1,000+ Customers, **12 years** of best-in-class solutions

LOGPOINT

Products Solutions Pricing Resources & Support Partner resources

Book a demo

Shenanigans of Scheduled Tasks

August 23rd, 2024 - 12 min read

Updated August 26, 2024

Scheduled Tasks are the automated processes configured to run automatically at a specified time or when certain conditions are met. By eliminating the need for manual intervention, these tasks enable the execution of repetitive or routine tasks while ensuring efficiency and consistency. In Windows, the Task Scheduler service manages and executes these automated tasks, similar to cron jobs on Unix systems. It monitors the conditions for any automated task, be it time or event criteria, and executes the tasks as soon as certain conditions are met. Attackers can also abuse scheduled tasks to persist malicious code or execute unauthorized activities on a compromised system.

Swachchhanda Shrawan Poudel
Security Research

Jump To Section

Share This Sto

https://www.logpoint.com/e...

Task Scheduler

and have them
ning scripts,

. It operates in

aging all tasks.

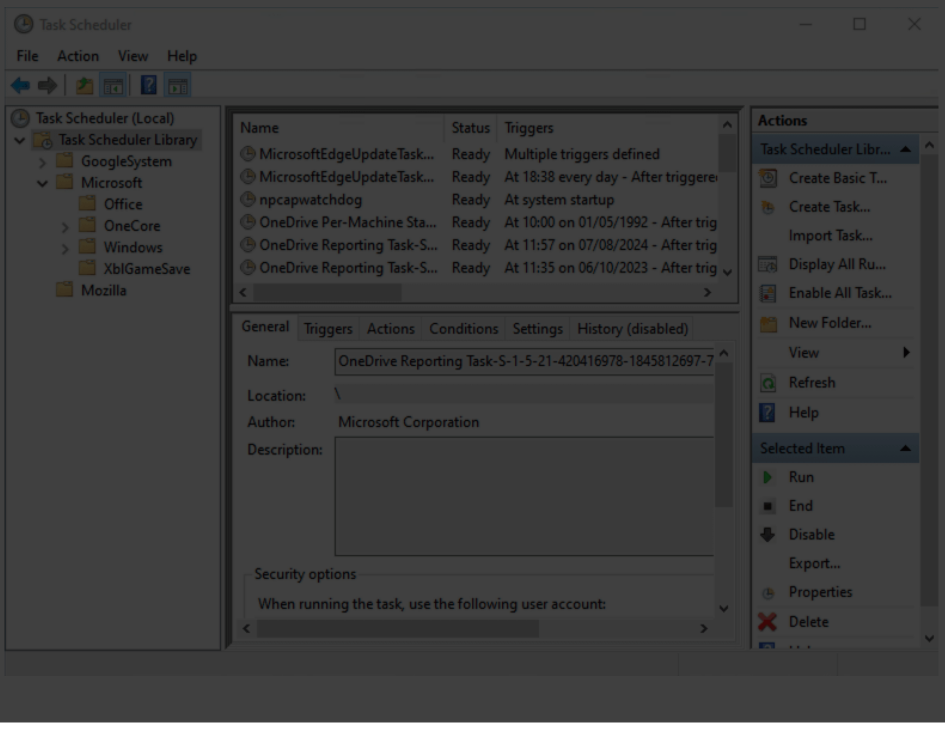
ML format in

ple, system

ng an email, or
displaying a message.

3. **Principals:** The security context in which the task is executed
4. **Conditions**: Additional requirements for the job, such as only operating while the machine is idle or connected to AC power.
5. **Settings**: Settings control task execution circumstances, such as task priority, the ability to run numerous instances, handling during idle periods, and other external considerations.
6. **Registration Information:** It contains metadata such as a task's creation date, the task author, and others.

4. **Task Scheduler User Interface**: The graphical interface, accessible through taskschd.msc or the Control Panel, enables users to interact with Task Scheduler visually, facilitating the creation and management of tasks.



Name	Status	Triggers
MicrosoftEdgeUpdateTask...	Ready	Multiple triggers defined
MicrosoftEdgeUpdateTask...	Ready	At 18:38 every day - After triggere
npcapwatchdog	Ready	At system startup
OneDrive Per-Machine Sta...	Ready	At 10:00 on 01/05/1992 - After trig
OneDrive Reporting Task-S...	Ready	At 11:57 on 07/08/2024 - After trig
OneDrive Reporting Task-S...	Ready	At 11:35 on 06/10/2023 - After trig

General Triggers Actions Conditions Settings History (disabled)

Name: OneDrive Reporting Task-S-1-5-21-420416978-1845812697-7

Location: \

Author: Microsoft Corporation

Description:

Security options

When running the task, use the following user account:

5. **Task Scheduler API**: An interface that allows developers to create and manage tasks programmatically. This API may be accessible using scripting languages like PowerShell or custom applications.

