

Uncovering cyber threat actors' persistence mechanisms with **Belkasoft**



After [gaining access](#) to a target network, bad actors usually set their sight on a new goal—establishing a persistent presence in the network and maintaining it. Why is this phase so important to attackers?

In this whitepaper, which is a continuation of Belkasoft's Digital **Incident Response** series, we will examine the persistence mechanisms—in terms of programs, files, and keys—that are widely used in cyberattacks targeted at Windows devices.

Note: Our **Incidence Response** series commenced with the [Hunting for Initial Infection Vector with Belkasoft](#) whitepaper. The [Tracking Potentially Malicious Files with Evidence of Execution Using Belkasoft](#) paper came next.

Persistence

Persistence

The adversary is trying to maintain their foothold.

Persistence consists of techniques that adversaries use to keep access to systems across restarts, changed credentials, and other interruptions that could cut off their access. Techniques used for persistence include any access, action, or configuration changes that let them maintain their foothold on systems, such as replacing or hijacking legitimate code or adding startup code.

Figure 1: Persistence definition from MITRE ATT&CK

Answer to the question: *Why do attackers try to establish a persistent presence in an already compromised network?*

For one, an initially compromised host may be rebooted—and this means access might get cut off. Bad actors use persistence techniques to avoid losing access to systems or networks.

For another, attackers' actions may be detected, so it makes sense for them to utilize multiple persistence mechanisms even on different hosts (compromised during lateral movement phase). This way, they might even manage to obtain redundant access capabilities.

Invariably, hunting for persistence mechanisms is the fastest and most convenient way to uncover malware, some of which might even be undetectable—at the time of an investigation, at least.

These are persistence mechanisms used by regular malware authors and (sometimes) advanced bad actors:

- Run keys
- Startup folders
- Scheduled tasks
- Services

You can view every one of them under the **Persistence** node, which you can find on the **Incident Investigations** tab in **Belkasoft Evidence Center X (Belkasoft X)**.

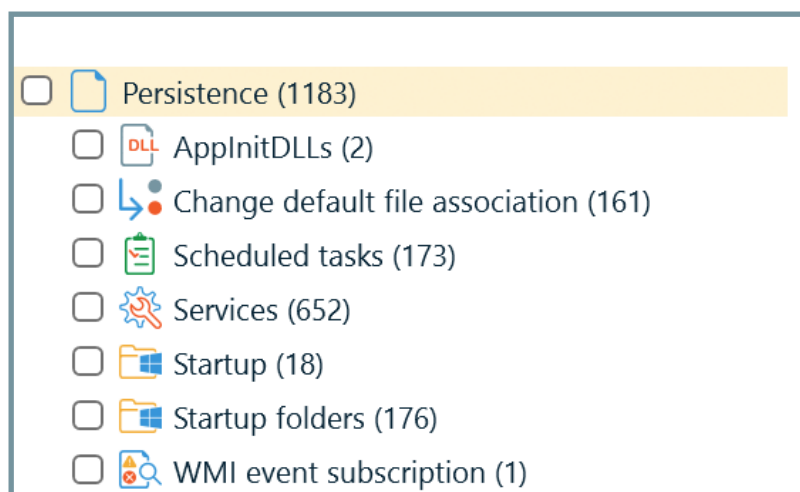


Figure 2: Persistence node on the Incident Investigations tab in Belkasoft X

Run keys, Startup folders

Run keys are the keys that force programs to automatically run each time a user logs into Windows. Similarly, Startup folders are the folders that house programs that are automatically initialized when a user logs into Windows.

Therefore, a bad actor can achieve persistence by referencing a program with a registry run key or adding the program to a startup folder. In such scenarios, the programs will be executed under the context of a user and they will be assigned permissions level associated with the user account involved.

When you use Belkasoft, information extracted from Run keys is placed under the **Startup** node, while items from Startup folders are grouped under the **Startup folders** node.

As you can see on Figure 2, there are few records under Startup compared to Startup folders—and this is how things are usually. So, Startup is one of the best places you should begin checking for traces of persistence associated with malicious actors.

In our sample case, we have a suspicious entry with this name: “hdeqcrc”. After examining the data, we easily extrapolated that it was used to start a QBot binary from

C:\Users\IEUser\AppData\Roaming\Microsoft\Cpdfx oatpg through PowerShell:

Item properties	
General	
Key path	Software\Microsoft\Windows\CurrentVersion\Run
Last write time (UTC)	11-Apr-20 3:39:09 PM
Application	hdeqrc
Application data	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe "\$windowsupdate = \"C:\Users\IEUser\AppData\Roaming\Microsoft\Cpdfxatpg\egvmxii.exe\"; & \$windowsupdate"

Figure 3: Qbot binary persistence mechanism uncovered with Belkasoft X

The key in view lies in `Software\Microsoft\Windows\CurrentVersion\Run`. It exists under HKLM (accessible to all users) and HKCU (accessible to a currently logged-in user) parts of the Windows registry.

