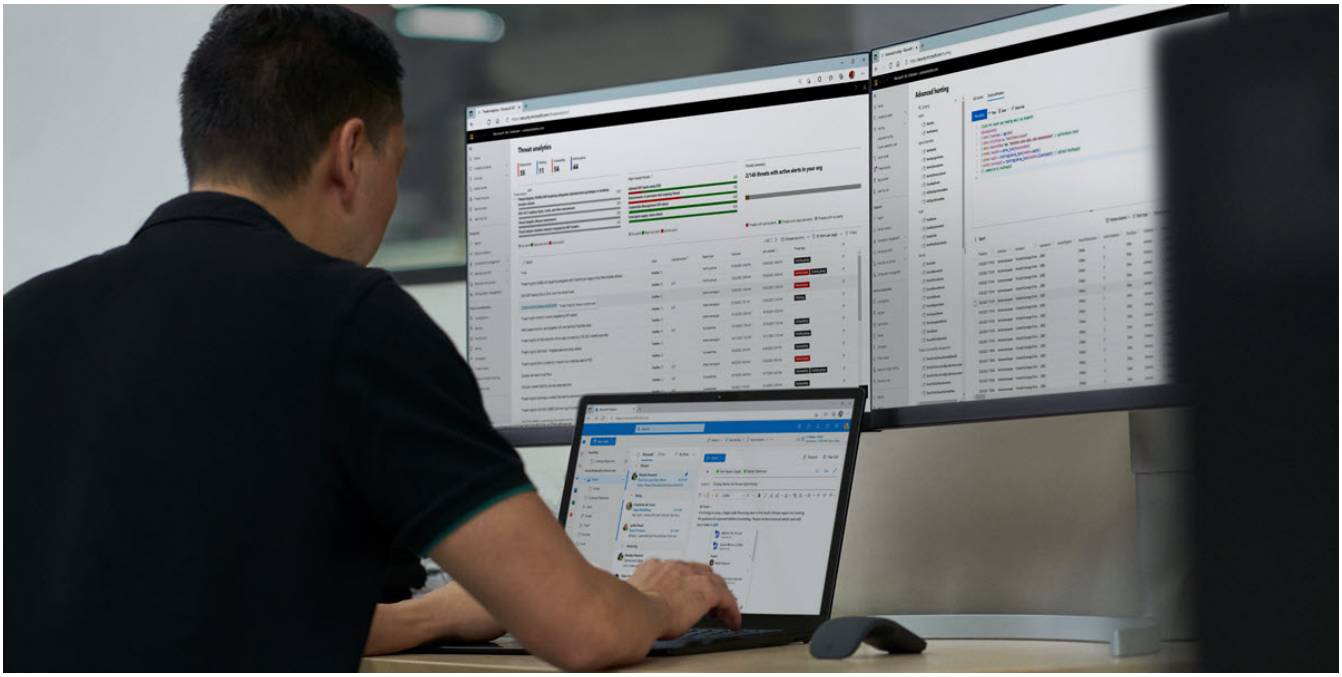
Tarrask malware uses scheduled tasks for defense evasion

microsoft.com/security/blog/2022/04/12/tarrask-malware-uses-scheduled-tasks-for-defense-evasion/

April 12, 2022



As Microsoft continues to track the high-priority state-sponsored threat actor HAFNIUM, new activity has been uncovered that leverages unpatched zero-day vulnerabilities as initial vectors. The Microsoft Detection and Response Team (DART) in collaboration with the Microsoft Threat Intelligence Center (MSTIC) identified a multi-stage attack targeting the Zoho Manage Engine Rest API authentication bypass vulnerability to initially implant a Godzilla web shell with similar properties detailed by the Unit42 team in a previous blog.

Microsoft observed HAFNIUM from August 2021 to February 2022, target those in the telecommunication, internet service provider and data services sector, expanding on targeted sectors observed from their earlier operations conducted in Spring 2021.