More information about Task Scheduler can be found here in Microsoft's official documentation link.

Task Scheduling and Execution in Windows

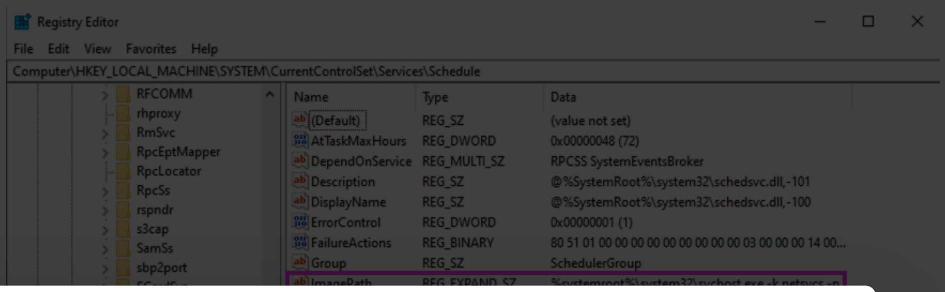
1. **Scheduling a Task**

- **User Action**: The Task Scheduler program allows you to create a scheduled task. It includes specifying when the task should execute (for example, at a specific time or upon system startup) and what it should perform (for example, run a script or launch an application).
- **Task Storage**: The job is saved to the Task Scheduler Library and registered in the Windows Registry. These two registries are formed for any new scheduled task created.

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\TASK_NAME
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{GUID}
```

Similar data is also stored within an extensionless XML file located in the `C:\Windows\System32\Tasks` directory.

2. **Monitoring and Preparing for Task Execution**



Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Schedule

RFCOMM
rhproxy
RmSvc
RpcEptMapper
RpcLocator
RpcSs
rspndr
s3cap
SamSs
sbp2port

Name	Type	Data
(Default)	REG_SZ	(value not set)
AtTaskMaxHours	REG_DWORD	0x00000048 (72)
DependOnService	REG_MULTI_SZ	RPCSS SystemEventsBroker
Description	REG_SZ	@%SystemRoot%\system32\schedsvc.dll,-101
DisplayName	REG_SZ	@%SystemRoot%\system32\schedsvc.dll,-100
ErrorControl	REG_DWORD	0x00000001 (1)
FailureActions	REG_BINARY	80 51 01 00 00 00 00 00 00 00 00 00 03 00 00 00 14 00...
Group	REG_SZ	SchedulerGroup
ImagePath	REG_EXPAND_SZ	%systemroot%\system32\svchost.exe -k netsvcs -p

uns as part of a
es.

vchost.exe
ervice group.

p may include
registry at

- **Trigger Condition**: When the planned time or event happens (for example, the system boots up or the user logs in), the Task Scheduler service identifies the requirement to execute a task.
- **Task Execution**: The Task Scheduler service gets ready to execute the task:
  - **Launching the Task**: It launches a new process, often `taskhostw.exe` (Task Host Window), in charge of carrying out the specified task.
  - **Running the Action**: The `taskhostw.exe` process then executes the task's action that might involve:
    - **Running a Script**: Launching `cmd.exe` or `powershell.exe` to execute a script.
    - **Starting an Application**: Launching the specified application, for example, `example.exe`.
- **User Context**: The job runs with the user rights set when it was created.

4. **Logging and Continuation**

- **Task Logging**: After completing the job, the job Scheduler reports the results in the Event Viewer under "Task Scheduler", which includes:
  - **Start and End Times**.
  - **Result of the Task** (success, failure, error codes).
  - **Output or Error Messages**.
- **Ongoing Monitoring**: The Task Scheduler service continues to operate and is ready to handle and execute any more scheduled tasks when triggered.

Scheduled Task for Persistence

Persistence is crucial for the attackers to have a successful attack. It allows them to maintain long-term access to the compromised system, enabling continued carnage and a foothold for more destructive objectives. One of the prevalent and widely used techniques used for persistence is the scheduled task. Adversaries commonly used the scheduled tasks to maintain persistence, execute their malicious payloads at specific times, or perform automatic tasks on compromised devices. Also, scheduled tasks allow the processes to execute with elevated privileges. Furthermore, they try to configure such tasks to run under the guise of legitimate processes, making their foothold ever stronger and making detection difficult. One of the most common tools used in Windows for scheduling such tasks is 'schtasks.exe'.

Schtasks

'`schtasks.exe`' is a Windows command-line application for managing scheduled tasks on local or remote computers, such as creating, removing, editing, executing, and terminating tasks. While it is an essential tool for system administrators, it is also a common choice of attackers to maintain persistence, launch malicious payloads, and carry out illegal operations. We have noticed the utilization of **schtasks**, particularly the use of `/create` command to schedule the task, in a diverse range of threat actor operations and malicious software. Below is the snippet of the command seen in one of the latest samples of Xworm to create a new scheduled task.

