Carnegie Mellon University

Copyright 2016 Carnegie Mellon University

Building and Scaling a Malware Analysis System

Brent Frye

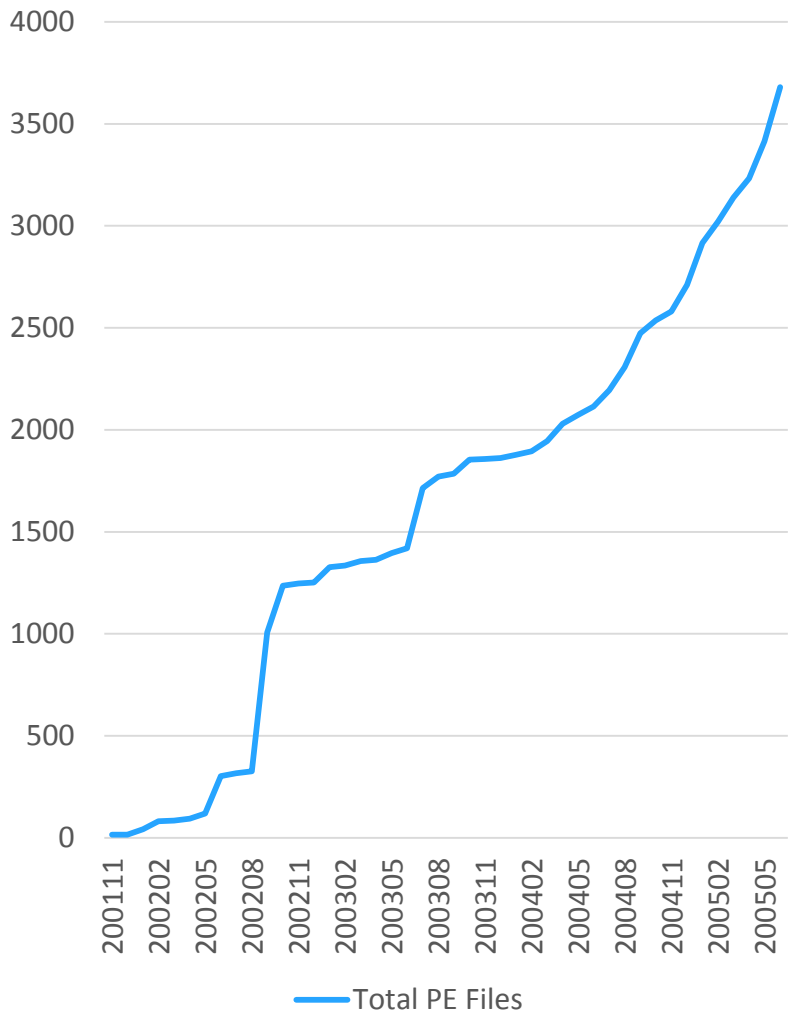
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Malware is Malicious Software

"Malware, also known as malicious code and malicious software, refers to a program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system or otherwise annoying or disrupting the victim."