Further investigation reveals forensic artifacts of the usage of Impacket tooling for lateral movement and execution and the discovery of a defense evasion malware called Tarrask that creates “hidden” scheduled tasks, and subsequent actions to remove the task attributes, to conceal the scheduled tasks from traditional means of identification.

The blog outlines the simplicity of the malware technique Tarrask uses, while highlighting that scheduled task abuse is a very common method of persistence and defense evasion—and an enticing one, at that. In this post, we will demonstrate how threat actors create scheduled tasks, how they cover their tracks, how the malware’s evasion techniques are used to maintain and ensure persistence on systems, and how to protect against this tactic.

Right on schedule: Maintaining persistence via scheduled tasks

Windows Task Scheduler is a service that allows users to perform automated tasks (scheduled tasks) on a chosen computer for legitimate administrative purposes (e.g., scheduled updates for browsers and other applications).

Throughout the course of our research, we’ve found that threat actors commonly make use of this service to maintain persistence within a Windows environment.

We’ve noted that the Tarrask malware generates several artifacts upon the creation of a scheduled task, whether using the Task Scheduler GUI or the schtasks command line utility. Profiling the use of either of these tools can aid investigators in tracking this persistence mechanism.

The following registry keys are created upon creation of a new task:

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache**Tree*TASK_NAME***
- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache**Tasks\{*GUID*}**

