Building a Budget-Friendly Malware Analysis Pipeline

Maximizing Efficiency without Breaking the Bank

./RE

Joshua Reynolds

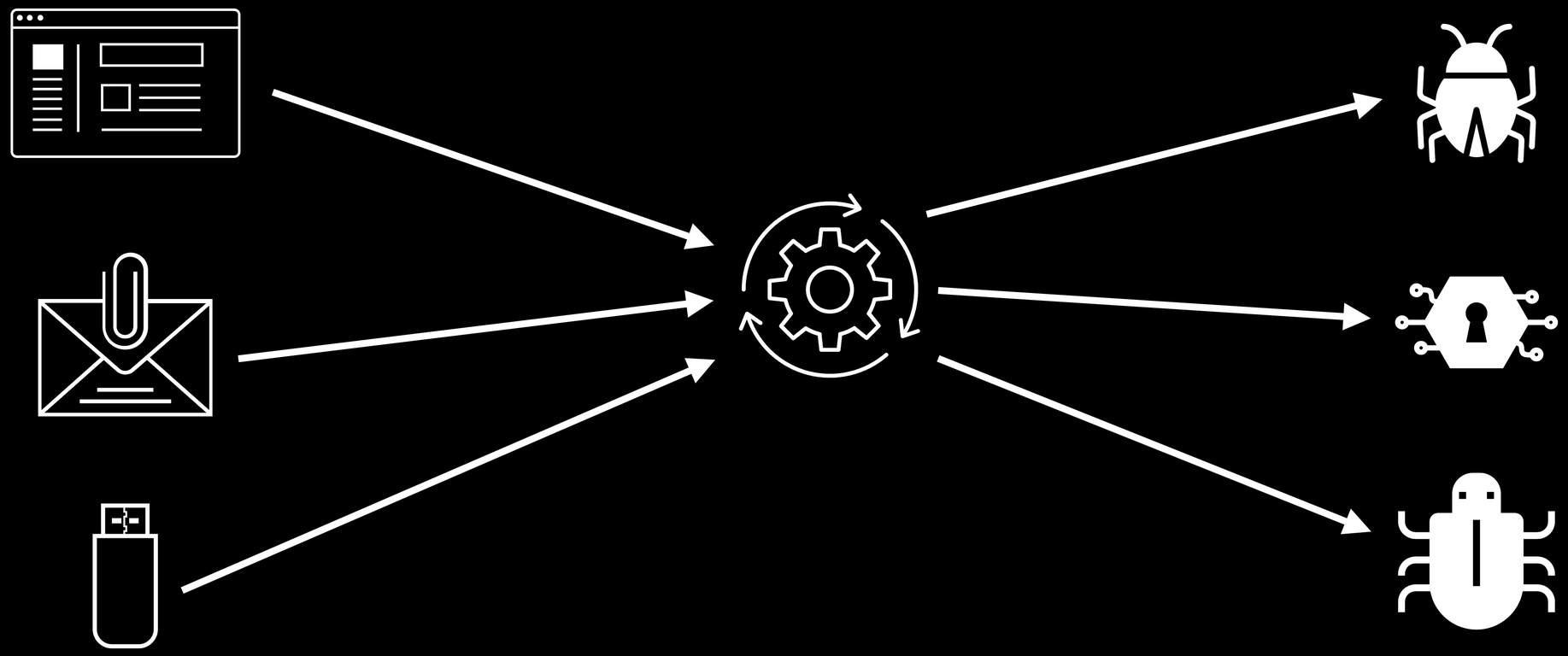
Founder, Invoke RE

- Over ten years of security-related experience working for industry leading companies
- Spoken at RSA, DEF CON and Virus Bulletin on ransomware and malicious document analysis
- Co-developed malware analysis course taught at Southern Alberta Institute of Technology
- @jershmagersh / @InvokeReversing
- info@invokere.com

Malware Analysis

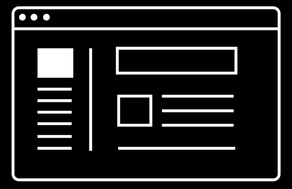
./RE

What is Malware Analysis?

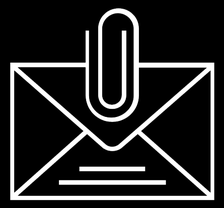


./RE

Example Infection Vector Scenarios

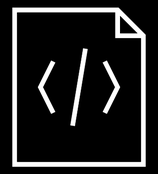


Malvertising leads to download of fake update



Email document to user containing malicious macros

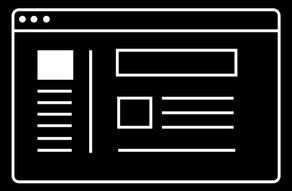
USB emulates keystrokes to execute PS payload



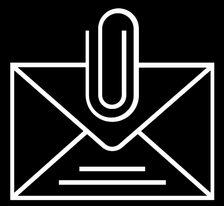
User downloads a ZIP containing a malicious script

./RE

Example Collection Scenarios

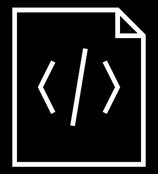


Fake update binary is collected using HTTP proxy



Malicious document download blocked at IPS

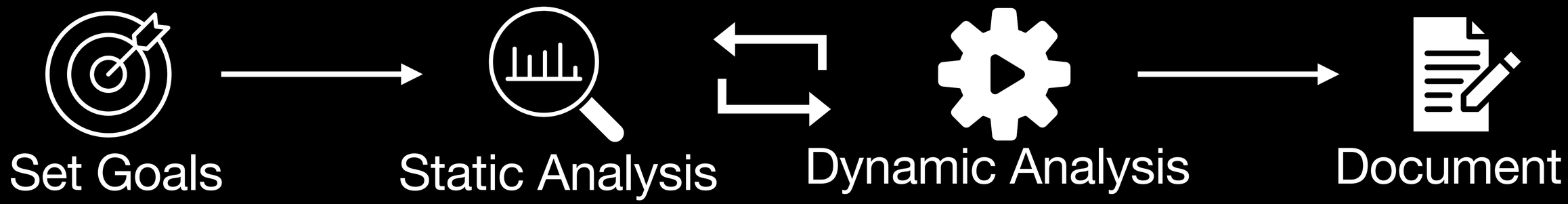
PowerShell launched by USB blocked by EDR



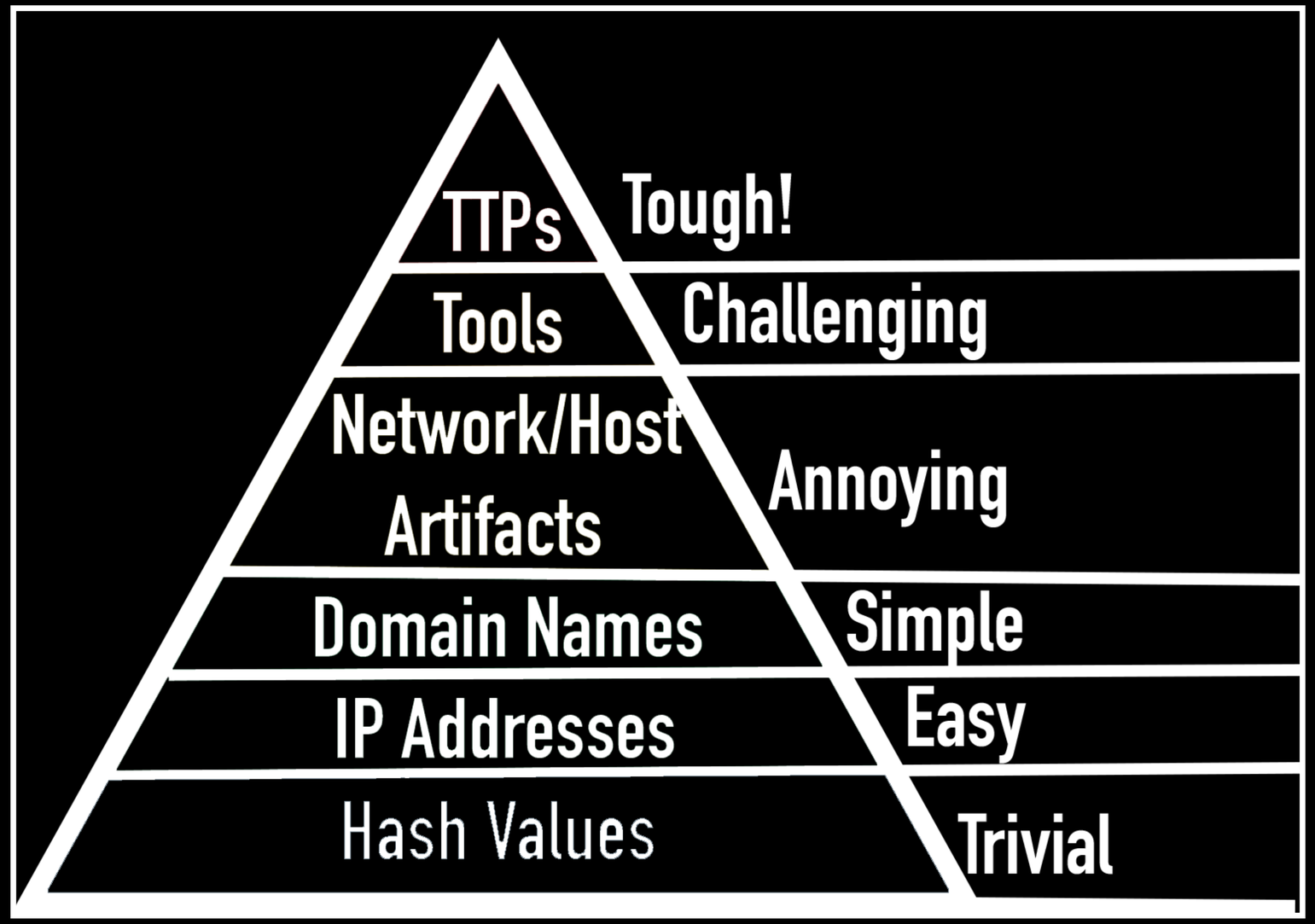
Malicious script download detected by IDS

