Tachyon Activity Record

Summary

Reference information about the Tachyon Activity Record (TAR) feature, sometimes referred to as either the inventory, or forensics feature, and previously known as Agent Historic Data Capture.

What is Tachyon Activity Record?

The Tachyon client, while running, continuously captures details of certain activities and events as they happen, similar to Windows Task Manager or Perfmon. During startup, the Tachyon client is able to detect some events that occurred when it was not running. Data is regularly written into a local, compressed and encrypted **persistent storage tables**, that are accessible to SCALE as SQL tables. The Tachyon client periodically aggregates data in order to minimize the amount of storage required, so that each capture source has a live, hourly, daily and monthly table. The whole process is designed to minimize impact on device performance, storage and security.

Persistent storage tables cannot be deleted or modified because they are managed by the Tachyon client and used by the Tachyon Activity Record feature. Whereas User Defined Persistent Storage Tables can be created, deleted and modified using SCALE because they are managed by Tachyon instructions run by users.

The Tachyon Activity Record schema is provided below.

> To use any TAR features you must have Inventory enabled by setting **Module.Inventory.enabled=true** in the client configuration file. This setting, and configuration options for each capture source are described in 1E Client 8.0 - Tachyon client settings: Capture source settings.
>
> With some exceptions, the defaults are the same for each capture source, but aggregation and retention settings can be set individually for each table.

The Tachyon client has two mechanisms of knowing when an event occurs that is of interest.

- **Event-based** relies on a source external to the Tachyon client (normally the operating system) providing a notification to indicate that something has happened
- **Polling-based** is where the Tachyon client will periodically check a source of data and work out what has changed by looking at differences in the data returned. Polling intervals means some brief events that occur between polls can be missed.

On Windows, the Tachyon client is able to use Event Tracing for Windows (ETW). However, if desired, the individual capture sources can be configured to use polling instead of ETW.

Other data collection methods that run periodically (polling-based) for a short period:

- Windows performance counters for disk, memory, network and processor performance
- a proprietary **metric** collection process that tests the Operating System and its software.

> Starting with Tachyon 5.1, the 1E Client UI component (which runs in each user's session) provides data to the Tachyon client about the user's interaction with the device and foreground applications. This data is available by querying the Device interaction and Software interaction tables.
>
> For this data to be captured, you must have Interaction enabled (set **Module.Interaction.Enabled=true**) as well as user interaction sampling (set **Module.Interaction.SampleUserInteraction=true**). Both are enabled by default.
>
> For more details on configuring the Interaction module, see 1E Client 8.0 - Tachyon client settings: Interaction module settings.

Capture sources

The table below lists currently supported data capture sources, on which OS they are supported, and which capture method is used by default. See Constraints of Legacy OS regarding Windows XP, Vista and Windows Server 2003.

TAR data source	Description	Required by	Windows	macOS	Linux	Solaris	Android

ARP cache entries $ARP_xxx	The Tachyon client captures translations between IP addresses and MAC (physical) addresses, known as ARP (Address Resolution Protocol). ARP cache polling is every 30 seconds.		• Introduced in 3.2 • Polling on all versions of Windows	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*
Boot performance $BootPerformance_Live	The Tachyon client captures boot performance related metrics from events logged by Windows OS.	Experience	• Introduced in 8.0 • Windows Event Log	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*
Device interaction $DeviceInteraction_xxx	The Tachyon client captures user interaction (keyboard and mouse activity) with the device for each (local and remote) user session. Data is captured by the 1E Client UI, whose behaviour is controlled by the Interaction module.	Experience	• Introduced in 5.1 • Data from the 1E Client UI	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*
Device performance $DevicePerformance_xxx	The Tachyon client captures metrics for device performance by interrogating Windows Performance Counters. These metrics cover disk, memory, network and processor performance. Device performance polling is every 10 seconds.	Experience	• Introduced in 5.0 • Windows Performance Counters	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*
Device resource demand $DeviceResourceDemand_xxx	The Tachyon client captures high-level device resource allocation and utilization - specifically for CPU, disk, network and memory. This data is used by the VDI monitoring feature of Tachyon Experience to show resource usage patterns, and to help identify under- and over-provisioned virtual servers. Device resource demand polling is every 10 seconds.	Experience	• Introduced in 5.1 • Windows Performance Counters	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*
DNS resolutions $DNS_xxx	The Tachyon client captures whenever a DNS address is resolved. When using the polling method, the polling interval is every 30 seconds.		• Introduced in 2.1 • Polling on Windows 8 and below • ETW on Windows 8.1 and above	• Introduced in 2.1 (not available for Mojave and later) • Polling	*Not yet available*	*Not yet available*	*Not yet available*
Operating System performance $OperatingSystemPerformance_xxx	The Tachyon client captures metrics for performance and sensitive processes by running a metrics executable every four hours by default, that captures 15 metrics. Operating system performance polling is every 4 hours (14,400 seconds).	Experience	• Introduced in 5.0 • Proprietary metric collection	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*
Performance event $PerformanceEvent_xxx	The Tachyon client captures a summary of operating system events which relate to device performance and stability, such as application/operating system crashes, patch installation and uninstallation, etc. Events are captured as they are generated by the operating system.	Experience	• Introduced in 5.1 • Windows Event Log	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*

Process executions $Process_xxx	The Tachyon client captures whenever a process starts on the device. When using the polling method, the polling interval is every 30 seconds.		• Introduced in 2.1 • Polling on Windows XP • ETW on Windows Vista and above	• Introduced in 2.1 • Polling	• Introduced in 2.1 • Polling	• Introduced in 2.1 • Polling	*Not yet available*
Process stabilizations $ProcessStabilization_xxx	The Tachyon client captures the time taken for a process to be considered stable. This is captured when a process starts on a device, provided that process is in a list of processes selected for monitoring in the 1E Client configuration file.		• Introduced in 3.2 • ETW on Windows Vista and above	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*
Process usage $ProcessUsage_Daily	The Tachyon client captures details about running processes from start to end. When using the polling method, the polling interval is every 30 seconds.	Tachyon Powered Inventory	• Introduced in 3.2 • Polling on Windows XP • ETW on Windows Vista and above	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*
Sensitive processes $SensitiveProcess_xxx	The Tachyon client captures metrics for performance and sensitive processes by running a metrics executable every four hours by default, that captures 15 metrics. Sensitive processes polling is every 4 hours (14,400 seconds).	Experience	• Introduced in 5.0 • Proprietary metric collection	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*
Software installations $Software_xxx	The Tachyon client captures which software is present on a device, and when it is installed and uninstalled. Software polling is every 120 seconds.		• Introduced in 2.1 • Polling on all versions of Windows	• Introduced in 2.1 • Polling	• Introduced in 2.1 • Polling	• Introduced in 2.1 • Polling	*Not yet available*
Software interaction $SoftwareInteraction_xxx	The Tachyon client captures user interaction (keyboard and mouse activity) with each application that enters the foreground. Data is captured by the 1E Client UI, whose behaviour is controlled by the Interaction module.	Experience	• Introduced in 5.1 • Data from the 1E Client UI	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*
Software performance $SoftwarePerformance_xxx	The Tachyon client captures metrics for software performance in terms of disk I/O, memory and processor usage. Software performance polling is every 10 seconds.	Experience	• Introduced in 5.0 • Windows Performance Counters	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*
SoftwarePerformance.DiskUsage $SoftwarePerformance_xxx	The Tachyon Client captures details about disk i/o operations for all running processes. We register 'Microsoft-Windows-Kernel-Disk' Event tracing provider to gather data	Experience	• Introduced in 8.0 • Event tracing	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*

SoftwarePerformance.ProcessNetworkUsage $SoftwarePerformance_xxx	The Tachyon Client captures details about network activity for all running processes. We use a real time 'NT Kernel Logger' event tracing session to gather network traffic data	Experience	• Introduced in 8.0 • Event tracing	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*
TCP outbound connections $TCP_xxx	The Tachyon client captures whenever an outbound TCP connection is made. When using the polling method, the polling interval is every 30 seconds.		• Introduced in 2.1 • Polling on Windows XP • ETW on Windows Vista and above	• Introduced in 2.1 • Polling	• Introduced in 2.1 • Polling	*Not yet available*	*Not yet available*
User usage $UserUsage_Daily	The Tachyon client captures details about user sessions from login to logout. System accounts, and accounts used to run services, are excluded. The polling interval is every 30 seconds.	Tachyon Powered Inventory	• Introduced in 3.2 • Polling on all versions of Windows	*Not yet available*	*Not yet available*	*Not yet available*	*Not yet available*

How do I retrieve the data from Tachyon client devices?

