Safeguarding Civilization

Mind the Gap, Bro!

Using Network Monitoring to Overcome Host Invisibility

Joe Slowik; Senior Adversary Hunter, Dragos Inc.
jslowik@dragos.com
@jfslowik

DRAGOS

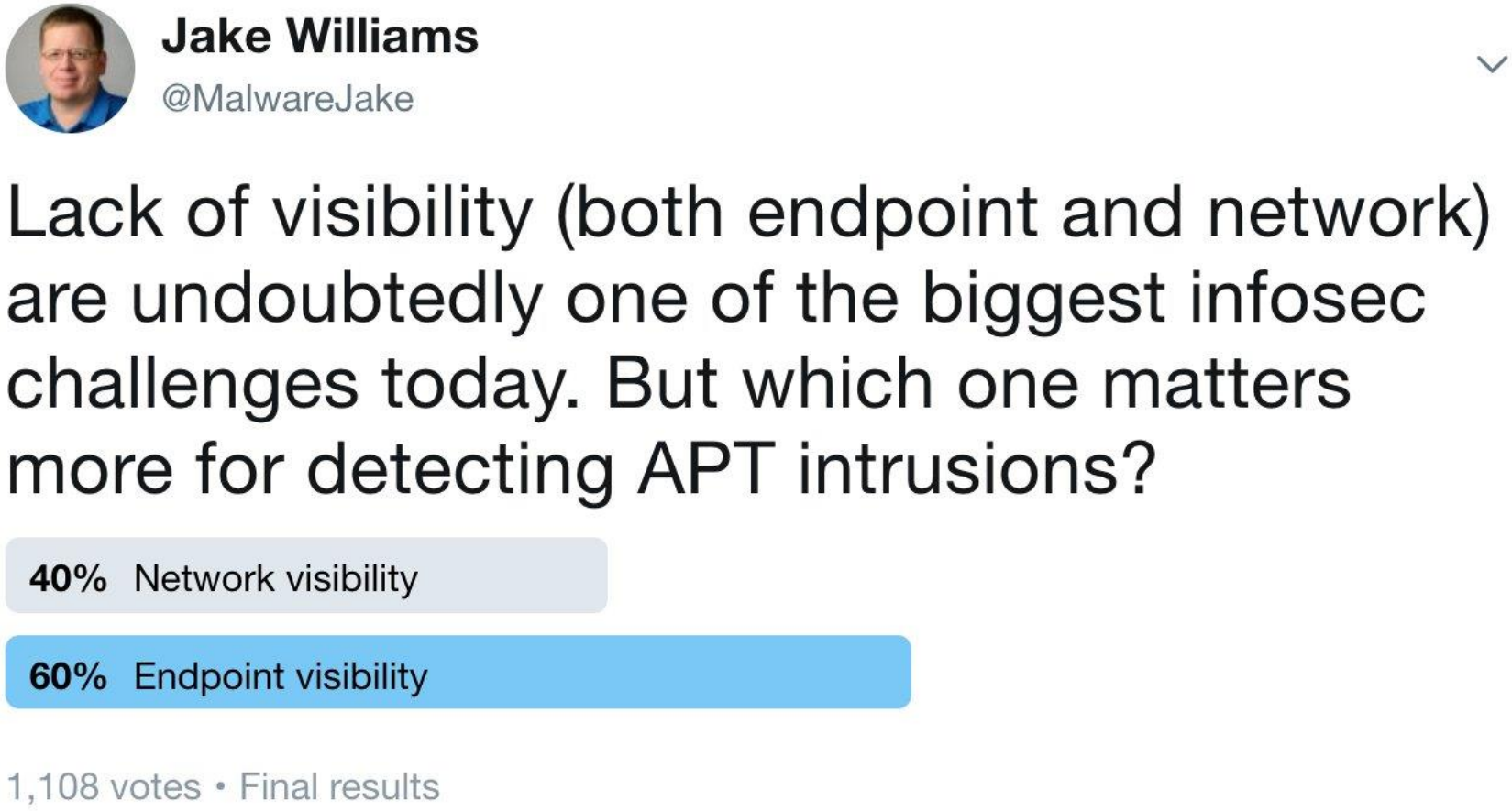
Introduction

- Joe Slowik, Threat Intelligence & Hunter
- Current: Dragos Adversary Hunter
- Previous:
  - Los Alamos National Lab: IR Lead
  - US Navy: Information Warfare Officer
  - University of Chicago: Philosophy Drop-Out

