National Cyber Security Centre
a part of GCHQ

Advisory: Hostile state actors compromising UK organisations with focus on engineering and industrial control companies

5 April 2018

© Crown Copyright 2018

About this document

In the past year, the NCSC has noted widespread targeting of UK infrastructure devices by hostile state actors. This has primarily focused on engineering and industrial control companies and is ongoing.

This advisory provides an update on the current threat and guidance for any organisations affected.

Handling of the Report

Information in this report has been given a Traffic Light Protocol (TLP) of WHITE, which means, subject to standard Copyright rules, it may be distributed without restriction.

Disclaimer

This report draws on reported information, as well as information derived from industry sources.

Introduction

The NCSC is aware of an ongoing attack campaign against multiple companies involved in the Critical National Infrastructure (CNI) supply chain. These attacks have been ongoing since at least March 2017.

The targeting is focused on engineering and industrial control companies and has involved the harvesting of NTLM [1] credentials via Server Message Block (SMB) using strategic web compromises and spear-phishing.

This advisory highlights the sustained risk to UK companies involved in these sectors, provides further details on the activity and offers guidance for any organisations affected.

Further information on this activity was published on 15 March by the US Department of Homeland Security.[2] The activity has also been highlighted previously by threat intelligence companies in open sources as Berserk Bear, Energetic Bear, Dragonfly, Havex and Crouching Yeti[3].

Details

The NCSC is aware of an attack campaign against multiple companies involved in the CNI supply chain. These attacks have been ongoing since at least March 2017 and are ongoing as of 5 April 2018.

The precise timeline for each compromise may vary. However, in several of the compromises identified, once the attacker interacted with the network, exploitation was typically achieved within a week, with ongoing access maintained over the course of months.

Outline of attack

The following diagrams outline the various steps taken to compromise target networks and propagate through them.

Initial Infection

1. The attacker gets the target PC to communicate with a malicious fileserver under actor control using one of two methods:

[1] Windows Challenge/Response (NTLM) is the authentication protocol used on networks that include systems running the Windows operating system and on stand-alone systems.
[2] https://www.us-cert.gov/ncas/alerts/TA18-074A
[3] https://www.symantec.com/blogs/threat-intelligence/dragonfly-energy-sector-cyber-attacks
https://www.baesystems.com/en/feature/havex
https://www.kaspersky.co.uk/resource-center/threats/crouching-yeti-energetic-bear-malware-threat

a. The attacker carries out a watering hole attack, compromising a website of interest to the target, and adding a link to a resource located on the malicious fileserver.
b. The attacker sends a spear-phishing email from a compromised account containing a document of interest (sometimes a known contact of the target). In several instances, stolen CVs have been used, which are configured to load a remote template from the malicious fileserver.

2. Running Inveigh PowerShell scripts[4] on the fileserver, the attacker harvests all the NTLM hashes sent to it by the target hosts that are attempting to logon and load the various resources.

