HelloKitty Ransomware Lacks Stealth, But Still Strikes Home

labs.sentinelone.com/hellokitty-ransomware-lacks-stealth-but-still-strikes-home/

Jim Walter

Game studio CD Projekt Red recently disclosed that it became a victim of a targeted, highly-impactful ransomware. In the days following the disclosure, it was revealed that the ransomware family most likely behind the attack was “HelloKitty”.

HelloKitty is a ransomware family that emerged in late 2020. While it lacks the sophistication of some of the more well-known families such as Ryuk, REvil, and Conti, it has nevertheless struck some notable targets, including CEMIG0. In this post, we analyse a recent HelloKitty sample and outline the basic behaviors and traits associated with this family of ransomware.

Execution and Behavior

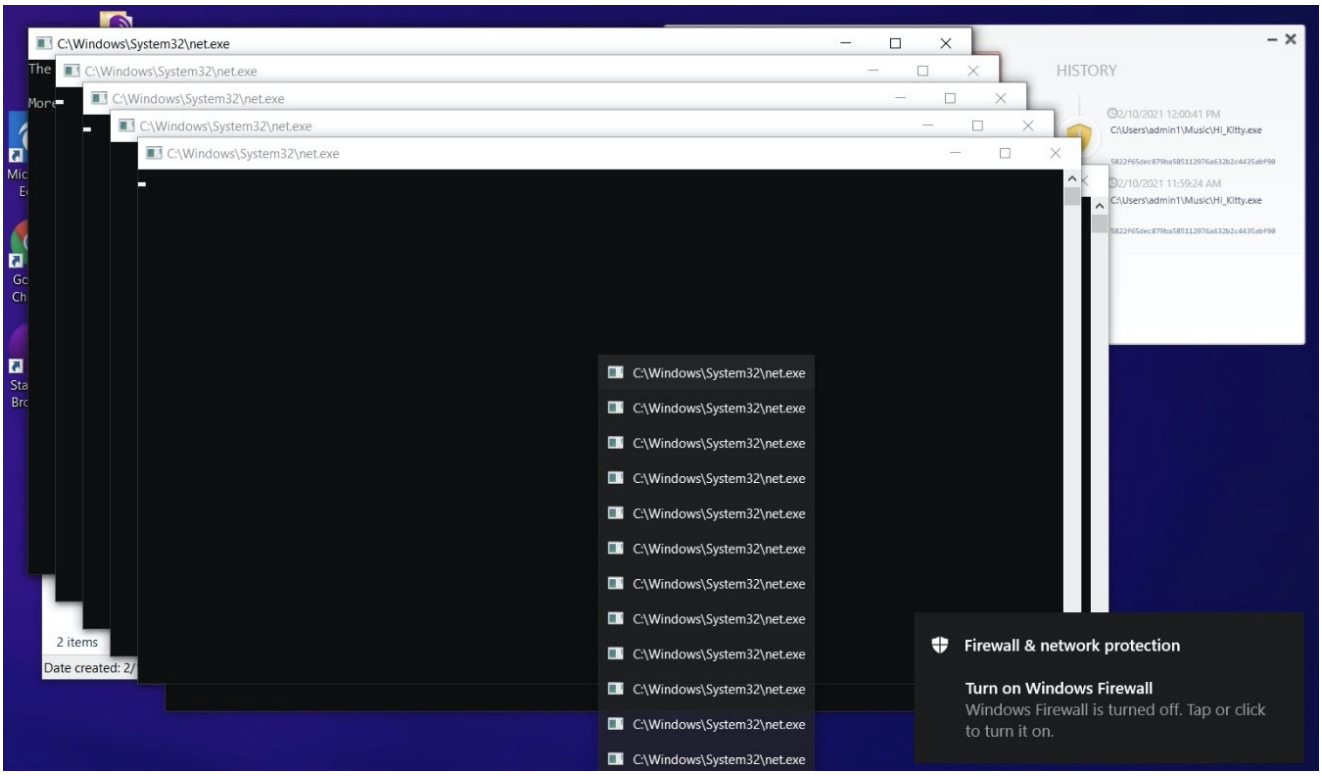
The “HelloKitty” name is based on internal mutex names, which are apparent upon execution.

```
00405a44 ff 15 48        CALL        dword ptr [->KERNEL32.DLL::SetErrorMode]
         20 41 00
00405a4a 83 3d 1c        CMP         dword ptr [DAT_0042741c],0x0                = ??
         74 42 00 00
00405a51 75 51           JNZ         LAB_00405aa4
00405a53 68 08 94        PUSH        u_HelloKittyMutex_00419408                  = u"HelloKittyMutex"
         41 00
00405a58 6a 00           PUSH        0x0
00405a5a 68 00 00        PUSH        0x100000
         10 00
```

