Security Exclusions, Endpoint Controls, ...and You!

$whoami

- Exec. Director of Threat Intelligence at SCYTHE
- IANS Faculty, former SANS Instructor
- Former NSA Hacker, endorsed by Shadow Brokers
  - aka Russian Intelligence
- Digital terrorist, breaker of software, responder of incidents, reverser of malware, injector of code, spaces > tabs
- **Dislikes:** those who call themselves “thought leaders,” “crypto bros,” and anyone who **needlessly adds blockchain** to a software solution

Agenda

- Security Exclusions
- Do We Really Need Another Model?
- Introducing the Equilateral of Exclusion Risk
- Exclusion Gotchas
- Closing Thoughts

Security Exclusions

Exclusions are a reality for most orgs…

Security Exclusions – Background

- Simply put, a security exclusion is a tool to prevent alarming on something you otherwise would

- Detection engineering teams build use cases, but these may create too much noise, leading to the need for exclusions

- Exclusions can be thought of as false positive reduction tools
  - If the detection rule works, but results in too many false positives, exclusions are needed

Detection Engineering: Reality Check

- Most organizations don't have a full-time detection engineer
- Without detection engineering to create custom use cases, orgs primarily use out of the box detections provided by their security tool vendors
- When these detection cause false positives, they either:
  - Disable the detection rule entirely
  - Create an exception for the rule

Security Haves and Have Nots

- Organizations with full-time professional detection engineers probably don't need this framework
  - They are the "security have's"
- The "security have nots" are left to implement the simplest exclusions they can apply or disable a detection entirely
  - "Simplest" rarely aligns with "best for security"

Exclusion Rule Necessity Examples

- A custom developed application that is critical to business functionality uses a licensing routine that is heavily obfuscated
  - This obfuscation is identified as malicious by the EDR
- A major sporting event venue sells commemorative screen savers that are packed using commercial tools
  - For copy protection, during installation, the machine ID is encoded in the screen saver binary, so the hash is different
- A business critical application that updates frequently creates a RWX section of memory and unpacks itself there
  - This is detected as malware by the EDR

Do We Need Another Model?

Signs point to yes…

A Word on Models

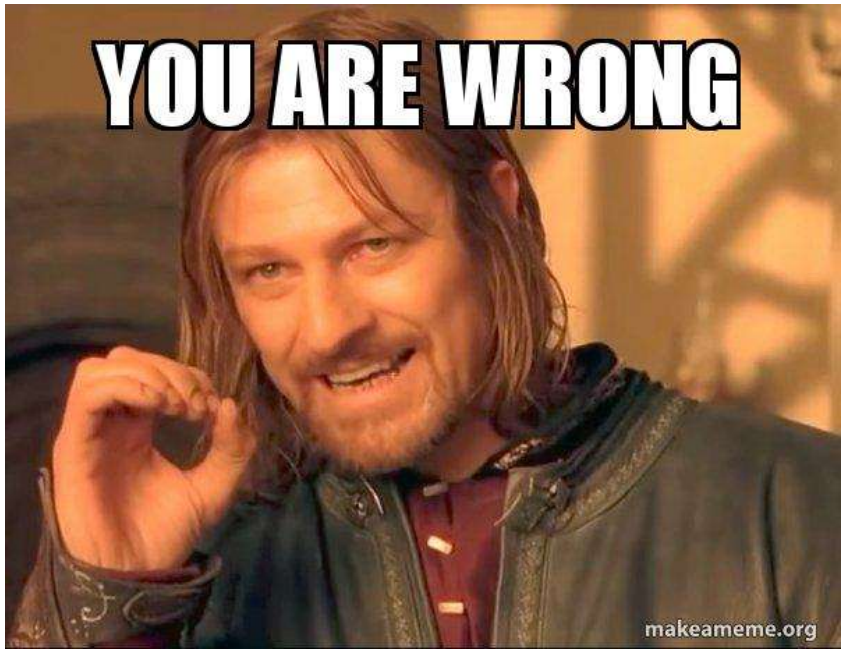
- Models introduce academic rigor into areas where we might otherwise try to reinvent the wheel

- More importantly in cybersecurity, stakeholders often think we're just making this up as we go
  - Truth be told, we often are