Live and aggregated Tachyon Activity Record data is stored in the following persistent storage tables. You can simply query these using SELECT statements.

TAR data source	Live tables	Hourly tables	Daily tables	Monthly tables
ARP cache entries	$ARP_Live	$ARP_Hourly	$ARP_Daily	$ARP_Monthly
Boot performance	**$BootPerformance_Live**	**n/a**	**n/a**	**n/a**
Device interaction	$DeviceInteraction_Live	$DeviceInteraction_Hourly	$DeviceInteraction_Daily	$DeviceInteraction_Monthly
Device performance	$DevicePerformance_Live	$DevicePerformance_Hourly	$DevicePerformance_Daily	$DevicePerformance_Monthly
Device resource demand	$DeviceResourceDemand_Live	$DeviceResourceDemand_Hourly	$DeviceResourceDemand_Daily	$DeviceResourceDemand_Monthly
DNS resolutions	$DNS_Live	$DNS_Hourly	$DNS_Daily	$DNS_Monthly
Operating System performance	$OperatingSystemPerformance_Live	$OperatingSystemPerformance_Hourly	$OperatingSystemPerformance_Daily	$OperatingSystemPerformance_Monthly
Performance event	$PerformanceEvent_Live	$PerformanceEvent_Hourly	$PerformanceEvent_Daily	$PerformanceEvent_Monthly
Process executions	$Process_Live	$Process_Hourly	$Process_Daily	$Process_Monthly
Process stabilizations	$ProcessStabilization_Live	$ProcessStabilization_Hourly	$ProcessStabilization_Daily	$ProcessStabilization_Monthly
Process usage	**n/a**	**n/a**	**$ProcessUsage_Daily**	**n/a**
Sensitive processes	$SensitiveProcess_Live	$SensitiveProcess_Hourly	$SensitiveProcess_Daily	$SensitiveProcess_Monthly
Software installations	$Software_Live	$Software_Hourly	$Software_Daily	$Software_Monthly
Software interaction	$SoftwareInteraction_Live	$SoftwareInteraction_Hourly	$SoftwareInteraction_Daily	$SoftwareInteraction_Monthly
Software performance	$SoftwarePerformance_Live	$SoftwarePerformance_Hourly	$SoftwarePerformance_Daily	$SoftwarePerformance_Monthly
TCP outbound connections	$TCP_Live	$TCP_Hourly	$TCP_Daily	$TCP_Monthly
User usage	**n/a**	**n/a**	**$UserUsage_Daily**	**n/a**

Example - querying historic captured data

```
/* Sum the number of connections made per process today */
SELECT  SUM(ConnectionCount) AS Connections
,      ProcessName
FROM   $TCP_Daily
WHERE   TS = DATETRUNC(STRFTIME("%s", "now"), "day")
GROUP BY ProcessName;
```

Note the below example uses LIKE because the inventory tables are not created with COLLATE NOCASE, and need to be queried in a case-sensitive fashion. If ProcessName = "chrome.exe" is used then it will not match "Chrome.exe" or "chrome.EXE".

Example - handling case-sensitivity

```
SELECT * FROM $Process_Live WHERE ProcessName LIKE "chrome.exe"
```

How is the data managed?

The Tachyon client automatically aggregates and grooms data in each inventory table, according to aggregation intervals and data retention settings which are configurable in the 1E Client configuration file.

- Default aggregation cycle interval is every 60 seconds, therefore it may take up to a minute before an event appears in an aggregated table
- Default retention for live tables is 5000 entries provided at least 3 aggregation cycles have occurred (older entries are deleted to make room for new entries)
- Default retention for hourly tables is a rolling 24 hours.
- Default retention for daily tables is a rolling 31 days.
- Default retention for monthly tables is a rolling 12 months.

Each aggregated table is built from the live table, and does not have a dependency on other aggregated tables. For example, Monthly is fed by Live, not fed by Daily. This allows retention settings to be configured independently for each table.

Data is stored in a local, compressed and encrypted persistent store, which persists during a Tachyon client upgrade, uninstall and re-installation, unless specifically deleted.

If the Tachyon client is unable to write to storage (out of disk space or other file-system problems), it will fail but continue monitoring in the hope this situation will improve later.

Tachyon Activity Record schema

The following table shows the fields which exist only in the Live and Aggregated (Hourly, Daily, Monthly) tables. This table is provided to help you avoid schema issues.

TAR data source	Fields that exist only in Live tables	Fields that exist only in Aggregated tables
ARP cache entries	n/a	n/a
Boot performance	n/a	n/a
Device interaction	InteractionSeconds, LogonSeconds, PresentSeconds	AverageIdleResponsivenessMsCount, AverageInteractiveResponsivenessMsCount, AverageSessionResponsivenessMsCount, InteractionMinutes, LogonMinutes, PresentMinutes
Device performance	n/a	SampleCount
Device resource demand	n/a	SampleCount
DNS resolutions	n/a	LookupCount
Operating System performance	n/a	ExecutionCount
Performance event	EventData	EventCount
Process executions	CommandLine, ProcessId, ParentProcessId	ExecutionCount

Process stabilizations	ProcessId, StabilizationTimeMs	ExecutionCount, TotalStabilizationTimeMs
Process usage	n/a	All fields only available in $ProcessUsage_Daily.
Sensitive processes	n/a	DetectionCount
Software installations	IsUninstall	InstallCount, UninstallCount
Software interaction	InteractionSeconds, LogonSeconds, PresentSeconds	AverageIdleResponsivenessMsCount, AverageInteractiveResponsivenessMsCount, AverageSessionResponsivenessMsCount, InteractionMinutes, LogonMinutes, PresentMinutes
Software performance	n/a	SampleCount
TCP outbound connections	ProcessId	ConnectionCount
User usage	n/a	All fields only available in $UserUsage_Daily.

Timestamps

The timestamp column (TS) in each table is stored in Unix Epoch format, defined as the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970.

To convert to a readable text format use the EPOCHTOJSON function. See also datetime handling.

Example - converting Unix Epoch timestamps

```
SELECT Fqdn, EPOCHTOJSON(TS) AS TS_ FROM $DNS_Hourly WHERE Fqdn LIKE "%facebook%";
```

Timestamps are truncated in the aggregated tables.

- Hourly - time is truncated to each hour - so an event that occurred at 2017-01-27 18:03:54 would be included in the summary for **2017-01-27 18:00:00**
- Daily - time is truncated to midnight on each day - so an event that occurred at 2017-01-27 18:03:54 would be included in the summary for **2017-01-27 00:00:00**
- Monthly - time is truncated to midnight on the first day of each month - so an event that occurred at 2017-01-27 18:03:54 would be included in the summary for **2017-01-01 00:00:00**

ARP cache entries

The following table shows fields available in the **$ARP_** tables.

Field	Datatype	Description	Sample value	Tables
CacheCount	integer	The number of times that the combination of IpAddress, MacAddress and Subnet were seen in the ARP cache for this time period.	1234	All
IpAddress	string	The IP address that was resolved using ARP.	192.168.11.12	All
MacAddress	string	The MAC (physical) address to which the IP address was resolved.	58-82-a8-93-4c-da	All
Subnet	string	The CIDR-format IP subnet to which the resolved IP address belongs.	192.168.11.0/8	All
TS	integer	When the record was added to the table. See Timestamps.	1500756083	All

The Tachyon client polls the operating system ARP cache periodically. Since the lifetime of an entry in the ARP cache can be variable, if an entry in the ARP cache is encountered which is already present in the Tachyon client's database, the Tachyon client will increment the CacheCount field on the table for that row, and update the timestamp (TS) field to the current time. To that end, the CacheCount field can be used to determine how frequently a particular entry was observed in the operating system's cache.

Boot performance

The following table shows fields in the **$BootPerformance_Live** table.

