IT NATION™ SECURE

hosted by CONNECTWISE®

Next Generation Threat Hunting

with Algorithm Genomic Databases

Presented by

Bob Miller, COO, Global Data Systems

IT NATION SECURE

Before we get started…

3 **Make notes of questions for the end.**

2 **Put phones on vibrate.**

1 **Make sure you download the app for slides and survey.**

IT Nation Events : ITN Secure23

Agenda

1. Introduction
2. Overview
3. Why does it work
4. How does it work
5. System Integration Example
6. Questions

Introduction

- Degree in Computer Science & Mathematics
- Software Developer & Network Engineer
- 30 years in high technology industries
- CLEC, Aerospace, On-Demand Logistics, MSP, MSSP
- Patented Inventor (Satellite Safety Device SPOT)
- Eight Startups, Constant Product Development
- Innovation Engineering Certification
- C-Level positions for 20 years

Bob Miller, COO
Global Data Systems

Overview

- Cybersecurity platform which identifies new malware from their “ancestorial DNA”
- It conducts automated, deep static analysis of code after automated de-obfuscation and reverse engineering to extract functions and reduce data dimensions of malware
- Conducts automated investigations to identify and attribute malicious code by type, family, and campaign to link persistent threats
- Provides a risk assessment which provides an evasiveness/risk score of findings for suspect files
- The trend between the time a malicious piece of code enters a system and begins to attempt compromise is shortening in response to better coordinated threat hunting tactics
- Instead of months, we now see weeks and in some cases days

