fingerprint Library
- **Dynamic LLM-Based Application Fingerprinting**

Lessons Learnd

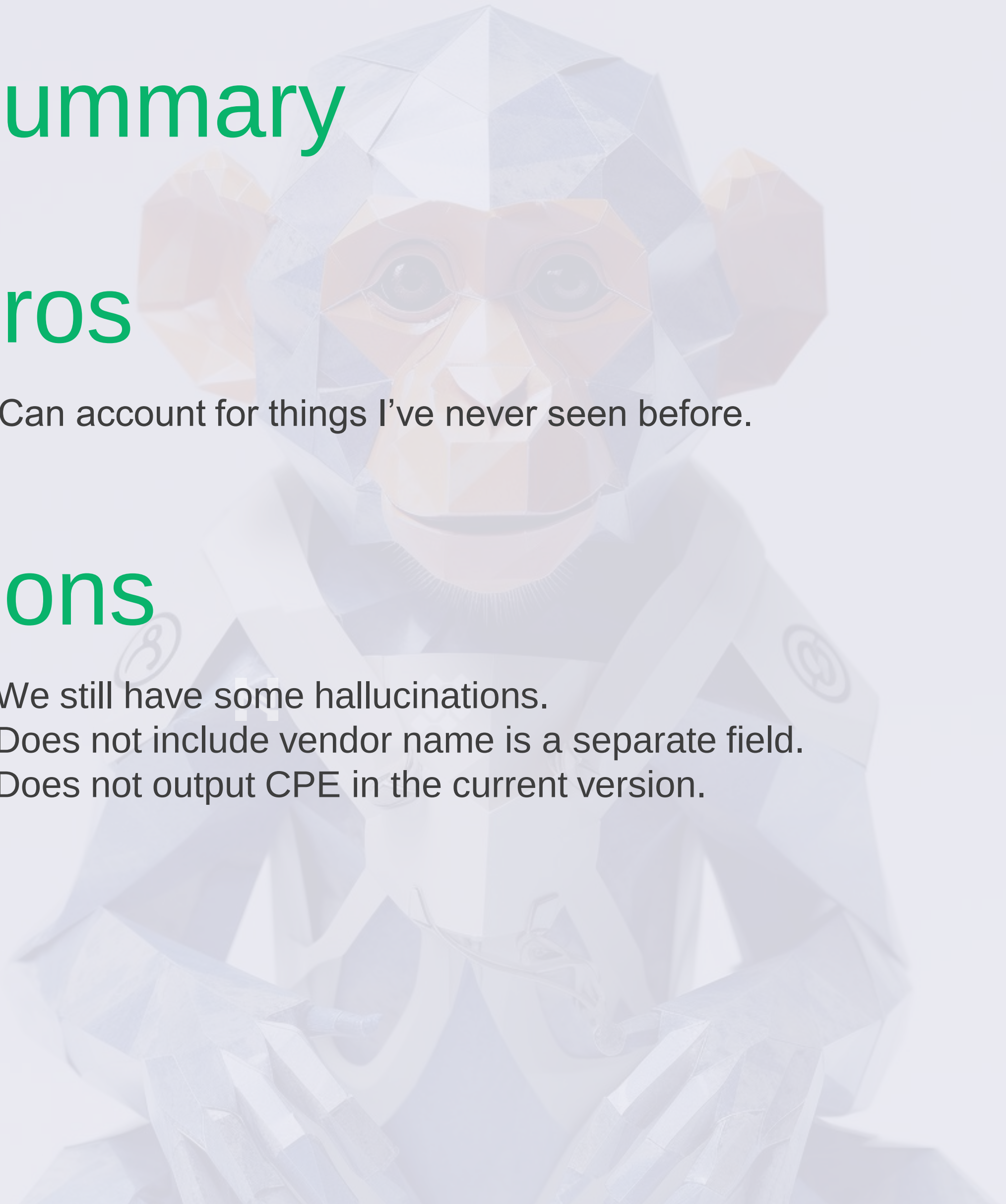


- Large context windows != Accuracy
- Break problem into smaller parts
- Use explicit instructions
- Run multiple iterations
- Generate confidence scores
- Generate justification
- XML > JSON

Share Fingerprinting

“What is this share used for?”

- Static Hardcoded Application Fingerprint Library
- **Dynamic LLM-Based Application Fingerprinting**



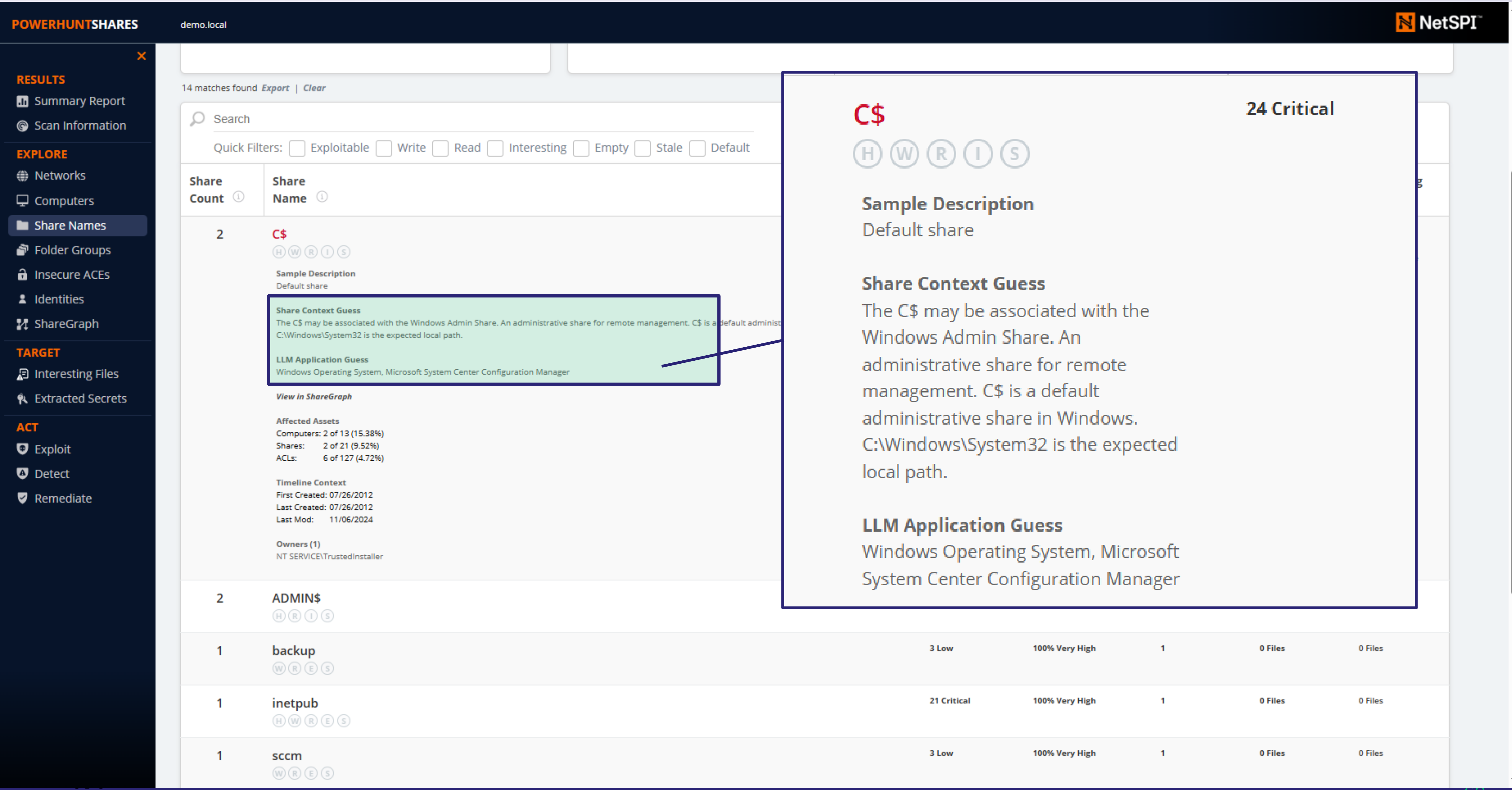
Summary

Pros

- Can account for things I’ve never seen before.

Cons

- We still have some hallucinations.
- Does not include vendor name is a separate field.
- Does not output CPE in the current version.



14 matches found *Export* | *Clear*

Search

Quick Filters: □ Exploitable □ Write □ Read □ Interesting □ Empty □ Stale □ Default

Share Count	Share Name					
2	C$ H W R I S					
2	ADMIN$ H R I S					
1	backup W R E S	3 Low	100% Very High	1	0 Files	0 Files
1	inetpub H W R E S	21 Critical	100% Very High	1	0 Files	0 Files
1	sccm W R E S	3 Low	100% Very High	1	0 Files	0 Files

Sample Description
Default share

Share Context Guess
The C$ may be associated with the Windows Admin Share. An administrative share for remote management. C$ is a default administ
C:\Windows\System32 is the expected local path.

LLM Application Guess
Windows Operating System, Microsoft System Center Configuration Manager

View in ShareGraph

Affected Assets
Computers: 2 of 13 (15.38%)
Shares: 2 of 21 (9.52%)
ACLs: 6 of 127 (4.72%)

Timeline Context
First Created: 07/26/2012
Last Created: 07/26/2012
Last Mod: 11/06/2024

Owners (1)
NT SERVICE\TrustedInstaller

C$ 24 Critical

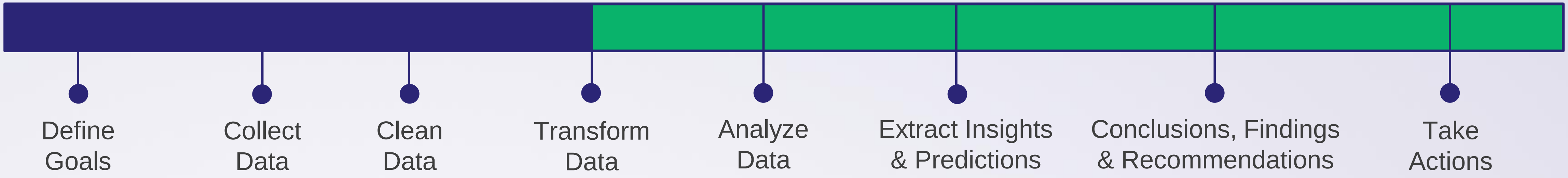
H W R I S

Sample Description
Default share

Share Context Guess
The C$ may be associated with the Windows Admin Share. An administrative share for remote management. C$ is a default administrative share in Windows. C:\Windows\System32 is the expected local path.

LLM Application Guess
Windows Operating System, Microsoft System Center Configuration Manager

PowerHuntShares Process



Transform Data

Data + Context + Context

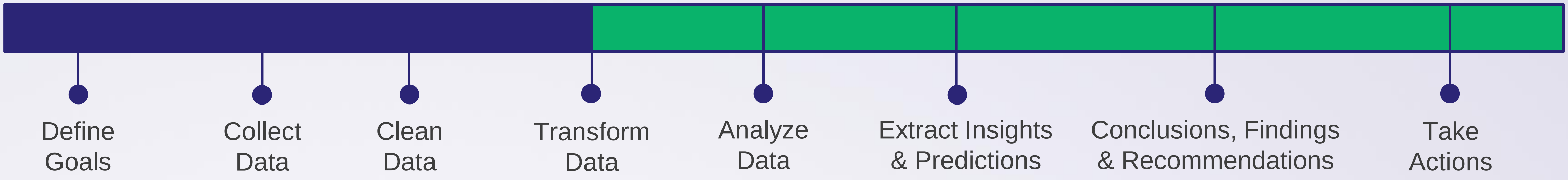
Static Data Labeling

- Highly Exploitable, Interesting Files, Secrets Extraction, Stale, Empty

Dynamic Data Enrichment

- **Fingerprinting**

PowerHuntShares Process



Define Goals

Collect Data

Clean Data

Transform Data

Analyze Data

Extract Insights & Predictions

Conclusions, Findings & Recommendations

Take Actions

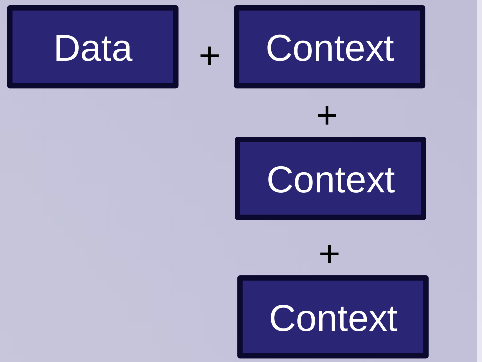
Transform Data

Static Data Labeling

- Highly Exploitable, Interesting Files, Secrets Extraction, Stale, Empty

Dynamic Data Enrichment

- Fingerprinting, **Risk Scoring**



Data + Context + Context + Context

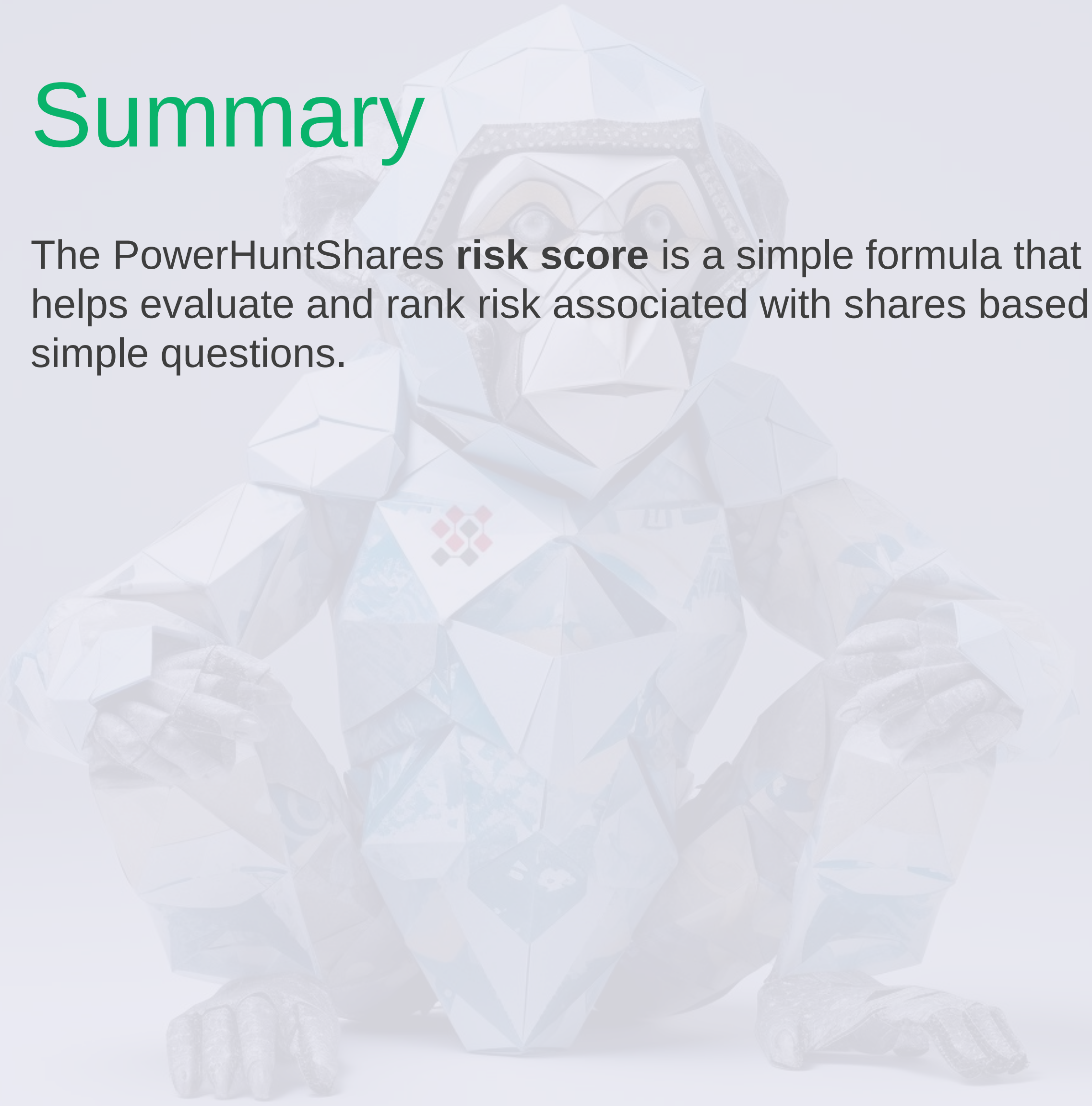
Risk Scoring

“Be honest, how bad is it?”

Risk Scoring

“Be honest, how bad is it?”

- **Summary**



Summary

The PowerHuntShares **risk score** is a simple formula that helps evaluate and rank risk associated with shares based simple questions.

Risk Scoring

"Be honest, how bad is it?"

- Summary
- **Why Risk Scores?**

Summary

The PowerHuntShares **risk score** is a simple formula that helps evaluate and rank risk associated with shares based simple questions.

Why Risk Scores?

- Help prioritize exploitation
- Help prioritized remediation
- Add context related to abuse impact

Why *Another* Risk Rating?

- **CVSS** didn't provide the data context and volume in the way I wanted.

Risk Scoring

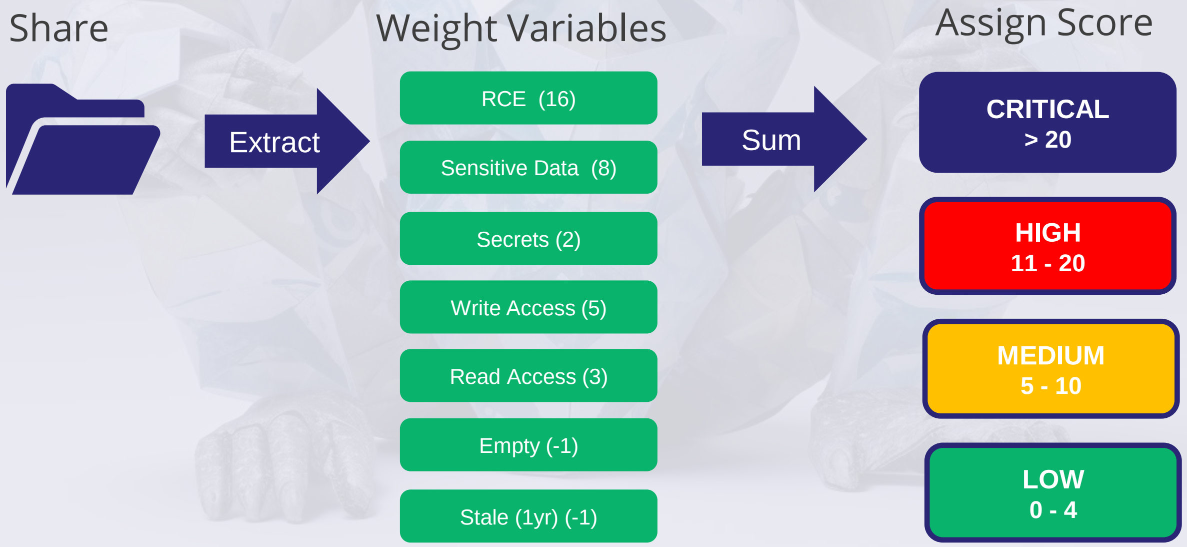
"Be honest, how bad is it?"

- Summary
- Why Risk Scores
- **Formula Abstract**

Summary

The PowerHuntShares **risk score** is a simple formula that helps evaluate and rank risk associated with shares based simple questions.

Formula Abstract



Share

Extract

Weight Variables

RCE (16)

Sensitive Data (8)

Secrets (2)

Write Access (5)

Read Access (3)

Empty (-1)

Stale (1yr) (-1)

Sum

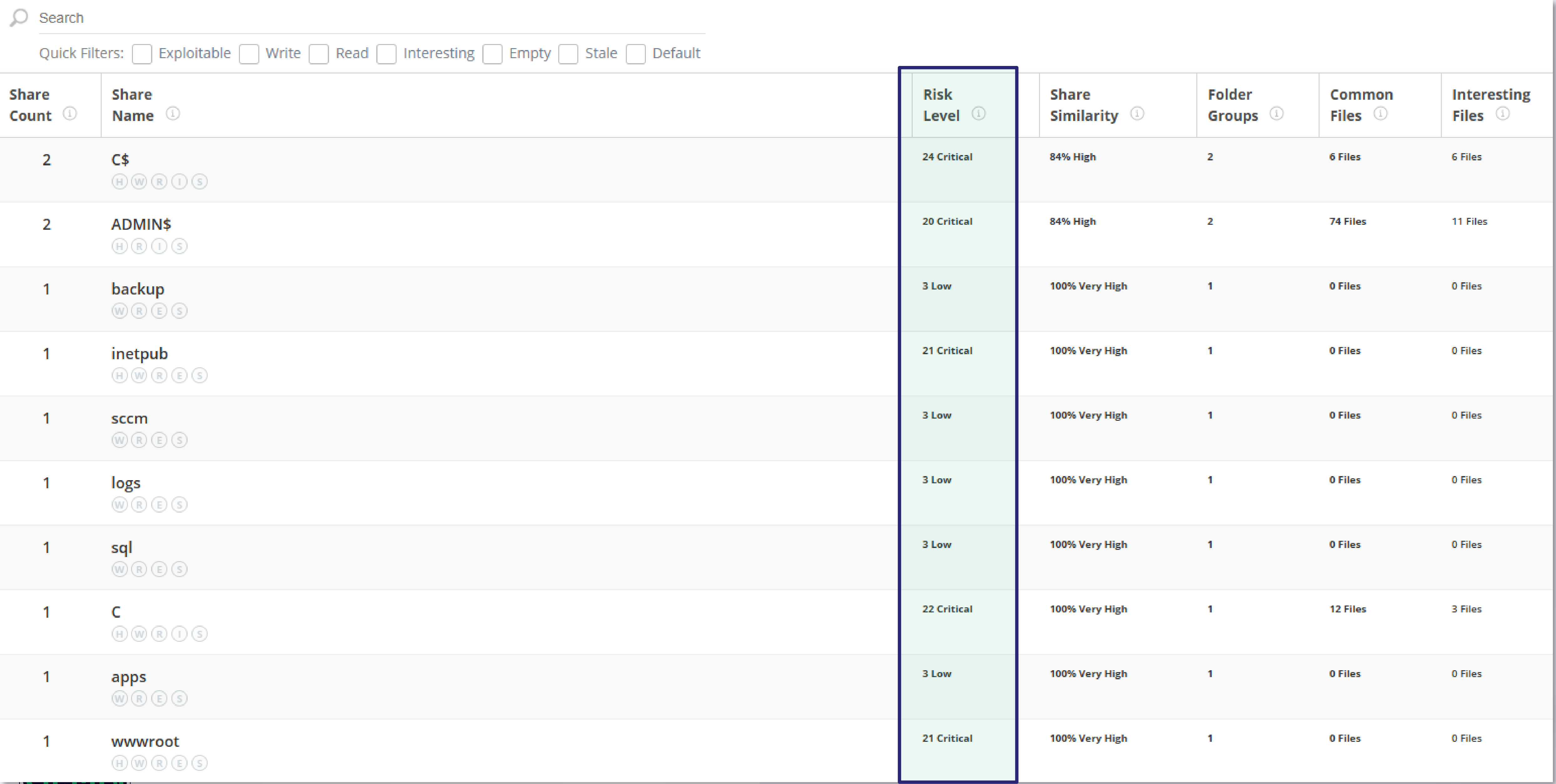
Assign Score

CRITICAL
> 20

HIGH
11 - 20

MEDIUM
5 - 10

LOW
0 - 4

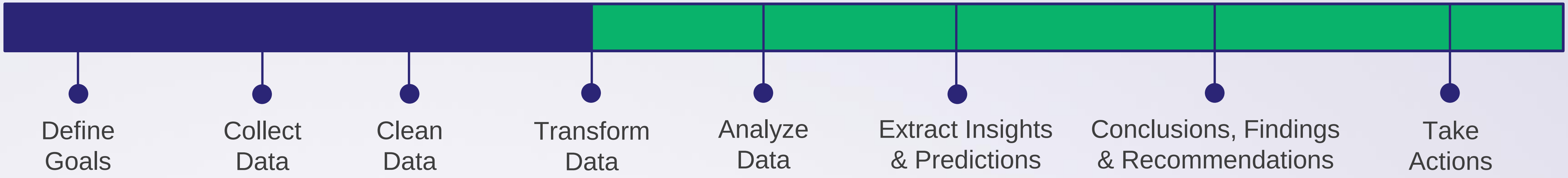


Search

Quick Filters: □ Exploitable □ Write □ Read □ Interesting □ Empty □ Stale □ Default

Share Count	Share Name	Risk Level	Share Similarity	Folder Groups	Common Files	Interesting Files
2	C$ H W R I S	24 Critical	84% High	2	6 Files	6 Files
2	ADMIN$ H R I S	20 Critical	84% High	2	74 Files	11 Files
1	backup W R E S	3 Low	100% Very High	1	0 Files	0 Files
1	inetpub H W R E S	21 Critical	100% Very High	1	0 Files	0 Files
1	sccm W R E S	3 Low	100% Very High	1	0 Files	0 Files
1	logs W R E S	3 Low	100% Very High	1	0 Files	0 Files
1	sql W R E S	3 Low	100% Very High	1	0 Files	0 Files
1	C H W R I S	22 Critical	100% Very High	1	12 Files	3 Files
1	apps W R E S	3 Low	100% Very High	1	0 Files	0 Files
1	wwwroot H W R E S	21 Critical	100% Very High	1	0 Files	0 Files

PowerHuntShares Process



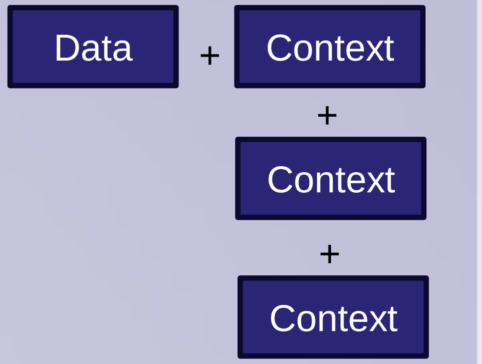
Transform Data

Static Data Labeling

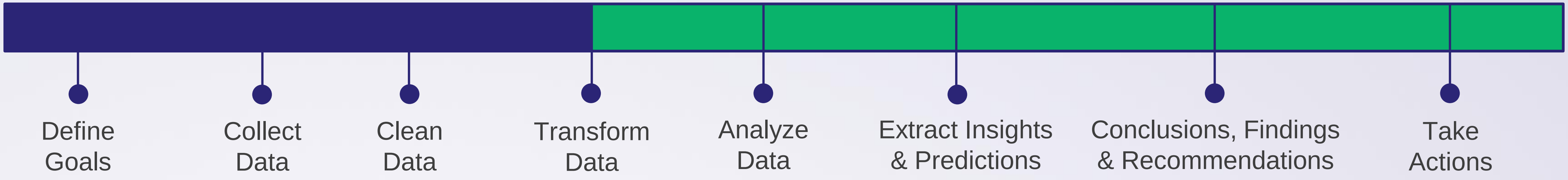
- Highly Exploitable, Interesting Files, Password, Extraction, Stale, Empty

Dynamic Data Enrichment

- Fingerprinting, **Risk Scoring**



PowerHuntShares Process



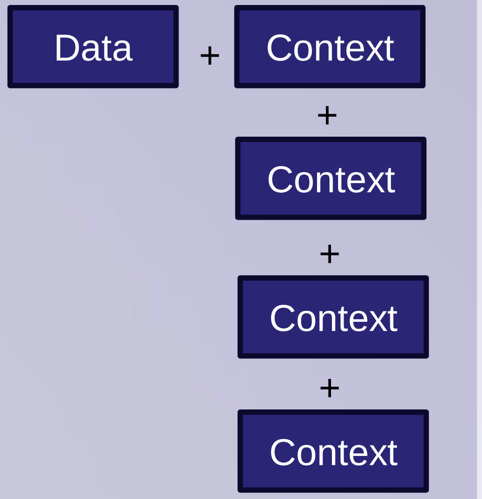
Transform Data

Static Data Labeling

- Highly Exploitable, Interesting Files, Password, Extraction, Stale, Empty

Dynamic Data Enrichment

- Fingerprinting, Risk Scoring, **Peer Comparison**



https://github.com/NetSPI/PowerHuntShares

Peer **Comparison**

“ So, we have 1,000 critical risk shares, really?...

...Good to know, but how do we compare to our peers? ”

Summary

Companies want to understand what’s normal and where they fall short and when they are overachieving.