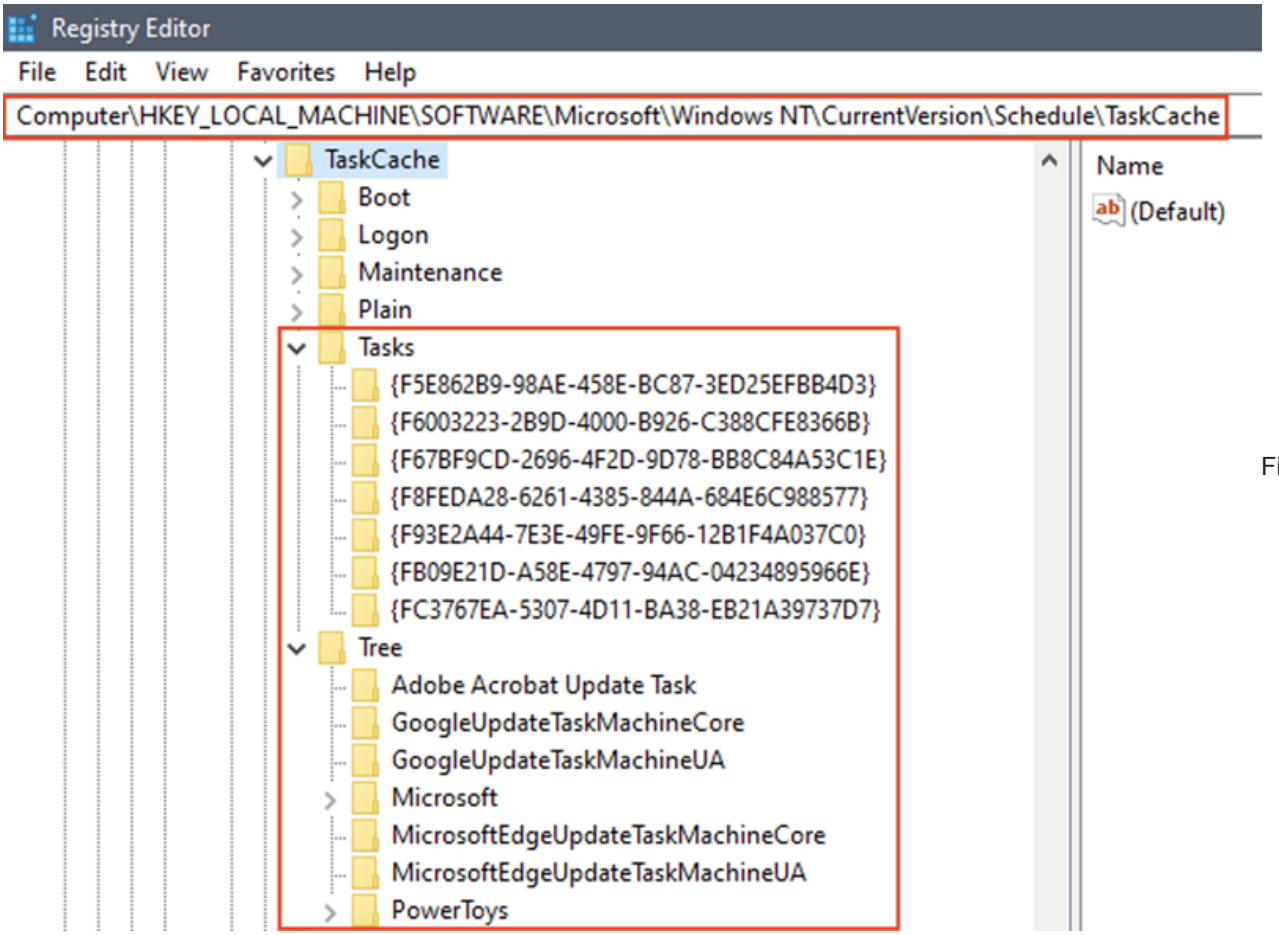