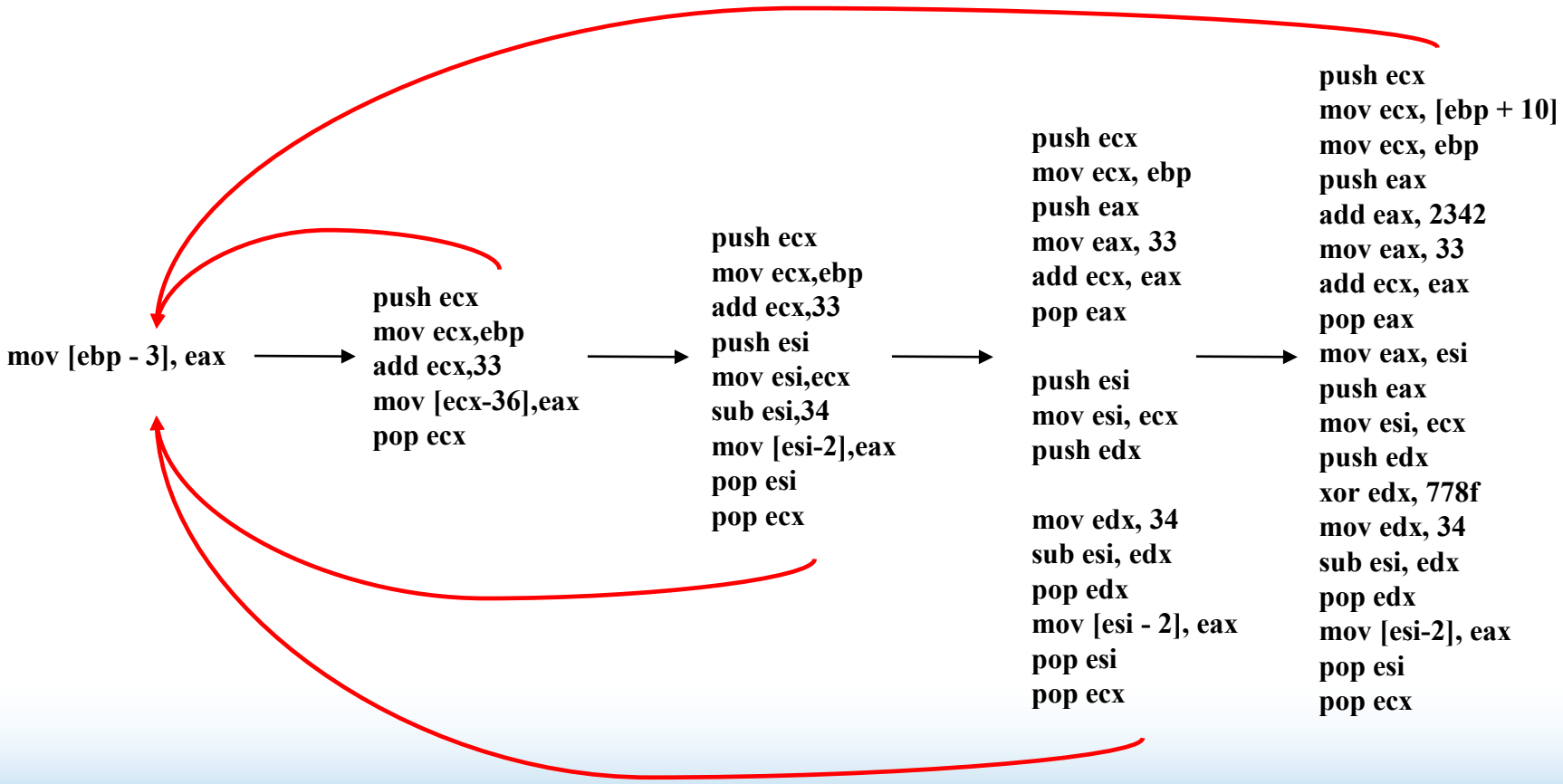
(Overview cont.) A little history…

- **Signature-based Detection:** This method relies on known "signatures“. Strings used, file hashes
- **Heuristic Analysis:** This approach aims at generic malware detection by statically examining files. Looks for rare instructions or junk code again against known threats
- **Behavioral Monitoring:** This method involves monitoring the behavior and characteristics of files to identify harmful patterns. Modifying system code or configurations.