Field	Datatype	Description	Sample value	Tables
BootAutoChkTimeSeconds	real	--	0	

BootCriticalServicesInitTimeSeconds	real	Time in seconds to initialize critical services enumerated in Registry at HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Diagnostics\Performance\Boot	46.38600159	
BootDevicesInitTimeSeconds	real	Time taken in seconds to identify and initialize devices	0.894999981	
BootDriverInitTimeSeconds	real	Time taken in seconds to initialize drivers that were loaded by winload.exe	3.105999947	
BootEndTime	string	Timepoint at which the last Boot Event ended	2021-07-01T04:50:53.874Z	
BootExplorerInitTimeSeconds	real	Time taken by the system to create the desktop window manager (DWM) process, which initializes the desktop and displays it for the first time.	43.48400116	
BootKernelInitTimeSeconds	real	Time taken by windows kernel to initialize data structures and components.	0.046999998	
BootMachineGroupPolicyProcessingTimeSeconds	real	Time in seconds taken at boot time to apply Machine Group Policy settings.	1	
BootMachineProfileProcessingTimeSeconds	real	Time in seconds taken to load and apply machine profile settings at system startup.	0.169	
BootMainPathBootTimeSeconds	real	S tarts when you see the Start Windows splash screen and ends when the desktop appears.	74.66500092	
BootNumStartupApps	integer	Total number of application marked as startup apps using 'Run' or 'RunOnce' registry keys or placed in the startup folder	8	
BootOsLoaderTimeSeconds	real	Time taken by Winload.exe to load essential system drivers and initializes the system to the point where the Windows kernel can begin execution.	2.105999947	
BootOtherLogonInitActivityTimeSeconds	real	--	0.352999985	
BootPostBootTimeSeconds	real	Starts when the desktop appears. During this time, services and application may be starting but it is not considered complete until the system has reached a certain idle state.	47.95700073	
BootPrefetchInitTimeSeconds	real	Time taken by Prefetcher to initialize according to the boot plan.	0	
BootSessionInitStartTimeSeconds	real	--	4.842999935	
BootStartupAppsTimeSeconds	real	Time in seconds taken by startup applications to initialize.	18.21299934	
BootStartupDegradationApps	string	';' delimited list of application which caused boot performance degradation	Antimalware Service Executable:47.254;Cortex XDR Service:26.835	
BootStartupDegradationTimeSeconds	real	Total degradation time in seconds	74.089	
BootTimeSeconds	real	Time in seconds to boot the machine. Is the addition of BootMainPathBootTimeSeconds and BootPostBootTimeSeconds	122.6220016	
BootUserGroupPolicyProcessingTimeSeconds	real	Time in seconds taken at boot time to apply User Group Policy settings.	377	
BootUserLogonWaitTimeSeconds	real	Time in seconds the OS waited on the logon screen for the user authentication input	24.96800041	
BootUserProfileProcessingTimeSeconds	real	Time in seconds taken to load and apply user profile settings at system startup.	17.83099937	
BootWinLogonStartTimeSeconds	real	Time elapsed between the user logon screen appears and the Explorer process starts. The service control manager starts services, and Group Policy scripts run.	12.82499981	
RebootCount	integer	Number of times the machine has been rebooted on the day of last reboot.	1	
TS	integer	When the record was added to the table. See Timestamps.	1625204343	

The Tachyon Client polls the Windows Event Logs to search for a new Boot Event (Microsoft-Windows-Diagnostics-Performance/Operational [EventID =100]) . If a newer event different from the one currently stored in the Live table is present, it parses the log to extract the various boot performance metrics. We also poll other event logs (Microsoft-Windows-GroupPolicy/Operational [EventID=8000], [EventID=8001], System / [EventID=1074]) and Startup XML files generated by the OS to gather data for all the above mentioned fields. This information is used by the Tachyon Experience application.

Device interaction

The following table shows fields available in the **$DeviceInteraction**_tables.

Field	Datatype	Description	Sample value	Tables
AverageIdleResponsivenessMs	real	The number of milliseconds, on average, that the foreground application took to respond to a probe. A lower value means the foreground application is likely to feel more responsive. This field considers only samples where the user was "idle" - i.e. not interacting with the device using the keyboard/mouse.	0.73310155982222314	All
AverageIdleResponsivenessMsCount	integer	How many aggregated samples were used to derive the value for AverageIdleResponsivenessMs field.	500	• $DeviceInteraction_Hourly • $DeviceInteraction_Daily • $DeviceInteraction_Monthly
AverageInteractiveResponsivenessMs	real	The number of milliseconds, on average, that the foreground application took to respond to a probe. A lower value means the foreground application is likely to feel more responsive. This field considers only samples where the user was interacting with the device using the keyboard/mouse.	1.3999095275108098	All
AverageInteractiveResponsivenessMsCount	integer	How many aggregated samples were used to derive the value for AverageInteractiveResponsivenessMs field.	500	• $DeviceInteraction_Hourly • $DeviceInteraction_Daily • $DeviceInteraction_Monthly
AverageSessionResponsivenessMs	real	The number of milliseconds, on average, that the foreground application took to respond to a probe. A lower value means the foreground application is likely to feel more responsive. This field considers both idle and interactive samples.	1.0665055436665165	All
AverageSessionResponsivenessMsCount	integer	How many aggregated samples were used to derive the value for AverageSessionResponsivenessMs field.	500	• $DeviceInteraction_Hourly • $DeviceInteraction_Daily • $DeviceInteraction_Monthly

BusyCursorSeconds	integer	The number of seconds that the user was presented with an hourglass ("busy") cursor.	816	All
InteractionSeconds	integer	The number of seconds that the user was interacting (keyboard and mouse activity) with the device within the aggregated live sample (which corresponds to one minute by default).	8	• $Device Interaction_Live
InteractionMinutes	integer	The number of minutes that the user was interacting (keyboard and mouse activity) with the device within the hour, day or month. Note: if a user interacts at all with the device (even a single click/keystroke) in a minute period, that minute is counts towards the total InteractionMinutes value.	316	• $Device Interaction_Hourly • $Device Interaction_Daily • $Device Interaction_Monthly
LogonSeconds	integer	The number of seconds that the user was logged on to the device within the aggregated live sample (which corresponds to one minute by default).	12	• $Device Interaction_Live
LogonMinutes	integer	The number of minutes that the user was logged on to the device within the hour, day or month.	1016	• $Device Interaction_Hourly • $Device Interaction_Daily • $Device Interaction_Monthly
PresentSeconds	integer	The number of second that the user was deemed to be "present" at the device within the aggregated live sample (which corresponds to one minute by default). A user's presence is implied if the user is logged on and the device is not locked.	37	• $Device Interaction_Live

PresentMinutes	integer	The number of minutes that the user was deemed to be "present" at the device within the hour, day or month. A user's presence is implied if the user is logged on and the device is not locked.	712	• $DeviceInteraction_Hourly • $DeviceInteraction_Daily • $DeviceInteraction_Monthly
RemoteHost	string	The FQDN (or hostname or IP address, depending on availability) of the device connected remotely to this one, e.g. over a Remote Desktop session. An empty value implies a local session - i.e. the user was logged at the console. The Device interaction hourly, daily and monthly tables will aggregate sessions for each distinct combination of UserName and RemoteHost in the given period. So if an individual user connects to this device from three other devices and also from the console in a given hour/day/month, this data will be aggregated into four distinct records.	myotherdevice.acme.local	All
TS	integer	When the record was added to the table. See Timestamps.	1500756083	All
UserName	string	The name of the user interacting with the device.	1E\bill.gates	All
UserWaitSeconds	integer	The number of seconds within this sample that the user was deemed to be waiting for the device to respond. This is the total number of seconds where either the user was presented with an hourglass ("busy") cursor, and/or where the foreground application was taking longer than the defined threshold (25ms by default) to respond to probes. The threshold can be configured using the Module.Inventory.SlowMessageThresholdMs setting. **TODO - link? We're missing this setting?**	17	All

New in 5.1, this capture source is used by the Tachyon Experience application and gets data from the 1E Client UI component (controlled by the Interaction module).