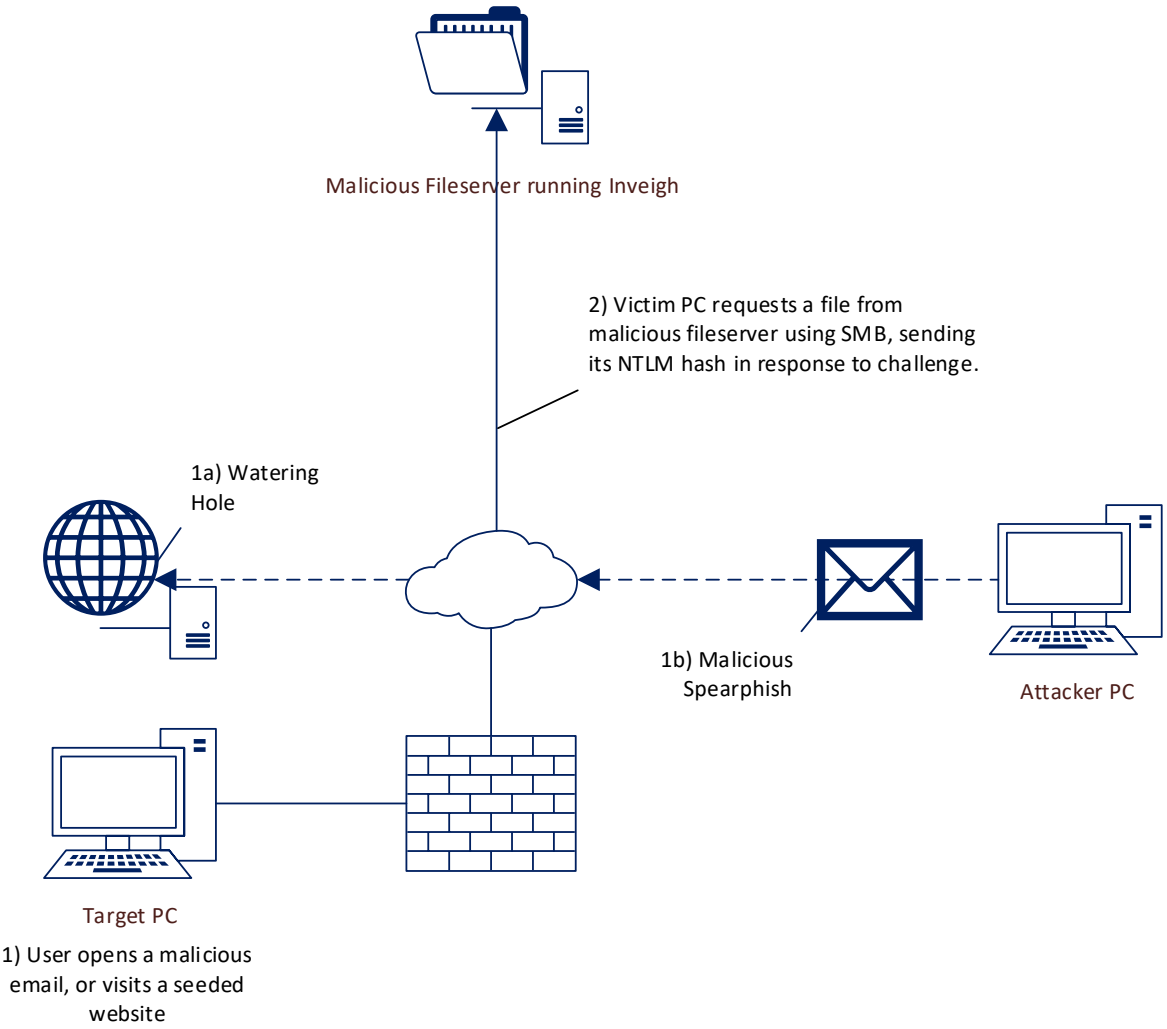


Figure 1: Initial Infection

Exploitation and lateral movement

[4] Publicly available SMB/HTTP man-in-the-middle tool.

3. Using the NTLM hash acquired in the previous step, the following actions are possible:
    a. The attacker uses the Inveigh-Relay PowerShell script to replay the hash against an exposed area of the network.
    b. The attacker cracks the captured hash offline to obtain user credentials needed to access the network via a VPN/RDP or other remote access protocol that is enabled.

4. Once access to the network has been obtained, the attacker will typically enumerate shares they can access, and into these, place shortcut/link files (.lnk) with an icon that is located on the malicious fileserver.

5. Each host that views the shortcut in file explorer (even if it is not opened) attempts to load the icon from the malicious fileserver, thus sending their NTLM hashes out to be replayed/harvested.