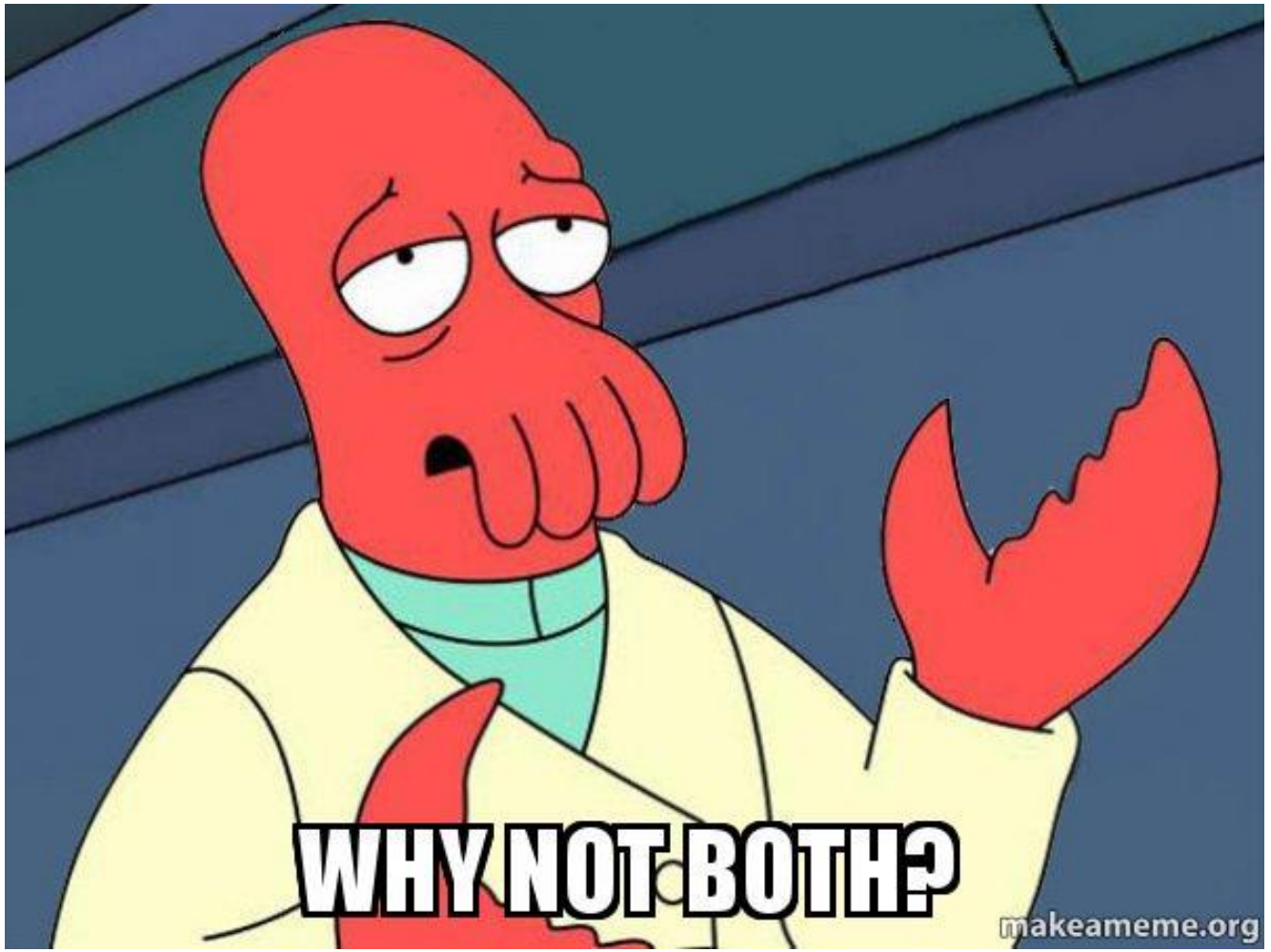
Agenda

- Network vs. Host Visibility
- Network to Capture Host
  - Bro
  - YARA
- Use-Cases & Examples
- Limitations

The Challenge



The Ideal Answer



The Monitoring Landscape

- Host-based monitoring is vital but often less mature
- Network-based monitoring more likely but incomplete
- Best answer is ‘both’ in support of one another

Visibility and Environment Type

- Visibility challenges differ by environment type
- Example: Large Windows Domain vs. ICS Network
- Different challenges – but also opportunities

Network vs. Host

- Host: ‘higher fidelity’, ground truth – but difficult to push out, manage
- Network: easier to implement, more centralized, but leaves out some details

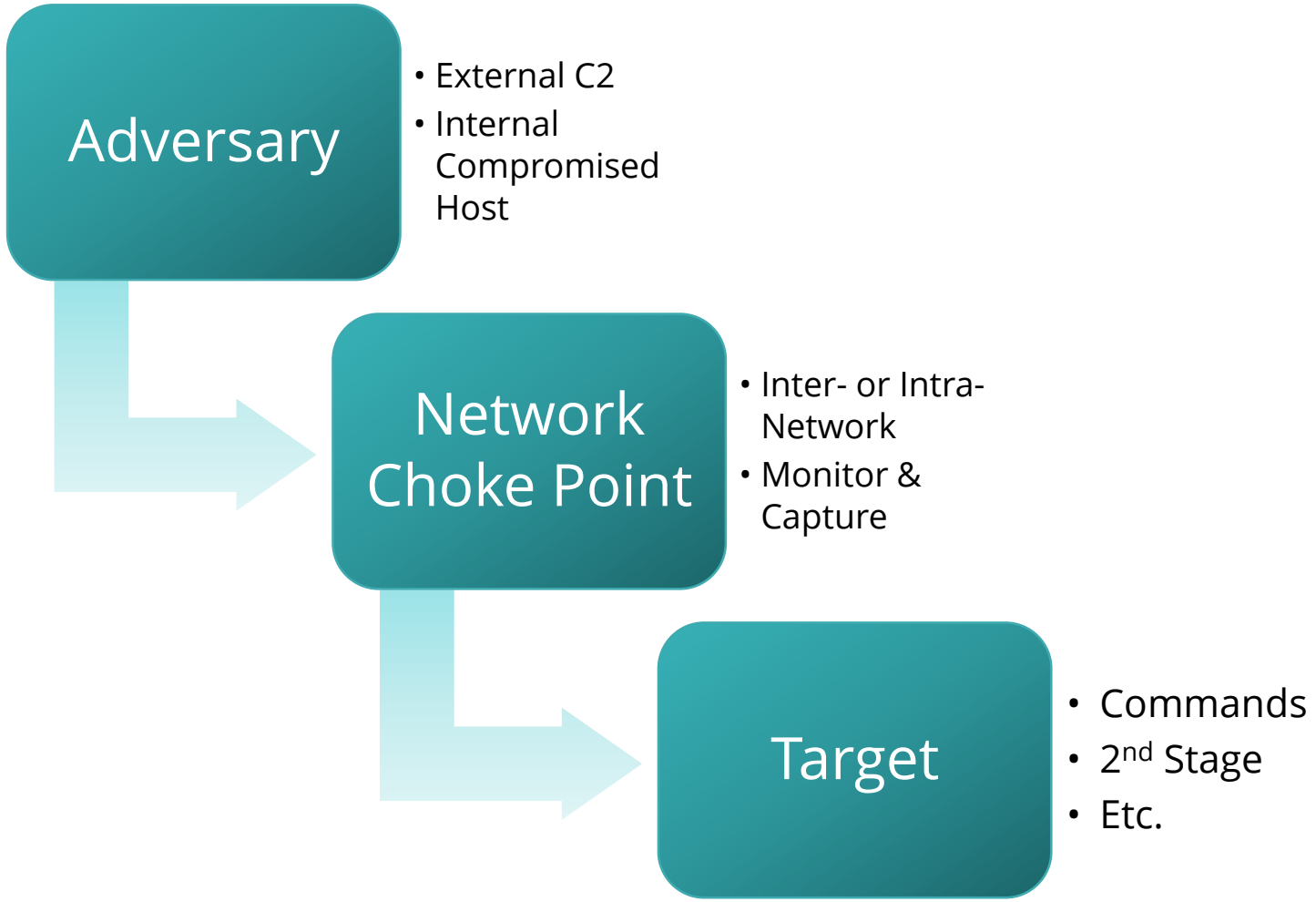
Using Network to Capture Host

- Network visibility can be leveraged to see elements of host activity:
  - Files moving across the wire
  - Commands via visible protocols
- Even if clear-text unavailable, sufficient data can be gleaned to inform investigation

Solution: Leverage Dependencies

- If host is inaccessible, leverage network
- Data, commands, etc. *must* come from somewhere to execute, control, etc.
- Key: identifying and parsing traffic

Solution: Leverage Dependencies



What's Up, Bro?