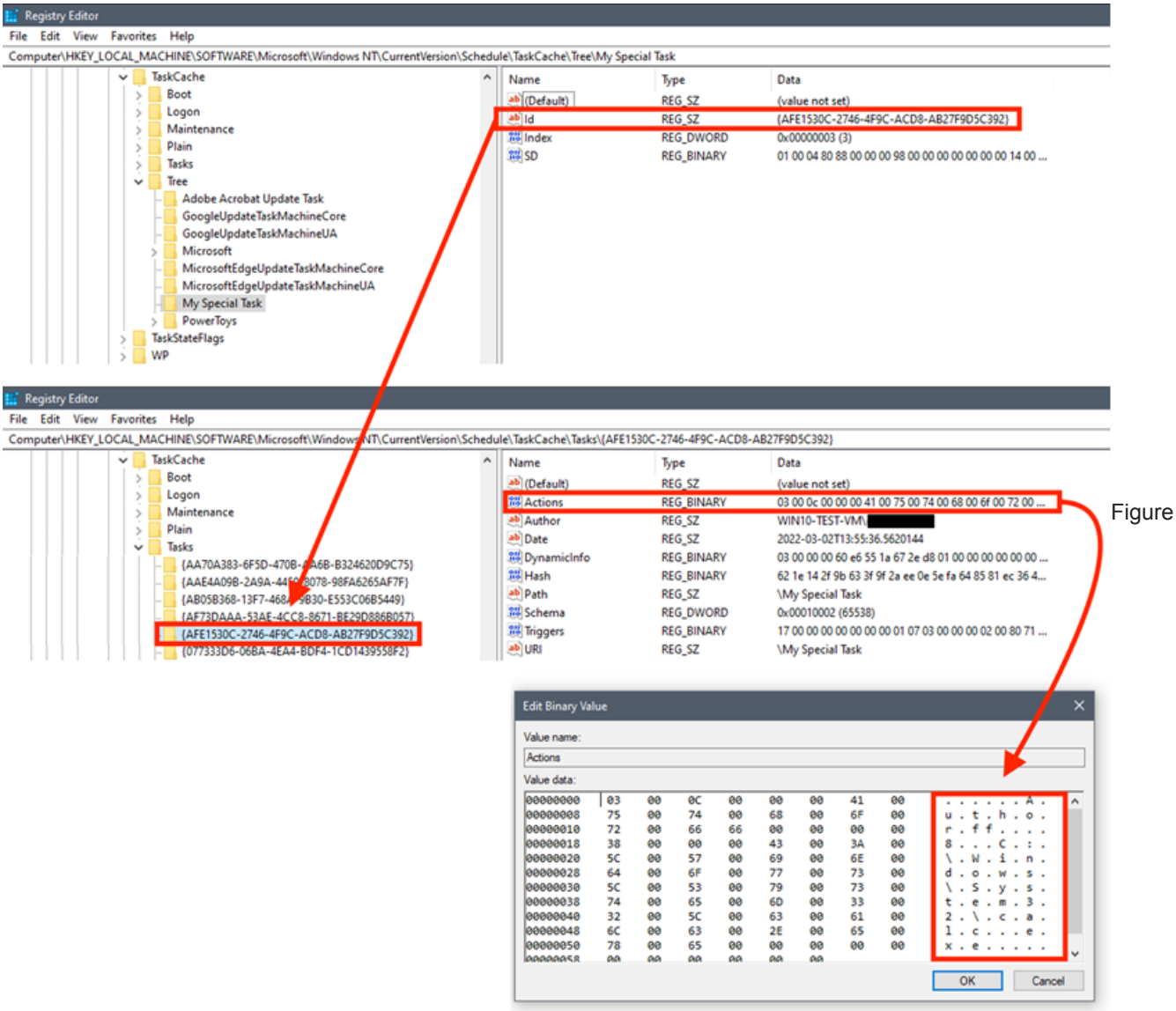