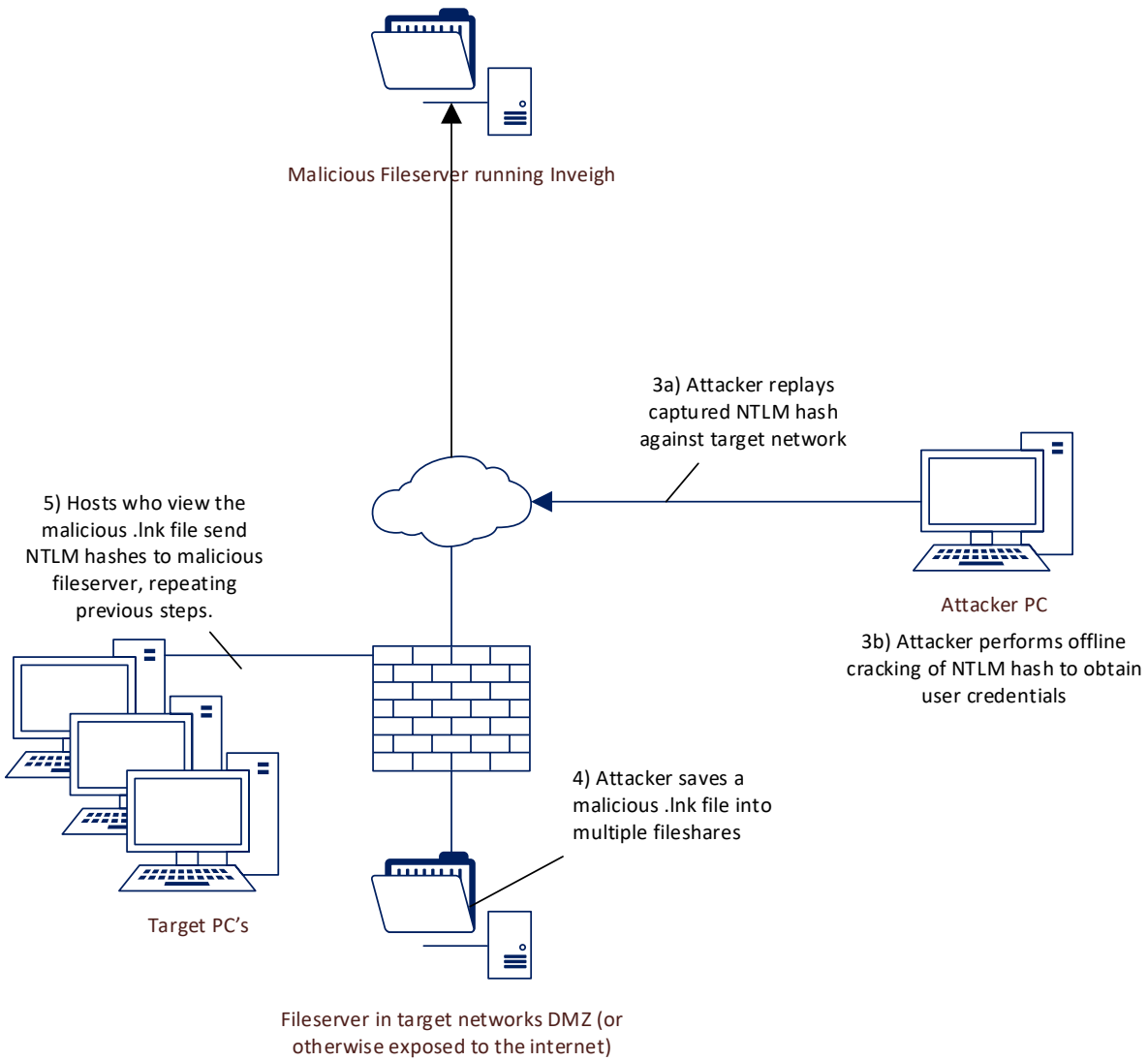


Figure 2: Exploitation and Lateral Movement

Post exploitation

Once access has been gained on the network, the attacker will pivot between various machines using harvested credentials, penetration testing and network administration tools. In some cases, the actors have also deployed custom malware/downloaders to maintain persistence and they will also take advantage of and actively look for any password stores or VPN/remote access guides. If administrative access is obtained, the attacker will likely add a new domain admin account to the network. Likewise, if access to the webserver is obtained, they will likely deploy a web-shell to facilitate ongoing access.

The attacker will also use cracked or stolen credentials to access the company's mail server and harvest the contact list of the compromised user. The compromised mail server may then be used to send spear-phishing emails emanating from the victim company to additional targets, increasing their perceived legitimacy.

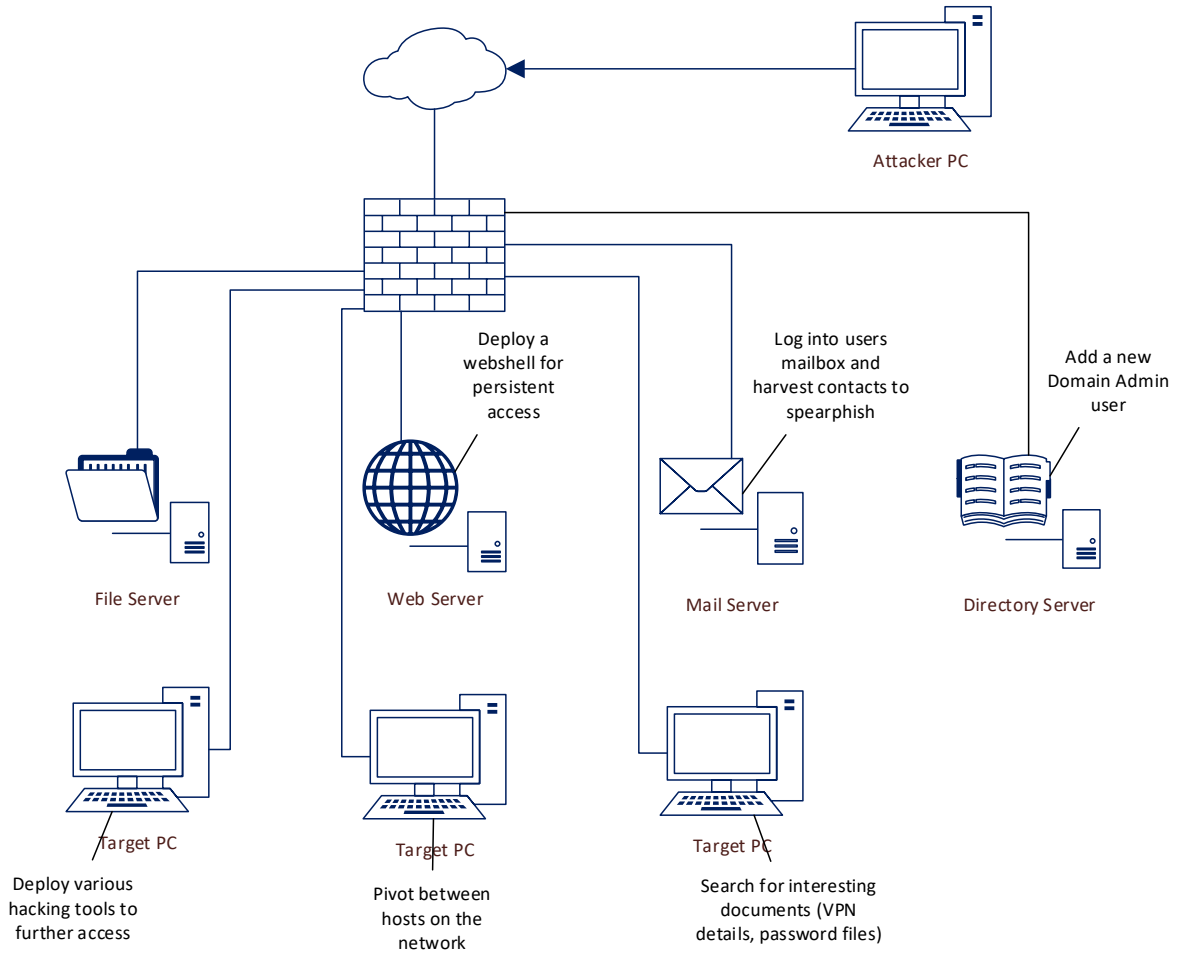


Figure 3: Post-exploitation

Tools used

The following is a list of publicly available and custom tools used by the attacker in each stage of these attacks.