- Bro = open-source network traffic analyzer
- Enables session-level analysis rather than packet

- Developed at LBNL – w00t DOE
- Continued development adds functionality

Bro for File Carving

- Bro automates file-carving from traffic
  - Better than manually parsing from PCAP
- Applies to various protocols – most significant limitation is encryption
  - *We will come back to this point*

Bro for File Carving

```
##! Extract all files to disk.

@load base/files/extract

event file_new(f: fa_file)
      {
      Files::add_analyzer(f, Files::ANALYZER_EXTRACT);
      }
```

https://github.com/hosom/file-extraction/blob/master/scripts/plugins/extract-all-files.bro

Bro for File Carving, Complex

```
@load base/files/extract
@load base/files/hash

redef FileExtract::prefix = "./";
global test_file_analysis_source: string = "" &redef;
global test_file_analyzers: set[Files::Tag];
global test_get_file_name: function(f: fa_file): string =   function(f: fa_file): string { return ""; }
&redef;
global test_print_file_data_events: bool = F &redef;
global file_count: count = 0;
global file_map: table[string] of count;
function canonical_file_name(f: fa_file): string
          {
          return fmt("file #%d", file_map[f$id]);
          }

event file_chunk(f: fa_file, data: string, off: count)
          {
          if ( test_print_file_data_events )
                    print "file_chunk", canonical_file_name(f), |data|, off, data;
          }

To be Continued!
```

File Carving Advantage

- Simply carving files and checking hashes against ‘dirty lists’ = pointless
- BUT – paired with analysis engine, very valuable:
  - Sandbox
  - YARA
  - Detection Scripts

File Carving Scope

- Pull files from anything Bro has an analyzer for:
  - HTTP
  - SMB
  - FTP
- If Bro can see it, you can grab it

Conceptual Flow

Traffic Captured, Items Carved

Initial Filter, Items of Interest Pass to Analysis Engine

Leverage Tools in Engine to Identify Malicious Activity

Detection Possibilities

- YARA:
  - Malware detection
  - Potential DLP/exfiltration monitoring
- Detection Scripts:
  - Unpack and examine Office Macros
  - PowerShell, WMI, and other scripting language detectors

Yet Another REGEX Alternative

- YARA = awesomesauce
- Flexible, powerful means of analyzing any filetype – strings and binary content

NotPetya Example

```
rule embedded_psexec{
        meta:
                description = "Look for indications of embedded psexec"
                author = "Dragos Inc"
        strings:
                $mz = "!This program cannot be run in DOS mode." ascii wide
                $s1 = "-accepteula -s" ascii wide
                $s2 = ",Sysinternals" ascii wide

        condition:
                all of ($s*) and #mz > 1}

rule shutdown_scheduling{
        meta:
                description = "Shutdown scheduling"
                author = "Dragos Inc"
        strings:
                $s1 = { 68 44 43 01 10 8d 85 d8 f9 ff ff 50 ff 15 1c d2 00 10 85 c0 74 }
                $s2 = { f6 05 44 f1 01 10 04 b8 6c 43 01 10 75 05 }
                $s3 = { 56 57 8d 8d ?? ?? ?? ff 51 50 8d 85 ?? ?? ?? ff 68 a8 42 01 10 }
        condition:
                all of ($s*)}
```

OlympicDestroyer Example

```
rule olympic_destroyer_service_manipulator
{
        meta:
                description = “Service manipulator functionality"
                author = "Joe Slowik, Dragos Inc"
                sha256 =
"ae9a4e244a9b3c77d489dee8aeaf35a7c3ba31b210e76d81ef2e91790f052c85"
        strings:
                $a = { 55 8B EC 83 EC 28 56 68 00 00 00 80 68 ?? ?? ?? 00 33 F6 56 FF 15
?? ?? 40 00 89 ?? ?? 3B C6 0F ?? ?? ?? ?? 00 53 8B ?? ?? ?? ?? 00 57 8D ?? ?? 51 8D ?? ?? 51
8D ?? ?? 51 56 56 6A 03 68 3F 01 00 00 50 89 ?? ?? 89 ?? ?? 89 ?? ?? FF ?? FF ?? ?? 8B ?? ??
?? ?? 00 6A 08 FF ?? 50 FF ?? ?? ?? 40 00 8D ?? ?? 51 8D ?? ?? 51 8D ?? ?? 51 FF ?? ?? 89 ??
?? 50 6A 03 68 3F 01 00 00 }
                $b = { 8B ?? ?? 68 00 00 00 10 FF ?? FF ?? ?? FF ?? ?? ?? 40 00 89 ?? ??
3B C6 74 ?? 8D ?? ?? 51 56 56 50 89 ?? ?? FF ?? FF ?? ?? 6A 08 FF ?? 50 FF ?? ?? ?? 40 00 56
56 56 56 56 56 6A FF 6A 04 6A FF FF ?? ?? 89 ?? ?? FF ?? ?? ?? 40 00 8D ?? ?? 50 FF ?? ??
FF ?? ?? FF ?? ?? FF D3 85 C0 }
        condition:
                uint16(0) == 0x5a4d and all of them
}
```

Putting it All Together

- Host-relevant artifacts pulled down via Bro
- Sort, process, etc. via scripts or whatever is appropriate
- Leverage YARA to look for activity of interest
  - Includes YARA at end of processing scripts

Putting into Practice

- Sensors in place, scripts set up, etc.
- So – what can you actually *look for* that makes up for lack of host detection?