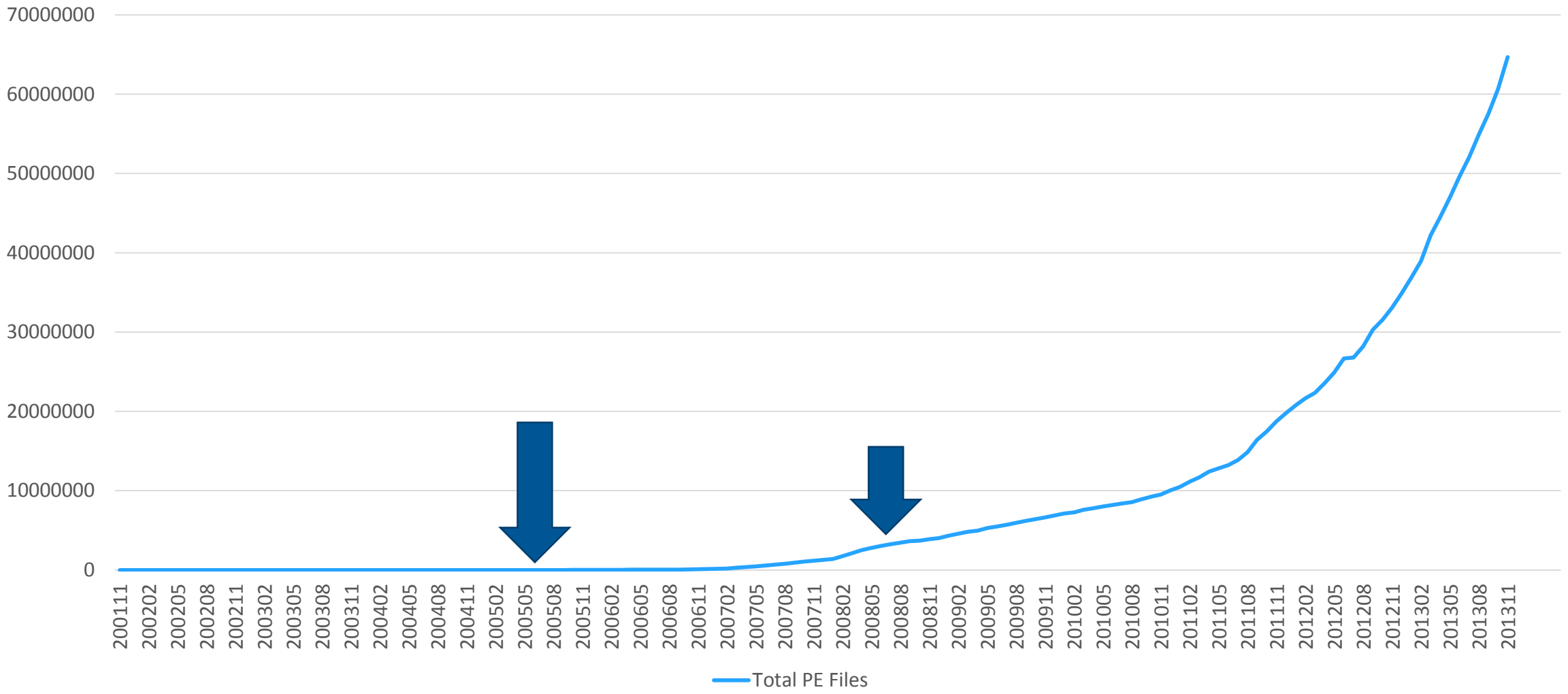
- NIST SP800-83

PE File Collection in Artifact Catalog, pre June 2005

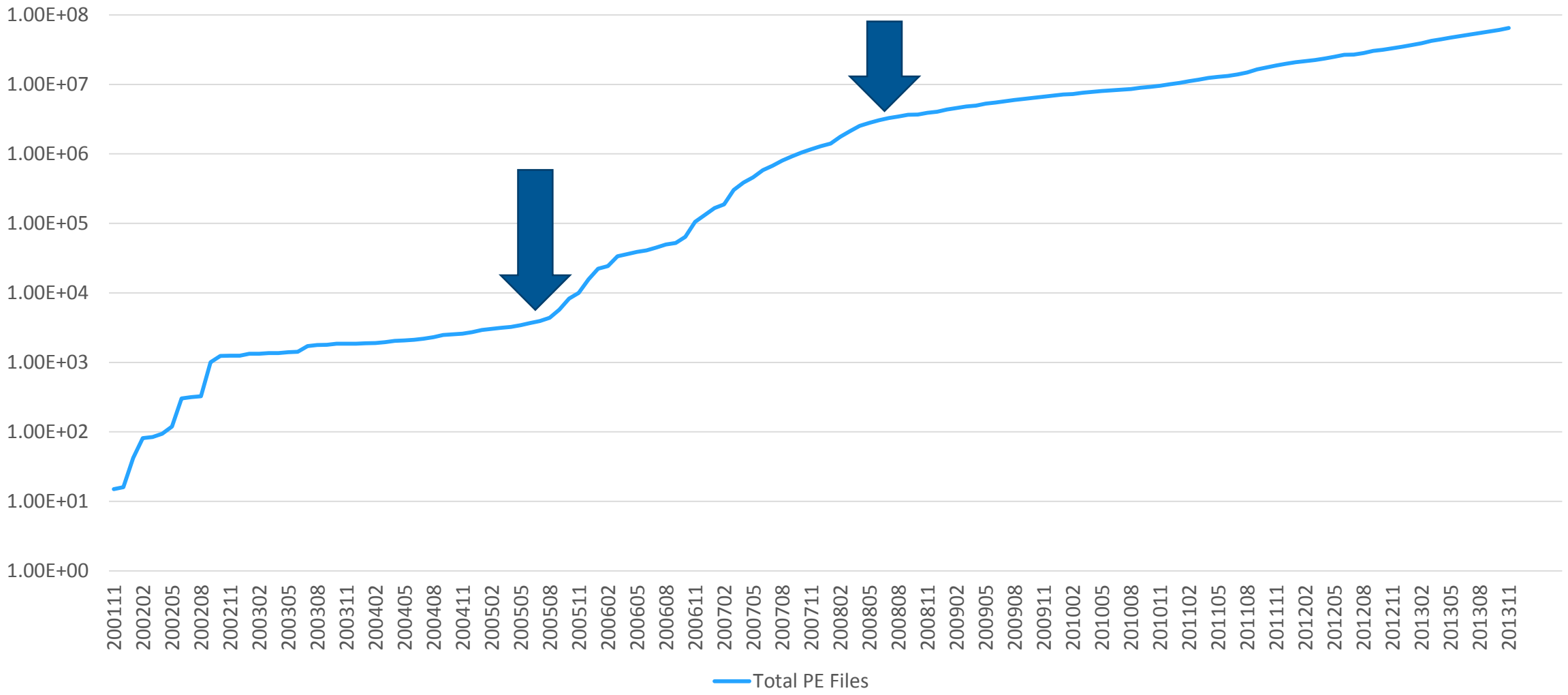


- Less than 4000 PE files in collection
- Manual collections
- Manual analysis
- Linear growth?

