Makop: The Toolkit of a Criminal Gang

By L M

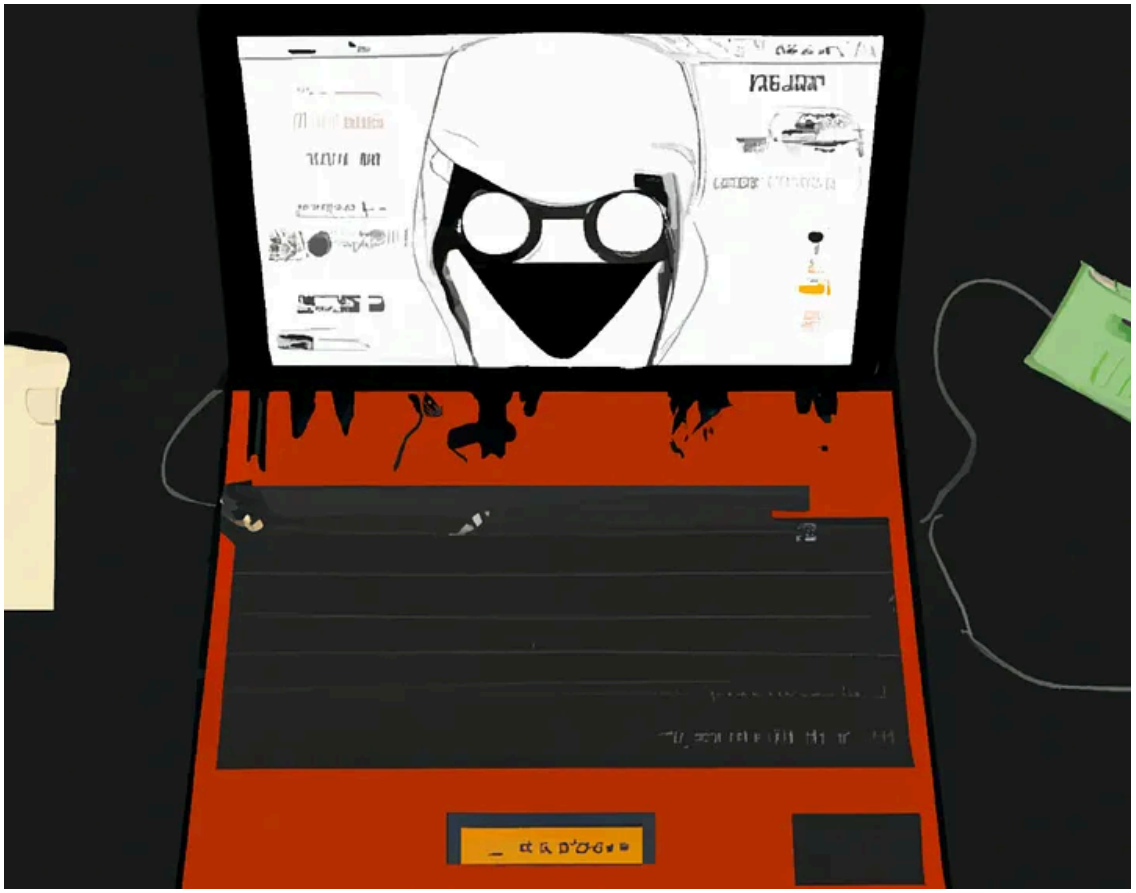
Published: 2023-04-07 · Archived: 2026-04-05 20:36:25 UTC

7 min read

Mar 12, 2023

Dissecting the malicious arsenal of the Makop ransomware gang.

Press enter or click to view image in full size



Executive summary

- Insights from a recent intrusion authored by Makop ransomware operators show persistence capability through dedicated .NET tools.
- Makop toolkit includes both off-the-shelf tools and custom-developed ones, including tools from the Chinese underground ecosystem.

- Makop gang did not conduct any significative retooling since 2020, which is a clear indicator of their effectiveness even after three years and hundreds of successful compromises.
- The gang leverages exposed remote administration services and internet-facing vulnerabilities to gain and maintain access to victim networks.

Introduction

The Makop ransomware operators started their infamous criminal business in 2020 leveraging a new variant of the notorious Phobos ransomware. During the last years, the gang maintained a solid presence in the criminal underground even if they did not join the double extortion practice.

Their operations are based on the human operator ransomware practice where most of the intrusion is handled by hands-on keyboard criminals, even in the encryption stage.