While this registry key is typically used by legitimate programs to initialize processes automatically when users log into computers, it also sees a lot of use—as we have just shown—in malware.

Now, when it comes to Startup folders, we have to emphasize that the bad actors may not stop at putting malicious executable itself in an item; they might also use a [shortcut \(.LNK\) for the file](#).

Scheduled tasks

At this point, we will examine the next most common persistence mechanism—scheduled tasks.

Scheduled tasks are tasks that force Windows to launch programs or runs scripts at predefined times. They allow adversaries to start malicious executables or bad scripts (initial or recurring).

In most cases, there are several scheduled tasks. From Figure 2, you may have noticed that there are 173 different tasks. Some tips come in handy here.

For one, you will do well to focus on those tasks that exist in `C:\Windows\System32\Tasks`—because that is the most common location for malicious tasks. For another, you should try using their creation times to uncover the most recent of the lot. You can do this by sorting by time column in Belkasoft X.

Through the described tips, we managed to uncover another QBot's persistence mechanism:

Item properties	
General	
Task ID	{C1116BC4-4CB6-413E-94F0-D8EF2C386E81}
Task name	\{53F3B560-FC76-45B4-8A10-8DE362DFCDD5}
Hash	8DF800A4B19F1AEF0F951BA943EB03DA09E2B8EC9F88731F0DC6C20A9D082610
The author of the document	MSEDGEWIN10\IEUser
URI	\{53F3B560-FC76-45B4-8A10-8DE362DFCDD5}
Triggers	TimeTrigger
Actions	03000C00000041007500740068006F0072006666000000007200000043003A005C00570069006E0064006F00770073005C00530079007300740065006D00330032005C00570069006E0064006F007700730050006F007700650072005300680065006C006C005C00760031002E0030005C0070006F007700650072007300680065006C006C002E00650078006500D200000022002400770069006E0064006F007700730075007000640061007400650020003D0020005C00220043003A005C00550073006500720073005C004900450055007300650072005C0041007000700044006100740061005C0052006F0061006D0069006E0067005C004D006900630072006F0073006F00660074005C00430070006400660078006F0061007400700067005C006500670076006D007800690069002E006500780065005C0022003B002000260020002400770069006E0064006F00770073007500700064006100740065002200000000000000
Dynamic info	11-Apr-20 6:52:49 PM
Schema	0x00010002 (65538)
Date	2020-04-11T08:52:49
User Id	MSEDGEWIN10\IEUser
Logon type	InteractiveToken
Run level	LeastPrivilege
Command	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Arguments	"\$windowsupdate = \"C:\Users\IEUser\AppData\Roaming\Microsoft\Cpdfxoaatpg\egvmxii.exe\"; & \$windowsupdate"
White list	Third-Party

Figure 4: QBot malware utilizing a scheduled task as a persistence mechanism

Here too, like in the previous example, PowerShell was used to run a malicious binary.

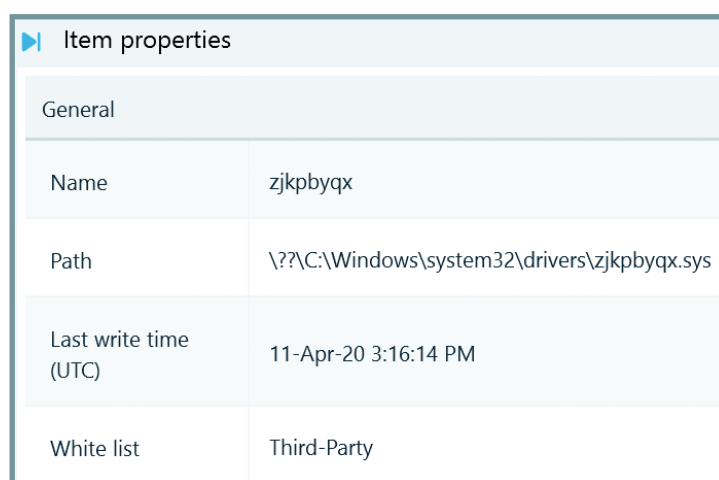
Services

In Windows, services are the core components that help with the creation and management of long-running processes. A windows service corresponds to a specific type of application that is configured to launch and operate in the background. Some services get to run even before a user logs into Windows.