Use Cases

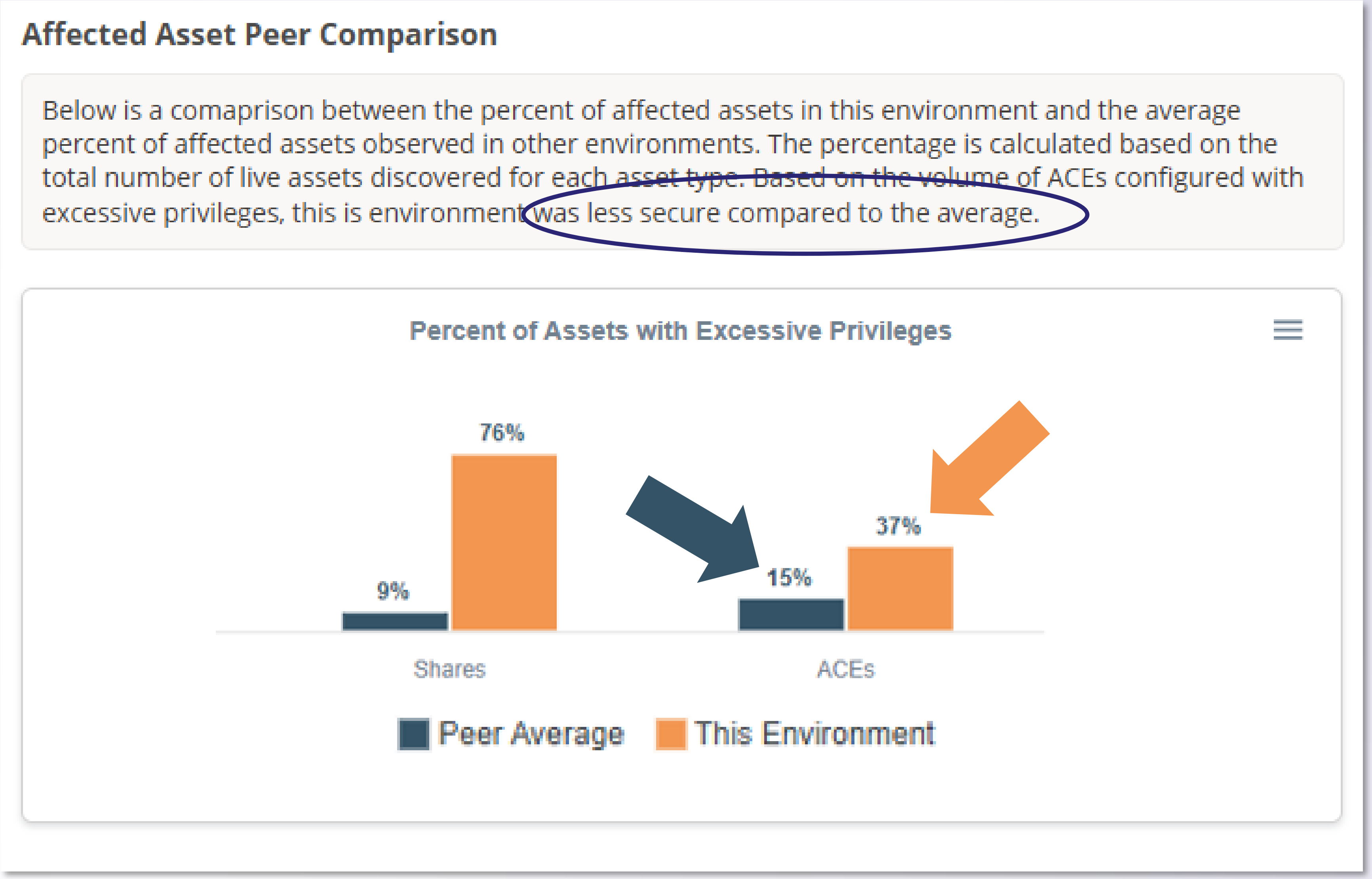
1. **Acquire Budget.**
2. **Use as KPI.**

Tested Approaches

• **Do nothing.**	PowerHuntShares v1
• **Historical Averages.**	PowerHuntShares v2
• **Predictive Models**.	PowerHuntShares v3?

Peer **Comparison**

Historical Average



Affected Asset Peer Comparison

Below is a comaprison between the percent of affected assets in this environment and the average percent of affected assets observed in other environments. The percentage is calculated based on the total number of live assets discovered for each asset type. Based on the volume of ACEs configured with excessive privileges, this is environment was less secure compared to the average.

Peer **Comparison**

Predictive Models

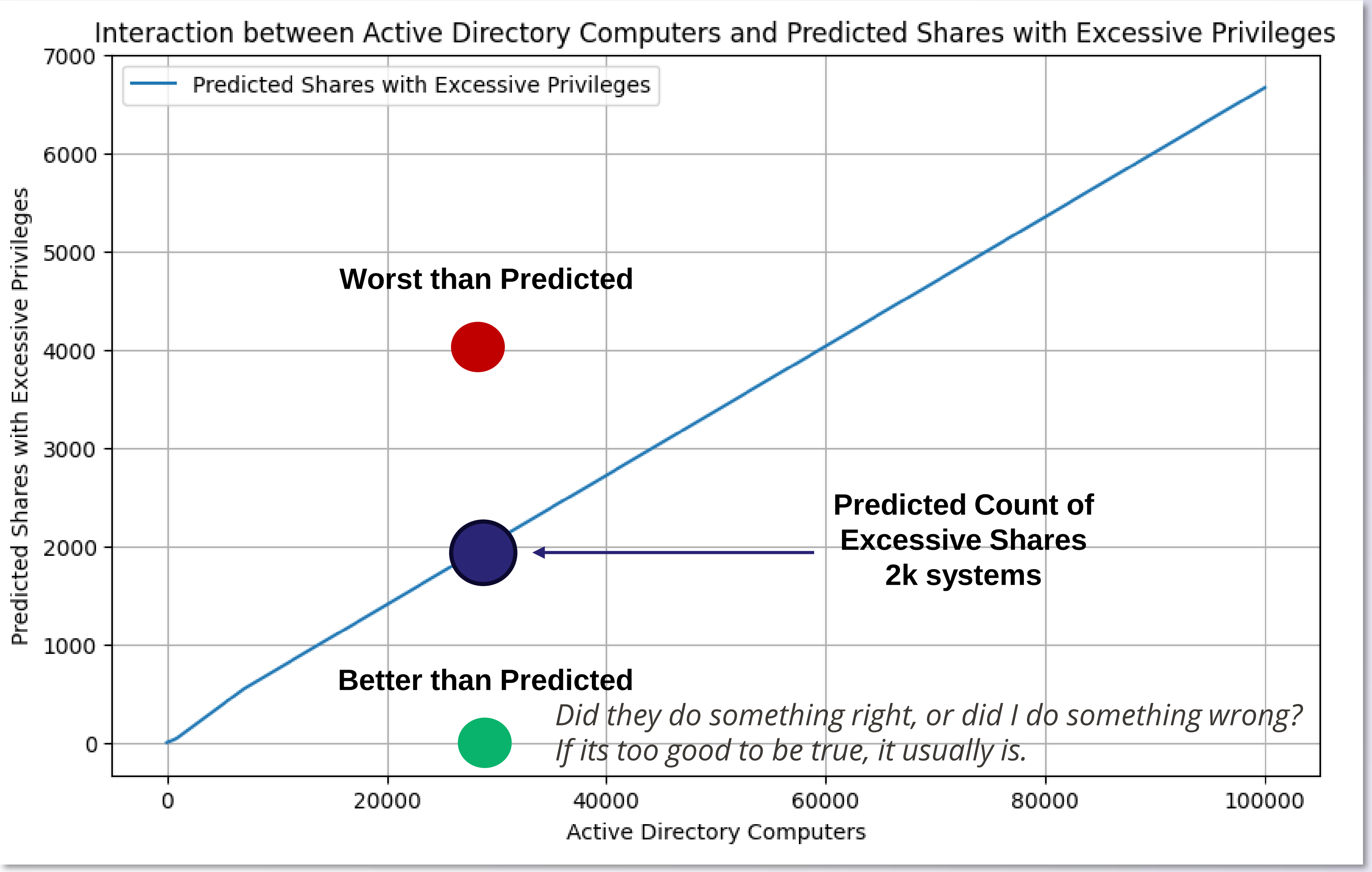
Linear Regression

Randomforest

Neural Network

```
import pandas as pd
import numpy as np
from sklearn.model_selectior
from sklearn.preprocessing i
from sklearn.metrics import
import tensorflow as tf
from tensorflow.keras.models
from tensorflow.keras.layers
import matplotlib.pyplot as
import shap

# Load data
file_path = r"C:\tools\data2
df = pd.read_csv(file_path)
```



Worst than Predicted

Predicted Count of Excessive Shares 2k systems

Better than Predicted

Did they do something right, or did I do something wrong?
If its too good to be true, it usually is.

Neural Network

Peer **Comparison**

Predictive Models

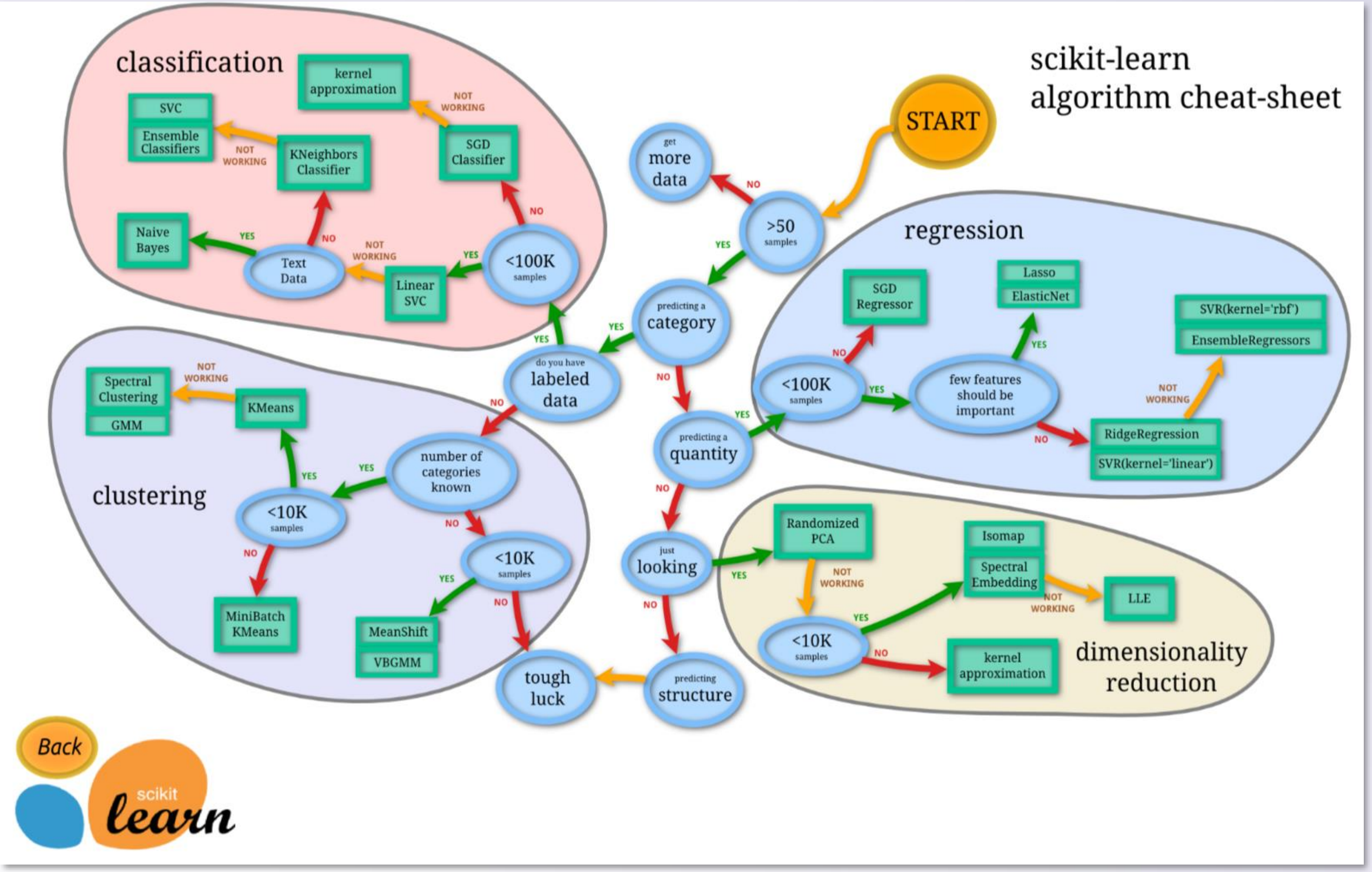
Linear Regression

Randomforest

Neural Network

```
import pandas as pd
import numpy as np
from sklearn.model_selectio
from sklearn.preprocessing i
from sklearn.metrics import
import tensorflow as tf
from tensorflow.keras.models
from tensorflow.keras.layers
import matplotlib.pyplot as
import shap

# Load data
file_path = r"C:\tools\data2
df = pd.read_csv(file_path)
```



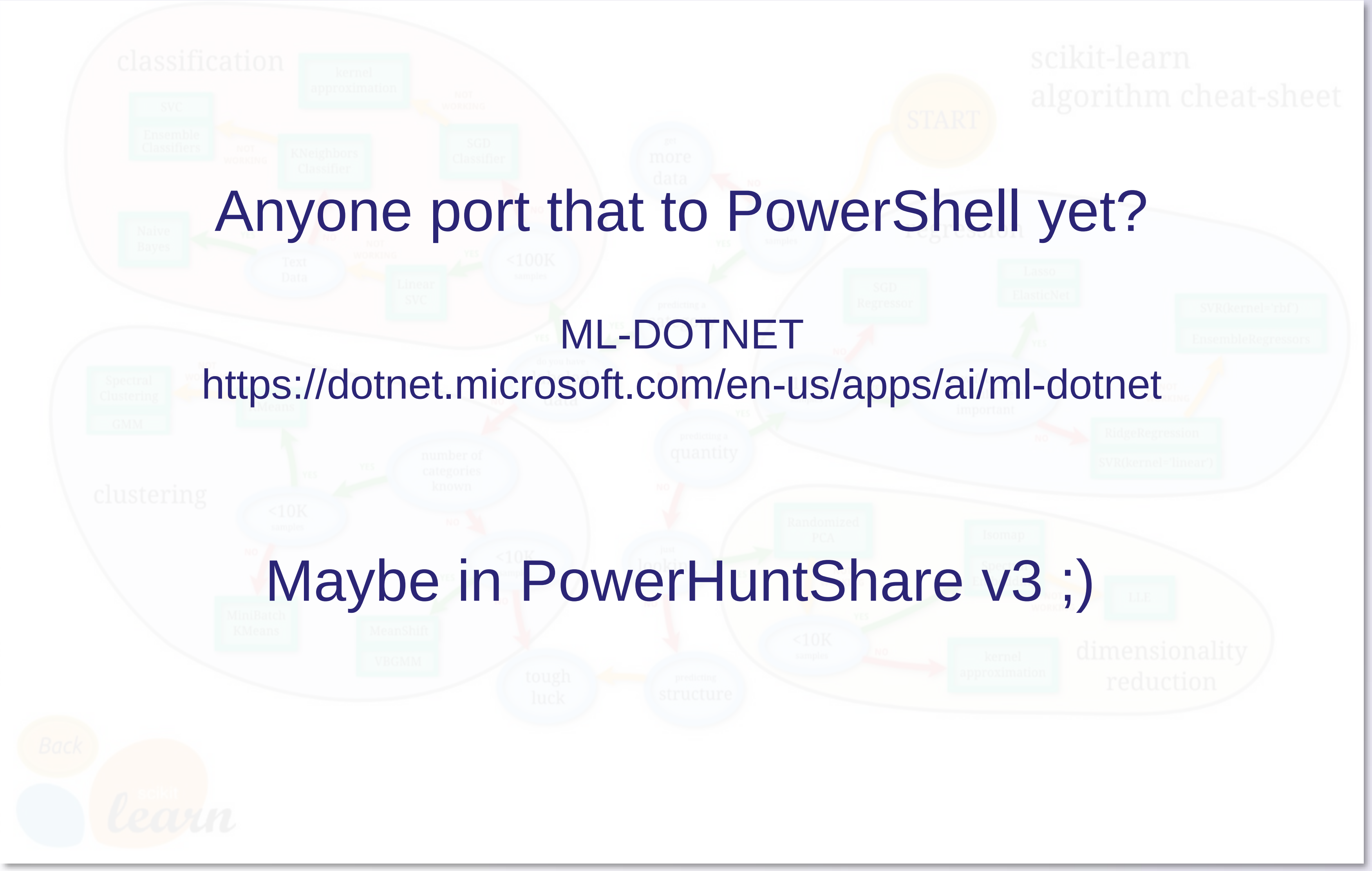
Peer **Comparison**

Predictive Models

Linear Regression
Randomforest
Neural Network

```
import pandas as pd
import numpy as np
from sklearn.model_selection
from sklearn.preprocessing i
from sklearn.metrics import
import tensorflow as tf
from tensorflow.keras.models
from tensorflow.keras.layers
import matplotlib.pyplot as
import shap

# Load data
file_path = r"C:\tools\data2
df = pd.read_csv(file_path)
```

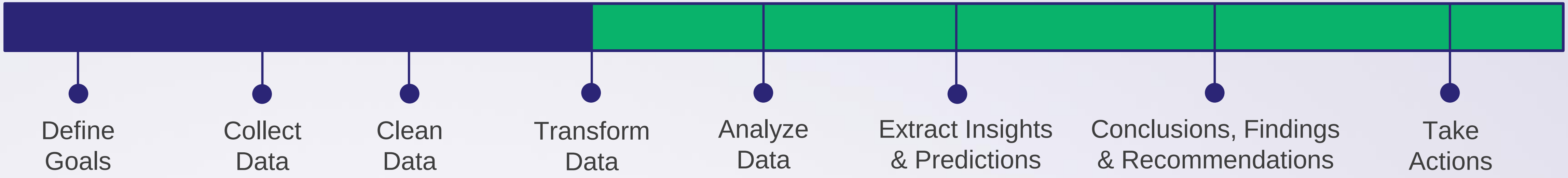


Anyone port that to PowerShell yet?

ML-DOTNET
https://dotnet.microsoft.com/en-us/apps/ai/ml-dotnet

Maybe in PowerHuntShare v3 ;)

PowerHuntShares Process



Transform Data

Static Data Labeling

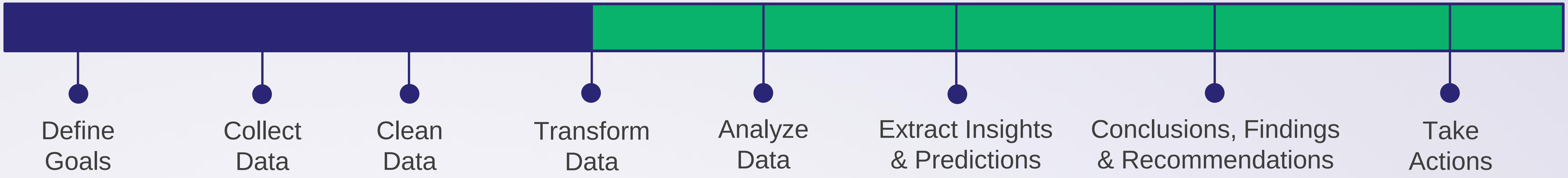
- Highly Exploitable, Interesting Files, Secrets Extraction, Stale, Empty

Dynamic Data Enrichment

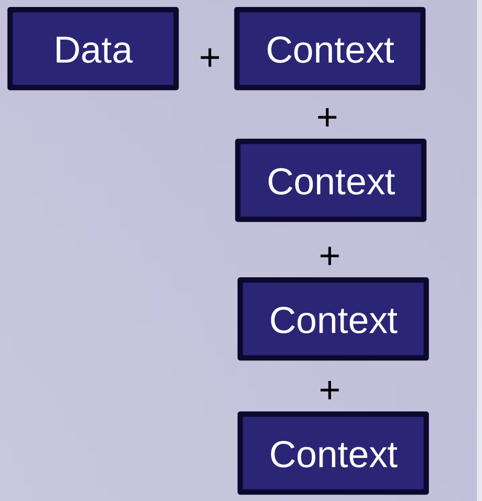
- Fingerprinting, Risk Scoring, **Peer Comparison**

Data + Context + Context + Context + Context

PowerHuntShares Process



Transform Data



Static Data Labeling

- Highly Exploitable, Interesting Files, Secrets Extraction, Stale, Empty

Dynamic Data Enrichment

- Fingerprinting, Risk Scoring, Peer Comparison, **Grouping & Similarity Scoring**

Grouping & Similarity

“How can I group similar shares so I can take fewer targeted actions?”

Grouping & Similarity

“How can I group similar shares so I can take fewer targeted actions?”

Why Group Shares?

Offensive Action Target Consolidation

- Secrets extraction
- Sensitive data extractions
- Remote code execution

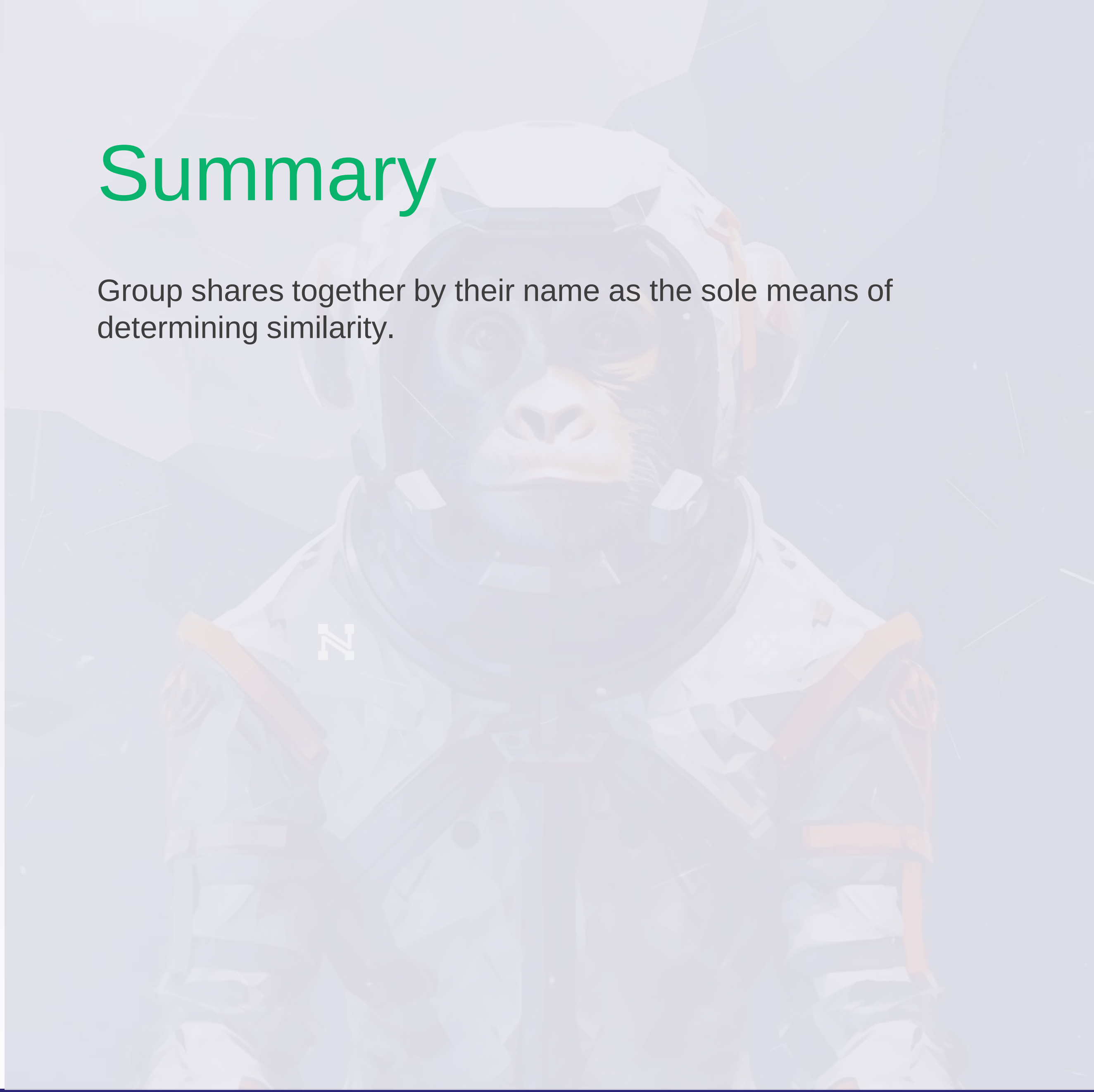
Defensive Action Target Consolidation

- Groups assets part of the same process or application with confidence
- Prioritize large groups of vulnerable assets at once
- Remediate groups of similar assets at the same time

Grouping & Similarity

“How can I group similar shares so I can take targeted actions?”

- Group by **Share Name**



Summary

Group shares together by their name as the sole means of determining similarity.

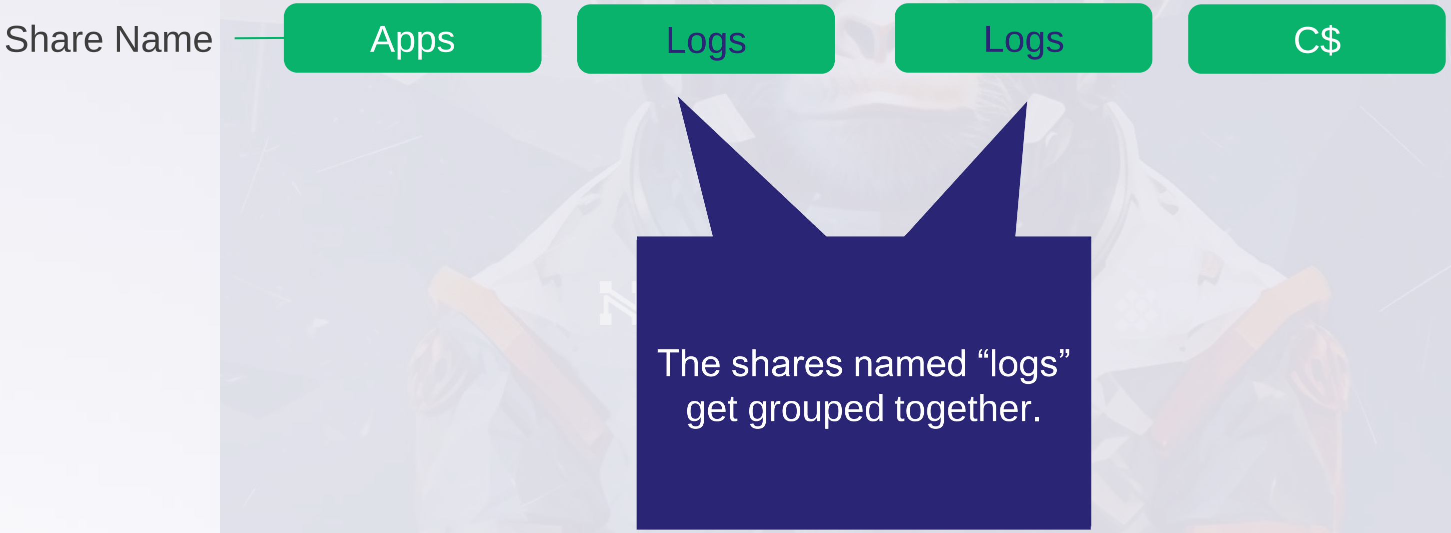
Grouping & Similarity

"How can I group similar shares so I can take targeted actions?"

- Group by **Share Name**

Summary

Group shares together by their name as the sole means of determining similarity.



Share Name

Apps

Logs

Logs

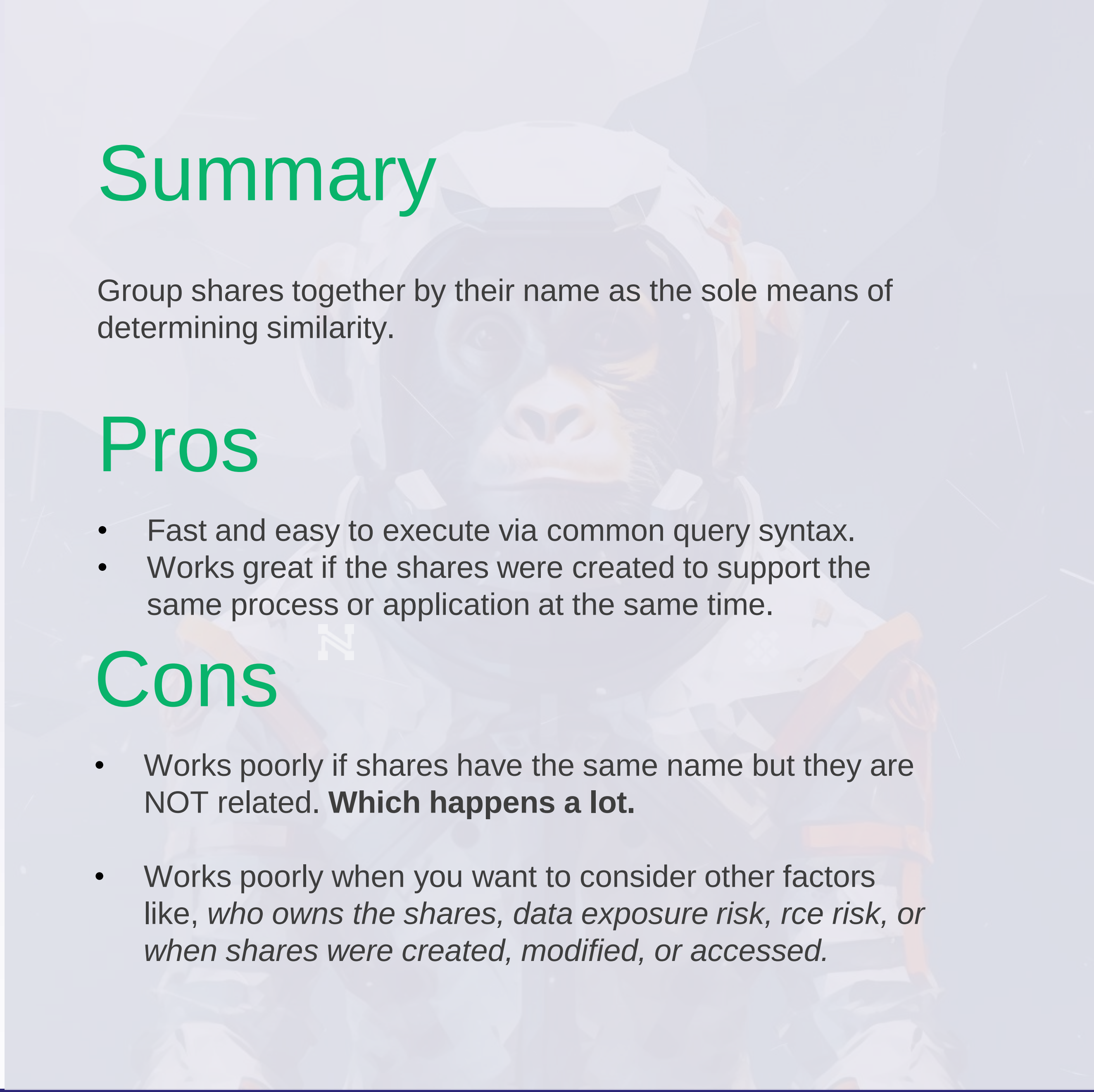
C$

The shares named "logs" get grouped together.

