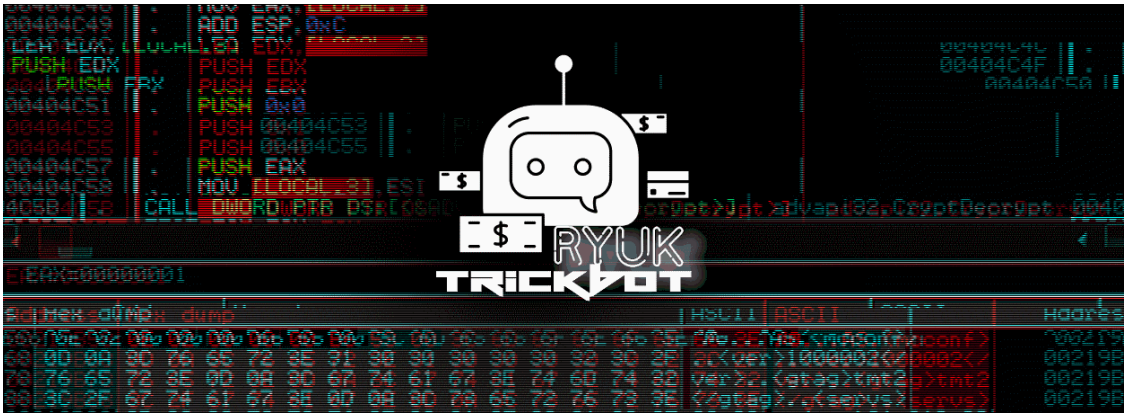
TrickBot malware uses obfuscated Windows batch script to evade detection

By Ax Sharma

Published: 2020-11-24 · Archived: 2026-04-05 14:37:38 UTC



With the 100th release of TrickBot, the malware came equipped with new and advanced evasive capabilities. One such capability is its use of an obfuscated batch script launcher to jumpstart malicious executables.

The fact that batch scripts need no interpreter but Microsoft Windows' inbuilt command prompt makes this evasion technique self-contained and minimalistic.

TrickBot deploys ransomware via obfuscated BAT scripts

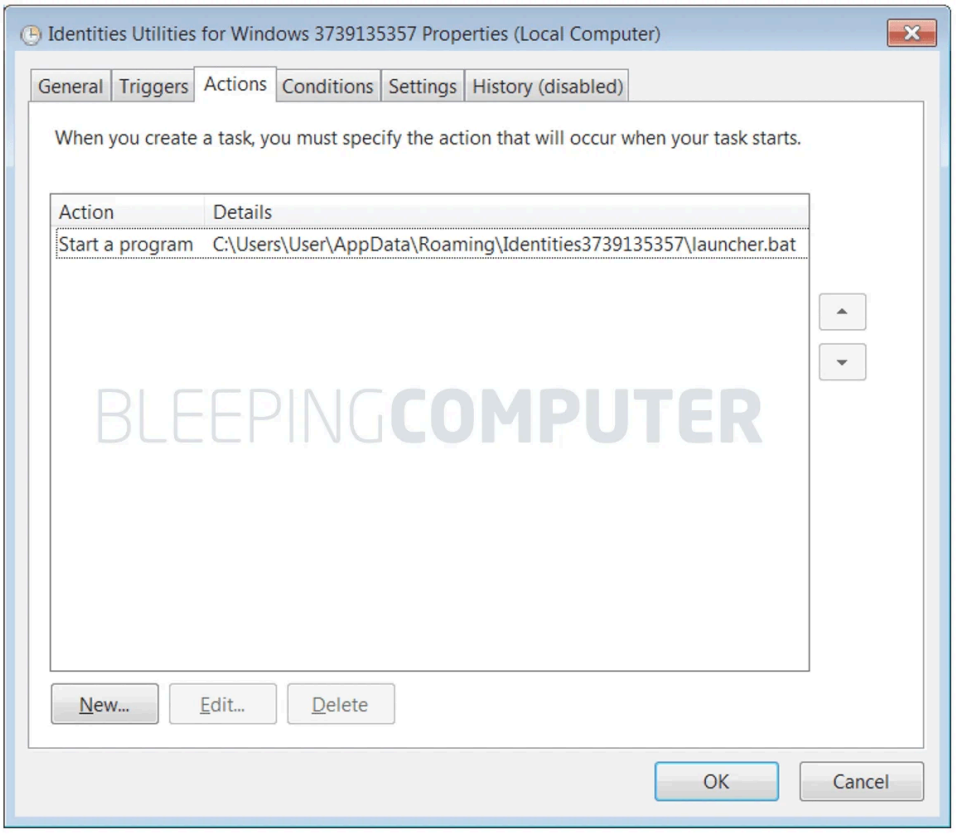
Over the weekend, BleepingComputer's Lawrence Abrams analyzed the hundredth build of TrickBot and its new features.


DIE LINKE.
0:00

TrickBot is a malware infection commonly installed via malicious phishing emails or other malware. When installed, TrickBot will quietly run on a victim's computer while it downloads other modules to perform different tasks.

TrickBot is known to finish an attack by giving access to threat actors who deploy either the Ryuk or Conti ransomware on the compromised network.

In our analysis, BleepingComputer had observed a BAT script *launcher.bat* being run by a scheduled task set up by TrickBot.