- **Sandboxing:** Executing code in a controlled, isolated environment (a "sandbox") for observation
- **All still rely on known code, unique patterns and behaviors**

Why does it work?

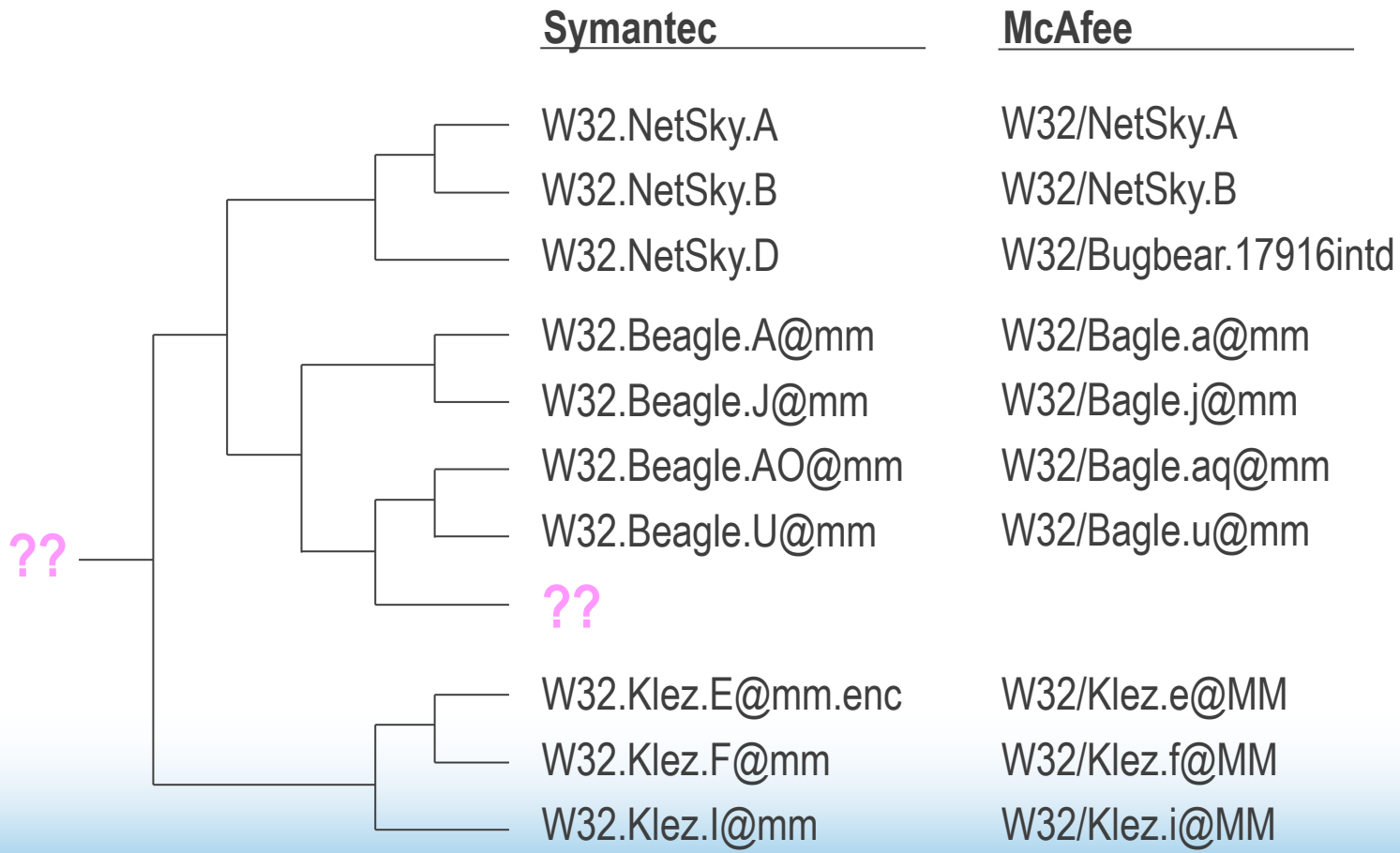
- Because “**family**” matters
- Software is made up of procedures and functions
- Malware authors often reuse and modify existing code to create new threats.
- Roughly **80% to 90%** of malicious code is based off previously authored variants.
- In effect, they use “**ancestor**” procedures/functions and apply tactics to obscure the underlying identity/purpose of the programs.