```
schtasks  /create /sc minute /mo 1 /tn "Nafifas" /tr "'C:\Users\Admin\AppData\Roaming\svchost\svcho
```

This command creates a scheduled " Nafifas " task that runs every minute. The task executes the file `svchost.exe`located in the `C:\Users\Admin\AppData\Roaming\svchost\` directory. It's an expected behavior seen in adversaries to drop their payloads from user-writeable directories like `\AppData\Roaming\`. The `/f` flag forces the task's creation, replacing any existing task with the same name. An attacker uses this task to maintain persistence, repeatedly running a potentially malicious executable.

In this instance, the scheduled task is executing the `svchost.exe` process from the `\AppData\Roaming` directory, which is not a usual location for legitimate processes like `svchost.exe`, usually executed from `%SystemRoot%\System32`. The malware attempts to evade detection by posing as a legitimate system process and running from an unusual directory. This strategy helps conceal its activity, making it harder for users and security tools to recognize it as malicious and allowing the malware to maintain its foothold.

y the existing

d scans of

t SIEM

points. Here is a

logging.

ng at the

of

In Windows, the **Microsoft-Windows-TaskScheduler/Operational** log is a specific event log within the Windows Event Viewer dedicated to recording detailed operational events for the Task Scheduler service. This log provides comprehensive information about the activities and operations carried out by the Task Scheduler, including task registration, running, termination, and errors. Event ID 200 is generated when a new task action starts and likewise for other actions.

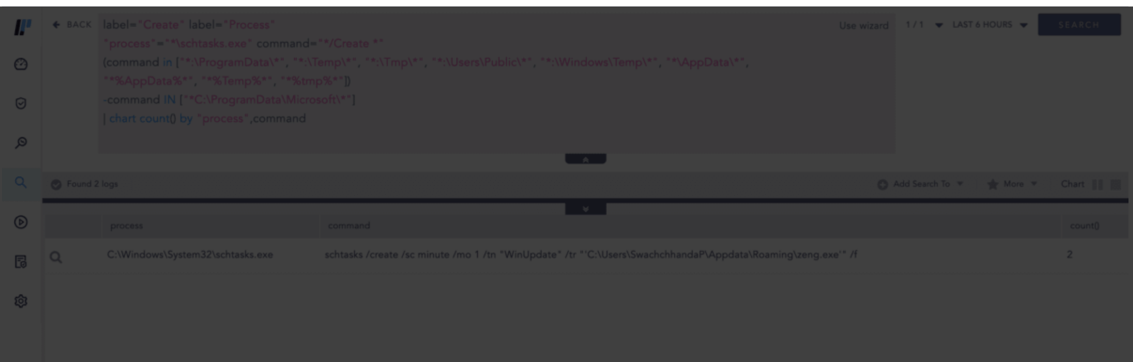
Scheduled Task logs are also available through the **Microsoft-Windows-Security-Auditing** events. To access these logs, you must enable the "**Audit Other Object Access Events**" policy. Once enabled, the following Event IDs are generated:

- **4698**: Scheduled task creation.
- **4699**: Scheduled task deletion.
- **4700**: Scheduled task enabled.
- **4701**: Scheduled task disabled.
- **4702:** Scheduled task updated.

Persistent Tasks from Suspicious Locations

Malware or threat actors frequently drop their payloads in publicly writable directories, utilizing them for initial deployment and persistence. It is crucial to monitor scheduled tasks originating from these specific directories.