Tool Name	Description	Detail
Angry IP Scanner	Publicly available IP scanning tool.	Used on attacker infrastructure to survey external network endpoints, or on internal victim machines to survey the internal network.
Backdoor.goodor	Custom downloader written in Google Go.	Calls out to one of nine hardcoded IP addresses. Yara and Snort rules provided. SHA256: b5278301da06450fe4442a25dda2d83d21485be63598642573f59c59e980ad46
CrackMapExec	Publicly available post exploitation tool.	Can be used from attacker infrastructure to reuse credentials on networks that have the appropriate protocols exposed. Otherwise, it may be deployed on the initial network foothold to further pivot around victim machines.
Dorshell	A backdoor downloader built using Shellter.	Yara rules provided. SHA256: b8bc0611a7fd321d2483a0a9a505251e15c22402e0cfdc62c0258af53ed3658a
Get-GPPPassword	PowerShell script to retrieve passwords of accounts provisioned by group policy.	Used on victim machines.
Inveigh	Publicly available spoofing and man-in-the-middle tool.	Can be used on attacker infrastructure and also on victim networks to facilitate SMB relaying.
Malicious JavaScript downloader	Obfuscated JavaScript that downloads second stage malware.	Executed on victim to download second stage. Yara rule provided. SHA256: 371d8fef7f976bf386cbcbe95358eb92dc764bdffaf3ced72b931c2e19e94378
Malicious Shortcut File	Windows shortcut file with its icon resource set to be loaded from remote fileserver.	Used to harvest additional credentials from the victim network. Yara rule provided.
Malicious Word document	Word document that loads a template file from a remote fileserver.	Word documents used were legitimate documents that had been acquired and weaponised using Phishery.
Mimikatz	Publicly available multifunction tool primarily used for password extraction.	Used on victim machines.
Phishery	Publicly available framework for spear-phishing.	Used to craft the spear-phishing documents used in these attacks.
Powershell	Inbuilt Windows utility for advanced scripting.	Used on victim machines to facilitate other tools, download additional code etc.

Psexec	Publicly available tool from Sysinternals for remote command execution.	Used on victim machines.
RDP Bruteforcer	RDP brute forcing tool.	Observed as brute.exe. Likely used on both attacker and victim machines. Yara rule provided. SHA256: 8234bf8a1b53efd2a452780a69666d1aedcec9eb1bb714769283ccc2c2bdcc65
Screenutil	Screenutil command line tool used to capture screenshots.	Used on victim machines.
Z_Webshell	Comprehensive ASPX web-shell.	Deployed on customer webserver to maintain persistence. Yara rule provided. SHA256: ace12552f3a980f1eed4cadb02afe1bfb851cafc8e58fb130e1329719a07dbf0

Table 1: Malicious Tools used

Infrastructure

The attacker makes use of both compromised infrastructure and purchased VPS infrastructure, and therefore it should be noted that the IP addresses listed below may not be wholly malicious, or may be in use by multiple malicious actors, and should be used for intelligence gathering purposes only.

IP Address	Use
2.229.10.193	Callback for Goodor malware
5.150.143.107	Callback for Goodor malware
5.153.58.45	Fileserver used to capture NTLM hashes
41.205.61.221	Callback for Goodor malware
41.78.157.34	Callback for Goodor malware
62.8.193.206	Fileserver used to capture NTLM hashes
78.47.199.220	Endpoint used to access network
81.149.16.168	Endpoint used to access webshell
82.222.188.18	Callback for Goodor malware
85.25.100.104	Endpoint used to access network
85.255.235.109	Endpoint used to access network
85.255.235.147	Endpoint used to access network
85.185.45.174	Endpoint used to access network
85.159.65.114	Hosting malicious files
91.183.104.150	Staging infrastructure
111.93.118.90	Endpoint used to access network
130.25.10.158	Callback for Goodor malware
139.162.108.53	Endpoint used to access network
139.162.114.70	Endpoint used to access network
149.210.156.198	Endpoint used to Access webshell/mailserver
151.80.163.14	Endpoint used to access network
167.114.44.147	Callback/hoster of malicious files
173.212.212.56	Endpoint used to access network
176.53.11.130	Callback for Goodor malware
184.154.150.66	Fileserver used to capture NTLM hashes
185.22.184.71	Endpoint used to access network

187.130.251.249	Callback for second stage executable
193.213.49.115	Callback for Goodor malware
195.250.149.195	Endpoint used to access network
195.87.199.197	Callback for Goodor malware
203.113.4.230	Fileserver used to capture NTLM hashes

Table 2: Infrastructure used

Detection

YARA Rules

Backdoor.goodor

```
rule Bytes_used_in_AES_key_generation {
  meta:
    author = "NCSC"
    hash =
"b5278301da06450fe4442a25dda2d83d21485be63598642573f59c59e980ad46"
  strings:
    $a1 = {35 34 36 35 4B 4A 55 54 5E 49 55 5F 29 7B 68 36 35 67 34 36 64
66 35 68}
    $a2 = {fb ff ff ff 00 00}
  condition:
    all of ($a*)
}

rule Partial_Implant_ID {
  meta:
    author = "NCSC"
    hash =
"b5278301da06450fe4442a25dda2d83d21485be63598642573f59c59e980ad46"
  strings:
    $a1 = {3838313435364643}
    $a2 = {fb ff ff ff 00 00}
  condition:
    all of ($a*)
}

rule Sleep_Timer_Choice {
  meta:
    author = "NCSC"
    hash =
"b5278301da06450fe4442a25dda2d83d21485be63598642573f59c59e980ad46"
  strings:
    $a1 = {8b0424b90f00000083f9ff743499f7f98d420f}
    $a2 = {fb ff ff ff 00 00}
  condition:
    all of ($a*)
}

rule User_Function_String {
```

```
  meta:
    author = "NCSC"
    hash =
"b5278301da06450fe4442a25dda2d83d21485be63598642573f59c59e980ad46"
  strings:
    $b1 = {fb ff ff ff 00 00}
    $a2 = "e.RandomHashString"
    $a3 = "e.Decode"
    $a4 = "e.Decrypt"
    $a5 = "e.HashStr"
    $a6 = "e.FromB64"
  condition:
    $b1 and 3 of ($a*)
}
```