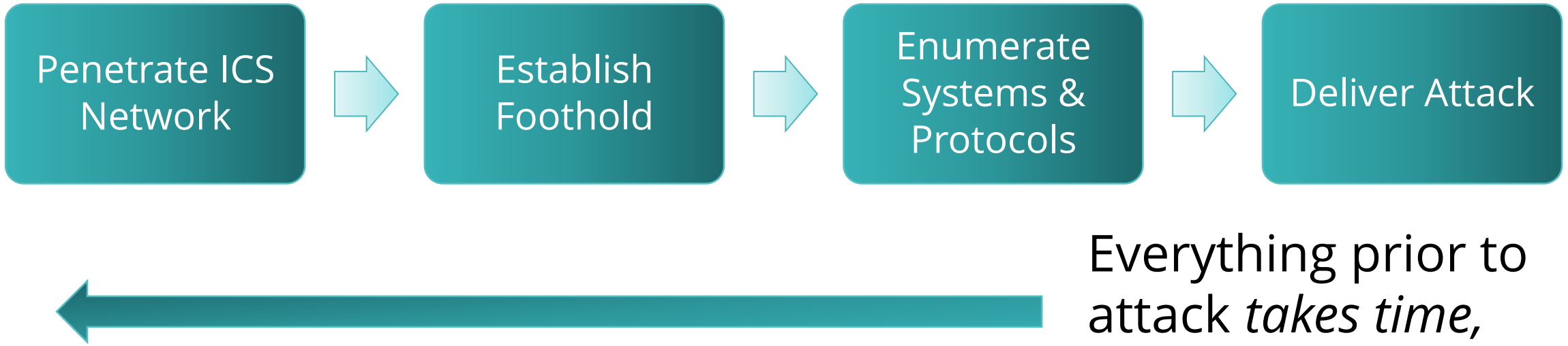
Contextuality

- Answer: depends!
- Environment dictates what you can see, and what you'll need to
- Example environment: ICS
  - AV coverage spotty
  - Host coverage VERY rare
  - Network capture pretty good

CRASHOVERRIDE Modules

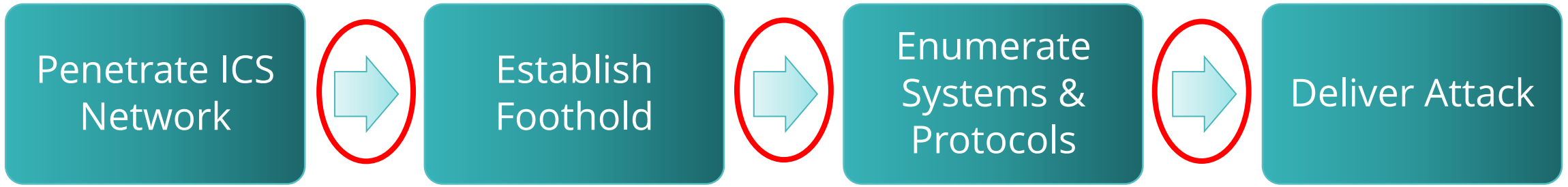
- CRASHOVERRIDE:
  - Modular malware framework
  - Responsible for 2016 Ukraine power outage
- Purpose-built ICS attack framework and payload

CRASHOVERRIDE Attack



Everything prior to attack *takes time, access, and work*

CRASHOVERRIDE Attack



Goal: Identify staging and prepositioning!

CRASHOVERRIDE Movement

```
EXEC xp_cmdshell 'net use L: \\X.X.X.X\C$ <Password>
/USER:<User>'

EXEC xp_cmdshell 'cscript C:\Delta\remote.vbs /s:X.X.X.X
/u:<Domain>\<User> /p:<Password> /t:-r move
C:\intel\imapi.txt C:\Intel\imapi.exe';
```

CRASHOVERRIDE Movement

```
Function CopyFiles(RemoteMachine, Username, Password, SrcFile, DestFile)
        WshNetwork.MapNetworkDrive "", "\\" & RemoteMachine & "\IPC$", false,
Username, Password
        If Err.Number <> 0 Then
                Wscript.StdOut.Write "Error: " & Err.Description
                CopyFiles = 1
                Exit Function
        End If

        DestFile = "\\" & RemoteMachine & "\" + Replace(DestFile, ":", "$")
        Set File = FSO.GetFile(SrcFile)
        File.Copy DestFile, True
        WshNetwork.RemoveNetworkDrive "\\" & RemoteMachine & "\IPC$"
        If Err.Number <> 0 Then
                Wscript.StdOut.Write "Error: " & Err.Description
                CopyFiles = 2
                Exit Function
        End If
        CopyFiles = 0
End Function
```

Movement Conclusions

- Leveraging ‘living off the land techniques’
  - Net Use
  - PSEXEC
  - Wscript
- Leaves protocol trail – primarily SMB

Detection Strategy

- Capture file transfer activity
- Parse files, analyze for malicious intent
- Take advantage of adversary need to ‘drill down’ into network

Bro SMB Capture

```
@load base/frameworks/files
@load ./main

module SMB;

export { ## Default file handle provider for SMB.
         global get_file_handle: function(c: connection, is_orig: bool): string;

         ## Default file describer for SMB.
         global describe_file: function(f: fa_file): string;}

function get_file_handle(c: connection, is_orig: bool): string
         {if ( ! (c$smb_state?$current_file &&
                  (c$smb_state$current_file?$name ||
                   c$smb_state$current_file?$path)) )
                  {
                  # TODO - figure out what are the cases where this happens.
                  return "";                  }

To Be Continued!
```

Malware Overview

- Custom ICS protocol implementation frameworks
- Destructive module to impede restoration
- ‘Off the shelf’ items
  - PSExec
  - Mimikatz (packed)

IEC-104 Impact Module

File information

Identification | Details | Content | Analyses | Submissions | ITW | Comments

2016-12-19 10:06:04	**3/55**
2016-12-26 10:06:29	**1/56**
2017-01-02 10:08:25	**2/57**
2017-01-09 10:14:22	**1/56**
2017-01-16 11:31:13	**2/58**
2017-02-01 04:34:04	**3/57**
2017-02-18 10:09:49	**5/59**
2017-02-25 10:16:36	**8/58**
2017-06-12 14:50:30	**9/61**
2017-06-12 15:28:19	**8/61**