```
label="Create" label="Process"
"process"="*\schtasks.exe" command="*/Create *"
(command in ["*:\ProgramData\*", "*:\Temp\*", "*:\Tmp\*", "*:\Users\Public\*",
 "*:\Windows\Temp\*", "*\AppData\*", "*%AppData%*", "*%Temp%*", "*%tmp%*"])
```



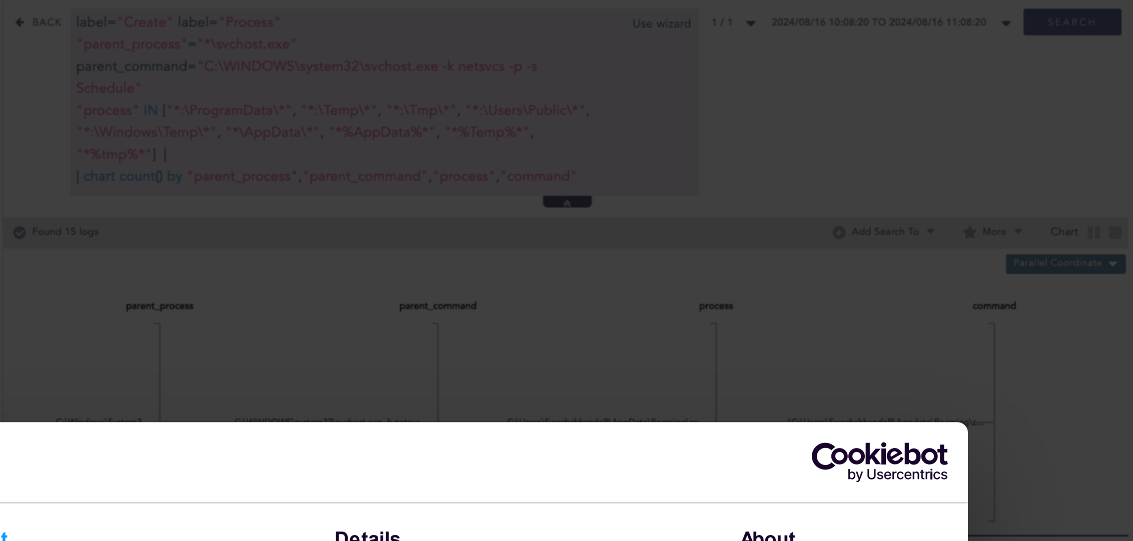
```
label="Create" label="Process"
"process"="*\schtasks.exe" command="*/Create *"
(command in ["*:\ProgramData\*", "*:\Temp\*", "*:\Tmp\*", "*:\Users\Public\*", "*:\Windows\Temp\*", "*\AppData\*",
"*%AppData%*", "*%Temp%*", "*%tmp%*"])
-command IN ["*C:\ProgramData\Microsoft\*"]
| chart count() by "process",command
```

Found 2 logs

process	command	count()
C:\Windows\System32\schtasks.exe	schtasks /create /sc minute /mo 1 /tn "WinUpdate" /tr "'C:\Users\SwachchhandaP\Appdata\Roaming\zeng.exe'" /f	2

The management of scheduled tasks' execution on Windows 10 is handled by "svchost.exe" through the command line "`C:\WINDOWS\system32\svchost.exe -k netsvcs -p -s Schedule`". Before Windows 10 Version 1511, it was executed by taskeng.exe. Analyzing the subprocesses of this particular process enables the detection of any irregular patterns that could indicate the presence of potentially harmful scheduled tasks.

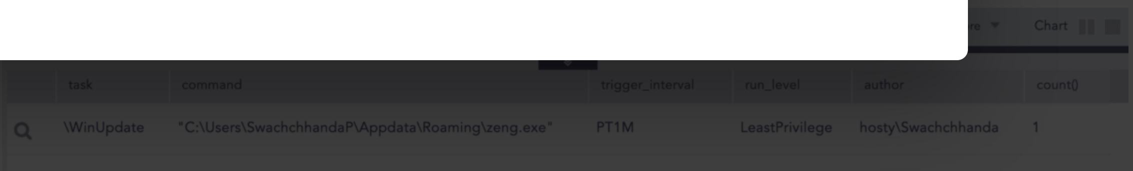
```
label="Create" label="Process" ("parent_process"="*\taskeng.exe")
OR ("parent_process"="*\svchost.exe" parent_command="C:\WINDOWS\system32\svchost.exe -k netsvcs -p
"process" IN ["*:\ProgramData\*", "*:\Temp\*", "*:\Tmp\*", "*:\Users\Public\*",
"*:\Windows\Temp\*", "*\AppData\*", "*%AppData%*", "*%Temp%*", "*%tmp%*"]
```