Grouping & Similarity

“How can I group similar shares so I can take targeted actions?”

- Group by **Share Name**



Summary

Group shares together by their name as the sole means of determining similarity.

Pros

- Fast and easy to execute via common query syntax.
- Works great if the shares were created to support the same process or application at the same time.

Cons

- Works poorly if shares have the same name but they are NOT related. **Which happens a lot.**

- Works poorly when you want to consider other factors like, *who owns the shares, data exposure risk, rce risk, or when shares were created, modified, or accessed.*

Grouping & Similarity

“How can I group similar shares so I can take targeted actions?”

- Group by **Share Name**

Summary

Group shares together by their name as the sole means of determining similarity.

Example Queries

SQL QUERY

```
SELECT ShareName, COUNT(ShareName) AS ShareCount
FROM Shares
GROUP BY ShareName
ORDER BY ShareCount DESC;
```

PowerShell Example

```
$Shares | Group-Object | Sort-Object Count -Descending |
Select-Object Count, Name
```

Grouping & Similarity

“How can I group similar shares so I can take targeted actions?”

- Group by **Share Name**

Summary

Group shares together by their name as the sole means of determining similarity.

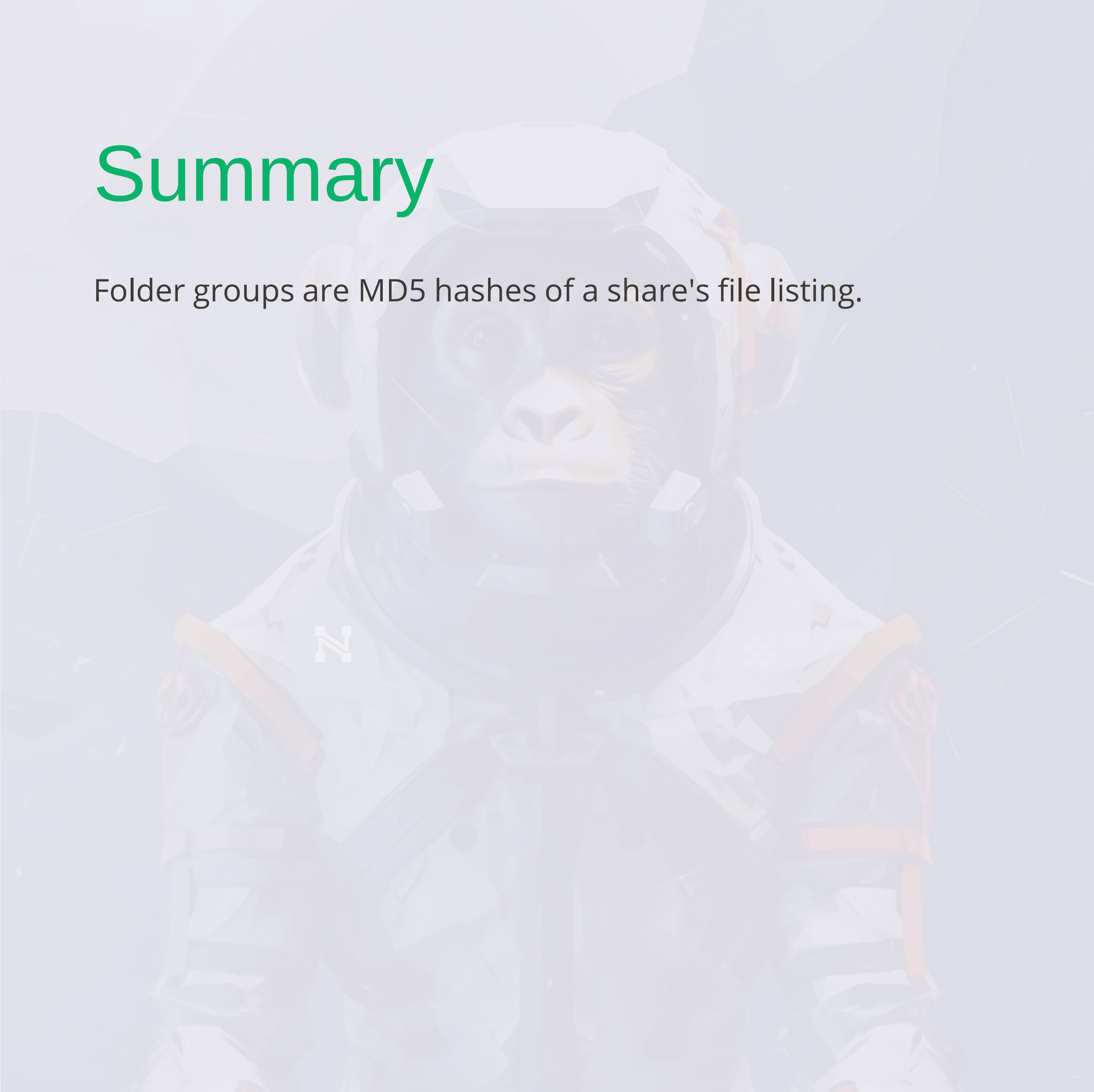
Example Output

ShareName	Count
Logs	2
Apps	1
C$	1

Grouping & Similarity

“How can I group similar shares so I can take targeted actions?”

- Group by Share Name
- Group by **Folder Group (Dir Hash)**



Summary

Folder groups are MD5 hashes of a share's file listing.

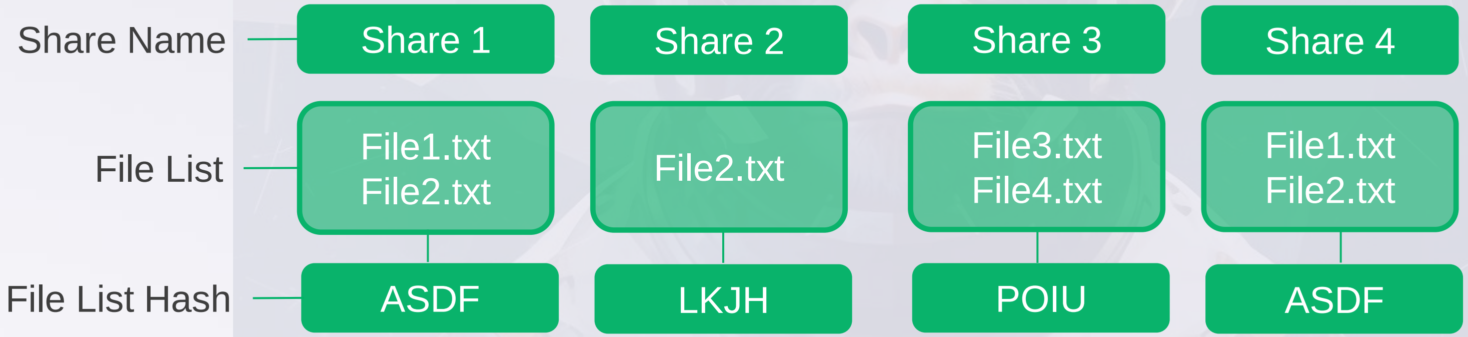
Grouping & Similarity

“How can I group similar shares so I can take targeted actions?”

- Group by Share Name
- Group by **Folder Group (Dir Hash)**

Summary

Folder groups are MD5 hashes of a share's file listing.



Share Name	Share 1	Share 2	Share 3	Share 4
File List	File1.txt File2.txt	File2.txt	File3.txt File4.txt	File1.txt File2.txt
File List Hash	ASDF	LKJH	POIU	ASDF

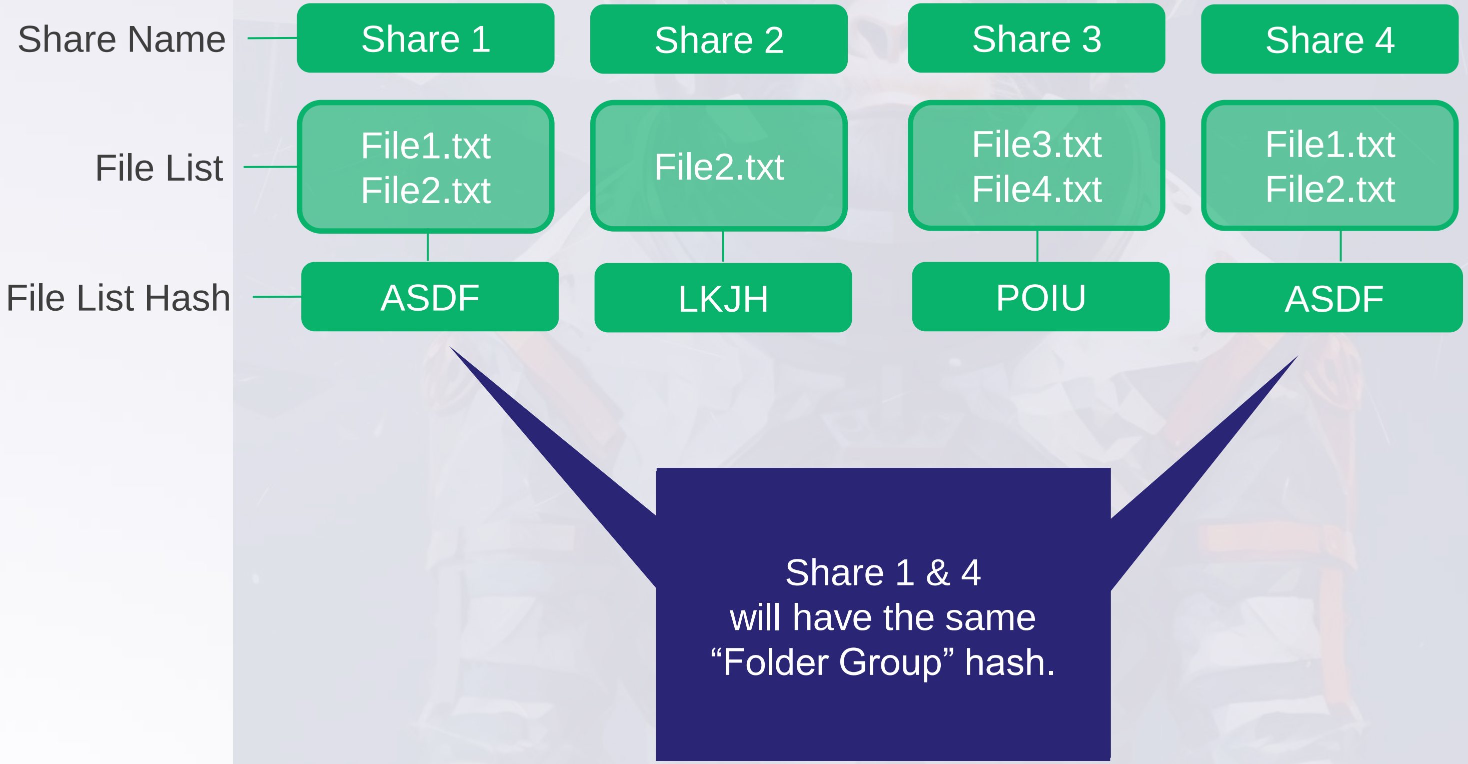
Grouping & Similarity

“How can I group similar shares so I can take targeted actions?”

- Group by Share Name
- Group by **Folder Group (Dir Hash)**

Summary

Folder groups are MD5 hashes of a share's file listing.



Share Name

File List

File List Hash

Share 1

File1.txt
File2.txt

ASDF

Share 2

File2.txt

LKJH

Share 3

File3.txt
File4.txt

POIU

Share 4

File1.txt
File2.txt

ASDF

Share 1 & 4
will have the same
“Folder Group” hash.

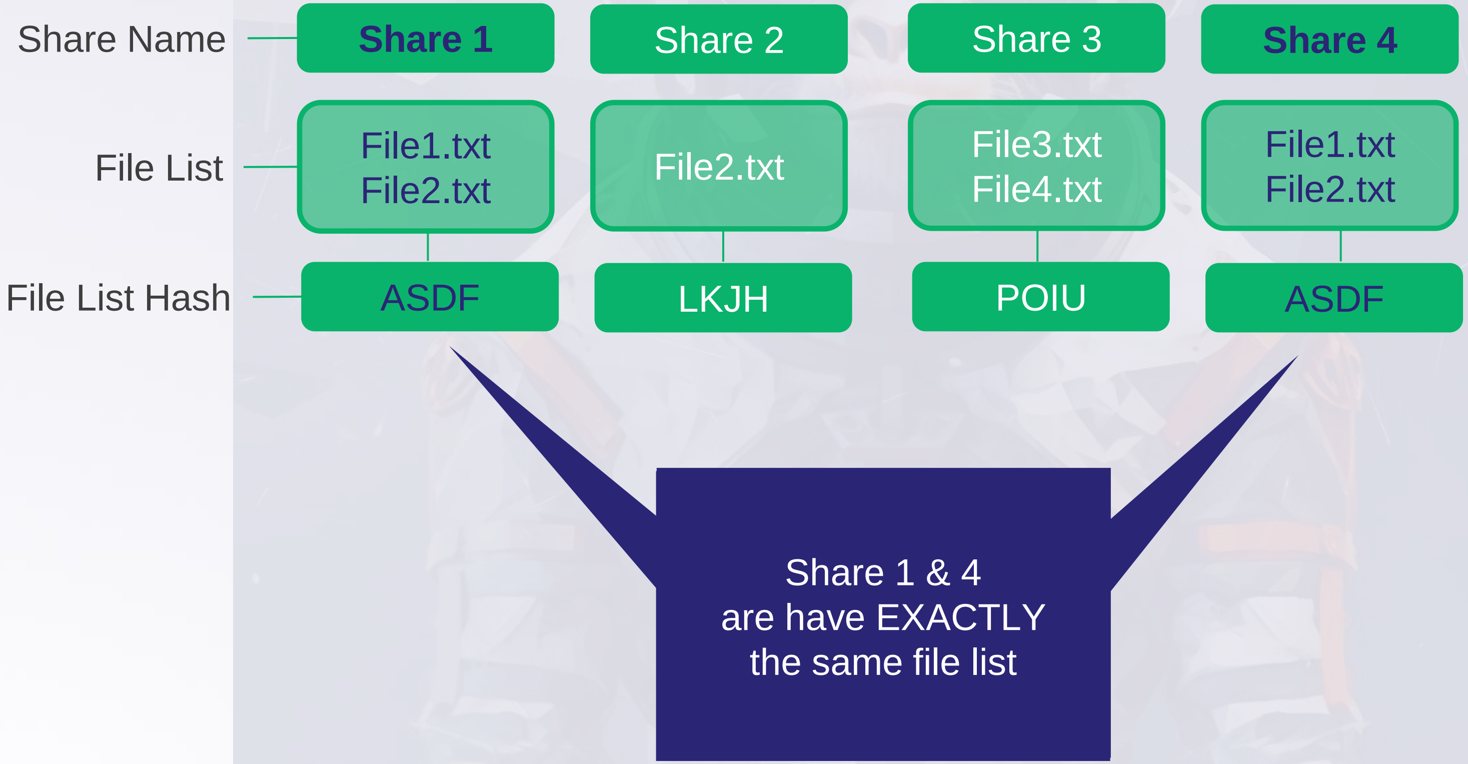
Grouping & Similarity

“How can I group similar shares so I can take targeted actions?”

- Group by Share Name
- Group by **Folder Group (Dir Hash)**

Summary

Folder groups are MD5 hashes of a share's file listing.



Share Name	**Share 1**	Share 2	Share 3	**Share 4**
File List	File1.txt File2.txt	File2.txt	File3.txt File4.txt	File1.txt File2.txt
File List Hash	ASDF	LKJH	POIU	ASDF

Share 1 & 4
are have EXACTLY
the same file list

Grouping & Similarity

“How can I group similar shares so I can take targeted actions?”

- Group by Share Name
- Group by **Folder Group (Dir Hash)**

Summary

Folder groups are MD5 hashes of a share's file listing.

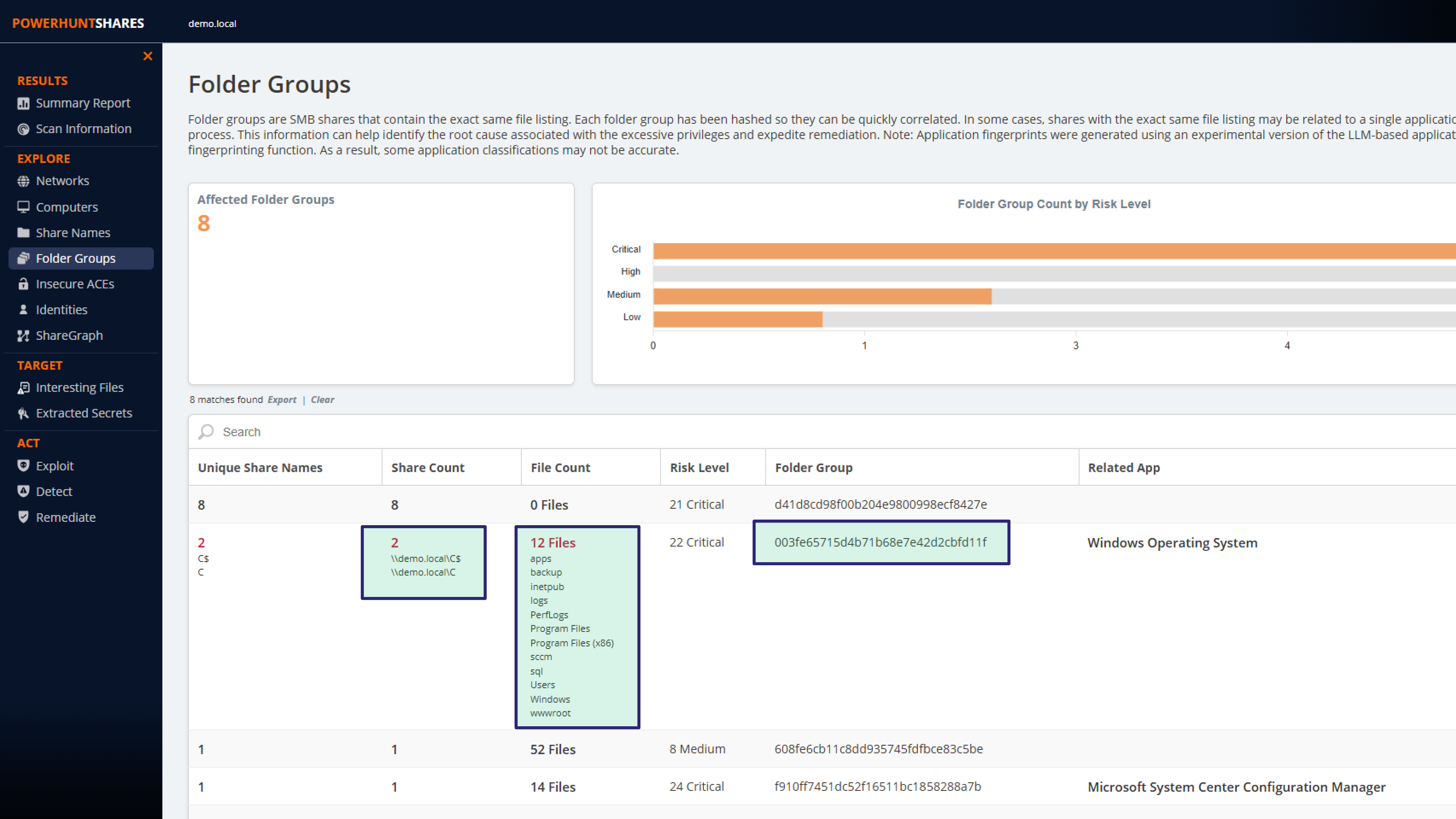
Pros

- Condensed representation of fil list for quick display, filtering and sorting.
- Fast and easy to execute via common query syntax & functions.
- Great for finding shares that have the EXACT SAME list of files at the root level.

Cons

- Works poorly when the shares DO NOT have the exact same list of files but are used by the same application. **Which happens a lot.**

- Folder groups functionality in PowerHuntShares does not currently include nested folder listings.



Folder Groups

Folder groups are SMB shares that contain the exact same file listing. Each folder group has been hashed so they can be quickly correlated. In some cases, shares with the exact same file listing may be related to a single applicatic process. This information can help identify the root cause associated with the excessive privileges and expedite remediation. Note: Application fingerprints were generated using an experimental version of the LLM-based applicat fingerprinting function. As a result, some application classifications may not be accurate.

Affected Folder Groups

8

Folder Group Count by Risk Level

8 matches found *Export* | *Clear*

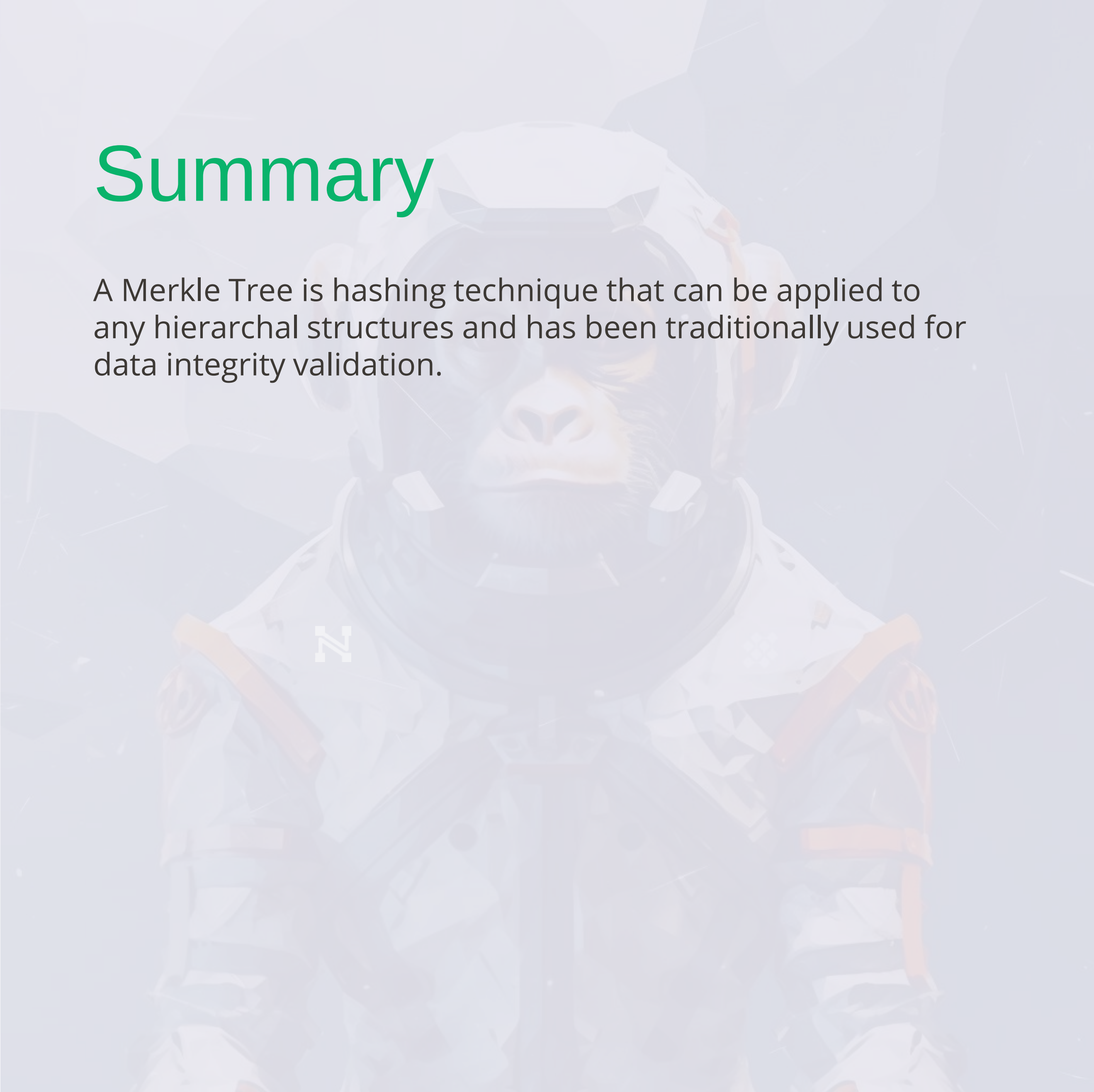
Search

Unique Share Names	Share Count	File Count	Risk Level	Folder Group	Related App
8	8	0 Files	21 Critical	d41d8cd98f00b204e9800998ecf8427e	
2 C$ C	2 \\demo.local\C$ \\demo.local\C	12 Files apps backup inetpub logs PerfLogs Program Files Program Files (x86) sccm sql Users Windows wwwroot	22 Critical	003fe65715d4b71b68e7e42d2cbfd11f	Windows Operating System
1	1	52 Files	8 Medium	608fe6cb11c8dd935745fdfbce83c5be	
1	1	14 Files	24 Critical	f910ff7451dc52f16511bc1858288a7b	Microsoft System Center Configuration Manager

Grouping & Similarity

“How can I group similar shares so I can take targeted actions?”

- Group by Share Name
- Group by Folder Group (Dir Hash)
- Group by **Merkle Hash (Nested Dir Hash)**



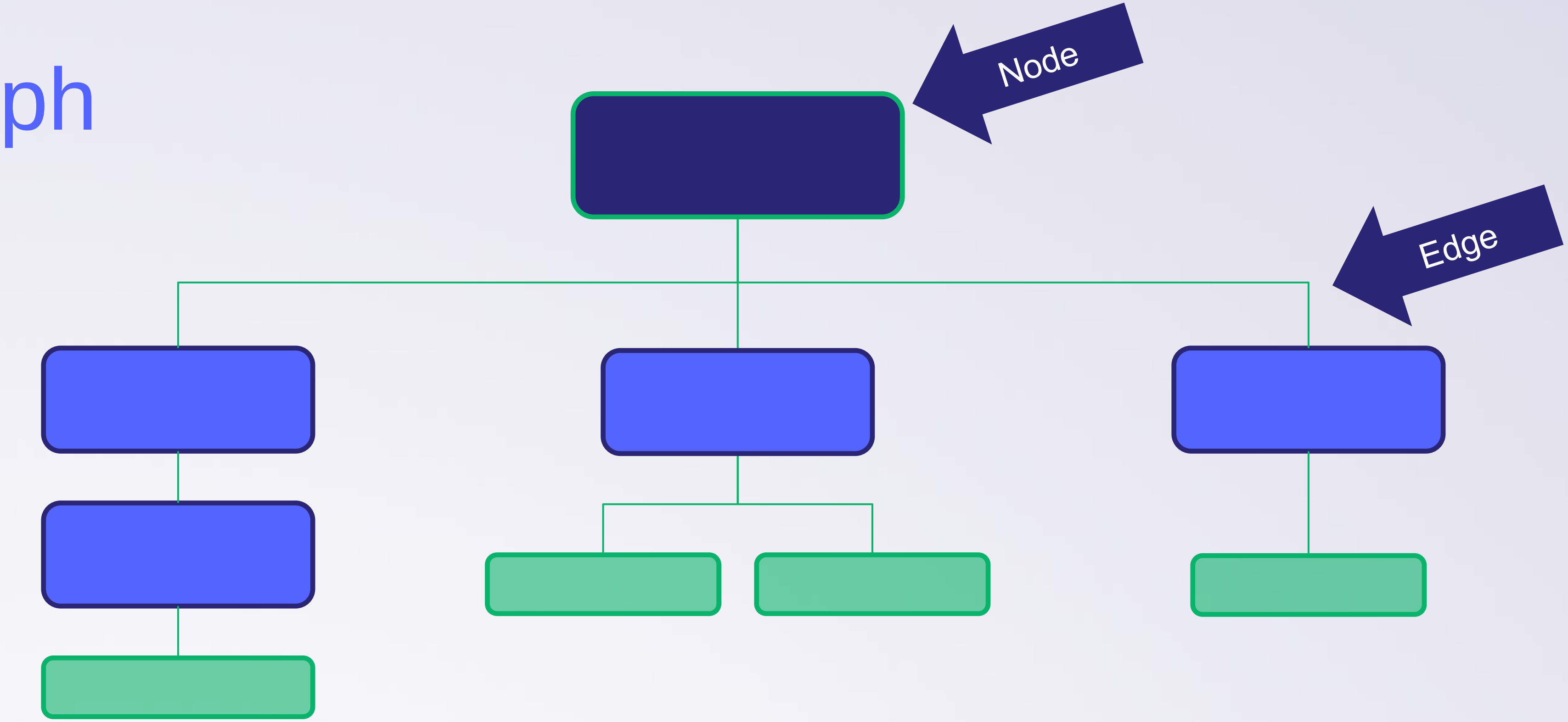
Summary

A Merkle Tree is hashing technique that can be applied to any hierarchal structures and has been traditionally used for data integrity validation.