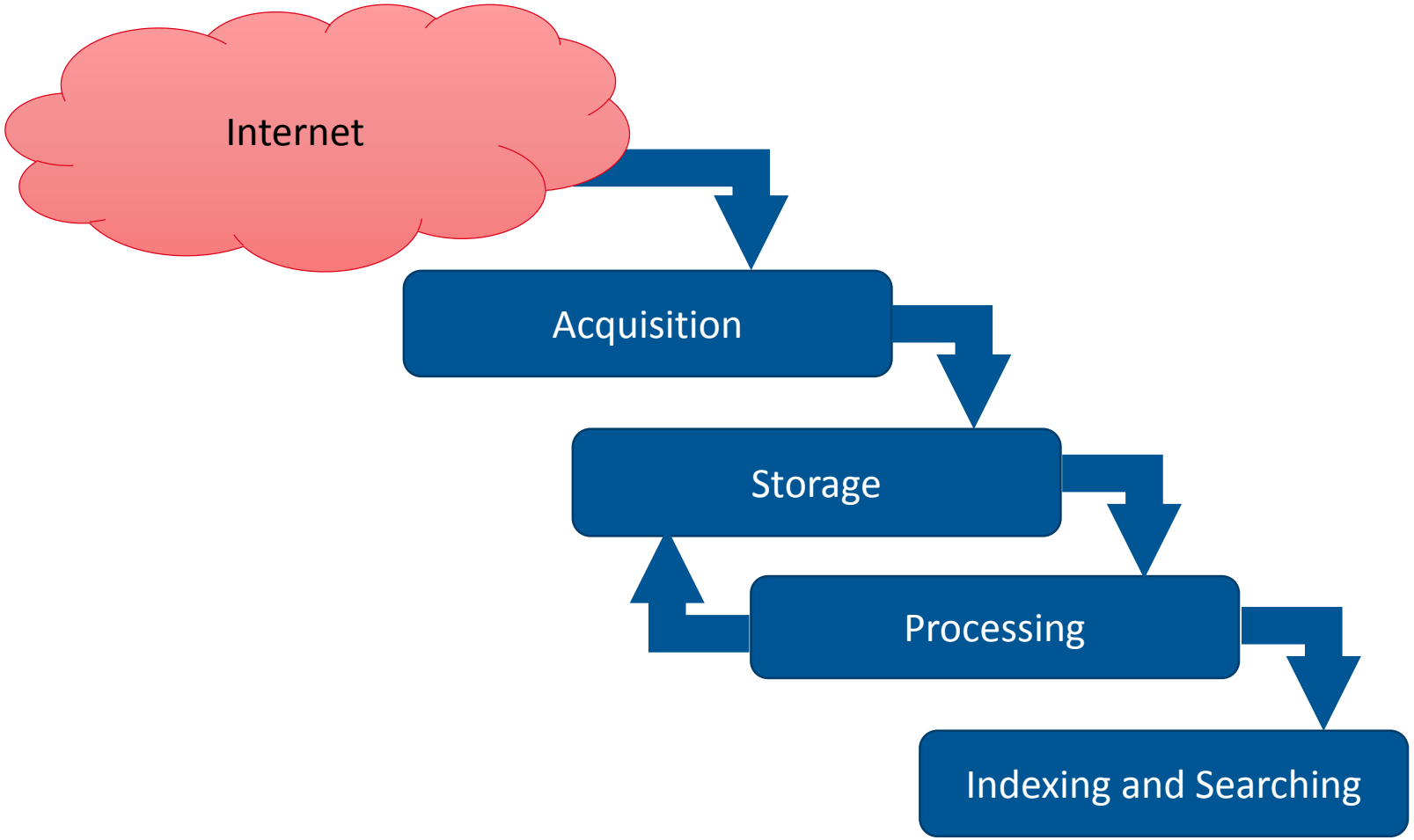
PE File Collection in Artifact Catalog



PE File Collection in Artifact Catalog, log scale



Overview of a Malware Processing System

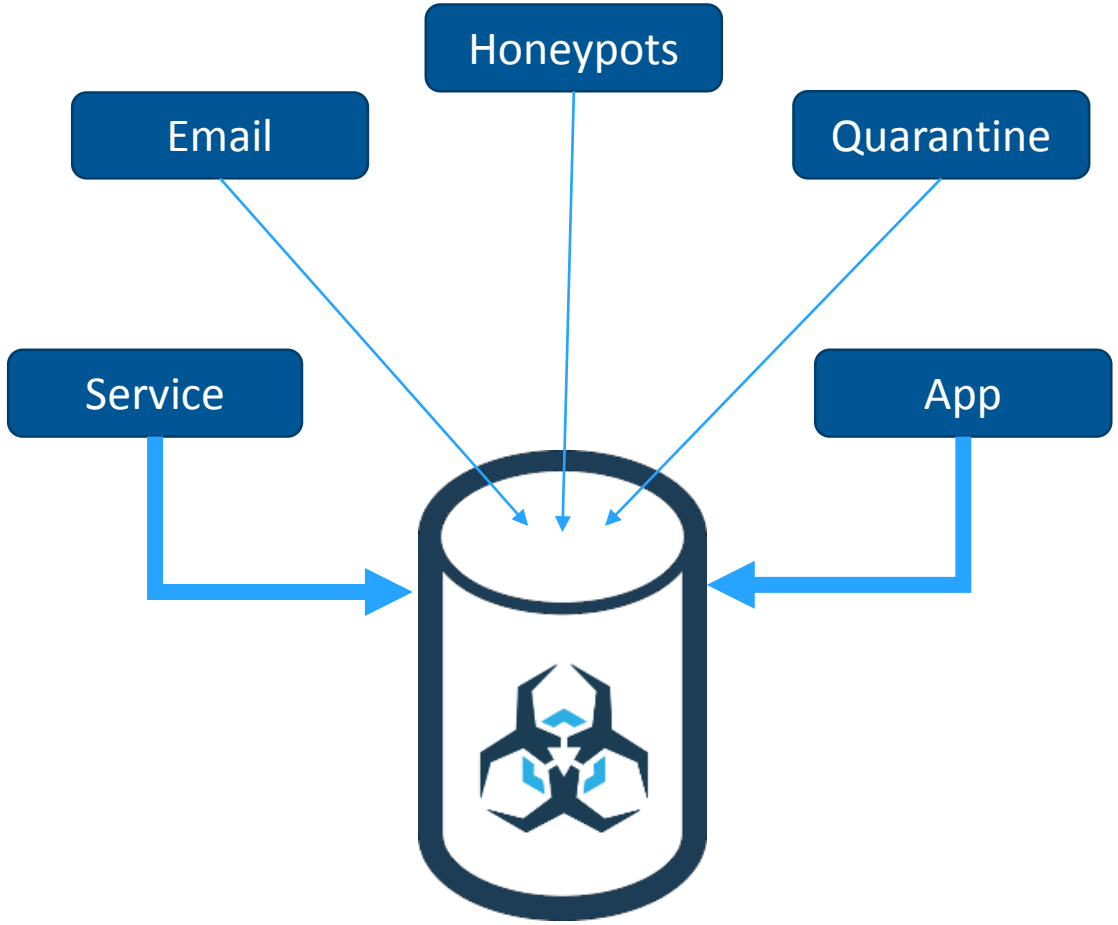


Polling Question #1

How automated is your current malware collections and analysis processing?