```
label="Create" label="Process"
"parent_process"="*\svchost.exe"
parent_command="C:\WINDOWS\system32\svchost.exe -k netsvcs -p -s
Schedule"
"process" IN ["*:\ProgramData\*", "*:\Temp\*", "*:\Tmp\*", "*:\Users\Public\*",
"*:\Windows\Temp\*", "*\AppData\*", "*%AppData%*", "*%Temp%*",
"*%tmp%*"] |
| chart count() by "parent_process","parent_command","process","command"
```

Found 15 logs

on of such



task	command	trigger_interval	run_level	author	count()
\WinUpdate	"C:\Users\SwachchhandaP\Appdata\Roaming\zeng.exe"	PT1M	LeastPrivilege	hosty\Swachchhanda	1

Suspicious Scheduled Programs Execution

Malware or threat actors tend to execute their malicious scripts repeatedly using Windows shell script programs such as cmd, Powershell, Wscript, and Csript while also abusing the other common LOLBAS. It's also a good idea to look at such execution patterns.

The query below can help detect the creation of such scheduled tasks that execute potentially suspicious programs.

```
(label="Create" label="Process" "process"="*\schtasks.exe" command="*/Create *" )
OR (norm_id=WinServer "event_id"=4698)
command IN ["*\InstallUtil.exe*", "*\RUNDLL32.EXE*", "*\msiexec.exe*", "*\RegSvcs.exe*",
"*\MSHTA.EXE*", "*\Cmd.Exe*", "*\PowerShell.EXE*", "*\cscript.exe*",
"*\CONTROL.EXE*", "*\RegAsm.exe*", "*\REGSVR32.EXE*","*\wscript.exe*",
"*\Microsoft.Workflow.Compiler.exe*", "*\MSBuild.exe*", "*\msxsl.exe*"]
```

The above query only detects creation, but we can track svchost.exe, which is associated with scheduling tasks, to detect process execution details.

```
label="Create" label="Process" ("parent_process"="*\taskeng.exe")
OR ("parent_process"="*\svchost.exe" parent_command="C:\WINDOWS\system32\svchost.exe -k netsvcs -p
"process" IN ["*\InstallUtil.exe", "*\RUNDLL32.EXE", "*\msiexec.exe", "*\RegSvcs.exe",
"*\MSHTA.EXE", "*\Cmd.Exe", "*\PowerShell.EXE", "*\cscript.exe",
"*\CONTROL.EXE", "*\RegAsm.exe", "*\REGSVR32.EXE","*\wscript.exe",
"*\Microsoft.Workflow.Compiler.exe", "*\MSBuild.exe", "*\msxsl.exe"]
```

Alternate via network connection event

In specific campaigns, the malware communicates with an external domain, commonly a Command and Control (C&C) server, to exchange information, including data and instructions. This communication is facilitated through scheduled tasks, enabling the malware to interact with malicious domains, download payloads, or transmit collected data based on the threat actor's requirements. We can look for the URL used in the scheduled task command line.

```
label="Create" label="Process"
("process"="*\schtasks.exe" command="*/Create *" )
OR ("parent_process"="*\taskeng.exe")
OR ("parent_process"="*\svchost.exe"
parent_command="C:\WINDOWS\system32\svchost.exe -k netsvcs -p -s Schedule")
command IN ["*http:\\*", "*https:\\*"])
```

Scheduled Task File Creation

When a new scheduled task is created, an extensionless XML file is automatically created in the `%systemroot%\System32\Tasks` directory. Keep an eye on generating these files unrelated to recognized software or valid processes. The file name matches the name of the scheduled task.

```
label=File label=Create
path in ["C:\Windows\System32\Tasks", "C:\Windows\Tasks]"]
```

Masqueraded Scheduled Tasks

Masqueraded scheduled tasks refer to instances where an attacker creates or modifies scheduled tasks on a system to appear legitimate, thereby disguising malicious activities. By masquerading, attackers aim to avoid detection by system administrators or security software. These tasks can be used to maintain persistence, execute malicious payloads, or perform other actions at specific times or intervals.

Mimicking Malicious scheduled tasks as legitimate processes, such as 'svchost.exe', 'lsass.exe', and others, is one common defense evasion technique used by threat actors. The following query can help us look for the execution of such processes.