While still somewhat unclear, current intelligence indicates that the primary delivery method of HelloKitty binaries is via phish email or via secondary infection in conjunction with other malware.

Once launched, HelloKitty will attempt to disable and terminate a number of processes and services so as to reduce interference with the encryption process. This includes processes and services associated with IIS, MSSQL, Quickbooks, Sharepoint, and more. These actions are carried out via `taskkill.exe` and `net.exe`.

In the analyzed sample, this is all done in a very non-stealthy manner. All spawned CMD windows are in the foreground and fully visible. This ‘lack of discreteness’ is atypical for modern ransomware, or any successful malware, for that matter.



A full list of processes from the analyzed sample are listed below:

```
dsa*
Ntrtsca
ds_moni
Notifie
TmListe
iVPAgen
CNTAoSM
IBM*
bes10*
black*
robo*
copy*
store.e
sql*
vee*
wrsa*
wrsa.ex
postg*
sage*
MSSQLServerADHelper100
MSSQL$ISARS
MSSQL$MSFW
SQLAgent$ISARS
SQLAgent$MSFW
SQLBrowser
ReportServer$ISARS
SQLWriter
WinDefend
mr2kserv
MSExchangeADTopology
MSExchangeFBA
MSExchangeIS
MSExchangeSA
ShadowProtectSvc
SPAdminV4
SPTimerV4
SPTraceV4
SPUserCodeV4
SPWriterV4
SPSearch4
IISADMIN
firebirdguardiandefaultinstance
ibmiasrw
QBCFMonitorService
QBVSS
QBPOSDBServiceV12
"IBM Domino Server(CProgramFilesIBMDominodata)"
"IBM Domino Diagnostics(CProgramFilesIBMDomino)"
"Simply Accounting Database Connection Manager"
QuickBooksDB1
QuickBooksDB2
QuickBooksDB3
QuickBooksDB4
QuickBooksDB5
QuickBooksDB6
```

```
QuickBooksDB7
QuickBooksDB8
QuickBooksDB9
QuickBooksDB10
QuickBooksDB11
QuickBooksDB12
QuickBooksDB13
QuickBooksDB14
QuickBooksDB15
QuickBooksDB16
QuickBooksDB17
QuickBooksDB18
QuickBooksDB19
QuickBooksDB20
QuickBooksDB21
QuickBooksDB22
QuickBooksDB23
QuickBooksDB24
QuickBooksDB25
```

Additional processes and services that are terminated are identified via PID. For example:

```
taskkill.exe /f /PID "8512"
taskkill.exe /f /PID "8656"
```

If HelloKitty is unable to stop any specific processes or services, it will leverage the Windows Restart Manager API to further assist in termination.