Figure 1. Tarrask malware creates new registry keys along with the creation of new scheduled tasks

The first subkey, created within the **Tree** path, matches the name of the scheduled task. The values created within it (Id, Index, and SD) contain metadata for task registration within the system. The second subkey, created within the **Tasks** path, is a GUID mapping to the **Id** value found in the **Tree** key. The values created within (Actions, Path, Triggers, etc.) contain the basic parameters necessary to facilitate execution of the task.

To demonstrate the value in the artifacts generated, shown in the following figures, we have created “My Special Task” which is set to execute the binary “C:\Windows\System32\calc.exe” on a regular interval.



Figure

2. XML file matches name of the task

Similar information is also stored within an extensionless XML file created within *C:\Windows\System32\Tasks*, where the name of the file matches the name of the task. This is displayed in Figure 2, where we name the task “My Special Task” as an example.

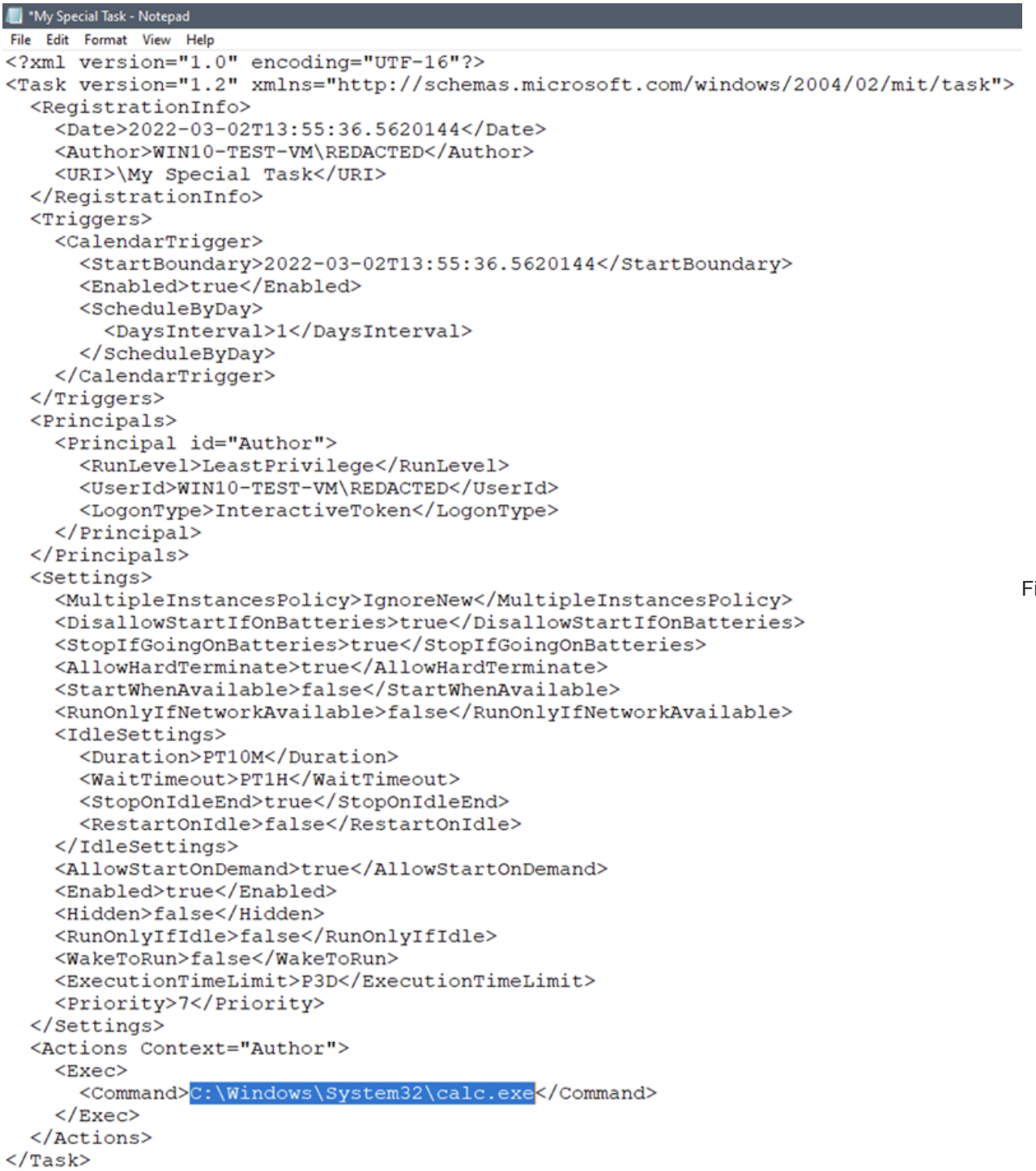


Figure 3. Extensionless XML file

Note that the “Actions” value stored within the Tasks\{GUID} key points to the command line associated with the task. In Figure 2, there is a reference to “C:\Windows\System32\calc.exe” within the “Edit Binary Value” dialog, and there is a path referenced within the “<Command>” section in the extensionless XML file in Figure 3. The fact that this value is stored within two different locations can prove useful in recovering information regarding the task’s purpose in the event the threat actor has taken steps to cover their tracks.

Finally, there are two Windows event logs that record actions related to the creation and operation of Scheduled Tasks – Event ID 4698 within the Security.evtx log, and the Microsoft-Windows-TaskScheduler/Operational.evtx log.

Neither of these are audited by default and must be explicitly turned on by an administrator. Microsoft-Windows-TaskScheduler/Maintenance.evtx will exist by default, but only contains maintenance-related information for the Task Scheduler engine.

Effectively hiding scheduled tasks

In this scenario, the threat actor created a scheduled task named “WinUpdate” via HackTool:Win64/Tarrask in order to re-establish any dropped connections to their command and control (C&C) infrastructure. This resulted in the creation of the registry keys and values described in the earlier section, however, the threat actor deleted the SD value within the Tree registry path.