Dorshell

```
rule generic_shellcode_downloader_specific {
  meta:
    author = "NCSC"
    hash =
"b8bc0611a7fd321d2483a0a9a505251e15c22402e0cfdc62c0258af53ed3658a"
  strings:
    $push1 = {68 6C 6C 6F 63}
    $push2 = {68 75 61 6C 41}
    $push3 = {68 56 69 72 74}
    $a = {BA 90 02 00 00 46 C1 C6 19 03 DD 2B F4 33 DE}
    $b = {87 C0 81 F2 D1 19 89 14 C1 C8 1F FF E0}
  condition:
    (uint16(0) == 0x5A4D and uint16(uint32(0x3C)) == 0x4550) and ($a or
$b) and @push1 < @push2 and @push2 < @push3
}

rule generic_shellcode_downloader {
  meta:
    author = "NCSC"
    hash =
"b8bc0611a7fd321d2483a0a9a505251e15c22402e0cfdc62c0258af53ed3658a"
  strings:
    $push1 = {68 6C 6C 6F 63}
    $push2 = {68 75 61 6C 41}
    $push3 = {68 56 69 72 74}
  condition:
    (uint16(0) == 0x5A4D and uint16(uint32(0x3C)) == 0x4550) and @push1 <
@push2 and @push2 < @push3
}
```

Malicious batch files

```
rule Batch_Script_To_Run_PsExec {
  meta:
    author = "NCSC"
    hash =
"b7d7c4bc8f9fd0e461425747122a431f93062358ed36ce281147998575ee1a18"
  strings:
    $ = "Tokens=1 delims=" ascii
    $ = "SET ws=%1" ascii
    $ = "Checking %ws%" ascii
    $ = "%TEMP%\\%ws%ns.txt" ascii
    $ = "ps.exe -accepteula" ascii
  condition:
    3 of them
}

rule Batch_Powershell_Invoke_Inveigh {
  meta:
    author = "NCSC"
    hash =
"0a6b1b29496d4514f6485e78680ec4cd0296ef4d21862d8bf363900a4f8e3fd2"
  strings:
    $ = "Inveigh.ps1" ascii
    $ = "Invoke-Inveigh" ascii
    $ = "-LLMNR N -HTTP N -FileOutput Y" ascii
    $ = "powershell.exe" ascii
  condition:
    all of them
}
```

Malicious Javascript

```
rule Obfuscated_Javascript {
  meta:
    author = "NCSC"
    hash =
"371d8fef7f976bf386cbcbe95358eb92dc764bdffaf3ced72b931c2e19e94378"
  strings:
    $a1 = "replace(/[^A-Za-z0-9\\+\\/\\=]/g,\"\")"
    $a2 = "(1E3)"
    $a3 = /.=.<<2|.>>4/
    $a4 = /.=(.&15)<<4|.>>2/
    $b1 = /this\[.\(/
  condition:
    all of ($a*) and #b1 > 4
}
```

Malicious Link

```
rule lnk_detect {
  meta:
    author = "NCSC"
    description = "malicious lnk properties"
  strings:
    $lnk_magic = {4C 00 00 00 01 14 02 00 00 00 00 00 C0 00 00 00 00 00 00
46}
    $lnk_target = {41 00 55 00 54 00 4F 00 45 00 58 00 45 00 43 00 2E 00
42 00 41 00 54}
    $s1 = {5C 00 5C 00 31 00}
    $s2 = {5C 00 5C 00 32 00}
    $s3 = {5C 00 5C 00 33 00}
    $s4 = {5C 00 5C 00 34 00}
    $s5 = {5C 00 5C 00 35 00}
    $s6 = {5C 00 5C 00 36 00}
    $s7 = {5C 00 5C 00 37 00}
    $s8 = {5C 00 5C 00 38 00}
    $s9 = {5C 00 5C 00 39 00}
  condition:
    (($lnk_magic at 0) and $lnk_target) and ($s1 or $s2 or $s3 or $s4 or
$s5 or $s6 or $s7 or $s8 or $s9)
}
```

RDP Bruteforcer

```
rule RDP_Brute_Strings {
  meta:
    author = "NCSC"
    hash =
"8234bf8a1b53efd2a452780a69666d1aedcec9eb1bb714769283ccc2c2bdcc65"
  strings:
    $ = "RDP Brute" ascii wide
    $ = "RdpChecker" ascii
    $ = "RdpBrute" ascii
    $ = "Brute_Count_Password" ascii
    $ = "BruteIPList" ascii
    $ = "Chilkat_Socket_Key" ascii
    $ = "Brute_Sync_Stat" ascii
    $ = "(\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}:\\d{1,5})" wide
    $ = "BadRDP" wide
    $ = "GoodRDP" wide
    $ = "@echo off{0}:loop{0}del {1}{0}if exist {1} goto loop{0}del
{2}{0}del \"{2}\"" wide
    $ = "Coded by z668" wide
  condition:
    4 of them
}
```