- This assumption is exacerbated by low rates of formal education in cybersecurity relative to other fields

Academic Rigor Matters in Cybersecurity

- It's okay if you disagree on this point
  - But you're wrong
  - You just are

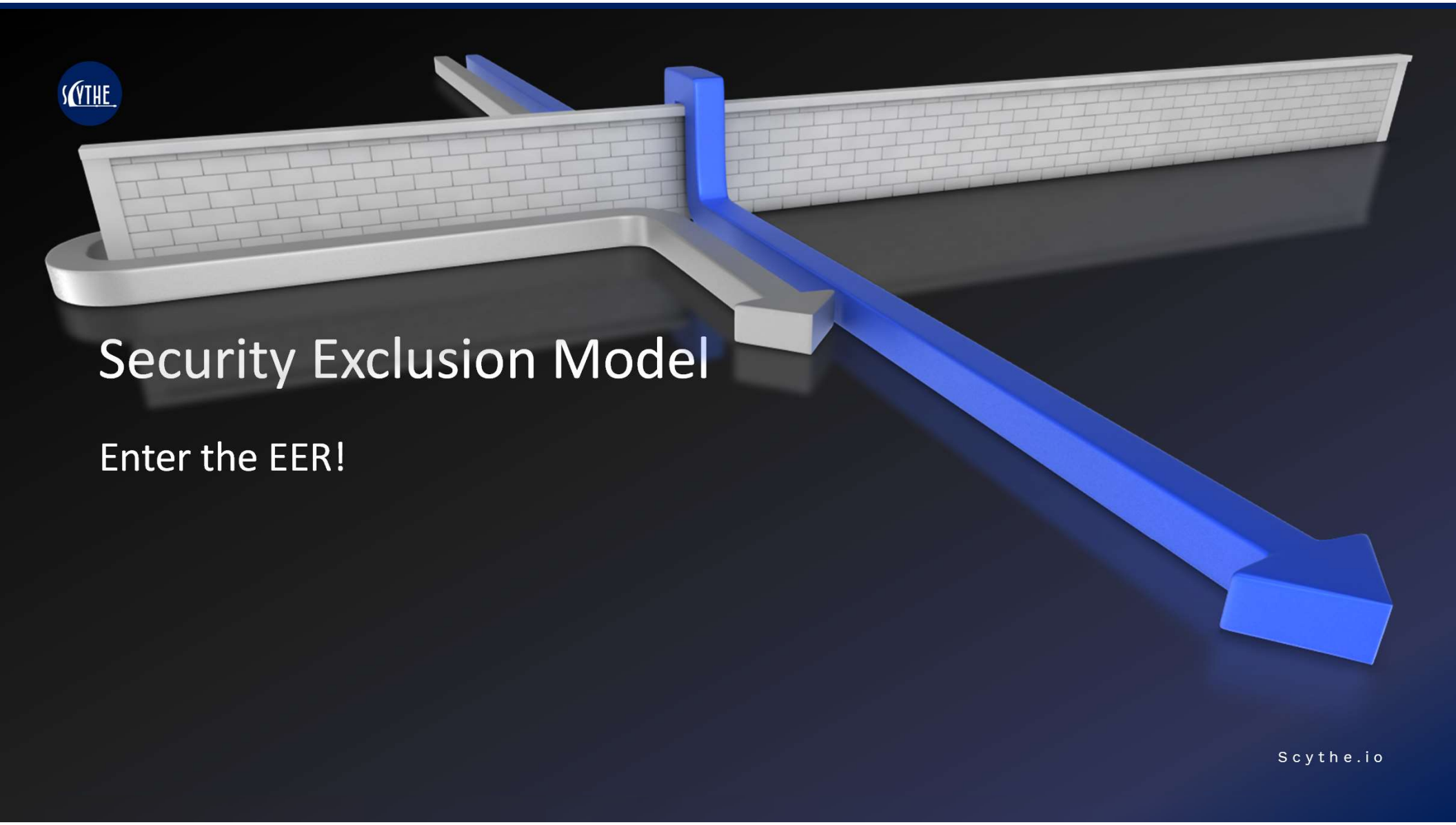


- Use **every** tool at your disposal to stop fighting the "we need degrees" debate and focus instead on fighting the **threat actor**

Independent Validation of Model Necessity

- In my work at IANS, I've taken many "Ask an Expert" calls from clients discussing what types of exclusions are best
  - IANS clients are relatively high on the maturity scale
  - I sincerely believe there are many lower maturity organizations that are just as confused but have nobody to (easily) ask

- Side note: many questions focus on whether exclusions are even necessary at all
  - They **absolutely are** and I'll fight anyone who argues otherwise



Security Exclusion Model

Enter the EER!