WDIW? This may hurt a little…



WDIW? Phylogeny

- Representative of the evolutionary history and relationships between groups of organisms. In this case, variants.



How does it work? Step 1: Normalization

WARNING: Computer Science Dead Ahead!

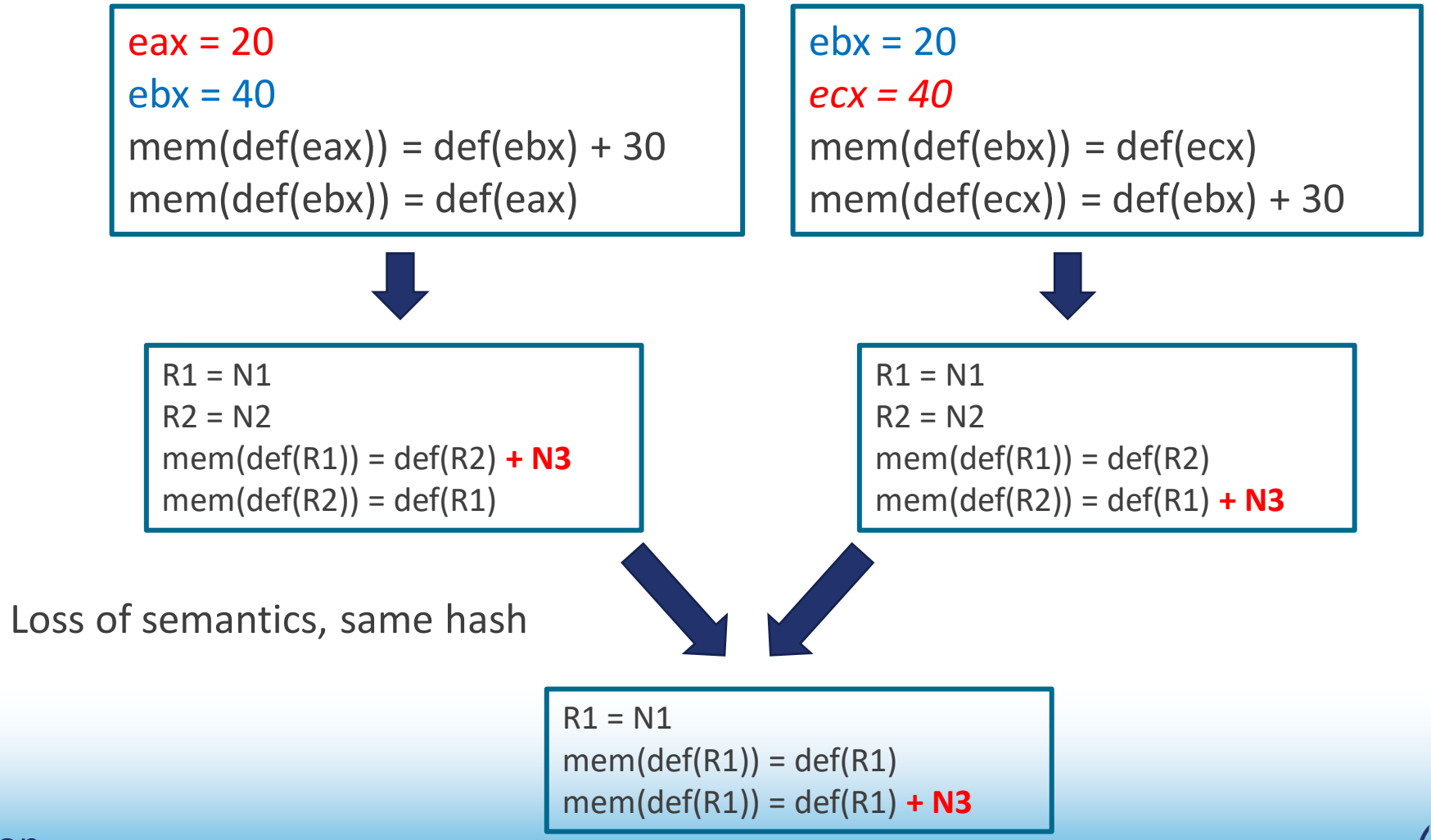
- “Normalized Semantics of Code”
  - Refers to the standardized or consistent interpretation of the **meaning and behavior** of code in a programming language.
  - Code needs to adhere to certain syntactic and semantic rules to be valid and executable.
  - **Syntactic rules** govern the structure and grammar of the code.
  - **Semantic rules** focus on the interpretation of the code. How the code behaves when executed.
- Establishes a uniform understanding of the **semantics of code** across different platforms, compilers, or programming environments.