./RE

Example Malware Analysis Workflow

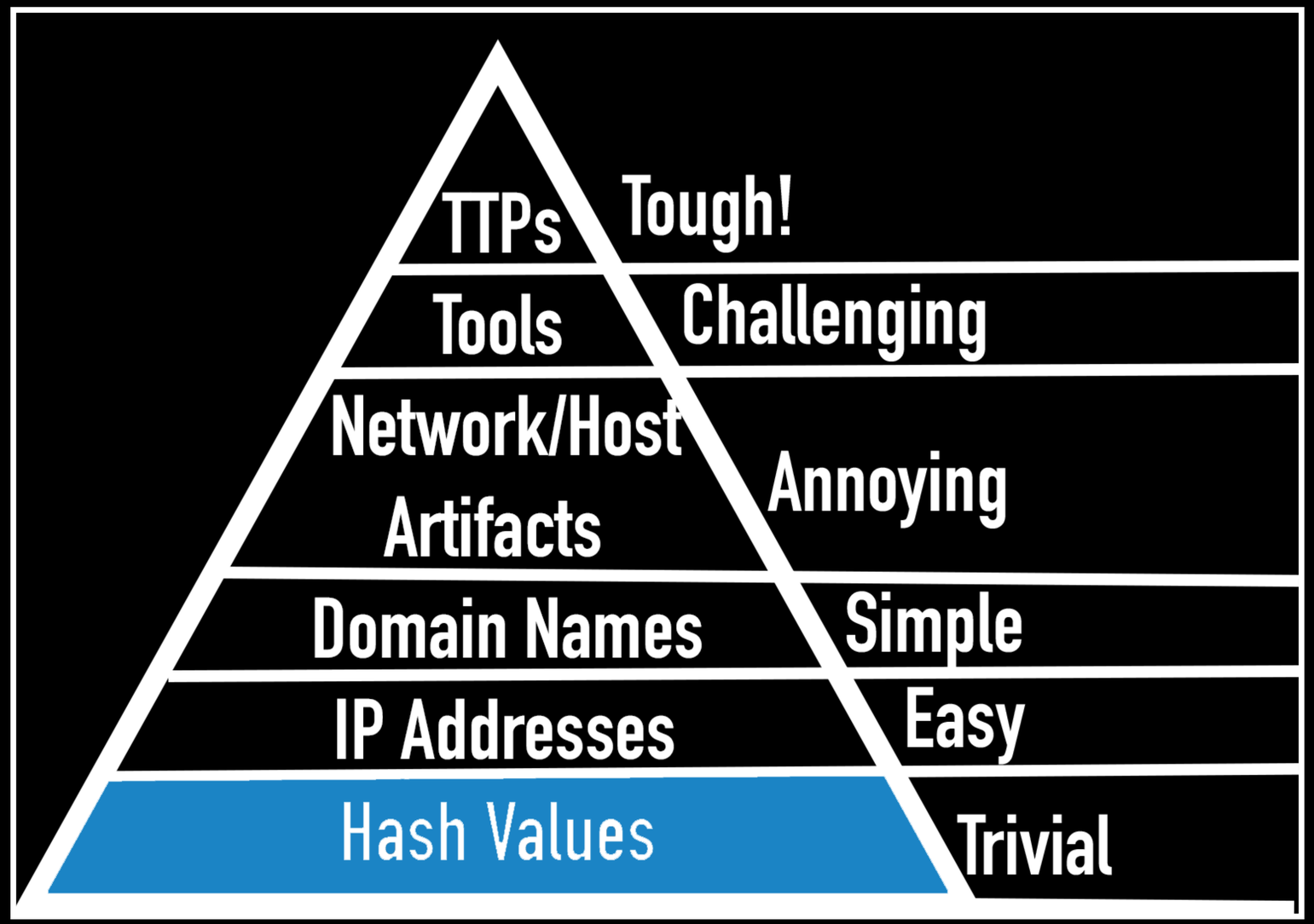


The Pyramid of Pain



./RE

The Pyramid of Pain

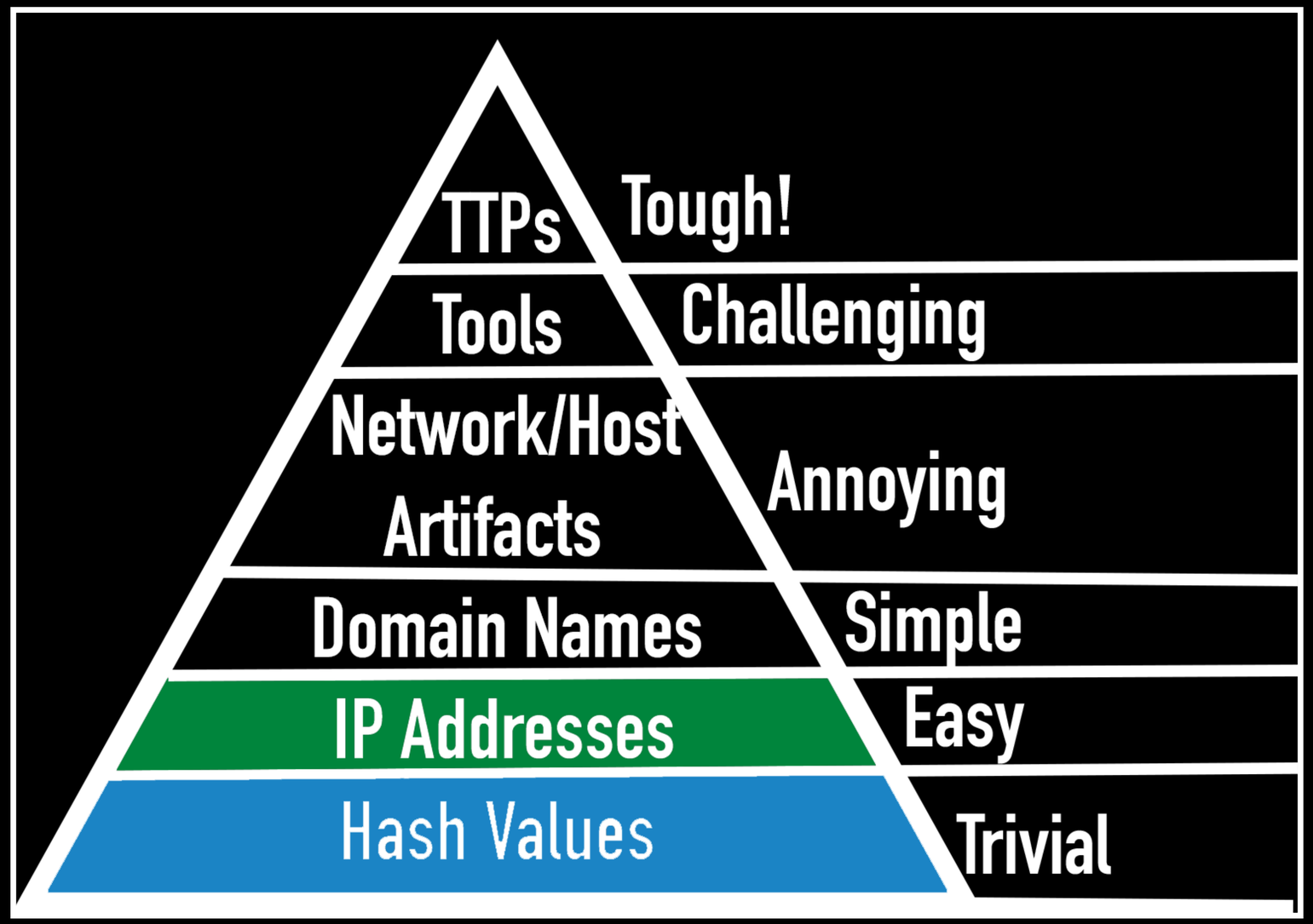


./RE

File Hash Example

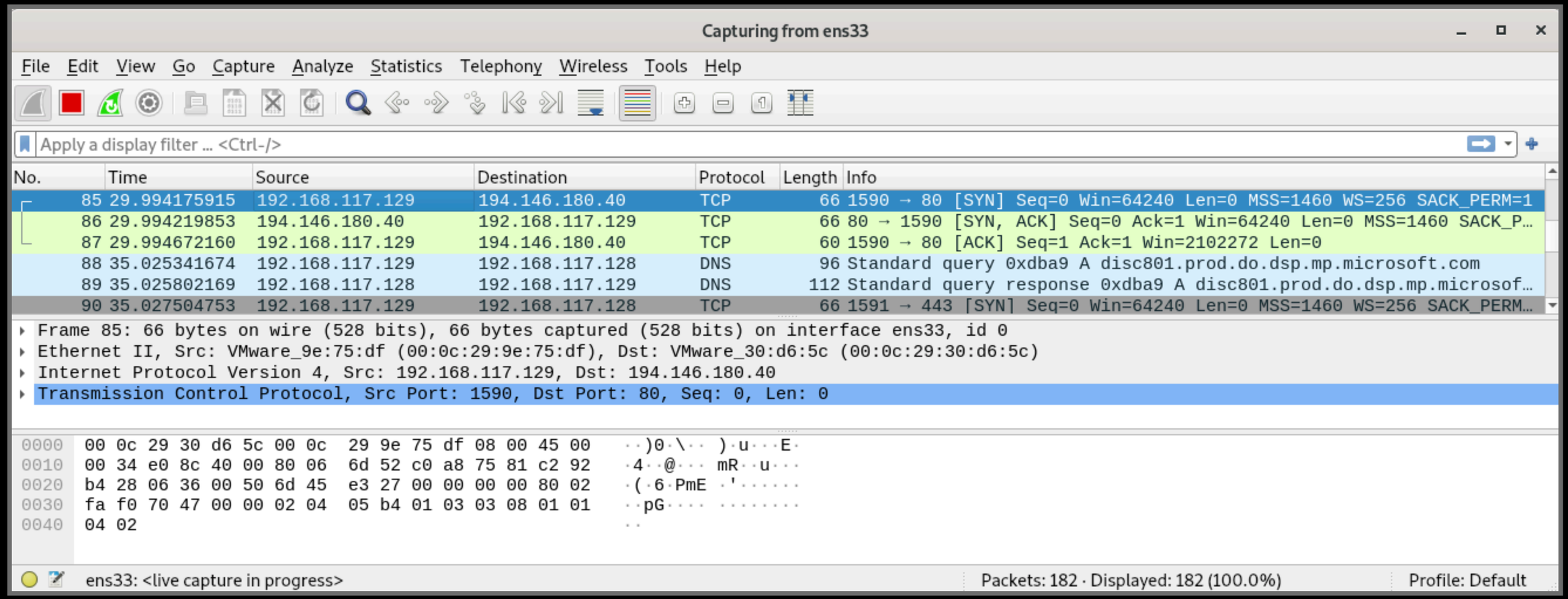
```
PS C:\windows\system32> Get-FileHash calc.exe

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          3091E2ABFB55D05D6284B6C4B058B62C8C28AFC1D883B699E9A2B5482EC6FD51       C:\windows\system32\calc.exe
```