1) I don’t collect malware or perform malware analysis
2) Manual analysis only; no queues or scheduling
3) Partially automated collections and/or analysis, some manual effort
4) Fully automated collections and analysis
5) We outsource all malware analysis to another team or organization

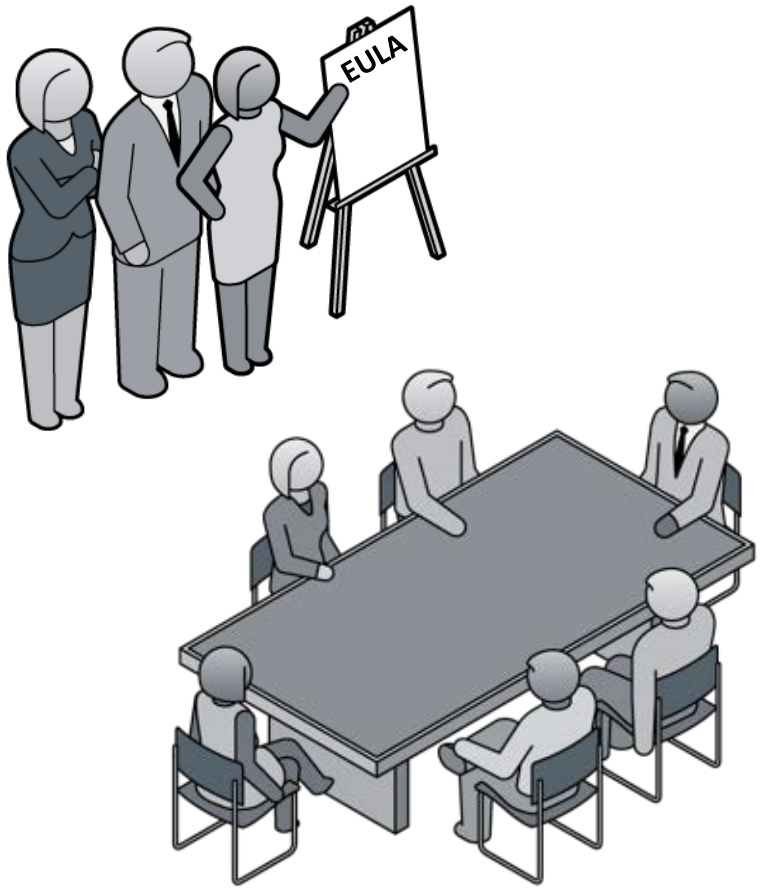
Acquisition Methods



First, you need to determine how you will acquire malware samples:

- Mine honeynets
- Export AV quarantine
- Host a submission form
- Request email submissions
- Malware hosting services

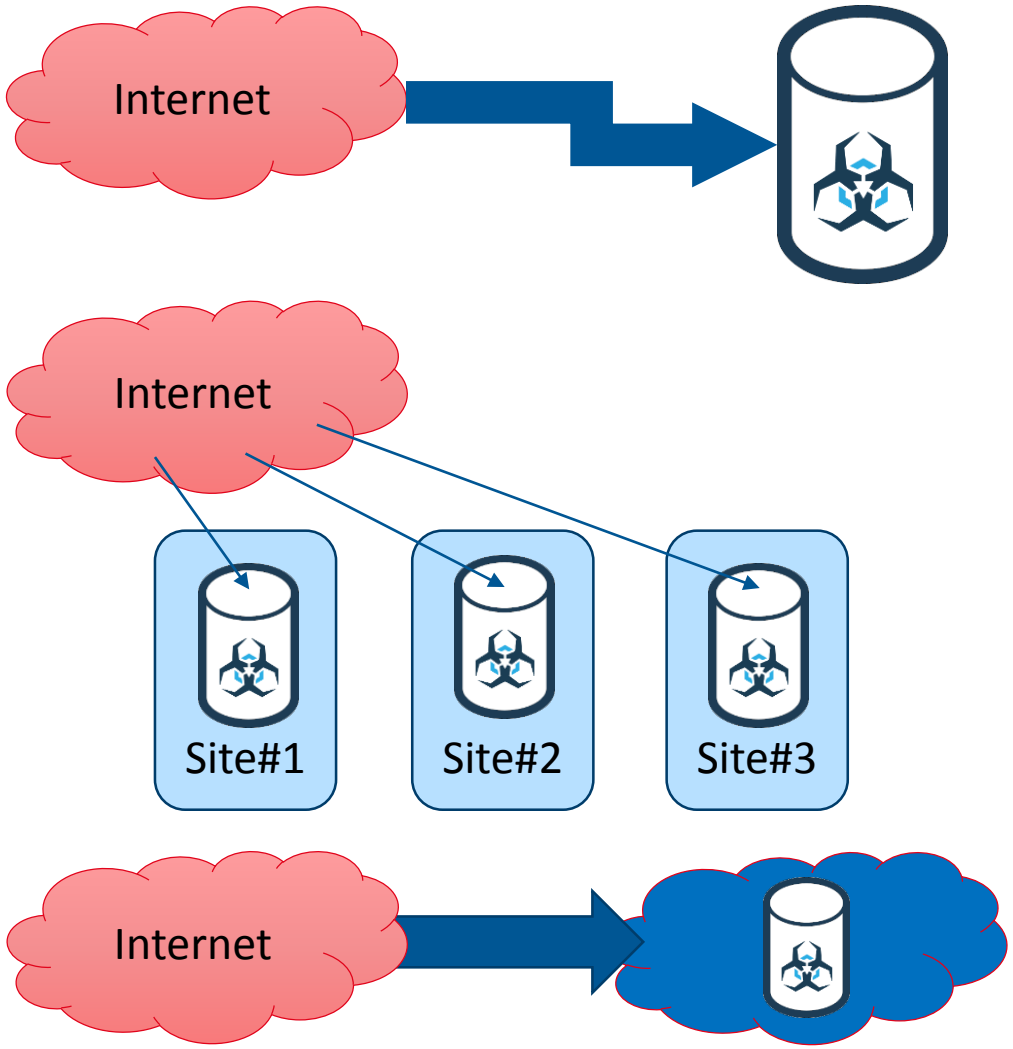
Policies Matter



Be mindful of relevant policies

- Organizational policies
  - Email services
  - Downloading files
- Hosting providers
- Legal authorities