The table contains one row per combination of (period + user + remote (or local) device), and summarizes both user presence/activity and overall responsiveness of applications for that user's session:

- User presence and activity is represented as logon time, present time and interaction time. Note that by definition: logon time >= present time >= interaction time.
- Application responsiveness is measured in milliseconds, and is the time taken for the foreground application to respond to a probe. Separate values are stored depending on whether the user is interacting with the device or is "idle".
- The table also includes data for "busy time" and "wait time". Busy time is when the user is presented with an hourglass cursor; wait time is when the user is either presented with an hourglass cursor OR when the foreground application is slow to respond to a probe.

Device performance

The following table shows fields available in the **$DevicePerformance**_tables.

Field	Datatype	Description	Sample value	Tables
DiskOthersAverageQueueLength	real	Average queue length for non-system disk(s). A high value indicates that these disk(s) are not keeping up with the I/O backlog.	0.041	All
DiskOthersAverageSecondsPerWrite	real	Average time taken for non-system disk(s) to perform a write. A high value indicates that these disk(s) are taking too long to service write requests.	0.00177	All
DiskOthersFreeSpaceMegabytes	integer	The free disk space across non-system disk(s). A lower value indicates that these disks are running low on space and space may need to be released.	1513894	All
DiskOthersSplitIoPerSecond	real	The I/O operations which were broken into multiple requests across non-system disk(s). A high value may indicate excessive disk fragmentation.	0.02669	All
DiskOthersUsageTimePercent	real	The percentage of time that non-system disk(s) were servicing requests. A high value indicates these disk(s) may be excessively busy.	3.1447	All
DiskSystemAverageQueueLength	real	Average queue length for the system disk. A high value indicates that this disk is not keeping up with the I/O backlog.	0.06877	All

DiskSystemAverageSecondsPerWrite	real	Average time taken for the system disk to perform a write. A high value indicates that this disk is taking too long to service write requests.	0.001319	All
DiskSystemFreeSpaceMegabytes	integer	The free disk space on the system disk. A lower value indicates that this disk is running low on space and space may need to be released.	27261	All
DiskSystemSplitIoPerSecond	real	The I/O operations that were broken into multiple requests on the system disk. A high value may indicate excessive disk fragmentation.	0.9275	All
DiskSystemUsageTimePercent	real	The percentage of time that the system disk was servicing requests. A lower score indicates this disk may be excessively busy.	3.8339	All
MemoryHardPageFaultsPerSecond	real	The amount of memory pages that had to be read from disk-based storage. A higher value indicates this device is low on available physical memory	0.2385	All
MemoryPageFileUsagePercent	real	The percentage of the page file that is in use. A higher value indicates more high page file use, which may mean the device is low on available physical memory.	9.6219	All
MemoryUsageMegabytes	integer	The amount of physical memory in use.	27946	All
MemoryUsagePercent	real	The percentage of physical memory in use. A higher value indicates higher memory consumption, and therefore less available physical memory.	85.43	All
NetworkActiveTcpConnections	integer	The average number of active (inbound and output) TCP connections.	90	All
NetworkBroadcastRate	integer	The sum of multicast packets that have been sent and received per second	5	All
NetworkBytesReceivedPerSecond	integer	The average number of bytes received per second.	2145	All
NetworkBytesSentPerSecond	integer	The average number of bytes sent per second.	1579	All
NetworkNetRetransmitRate	integer	The rate of TCP segment (both IPv4 and IPv6) retransmissions done per second.	1	All
NetworkOutputQueueLength	integer	The length of the output packet queue (in packets).	0	All
NetworkPacketRate	integer	the sum of all TCP packets that have been sent and received per second.	71	All
NetworkPacketsOutboundDiscarded	integer	The number of outgoing packets dropped by the network adapter.	1	All
NetworkPacketsOutboundErrors	integer	The number of outgoing packets dropped by the network adapter due to errors at physical layer	2	All
NetworkPacketsReceivedDiscarded	integer	The number of incoming packets dropped due to non availability of receive buffers to store the incoming frames.	1	All
NetworkPacketsReceivedErrors	integer	The number of packets dropped by the network adapter due to various errors at physical layer	4	All
NetworkPacketsReceivedNonUnicastPerSecond	integer	The rate that non-unicast, that is, subnet broadcast or subnet multicast packets, are delivered to a higher-layer protocol.	7	All
NetworkPacketsSentNonUnicastPerSecond	integer	The rate that packets are requested to be transmitted to non-unicast, that is, subnet broadcast or subnet multicast, addresses by higher-layer protocols.	3	All
NetworkSaturationPercent	integer	Not implemented. Deprecated in v8.0 in favour of NetworkUtilizationPercent	0 ; NULL (v8.0 onwards)	All
ActiveUserSampleCount	integer	Number of samples used to form NetworkUsageAverageUserLoggedIn aggregated data	1	All
NetworkUsageAverageUserLoggedIn	integer	NetworkUtilizationPercentage value when the user is active on the machine.	23	All
NetworkUtilizationPercent	integer	A percentage value that represents the ratio of total network traffic to the reported maximum bandwidth supported by the interface.	25	All
ProcessorInterruptTimePercent	real	The amount of time the CPU spent servicing interrupts. A higher value may indicate faulty or misconfigured hardware/drivers.	0.23485	All
ProcessorQueueLength	real	The CPU queue length (backlog of processing work). A higher value means that the CPU is not keeping up with the workload.	0.33	All
ProcessorTimePercent	real	The CPU load. A higher value indicates that the CPU is fully loaded and additional processing power may be required.	15.10	All
ProcessorTimeSeconds	real	The average number of CPU seconds (amount of processor work) per second. A value of 1 indicates that a single CPU core was entirely busy for a second.	143.12	All
TCPv4ConnectionsEstablished	integer	The number of TCPv4 connections established by the endpoint	287	All
TCPv6ConnectionsEstablished	integer	The number of TCPv6 connections established by the endpoint	12	All

SampleCount	integer	Number of samples used to form this aggregated data	4244	• $DevicePerformance_Hourly • $DevicePerformance_Daily • $DevicePerformance_Monthly
TS	integer	When the record was added to the table. See Timestamps.	1500756083	All
UserRating	integer	Not yet implemented.	0	All
W irelessReceiveRate	integer	Th e maximum possible link layer receive transfer speed that can be expected from the Wireless networking on the endpoint.	130000	All
WirelessSignalQuality	integer	A percentage value that represents the signal quality of the wireless network capability on the endpoint.	80	All
WirelessTransmitRate	integer	The maximum possible link layer send transfer speed that can be expected from the Wireless networking on the endpoint.	71500	All
WirelessSampleCount	integer	Number of samples used to form Wireless aggregated data	1	All

New in 5.0, this capture source is used by the Tachyon Experience application.
The Network Metrics collect data for the Primary Network Interface.

Device resource demand

The following table shows fields available in the **$DeviceResourceDemand**_tables.

Field	Datatype	Description	Sample value	Tables
AllocatedMB	integer	How much memory, in megabytes, was allocated to this device on average during this sample.	8192	All
AllocatedMips	integer	How much CPU resource, in Mips (millions of instructions per second), was allocated to this device on average during this sample.	34848	All
CpuMips	integer	How much CPU resource in Mips was being used by this device on average during this sample.	1660	All
DiskKBps	integer	How much disk throughput (in kilobytes per second) was used by this device on average during this sample. Throughput is measured across all fixed disks.	739	All
MemoryMB	integer	How much memory, in megabytes, was used by this device on average during this sample.	5883	All
NetKBps	integer	How much network throughput (in kilobits per second) was used by this device on average during this sample. Throughput is measured across all network adapters.	121	All
Rttms	integer	The roundtrip time, in milliseconds, for user input to be processed by the virtualization infrastructure. Note that only Citrix virtualization technology is supposed when determining the roundtrip time.	3	All
SampleCount	integer	Number of samples used to form this aggregated data	123	• $DeviceResourceDemand_Hourly • $DeviceResourceDemand_Daily • $DeviceResourceDemand_Monthly
Server	string	Where available, the FQDN (or hostname or IP address) of the virtual server that is hosting this virtual machine. If the device is not virtualized, or if the virtual server information is not available, this field will be empty. Note that only Hyper-V and Citrix virtualization technologies are supported when determining the name of the virtual server.	myvirtualserver.acme.local	All
TS	integer	When the record was added to the table. See Timestamps.	1500756083	All

New in 5.1, this capture source is used by the Tachyon Experience application.