The Pyramid of Pain



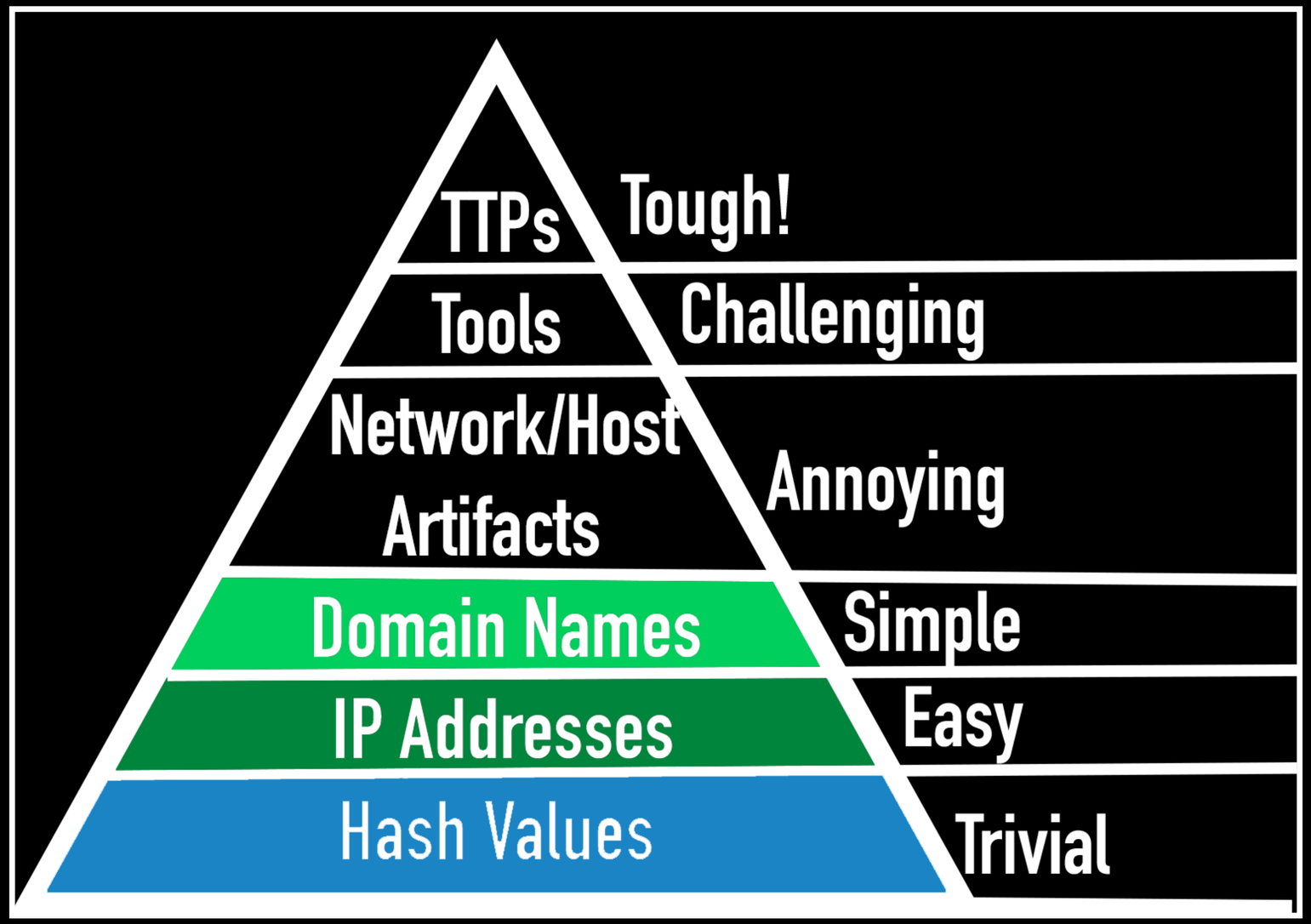
./RE

IP Address Example



./RE

The Pyramid of Pain

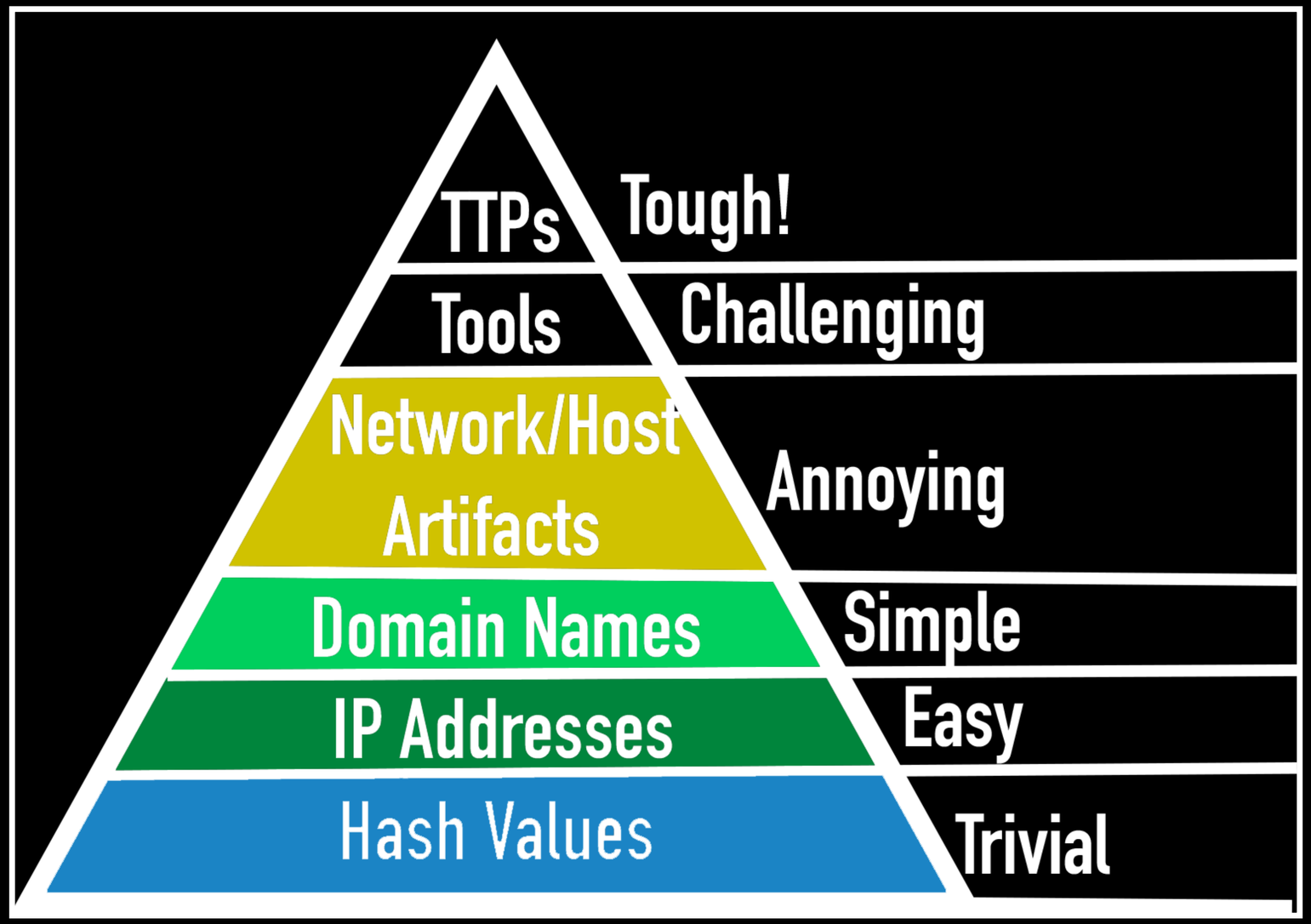


./RE

DNS Capture Example

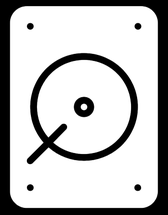
```
remnux@remnux:~$ fakedns 2>&1 | grep -v "microsoft.com ->"
fakedns[INFO]: dom.query. 60 IN A 192.168.117.128
fakedns[INFO]: Response: plumberspro.us -> 192.168.117.128
fakedns[INFO]: Response: ntp.ubuntu.com -> 192.168.117.128
fakedns[INFO]: Response: ntp.ubuntu.com -> 192.168.117.128
fakedns[INFO]: Response: godstar.hopto.org -> 192.168.117.128
fakedns[INFO]: Response: grigori.ddns.net -> 192.168.117.128
fakedns[INFO]: Response: checkip.dyndns.org -> 192.168.117.128
fakedns[INFO]: Response: plumberspro.us -> 192.168.117.128
fakedns[INFO]: Response: godstar.hopto.org -> 192.168.117.128
fakedns[INFO]: Response: grigori.ddns.net -> 192.168.117.128
```

The Pyramid of Pain



./RE

What are host and network artifacts?