Introducing the EER

- Equilateral of Exclusion Risk (EER) is a model demonstrating the relative risks of different exclusions that can be applied to endpoint security controls

- Yes, equilateral is hard to spell
  - And even to say
- But marketing loves a good acronym and this just works
  - Not many shapes begin with "E"

Using the EER

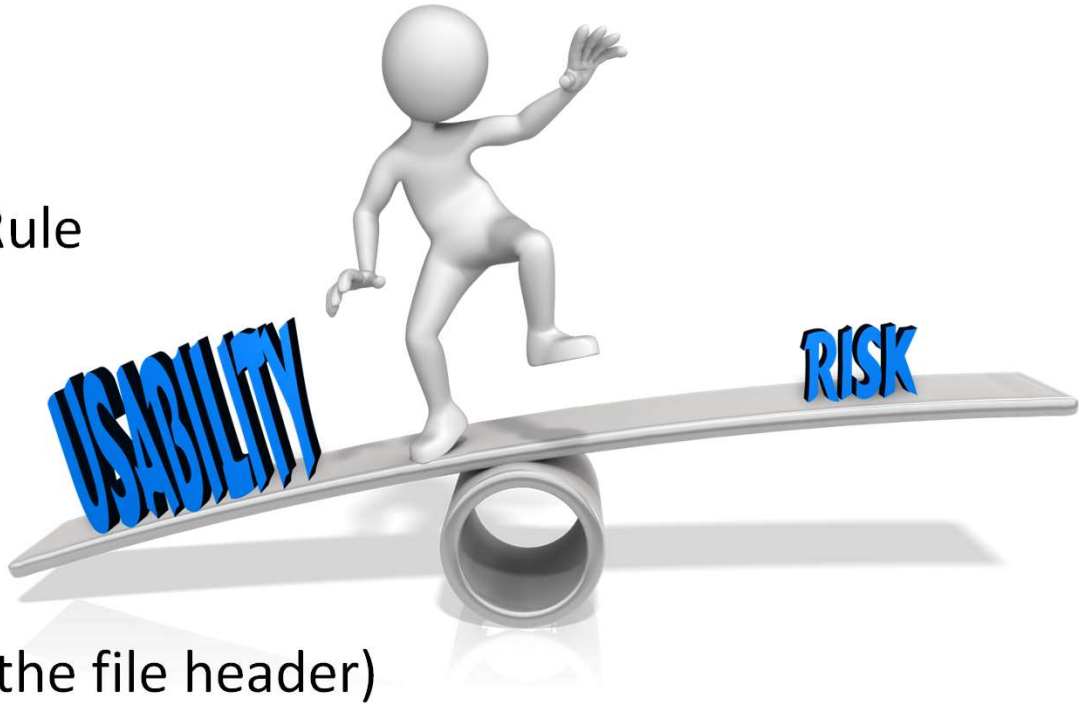
- When faced with the need to write an exclusion, use the exclusions highest on the EER model that are supported by your endpoint security tooling
  - The EER addresses creating the best detection rule exclusions on a given endpoint, user, or group of endpoints
- The simplest exclusions often involve exempting an endpoint, user, or group of endpoints from the application of a rule
  - Selecting the right endpoints/groups to apply detection logic to is not addressed by the EER