Merkle Trees

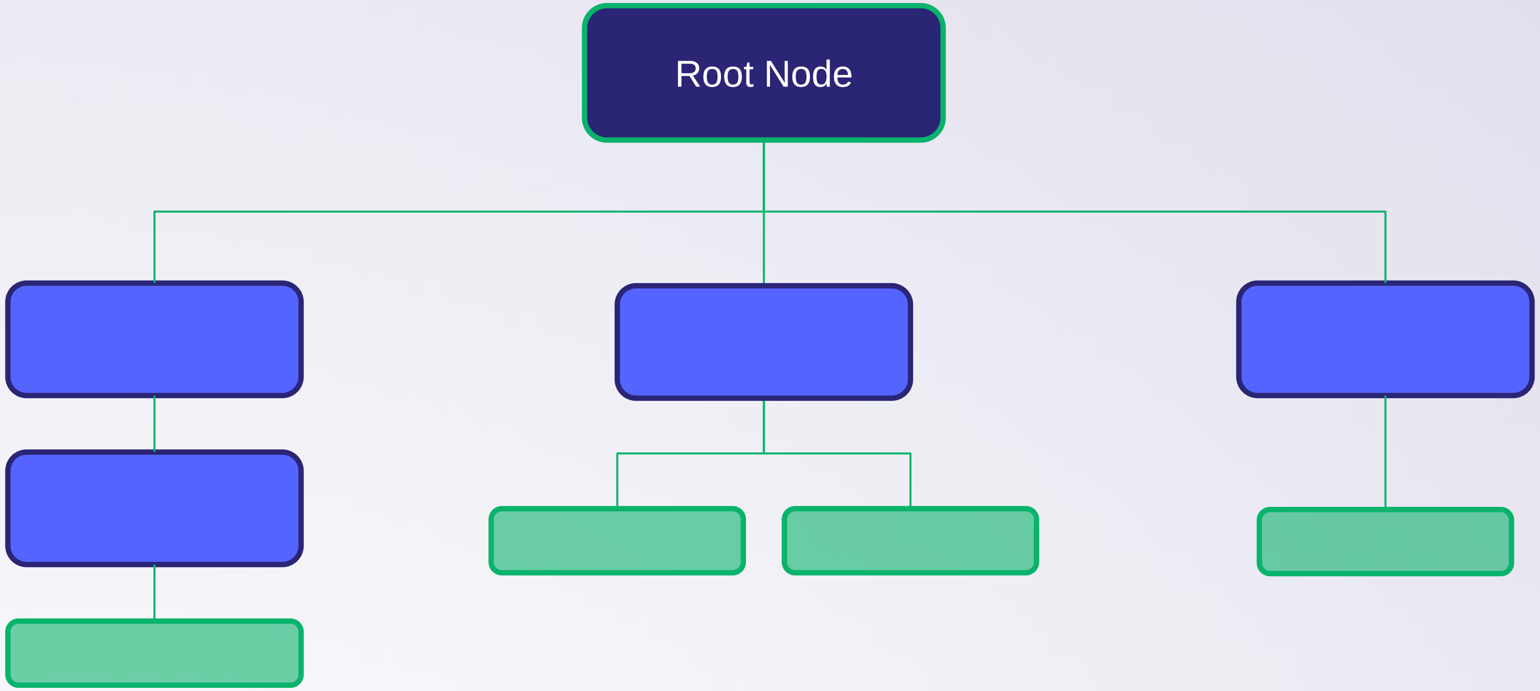
Hierarchal Graph

A Merkle Tree is hashing technique that can be applied to any hierarchal structures and has been traditionally used for data integrity validation.



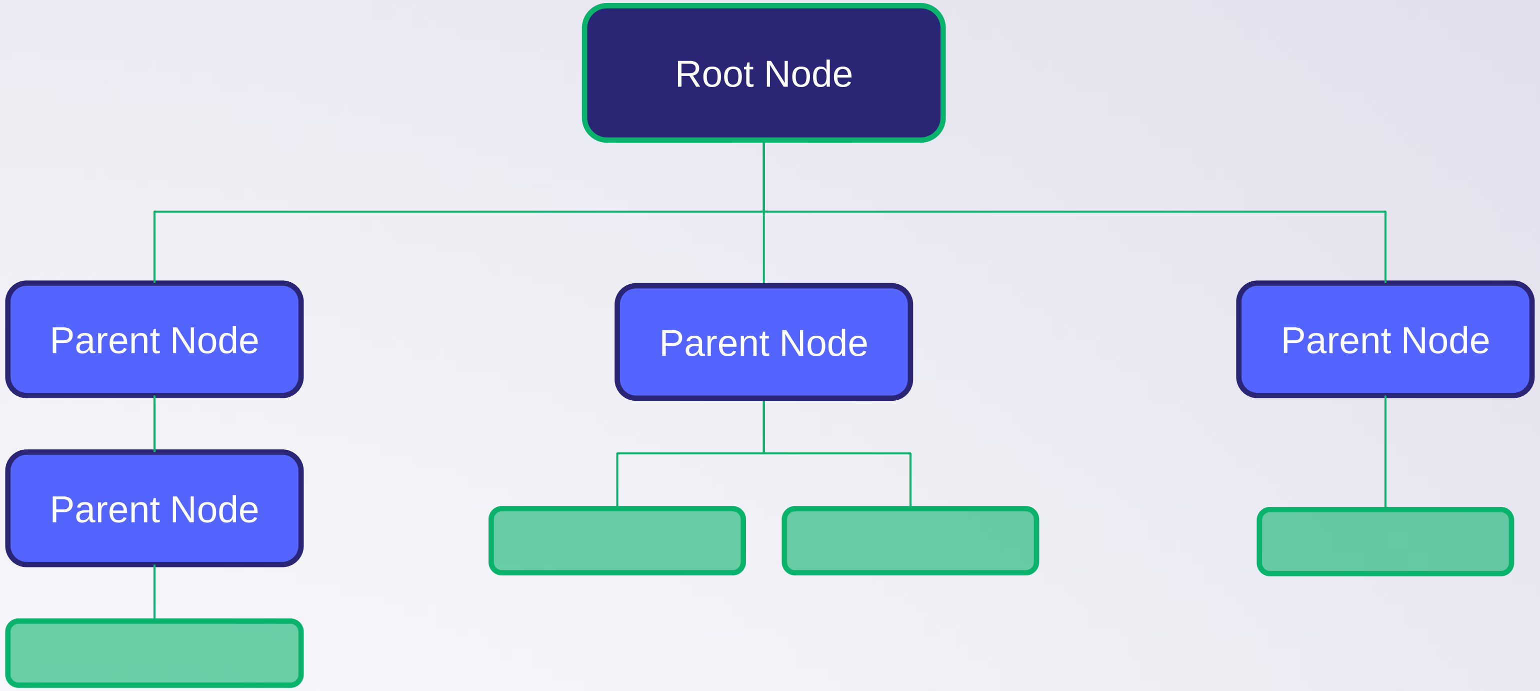
Merkle Trees

Root Nodes



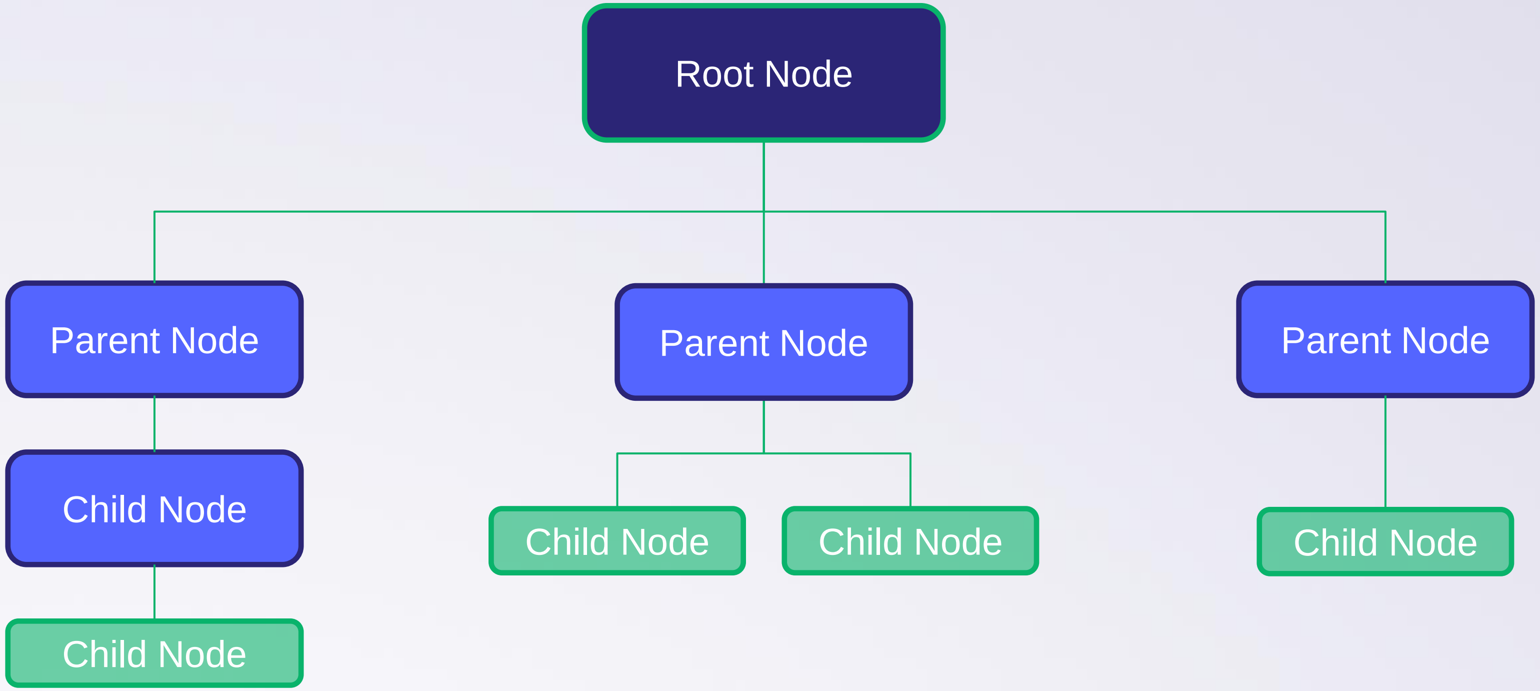
Merkle Trees

Parent Nodes



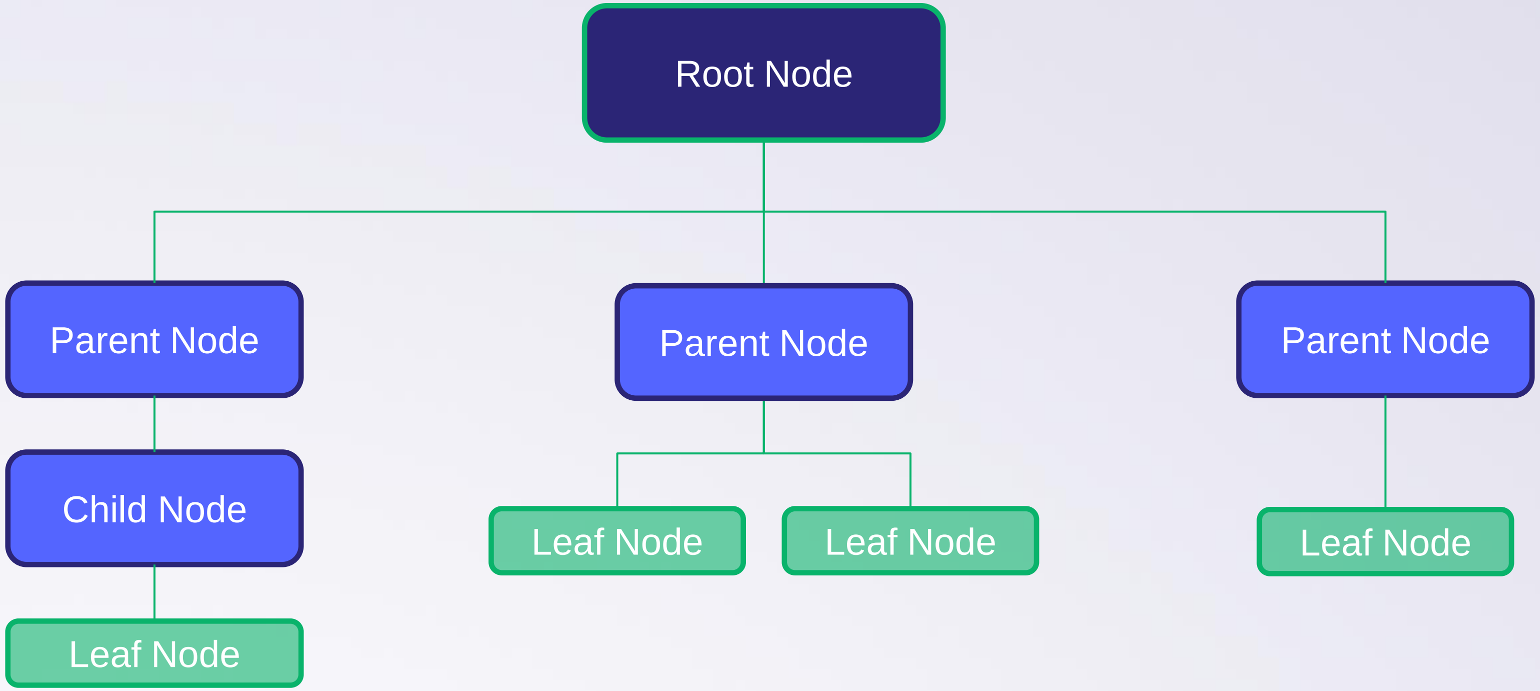
Merkle Trees

Child Nodes



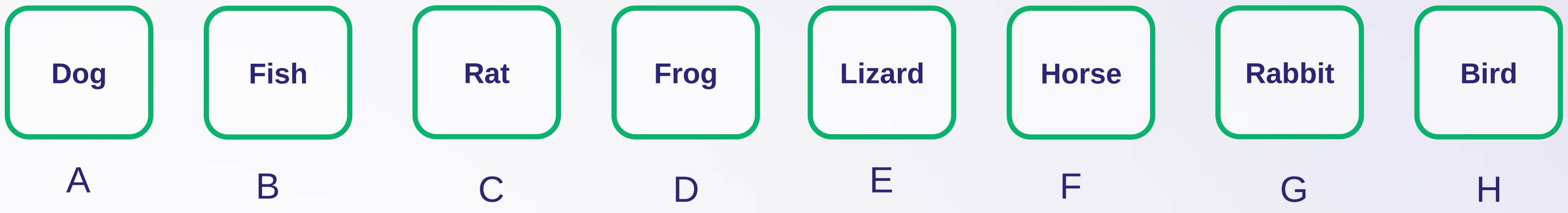
Merkle Trees

Leaf Nodes



Merkle Trees

Hashing Process



Dog A

Fish B

Rat C

Frog D

Lizard E

Horse F

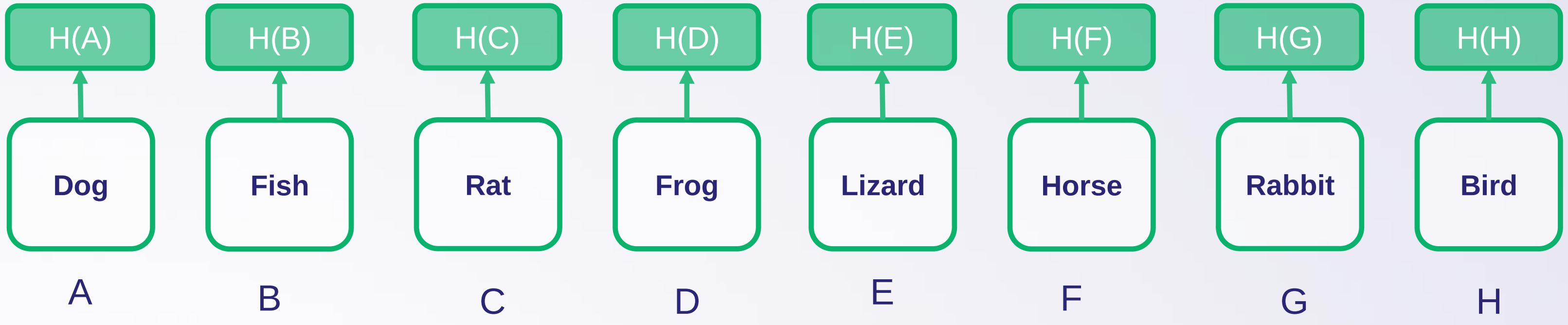
Rabbit G

Bird H

Merkle Trees

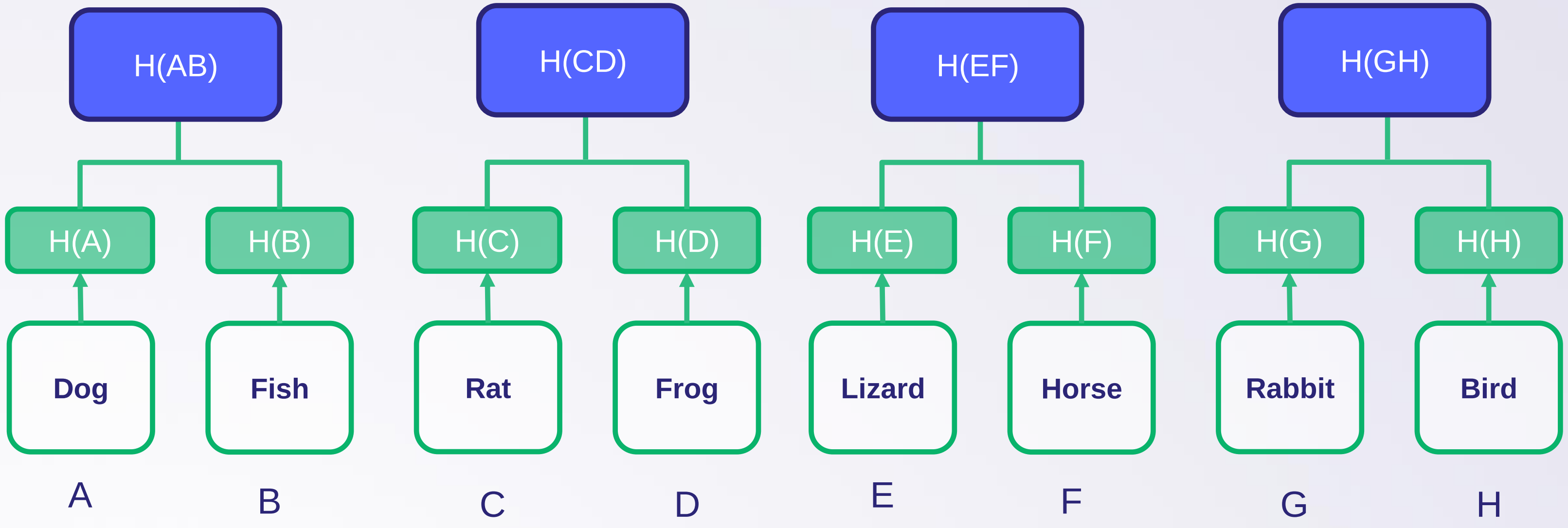
Hashing Process

1. Hash the leaf node data.



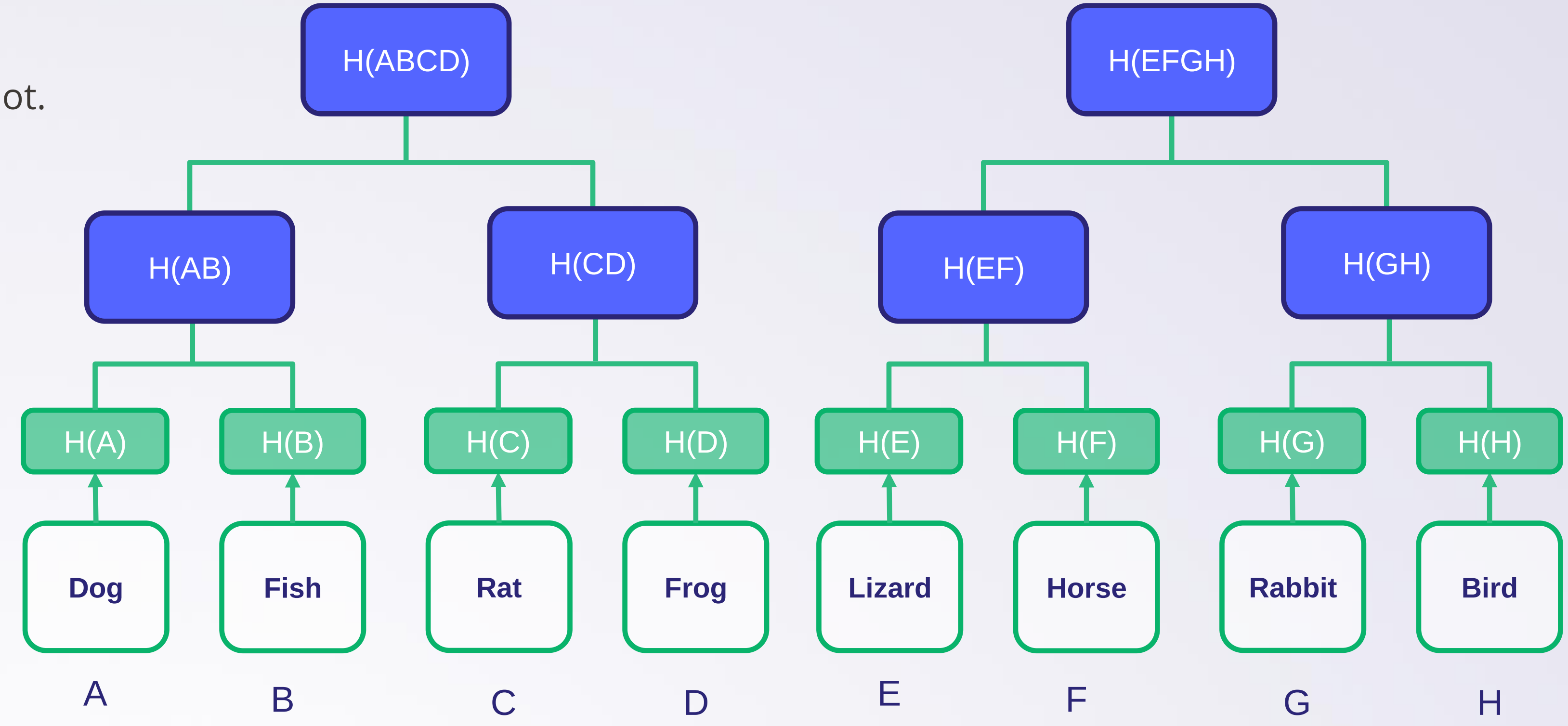
Merkle Trees
Hashing Process

1. Hash the leaf node data.
2. Group the leaf nodes into pairs and hash their hashes.



Merkle Trees
Hashing Process

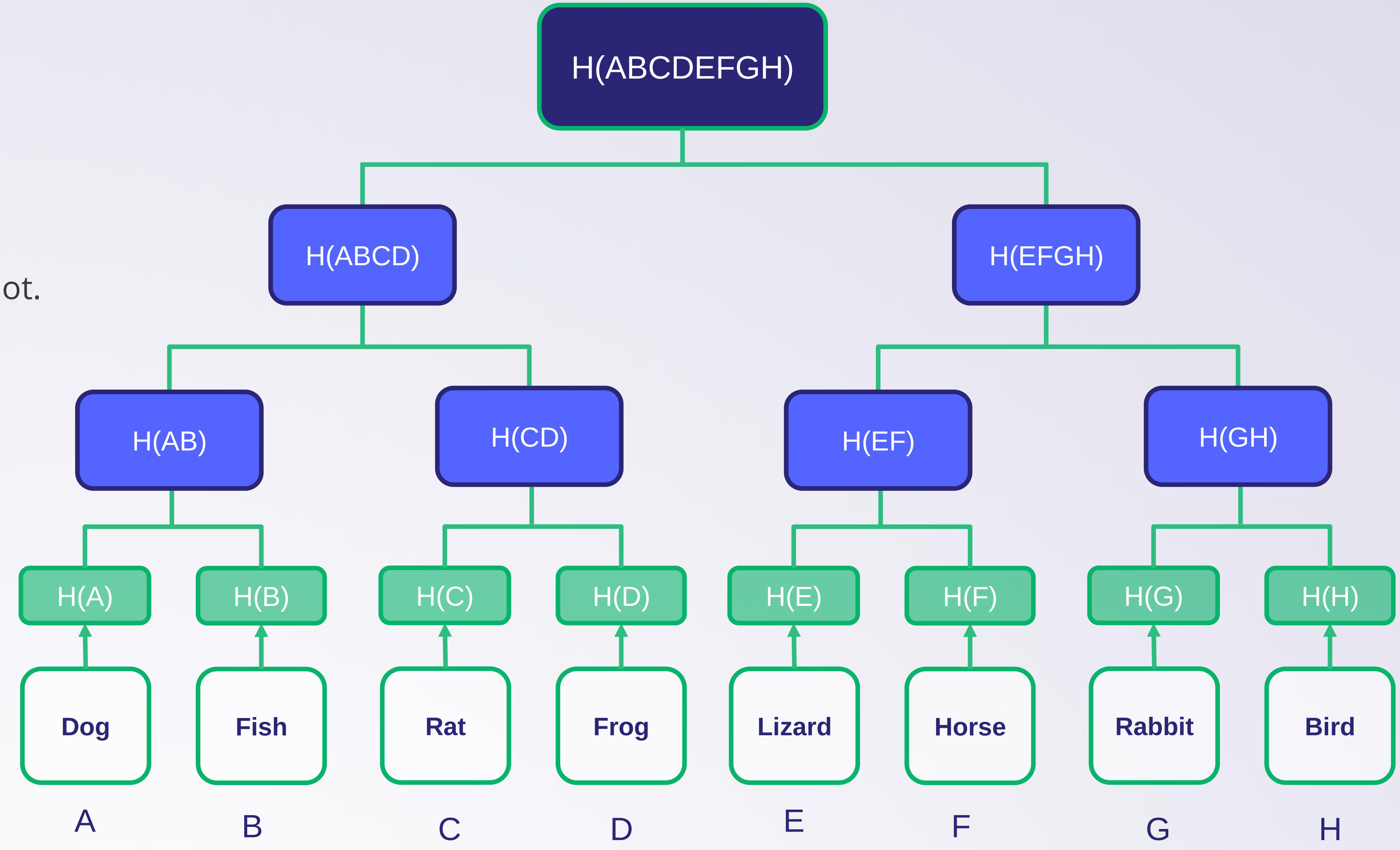
1. Hash the leaf node data.
2. Group the leaf nodes into pairs and hash their hashes.
3. Repeat with the parent nodes until root.



Merkle Trees

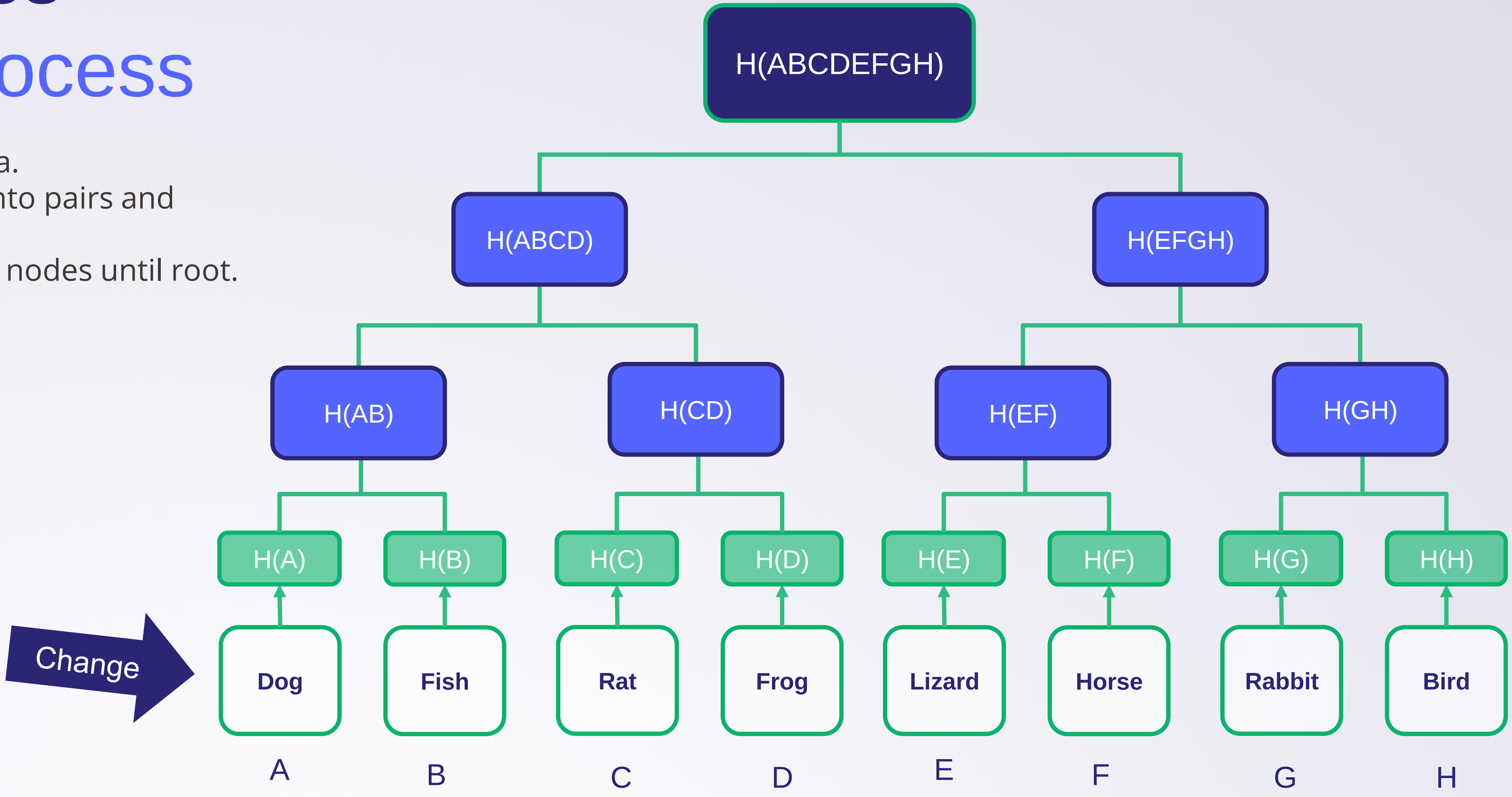
Hashing Process

1. Hash the leaf node data.
2. Group the leaf nodes into pairs and hash their hashes.
3. Repeat with the parent nodes until root.