Files residing on the filesystem

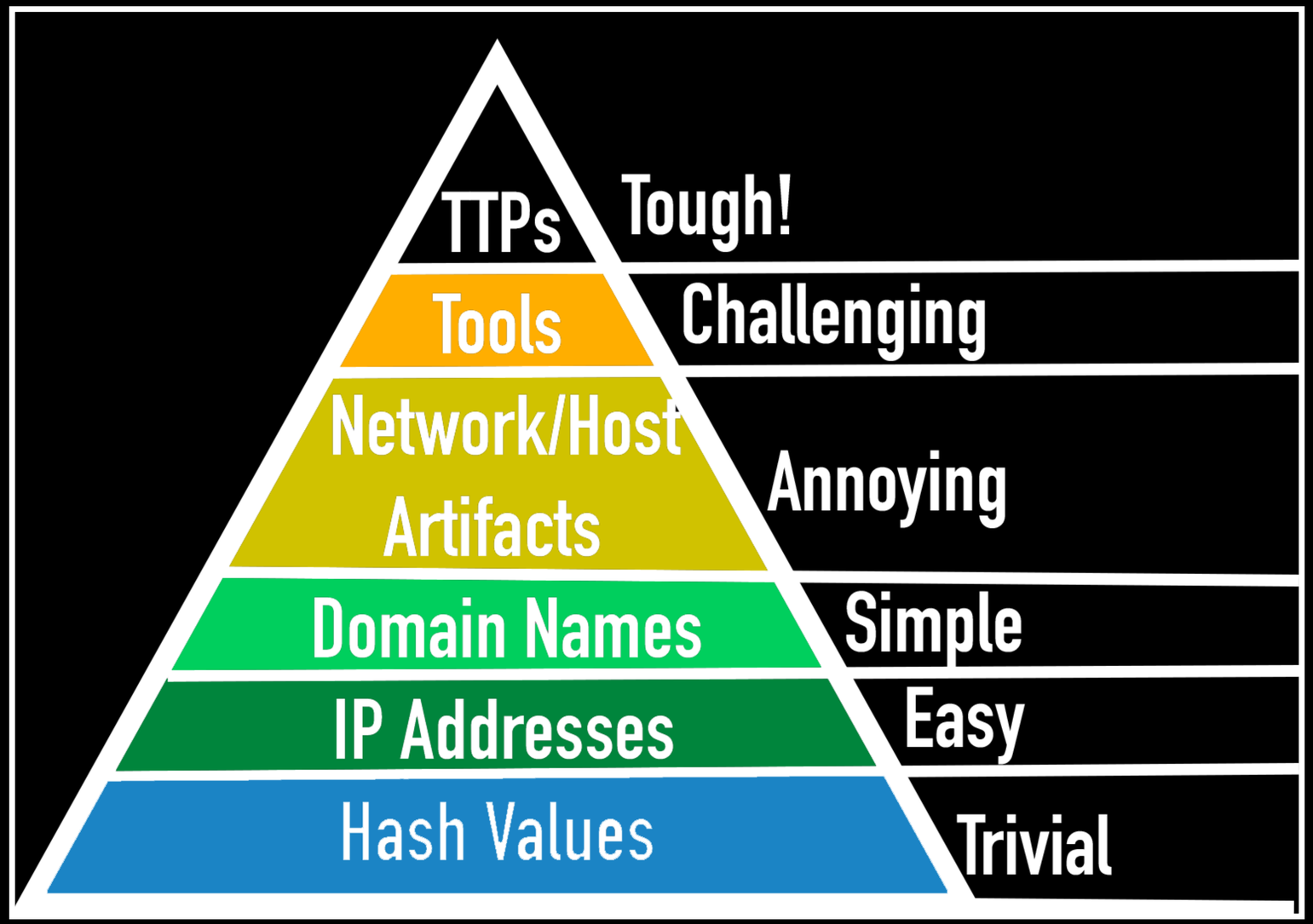
Registry hives, keys and values

Network characteristics

Host Artifact Example

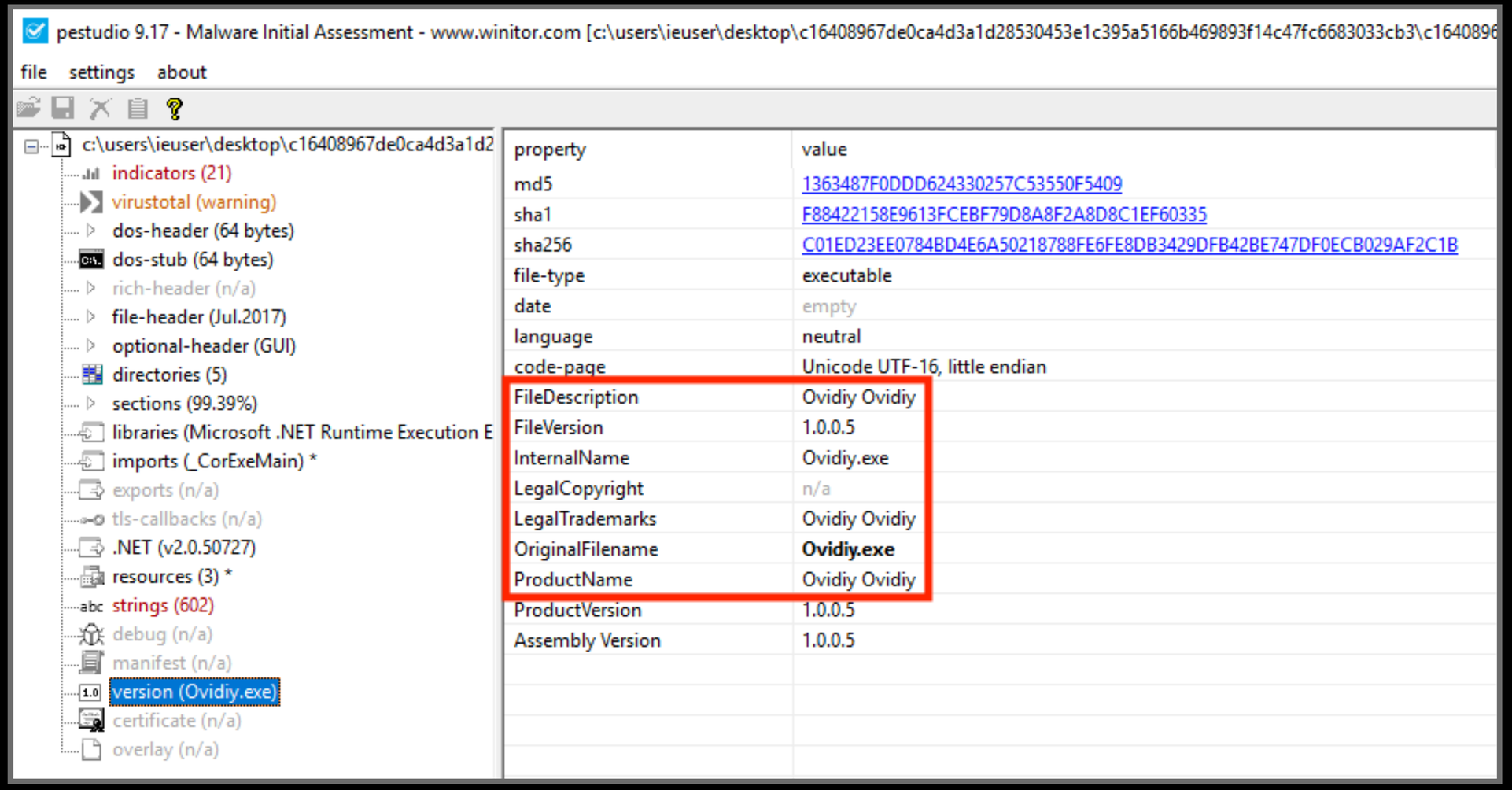
```
.text:012D7F29                 push    eax             ; lpdwDisposition
.text:012D7F2A                 lea     eax, [ebp+phkResult]
.text:012D7F2D                 push    eax             ; phkResult
.text:012D7F2E                 push    0               ; lpSecurityAttributes
.text:012D7F30                 push    20006h          ; samDesired
.text:012D7F35                 push    0               ; dwOptions
.text:012D7F37                 push    0               ; lpClass
.text:012D7F39                 push    0               ; Reserved
.text:012D7F3B                 push    offset aSoftwareMicros_0 ; "SOFTWARE\\Microsoft\\Windows\\CurrentVe"...
.text:012D7F40                 push    HKEY_LOCAL_MACHINE ; hKey
.text:012D7F45                 call    ds:RegCreateKeyExA
.text:012D7F4B                 test    eax, eax
.text:012D7F4D                 jnz     short loc_12D7FC6
.text:012D7F4F                 mov     ecx, esi
.text:012D7F51                 lea     edx, [ecx+1]
.text:012D7F54
.text:012D7F54 loc_12D7F54:                            ; CODE XREF: sub_12D7EF0+69↓j
.text:012D7F54                 mov     al, [ecx]
.text:012D7F56                 inc     ecx
.text:012D7F57                 test    al, al
.text:012D7F59                 jnz     short loc_12D7F54
.text:012D7F5B                 sub     ecx, edx
.text:012D7F5D                 lea     eax, [ecx+1]
.text:012D7F60                 push    eax             ; cbData
.text:012D7F61                 push    esi             ; lpData
.text:012D7F62                 mov     esi, ds:RegSetValueExA
.text:012D7F68                 push    1               ; dwType
.text:012D7F6A                 push    0               ; Reserved
.text:012D7F6C                 push    offset aSm1     ; "SM_1"
.text:012D7F71                 push    [ebp+phkResult] ; hKey
.text:012D7F74                 call    esi ; RegSetValueExA
.text:012D7F76                 mov     ecx, ebx

; CHAR aSoftwareMicros_0[]
aSoftwareMicros_0 db 'SOFTWARE\Microsoft\Windows\CurrentVersion',0
                                                       ; DATA XREF: sub_12D7EF0+4B↑o
                                                       ; sub_12D7FE0+59↑o ...
```