This data is particularly useful when it has been collected virtual machines - it allows you to compare allocated resources with actual resource demand. This can give insight into whether virtual resources are under- or over-provisioned.

DNS resolutions

The following table shows fields available in the **$DNS**_tables.

Field	Datatype	Description	Sample value	Tables
Fqdn	string	The FQDN which is being resolved.	client-office365-tas.msedge.net	All
LookupCount	integer	Sum of resolutions per FQDN within the hour, day, month.	1234	• $DNS_Hourly • $DNS_Daily • $DNS_Monthly
TS	integer	When the record was added to the table. See Timestamps.	1500756083	All

When using polling, the local DNS cache is queried for all unique FQDNs. This includes an initial scan of cache entries created before the Tachyon client starts, which are stored with the same timestamp. New entries that appear in the cache are deemed to correspond to new resolutions and stored with the timestamp of when the polling occurred.

When using ETW, the Tachyon client attempts to capture DNS queries at the point that they are made. The query is captured, not the result of that query. That is, the Tachyon client will capture a request to resolve an FQDN which may ultimately not be resolvable. The DNS cache is not scanned.

Operating System performance

The following table shows fields available in the **$OperatingSystemPerformance**_tables.

<table>
<tr><th>Field</th><th>Datatype</th><th>Description</th><th>Sample value</th><th>Tables</th></tr>
<tr><td>CpuSeconds</td><td>real</td><td>The time taken in seconds (on average) to run the test for the corresponding metric.</td><td>9.1E-05</td><td>All</td></tr>
<tr><td>ExecutionCount</td><td>integer</td><td>The number of times that the test for this metric was run within the hour, day, month.</td><td>1</td><td>$OperatingSystemPerformance_Hourly
$OperatingSystemPerformance_Daily
$OperatingSystemPerformance_Monthly</td></tr>
<tr><td>Metric</td><td>string</td><td colspan="2">A row for each of the following 15 metrics:
<table>
<tr><th>Metric</th><th>Description</th></tr>
<tr><td>BootTime</td><td>Most recent time taken in seconds for the device to boot to the logon prompt. A long boot up duration may influence user satisfaction..</td></tr>
<tr><td>CreateFile</td><td>Time taken to create an empty, temporary file. A high value indicates that the operating system may be underperforming for basic file operations.</td></tr>
<tr><td>CreateProcess</td><td>The time taken to create a new process. A high value indicates that the operating system is taking a long time to create new processes.</td></tr>
<tr><td>CreateThread</td><td>The time taken to create a new thread within a process. A high value indicates that the operating system is taking a long time to create new threads.</td></tr>
<tr><td>CreateWindow</td><td>The time taken to generate an empty window. A high value indicates poor Windows desktop performance, which may cause applications to appear unresponsive.</td></tr>
<tr><td>DiskRand</td><td>The time taken to perform a random access disk operation on the system drive. A high value indicates poor random disk access performance.</td></tr>
<tr><td>DiskSeq</td><td>The time taken to perform a sequential access disk operation on the system drive. A high value indicates poor sequential disk access performance.</td></tr>
<tr><td>LoadDLL</td><td>The time taken to load and unload a DLL in a process. A high value indicates library loading is slow, which may affect the startup time of applications.</td></tr>
<tr><td>Memory</td><td>The time taken to allocate, zero and free a block of memory. A high value indicates that the operating system is slow to serve memory requests, which may affect application performance.</td></tr>
<tr><td>MessageDispatch</td><td>The time taken to dispatch and confirm processing of a windows message. A high value indicates poor message processing throughput, which may affect application responsiveness.</td></tr>
<tr><td>OpenHandle</td><td>The time taken to acquire a basic operating system resource. A high value indicates that applications may underperform because the operating system is slow to service resource requests.</td></tr>
<tr><td>RegReadHKLM</td><td>The time taken to read from the HKLM Windows Registry hive. A high value indicates poor registry read performance, which in turn may affect application performance.</td></tr>
<tr><td>RegWriteHKCU</td><td>The time taken to write to the HKCU Windows Registry hive. A high value indicates poor registry write performance, which in turn may affect application performance.</td></tr>
<tr><td>RegWriteHKLM</td><td>The time taken to write to the HKLM Windows Registry hive. A high value indicates poor registry write performance, which in turn may affect application performance.</td></tr>
<tr><td>UDPSend</td><td>The time taken to perform a basic loopback UDP send. A high value may indicate problems with the operating system network stack</td></tr>
</table></td><td>All</td></tr>
<tr><td>TS</td><td>integer</td><td>When the record was added to the table. See Timestamps.</td><td>1500756083</td><td>All</td></tr>
</table>

New in 5.0, this capture source is used by the Tachyon Experience application.

The Operating System Performance tables store the average time taken to run various tests against the operating system. These tests measure Operating System performance by timing common OS-level tasks such as creating processes and threads, reading and writing to the registry, etc.

Performance event

The following table shows fields available in the **$PerformanceEvent**_tables.

<table>
<tr><th>Field</th><th>Datatype</th><th>Description</th><th>Sample value</th><th>Tables</th></tr>
<tr><td>EventData</td><td>string</td><td>JSON-formatted data, specific to the EventType, containing additional information about this event.</td><td>(json data)</td><td>$PerformanceEvent_Live</td></tr>
<tr><td>EventCount</td><td>integer</td><td>A count of the number of events of this EventType and ReferenceItem combination for the hour, day, or month.</td><td>123</td><td>$PerformanceEvent_Hourly
$PerformanceEvent_Daily
$PerformanceEvent_Monthly</td></tr>
<tr><td>EventType</td><td>string</td><td>The type of event, in dotted-namespace notation. Events captured are:
<table>
<tr><th>EventType</th><th>Description</th><th>ReferenceItem</th></tr>
<tr><td>Device.PerformanceAnomaly</td><td>Reserved for future use</td><td>N/A</td></tr>
<tr><td>OperatingSystem.Boot</td><td>Occurs when the operating system starts up</td><td>N/A</td></tr>
<tr><td>OperatingSystem.Crash</td><td>Occurs when the operating system crashes (BSOD)</td><td>N/A</td></tr>
<tr><td>OperatingSystem.PerformanceAnomaly</td><td>Reserved for future use</td><td>N/A</td></tr>
<tr><td>OperatingSystem.ServiceFailure</td><td>Occurs when a service fails to start</td><td>Service description</td></tr>
<tr><td>OperatingSystem.Upgrade</td><td>Occurs when the version number of the operating system changes</td><td>N/A</td></tr>
<tr><td>Patch.Install</td><td>Occurs when a patch is installed</td><td>N/A</td></tr>
<tr><td>Patch.Uninstall</td><td>Occurs when a patch is removed</td><td>N/A</td></tr>
<tr><td>Software.Crash</td><td>Occurs when a process crashes</td><td>Executable path of faulted process</td></tr>
<tr><td>Software.Hang</td><td>Occurs when an application becomes unresponsive</td><td>Executable path of fauled process</td></tr>
<tr><td>Software.Install</td><td>Occurs when an application is installed</td><td>Product name</td></tr>
<tr><td>Software.PerformanceAnomaly</td><td>Reserved for future use</td><td>N/A</td></tr>
<tr><td>Software.Uninstall</td><td>Occurs when an application is removed</td><td>Product name</td></tr>
</table></td><td>Software.Install</td><td>All</td></tr>
<tr><td>ReferenceItem</td><td>string</td><td>A simplified representation, where applicable, of the object to which the event pertains. See the EventType field description for more details.</td><td>Microsoft Edge WebView2 Runtime</td><td>All</td></tr>
<tr><td>TS</td><td>integer</td><td>When the record was added to the table. See Timestamps.</td><td>1500756083</td><td>All</td></tr>
</table>

New in 5.0, this capture source is used by the Tachyon Experience application.

The data in this table is used to derive a count-over-time of events which may be of relevance when diagnosing performance or end-user experience issues.

Process executions

All platforms except Android. The following table shows fields available in the **$Process**_tables.