Merkle Trees
Hashing Process

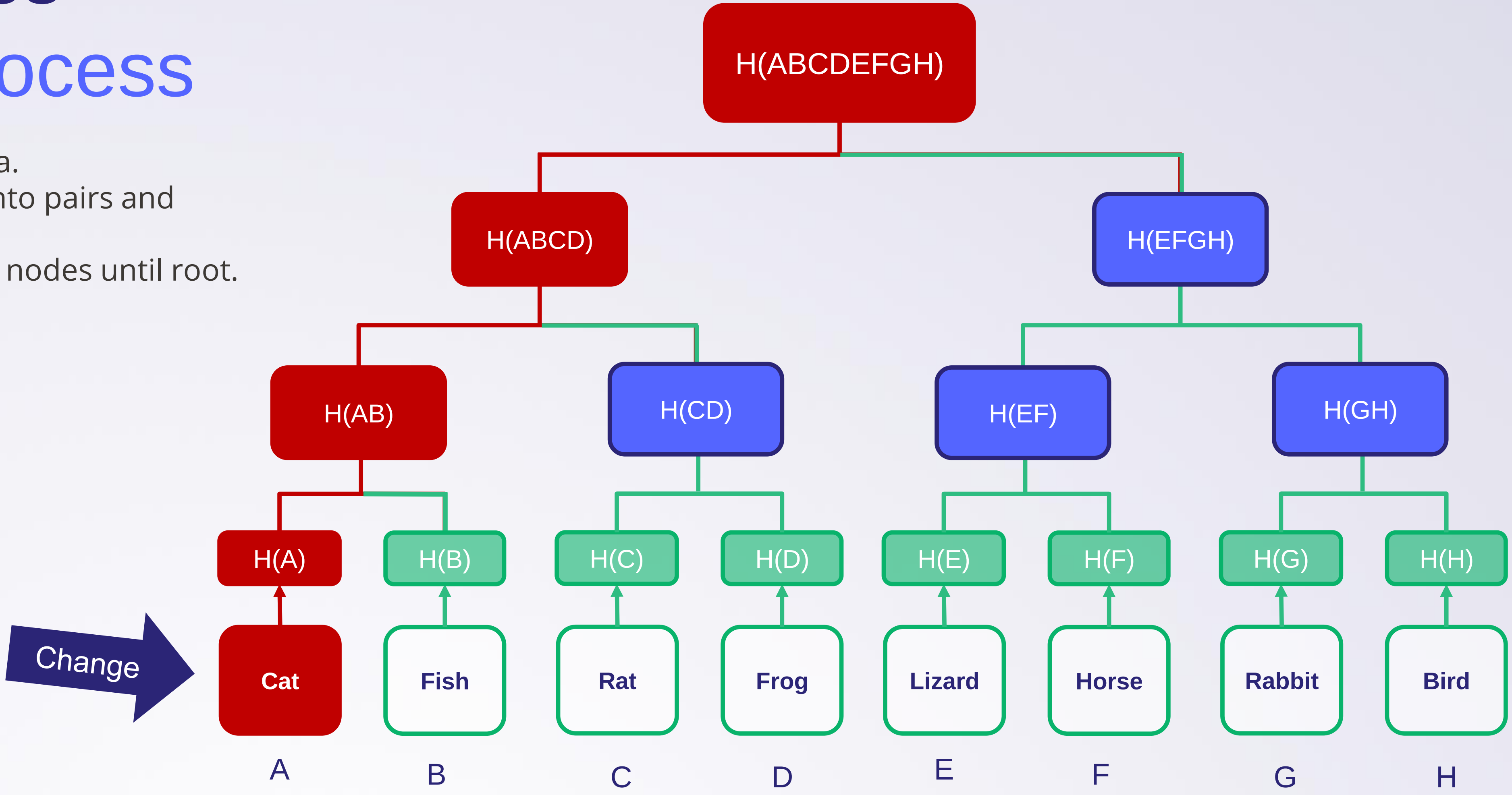
1. Hash the leaf node data.
2. Group the leaf nodes into pairs and hash their hashes.
3. Repeat with the parent nodes until root.



Merkle Trees

Hashing Process

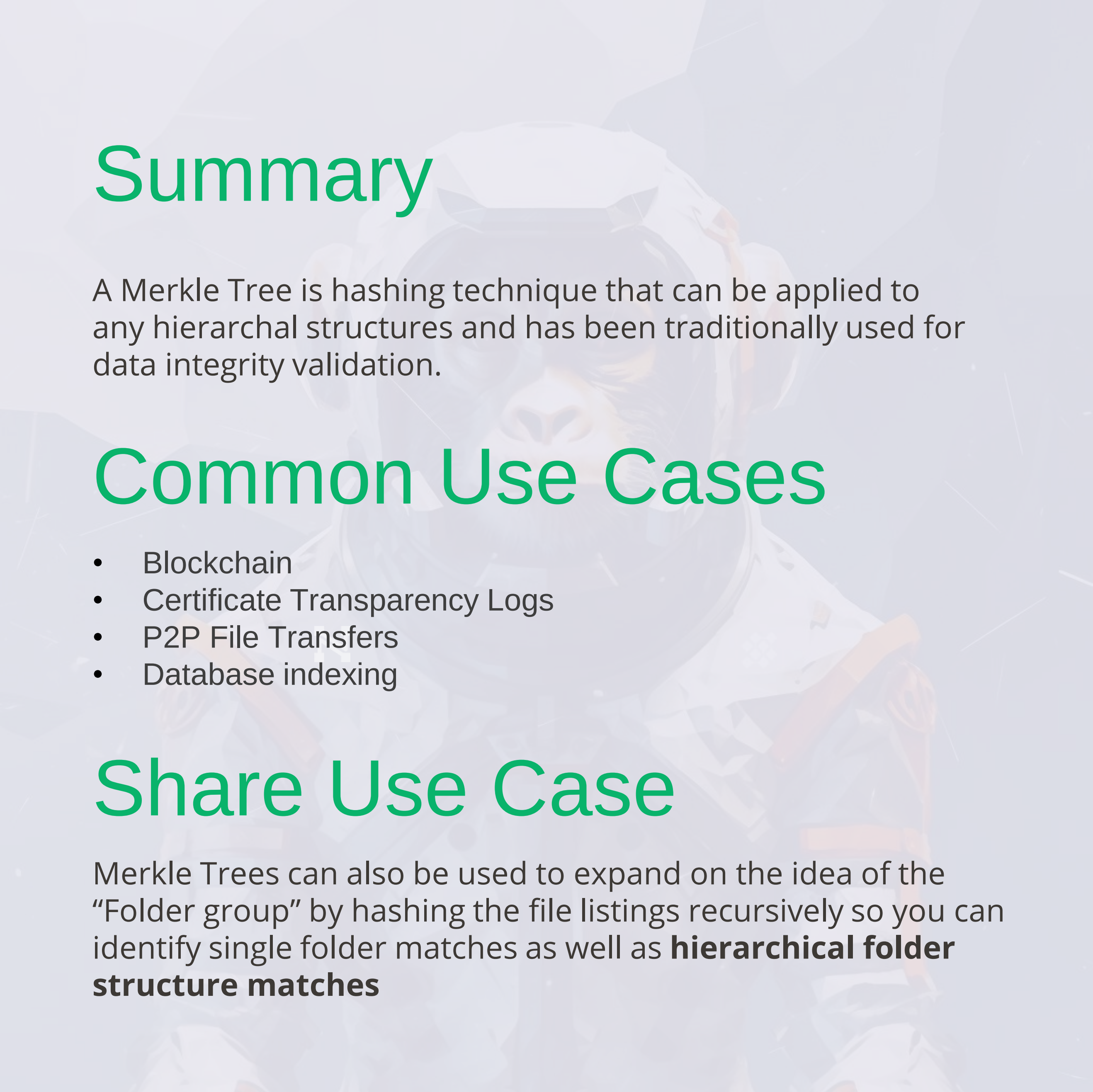
1. Hash the leaf node data.
2. Group the leaf nodes into pairs and hash their hashes.
3. Repeat with the parent nodes until root.



Grouping & Similarity

“How can I group similar shares so I can take targeted actions?”

- Group by Share Name
- Group by Folder Group (Dir Hash)
- Group by **Merkle Hash (Nested Dir Hash)**



Summary

A Merkle Tree is hashing technique that can be applied to any hierarchal structures and has been traditionally used for data integrity validation.

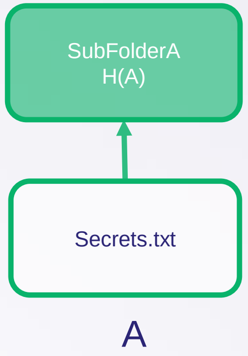
Common Use Cases

- Blockchain
- Certificate Transparency Logs
- P2P File Transfers
- Database indexing

Share Use Case

Merkle Trees can also be used to expand on the idea of the “Folder group” by hashing the file listings recursively so you can identify single folder matches as well as **hierarchical folder structure matches**

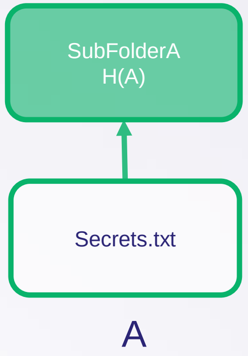
Merkle Tree (Modified)
Share Use Case



Merkle Tree (Modified)

Share Use Case

1. HASH 1 - H(A) → Simple Folder List Hashes

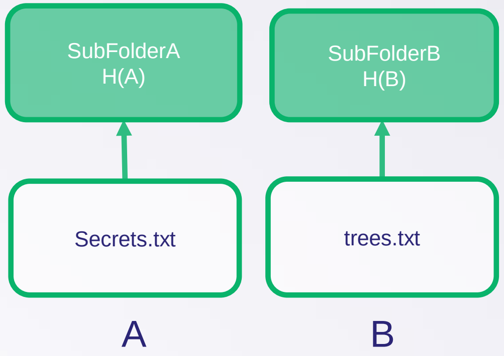


Merkle Tree (Modified)

Share Use Case

1. HASH 1 - H(A)
2. HASH 2 - H(B)

Simple Folder List Hashes

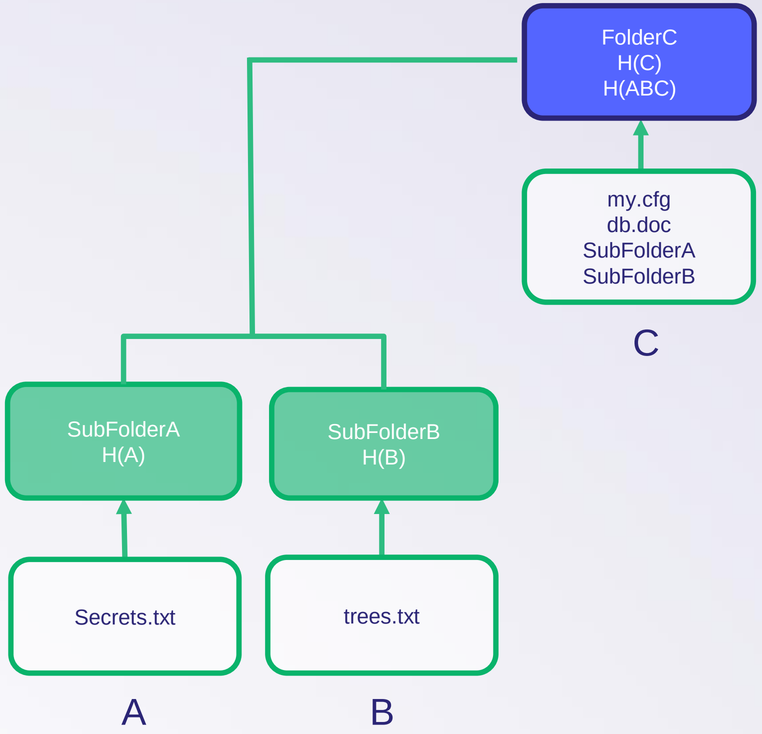


Merkle Tree (Modified)

Share Use Case

1. HASH 1 - H(A)
2. HASH 2 - H(B)
3. HASH 3 - H(C)

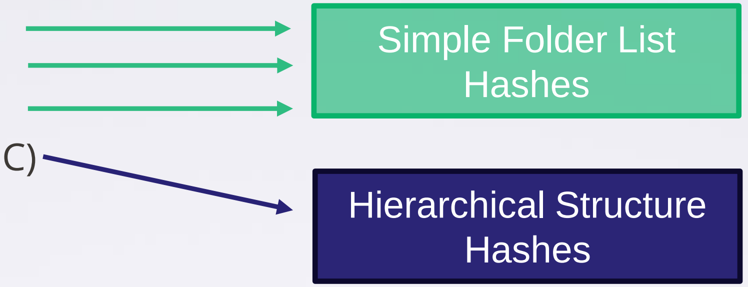
Simple Folder List Hashes



Merkle Tree (Modified)

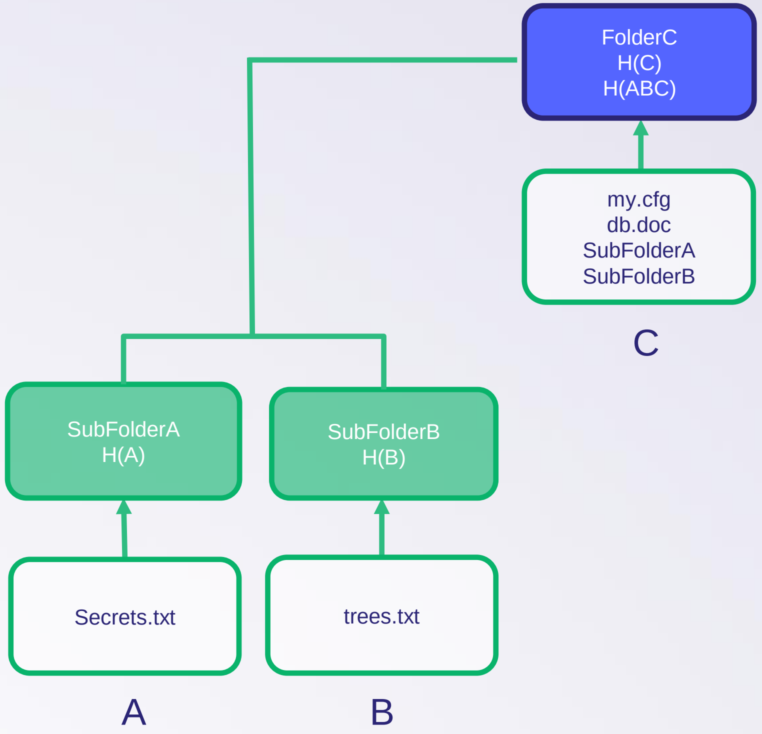
Share Use Case

1. HASH 1 - H(A)
2. HASH 2 - H(B)
3. HASH 3 - H(C)
4. HASH 4 - H(ABC)

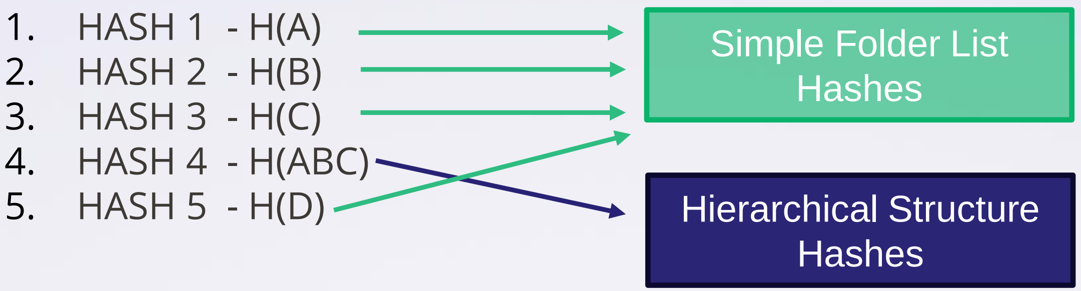


Simple Folder List Hashes

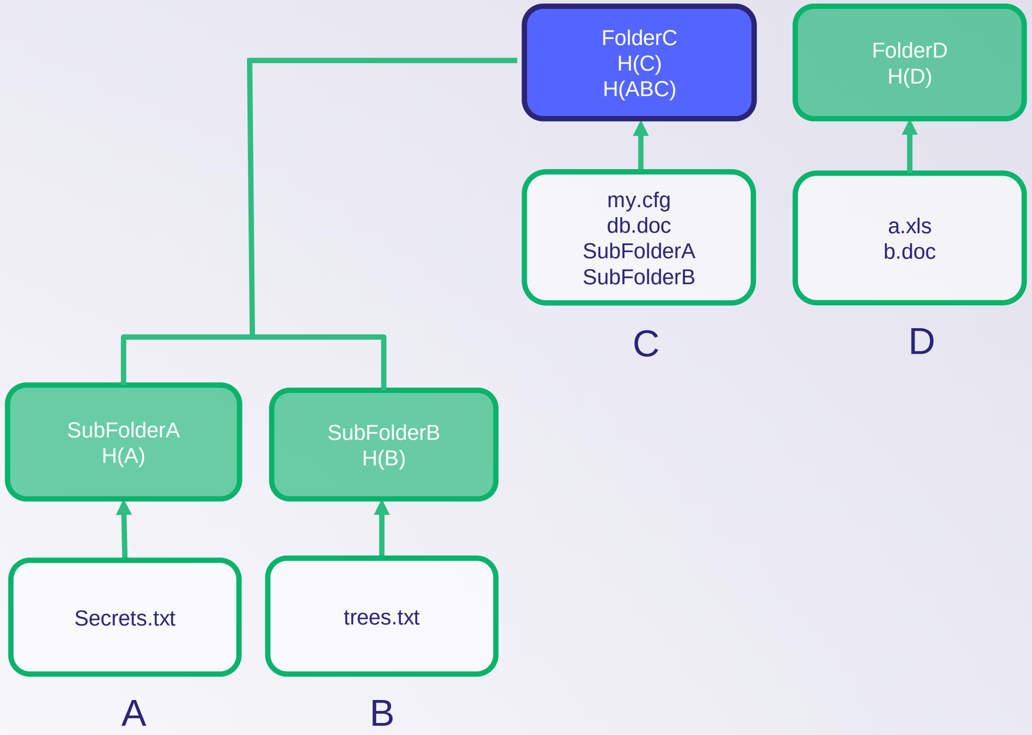
Hierarchical Structure Hashes



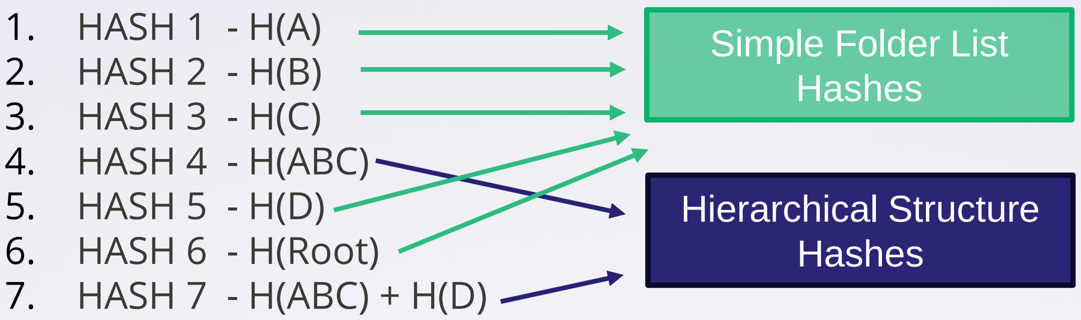
Merkle Tree (Modified)
Share Use Case



1. HASH 1 - H(A)
2. HASH 2 - H(B)
3. HASH 3 - H(C)
4. HASH 4 - H(ABC)
5. HASH 5 - H(D)



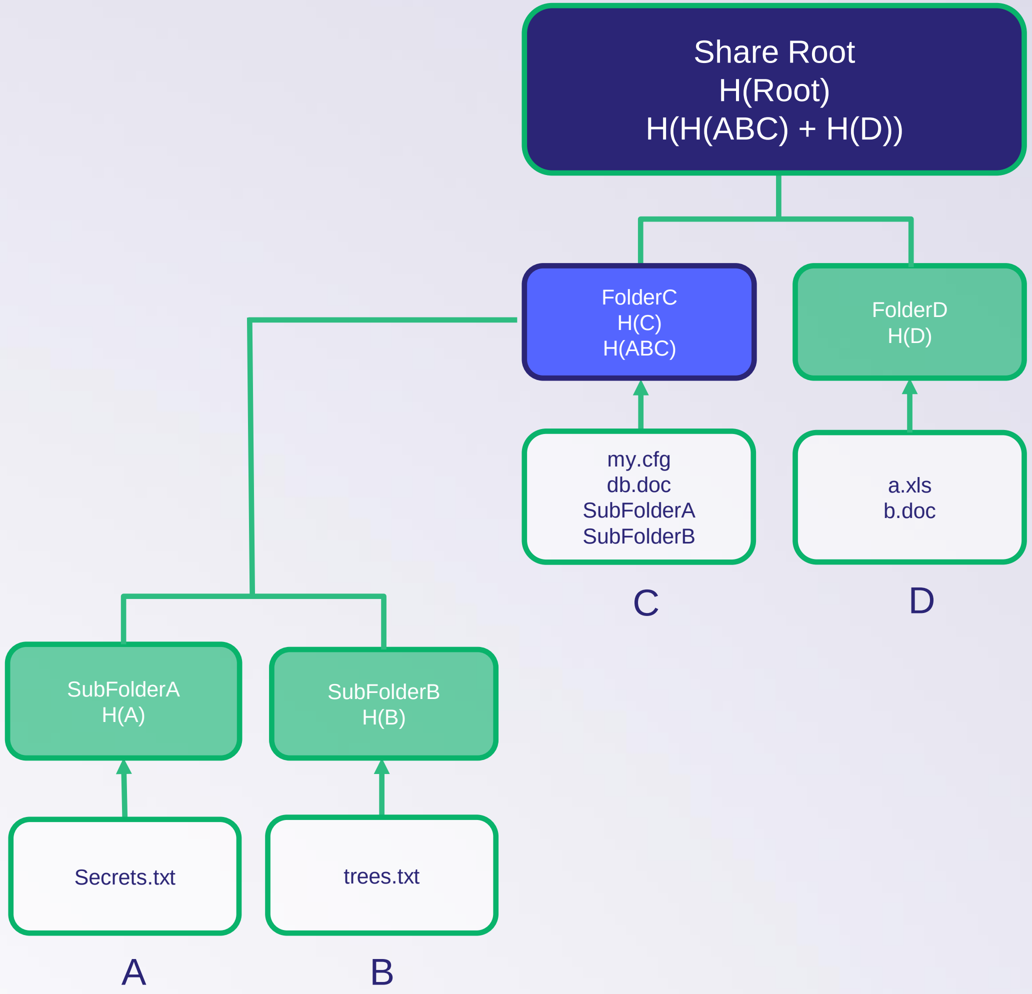
Merkle Tree (Modified)
Share Use Case



1. HASH 1 - H(A)
2. HASH 2 - H(B)
3. HASH 3 - H(C)
4. HASH 4 - H(ABC)
5. HASH 5 - H(D)
6. HASH 6 - H(Root)
7. HASH 7 - H(ABC) + H(D)

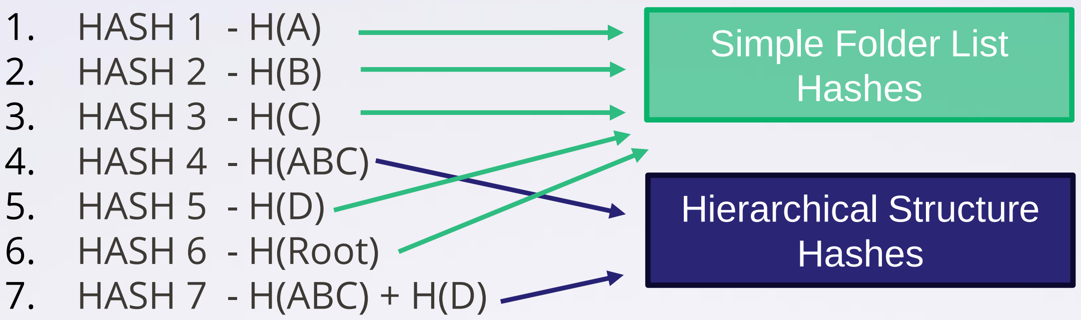
Simple Folder List Hashes

Hierarchical Structure Hashes



Merkle Tree (Modified)

Share Use Case

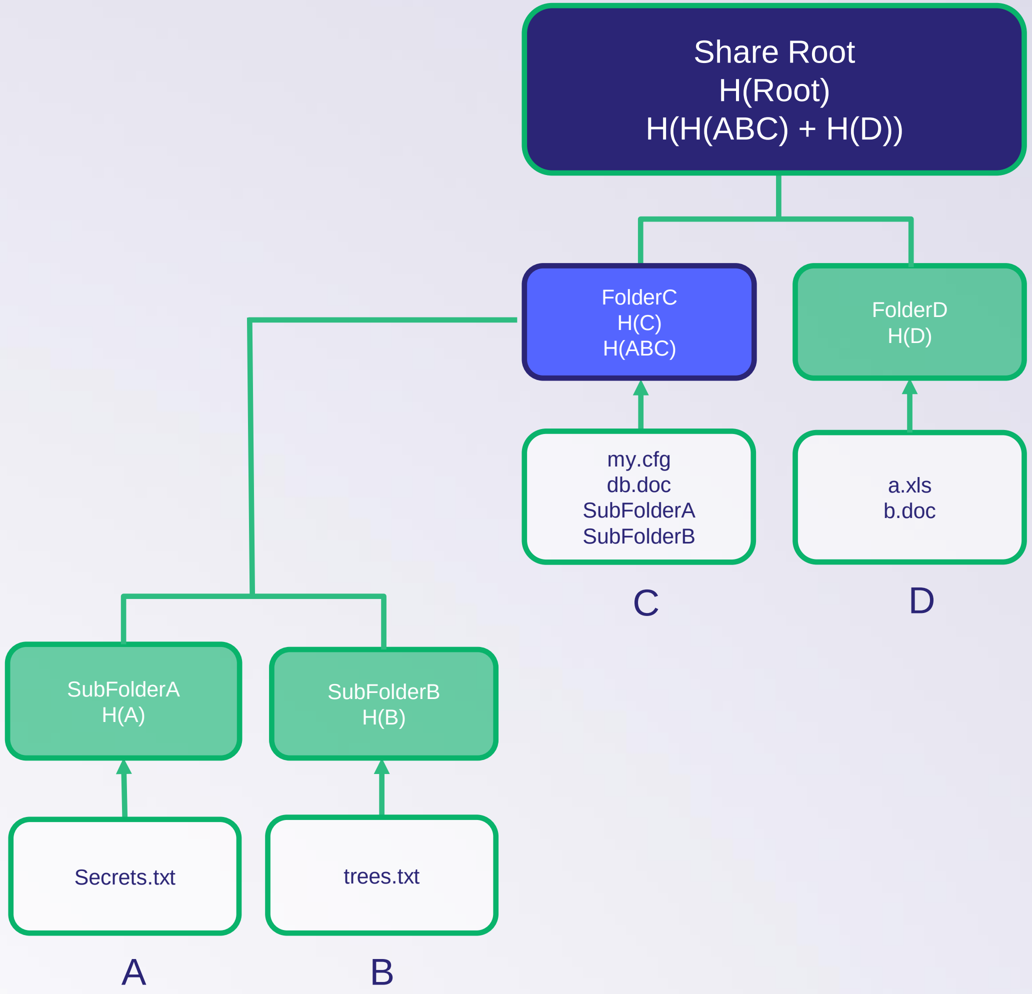


1. HASH 1 - H(A)
2. HASH 2 - H(B)
3. HASH 3 - H(C)
4. HASH 4 - H(ABC)
5. HASH 5 - H(D)
6. HASH 6 - H(Root)
7. HASH 7 - H(ABC) + H(D)