```
-k netsvcs -p
"C:\Windows\WinSxS\*"]
```

This query detects such anomalous behaviors by identifying processes that start from unusual paths but exclude scheduled task processes from known safe locations.

Scheduled Task Registry Modification

Windows Sysmon provides a range of valuable telemetry for threat hunting, including telemetry for the Windows registry. When a new scheduled task is created, two specific registry entries are generated:

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\TASK_NAME
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{GUID}
```

Therefore, it is advisable to enable the entry of these registry keys in the sysmon configuration to identify any potentially malicious scheduled tasks.

```
<Sysmon schemaversion="4.30">
  <EventFiltering>
    <RuleGroup name="" groupRelation="or">
      <RegistryEvent onmatch="include">
        <TargetObject name="technique_id=T1053,technique_name=Scheduled Task" condition="contains a
        <TargetObject name="technique_id=T1053,technique_name=Scheduled Task" condition="begin with
      </RegistryEvent>
    </RuleGroup>
  </EventFiltering>
</Sysmon>
```

The provided query can help pinpoint such events. Nonetheless, it's crucial to proceed with caution as the query might produce noise. Hence, we advise incorporating a filter to eliminate false positives, given that numerous valid processes continuously generate and update scheduled tasks. Subsequently, establishing a baseline query specific to your organization's environment can aid in identifying any questionable scheduled task executions.

```
norm_id=WindowsSysmon event_id IN [12, 13, 14]
target_object="*\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tree\*"
```

Scheduled Task via a COM Interface

In a recent campaign of Latrodectus, a unique approach was observed to bypass the usual `schtasks.exe` command-line tool while creating the scheduled tasks. Instead, it was observed leveraging the COM object (`Schedule.Service`) to interact with the Task Scheduler through the `taskschd.dll` module. Since, various legitimate processes also load `taskschd.dll`, merely detecting its use is not enough to identify malicious intent. As malicious payloads are generally dropped in publicly writable directories, examining suspicious processes originating from these paths that load this module could indicate malicious scheduled task creation. Following query is catered to hunt for such events.

```
label="Image" label=Load
image="*\taskschd.dll"
"process" in ["*\AppData\Local\Temp\*",  "*\AppData\Roaming\*",
"*C:\Users\Public\*", "*C:\Windows\Temp\*", "*C:\Temp\*", "*\Downloads\*",
"*\Desktop\*"]
```

Logpoint Alerts

In addition to the detection queries mentioned above, you can utilize Logpoint's out-of-the-box alert rules. Activating these alerts can assist in pinpointing any potentially suspicious scheduled tasks within your enterprise system. The following alerts are associated explicitly with suspicious scheduled tasks.

Alert Name
Scheduled Task Creation Detected
Default PowerSploit and Empire Schtasks Persistence

ersistence. Prior code to execute your

int sensor proactive monitoring and mitigation of suspicious activities in their infancy, safeguarding your business operations. Additionally, Logpoint Security Research continuously develops analytics, sharing insights through blog posts like the Emerging Threats Protection Reports. Stay updated by following Logpoint on LinkedIn.

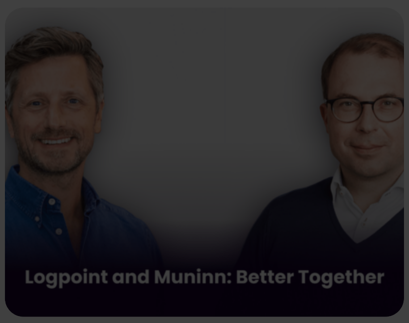
Related Posts

Uncover more resources with Logpoint's latest release

October 30th, 2024

Latrodectus: The Wrath of Black Widow

October 22nd, 2024



Logpoint and Muninn: Better Together

October 1st, 2024

Case Management
Behavior Analytics
Cyber Defense Platform
Pricing
Sizing Calculator

EAL3+ Certificate
Newsletter

Careers at Logpoint
Media Room
Logpoint in the media
Blog & Webinars

Documentation
Community
Contact
Status

☎ +45 7060 6100

Cookiebot by Usercentrics

Consent | Details | About

This website uses cookies

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or that they've collected from your use of their services.

Necessary	Preferences	Statistics	Marketing
●	○	○	○