Makop ransomware gang is classified as a tier-B ransomware actor, but despite this, they keep hitting companies in Europe and Italy. Technical details of the Makop ransomware encryption tool have been greatly deepened by the Lifars security team (link), so, in this article, I am going to focus on other parts of the Makop gang arsenal leveraged to conduct digital extortions.

Technical Details

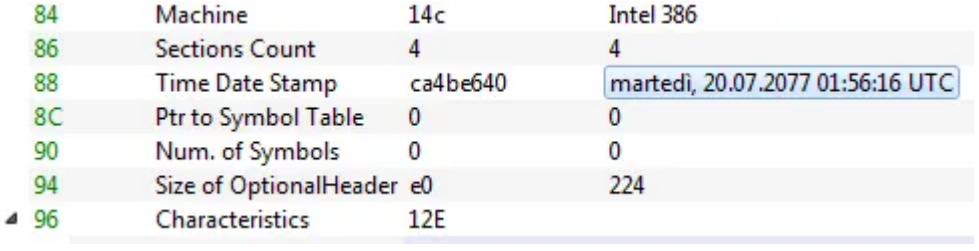
Makop ransomware operator arsenal is a hybrid one: it contains both cust-developed tools and off-the-shelf software taken from public repositories. In particular, recent investigations were able to identify four of them: the ARestore escalation tool, the backdoor, and other publicly available toolkits such as Advanced_Port_Scanner and a particular popular Chinese hack tool.

Custom tools

After the initial access, Makop criminals are still using an old tool dated back to their first operations in cyberspace. The “ARestore” tool is .NET executable built in 2020 and partially obfuscated. Also, the compilation time in the PE header looks time stomped, but the metadata from the .NET assembly modules reveal a more plausible date matching the time scale of the Makop operations.

filename: ARestore.exe

md5: 7f86b67ac003eda9d2929c9317025013

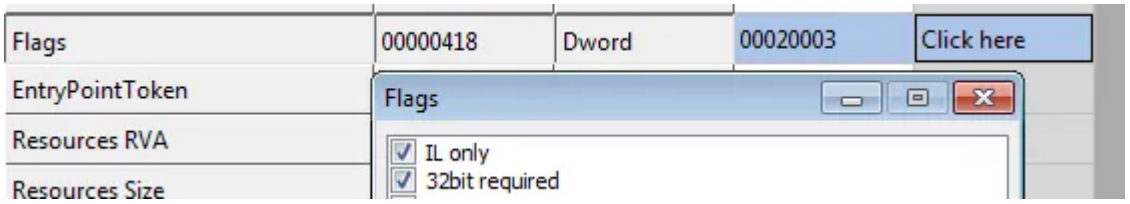


84	Machine	14c	Intel 386
86	Sections Count	4	4
88	Time Date Stamp	ca4be640	martedì, 20.07.2077 01:56:16 UTC
8C	Ptr to Symbol Table	0	0
90	Num. of Symbols	0	0
94	Size of OptionalHeader	e0	224
96	Characteristics	12E	

```
[assembly: RuntimeCompatibility(WrapNonExceptionThr
[assembly: AssemblyCopyright("Copyright ©  2020")]
[assembly: AssemblyTrademark("")]
[assembly: ComVisible(false)]
[assembly: AssemblyCompany("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyProduct("ARestore")]
```

Figure. Tampered PE timestamp (left) and .NET assembly copyright year (right)

The obfuscated part of the code is based on a switch-case state-machine looping and jumping through labels in the MSIL code. Despite this, the tool does not contain any evasion or anti-debugging techniques and contains IL-only, 32-bit code.




```
switch (num2)
{
case 0:
    goto IL_729;
case 1:
    num3 = 225 - 75;
    num = 334;
    if (!true)
    {
        goto IL_2D7F;
    }
    goto IL_B72;
case 2:
    goto IL_A5C;
case 3:
    array[15] = (byte)num3;
    num2 = 34;
    continue;
case 4:
    goto IL_55F;
case 5:
    goto IL_568E;
case 6:
    goto IL_35F9;
```

Figure. .NET flags (left) and obfuscation pattern (right)

The tool is designed for two main purposes: generating comb lists of local windows user names and potential passwords, and testing them locally. The tool is able to automatically retrieve local users from groups, filter for administration, and then test the password. The crooks currently use it after the initial access phase of their attack chain.