HelloKitty will also utilize WMI to gather system details and help identify running processes and any potentially problematic processes. This is done both by name and by PID. A number of examples are shown below:

```
start iwbemservices::execquery - rootcimv2 : select __path, processid, csname,
caption, sessionid, threadcount, workingsetsize, kernelmodetime, usermodetime,
parentprocessid from win32_process

start iwbemservices::execquery - rootcimv2 : select __path, processid, csname,
caption, sessionid, threadcount, workingsetsize, kernelmodetime, usermodetime,
parentprocessid from win32_process where (  caption = "store.exe")

start iwbemservices::execquery - rootcimv2 : select __path, processid, csname,
caption, sessionid, threadcount, workingsetsize, kernelmodetime, usermodetime,
parentprocessid from win32_process where (  caption = "wrsa.exe")

start iwbemservices::execquery - rootcimv2 : select __path, processid, csname,
caption, sessionid, threadcount, workingsetsize, kernelmodetime, usermodetime,
parentprocessid from win32_process where (  processid = 3036)

start iwbemservices::execquery - rootcimv2 : select __path, processid, csname,
caption, sessionid, threadcount, workingsetsize, kernelmodetime, usermodetime,
parentprocessid from win32_process where (  processid = 4460)

start iwbemservices::execquery - rootcimv2 : select __path, processid, csname,
caption, sessionid, threadcount, workingsetsize, kernelmodetime, usermodetime,
parentprocessid from win32_process where (  processid = 3052)

start iwbemservices::execquery - rootcimv2 : select __path, processid, csname,
caption, sessionid, threadcount, workingsetsize, kernelmodetime, usermodetime,
parentprocessid from win32_process where (  processid = 4476)

start iwbemservices::execquery - rootcimv2 : select __path, processid, csname,
caption, sessionid, threadcount, workingsetsize, kernelmodetime, usermodetime,
parentprocessid from win32_process where (  processid = 1560)

start iwbemservices::execquery - rootcimv2 : select __path, processid, csname,
caption, sessionid, threadcount, workingsetsize, kernelmodetime, usermodetime,
parentprocessid from win32_process where (  processid = 8124)

start iwbemservices::exe
```

Encryption and Ransom Note

Encryption is initiated and completed very quickly once applicable services and processes have been terminated. Specific encryption recipes and routines can vary across variants of HelloKitty. Generally speaking, they tend to use a combination of AES-256 & RSA-2048 or even NTRU+AES-128.

Once encrypted, affected files receive the `.crypted` extension.

Ransom notes are typically customized to directly reference the victim and victim's environment. Victims are instructed to visit a TOR-based payment and support portal. The following example has been sanitized:

It is also important to note that as of this writing, the onion address associated with HelloKitty ransom notes is not active.

`6x7dp6h3w6q3ugjv4yv5gycj3femb24kysgry5b44hhgfwc5ml5qrdad.onion`

Conclusion

HelloKitty may be easier to spot than other modern ransomware families, but upon execution it is no less dangerous. There are currently no known 'weaknesses' in the encryption routines, and there are no thirdy-party decrypters available for the HelloKitty ransomware.

Therefore, the only true defense is prevention. While this family does not appear to be actively leaking victim data at the moment, that could change at any point, in addition to them choosing to adopt some of the more recent extortion methods that go along with ransomware (DDoS).

Actors behind the more recent campaign(s) are reportedly attempting to auction the CD Projekt data off in various 'underground' forums. At present this sale of this data does appear to be legitimate. Time will tell if additional victim data is dealt with in the same way.

To protect yourself against HelloKitty, make sure you are armed with a modern Endpoint Security platform, which is configured correctly and up to date. The SentinelOne Singularity Platform is fully capable of preventing and detecting all malicious behaviors associated with the HelloKitty ransomware family.

IOCs

SHA1
fadd8d7c13a18c251ded1f645ffea18a37f1c2de

SHA256
501487b025f25ddf1ca32deb57a2b4db43ccf6635c1edc74b9cff54ce0e5bcfe

MITRE ATT&CK
Data from Local System – T1005
Modify Registry – T1112
Query Registry – T1012
System Information Discovery – T1082
Data Encrypted for Impact – T1486
File Deletion – T1070.004
Command and Scripting Interpreter: Windows Command Shell – T1059.003
Windows Management Instrumentation – T1047