Z_Webshell

```
rule Z_WebShell {
  meta:
    author = "NCSC"
    hash =
"ace12552f3a980f1eed4cadb02afe1bfb851cafc8e58fb130e1329719a07dbf0"
  strings:
    $ = "Z_PostBackJS" ascii wide
    $ = "z_file_download" ascii wide
    $ = "z_WebShell" ascii wide
    $ = "1367948c7859d6533226042549228228" ascii wide
  condition:
    3 of them
}
```

SNORT rules

Backdoor.goodor

```
Alert tcp any any <> any any (flow: established; msg: "backdoor.goodor
beacons"; content: "User-Agent|3a|Go-http-client/1.1|0d0a|Accept-
Encoding|3a|gzip"; pcre:"/\.(aspx|txt)\?[a-z0-9]{3}=[a-z0-9]{32}&/"; sid:
00000001; rev: 1; priority: 1;)
```

Log artefacts

The following information on open source tools used by the actor was gathered by executing the tools themselves and monitoring their activity.

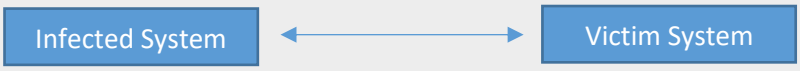
The information may vary between systems and was collected with enhanced logging enabled. The additional logging was gathered by enabling Audit Policy, which was achieved by auditing both success and failure attempts in all of the policies under ‘Audit Policy’ in the ‘Local Group Policy Editor’.

There was also an auditing entry added to the ‘Advanced Security Setting for local Disk’ that utilised several of the advanced permissions to enable more in-depth logging of the systems.

CrackMapExecWin

Description	This is the original CrackMapExec tool compiled for Windows. Primary function is as a post-exploitation tool that aids in the automation of assessing the security posture of large active directory networks.
Type	Command line execution tool
Target OS	Windows
Hashes	SHA1 - F4250B961BD1C8694A949429F739D9F424283612

MD5 - 0E7D5D16E03393605F5F4862F1B9CC37

Attack usage	 CrackMapExecWin could be deployed on the infected system and would be used to target other systems on the network. This tool is a python based utility that can interact with other tools such as PowerShell Empire, enabling an attack in further compromising the network.

Installing CrackMapExecWin on Infected system

Scenario	Running the crackmapexec.exe executable on the infected system
Log type and name	**Acquired information**
Event log - Security	Event ID: **4688** - A new process has been created **4689** - A process has exited Process name: [filepath]\crackmapexec.exe
Event log - Security	Event ID: **4690** - An attempt was made to duplicate a handle to an object Same source process ID as previous 4688 event
Event log - Security	Event ID: **4658** - The handle to an object was closed **4656** - A handle to an object was requested **4663** - An attempt was made to access an object Process name: [filepath]\crackmapexec.exe
Execution history - Prefetch	Prefetch file - C:\Windows\Prefetch\CRACKMAPEXEC.EXE-[PREFETCH_TRAILING_HEX].pf
Execution history - Registry entries	HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\CurrentLanguage HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows NT\CurrentVersion\Perflib\009 HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows NT\CurrentVersion\Perflib\CurrentLanguage HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\bam\UserSettings\[SID] HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\bam\UserSettings\[SID] HKEY_USERS\[SID]\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store

Executing PowerShell Command on Infected System

Scenario	Running the following command on the infected system *.\crackmapexec.exe localhost -u [user] -p ["password"] -lusers*
Event log - Security	Event ID: **4688** - A new process has been created **4689** - A process has exited New process name: [filepath]\crackmapexec.exe Creator process name: C:\Windows\System32\windowsPowershell\v1.0\powershell.exe
Event log - Security	Event ID: **4658** - The handle to an object was closed **4656** - A handle to an object was requested **4663** - An attempt was made to access an object New process name: [filepath]\crackmapexec.exe
Event log - Security	Event ID: **4690** - An attempt was made to duplicate a handle to an object Source process ID matches that of the previous events

Executing PowerShell command on infected system, targeting victim system	
Scenario	Running the following command on the infected system *.\crackmapexec.exe [target-IP] –u [username] –p [“password”] –lsa*

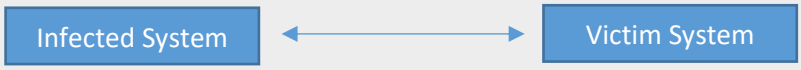
Infected System information	
Event log - Security	Event ID: **4658** - The handle to an object was closed **4688** - A new process has been created **4656** - A handle to an object was requested **4663** - An attempt was made to access an object New process name: [filepath]\crackmapexec.exe
Event log - Security	Event ID: **4690** - An attempt was made to duplicate a handle to an object Source process ID matches that of the previous events
Event log - Security	Event ID: **4656** - A handle to an object was requested Object name: [filepath]\logs\[Victim-IP]_[date].log Process name: [filepath]\crackmapexec.exe
Event log - Security	Event ID: **5158** - The Windows filtering platform has permitted a bind to a local port Application name: [filepath]\crackmapexec.exe
Event log - Security	Event ID: **5156** - The Windows filtering platform has permitted a connection Application name: [filepath]\crackmapexec.exe Destination address: [Victim IP] Destination port: 445
Event log - Security	Event ID: **4656** - A handle to an object was requested Object name: [filepath]\logs\[Victim IP].secrets Process name: [filepath]\crackmapexec.exe
Event log - Security	Event ID: **4656** - A handle to an object was requested. There were a number of these events identified that had read and write control accesses and then another set of events with delete accesses. Object name: C:\Users\[username]\AppData\Local\Temp_[8_char_string] C:\Users\[username]\AppData\Local\Temp_[8_char_string]\microsoft.vc90.crt.manifest C:\Users\[username]\AppData\Local\Temp_[8_char_string]\msvcr90.dll C:\Users\[username]\AppData\Local\Temp_[8_char_string]\msvcp90.dll C:\Users\[username]\AppData\Local\Temp_[8_char_string]\msvcm90.dll C:\Users\[username]\AppData\Local\Temp_[8_char_string]\python27.dll C:\Users\[username]\AppData\Local\Temp_[8_char_string]\unicodedata.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\bz2.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]_ctypes.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]_ssl.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\win32evtlog.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\win32api.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]_hashlib.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]_socket.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\select.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\pyexpat.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\Crypto.Cipher._DES.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\Crypto.Util._counter.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\Crypto.Cipher._AES.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\Crypto.Hash._SHA256.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\Crypto.Util.strxor.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\Crypto.Hash._MD4.pyd