Simple Folder List Hashes

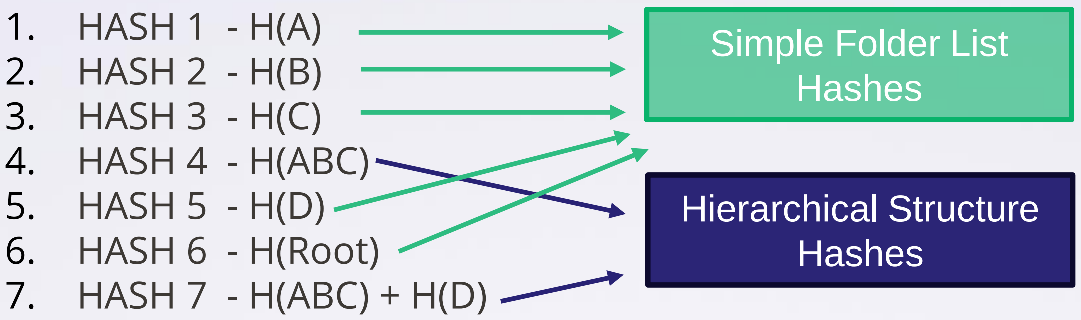
Hierarchical Structure Hashes

If we store all the hashes in a table, we can then perform simple SQL GROUP BY operations like the “Folder Groups”, but this time we can also see groups of folder hierarchies. ☺



Merkle Tree (Modified)

Share Use Case

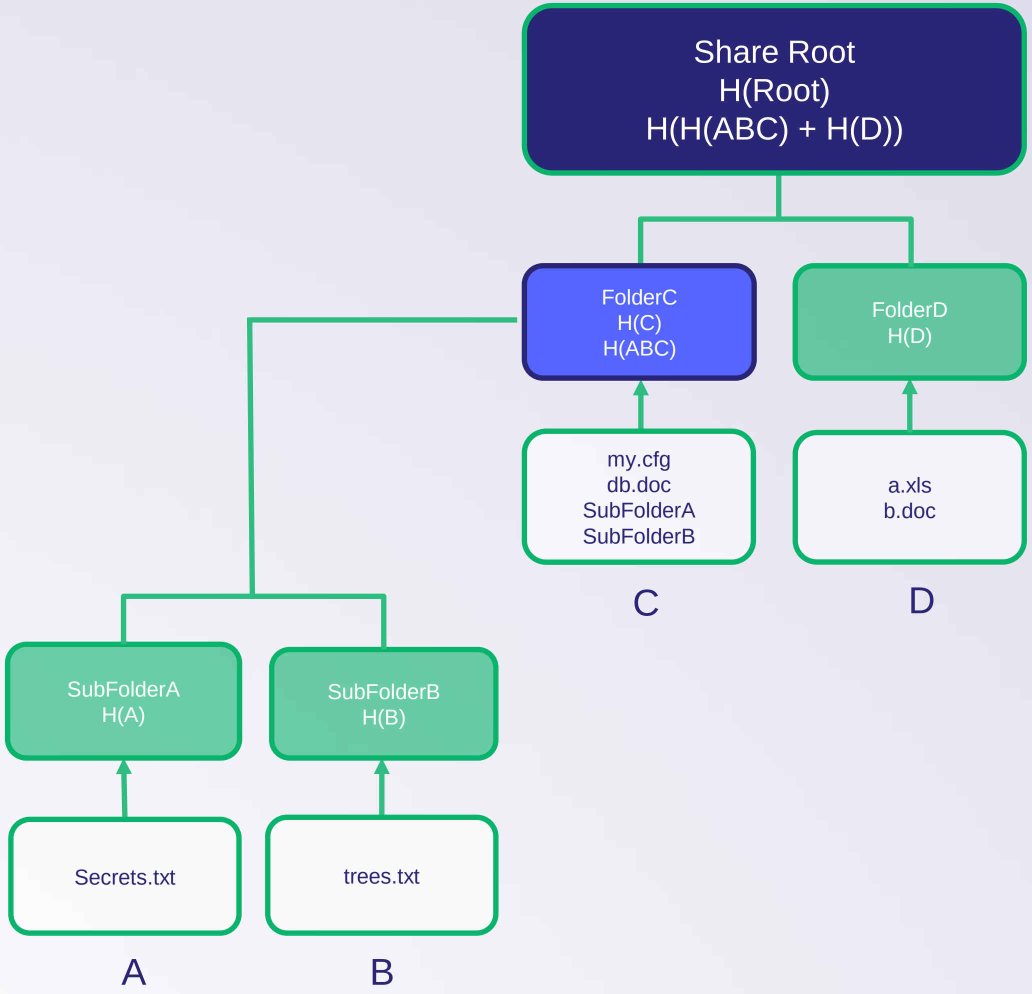


1. HASH 1 - H(A)
2. HASH 2 - H(B)
3. HASH 3 - H(C)
4. HASH 4 - H(ABC)
5. HASH 5 - H(D)
6. HASH 6 - H(Root)
7. HASH 7 - H(ABC) + H(D)

Simple Folder List Hashes

Hierarchical Structure Hashes

If we store all the hashes in a table, we can then perform simple SQL GROUP BY operations like the “Folder Groups”, but this time we can also see groups of folder hierarchies. ☺



Share Root
H(Root)
H(H(ABC) + H(D))

FolderC
H(C)
H(ABC)

FolderD
H(D)

my.cfg
db.doc
SubFolderA
SubFolderB

C

a.xls
b.doc

D

SubFolderA
H(A)

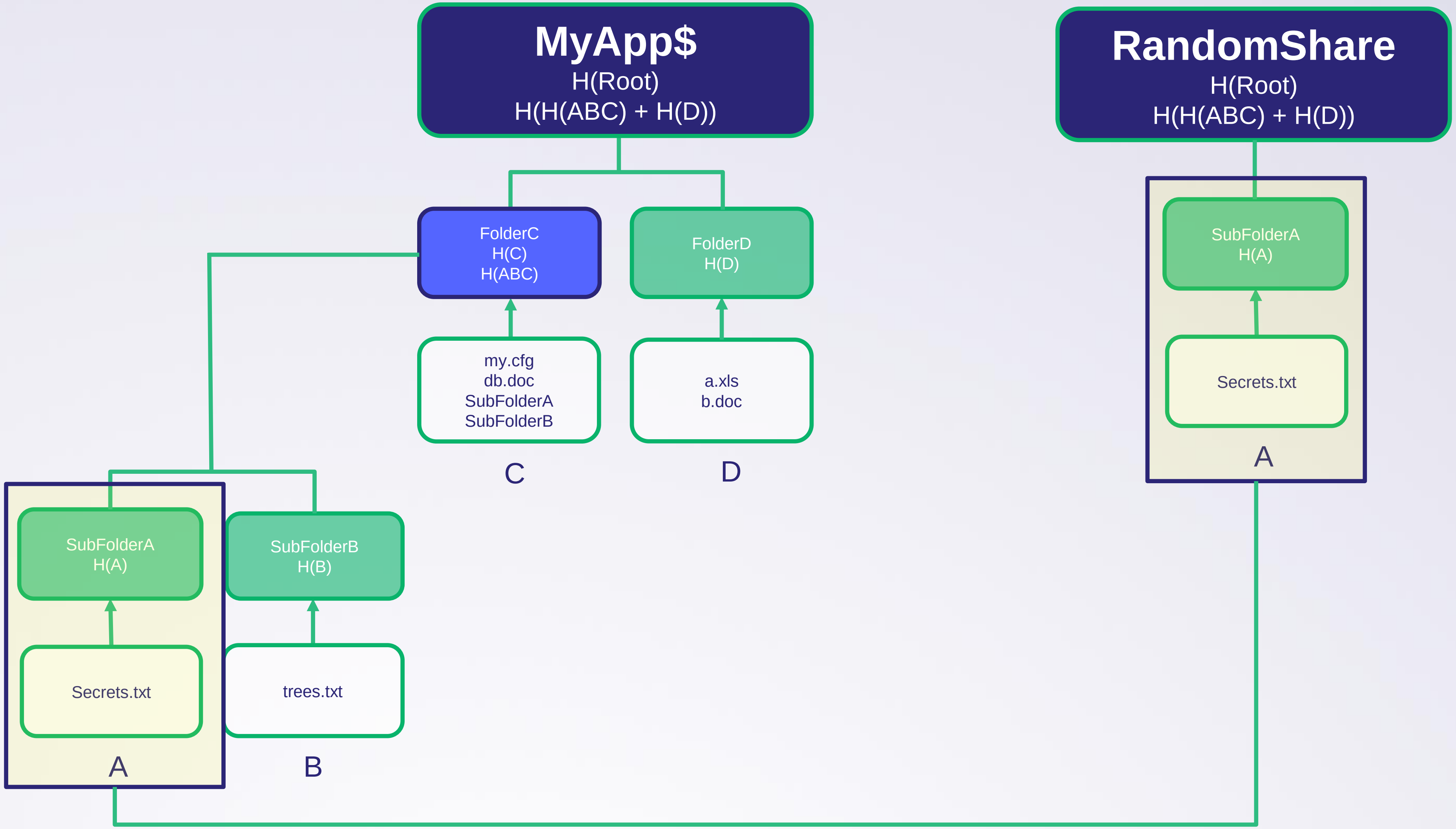
SubFolderB
H(B)

Secrets.txt

A

trees.txt

B



MyApp$
H(Root)
H(H(ABC) + H(D))
RandomShare
H(Root)
H(H(ABC) + H(D))
FolderC
H(C)
H(ABC)
FolderD
H(D)
my.cfg
db.doc
SubFolderA
SubFolderB
C
a.xls
b.doc
D
SubFolderA
H(A)
Secrets.txt
A
SubFolderA
H(A)
SubFolderB
H(B)
Secrets.txt
trees.txt
A
B

Grouping & Similarity

“How can I group similar shares so I can take targeted actions?”

- Group by Share Name
- Group by Folder Group (Dir Hash)
- Group by **Merkle Hash (Nested Dir Hash)**

Our Use Case

Merkle Trees can also be used to expand on the idea of the “Folder group” by hashing the file listings recursively so you can identify nested folder and file listing structure at any folder level.

Pros

- Can surfacing relationships between shares.
- Works great for grouping hierarchies with EXACT structural match.
- Can be used **for hunting for threats and vulnerabilities based on folder, registry memory, database, code structures etc.**

Cons

- Works poorly when the shares DO NOT have the exact same list of files but are used by the same application.
  Which happens a lot.

- Collecting recursive directly listings from shares deeper than 3 levels can take a long time in large environments.

Grouping & Similarity

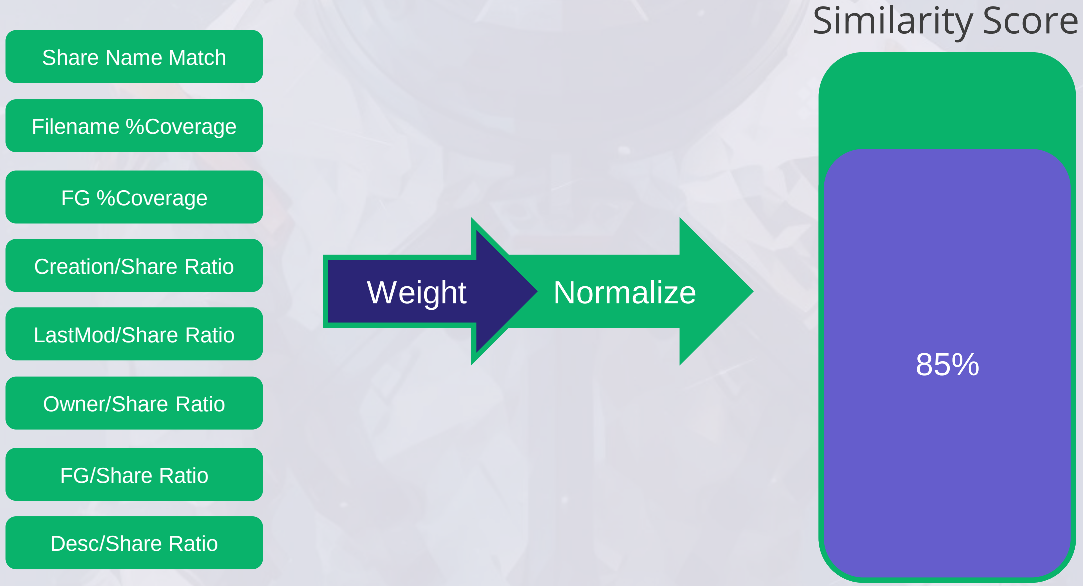
“How can I group similar shares so I can take targeted actions?”

- Group by Share Name
- Group by Folder Group (Dir Hash)
- Group by Merkle Hash (Nested Dir Hash)
- **Calculate weighted similarity score**

Summary

The weighted similarity score used to group shares in PowerHuntShares v2 is derived from multiple data points which are normalized to determine the percentage of similarity.

Logic Abstract



Share Name Match

Filename %Coverage

FG %Coverage

Creation/Share Ratio

LastMod/Share Ratio

Owner/Share Ratio

FG/Share Ratio

Desc/Share Ratio

Weight

Normalize

Similarity Score

85%

Grouping & Similarity

"How can I group similar shares so I can take targeted actions?"

- Group by Share Name
- Group by Folder Group (Dir Hash)
- Group by Merkle Hash (Nested Dir Hash)
- **Calculate weighted similarity score**

Summary

The similarity score in PowerHuntShares v2 is derived from the following meta data:

Pros

- More accurate than the other methods alone.
- More granular metrics provide more information for root cause analysis. Example: Date & owner differences can tell a story.

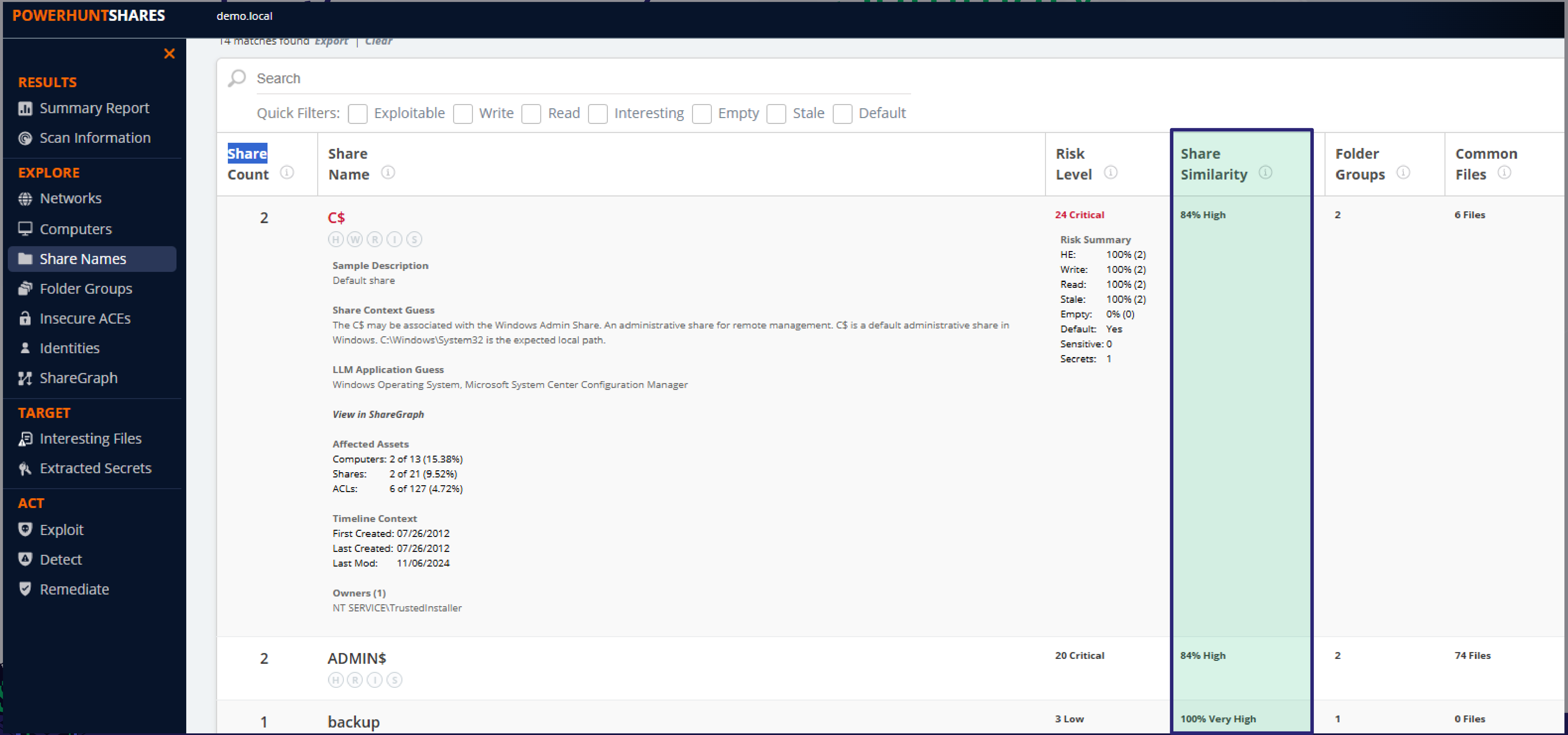
Cons

- Does not take into account fingerprints.
- Does not take into account Merkle Hashes.
- Does not take into account file contents.

Note: The same approach could be applied to almost any file storage medium. For example: AWS s3, Azure blob, or GCP storage.

Grouping & Similarity

Summary



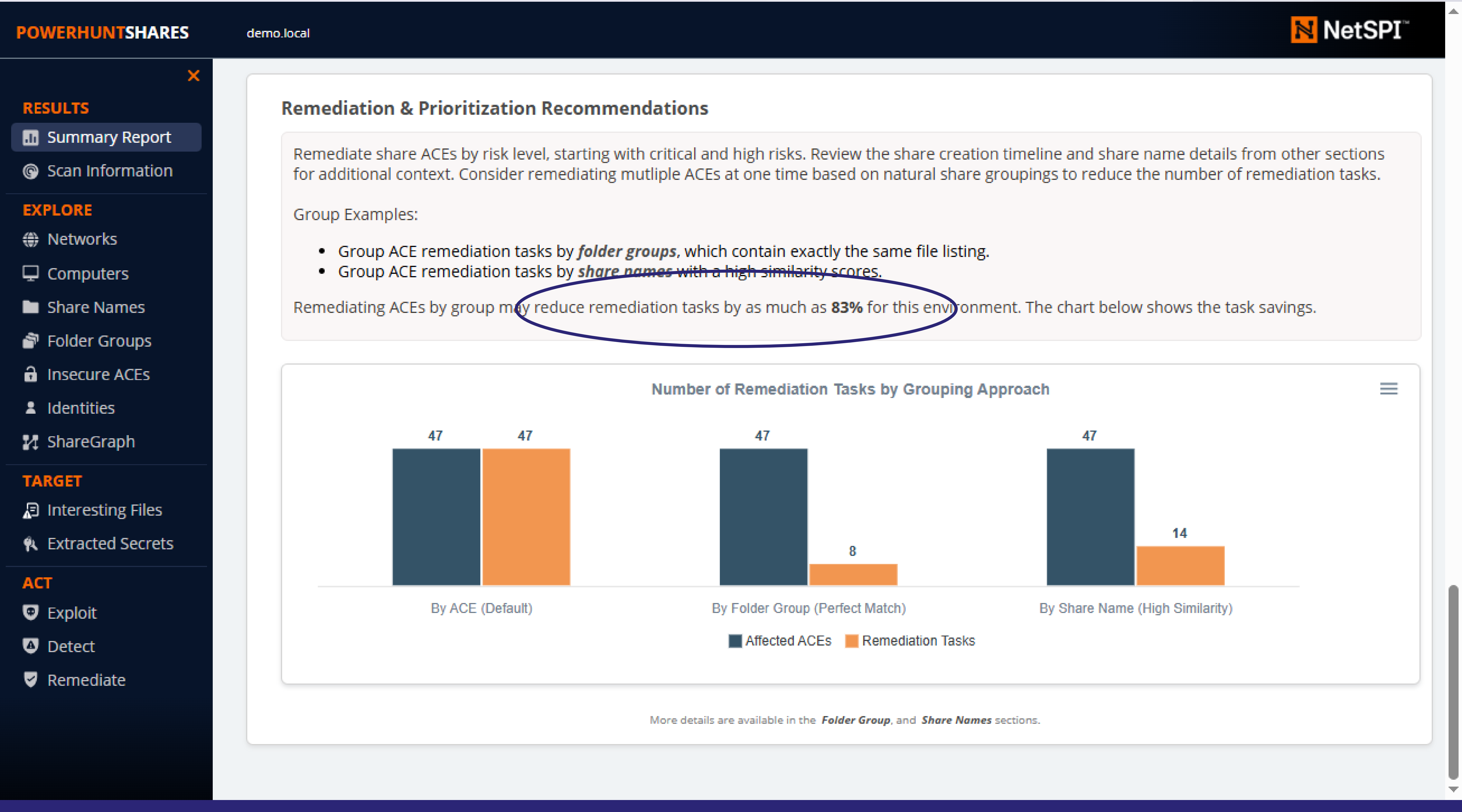
POWERHUNTSHARES demo.local

14 matches found *Export* | *Clear*

Search

Quick Filters: □ Exploitable □ Write □ Read □ Interesting □ Empty □ Stale □ Default

Share Count	Share Name	Risk Level	Share Similarity	Folder Groups	Common Files
2	C$ H W R I S **Sample Description** Default share **Share Context Guess** The C$ may be associated with the Windows Admin Share. An administrative share for remote management. C$ is a default administrative share in Windows. C:\Windows\System32 is the expected local path. **LLM Application Guess** Windows Operating System, Microsoft System Center Configuration Manager ***View in ShareGraph*** **Affected Assets** Computers: 2 of 13 (15.38%) Shares: 2 of 21 (9.52%) ACLs: 6 of 127 (4.72%) **Timeline Context** First Created: 07/26/2012 Last Created: 07/26/2012 Last Mod: 11/06/2024 **Owners (1)** NT SERVICE\TrustedInstaller	24 Critical **Risk Summary** HE: 100% (2) Write: 100% (2) Read: 100% (2) Stale: 100% (2) Empty: 0% (0) Default: Yes Sensitive: 0 Secrets: 1	84% High	2	6 Files
2	ADMIN$ H R I S	20 Critical	84% High	2	74 Files
1	backup	3 Low	100% Very High	1	0 Files



Remediation & Prioritization Recommendations

Remediate share ACEs by risk level, starting with critical and high risks. Review the share creation timeline and share name details from other sections for additional context. Consider remediating mutliple ACEs at one time based on natural share groupings to reduce the number of remediation tasks.

Group Examples:

- Group ACE remediation tasks by ***folder groups***, which contain exactly the same file listing.
- Group ACE remediation tasks by ***share names*** with a high similarity scores.

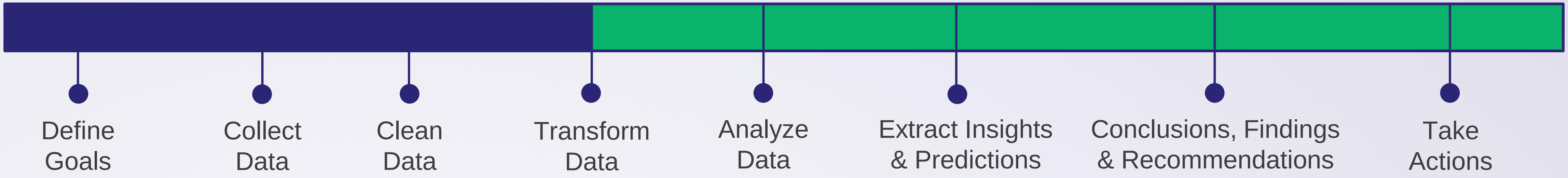
Remediating ACEs by group may reduce remediation tasks by as much as **83%** for this environment. The chart below shows the task savings.

Number of Remediation Tasks by Grouping Approach

47
47
By ACE (Default)
47
8
By Folder Group (Perfect Match)
47
14
By Share Name (High Similarity)
Affected ACEs
Remediation Tasks

More details are available in the ***Folder Group***, and ***Share Names*** sections.

PowerHuntShares Process



Define Goals

Collect Data

Clean Data

Transform Data

Analyze Data

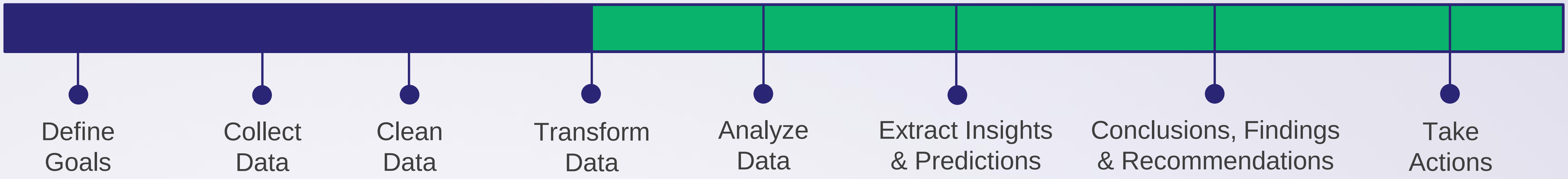
Extract Insights & Predictions

Conclusions, Findings & Recommendations

Take Actions

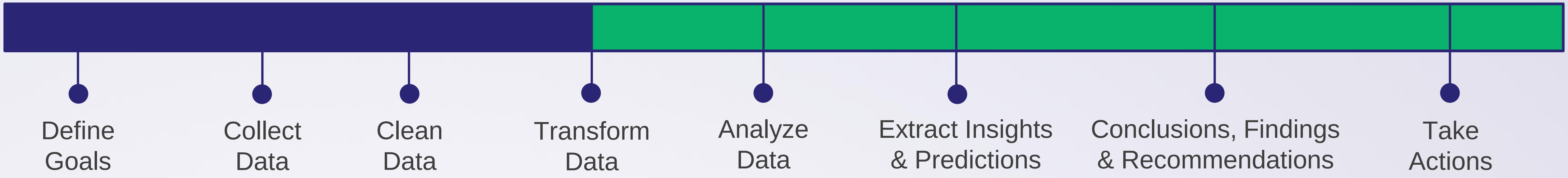
Reduce Risk
with Fewer Actions

PowerHuntShares Process



Reduce Risk
with Fewer Actions

PowerHuntShares Process



Define Goals

Collect Data

Clean Data

Transform Data

Analyze Data

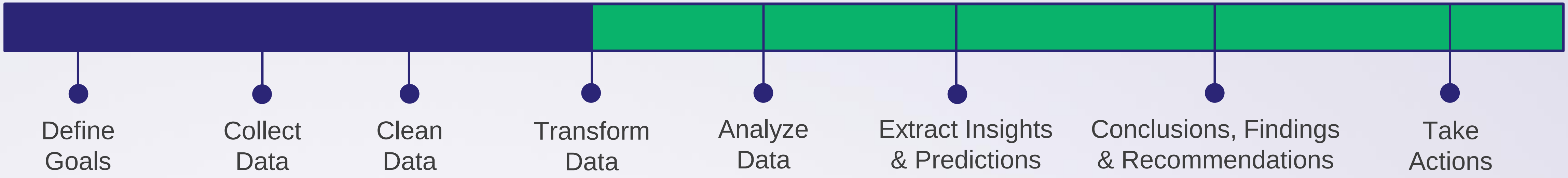
Extract Insights & Predictions

Conclusions, Findings & Recommendations

Take Actions

Reduce Risk
with Fewer Actions

PowerHuntShares Process



Define Goals

Collect Data

Clean Data

Transform Data

Analyze Data

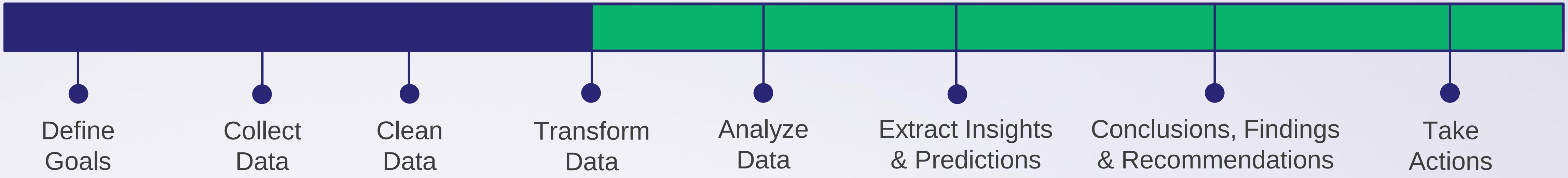
Extract Insights & Predictions

Conclusions, Findings & Recommendations

Take Actions

Reduce Risk
with Fewer Actions

PowerHuntShares Process



Define Goals

Collect Data

Clean Data

Transform Data

Analyze Data

Extract Insights & Predictions

Conclusions, Findings & Recommendations

Take Actions

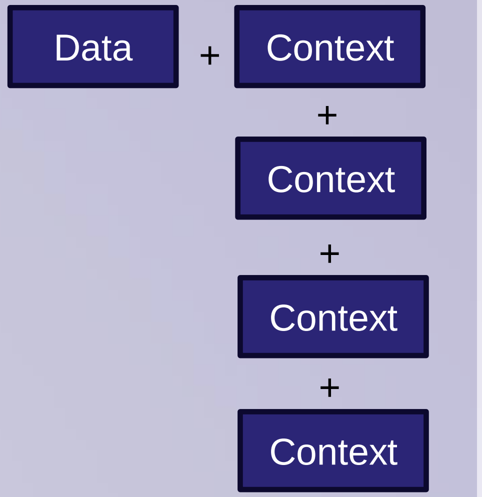
Transform Data

Static Data Labeling

- Highly Exploitable, Interesting Files, Secrets Extraction, Stale, Empty

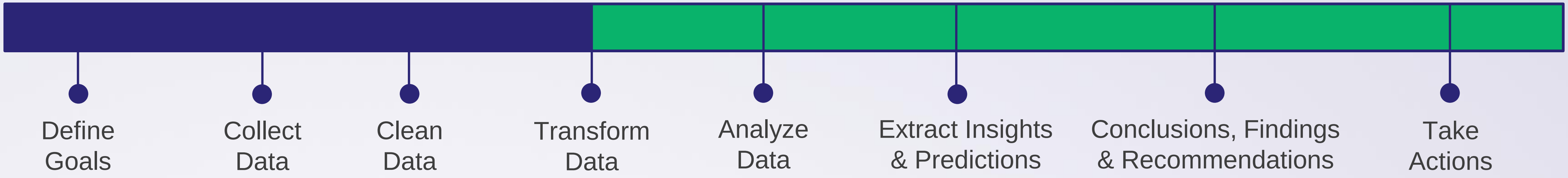
Dynamic Data Enrichment

- Fingerprinting, Risk Scoring, Peer Comparison, **Grouping & Similarity Scoring**