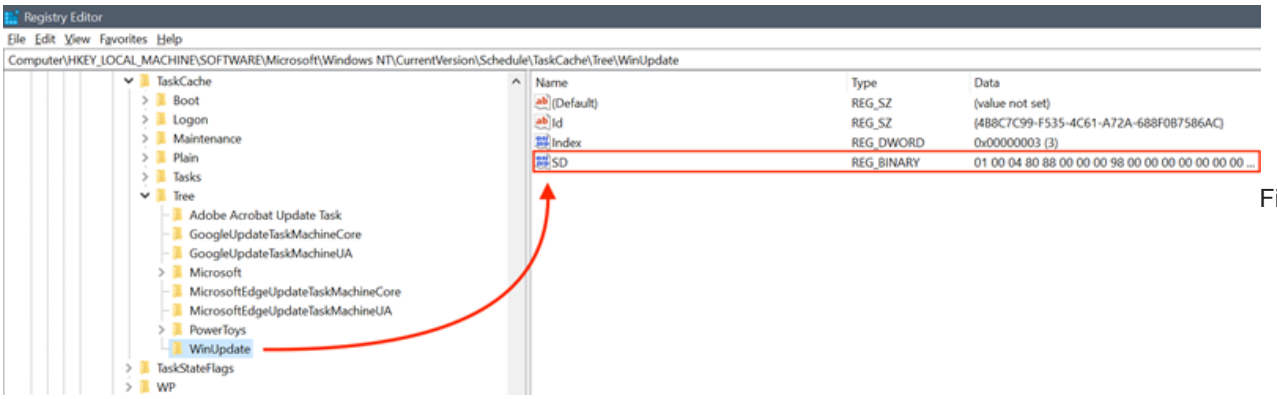


Figure 4. Deletion of the security descriptor (SD) value

In this context, SD refers to the Security Descriptor, which determines the users allowed to run the task. Interestingly, removal of this value results in the task “disappearing” from “schtasks /query” and Task Scheduler. The task is effectively hidden unless an examiner manually inspects the aforementioned registry paths.

Issuing a “reg delete” command to delete the SD value will result in an “Access Denied” error even when run from an elevated command prompt. Deletion must occur within the context of the SYSTEM user. It is for this reason that the Tarrask malware utilized token theft to obtain the security permissions associated with the lsass.exe process. Upon execution of the token theft, the malware could operate with the same privileges as LSASS, making the deletion possible.

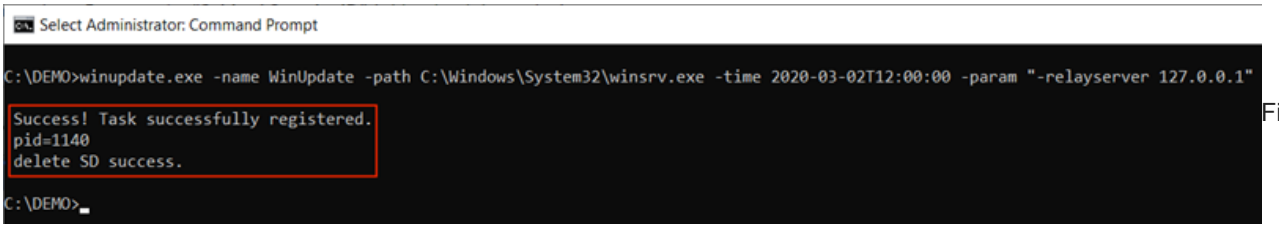


Figure 5. Successful deletion of SD in Command Prompt

It is also important to note that the threat actor could have chosen to completely remove the two registry keys within Tree and Tasks, and the XML file created within *C:\Windows\System32\Tasks*. This would effectively remove the on-disk artifacts associated with the scheduled task, but the task would continue to run according to the defined triggers until the system rebooted, or until the associated svchost.exe process responsible for executing the task was terminated.

It’s possible the threat actor wanted to ensure persistence across reboots and therefore chose not to perform those steps, instead deleting only the SD value; however, we also speculate that the threat actor was unaware that the task would continue to run even after these components were removed.

Recommendations and cyber resilience guidance

Job or task schedulers are services that have been present in the Windows operating system for many years. The attacks we described signify how the threat actor HAFNIUM displays a unique understanding of the Windows subsystem and uses this expertise to mask activities on targeted endpoints to maintain persistence on affected systems and hide in plain sight.