Equilateral of Exclusion Risk (EER) Key Principles

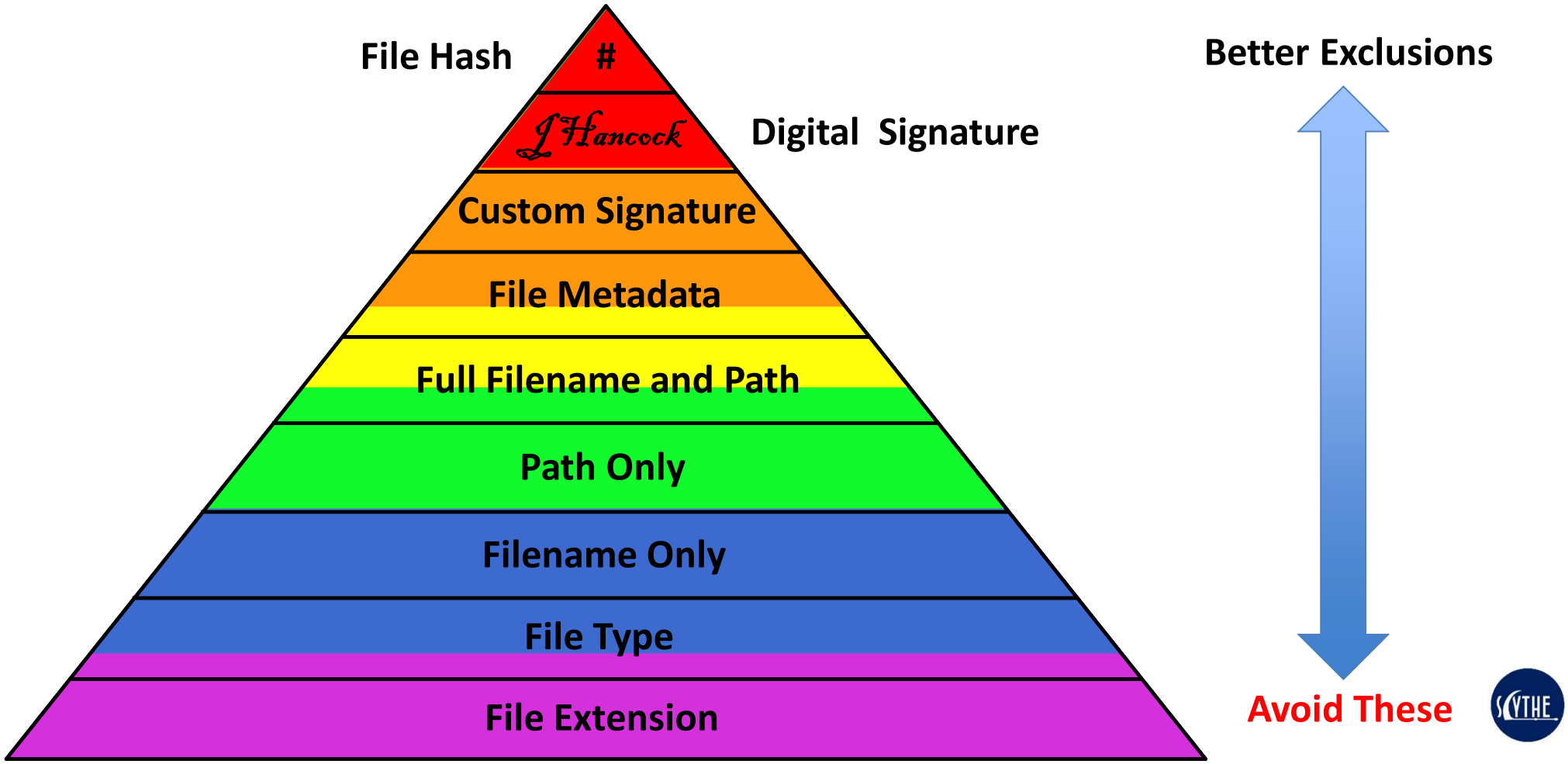
- There exist some activities that cannot be detected reliably without some exclusions built into the detection logic
- Every exclusion introduced some risk of a bypass
- Not all categories of exclusions introduce the same bypass risk
- Optimal exclusions may not be supported by the security controls deployed by the org
- Detection engineers should select the exclusion or exclusions with the lowest risk of bypass
- As controls are updated, exclusions should be reviewed