Data + Context + Context + Context + Context

PowerHuntShares Process



Transform Data

Static Data Labeling

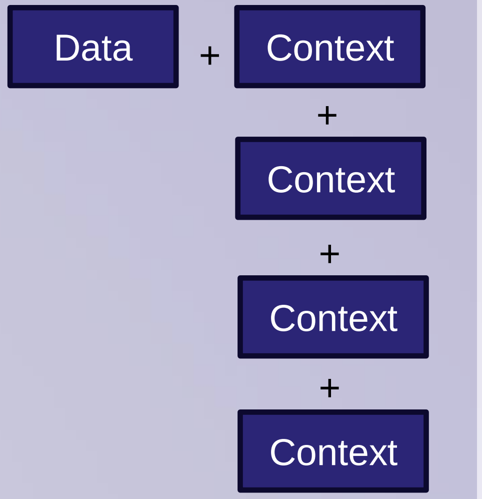
- Highly Exploitable, Interesting Files, Secrets Extraction, Stale, Empty

Dynamic Data Enrichment

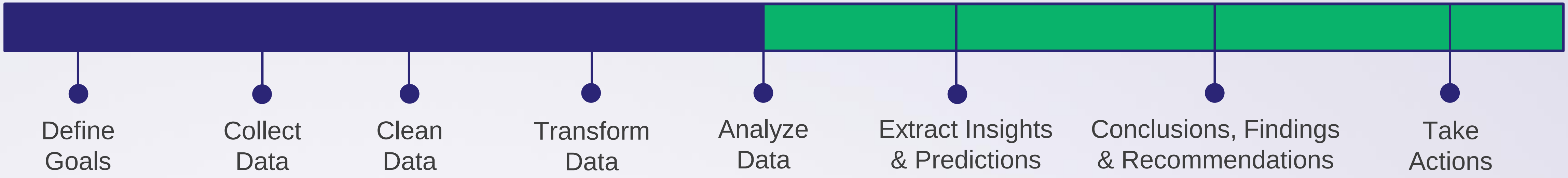
- Fingerprinting, Risk Scoring, Peer Comparison, Grouping & Similarity Scoring

Convert to JSON/CSV

Convert to graph dataset

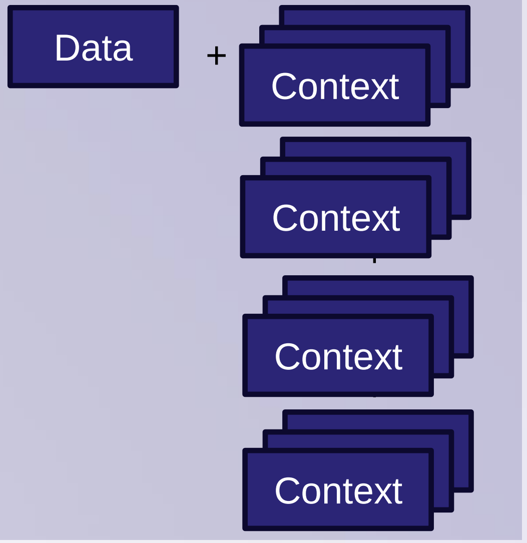


PowerHuntShares Process



Analyze Data

- Networks, Computers, Share Names, Folder Groups, Aces, Identities
- ShareGraph
- Share Creation Timeline
- Prioritized Recommendations



Exploring Data
Chart & Graphs

“How can I explore and visualize my data to gain insights and tell stories?”

Exploring Data
Chart & Graphs

- **Simple Charts with ApexCharts.js**

ApexCharts.js

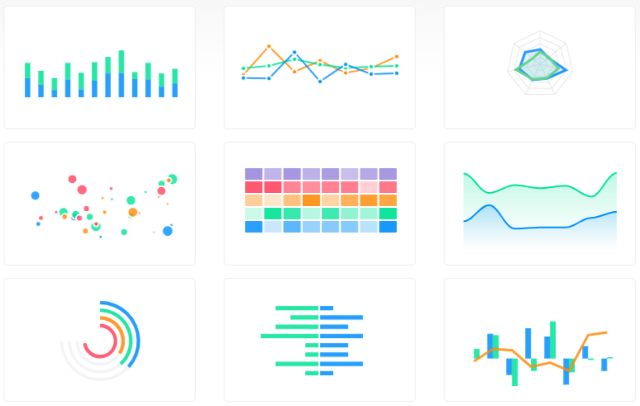
“Can you help me visualize this data in a chart?”

Quick Story

1. Asked ChatGPT for the top 5 open sources/free JavaScript chart libraries with specific features.

2. Provided it a use case and asked it to produce a simple web application with the **ApexCharts.js**.

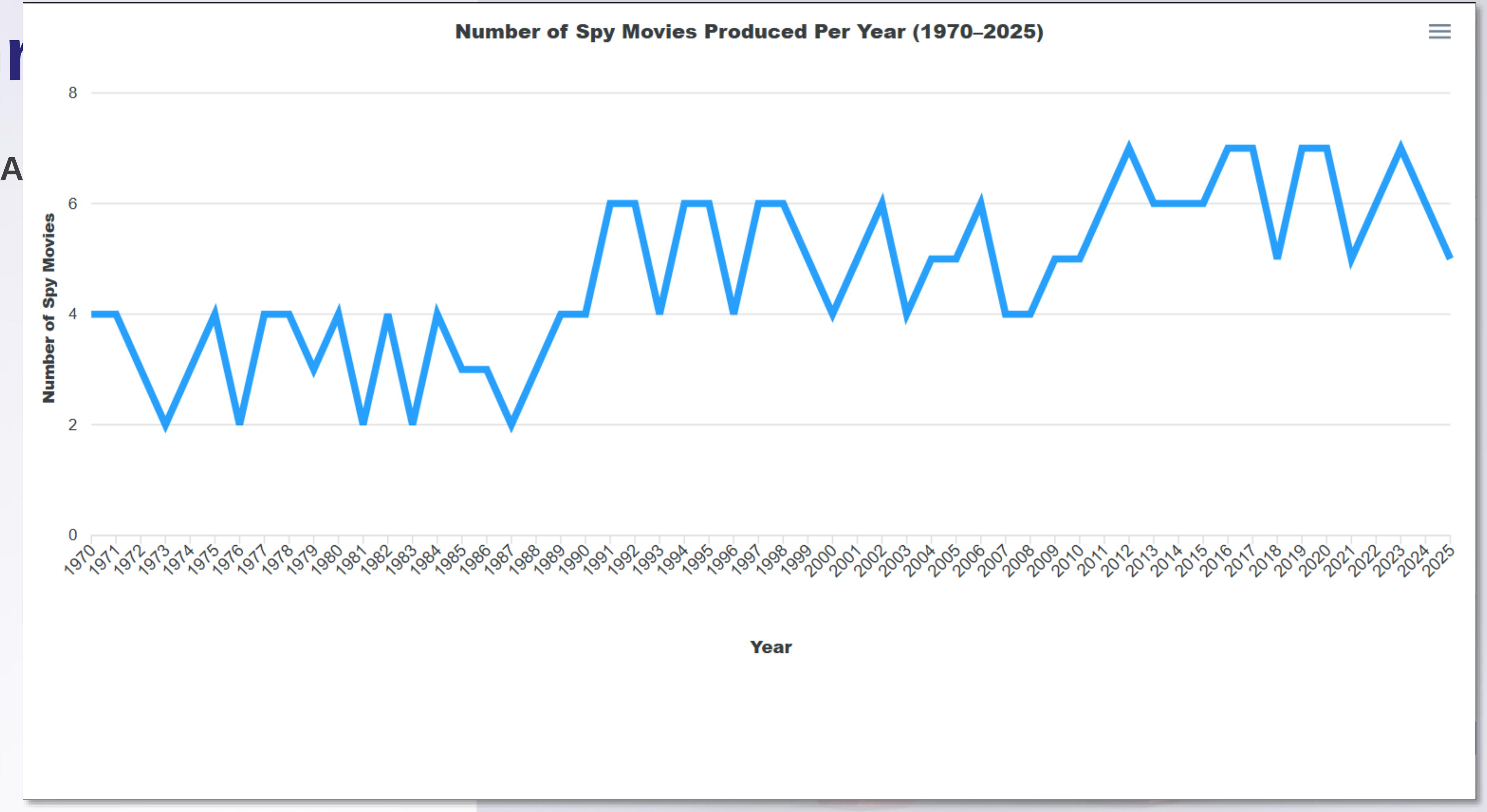
3. It’s be a love affair ever since.



www.apexcharts.com

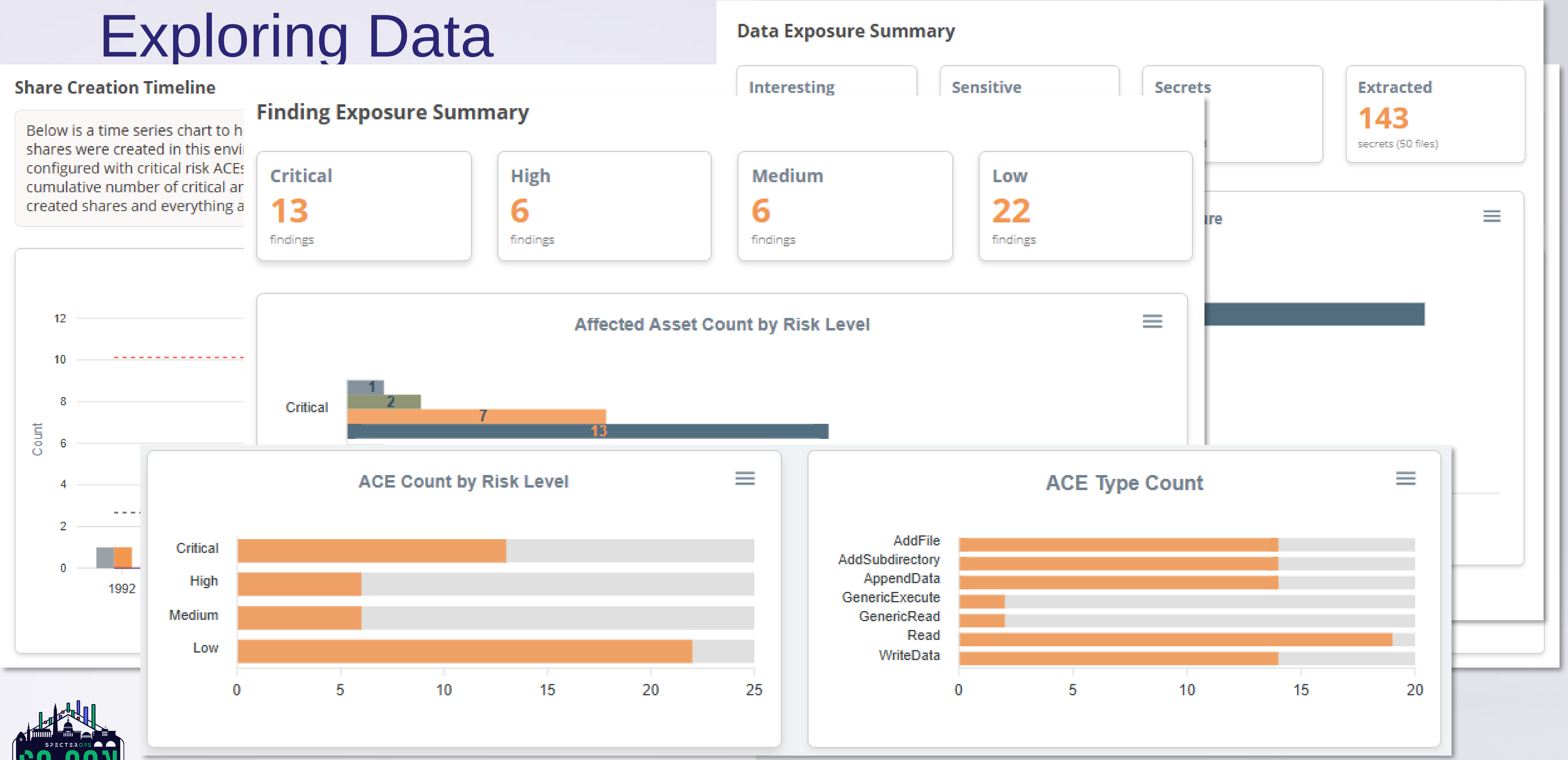
Exploring Data
Chart & Gr

- Simple Charts with A



Number of Spy Movies Produced Per Year (1970–2025)

Exploring Data



Share Creation Timeline

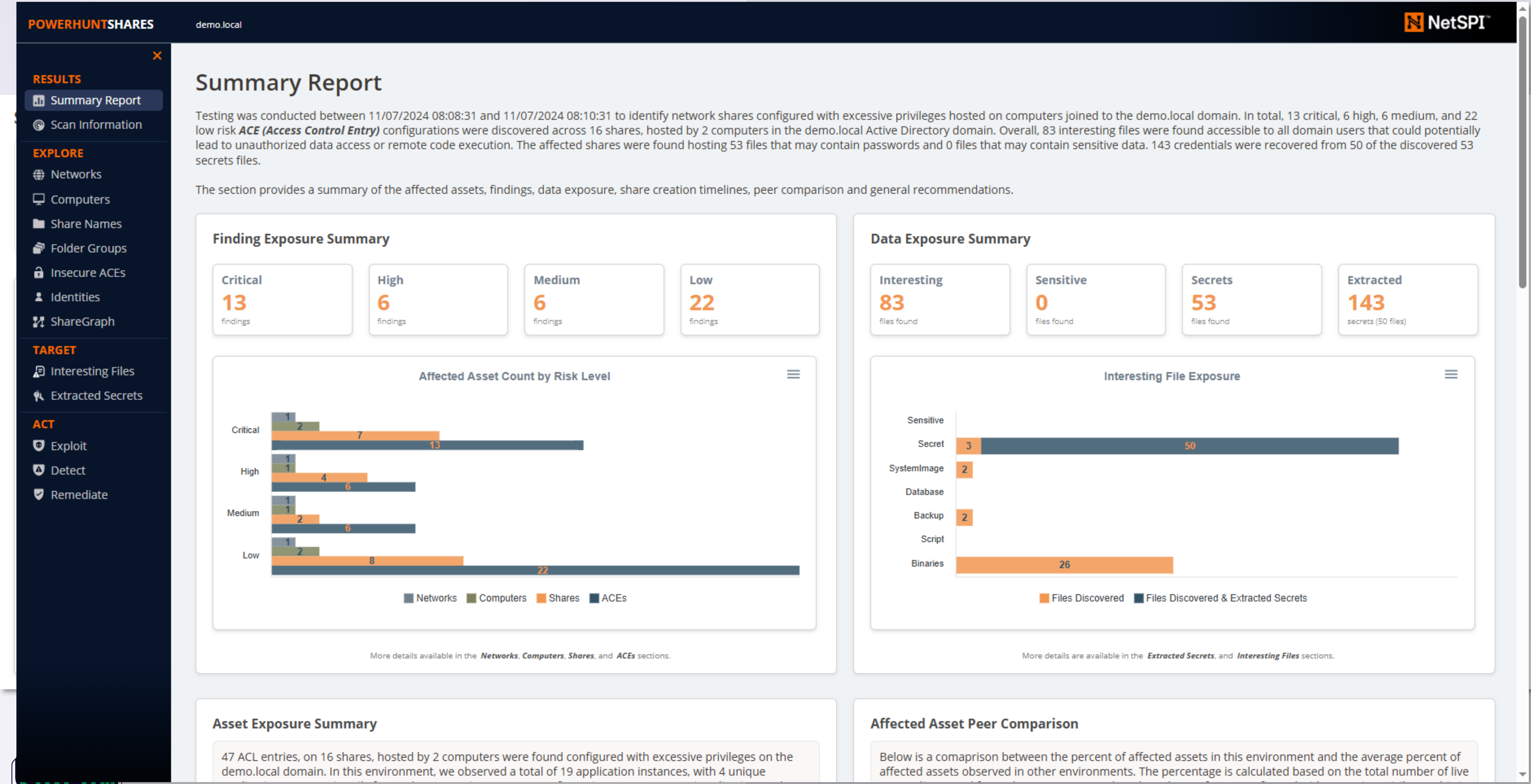
Below is a time series chart to h
shares were created in this envi
configured with critical risk ACEs
cumulative number of critical ar
created shares and everything a

Data Exposure Summary

Interesting	Sensitive	Secrets	Extracted
			143 secrets (50 files)

Finding Exposure Summary

Critical	High	Medium	Low
13 findings	6 findings	6 findings	22 findings



Summary Report

Testing was conducted between 11/07/2024 08:08:31 and 11/07/2024 08:10:31 to identify network shares configured with excessive privileges hosted on computers joined to the demo.local domain. In total, 13 critical, 6 high, 6 medium, and 22 low risk ***ACE (Access Control Entry)*** configurations were discovered across 16 shares, hosted by 2 computers in the demo.local Active Directory domain. Overall, 83 interesting files were found accessible to all domain users that could potentially lead to unauthorized data access or remote code execution. The affected shares were found hosting 53 files that may contain passwords and 0 files that may contain sensitive data. 143 credentials were recovered from 50 of the discovered 53 secrets files.

The section provides a summary of the affected assets, findings, data exposure, share creation timelines, peer comparison and general recommendations.

Finding Exposure Summary

Critical	High	Medium	Low
13 findings	6 findings	6 findings	22 findings

More details available in the ***Networks***, ***Computers***, ***Shares***, and ***ACEs*** sections.

Data Exposure Summary

Interesting	Sensitive	Secrets	Extracted
83 files found	0 files found	53 files found	143 secrets (50 files)

More details are available in the ***Extracted Secrets***, and ***Interesting Files*** sections.

Asset Exposure Summary

47 ACL entries, on 16 shares, hosted by 2 computers were found configured with excessive privileges on the demo.local domain. In this environment, we observed a total of 19 application instances, with 4 unique

Affected Asset Peer Comparison

Below is a comaprison between the percent of affected assets in this environment and the average percent of affected assets observed in other environments. The percentage is calculated based on the total number of live

Exploring Data
Chart & Graphs

- Simple Charts with ApexCharts.js
- **Exploring Data with Graphs: Cytoscape.js**

CytoScape.js

“Can you help me visualize these share relationships?”

Similar Story

1. Asked ChatGPT for the top 5 open sources/free JavaScript graphing libraries with specific features
2. Provided it a use case and asked it to produce a simple web application with the graph using **Cytoscape.js**.
3. It’s be a love affair ever since.

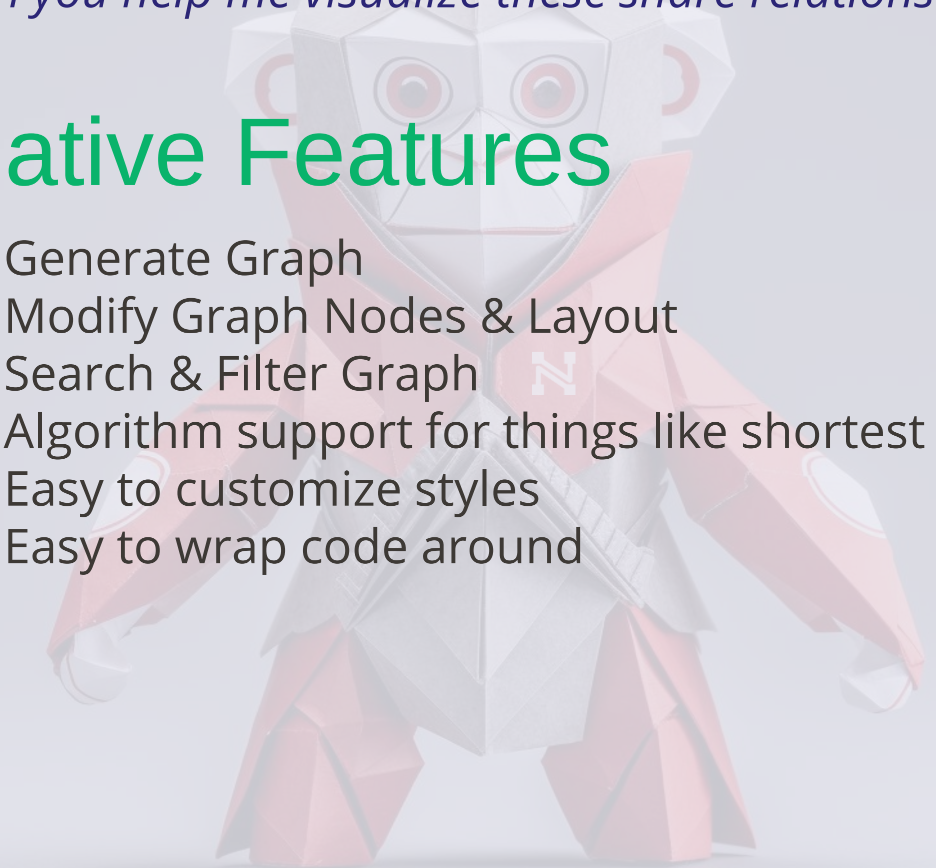
js.cytoscape.org

Exploring Data **Chart & Graphs**

- Simple Charts with ApexCharts.js
- **Exploring Data with Graphs: Cytoscape.js**

CytoScape.js

“Can you help me visualize these share relationships?”



Native Features

- Generate Graph
- Modify Graph Nodes & Layout
- Search & Filter Graph
- Algorithm support for things like shortest Path
- Easy to customize styles
- Easy to wrap code around

Exploring Data **Chart & Graphs**

- Simple Charts with ApexCharts.js
- **Exploring Data with Graphs: Cytoscape.js**

CytoScape.js Prompt Example

Please create an html graph using Cytoscape.js that includes:

Layout Options
1. Add buttons to change the layout to breadthfirst and the top five other layouts like grid.
2. Add buttons to show Pageranked most influential nodes in bright orange.
3. When Pageranked button is clicked resize nodes based on pagerank.
3. Add buttons to show Betweenness Centrality nodes in bright tan and create a px border in black.

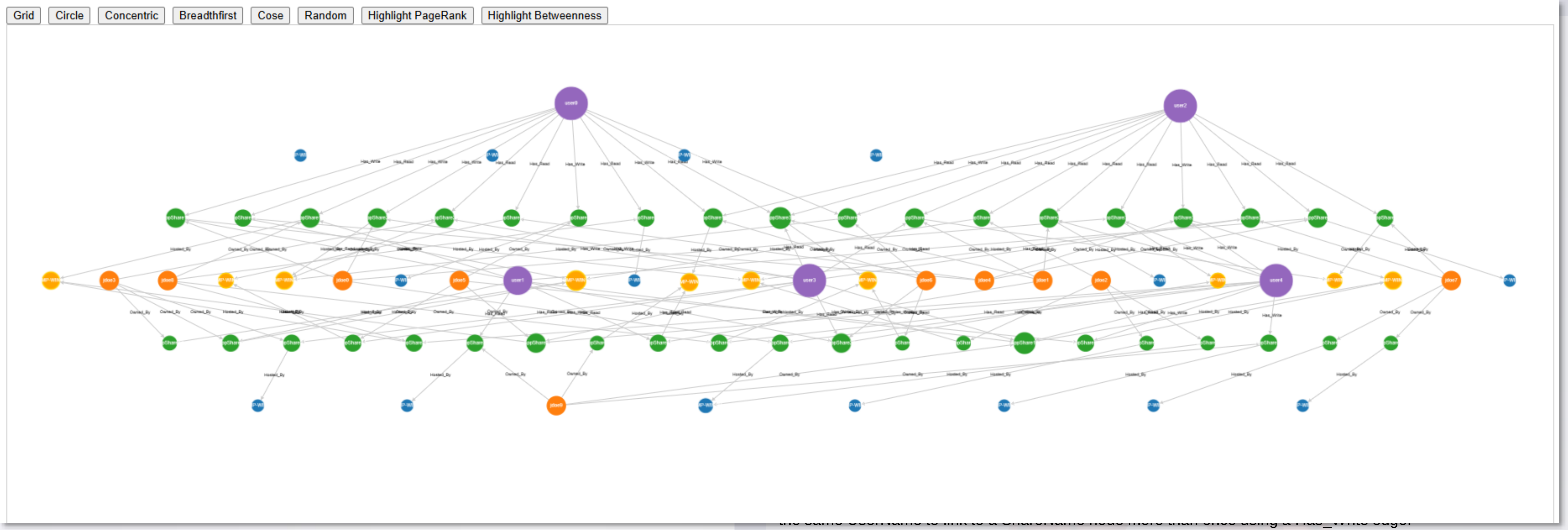
Nodes with the details below:
1. Four node types: ComputerName, ShareName, Owner, and User nodes.
2. Generate a list of 10 Owner nodes that look like user names.
3. Generate a list of 25 ComputerName nodes that look like they would be part of a common entrprise network.
4. Generate a list of 40 ShareName nodes that look like SMB shares used by applications
5. Generate a list of 5 UserName nodes that see like simple user names.
6. Ensure all nodes are large enough to be read.
7. Ensure all nodes are the same shape.
8. Ensure all nodes types have a different color.

Edges with the details below:
1. Four edge types: Owned_By, Hosted_By, Has_Write, and Has_Read.
2. Generate Owned_By edges between Owner nodes and ShareName nodes. Ensure each ShareName has one owner.
3. Generate Hosted_By edges between ShareName nodes and ComputerName nodes. Assign those Hosted_By edges randomly, but ensure at least 80% of ComputerName nodes have at least one ShareName node connected.
4. Generate 20 Has_Write edge between randomaly selected UserName nodes and ShareNames. Do not allow the same UserName to link to a ShareName node more than once using a Has_Write edge.
5. Generate 30 Has_Read edge between randomaly selected UserName nodes and ShareNames. Do not allow the same UserName to link to a ShareName node more than once using a Has_Read edge.
6. Ensure all nodes are large enough to be read.
7. Ensure all nodes are the same shape.

Please dont forget to add the nodes and edges.