The Pyramid of Pain



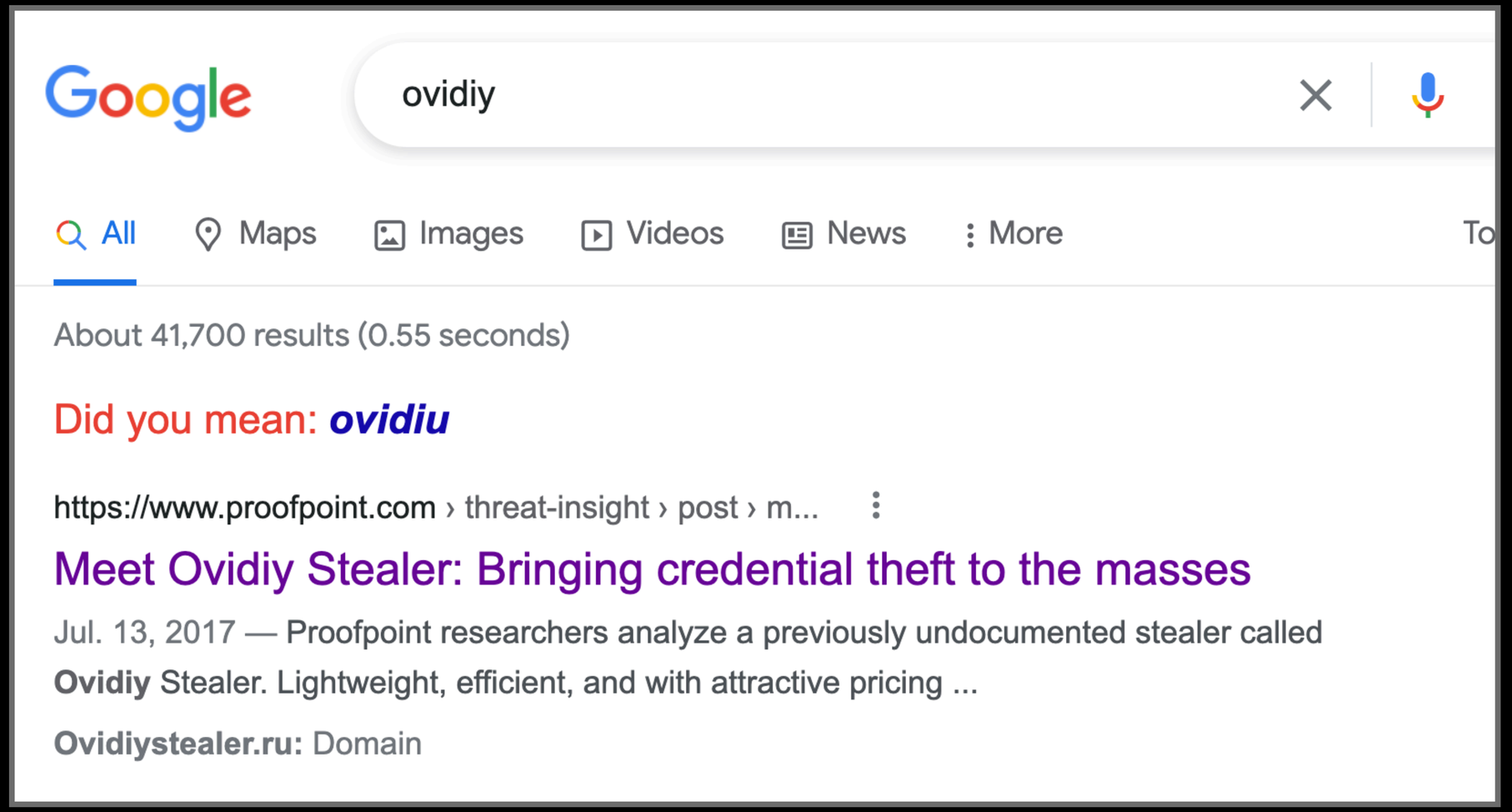
./RE

Tool Identification Example



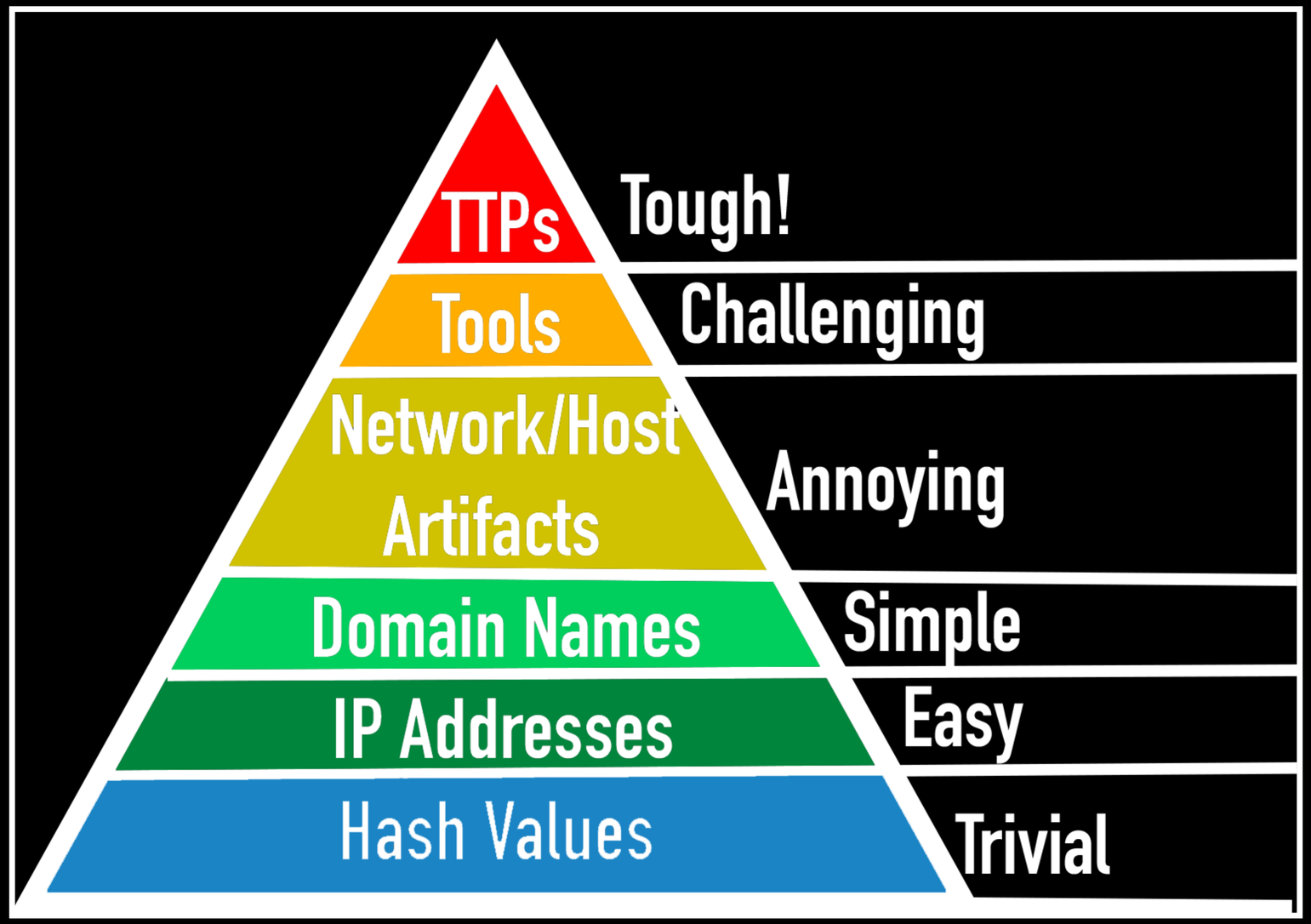
./RE

Tool Identification Example



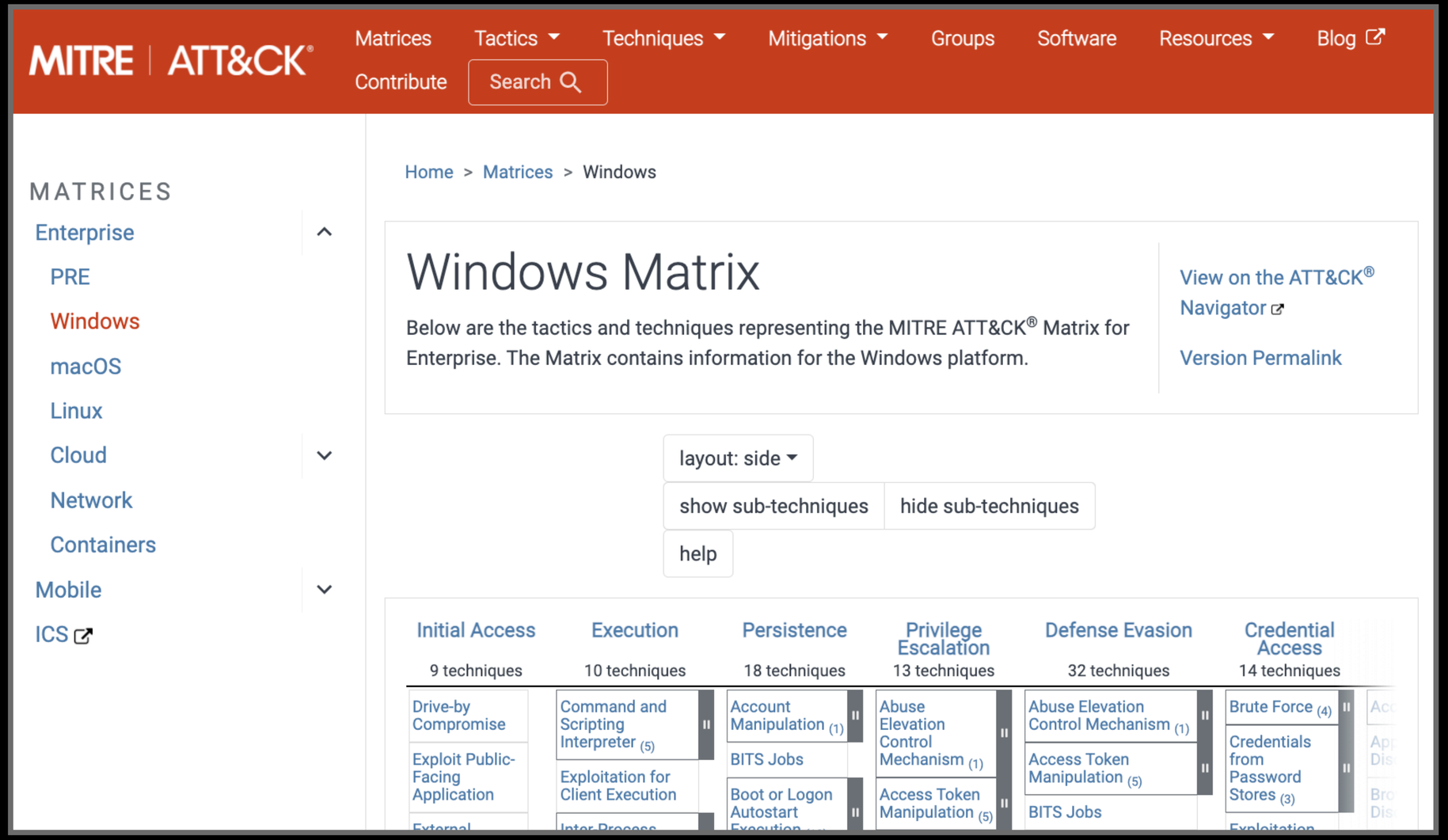
./RE

The Pyramid of Pain



./RE

ATT&CK Framework

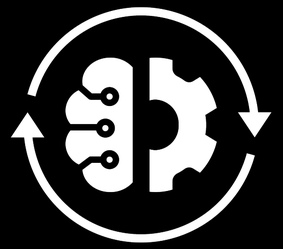


./RE

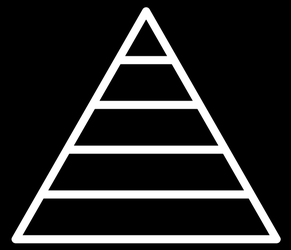
Automating Malware Analysis

./RE

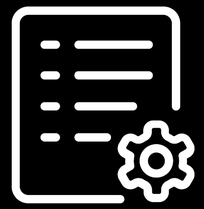
Automating Malware Analysis



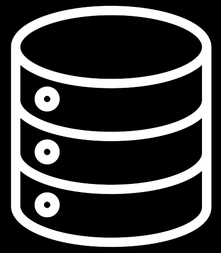
Goal of performing analysis automatically



Extract all levels of pyramid of pain



Extract malware configurations

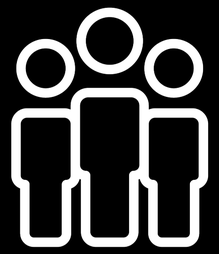


Store all information for future analysis

Considerations

Available budget - can someone do this for me?