EER Order Of Exclusions

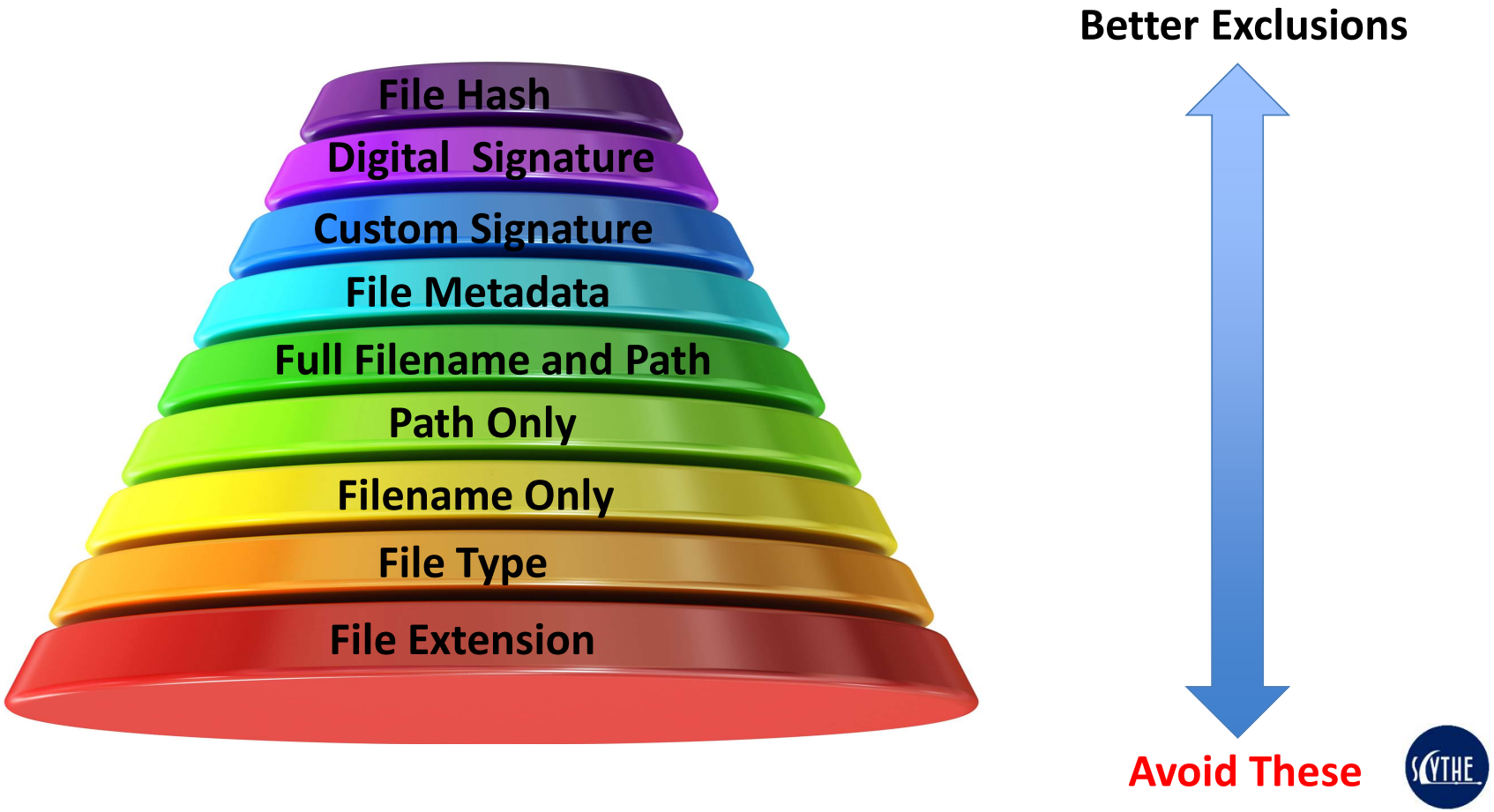
- In v1.0 of the EER, exclusions in order of preference are:
  - File Hash
  - Digital Signature
  - Custom Signature Match/Yara Rule
  - File Metadata
  - Full Filename + Path
  - Path Only
  - Filename only
  - File type (inspection of at least the file header)
  - File extension