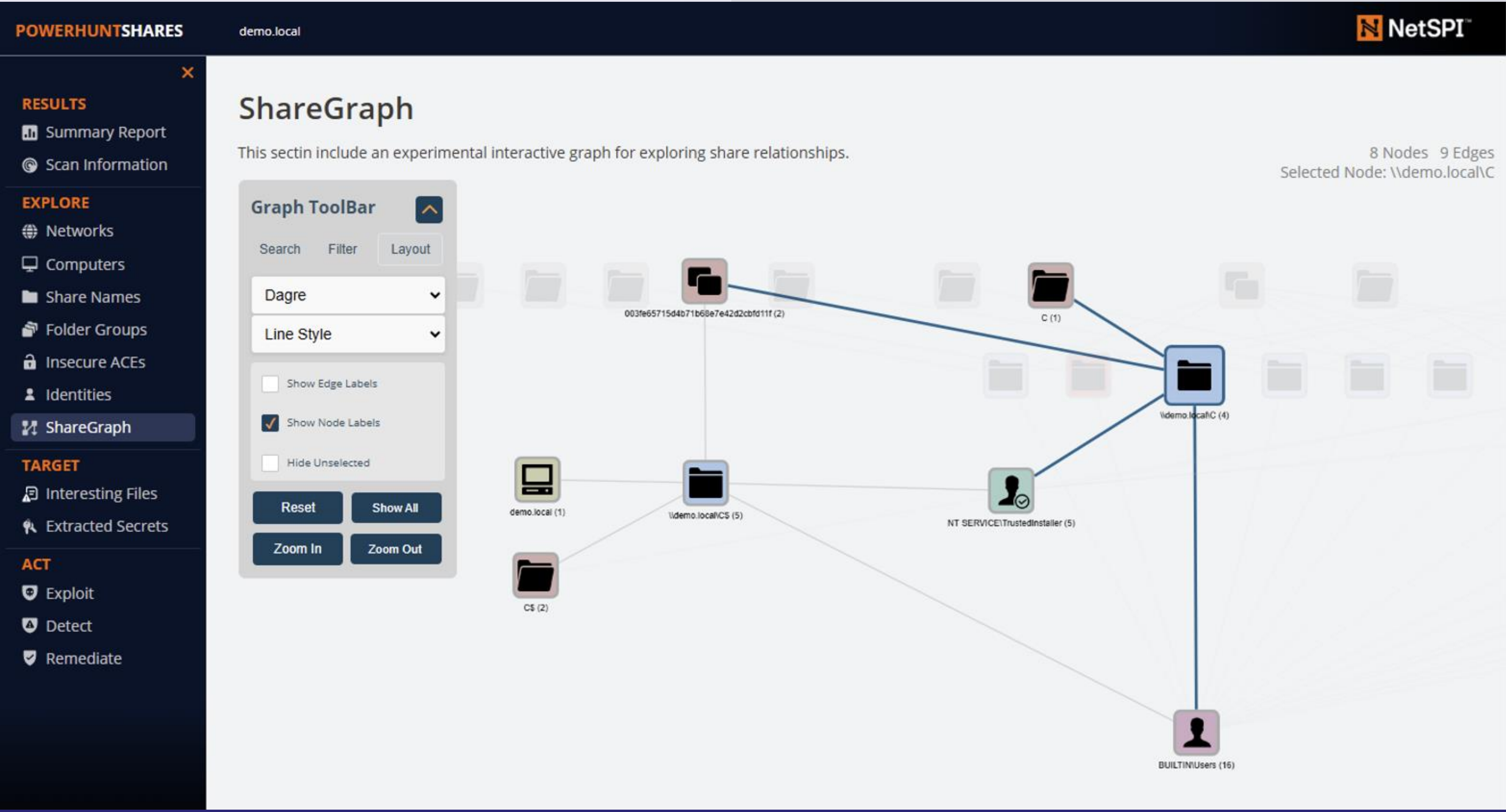
Exploring Data **Chart & Graphs**

CytoScape.js Prompt Example



the same UserName to link to a ShareName node more than once using a Has_Write edge.
5. Generate 30 Has_Read edge between randomaly selected UserName nodes and ShareNames. Do not allow the same UserName to link to a ShareName node more than once using a Has_Read edge.
6. Ensure all nodes are large enough to be read.
7. Ensure all nodes are the same shape.

Please dont forget to add the nodes and edges.



ShareGraph

This sectin include an experimental interactive graph for exploring share relationships.

8 Nodes 9 Edges
Selected Node: \\demo.local\C

Graph ToolBar

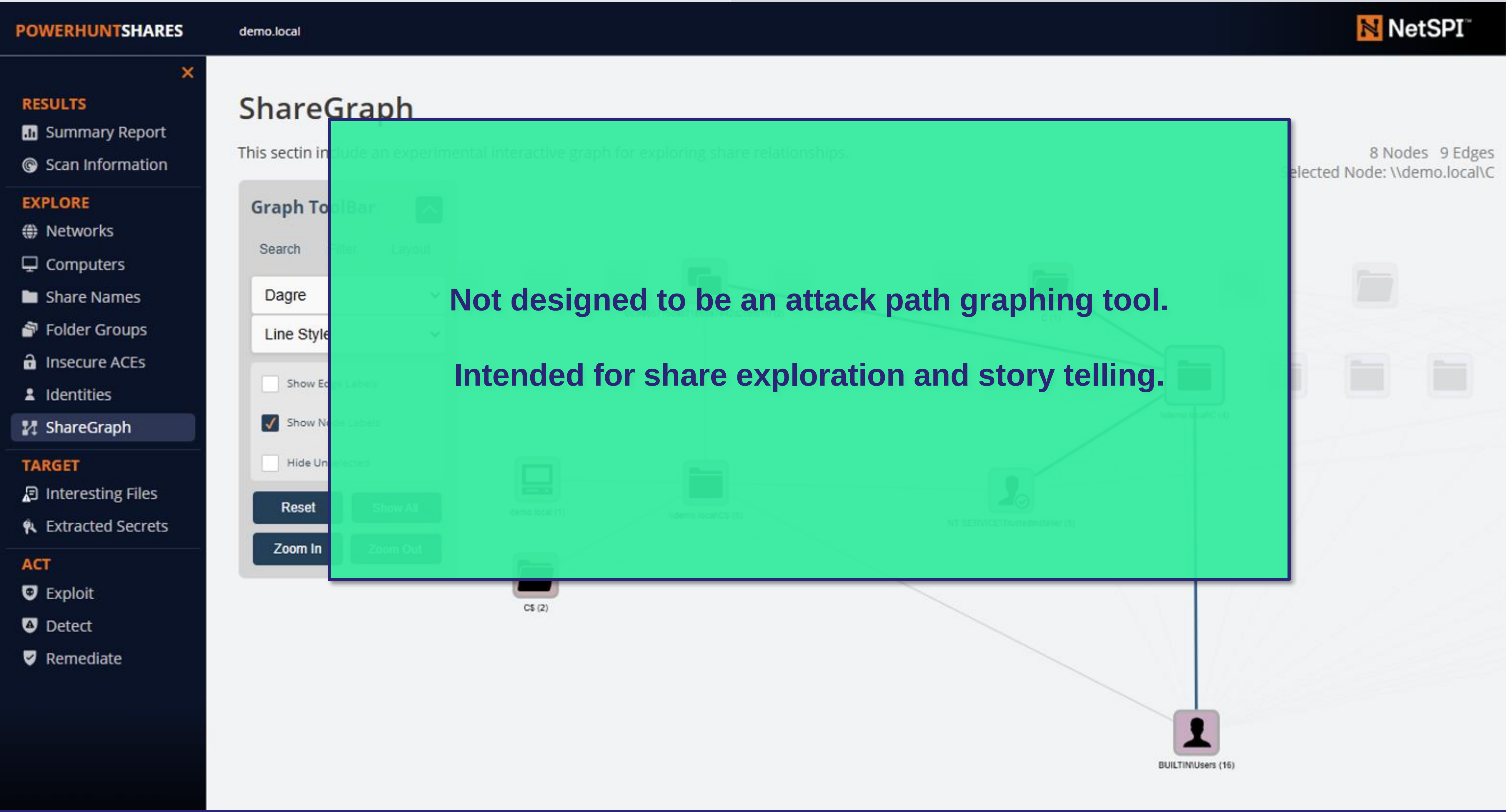
Dagre

Line Style

□ Show Edge Labels

☑ Show Node Labels

□ Hide Unselected



ShareGraph

This sectin include an experimental interactive graph for exploring share relationships.

8 Nodes 9 Edges
elected Node: \\demo.local\C

☐ Show Edge Labels
☑ Show Node Labels
☐ Hide Unselected

Not designed to be an attack path graphing tool.

Intended for share exploration and story telling.

Finding Nodes
That Matter

Finding Nodes that Matter

"Are there things I'm not thinking of and what other tools are available?"

Explored Neo4j Graph Data Science (GDS) library
https://neo4j.com/docs/graph-data-science/current/algorithms/

30 algorithms reviewed
I was looking for problems for these solutions ;)

Algorithms I liked in Neo4j

- Page Rank - What nodes have the most influence?
- Betweenness Centrality - What nodes act as a bridge?
- Louvain - What are natural node clusters?

All of the algorithms I liked were also available in Cytoscape.js ☺

Finding Nodes **That Matter**

- **Finding Nodes that Matter: PageRank**

  Supported by **Cytoscape.js** and **Neo4j**

Page Rank

“What are the most Influential Nodes?”

Why Should I Care?

- **Offense** can identify which nodes will provide access to resources, routes, etc.
- **Defense** can do the same and add preventative, detective and corrective controls to make them more resilient to attack

Simple Example

When experimenting with simple Active Directory environment graphs, Page Rank could be used to identify the most influential nodes… guess which node do you think was most influential?

Finding Nodes **That Matter**

- **Finding Nodes that Matter: PageRank**

  Supported by **Cytoscape.js** and **Neo4j**

Page Rank

“What are the most Influential Nodes?”

Why Should I Care?

- **Offense** can identify which nodes will provide access to resources, routes, etc.
- **Defense** can do the same and add preventative, detective and corrective controls to make them more resilient to attack

Simple Example

When experimenting with simple Active Directory environment graphs, Page Rank could be used to identify the most influential nodes… guess which node do you think was most influential?

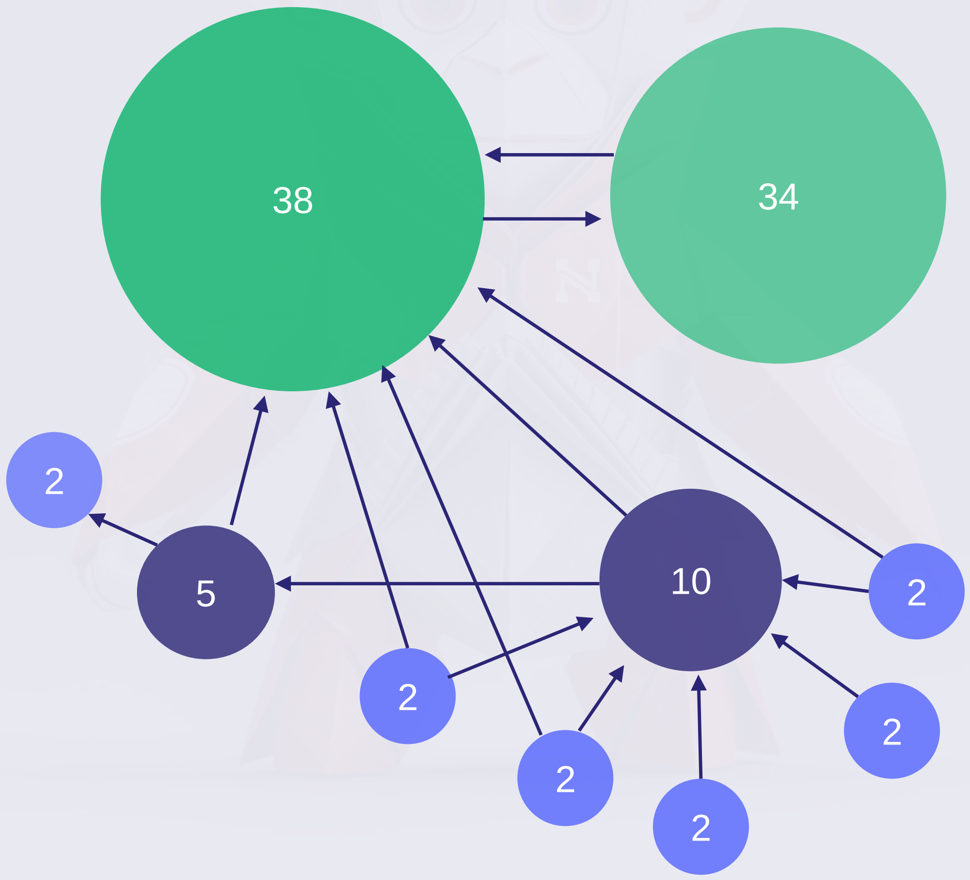
Domain

Finding Nodes **That Matter**

- **Finding Nodes that Matter: PageRank**
  Supported by **Cytoscape.js** and **Neo4j**

Page Rank

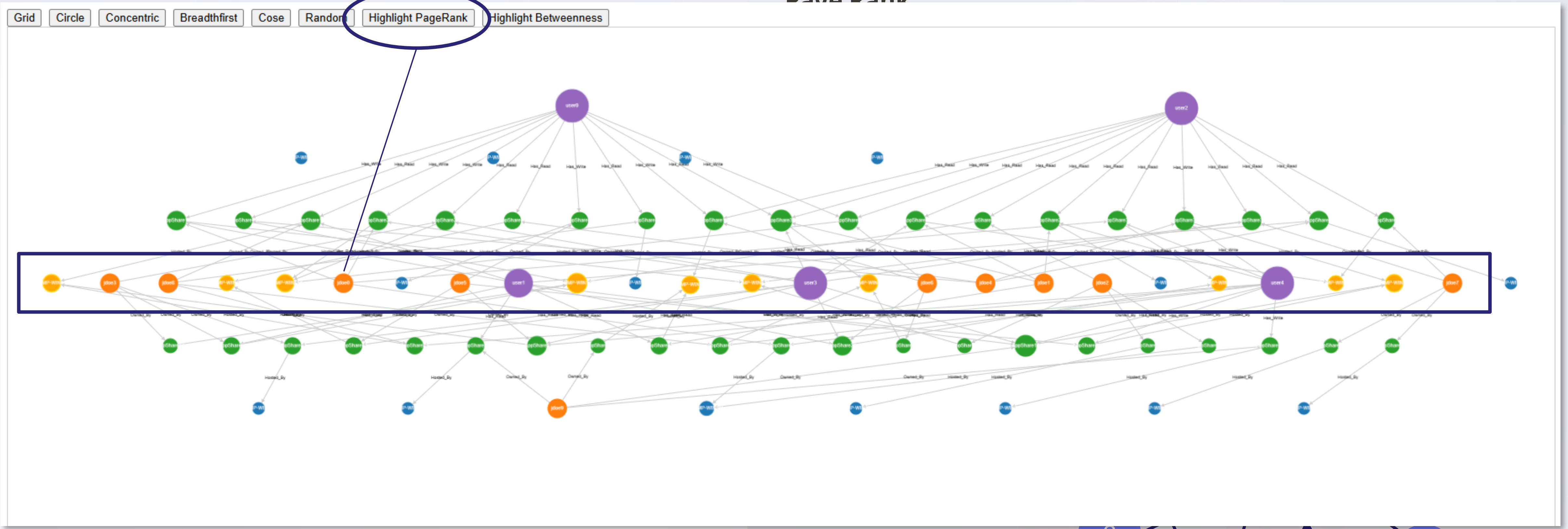
"What are the most Influential Nodes?"

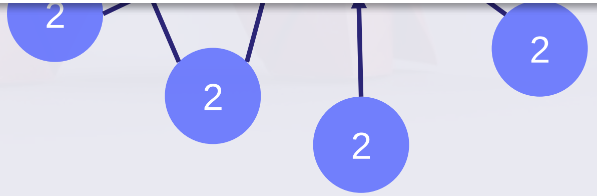


Finding Nodes **That Matter**

Page Rank

Page Rank





Finding Nodes **That Matter**

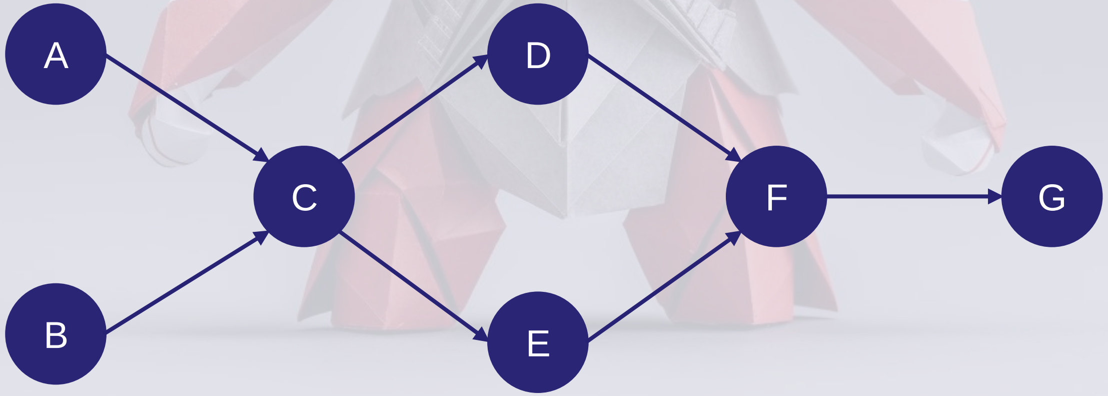
- Finding Nodes that Matter: PageRank
- **Finding Nodes that Matter: Betweenness**
  Supported by **Cytoscape.js** and **Neo4j**

Betweenness Centrality

*"Which nodes lie on the **shortest paths** between other nodes?" aka they act like bridges between communities of nodes.*

Why Should I Care?

- We may be able to determine which nodes are providing attackers with the greatest mobility.
- Prioritizing their remediation may help reduce risk or the speed at which attackers can move.



Finding Nodes **That Matter**

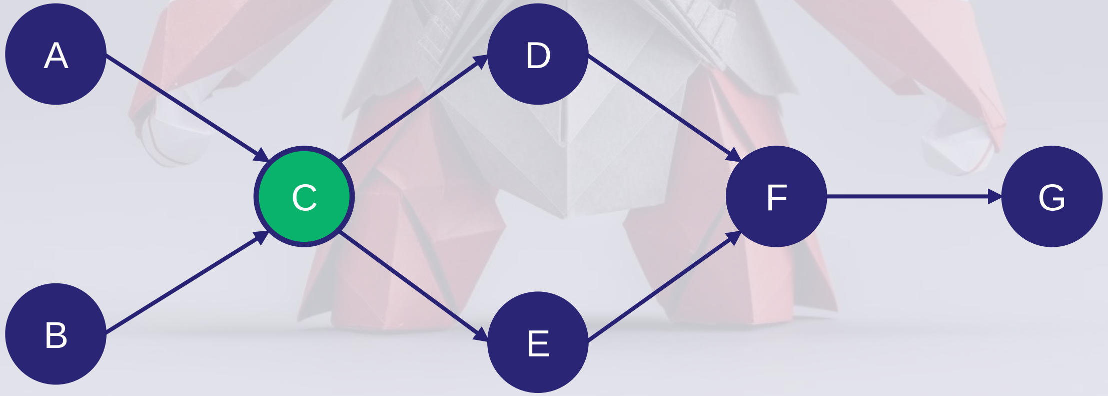
- Finding Nodes that Matter: PageRank
- **Finding Nodes that Matter: Betweenness**
  Supported by **Cytoscape.js** and **Neo4j**

Betweenness Centrality

*"Which nodes lie on the **shortest paths** between other nodes?" aka they act like bridges between communities of nodes.*

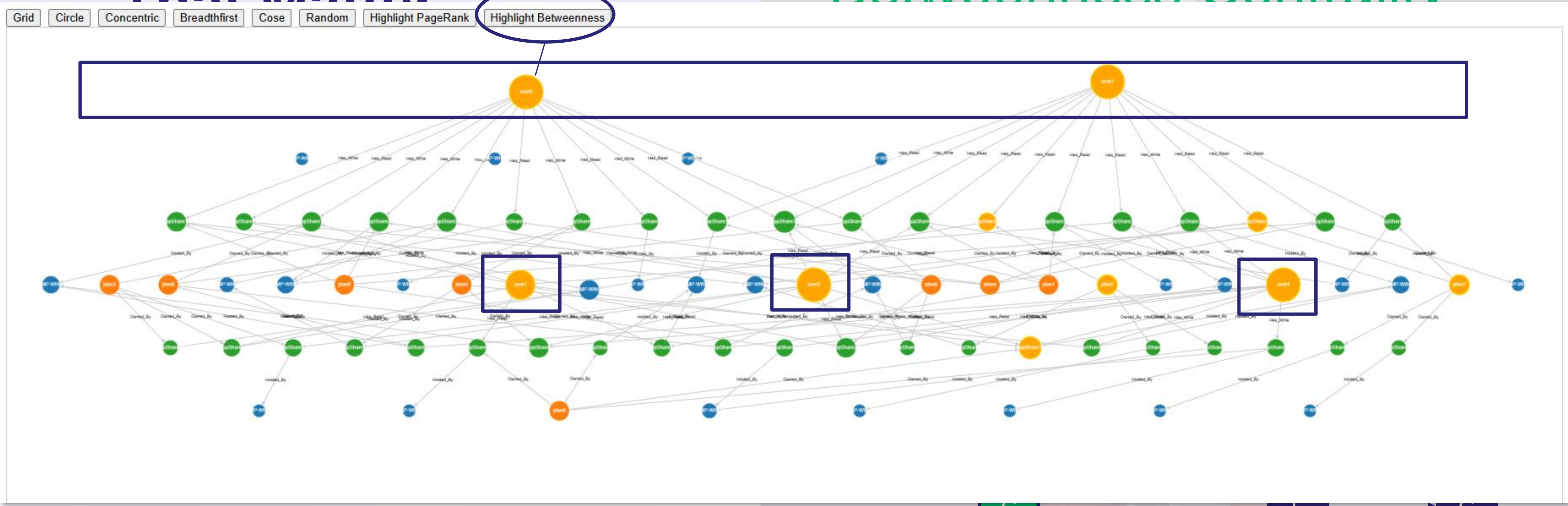
Why Should I Care?

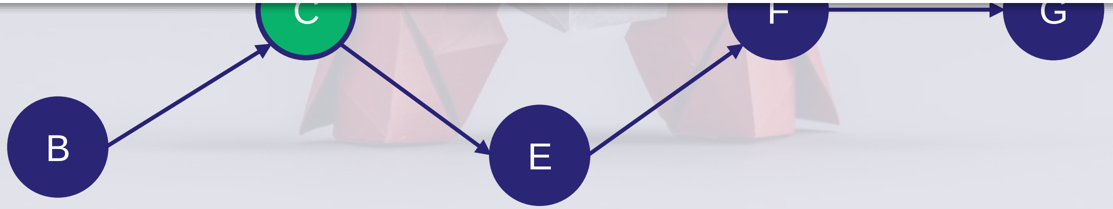
- We may be able to determine which nodes are providing attackers with the greatest mobility.
- Prioritizing their remediation may help reduce risk or the speed at which attackers can move.



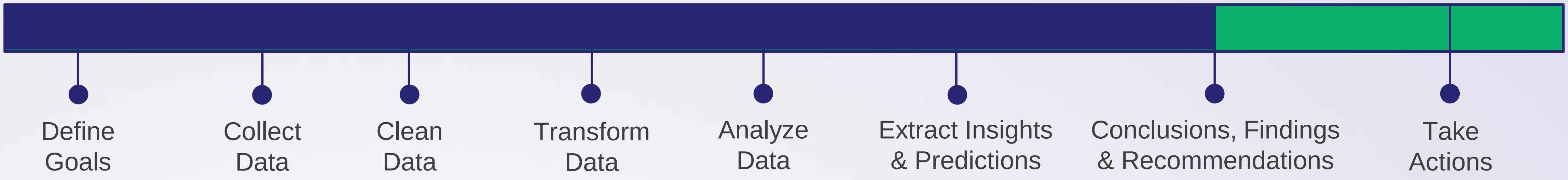
Finding Nodes That Matter

Betweenness Centrality





PowerHuntShares Process



Define Goals

Collect Data

Clean Data

Transform Data

Analyze Data

Extract Insights & Predictions

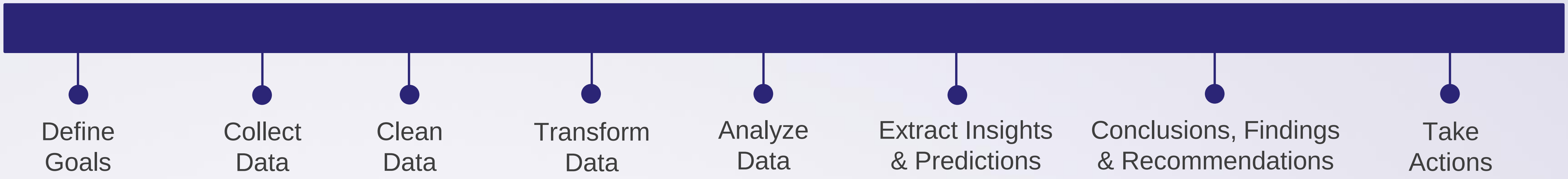
Conclusions, Findings & Recommendations

Take Actions

Conclusions, Findings, & Recommendations

- How many shares are vulnerable?
- What shares are most vulnerable?
- When were the shares created?
- What application will be affected if we fix this?
- How can I remediate shares efficiently?
- How they should and do compare to peers?

PowerHuntShares Process



Define Goals

Collect Data

Clean Data

Transform Data

Analyze Data

Extract Insights & Predictions

Conclusions, Findings & Recommendations

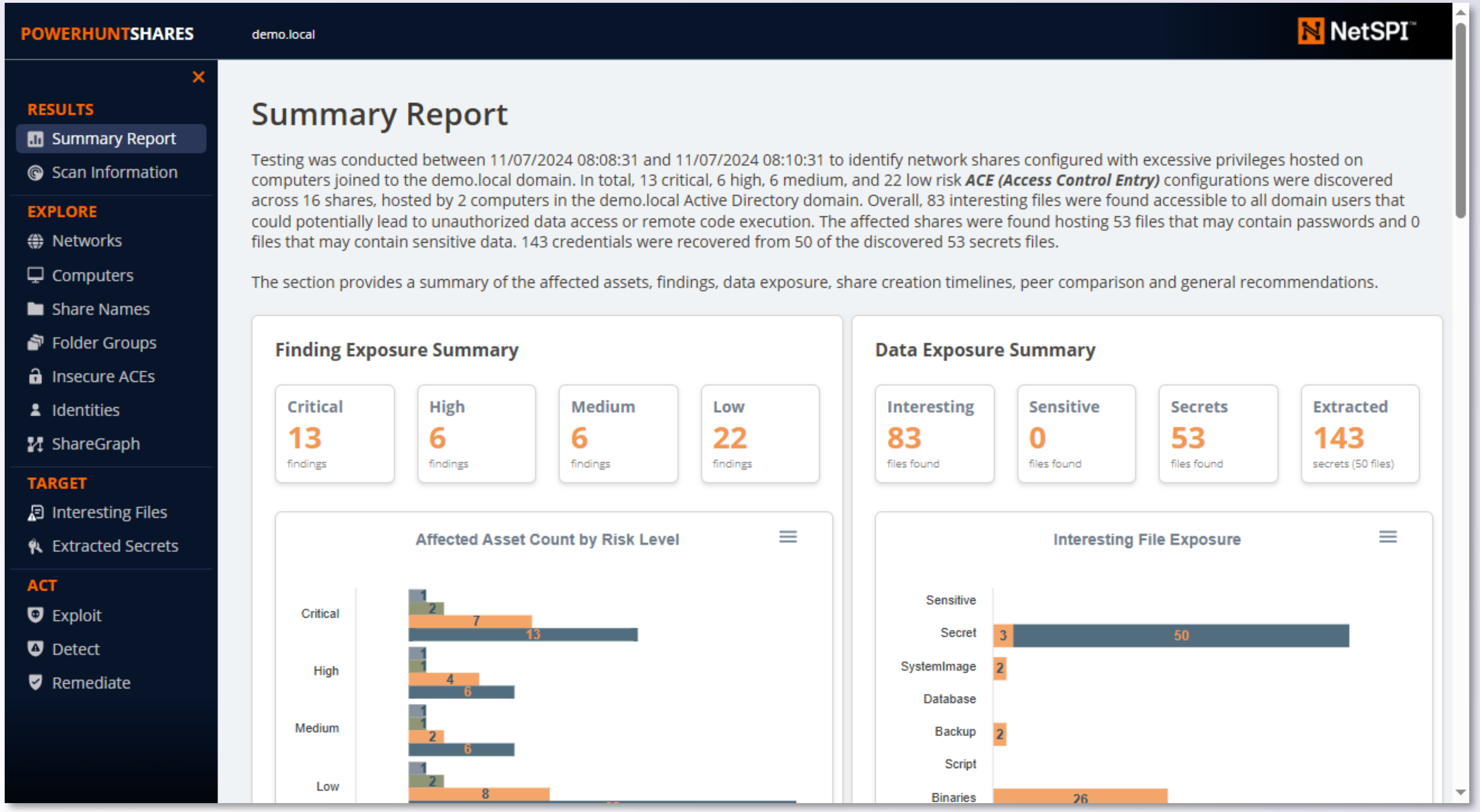
Take Actions

Take Actions

- Exploit
- Remediate
- Detect

PowerHuntShares

Demo



Summary Report

Testing was conducted between 11/07/2024 08:08:31 and 11/07/2024 08:10:31 to identify network shares configured with excessive privileges hosted on computers joined to the demo.local domain. In total, 13 critical, 6 high, 6 medium, and 22 low risk ***ACE (Access Control Entry)*** configurations were discovered across 16 shares, hosted by 2 computers in the demo.local Active Directory domain. Overall, 83 interesting files were found accessible to all domain users that could potentially lead to unauthorized data access or remote code execution. The affected shares were found hosting 53 files that may contain passwords and 0 files that may contain sensitive data. 143 credentials were recovered from 50 of the discovered 53 secrets files.

The section provides a summary of the affected assets, findings, data exposure, share creation timelines, peer comparison and general recommendations.

Finding Exposure Summary

Critical	High	Medium	Low
13 findings	6 findings	6 findings	22 findings

Data Exposure Summary

Interesting	Sensitive	Secrets	Extracted
83 files found	0 files found	53 files found	143 secrets (50 files)

Take Aways

Take Aways

- Play with your data!
- Use data analysis tools to help improve your quality of life as a defender or tester.
- Not all solutions require LLMs, but they can help save time!
- PowerHuntShares can be another tool in the box

SPECTEROPS

Thank you

Good luck and hack responsibly.

Scott Sutherland
BlueSky: @nullbind.bsky.social
x: @_nullbind
GitHub: nullbind