<table>
<tr><th>Field</th><th>Datatype</th><th>Description</th><th>Sample value</th><th>Tables</th></tr>
<tr><td>CommandLine</td><td>string</td><td>The full command-line of the process, including (on Windows) the executable name.

Sometimes the executable name part of the command-line is quoted, sometimes it's not - it's arbitrary based however the parent process launched the child; so you may see a mix of command-lines like...
"C:\Program Files (x86)\Microsoft Office\root\Office16\OUTLOOK.EXE"
\??\C:\Windows\system32\conhost.exe 0x4
C:\Windows\system32\svchost.exe -k UnistackSvcGroup</td><td>"C:\Windows\system32\VmConnect.exe" "1EUKDEVWKS1231" "TCH-CLI-WXPX86" -G "B2C72520-BBC6-4736-BBBC-5CCF50FE6666" -C "0"</td><td>$Process_Live</td></tr>
<tr><td>ExecutableHash</td><td>string</td><td>The MD5 hash of the process executable.</td><td>dae0bb0a7b2041115cfd9b27d73e0391</td><td>All</td></tr>
</table>

ExecutableName	string	The filename (including extension) of the process executable.	vmconnect.exe	All
ExecutablePath	string	The path and filename of the process executable. On Windows, this is the NT-device format version of the path (as a process does not necessarily need to have been launched from a device which has a drive-letter mapping).	\device\harddiskvolume8\windows\system32\vmconnect.exe	All
ExecutionCount	integer	Sum of executions per executable within the hour, day, month.	1234	• $Process_Hourly • $Process_Daily • $Process_Monthly
ParentExecutableName	string	The filename (including extension) of the executable of the process which spawned this one.	mmc.exe	All
ParentProcessId	integer	The process ID of the process which spawned this one.	2088	• $Process_Live
ProcessId	integer	Operating-system dependent process ID.	178	• $Process_Live
TS	integer	When the record was added to the table. See Timestamps.	1500756083	All
UserName	string	The name of the user in whose session the process was launched (or blank if it is a system-launched process)	1E\bill.gates	All

On Windows, the 1E Client service runs as LOCAL SYSTEM, therefore details of almost every process will be available to the Tachyon client features; however some processes may not be accessible because of permissions.

The Tachyon client captures process starts; it does not track how long the process has been running, or how much CPU-time (or user/kernel/active time) the process has used.

Each time the Tachyon client starts it does an initial scan of processes before it starts capturing, and will use that time to record when these processes started.

The UserName field is derived from the session in which the process was executed, and doesn't necessarily reflect the user in whose context the process was executed.

Process stabilizations

Windows only. The following table shows which OS and polling methods are available for **Process stabilizations**.

Field	Datatype	Description	Sample value	Tables
ExecutableName	string	The filename (including extension) of the process executable.	vmconnect.exe	All

Execution Count	integer	Sum of executions per executable and username within the hour, day, month. For example, vmconnect.exe run by 1e\user1 and vmconnect.exe run by 1e\user2 will have separate rows and thus will be summed separately.	53	• $ProcessStabilization_Hourly • $ProcessStabilization_Daily • $ProcessStabilization_Monthly
ProcessId	integer	Operating-system dependent process ID.	178	• $ProcessStabilization_Live
StabilizationTimeMs	integer	The time taken for the process to be considered stable, measured in milliseconds. This will be a multiple of 100.	4500	• $ProcessStabilization_Live
TotalStabilizationTimeMs	integer	Sum of the time taken to be considered stable per executable and username within the hour, day, month. For example, vmconnect.exe run by 1e\user1 and vmconnect.exe run by 1e\user2 will have separate rows and thus will be summed separately.	864300	• $ProcessStabilization_Hourly • $ProcessStabilization_Daily • $ProcessStabilization_Monthly
TS	integer	When the record was added to the table. See Timestamps.	1500756083	All
UserName	string	The name of the user in whose session the process was launched (or blank if it is a system-launched process)	1e\bill.gates	All

The following table shows fields available in the **$ProcessStabilization**_tables.

On Windows, the 1E Client service runs as `LOCAL SYSTEM`, therefore details of almost every process will be available; however some processes may not be accessible because of permissions. The Tachyon client captures only information that can be accessed by `LOCAL SYSTEM` - as such it does not check the UI responsiveness of a process.

By default, process stabilization monitoring is not active. To enable, the process names must be specified in the 1E Client configuration file as follows:

- Add `Module.Inventory.ProcessStabilization.MonitoredProcesses=<string>` to the 1E Client configuration file.

- This is a list of comma separated values, and the case is not significant. For example, `winword.exe` and `WINWORD.EXE` are treated the same.
- The list of monitored processes does not currently have a limit, however adding a large list of processes to monitor can cause performance degradation and the process stabilization time will become less accurate.

A process' resource usage is tracked, and it will be considered stable once it's resource utilisation has stopped fluctuating. The margin in which a process is considered stable can be modified in the 1E Client configuration file. Changing from default is **not** recommended.

- This margin is controlled by the `fuzziness` configuration setting.
- Add `Module.Inventory.ProcessStabilization.Fuzziness=<integer>` to the 1E Client configuration file. It cannot be lower than 1, and cannot exceed 66. The default is 5.

A process that exits before it is considered stable is not be recorded. Currently, such processes are discarded. A warning is logged when this occurs.

The accuracy of process monitoring decreases if more processes need to be monitored concurrently. For example, accuracy will decrease if many processes are started at the same time. Warnings are logged when this occurs.

The accuracy of the process monitoring decreases if the system is under considerable load, for example high disk or CPU stress.

Aggregation is grouped by the `UserName` and `ExecutableName` fields. Unlike process executions, process stabilization values for `UserName` and `ExecutableName` are lower case.

Process usage

Windows only. The following table shows fields available in the **$ProcessUsage_Daily** table.

Field	Datatype	Description	Sample value	Tables
CommandLine	string	This is a single instance of the command used to launch that instance, most probably the first one. It will not contain any differences if other instances are launched with a slightly different comand line. It is an indication of a typical command line for this instance.	C:\Program Files\Git\mingw64\libexec\git-core\git-credential-manager.exe	
Duration	integer	The number of minutes covered by the individual execution(s) of at least one instance of this executable. Duration can never be more than 1440 minutes, being the number of minutes in a day.	1	
ExecutableHash	string	The MD5 hash of the binary that contains the entry point (usually an exe)	ad3ec70ae9e82582bdf6aa6fd5811376	
ExecutableName	string	The name of the binary that contains the entry point obtained from stamped version information where possible, the filename if not.	git-credential-manager.exe	
ExecutableSize	integer	The size of the binary that is hashed below	131168	
ExecutableVersion	string	The version information stamped into the executable where available.	1.5.0.0	
ExecutionCount	integer	The number of instances observed during the Duration period	2	
IsOSProcess	integer	A value of 1 indicates that this is categorised as an operating system by the rules in place. A value of 0 indicates that it is not.	0	
LastSeen	integer	The UTC Timestamp of when the last instance of the executable (of all the accumulated subjects of this record) was last seen (polling) or actually exited (events). Whilst any instance is running, for the current day records, LastSeen will creep across the day and duration will increase as time passes if the process remains running. Once midnight is crossed then the daily records for yesterday are 'closed off' by setting LastSeen = TS + 86400 (the number of seconds in a day), which is midnight of the next day. If all instances of one binary are exited and never run again that day, then the LastSeen field for that daily record should 'stick' at one value and never ever change again. In other words the maximum difference between TS and LastSeen in a single row is at most 86400, being the number of seconds in a day. Tracking of an execution summary from one day to another ("carry-over") can be achieved by looking for a record based on $TS_{tomorrow}$ = $LastSeen_{today}$ with all the other key information the same. If that exact key record with the 'carry over' conditions is not found then the process did not theoretically continue across midnight. Note that a process that dies after 23:59:00 and starts before 00:01:00 the next day will appear to be a continuous process in the summary tables. Even though it could theoretically have stopped for nearly two minutes. This is because the resolution of the table is to the start of the minute the event occurred in.	1526982245	
TS	integer	When the record was added to the table. See Timestamps. Midnight UTC that is the start day of the 24 hours covered by this record.	1526947200	

The Tachyon client captures executable usage; this is from the moment the executable is turned into a process, hence the process usage. The Process Usage data presented is grouped by executable binary, and parallel runs are accumulated in the ExecutionCount, but not in the Duration, where coverage time period is desired instead.