Privacy - can we upload to cloud services?



Talent - Do we have in-house talent?

Assemblyline

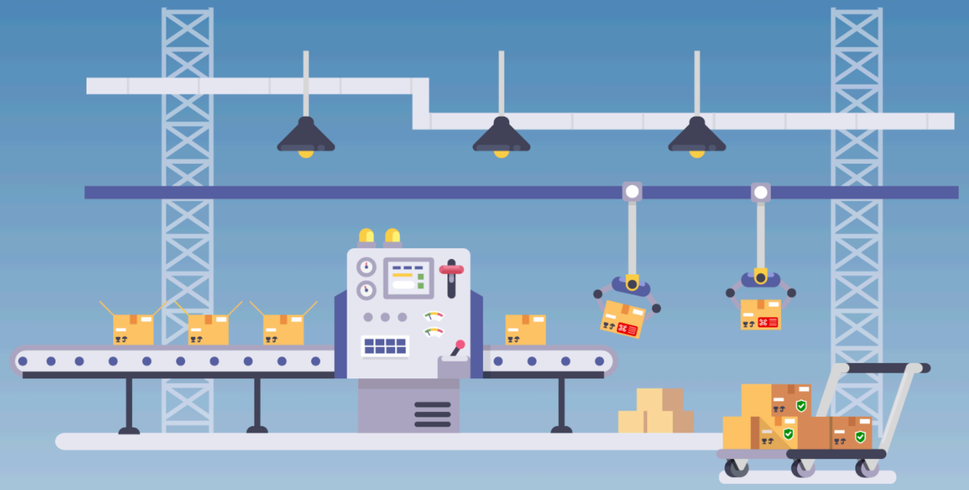
Assemblyline 4

A scalable file triage and malware analysis system integrating the cyber security community's best tools.

How it works

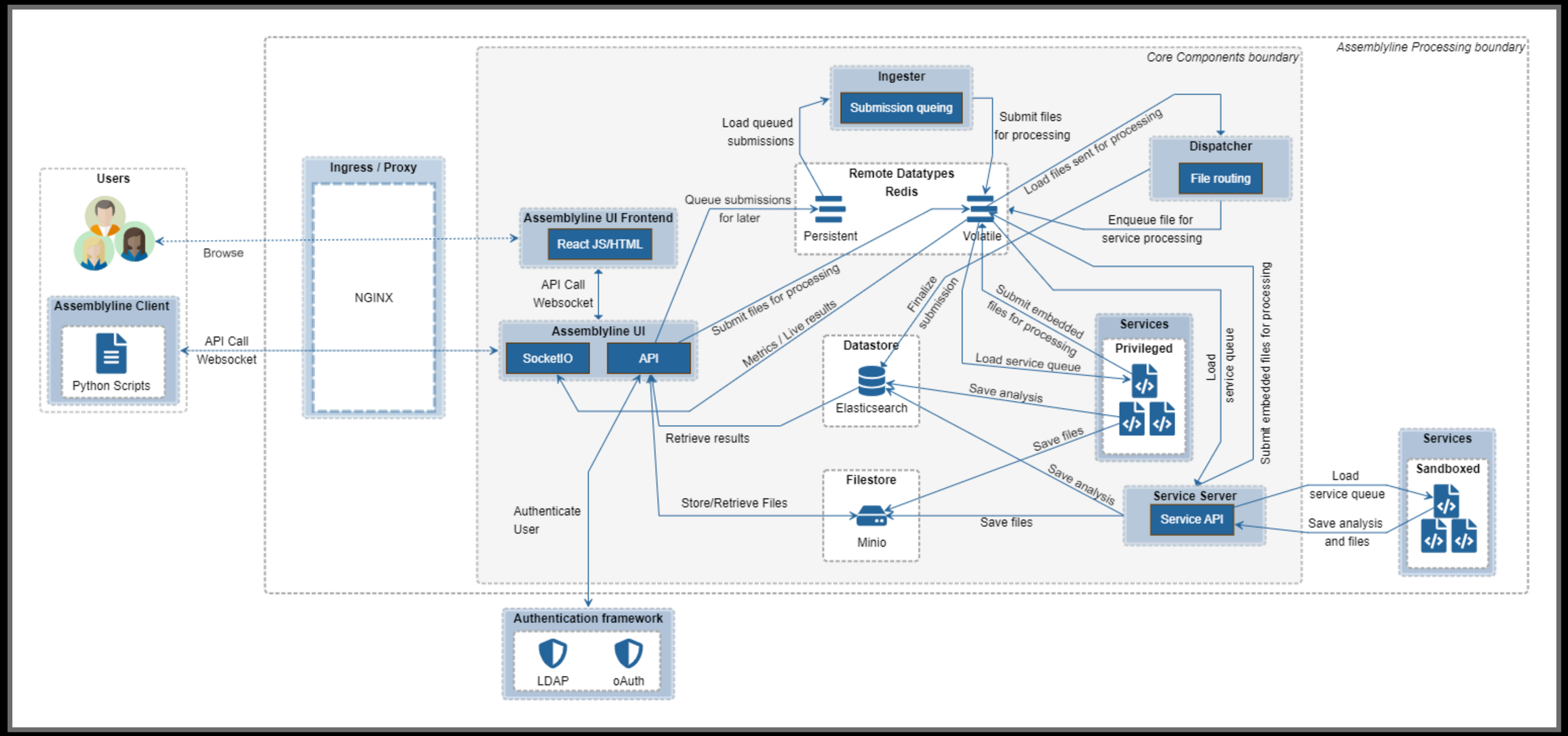
View source code

Join our community



./RE

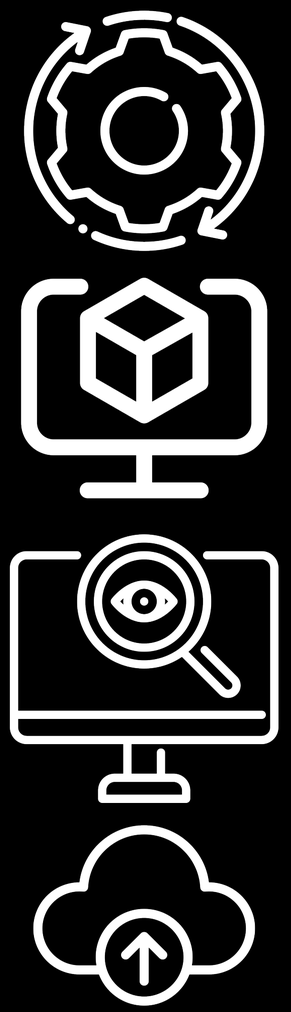
Assemblyline



Source: https://cybercentrecanada.github.io/assemblyline4_docs/overview/architecture/

./RE

What is a sandbox?