HDIW? Step 2: Canonical Form

WARNING: Plus, some math, we are in it now!

- “Canonical form”
  - Provides for a **consistent** understanding of how code will execute.
  - Finds a way to represent different code sequences that have different semantics in a **standardized way**.
  - Represents a function as a mathematical equation
    - *(Here be magic!)*

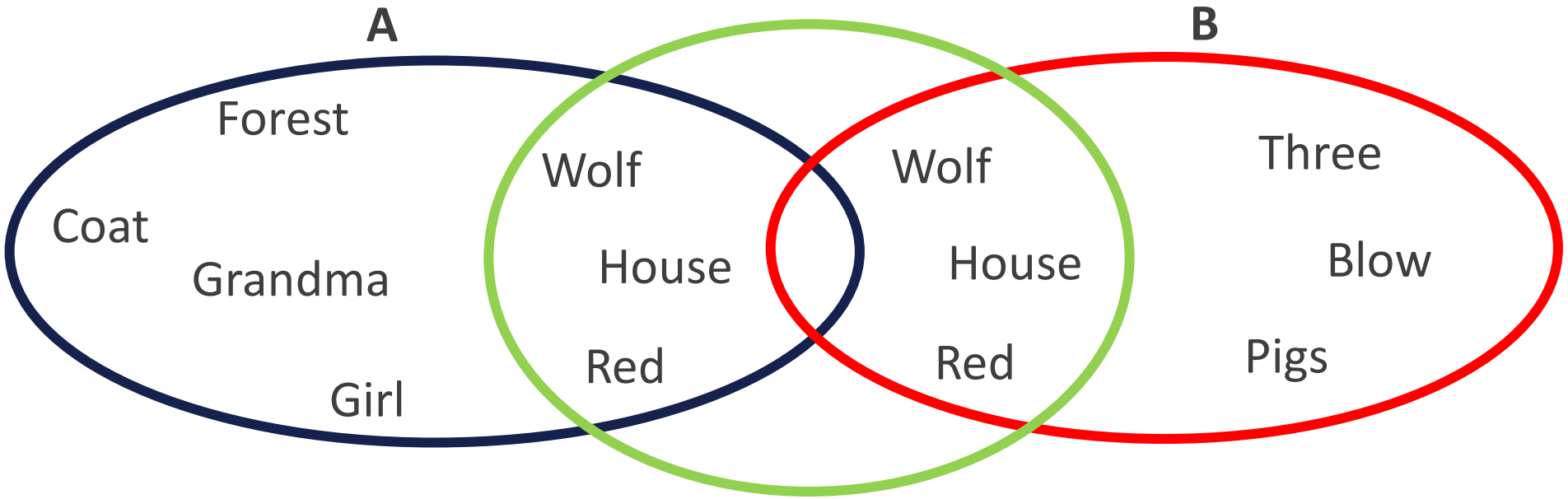
HDIW? Step 2: Canonical Form (cont.)