Engine	Signature	Version	Update
Ad-Aware	-	3.0.3.794	20161219
AegisLab	-	4.2	20161219
AhnLab-V3	-	3.8.2.16235	20161219
ALYac	-	1.0.1.9	20161219
Antiy-AVL	-	1.0.0.1	20161219
Arcabit	-	1.0.0.791	20161219
Avast	-	8.0.1489.320	20161219
AVG	-	16.0.0.4739	20161219
Avira	-	8.3.3.4	20161219
AVware	-	1.5.0.42	20161219
Baidu	Win32.Trojan.WisdomEyes.16070401.9500.9926	1.0.0.2	20161207

Take-Away

- From an AV perspective, not much
- From an ICS-specific perspective, many items in payload would have been interesting
- Adding ‘custom’ detection midpoint would identify payload prepositioning

Take-Away: Specifics

```
rule crashoverride_configReader{
        meta:
                description = "CRASHOVERRIDE v1 Config File Parsing"
                author = "Dragos Inc"
                sha256 = "7907dd95c1d36cf3dc842a1bd804f0db511a0f68f4b3d382c23a3c974a383cad"
        strings:
                $s0 = { 68 e8 ?? ?? ?? 6a 00 e8 a3 ?? ?? ?? 8b f8 83 c4 ?8 }
                $s1 = { 8a 10 3a 11 75 ?? 84 d2 74 12 }
                $s2 = { 33 c0 eb ?? 1b c0 83 c8 ?? }
                $s3 = { 85 c0 75 ?? 8d 95 ?? ?? ?? ?? 8b cf ?? ?? }
        condition:
                uint16(0) == 0x5a4d and all of them}

rule dragos_crashoverride_moduleStrings {
        meta:
                description = "IEC-104 Interaction Module Program Strings"
                author = "Dragos Inc"
        strings:
                $s1 = "IEC-104 client: ip=%s; port=%s; ASDU=%u" nocase wide ascii
                $s2 = " MSTR ->> SLV" nocase wide ascii
                $s3 = " MSTR <<- SLV" nocase wide ascii
                $s4 = "Unknown APDU format !!!" nocase wide ascii
                $s5 = "iec104.log" nocase wide ascii
        condition:
                any of ($s*)
```

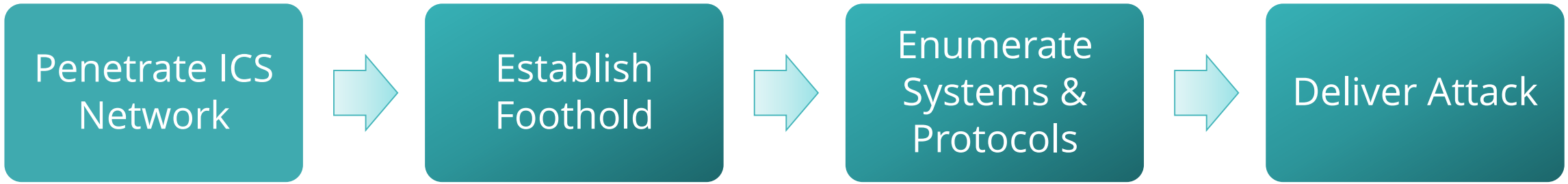
Environment-Specific Defense

- Build detections around environment
- Implement them at network choke-points
- Detect suspicious items in advance of attack
  - Malicious code must be brought into environment
  - Take advantage of attacker dependencies

TRISIS Malicious Logic Files

- TRISIS:
  - Third ICS-impacting malware
  - First to target safety systems
- Establish backdoor to replace safety system logic

TRISIS Attack



TRISIS Attack in Context

- *Wait a minute – that looks just like CRASHOVERRIDE!*
- YES!
  - Same dependencies to access ICS
  - Similar challenges in establishing C2
  - Attack requires moving attack payload into network