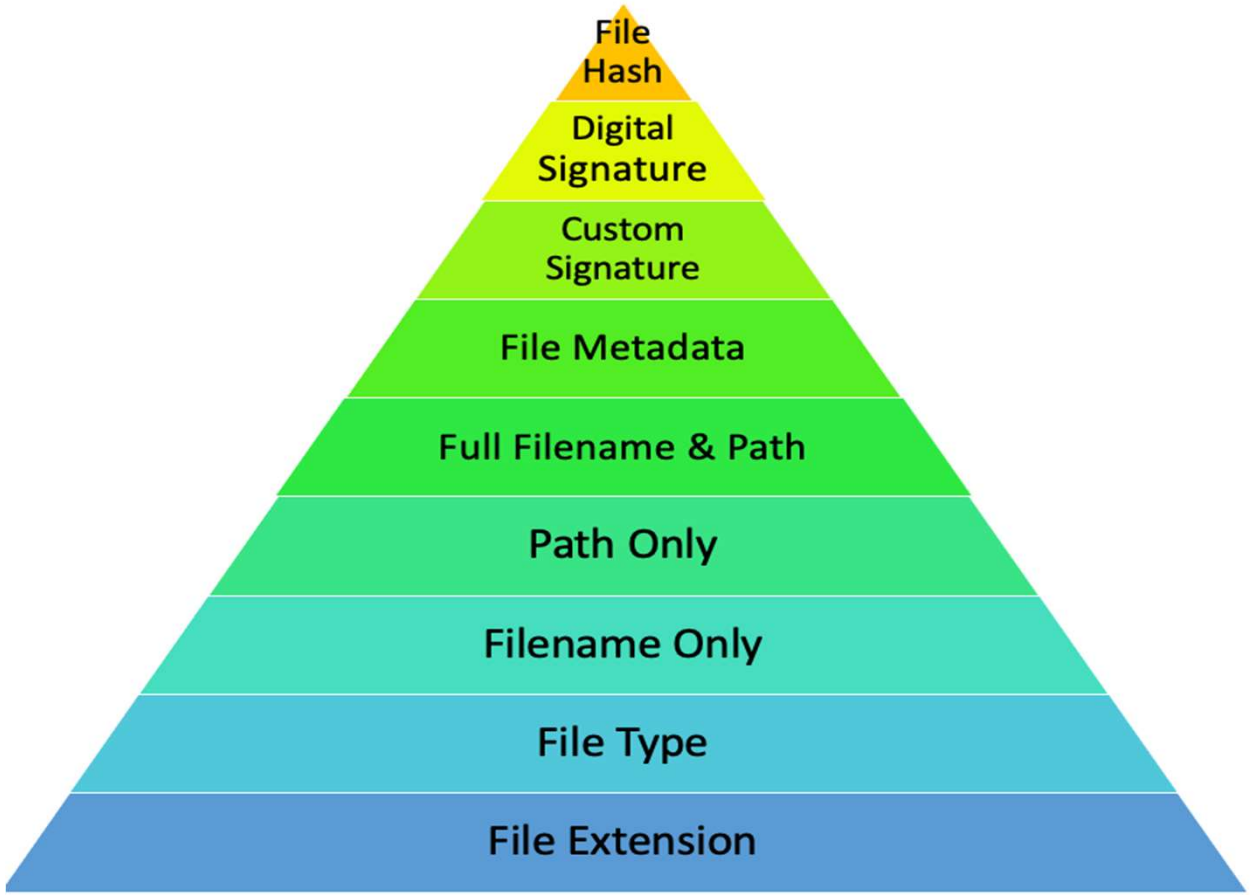
EER Order Of Exclusions Diagram



EER Order Of Exclusions Diagram (Not An Equilateral)