Sensitive processes

A "sensitive process" is one flagged by the Tachyon Performance Metrics program as one which consumes extra CPU when files and processes are created, registry entries are read, etc., suggesting such processes are monitoring such low level O/S operations. Antivirus and other security software legitimately does this (as does for example Windows Explorer and the 1E Client itself), but other processes that do it *may* be a security hazard.

Windows only. The following table shows fields available in the **$SensitiveProcess**_tables.

Field	Datatype	Description	Sample value	Tables
CpuSeconds	real	Average CPU used by the process executable during the sample intervals.	0.456	All
DetectionCount	integer	Sum of the number of samples within the hour, day, month in which the process executable was detected.	1	• $SensitiveProcess_Hourly • $SensitiveProcess_Daily • $SensitiveProcess_Monthly
ExecutablePath	string	The path and filename of the process executable. On Windows, this is the NT-device format version of the path (as a process does not necessarily need to have been launched from a device which has a drive-letter mapping).	c:\windows\system32\conhost.exe	All
Product	string	The title of the software product.	Microsoft® Windows® Operating System	All
TS	integer	When the record was added to the table. See Timestamps.	1500756083	All
Version	string	Version of the process executable.	10.0.17763.404	All

New in 5.0 this capture source is used by the Tachyon Experience application.

> ⚠ For Windows XP permissions restrictions mean that not all sensitive processes are detected.

Software installations

All platforms except Android. The following table shows fields available in the **$Software**_tables.

Field	Datatype	Description	Sample value	Tables
Architecture	string	The platform architecture of the software product.	x64	All
InstallCount	integer	Sum of installs per software product version within the hour, day, month. 0 if uninstalled, or present but not detected as installed.	1234	• $Software_Hourly • $Software_Daily • $Software_Monthly
IsUninstall	integer	0 = install, 1 = uninstall.	0	• $Software_Live
Product	string	The title of the software product that was installed/uninstalled.	Google Chrome	All
Publisher	string	The publisher of the software product that was installed/uninstalled.	Google Inc.	All
TS	integer	When the record was added to the table. See Timestamps. The Tachyon client assumes a "new" installation/uninstallation occurred at the point of polling.	1500756083	All
UninstallCount	integer	Sum of uninstalls per software product version within the hour, day, month. 0 if installed, or present but not detected as installed.	1233	• $Software_Hourly • $Software_Daily • $Software_Monthly
Version	string	The version of the software that was installed/uninstalled.	55.0.2883.87	All

Each time the Tachyon client starts it does an initial scan of installed software before it starts capturing. Since the Tachyon client has no way of knowing when this install/uninstall happened, it will mark the event as having occurred "now".

On Windows, software installations are read from the registry key `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall` and `HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall`

Per-user installations are not yet supported.

Linux does not distinguish between O/S packages (even the kernel) and application packages; they are all software.

Software interaction

The following table shows fields available in the **$SoftwareInteraction**_tables.

Field	Datatype	Description	Sample value	Tables
AverageIdleResponsivenessMs	real	The number of milliseconds, on average, that this application took to respond to a probe when it was in the foreground. A lower value means the application is likely to feel more responsive. This field considers only samples where the user was "idle" - i.e. not interacting with the device using the keyboard/mouse.	0.73310155982222314	All
AverageIdleResponsivenessMsCount	integer	How many aggregated samples were used to derive the value for AverageIdleResponsivenessMs field.	500	• $SoftwareInteraction_Hourly • $SoftwareInteraction_Daily • $SoftwareInteraction_Monthly
AverageInteractiveResponsivenessMs	real	The number of milliseconds, on average, that this application took to respond to a probe when it was in the foreground. A lower value means the application is likely to feel more responsive. This field considers only samples where the user was interacting with the device using the keyboard/mouse.	1.3999095275108098	All
AverageInteractiveResponsivenessMsCount	integer	How many aggregated samples were used to derive the value for AverageInteractiveResponsivenessMs field.	500	• $SoftwareInteraction_Hourly • $SoftwareInteraction_Daily • $SoftwareInteraction_Monthly
AverageSessionResponsivenessMs	real	The number of milliseconds, on average, that this application took to respond to a probe when it was in the foreground. A lower value means the foreground application is likely to feel more responsive. This field considers both idle and interactive samples.	1.0665055436665165	All
AverageSessionResponsivenessMsCount	integer	How many aggregated samples were used to derive the value for AverageSessionResponsivenessMs field.	500	• $SoftwareInteraction_Hourly

				• $SoftwareInteraction_Daily • $SoftwareInteraction_Monthly
BusyCursorSeconds	integer	The number of seconds that the user was presented with an hourglass ("busy") cursor while this application was in the foreground.	816	All
InteractionSeconds	integer	The number of seconds that the user was interacting (keyboard and mouse activity) with this application within the aggregated live sample (which corresponds to one minute by default).	8	• $SoftwareInteraction_Live
InteractionMinutes	integer	The number of minutes that the user was interacting (keyboard and mouse activity) with this application within the hour, day or month. Note: if a user interacts at all with the device (even a single click/keystroke) in a minute period, that minute is counts towards the total InteractionMinutes value.	316	• $SoftwareInteraction_Hourly • $SoftwareInteraction_Daily • $SoftwareInteraction_Monthly
LogonSeconds	integer	The number of seconds that the user was logged on to the device within the aggregated live sample (which corresponds to one minute by default) with this application in the foreground.	12	• $SoftwareInteraction_Live
LogonMinutes	integer	The number of minutes that the user was logged on to the device within the hour, day or month with this application in the foreground.	1016	• $SoftwareInteraction_Hourly • $SoftwareInteraction_Daily • $SoftwareInteraction_Monthly
PresentSeconds	integer	The number of second that the user was deemed to be "present" at the device within the aggregated live sample (which corresponds to one minute by default) with this application in the foreground. A user's presence is implied if the user is logged on and the device is not locked.	37	• $SoftwareInt

				eraction_Live
PresentMinutes	integer	The number of minutes that the user was deemed to be "present" at the device within the hour, day or month with this application in the foreground. A user's presence is implied if the user is logged on and the device is not locked.	781	• $SoftwareInteraction_Hourly • $SoftwareInteraction_Daily • $SoftwareInteraction_Monthly
ProcessName	string	The name of the foreground process (without the .exe extension) to which this record pertains.	chrome	All
RemoteHost	integer	The FQDN (or hostname or IP address, depending on availability) of the device connected remotely to this one, e.g. over a Remote Desktop session. An empty value implies a local session - i.e. the user was logged at the console. The Software interaction hourly, daily and monthly tables will aggregate sessions for each distinct combination of UserName, ProcessName and RemoteHost in the given period. So if an individual user connects to this device from three other devices and also from the console in a given hour/day/month, and has this process in the foreground, this data will be aggregated into four distinct records.	myotherdevice.acme.local	All
TS	integer	When the record was added to the table. See Timestamps.	1500756083	All
UserName	string	Not yet implemented.	1E\bill.gates	All
UserWaitSeconds	integer	The number of seconds within this sample that the user was deemed to be waiting for the device to respond while this application was in the foreground. This is the total number of seconds where either the user was presented with an hourglass ("busy") cursor, and/or where the foreground application was taking longer than the defined threshold (25ms by default) to respond to probes. The threshold can be configured using the Module.Inventory.SlowMessageThresholdMs setting.	17	All

New in 5.1, this capture source is used by the Tachyon Experience application and gets data from the 1E Client UI component (controlled by the Interaction module).

The table contains one row per combination of (period + user + process name + remote (or local) device), and summarizes both user presence/activity and overall responsiveness for each application which enters the foreground for that user's session:

- User presence and activity is represented as logon time, present time and interaction time. Note that by definition: logon time >= present time >= interaction time.
- Application responsiveness is measured in milliseconds, and is the time taken for the foreground application to respond to a probe. Separate values are stored depending on whether the user is interacting with the application or is "idle".
- The table also includes data for "busy time" and "wait time". Busy time is when the user is presented with an hourglass cursor; wait time is when the user is either presented with an hourglass cursor OR when the application is slow to respond to a probe.