Acquisition – Network and Storage Considerations



Scaling the network

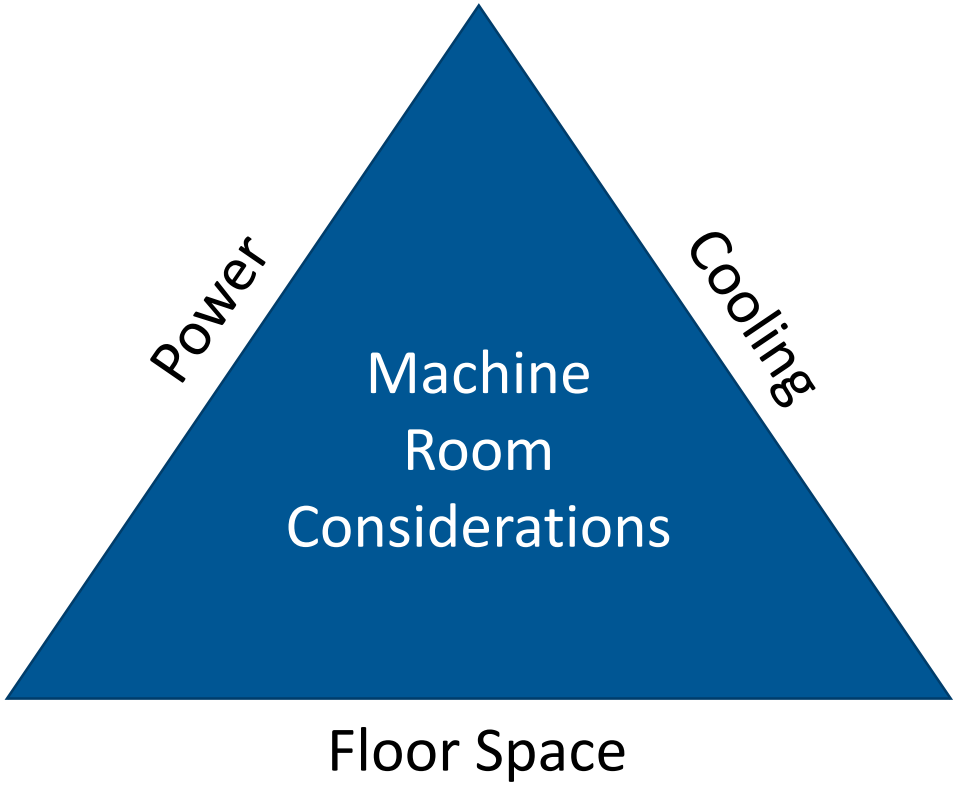
- Increase bandwidth at central site
- Multiple sites
- Cloud services

Scaling storage

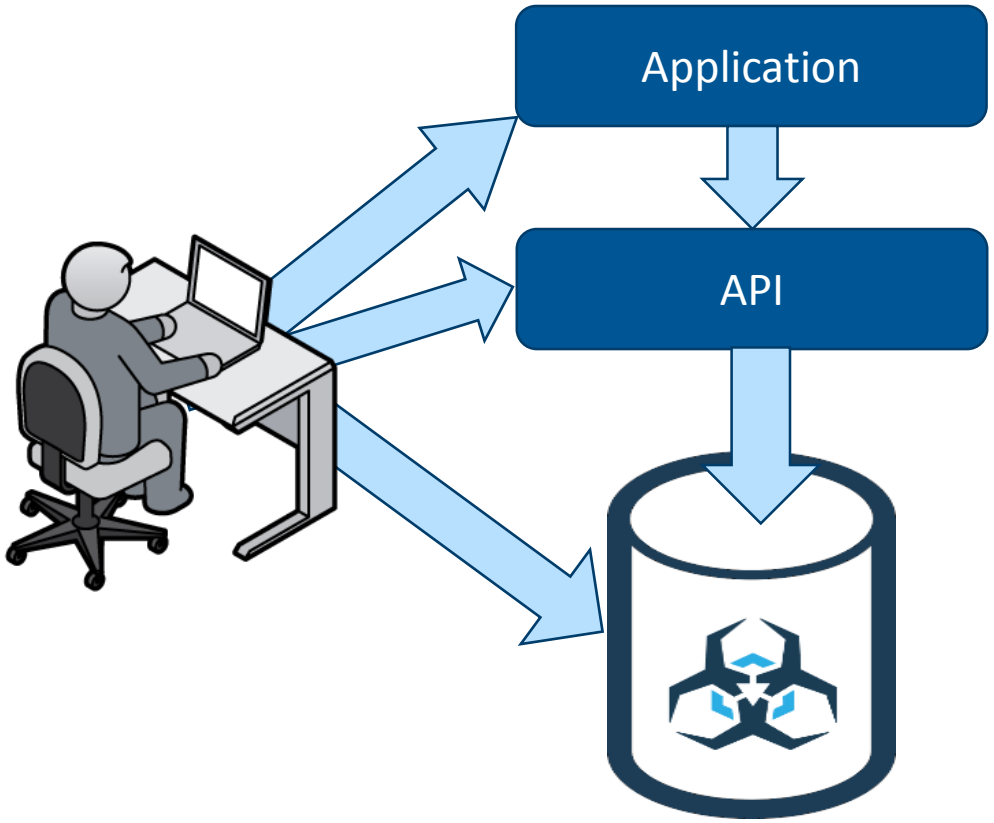
- NAS/SAN
- Distributed filestore
- Cloud storage

Parallelize wherever possible to improve bandwidth utilization/throughput

Machine Rooms



Consider User Access Methods



Direct Access

- Users access database or filestore directly
- Possibly a variant of API

API

- REST
- JSON or similar format

Application

- Command-line
- Web UI

Acquisition Metadata

Who?