HDIW? Bags of features

- Each function, when in canonical form, becomes a feature with a representative hash.
- Each set of **features** is grouped within a procedure.
- Each set of procedures is grouped in a “bag.”
- Genome database indexed by Bag, Procedure, and Feature.
- Represents normalized structure and function of “program.”

HDIW? Calculating “Likeness” via features

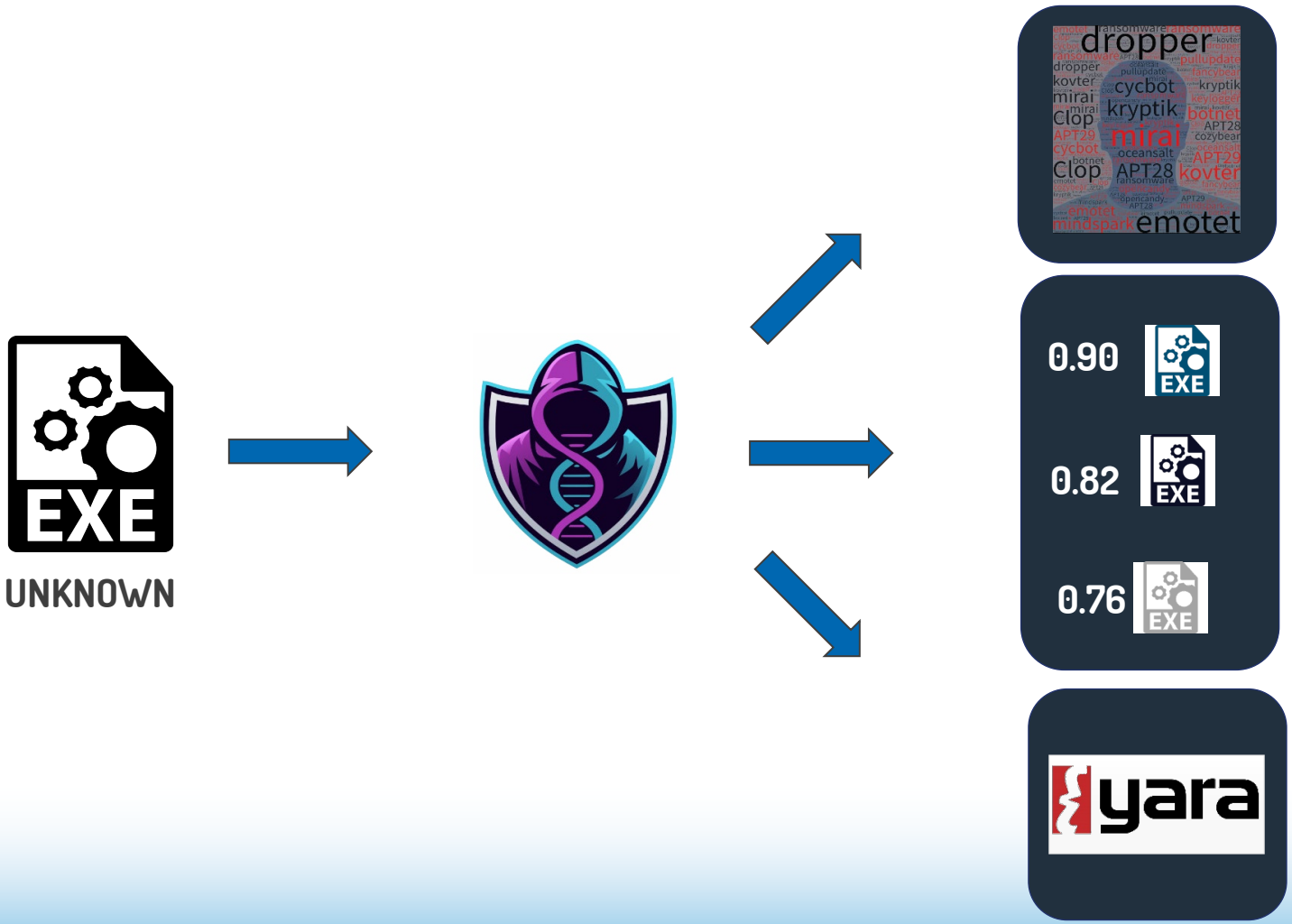


Similarity(A,B) = | A ∩ B | / | A ∪ B|

= 3 / 10

= 0.3

HDIW? Detection System

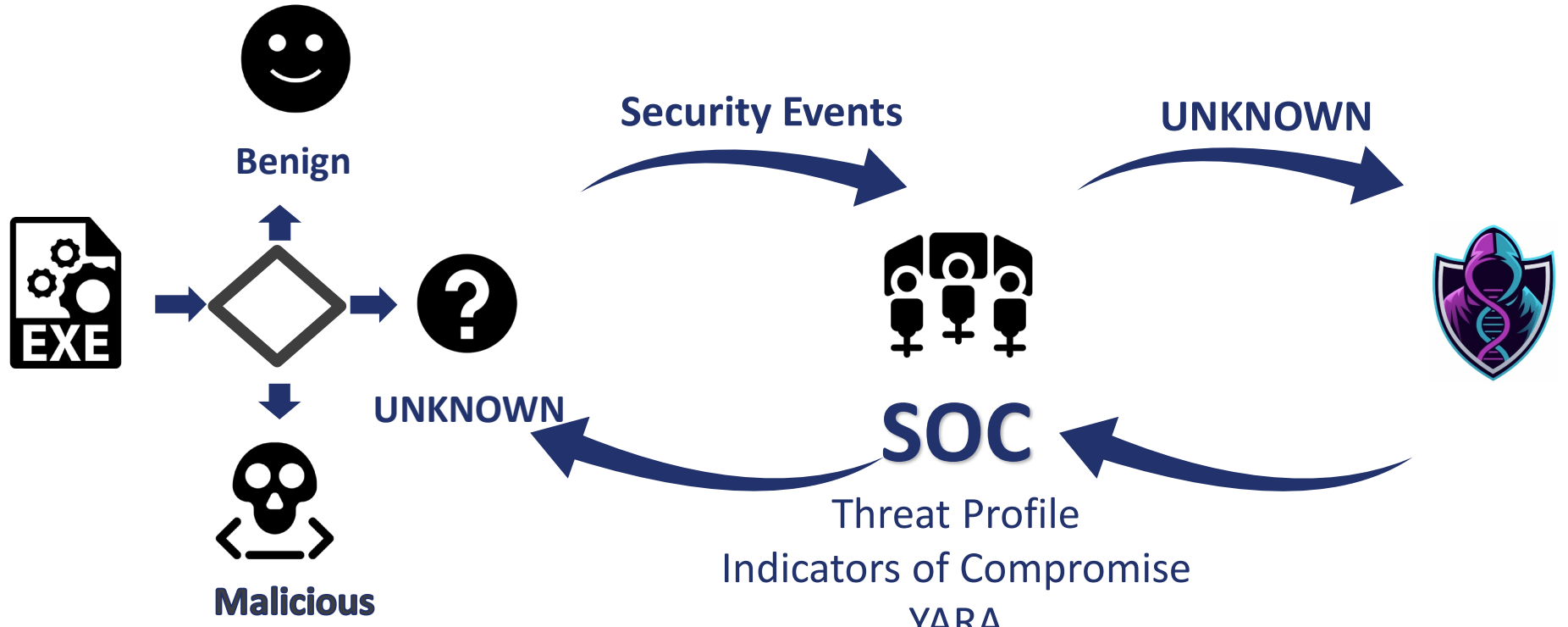


DETECT
[Family, Type]

ATTRIBUTE
[Shared Genome]

HUNT
[Code Based Yara Rule]