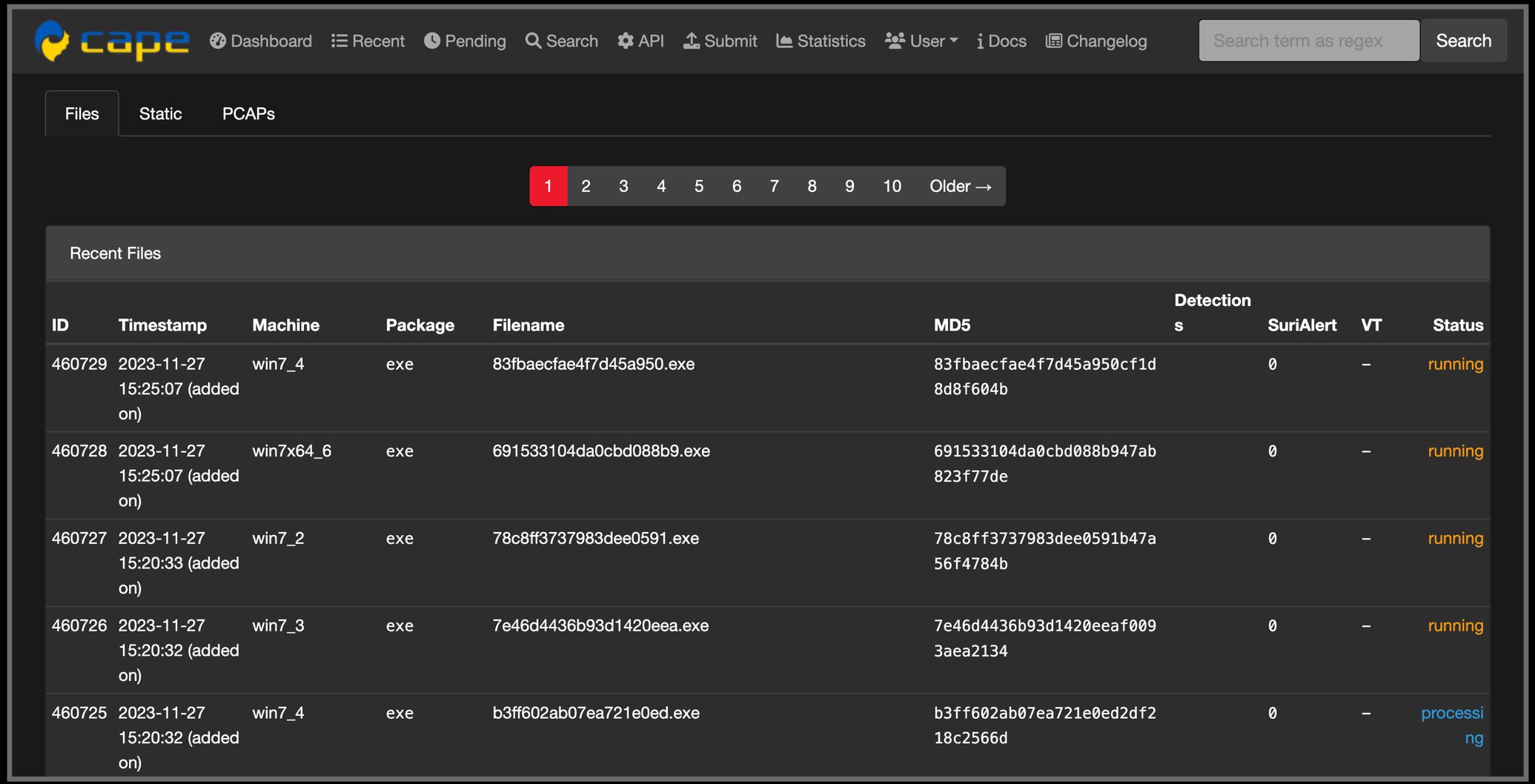
Performs dynamic analysis of samples

Spins up a virtual machine and executes sample

Monitors activity during sample execution

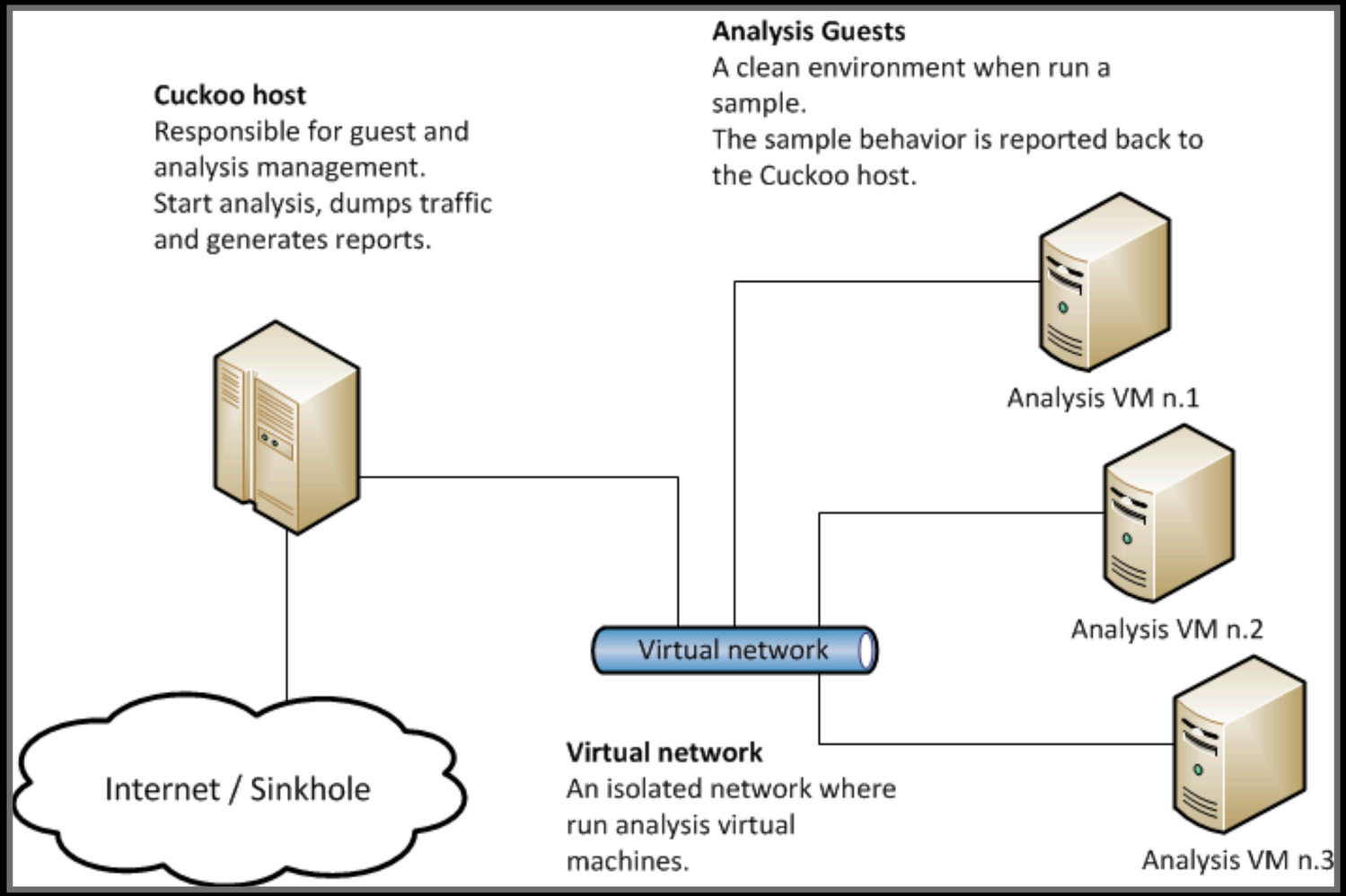
Uploads sandbox outputs to server for analysis

CapeV2



./RE

CapeV2



Source: https://capev2.readthedocs.io/en/latest/introduction/what.html#architecture

./RE

Limitations

Automation can result in false positives

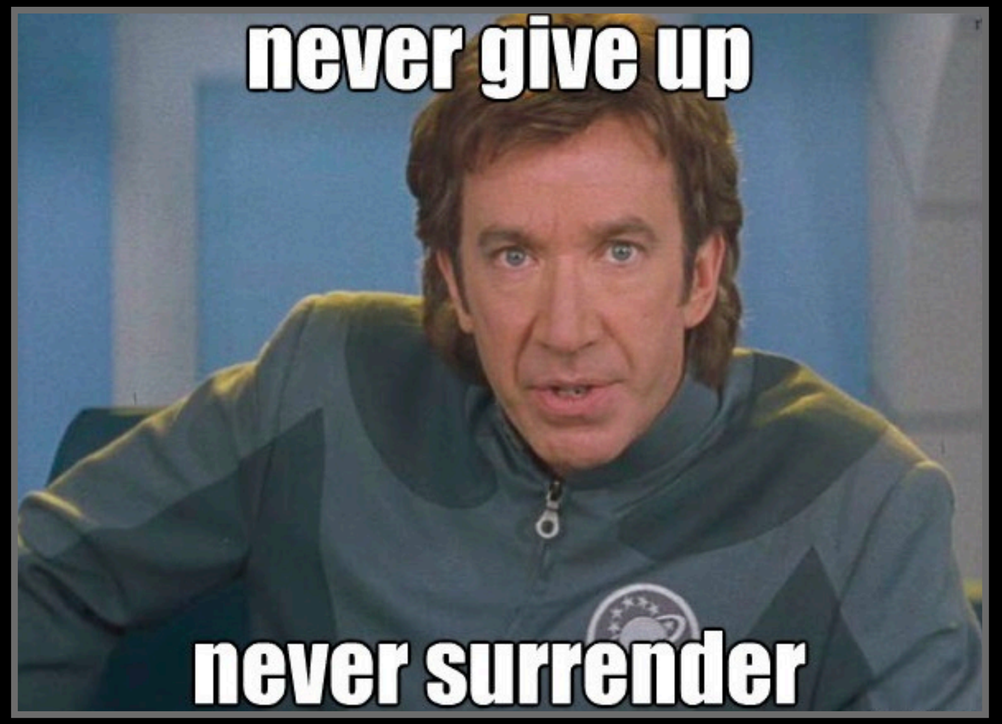
Automation can result in false negatives

Automation can be evaded

Demo

Further Reading & References

- https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html
- https://detect-respond.blogspot.com/2022/04/stop-using-hashes-for-detection-and.html
- https://cybercentrecanada.github.io/assemblyline4_docs/
- https://github.com/kevoreilly/CAPEv2
- https://capesandbox.com/analysis/



Questions?

Contact
@jershmagersh / @InvokeReversing
info@invokere.com

./RE