Who gave it? Who looked at it?

What?

What is it?

When?

When did we get it?

Where?

service, website

How?

email, web, service

How Many?

Keep all? First from source? First?

If you don’t collect this information as part of the acquisition process, there is generally no way to re-acquire this metadata

Storage

Flat files

RDMS

- Postgresql
- MySQL

NoSQL

- CouchDB
- MongoDB
- Cassandra
- Hadoop

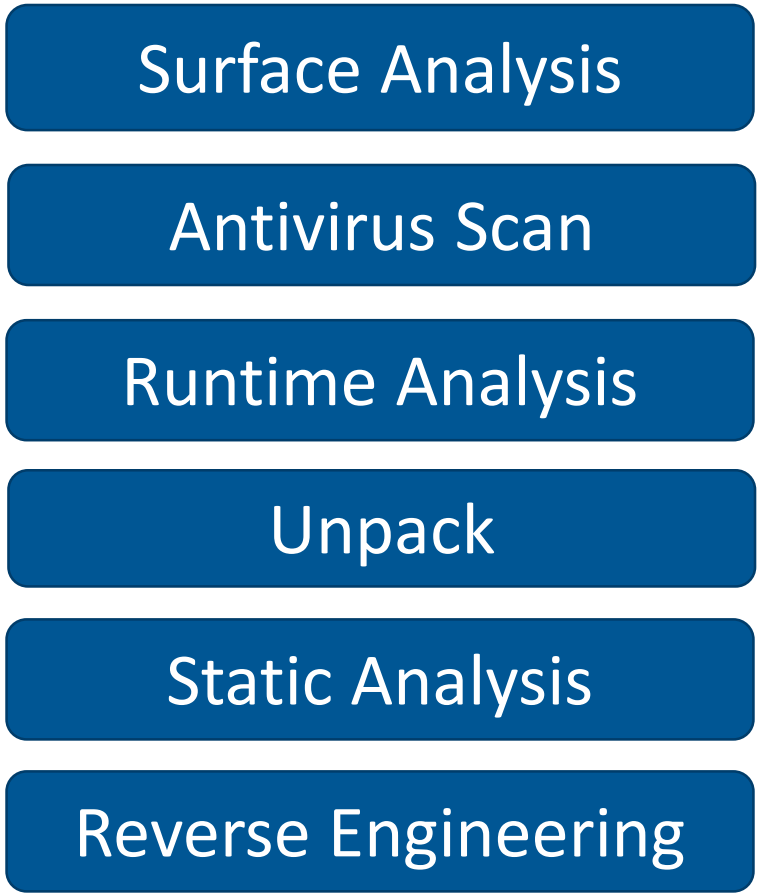
Name files so they can be easy to locate and so different files won't overwrite one another

Polling Question #2

How much malware does your organization collect (or soon hope to collect) each day?

1) Less than 100 per day
2) Up to 1000 per day
3) Up to 10,000 per day
4) Up to 100,000 per day
5) Over 100,000 per day

Processing Malware



Malware Analysis Goals

Before building a system, determine what you hope to achieve with the system.

- This is not an enterprise protection system
- If you just want indicators, you don’t need to reverse engineer everything in the code

How Do We Get Here?

We can either begin our malware processing from the Acquisition step or the Storage step.

Acquisition

- Push all new files onto a queue
- Possibly the first thing the queue does is to insert the file into datastore

Storage

- Store the data first
- Query the datastore to determine “new” files

Malware Processing - Surface Analysis

SURFACE ANALYSIS

d14deadbeef…
e2f00blahblahblah

What information do you want from the files that does not require running them?

- Filesize, filetype, mimetype
- Hashes: md5, sha1, sha256, ssdeep

Malware Processing – Antivirus Scan

Running each file through AV scanning can provide some information about the file

- Use multiple AV engines
- Names may not be accurate
- Some files may not be found using AV
  - Maybe false negative
  - Maybe the file isn’t malicious
- Beware of false positives

Malware Processing – Runtime Analysis

Execute the malware in a virtual environment and see what it does:

- Network access
- File create, update, delete
- Mutexes
- Service start, stop
- Memory dump
- Screenshots
- Often allowed to run for 5 minutes, additional post-processing time needed

CERT Anexa, Cuckoo Sandbox, Cisco AMP Threat Grid

Typical Runtime Analysis Results

Analysis report (XML or JSON)

PCAP file

Dropped Files

  Files created or altered

Indicators of Compromise

  IP addresses, hostnames, mutexes, filenames, registry entries, service changes

Log files

How will users access this data?

How much extra storage is needed?

Should we also process the dropped files?

- Any output of a malware run may be malicious
- May cause deep recursion loops

Malware Processing - Unpack

packer

1. Software that compresses other software.

2. A malicious tool that compresses and obfuscates software in order to defeat anti-reversing.

* The MAL: A Malware Analysis Lexicon

If a malware sample is packed, then it needs to be unpacked to be able to reverse engineer or to obtain useful static analysis results.

Non-malicious examples: gzip, pkzip

Common malware examples: UPX

Polling Question #3

How recent does malware analysis have to be in order to still be useful?