Software performance

Windows only. The following table shows fields available in the **$SoftwarePerformance**_tables.

Field	Datatype	Description	Sample value	Tables
ExecutablePath	string	The path and filename of the process executable.	c:\windows\explorer.exe	All
HandleCount	integer	How many open handles the process has	1234	All
InstanceCount	integer	How many instances of the process are active at the same time	2	All
IoReadKilobytesPerSecond	integer	kB read by the process per second	2	All

IoWriteKilobytesPerSecond	integer	kB written by the process per second	0	All
MemoryUsagePhysicalKilobytes	integer	kB used by the process in physical memory	35780	All
MemoryUsageVirtualKilobytes	integer	kB used by the process in virtual memory	9496	All
ProcessorTimePercent	real	Percentage of time the processor is running the process	1.11080672689126	All
ProcessorTimeSecondsPerSecond	real	Number of processor seconds consumed by this process per second, where a "processor second" is a single processor core which is fully busy for an entire second.	0.0666526784132013	All
DiskReadIops	integer	Disk read operations done by a process per second	6	All
DiskWriteIops	integer	Disk write operations done by a process per second	2	All
NetworkBytesExchangedPerSecond	integer	Total network bytes (Sent + Received) exchanged by a process per second	1485956	All
NetworkBytesReceivedPerSecond	integer	Network bytes received by a process per second	1597601	All
NetworkBytesSentPerSecond	integer	Network bytes sent by a process per second	1897601	All
NetworkTotalDataReceivedMegabytes	integer	MB received by the process	94	All
NetworkTotalDataSentMegabytes	integer	MB sent by the process	214	All
Product	string	The title of the software product.	Google Chrome	All
SampleCount	integer	How many samples the aggregated data is based on.	1233	• $SoftwarePerformance_Hourly • $SoftwarePerformance_Daily • $SoftwarePerformance_Monthly
NetworkSampleCount	integer	How many samples the per process Network aggregated data is based on.	3	All
DiskSampleCount	integer	How many samples the per process Disk i/o aggregated data is based on.	3	All
TS	integer	When the record was added to the table. See Timestamps.	1500756083	All
Version	string	The version of the software product.	55.0.2883.87	All

New in 5.0, this capture source is used by the Tachyon Experience application.

New in 8.0, the capture sources **SoftwarePerformace.DiskUsage** (collecting DiskReadIops, DiskWriteIops) and **SoftwarePerformance.ProcessNetworkUsage** (collecting NetworkBytesExchangedPerSecond, NetworkBytesReceivedPerSecond, NetworkBytesSentPerSecond, NetworkTotalDataReceivedMegabytes, NetworkTotalDataSentMegabytes) do not have Aggregation tables of their own. The software performance data collected by them is included in the **$SoftwarePerformance** tables. These capture sources can be disabled in isolation to one other but if the parent **SoftwarePerformance** capture source is disabled then they are also disabled.

TCP outbound connections

Windows, macOS and Linux only. Not Solaris or Android. The following table shows fields available in the **$TCP**_tables.

Field	Datatype	Description	Sample value	Tables
ConnectionCount	integer	Sum of connections to an IP Address and Port by a process within the hour, day, month.	123	• $TCP_Hourly • $TCP_Daily • $TCP_Monthly
IpAddr	string	The target remote IP address of the connection, either an IPv4 or IPv6 address.		All

ess		Windows support for IPV6 is limited; the Tachyon client will capture the connections, but the format used to represent the target IPV6 may differ slightly depending on the mechanism used, and may be subject to change in future versions of the Windows Tachyon client.	132.245.77.18 [2001:4860:4860::8888]	
Port	integer	The target remote port of the connection.	443	All
ProcessId	integer	The operating-system specific identifier of the process which instigated the connection. Not supported for Mac OSX earlier than Mac OSX Lion (10.7).	11828	• $TCP_Live
ProcessName	string	The executable filename of the process which instigated the connection Connections originated from system-oriented processes are captured as "(system)"	chrome.exe	All
TS	integer	When the record was added to the table. See Timestamps.	1500756083	All

The Tachyon client captures TCP connections, not UDP connections - as UDP is inherently connectionless (each packet sent is effectively a new connection).

Each time the Tachyon client starts it does an initial scan of connections before it starts capturing. A limitation of the Windows API is means that all established TCP connections, whether inbound or outbound, are captured; there is no way to distinguish between the two. This means that it is possible for the Tachyon client to double-capture a connection if that connection was established before the Tachyon client stops monitoring, and still exists when the Tachyon client starts monitoring again, for example between Tachyon client restarts. Unlike other capture sources, there is no persistent storage setting to prevent double-counting.

The Tachyon client captures initial "connect" requests, not just successful connection establishment. This means that an attempt to perform a connection will be captured, even if that connection does not complete, for example, because of a timeout, or the server-side does not permit the connection.

User usage

Windows only. The following table shows fields available in the **$UserUsage_Daily** table.

Field	Datatype	Description	Sample value	Tables
Duration	integer	The number of minutes covered by the individual user session(s) of at least one instance of this login. Duration can never be more than 1440 minutes, being the number of minutes in a day.	12	• $UserUsage_Daily
Email	string	The email address that is cached in the system for this user. This may not necessarily be the email address to use to contact the user via corporate email.	abrown@acme.org	• $UserUsage_Daily
FirstName	string	The forename that the system has cached for the user.	Alice	• $UserUsage_Daily
LastName	string	The surname that the system has cached for the user	Brown	• $UserUsage_Daily
LastSeen	integer	The UTC Timestamp of when the last instance of the user session (of all the accumulated subjects of this record) was last seen (polling) or actually exited (events), rounded down to the start of the minute in which the event occurred. Whilst any session is in progress, for the current day records, LastSeen will creep across the day and the duration will increase as time passes if the user remains logged in. That is Duration and LastSeen will increase each time you query the table (with at least a minute between queries). Once midnight is crossed then the daily records for yesterday are 'closed off' by setting LastSeen = TS + 86400 (the number of seconds in a day), which is midnight of the next day. If all users sessions for one user are exited and never occur again that day, then the LastSeen field for that daily record should 'stick' at one value and never ever change again. In other words the maximum difference between TS and LastSeen in a single row is at most 86400, being the number of seconds in a day.	1526990846	• $UserUsage_Daily

		Tracking of a user session summary from one day to another ("carry-over") can be achieved by looking for a record based on $TS_{tomorrow}$ = $LastSeen_{today}$ with all the other key information the same. If that exact key record with the 'carry over' conditions is not found then the user session did not theoretically continue across midnight. Note that a session that exits after 23:59:00 and starts again before 00:01:00 the next day will appear to be a continuous user session in the summary tables. Even though it could theoretically have not existed for nearly two minutes. This is because the resolution of the table is to the start of the minute the event occurred in. See Timestamps.		
SID	string	The Windows NT SID of the user.	S-1-5-21-xxx-yyy-zzz	• $UserUsage_Daily
TS	integer	When the record was added to the table. See Timestamps.	1526947200	• $UserUsage_Daily
Username	string	The user account name, with a domain prefix if applicable. For Windows devices not a in a domain, the 'domain' is the local machine name. For non-Windows devices such as Linux there is no domain part.	aliceb acme\Alice Brown	• $UserUsage_Daily

The Tachyon client captures user sessions (usage); this is from the moment the user instigates a login/logout, hence User Usage. The usage data presented is grouped by SID and Username, and parallel login durations are really the coverage of the time period, not the total time for all the individual sessions.

User and administrator accounts are included, either local or remote. System accounts, and accounts used to run services, are excluded.

Constraints of Legacy OS

In this documentation, the following are referred to as legacy OS. 1E does not provide support for the Tachyon client on these OS. This is because Microsoft has withdrawn support for these OS or they are not significantly used by business organizations.

• **Windows XP** • **Windows Vista** • **Windows 7** • **Windows 8.0**	• **Windows Server 2003** • **Windows Server 2008** • **Windows Server 2008 R2** • **Windows Server 2012** • **Windows Server 2012 R2**

Please contact 1E if you require support for these legacy OS.

If you experience an issue on these OS, then please try replicating the issue on a supported OS.