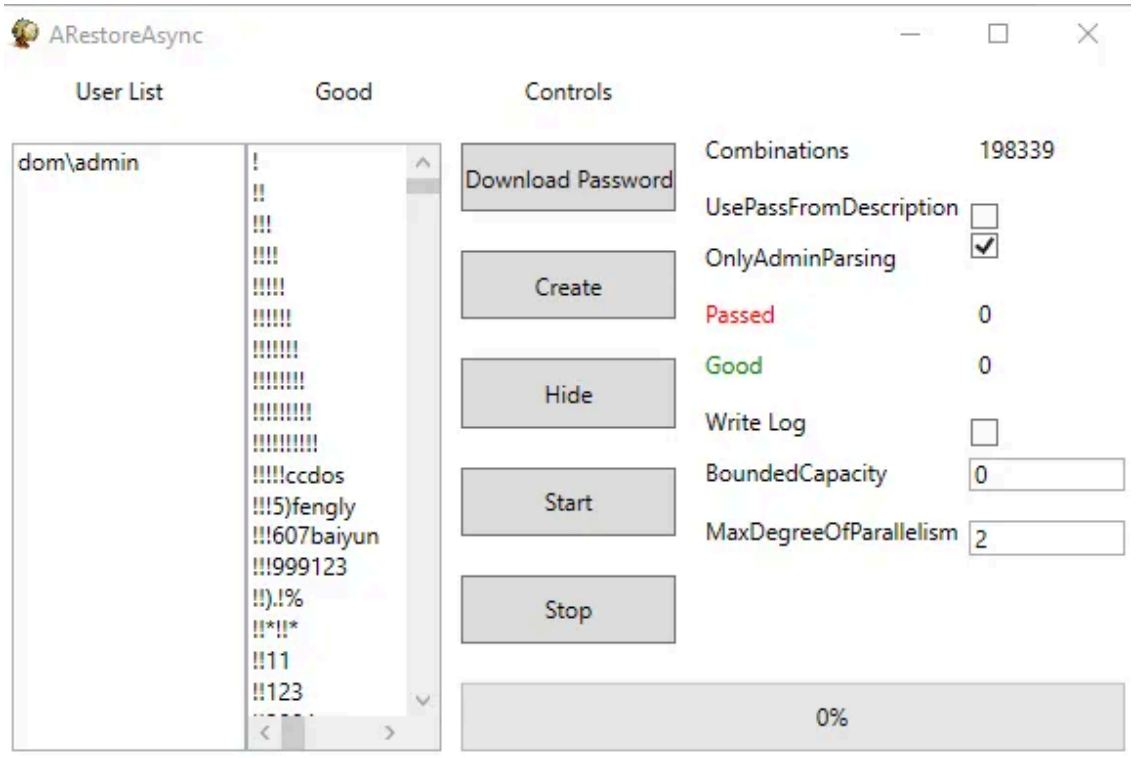
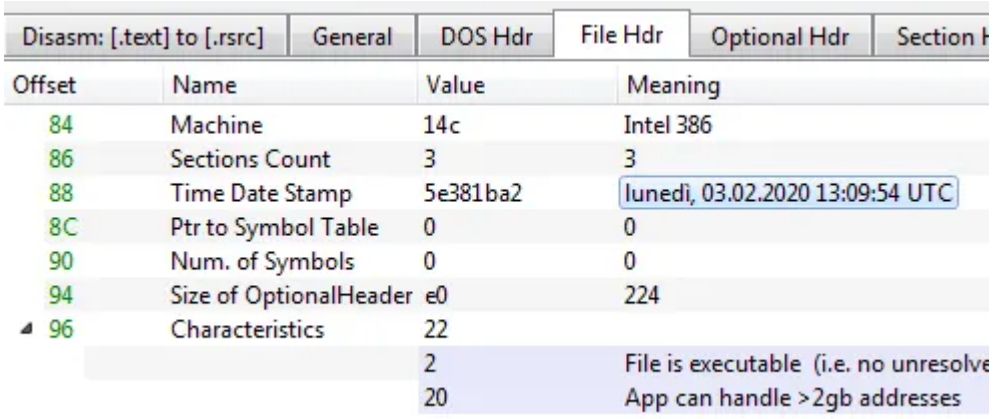