1) I don’t have any use for malware analysis
2) I need complete results in 30 minutes or less
3) I need complete results in 48 hours or less
4) As long as it’s done in a month or so, that’s OK with me
5) It doesn’t matter how old the analysis is, I’ll still find a use for it

Malware Processing – Static Analysis

Process file without executing it

- Strings
- Objdump
- PE file sections
  - Section Name
  - Section Size
  - Section Hash
  - Other filetypes (e.g., PDF, Office) have internal sections as well
- Extract functions

Malware Processing – Reverse Engineering

Deconstruct malicious code to understand how the malware behaves on a system at a binary level

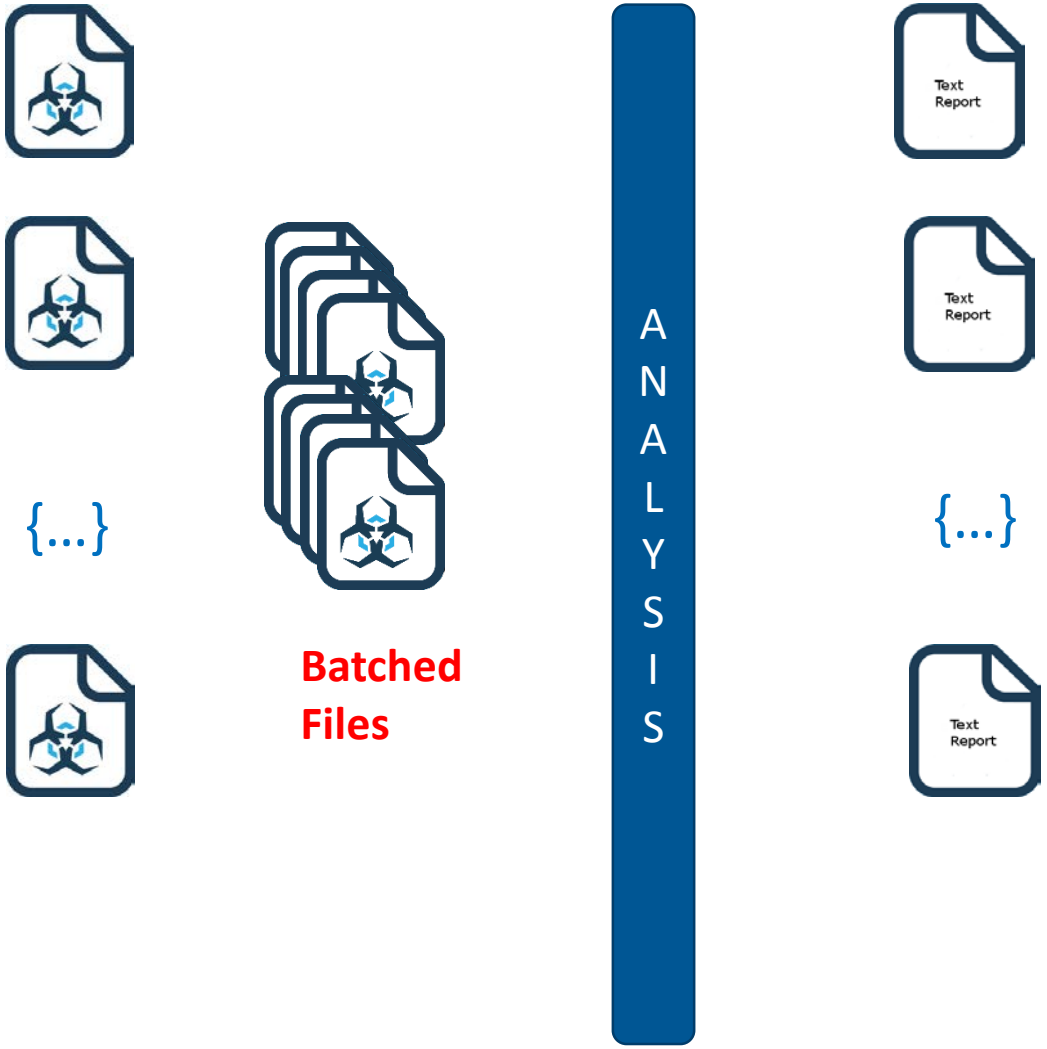
Dissassemble and Debuggers

- IDA Pro
- OllyDbg

Memory Dumper

- OllyDumpEx
- Volitility

Malware Processing Concept – Batching Files



Some processes have large overhead for startup and/or teardown

- AV Scanning
- Unpacking

Rather than processing single files, put a group together and process the group

- Must be able to extract individual results
- Entire group may fail due to one bad file