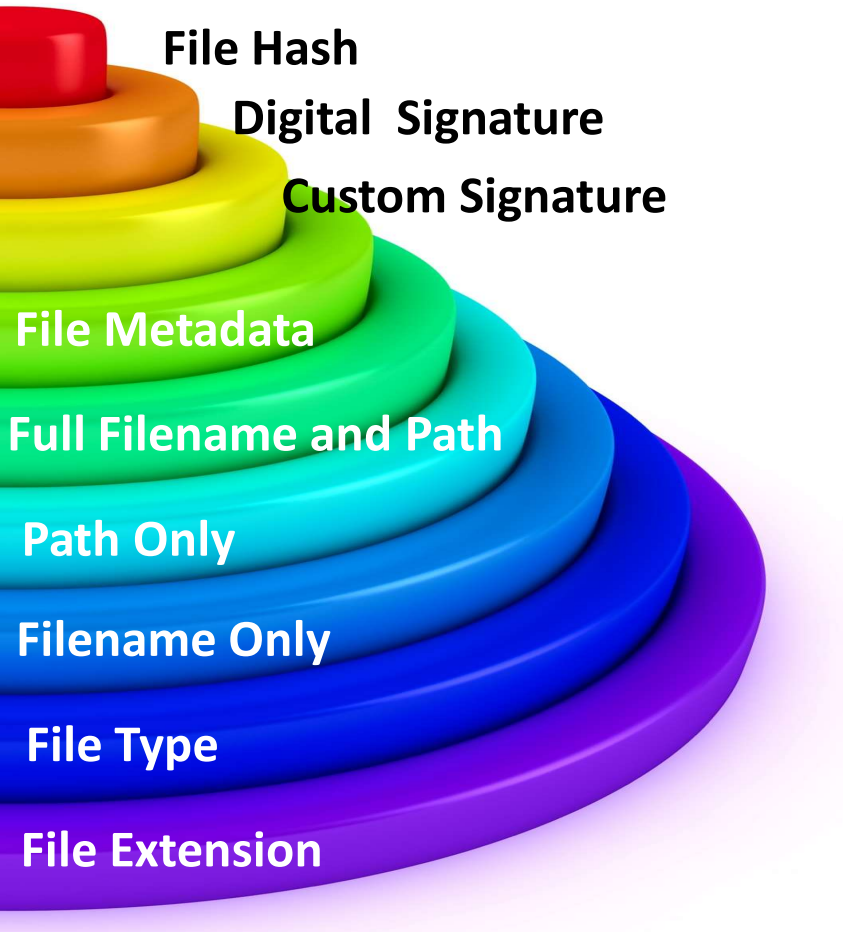
EER Order Of Exclusions Diagram (Wrong Colors)



Better Exclusions

Avoid These

EER Order Of Exclusions Diagram (Tower of Hanoi)

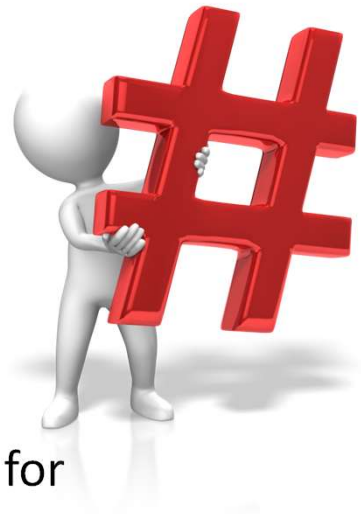


Better Exclusions

Avoid These

EER – File Hash

- File hashes are the most specific exclusion possible and should always be preferred when possible
- Benefits:
  - Very low chance of bypass by threat actors (near zero when modern hashing algorithms like SHA256 are used)
- Drawbacks:
  - Exclusion must be updated with each new software update
- Notes:
  - Fuzzy hashes, while useful for offline analysis, are not generally useful for real time detections and generally not supported for exclusions
  - Avoid the temptation to engage in "hash hunting" for detections



EER – Digital Signature

- Digital signatures offer high fidelity exclusions
- Benefits:
  - Mathematically validated exclusion that is more flexible than hashes
- Drawbacks:
  - Security controls vary wildly in how they validate digital signatures
  - Not all security controls support exclusions via digital signatures
- Notes:
  - Test carefully before using digital signature exclusions, particularly when threat modeling against nation-state actors

EER – Custom Signature Match

- Custom signature matches (e.g. YARA) are far more flexible than hashes or digital signatures
- Benefits:
  - Not dependent on excluded software being digitally signed
- Drawbacks:
  - Complex rules may result in high resource use on endpoints
  - Many teams lack the ability to write good signatures
  - Poorly written signatures may be the least secure exclusion
  - Many endpoint security controls do not support this exclusion