Figure. View of the “ARestore” tool

Makop operators also leverage other custom .NET assemblies to achieve further stages of the kill chain. For instance, they are using a particular persistence tool we name “PuffedUp” designed to ensure persistence after the initial access. Even this tool looks compiled and generated back in the early stage of the Makop operations and is still in use in current intrusions. Even this time the executable has been built in 2020, but it is not obfuscated at all.

filename:data.exe
md5:e245f8d129e8eadb00e165c569a14b71



```
private static void Main(string[] args)
{
    Console.WriteLine("припорфыв ");
    Console.WriteLine("раз два три");
    string location = Assembly.GetEntryAssembly().Location;
    program.jhasdas = Path.GetDirectoryName(location);
    program.Sasha();
}
```

Figure. Compilation timestamp (left), main routine (right)

During recent Makop intrusions, the tool has been coupled with another executable named “c.exe”, but, unfortunately, it has been erased by the attackers during the disengagement phase. Anyway, a quick look at the PuffedUp code reveals a plain logic to keep its execution persistent through a RUN registry key.

```
RegistryKey registryKey = Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\
  \CurrentVersion\\Run", true);
Console.WriteLine("четыре");
bool flag4 = !program.IsStartupItem();
if (flag4)
{
    registryKey.SetValue("VBCECompiler", Assembly.GetEntryAssembly().Location);
}
```

Figure. Run registry key setup in PuffedUp

Interestingly, the tool relies on a textual configuration file placed in the same folder. This particular file contains one or more 42 chars strings, that will be placed into the user clipboard. Apparently, a weird behavior that might make sense only with a more complete view of the Makop arsenal.

```
private static void Sasha()
{
    Console.WriteLine("припорфыв 1");
    string a = "";
    for (;;)
    {
        try
        {
            string text = "";
            foreach (string str in File.ReadAllLines(program.jhasdas + "/config.ini"))
            {
                text += str;
            }
```

Figure. PuffedUp configuration reading loop

Off-the-shelf tools

Makop ransomware operators extensively use off-the-shelf open-source and freeware tools to conduct lateral movement and system discovery. Along with the classical abuse of Microsoft SysInternal tools such as PsExec and other well-known open-source tools such as Putty and the never-missing Mimikatz, during recent operations, Makop abused even more peculiar software.

Get L M’s stories in your inbox

Join Medium for free to get updates from this writer.

Remember me for faster sign in

For instance “Advanced Port Scanner”, a freeware port scanning tool developed by the famous Radmin’s authors. The Makop criminals were recently using version 2.5.3869 of the tool, which dates back to 2019.

Advanced_Port_Scanner_2.5.3869.exe
md5: 6A58B52B184715583CDA792B56A0A1ED

The date of this particular version of the free software is particularly meaningful because it perfectly fits the build and compilation time of the other custom tools of the Makop intrusion arsenal. In fact, Makop criminals are still using tools built back in 2019 and 2020 to compromise small and medium enterprises around the world.

Press enter or click to view image in full size

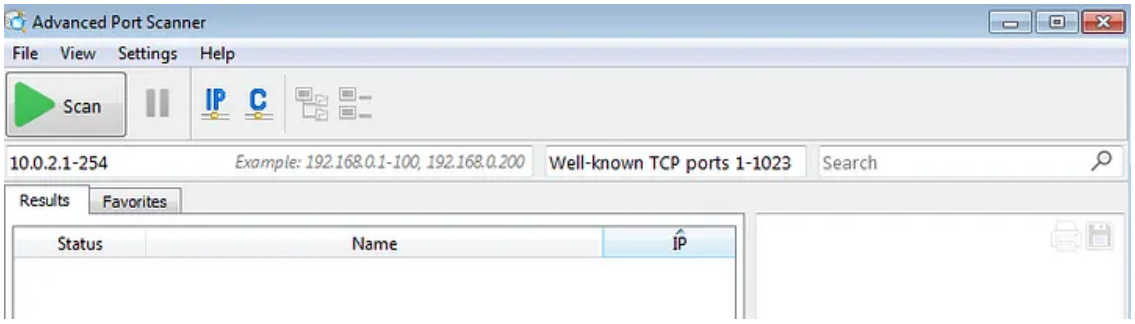


Figure. Advanced Port Scanner part of the Makop arsenal

Again, another tool in the Makop arsenal still dates to 2019: the “Everything” tool. Everything is freeware software maintained by Voidtools. As anticipated, the version abused by Makop ransomware operators in recent 2023 intrusions is still version 1.4.1.932, released in January 2019.