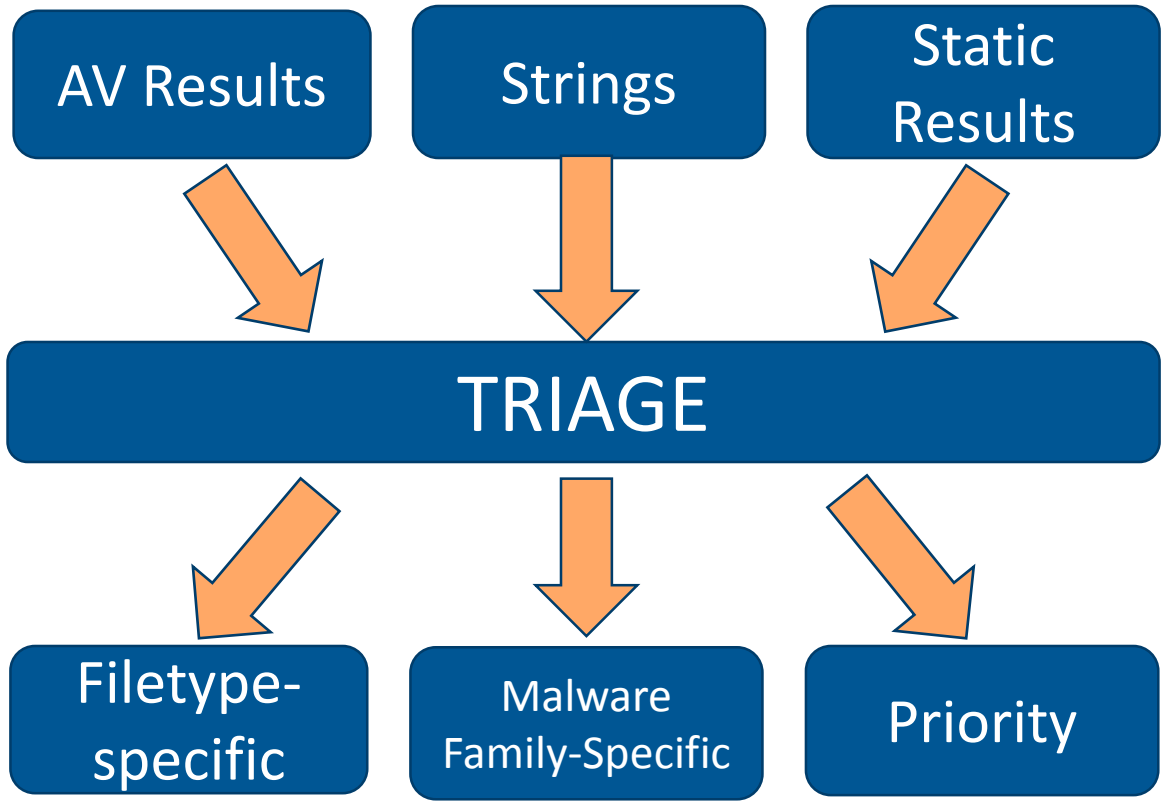
Malware Processing Concept – Triage

Tri*age – verb – assign degrees of urgency to (wounded or ill patients)

Generally, this comes before time-consuming dynamic analysis

- All malware is created differently, some less different than others
- For example, do we really need to analyze every one of the 800,000+ “Allaple” samples we have?
- Run on all candidate samples to deprioritize (or skip entirely) known malware

Malware Processing Concept – Triage-2



What can be used for triage?

- Results from static analysis
- Searching for specific strings
- Other information provided by source or third party

Indexing and Searching

Specific file(s)

Metadata (Acquisition, Surface Analysis)

- Few rows per record
- Traditional RDBMS may be sufficient

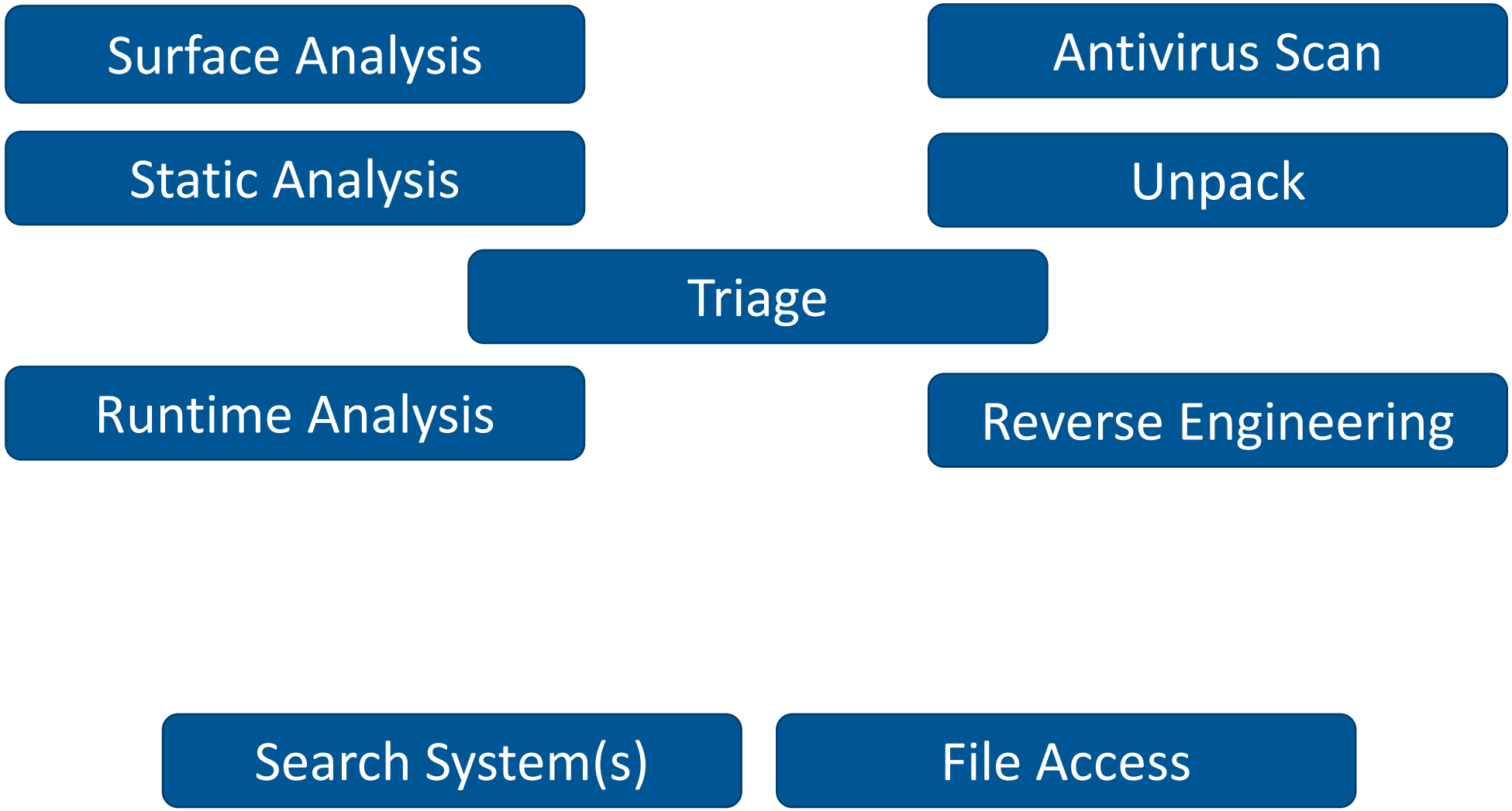
Analysis Results

- Possibly 1000s rows per record
- Hadoop or other NoSQL may be better to handle variety of structures

Full-text search

- Search possibly hundreds of terabytes
- CERT BigGrep

Workflow



Questions?