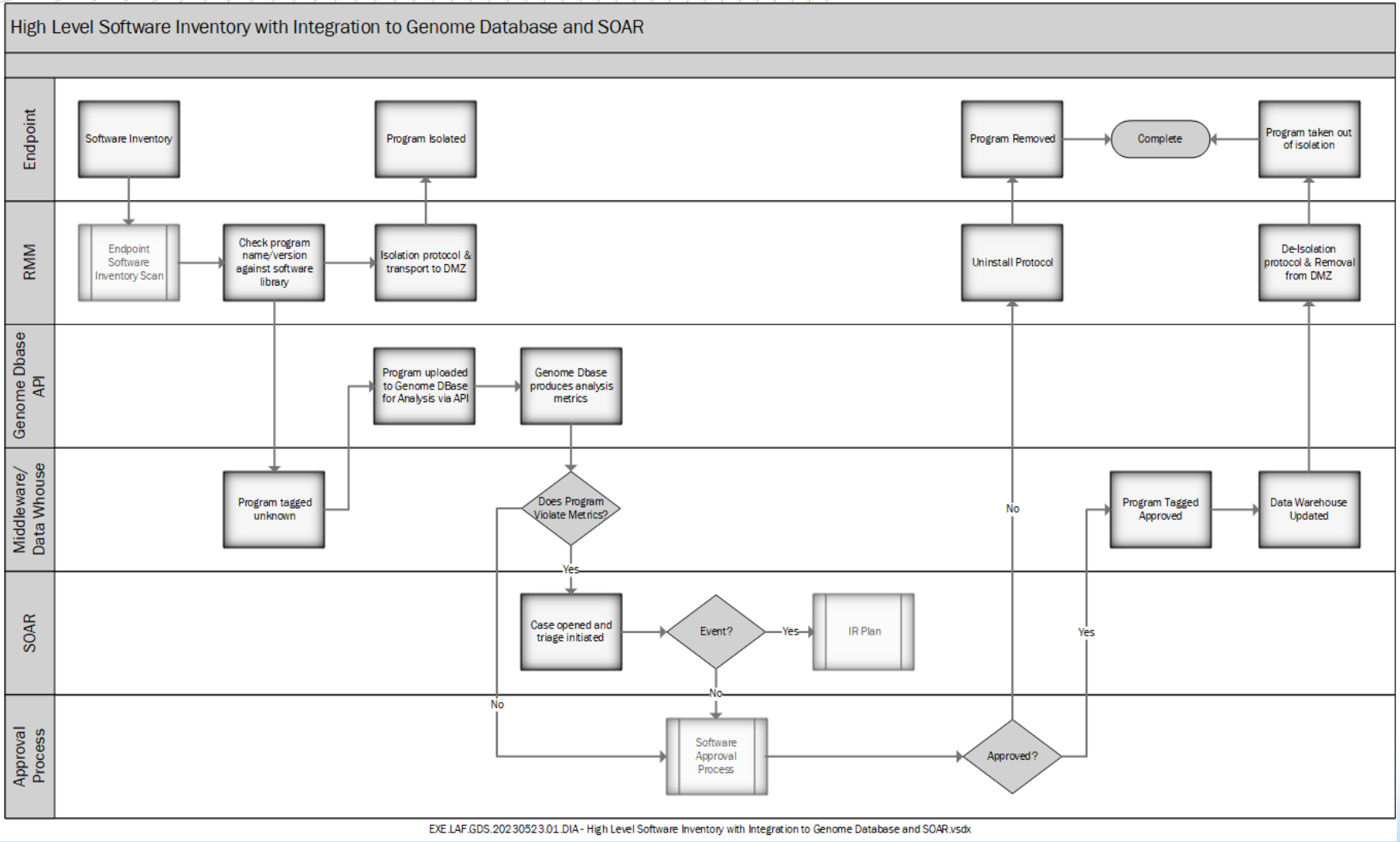
HDIW? Workflow



Benefits

- ID UNKNOWNs Missed by Others
  - Not susceptible to classic deceptions employed by malware
- Reliable Evidence for Attribution
  - Shared code is strong evidentiary connection between threats and between threat actors
- Build Enterprise Intelligence
  - Attackers reuse code; propagate intelligence about attackers over shared code
- Behavior of the “Ancestor”
    - Highlights tactics that can be used to mitigate
    - Helps target which telemetry may be the most useful

System Integration



Want to know more?

https://unknowncyber.com/

Questions?

https://www.linkedin.com/in/robertdmiller/

Contact us	
Local:	337.291.6500
24x7x365 Support:	888.435.7986 Opt. 1
Website:	www.getgds.com

Find us on Social Media	
Global Data Systems	
Global Data Systems	
@getgds	
getgds	
Global Data Systems	

Don't forget to fill out your

SESSION SURVEY