The tool is basically a search engine for local and network shared files inside a Windows environment: unlike the default Windows search, it is designed to locate files and folders by filename instantly, speeding up system information discovery.

Filename: Everything.exe
md5: b69d036d1dcfc5c0657f3a1748608148

The last tool interesting tool spotted in the Makop arsenal is a particular system administration tool rarely used in the Russian criminal underground. Its name is YDArk and it is an open-source tool available even on GitHub (link).

filename: YDArk.exe
md5: 9fd28d2318f66e4fe37a9a5bc1637928

Press enter or click to view image in full size

README.md

YDArk

免责声明: 这只是一个免费的软件, 如果您使用本软件, 给您直接或者间接造成损失、损害, 本人概不负责. 从您使用本软件的一刻起, 将视为您已经接受了本免责声明.

// 本软件加了VMProtect壳, 可能有些杀毒软件会报毒...请大家放心使用, 这属于杀毒软件误报.

// 本软件免费, 但未获得作者书面授权, 禁止用于商业用途; 另外禁止本软件用于恶意用途(比如作为病毒木马的一部分、破解网吧收费系统等等).

// 本软件仅限于学习交流，如侵权请在24小时进行删除.

//

Figure. YDArk GitHub page (source: GitHub)

YDArk is a powerful kernel manipulation tool that appeared in the Chinese underground communities back in 2020, where it was used to evade the memory scan of the anti-cheating program in gaming communities. The tool has been previously analyzed by SangYun Shin (link). YDArk can hide processes the rootkit way: at the kernel level. It manipulates the EPROCESS kernel object of the target process by changing its PID to 0 and redirecting forward and backward ActiveProcessLinks to the self's EPROCESS address.

Press enter or click to view image in full size

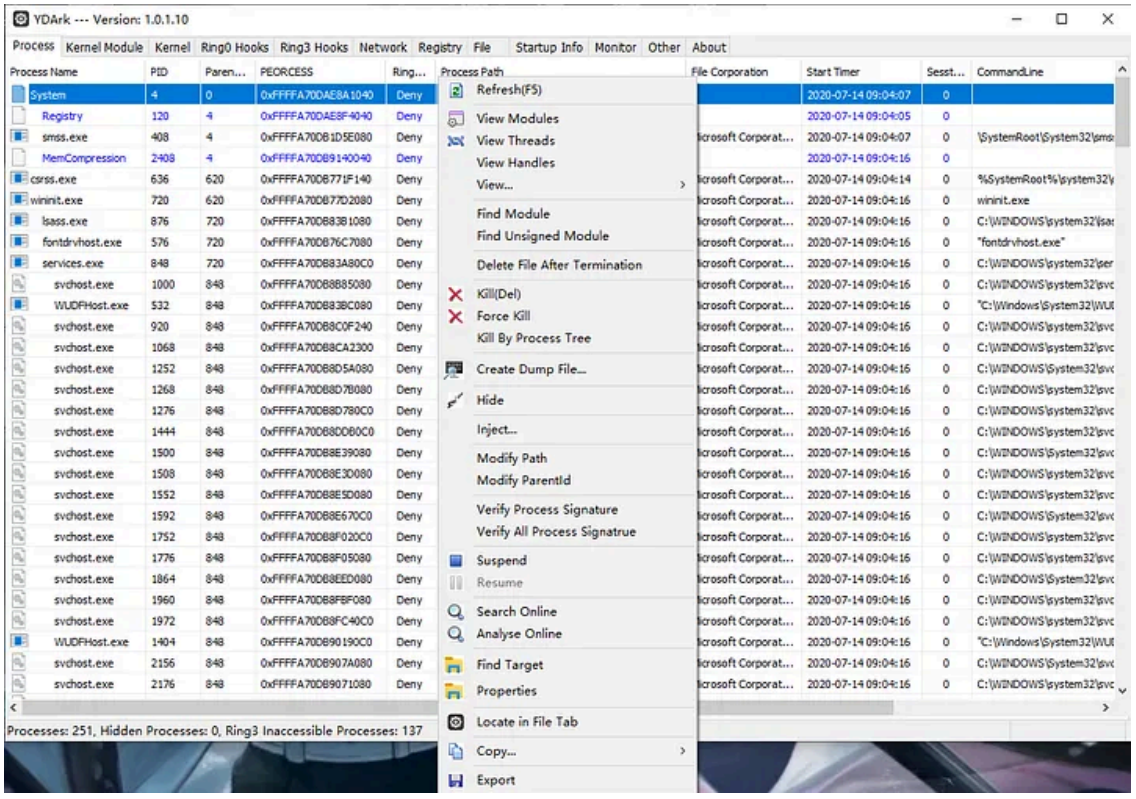


Figure. YDArk process hiding feature (source: GitHub)