Services are implemented with the services API and they handle low-level tasks that require little or no interaction with users. The major difference between services and other programs is that services typically run in the background (unnoticed) and they lack interfaces that users can click on or use.

For the stated reasons, bad actors take advantage of services all too often. The same tips we provided earlier on identifying malicious items apply here.

At this point, after we examined recently created services, we found a suspicious service due to its strange name, which is somewhat similar to previously identified malware:



The image shows a screenshot of the 'Item properties' window in Windows. The window has a title bar with a blue play button icon and the text 'Item properties'. Below the title bar is a tab labeled 'General'. The main content area is a table with the following data:

Name	zjkpbyqx
Path	\\??\C:\Windows\system32\drivers\zjkpbyqx.sys
Last write time (UTC)	11-Apr-20 3:16:14 PM
White list	Third-Party

Figure 5: A suspicious service

Belkasoft X has a built-in file system browser, so we used it to examine the suspicious file. We also exported it to perform deeper analysis tasks on it. Furthermore, since we already set things up in the processing stage, we obtained the file's hash sum.

Item properties	
General	
Name	zjkpbypqx.sys
Created (UTC)	11-Apr-20 3:16:14 PM
Modified (UTC)	11-Apr-20 3:16:14 PM
Access time (UTC)	11-Apr-20 3:20:01 PM
Entry changed (UTC)	11-Apr-20 3:16:14 PM
MFT created (UTC)	11-Apr-20 3:16:14 PM
MFT modified (UTC)	11-Apr-20 3:16:14 PM
MFT access time (UTC)	11-Apr-20 3:16:14 PM
MFT entry changed (UTC)	11-Apr-20 3:16:14 PM
File size (bytes)	72816
Md5	Not calculated
Sha1	Not calculated
Sha256	Not calculated
Offset (bytes)	15821750272
Is Deleted	No

Figure 6. A suspicious file's properties

After further analysis, we figured out that the file is signed by Microsoft. Windows Defender actually created the file—Boot Time Removal Tool. So, while the file may share a name similar to previously uncovered threats, it is not malicious.

In any case, the list of persistence mechanisms (and their descriptions) provided is not exhaustive. We only examined the most common mechanisms, so there are other things you must look out for.

Threats constantly evolve; bad actors are becoming more creative. For example, some attackers may use WMI Event Subscription to establish persistence on a target host.

[“WMI is the Microsoft implementation of Web-Based Enterprise Management \(WBEM\), which is an industry initiative to develop a standard technology for accessing management information in an enterprise environment. WMI uses the Common Information Model \(CIM\) industry standard to represent systems, applications, networks, devices, and other managed components.”](#)

A WMI Event Subscription triggers an action when an event occurs. In persistence, bad actors take advantage of the WMI event subscription technique to permanently bind a specific action to a Windows event.

Since exploitation frameworks—such as PowerShell Empire and Koadic—allow such procedures automatically, those frameworks see more use these days, especially in ransomware attacks.

Here is an example of the persistence mechanism in view created with Koadic:

Item properties	
General	
Name	K0adic
Filter	K0adic
Query	SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 300
Type	CommandLineEventConsumer
Arguments	C:\Windows\system32\mshta.exe C:\ProgramData\TRGBWFPQDH.hta

Figure 7: WMI Event Subscription example

Conclusion

It is easy to see that an uncovered persistence mechanism is an invaluable pivot point from which further investigations on threats can be conducted.

When it comes to attacks, persistence is hardly ever the final or most important phase. However, if persistence is not prevented or if this phase is not detected early enough, then networks and systems can be expected to lose some data to adversaries.

Incident responders and forensic analysts must have the means to obtain and analyze important information quickly. After all, time is always of the essence in incident response.

By using the right tool—such as [Belkasoft X](#)—investigators and researchers can cut down on time wasted on manual tasks—and this means they get to act fast to implement remediation.

Belkasoft X is a digital forensics and cyber incident response tool which significantly reduces the time needed to find, track, and examine forensics artifacts and digital footprints. Belkasoft solutions for corporate customers are designed to help you deal with various data breaches and respond in a quick manner so that to prevent data losses in future.



For more information contact us
at sales@belkasoft.com

TRY FREE AT
<https://belkasoft.com/trial>
30 days trial

Request a quote at
<https://belkasoft.com/pricing>

702 San Conrado Terrace, Unit 1
Sunnyvale CA 94085
+1 650-272-0384 (USA)
<https://belkasoft.com>