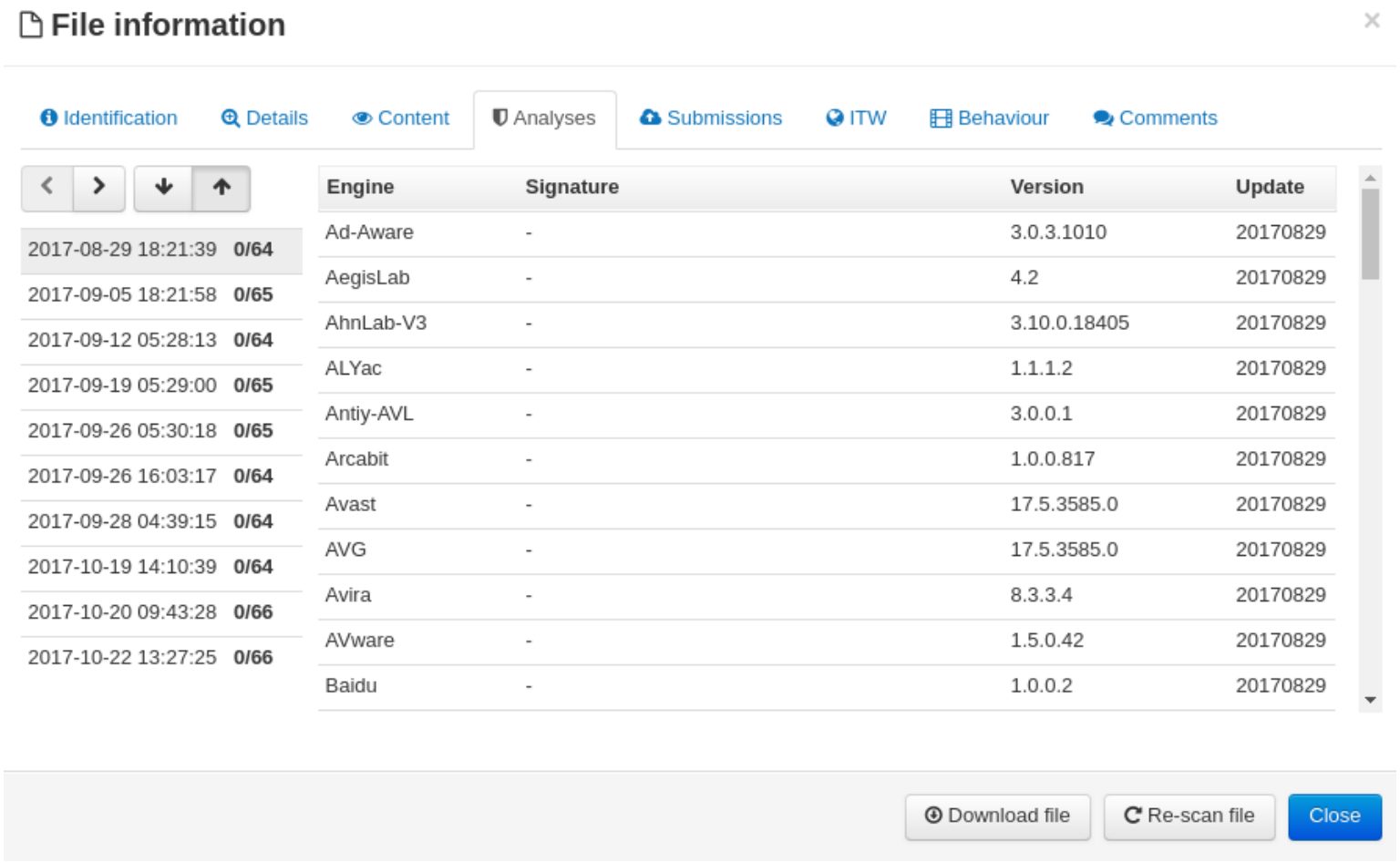
TRISIS Attack Path

- Malicious payload downloaded from engineering workstation to target SIS
- Payloads and upload/inject program (compiled Python) moved to workstation

TRISIS Defense

- Similar principles hold:
  - Add detection at chokepoints
  - Look for items of interest traversing
- Leverage network visibility to catch items otherwise only seen on host

TRISIS Defense != AV



TRISIS Observables

- AV failed to pick out TRISIS
- But numerous items ‘strange’ to ICS would allow for detection:
  - Compiled Python EXE
  - File headers and content for malicious logic files outside of known service/update times

TRISIS Identification - General

```
rule compiledPython{
        meta:
                description = "Identify compiled Python objects - Should be rare to non-
                                existent in ICS environments”
                author = "Dragos Inc."
        strings:
                $s1 = "PyImport_" nocase wide ascii
                $s2 = "PyErr_" nocase wide ascii
                $s3 = ".pyd" nocase wide ascii
                $s4 = "py2exe" nocase wide ascii
                $a1 = "cyberoam" nocase wide ascii fullword
                $a2 = "plctalk" nocase wide ascii fullword
                $a3 = "greenbow" nocase wide ascii fullword
                $a4 = "mbnet" nocase wide ascii fullword
                $a5 = "mbconnect" nocase wide ascii fullword
                ….
                $a** = "trilog" nocase ascii wide fullword
        condition:
                uint16(0) == 0x5a4d and 2 of ($s*) and 1 of ($a*)}
```

TRISIS Take-Aways

- Basically ZERO visibility on SIS
- Leverage network capture to fill in (some) blanks
- Look for items that either:
  - Never belong
  - Only appear during known, legit activity

DYMALLOY Screen Shots

- DYMALLOY is an ICS activity group targeting North America, Europe, Turkey
- Superficial similarity to legacy DRAGONFLY
- Part of ICS intrusion: exfil HMI screenshots

DYMALLOY TTPs

Initial Access:

- Phishing
- Strategic website compromise

Deploy Implants:

- RATs: Karagany.B, Heriplor
- Backdoors: DorShel, Goodor

Information Collection

- Mimikatz integrated into broader credential capture tool
- Framework for harvesting documents, intelligence info
- Exfiltrate HMI screenshots for process and network information

DYMALLOY Detection

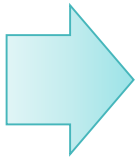
- Screenshot activity in ICS environment is an excellent alerting point
- Something that would *not* get picked up by traditional security solutions
- Deploy Bro to carve image files, analyze to determine file significance

Screenshot Identification

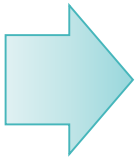
```
ExifTool Version Number         : 10.60
File Name                       : Windows7x64_TB-2018-01-12-20-00-08.png
Directory                       : .
File Size                       : 68 kB
File Modification Date/Time     : 2018:01:12 20:00:08-07:00
File Access Date/Time           : 2018:01:14 09:31:00-07:00
File Inode Change Date/Time     : 2018:01:12 20:00:08-07:00
File Permissions                : rw-------
File Type                       : PNG
File Type Extension             : png
MIME Type                       : image/png
Image Width                     : 1280
Image Height                    : 1024
Bit Depth                       : 8
Color Type                      : RGB
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Image Size                      : 1280x1024
Megapixels                      : 1.3
```

Screenshot Stupid Simple Alert

Identify Image File in Network Traffic FROM ICS



Carve File via Bro and Move to Analysis Machine

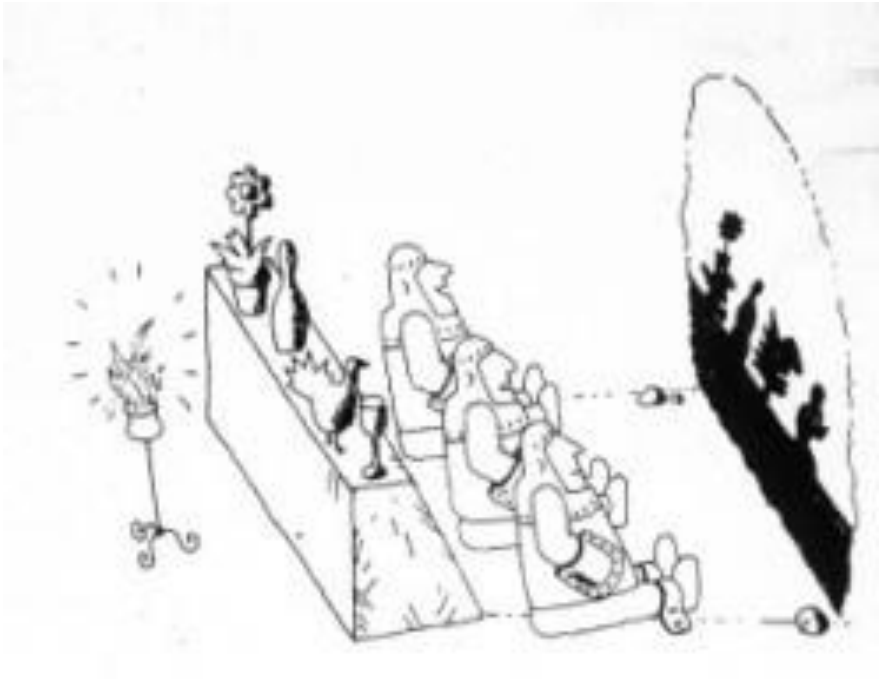


Analyze EXIF Data to See if Image Size Matches Set of Screen Resolutions

DYMALLOY Implementation

- YARA applied to EXIF results
- Pattern off of ‘common’ screen resolutions
- Leverage as alerting data point