The presence of this tool in the Makop arsenal is quite interesting because YDArk was previously found in other ransomware compromises:

- In April 2021 and December 2020, Sophos reported the abuse of YDArk in unspecified circumstances (link).
- In July 2022, Dark Lab security firm (link) reported the abuse of YDArk within a SonicWall SMA100 exploitation campaign aimed to leverage CVE-2019–7481 and CVE-2021–20028 on internet-exposed appliances to install Lockbit ransomware.

Conclusion

The Makop ransomware operators are conducting cyber extortion with a consistent cyber arsenal surviving detection for years. The absence of significative retooling in the Makop operator practice tells us the way to stop ransomware intrusion is still long.

If a tier-B human-operated ransomware gang targeting hundreds of companies worldwide does not need to update and change its arsenal after three years of operation it is a clear indication we are still lagging behind in enforcing an effective cyber attack deterrence strategy based on increasing the cost of attacks for cyber criminals and forcing them to retool.

The disclosure of the Makop cyber arsenal tools shall enable defenders to correlate even more intrusion attempts to the gang, to reach early detection of the abuse of both legit and custom-made tools.

Indicator of Compromise

Hash:

- 7f86b67ac003eda9d2929c9317025013 arestore.exe
- e245f8d129e8eadb00e165c569a14b71 data.exe
- 6A58B52B184715583CDA792B56A0A1ED Advanced_Port_Scanner_2.5.3869.exe
- b69d036d1dcfc5c0657f3a1748608148 Everything.exe
- 9fd28d2318f66e4fe37a9a5bc1637928 YDArk.exe

Yara Rules

```
import "pe"
rule PuffedUp{
meta:
author= "@luc4m"
date= "2023-03-12"
modified= "2023-03-12"
hash= "e245f8d129e8eadb00e165c569a14b71"
description="puffedup tool in makop ransomware toolkit"
tlp="CLEAR"
strings:
$main_1 = { 00 72 [4] 28 [4] 00 72 [4] 0A 72 [4] 28 [4] 00 29 }
$main_2 = { 0B 07 28 [4] 80 [4] 28 [4] 00 2A }
$sash_3 = { 72 [4] 0C [4] 72 [4] 0D 28 [4] 13 08 2C 06 }
```

```
$sash_4 = { 16 FE 01 13 0C 11 0C 2C 17 11 08 }
$sash_5 = { 1c 0D 00 20 [4] 28 [4] 00 00 DE 00 }
condition:
uint16(0) == 0x5a4d
and pe.imports("mscoree.dll")
and ( 2 of ($sash_*) or 1 of ($main_*) )
}
rule ARestore{
meta:
author= "@luc4m"
date= "2023-03-12"
modified= "2023-03-12"
hash= "7f86b67ac003eda9d2929c9317025013"
description="ARestore in makop ransomware toolkit"
tlp="CLEAR"
strings:
$junk_1= { 2B 09 28 [4] 14 16 9A 26 16 2D F9 14 2A }
$obj_1= { 38 [4] 26 20 [4] 38 [4] FE [4] 38 [4] 20 [4] 20 [4] 59 9C 20 [4] FE [4] 28 [4] 38 }
$obj_2= { FE [4] 20 [4] FE [4] 9C 20 [4] 38 [4] 12 }
$string_1 = "ADLogic" nocase
$string_2 = "GetUserFromGroupAsync" nocase
$string_3 = "WriteResultAsync" nocase
$string_4 = "ParseLoginAsync" nocase
$string_5 = "GenerateCredentials" nocase
$string_6 = "GetUserAsync" nocase
$string_7 = "IsAuthenticated" nocase
condition:
uint16(0) == 0x5a4d
and pe.imports("mscoree.dll")
and ( (1 of ($junk_*) or 1 of ($obj_*)) and 3 of ($string_*) )
}
```