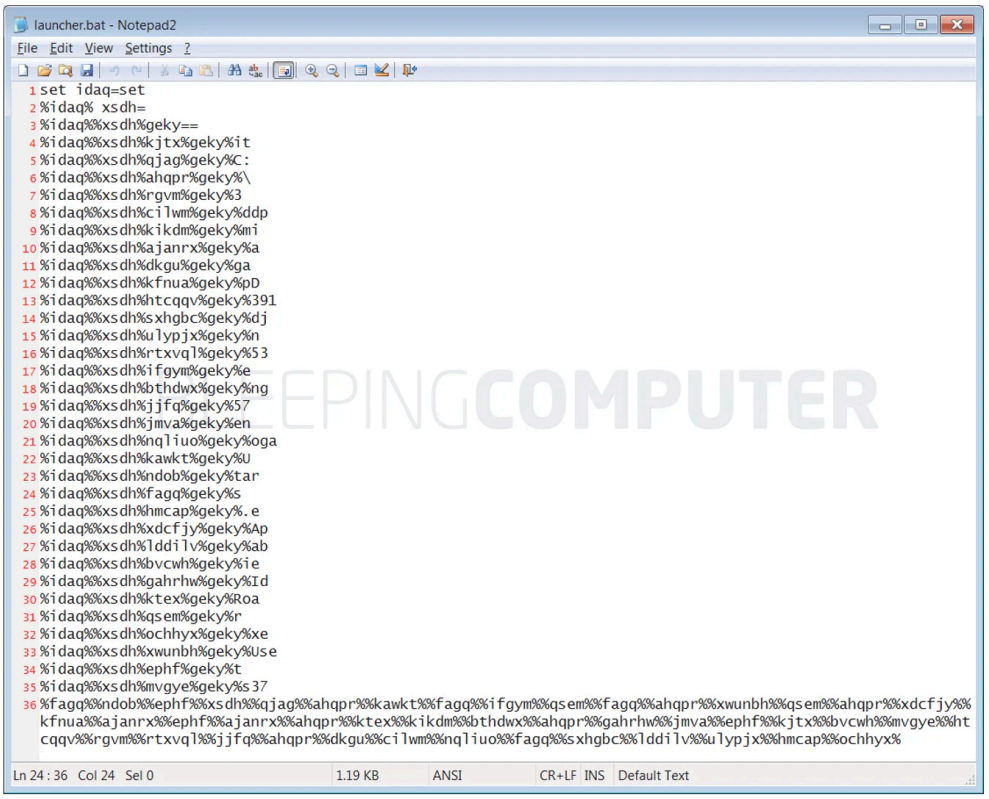


Scheduled task that runs *launcher.bat*
Source: BleepingComputer

Both the *launcher.bat* and the executable it launches are present in the same directory, as observed by BleepingComputer, whose location looks like:

```
C:\Users\(username)\AppData\Roaming\IdentitiesXXXXXXXXXX\
```

Yet, the use of an obfuscated batch script, shown below, to launch the executable is likely another feature to fly under the radar of enterprise security products.



The obfuscated batch script *launcher.bat* further runs the EXE payload
Source: BleepingComputer

Recently, researchers at Huntress Labs discovered another TrickBot sample that used a similar batch script with over 40 lines of obfuscated code.

When deciphered, all the code did was launch the malware, an action that could have been triggered by just a single line of code:

```
start C:\Users\[redacted]\AppData\Roaming\Identities1603031315\ulib8b4.exe
```

The binary in question, "ulib8b4.exe" is TrickBot's payload that performs a wide range of malicious activity, including stealing a domain's Active Directory Services database, spreading laterally on a network, screenlocking, stealing cookies and browser passwords, and stealing OpenSSH keys.

"System administrators often make use of batch scripting to make their lives easier and speed up their workflow," says John Hammond, Senior Security Researcher at Huntress Labs.

"But since this offers great access to the computer system, threat actors and malware families take advantage of *.bat* files just as well."

Hammond notes although antivirus products could easily scan plain-text batch scripts, the fact an attacker has gone through multiple steps to obfuscate a simple one-line command would make it virtually impossible for an "off-the-shelf" EDR or signature-based antivirus product to detect such samples.

Further, the signature detection can be avoided given there are various ways an attacker could obfuscate the same payload, each producing a different signature.

"On the surface, this code is completely unintelligible. It looks like random letters, in a random order, with random percent-signs thrown all around. But *cmd.exe* will interpret it and execute it, and that old-school shell is the tried and true built-in that hackers know will be on a target system," said Hammond.

Why are obfuscated batch scripts uniquely a problem?

BleepingComputer asked Hammond, considering obfuscation techniques are not limited to batch scripts why was the use of BAT files in malware uniquely a problem.

In other words, NodeJS files and Python scripts that contain plain text code, rather than binary data, could be just as well obfuscated.

Hammond told BleepingComputer, "You're absolutely right—it could very well have been any file or any different language of code. I think the most interesting gimmick with the BAT/cmd.exe script is that it is native and inherent to a Windows operating system, so it doesn't need any external compiler or some other means to get the code to execute on the target."

Furthermore, the researcher told us, since all of the characters in the batch script were ASCII printable characters, rather than binary code, it was easier to transmit the script over the wire while bypassing the scrutiny of antivirus programs.

"We talk a lot about 'live-off-the-land binaries' and this is a peculiar one because it is not so much a 'binary,' but a trick to sort of weaponizing one."

"And of course, with all the characters being ASCII printable characters, this snippet can be easily sent over the wire, and since there aren't any glaring 'bad strings' or malicious signatures, an EDR or AV program could overlook it," the researcher told BleepingComputer.

Huntress Labs' detailed insights on the obfuscation technique can be found in their report.

An improved version of this obfuscation technique has also been demonstrated by Hammond on YouTube.