Shadows on the Wall



- Ultimately, this approach remains an *approximation*
- Not a replacement for host visibility
- Making the best of what you have

Key Weaknesses

- Encryption
- Compound File Types
- Lack of sensors
- ‘Flat’ network topology
- Reactive, not preventative

Encryption

- Potentially the greatest issue
  - Many threat actors moving to HTTPS
  - Increasing use of encryption by default
- Not as applicable in some environments
  - E.g., ICS remains rare for encrypted traffic

Encryption Work-Arounds

- SSL intercept
  - Justifiable given shifting threat landscape
  - But a tough sell
- Identifying *host* work-arounds if possible
  - Yes, defeats purpose of this discussion
  - Shifts conversation to lack of host visibility

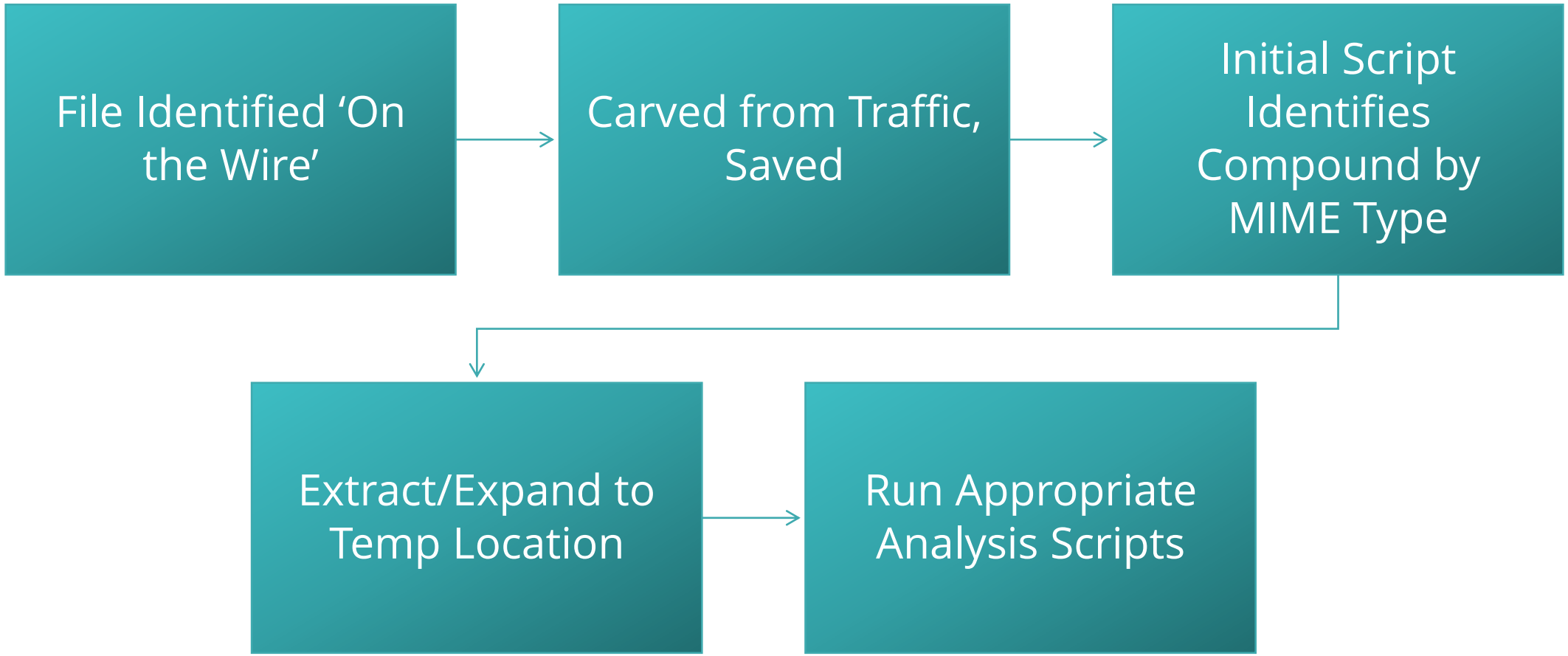
Encryption Work-Arounds

- Host and application *fingerprinting*
- JA3 project
  - https://github.com/salesforce/ja3
- Identify custom or anomalous encrypted communications via system and application fingerprint

Compound File Types

- This approach works REALLY WELL for things like PE files
- Compound or archive types – not so much:
  - Zip, RAR, etc.
  - DOCX, XLSX, etc.

More Complex Analysis



Stupid-Simple Example

```
#!/bin/bash
#Script for XML-type documents to unzip, scan with Yara, and look for
Phishery indicators (IP address)
yaraRules=$1

for f in *; do
        mkdir tmp
        7za x -otmp $f > /dev/null
        yara $yaraRules -r tmp/ >> ${f}_yara.results
        grep -oEr "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" tmp/ >>
${f}_grep.results
        rm -R tmp
done

#Remove empty result files
find . -name "*.results" -size 0 -exec rm {} \;
```

Compound File Types

- This is doable – just requires more effort
- Key is finding a sustainable workflow:
  - Won't overtax storage
  - Keep processing requirements to min

Sensor Coverage

- Network edge typically covered
  - Covers C2, downloads, etc.
- Internal traffic – less so
  - Needed to capture lateral movement
- Align coverage to choke-points as best as possible

Flat Topology

- Flat networks are BAD
  - But they still exist
- Similar to sensor coverage issue but less scope to ‘fix’
- Architecture item – hard to implement, but once you do good things

Still Reactive Only

- Method will tell you something bad happened – or is happening
- Damage is done!