	C:\Users\[username]\AppData\Local\Temp_[8_char_string]\Crypto.Random.OSRNG.winrandom.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\Crypto.Cipher._ARC4.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\Crypto.Cipher._DES3.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\gevent._semaphore.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\Crypto.Cipher._DES3.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\gevent._semaphore.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\greenlet.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\gevent._util.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\gevent.ares.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\gevent.core.pyd C:\Users\[username]\AppData\Local\Temp_[8_char_string]\pywintypes27.dll C:\Users\[username]\AppData\Local\Temp_[8_char_string]\include\pyconfig.h C:\Users\[username]\AppData\Local\Temp_[8_char_string]\crackmapexec.exe.manifest
Execution history - Prefetch	Prefetch file - C:\Windows\Prefetch\CRACKMAPEXEC.EXE-[prefetch_trailing_hex].pf

Victim System Information

Event log - Security	Event ID: **5156** - The Windows filtering platform has permitted a connection Source address: [Infected System IP] Destination address: [Victim System IP] Destination port: 445
Event log - Security	Event ID: **4624** - An account was successfully logged on Security ID: ANONYMOUS LOGON Account name: ANONYMOUS LOGON Logon process: NtlmSsp Authentication Package: NTLM
Event log - Security	Event ID: **4627** - Group membership information **4634** - An account was logged off Security ID: ANONYMOUS LOGON Account name: ANONYMOUS LOGON
Event log - Security	Event ID: **5140** - A network share object was accessed **5154** - A network share object was checked to see whether client can be granted desired access Source address: [Infected system IP] Share name: *\IPC$
Event log - Security	Event ID: **5154** - A network share object was checked to see whether client can be granted desired access Source address: [Infected system IP] Share name: *\ADMIN$ Share path: \??\C:\WINDOWS Relative target name: SYSTEM32\laZCxPea.tmp, JggjpEVh.tmp Accesses: ReadData, WriteData
Event log - Security	Event ID: **4656** - A handle to an object was requested **4659** - A handle to an object was requested with intent to delete Object name: C:\Windows\System32\laZCxPea.tmp, JggjpEVh.tmp

Angry IP scanner

Description	"Fast and friendly network scanner" - Purpose and functionality is to identify live systems via ping requests. Once a system is identified it will attempt to resolve the hostname along with mac address and will also attempt to scan ports.
Type	GUI
Target OS	Windows
Hashes	SHA1 - EC91544253C4254C290D9C027C63EB46E3C2756A MD5 - A85161524FA2A891EAF58C71D24F07A8
Attack usage	 Angry IP scanner could be deployed on the infected system and would be used to target other systems on the network. This would allow an attack to get a more in-depth understanding of the network and aid further attacks within the network.
Prerequisites	Java runtime environment

Installing Angry IP scanner on Infected system	
Scenario	Running the ipscan-[version*]-setup.exe executable on the infected system. *Version used in testing - 3.5.2

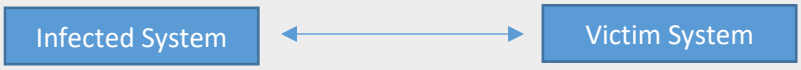
Log type and name	**Acquired information**
Event log - Security	Event ID: **4688** - A new process has been created **4689** - A process has exited Process name: [filepath]\ipscan.exe
Event log - Security	Event ID: **4690** - An attempt was made to duplicate a handle to an object Source process ID matches that of the previous events
Event log - Security	Event ID: **4688** - A new process has been created Process name: [filepath]\javaw.exe
Event log - Security	Event ID: **4656** - A handle to an object was requested **4663** - An attempt was made to access an object Process name: C:\Windows\Prefetch\IPSCAN.EXE-[prefetch_trailing_hex].pf
Execution history -Registry entries	HKEY_CLASSES_ROOT\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\microsoft.microsoftedge_8wekyb3d8bbwe\Children\001\Internet Explorer\DOMStorage\angryip.org HKEY_CLASSES_ROOT\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\microsoft.microsoftedge_8wekyb3d8bbwe\Children\001\Internet Explorer\EdpDomStorage\angryip.org HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\UFH\ARP HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Angry IP Scanner HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall\Angry IP Scanner HKEY_USERS\[SID]\Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\microsoft.microsoftedge_8wekyb3d8bbwe\Children\001\Internet Explorer\DOMStorage\angryip.org HKEY_USERS\[SID]\Software\Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\microsoft.microsoftedge_8wekyb3d8bbwe\Children\001\Internet Explorer\EdpDomStorage\angryip.org HKEY_USERS\[SID]\Software\JavaSoft\Prefs\ipscan HKEY_USERS\[SID]\Software\Microsoft\Windows\CurrentVersion\Search\JumplistData

HKEY_USERS\[SID]\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store
HKEY_USERS\[SID]\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\microsoft.microsoftedge_8wekyb3d8bbwe\Children\001\Internet Explorer\DOMStorage\angryip.org
HKEY_USERS\[SID]_Classes\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppContainer\Storage\microsoft.microsoftedge_8wekyb3d8bbwe\Children\001\Internet Explorer\EdpDomStorage\angryip.org

Initiating scan on Infected system, targeting victim system	
Scenario	With default settings on the Angry IP Scanner a scan was initiated, targeting the victim machine.