As such, we recognize that scheduled tasks are an effective tool for adversaries to automate certain tasks while achieving persistence, which brings us to raising awareness about this oft-overlooked technique. We also want to bring attention to the fact that threat actors may utilize this method of evasion to maintain access to high value targets in a

manner that will likely remain undetected. This could be especially problematic for systems that are infrequently rebooted (e.g., critical systems such as domain controllers, database servers, etc.).

The techniques used by the actor and described in this post can be mitigated or detected by adopting the following recommendations and security guidelines[1]:

- Enumerate your Windows environment registry hives looking in the *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree* registry hive and identify any scheduled tasks without SD (security descriptor) Value within the Task Key. Perform analysis on these tasks as needed.
- Modify your audit policy to identify Scheduled Tasks actions by enabling logging "TaskOperational" within Microsoft-Windows-TaskScheduler/Operational. Apply the recommended Microsoft audit policy settings suitable to your environment.
- Enable and centralize the following Task Scheduler logs. Even if the tasks are 'hidden', these logs track key events relating to them that could lead you to discovering a well-hidden persistence mechanism
  - Event ID 4698 within the Security.evtx log
  - Microsoft-Windows-TaskScheduler/Operational.evtx log
- The threat actors in this campaign used hidden scheduled tasks to maintain access to critical assets exposed to the internet by regularly re-establishing outbound communications with C&C infrastructure. Remain vigilant and monitor uncommon behavior of your outbound communications by ensuring that monitoring and alerting for these connections from these critical Tier 0 and Tier 1 assets is in place.

Indicators of compromise (IOCs)

The following list provides IOCs observed during our investigation. We encourage customers to investigate these indicators in their environments and implement detections and protections to identify past related activity and prevent future attacks against their systems.

SHA256	File Name	Details
54660bd327c9b9d60a5b45cc59477c75b4a8e2266d988da8ed9956bcc95e6795	winupdate.exe, date.exe, win.exe	Tarrask
a3baacffb7c74dc43bd4624a6abcd1c311e70a46b40dcc695b180556a9aa3bb2	windowsvc.exe, winsrv.exe, WinSvc.exe, ScriptRun.exe, Unique.exe, ngcsvc.exe, ligolo_windows_amd64.exe, proxy.zip, wshqos.exe, cert.exe, ldaputility.exe	Ligolo
7e0f350864fb919917914b380da8d9b218139f61ab5e9b28b41ab94c2477b16d	CertCert.jsp, Cert0365.jsp	Godzilla web shell

Microsoft 365 Defender Detections

How customers can identify this in Microsoft 365 Defender:

Microsoft Defender Antivirus

Microsoft Defender for Endpoint on detects implants and components as the following:

- HackTool:Win64/Tarrask!MSR
- HackTool:Win64/Ligolo!MSR

Microsoft Defender for Endpoint detects malicious behavior observed as the following:

Behavior:Win32/ScheduledTaskHide.A

Microsoft Sentinel Detections

Microsoft Sentinel customers can use the following detection queries to look for this activity:

- Tarrask malware hash IOC: This query identifies a hash match related to Tarrask malware across various data sources.
- Scheduled Task Hide: This query uses Windows Security Events to detect attempts by malware to hide the scheduled task by deleting the SD (Security Descriptor) value. Removal of SD value results in the scheduled task "disappearing" from "schtasks /query" and Task Scheduler.
- Microsoft Defender AV Hits: This query looks for Microsoft Defender AV detections related to Tarrask malware using SecurityAlerts table. In Microsoft Sentinel the SecurityAlerts table includes only the Device Name of the affected device, this query joins the DeviceInfo table to clearly connect other information such as Device group, IP, logged on users etc. This way, the Microsoft Sentinel user can have all the pertinent device info in one view for the alerts.

[1] The technical information contained in this article is provided for general informational and educational purposes only and is not a substitute for professional advice. Accordingly, before taking any action based upon such information, we encourage you to consult with the appropriate professionals. We do not provide any kind of guarantee of a certain outcome or result based on the information provided. Therefore, the use or reliance of any information contained in this article is solely at your own risk.