EER – File Metadata

- File metadata is fragile and easily tampered with, but offers additional selectors for exclusions
- Benefits:
  - More secure than simple file/path name
  - Requires significant attention to detail by threat actors for bypass
- Drawbacks:
  - Trivially bypassed by sophisticated threat actors
  - Few endpoint controls support metadata exclusions
- Notes:
  - Use multiple metadata fields or combine with other data elements

EER – Full Filename and Path

- Exclusions with full filename and path are really a combination of two other exclusions (filename and pathname)
- Benefits:
  - Combination of elements is harder to bypass than either alone
  - Prevents some attacks like DLL sideloading that would lead to bypass with filename alone

- Drawbacks:
  - Weak filesystem permissions or existing admin permissions renders this exclusion relatively useless

EER – File Path

- Exclusions that rely only on file path should be a last resort in almost every case since the potential for abuse is so high
  - The path should be as specific as possible
  - Avoid world-writeable paths and implement compensating controls
- Benefits:
  - Support for pesky applications that frequently update but are not digitally signed
- Drawbacks:
  - Trivially bypassed by competent threat actors, who know where you struggle with path exclusions and will capitalize on this knowledge

EER – File Name

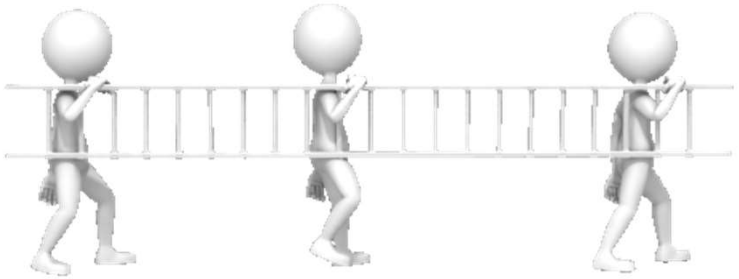
- Exclusions that rely only on file name should be avoided whenever possible
  - Using this exclusion is usually a sign of poor detection engineering or significant tool limitations
- Benefits:
  - So easy, a caveman can do it…
- Drawbacks:
  - So easy to bypass, a caveman can do it…
  - This is svchost.exe from the 2000's all over again

EER – File Type

- Exclusions that rely on the file type (as determined by at least the file header) are an option for some platforms
- Benefits:
  - More robust than file extensions since some internal consistency check takes place
  - Primary use case is when a particular file (e.g. AV definition file) is consistently generating false positives
- Drawbacks:
  - Very few file types are completely good or bad

EER – File Extension

- Exclusions that rely on the file extension are a last resort for the vast majority of use cases
- Benefits:
  - May be useful when a known-benign file type that shares headers with other dangerous file types
    - E.g. DICOM vs MS Office



- Drawbacks:
  - Very few file extensions are completely good or bad
  - Shell handlers care about the file extension, but practically nothing else does (rendering this extremely ineffective)

Exclusion Gotchas

Exclusion Gotcha: Digital Signatures

- Security controls vary wildly in how they validate signatures
- Validation failures:
  - Accepts any signature (even self-signed)
  - Only looks for a trusted digital signature
  - Does not validate the signing certificate chain
  - Inspects the subject name field without validating the signature
  - Honors signatures from revoked (stolen) certificates
  - Accept signatures with known weak algorithms like MD5 (Flame)

Exclusion Gotcha: File Paths

- When building a path-based exclusion, recognize that threat actors know where your pain points are
- A few frequent fliers:
  - Teams
  - GotoMeeting
  - Webex Meetings

- Be judicious about creating exclusions for too many paths
  - Where possible, supplement path-based exclusions with additional detections to avoid coverage gaps

Closing Thoughts

Closing Thoughts

- Exclusions don't need to be a dirty word in detection engineering
  - We just need to do it right
- The cost of bad exclusions is high
  - Too damn high to get it wrong
- Even if all this seems obvious to you, the EER model should still help
  - Saying "we followed the model" vs "I know this is best" often produces divergent outcomes

Jake Williams

@MalwareJake

SCYTHE

@scythe_io

SCYTHE