Infected & Victim System Information	
Event log - Security	Event ID: **5156** - The Windows filtering platform has permitted a connection Direction: Inbound & Outbound Source address: [Infected system IP] Destination address: [Victim system IP] With default settings there will be three Inbound & three outbound event logs seen as by default the number of ping probes to send is set to three

PsExec

Description	PsExec is part of the Sysinternals command line tools. PsExec is a legitimate tool that can be used to launch processes on remote systems on a network.
Type	Command line execution tool
Target OS	Windows
Hashes	SHA1 - E50D9E3BD91908E13A26B3E23EDEAF577FB3A095 MD5 - 27304B246C7D5B4E149124D5F93C5B01
Attack usage	 PsExec may be installed on a system for legitimate usage. As PsExec can be used to run processes on other systems, an attacker could utilise the tool for malicious purposes.

Running PsExec on Infected system	
Scenario	Opening a command prompt on the victim system remotely from the infected system.
Log type and name	**Acquired information**
Infected System Information	
Event log - Security	Event ID: **4688** - A new process has been created **4689** - A process has exited New process name: [filepath]\psexec.exe
Event log - Security	Event ID: **5156** - The Windows filtering platform has permitted a connection Direction: Outbound Source Address: [Infected system IP] Destination Address: [Victim system IP] Destination port: 445
Event log - Security	Event ID: **5158** - The Windows filtering platform has permitted a bind to a local port

	Application name: [filepath]\psexec.exe Source port: [#1]
Event log - Security	Event ID: **5156** - The Windows filtering platform has permitted a connection Application name: [filepath]\psexec.exe Direction: Outbound Source Address: [Infected system IP] Source port: [#1] (same port number as above, in event 5158) Destination Address: [Victim system IP]
Execution history -Registry entries	HKEY_USERS\[SID]\Software\Sysinternals\PsExec

Victim System Information	
Event log - System	Event ID: **7045** - A service was installed in the system Path: %SystemRoot%\PSEXESVC.exe
Event log - System	Event ID: **7036** - The service state has changed Execution: PSEXESVC
Event log - Security	Event ID: **4697** - A service was install in the system Path: %SystemRoot%\PSEXESVC.exe
Event log - Security	Event ID: **4688** - A new process has been created Process name: C:\Windows\PSEXESVC.exe
Event log - Security	Event ID: **5145** - A network share object was checked to see whether client can be granted desired access **5156** - The Windows filtering platform has permitted a connection Source address: [Infected system IP] Relative target name: PSEXESVC
Event log - Security	Event ID: **5140** - A network share object was accessed Source address: [Infected system IP] Share name: *\Admin$
Event log - Security	Event ID: **4648** - A logon was attempted using explicit credentials **4624** - An account was successfully logged on Process name: C:\Windows\PSEXESVC.exe Logon ID: [ID]
Event log - Security	Event ID: **4627** - Group membership information Logon ID: [ID] (same as above, in event 4624)
Event log - Security	Event ID: **4703** - A token right was adjusted Logon ID: [ID] (same as above, in event 4624)
Event log - Security	Event ID: **4688** - A new process has been created New process name: C:\Windows\cmd.exe Creator process name: C:\Windows\PSEXESVC.exe
Event log - Security	Event ID: **4656** - A handle to an object was requested **4663** - An attempt was made to access an object Object name: C:\Windows\Prefetch\PSEXESVC.EXE-[prefetch_trailing_hex].pf

Mitigation

A variety of mitigations will be of use in defending against the attacks detailed in this report:

Defend your organisation against spear-phishing, by taking a multi-layered approach.
See NCSC Guidance: https://www.ncsc.gov.uk/phishing

Protect your devices and networks by keeping them up to date: apply security patches promptly, use anti-virus and scan regularly to guard against known malware threats.
See NCSC Guidance: https://www.ncsc.gov.uk/guidance/mitigating-malware

Prevent and detect lateral movement in your enterprise networks.
See NCSC Guidance: https://www.ncsc.gov.uk/guidance/preventing-lateral-movement

Implement architectural controls for network segregation. This would help mitigate the exposure of the SMB issues described in the report.
See NCSC Guidance: https://www.ncsc.gov.uk/guidance/10-steps-network-security

Protect the management interfaces of your critical operational systems. In particular, use browse-down architecture to prevent attackers easily gaining privileged accesses to your most vital assets.
See NCSC blog post: https://www.ncsc.gov.uk/blog-post/protect-your-management-interfaces

Set up a security monitoring capability so you are collecting the data that will be needed to analyse network intrusions.
See NCSC Guidance: https://www.ncsc.gov.uk/guidance/10-steps-monitoring

Review and refresh your incident management processes.
See NCSC Guidance: https://www.ncsc.gov.uk/guidance/10-steps-incident-management