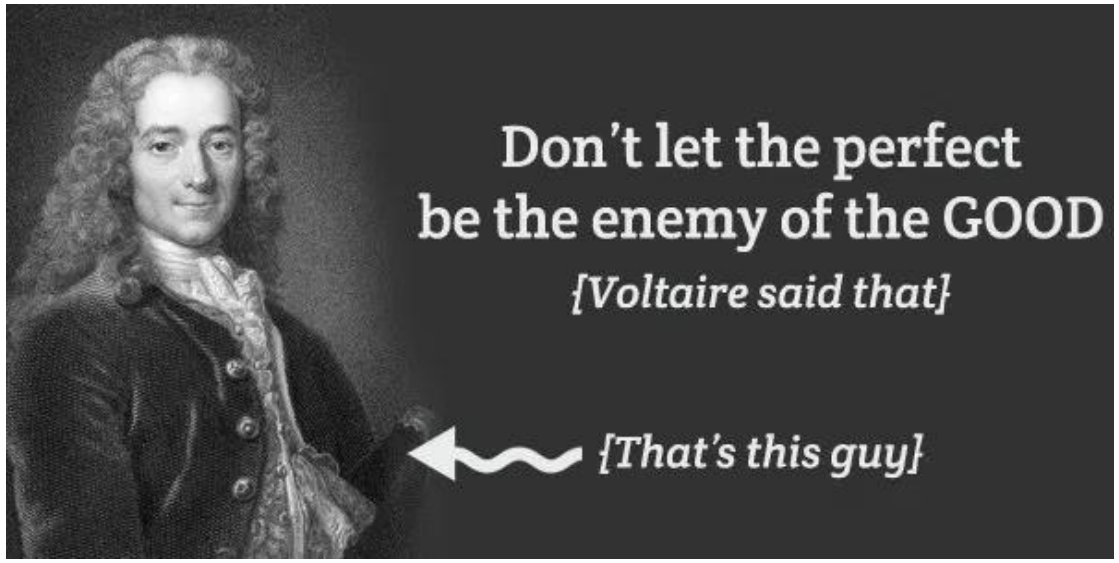
Minimize Response Time

- You might be reacting – but quicker than before
- Goal is to respond faster
- Picking off in network traffic means identifying badness before it spreads from ‘poor coverage’ areas

Wait, You Talked a Lot about ICS

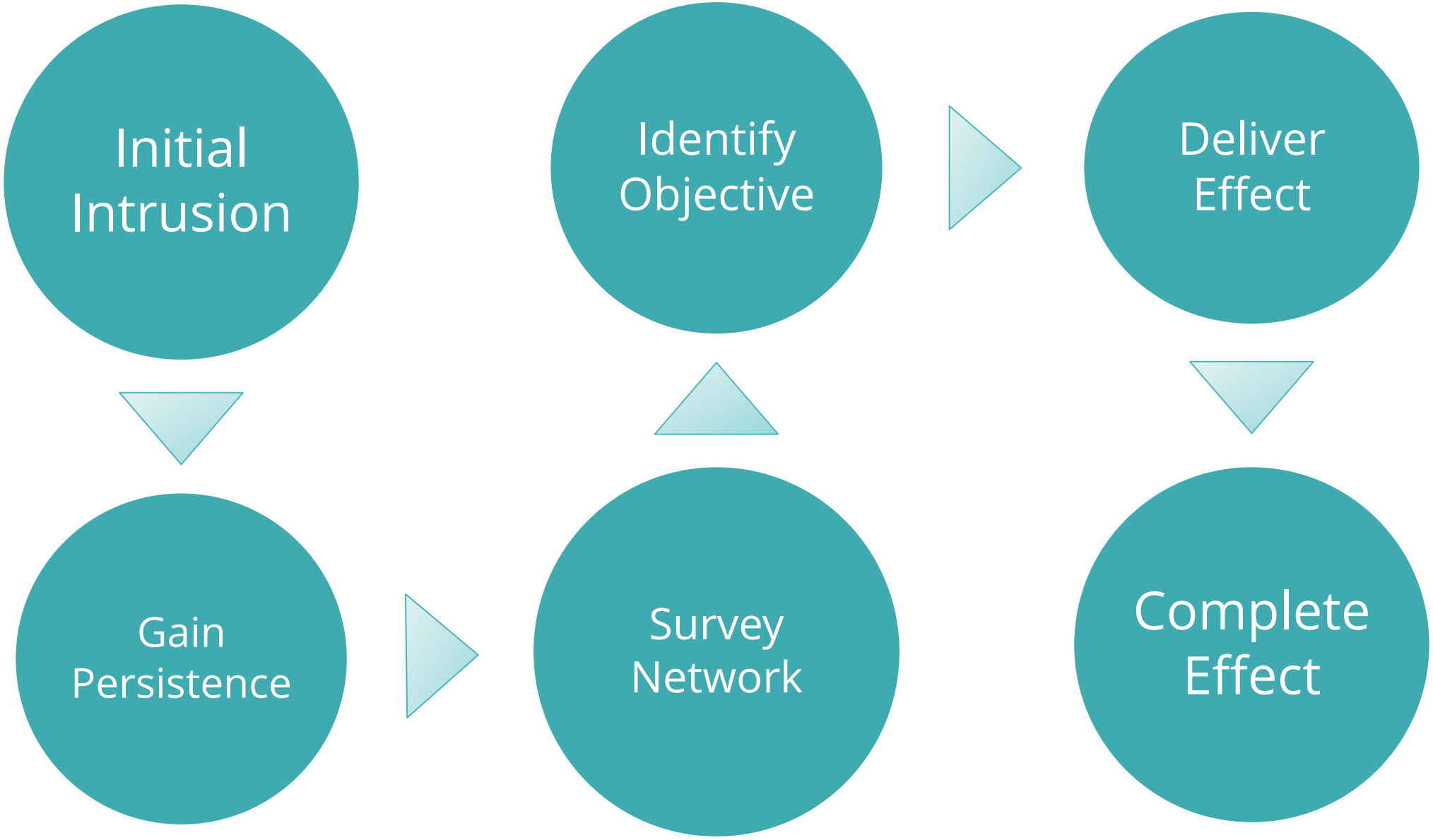
- ICS networks are well-tailored to this approach
  - *And it is also my day job*
- HOWEVER – aspects of this can apply to various other environments
- Purpose: apply what you can based on YOUR problems

Good != Perfect



- In imperfect situations, can still improve security posture

- Reducing response times can limit infections
- Identify activity earlier in attack chain


Initial Intrusion
Identify Objective
Deliver Effect
Gain Persistence
Survey Network
Complete Effect

Questions?

jslowik@dragos.com