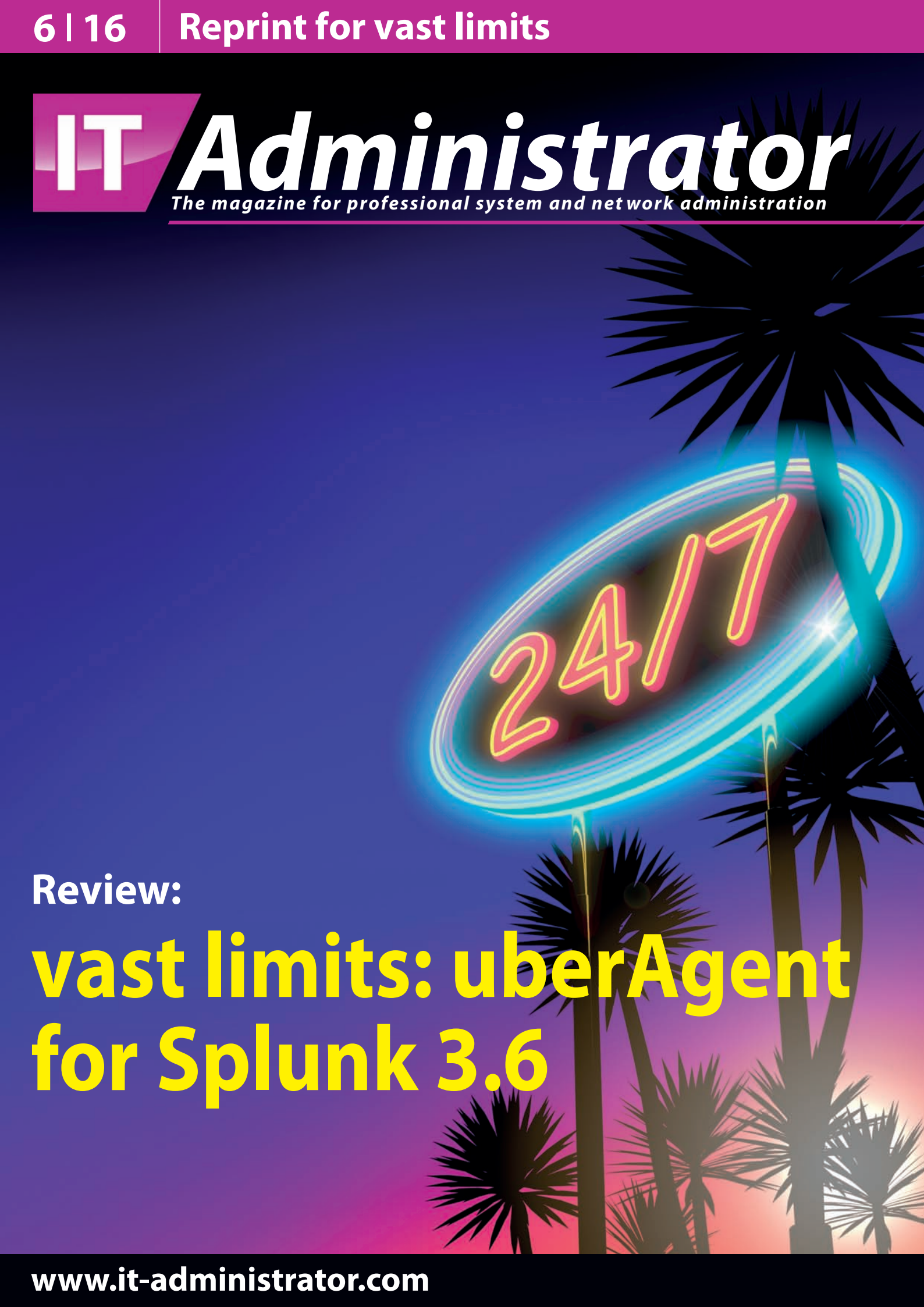
IT Administrator
The magazine for professional system and network administration



Review:

vast limits: uberAgent for Splunk 3.6

vast limits: uberAgent for Splunk 3.6

An agent in the know

Source: M.R. Bin Husin – 123RF

By Christian Knermann

The end user experience is shaped by many different factors, particularly when users roam around virtual environments. But when a problem occurs, the job of looking for the needle in a haystack begins. In a test, uberAgent, an extension for Splunk, helps to significantly ease the process.

As a rule, end users should notice virtually no difference at all in local applications and those handled on some far-off terminal server or on virtual desktops. For this reason, uberAgent supports widely used solutions to operate terminal servers and virtual desktops. In addition to Microsoft environments, this includes Citrix XenApp and XenDesktop as well as VMware Horizon View. An add-on for the RES Workspace Manager has also been developed.

Licensing is handled on the basis of per named user, per physical or virtual desktop, or per terminal server. This last option could be the best one for robust terminal servers that can serve a large number of users. But if physical computers are to be included in the monitoring process or if users work on virtual desktops and terminal servers in varying work settings, licensing per user becomes the option of choice.

In writing this article, we have assumed the reader is already fairly well acquainted with Splunk, the heart of uberAgent. For those not yet up to speed on the topic, you will find a good introduction in our workshop contained in the February issue that delves into the debugging process with Splunk. The article begins on page 64. In our test, we used version 6.3.3. of Splunk that had already been installed on a device. Our system comprised a server that combines the roles of search heads and indexers.

Two functionalities

uberAgent can be integrated into such an environment using two different approaches. The client component known as Endpoint can be installed on any physical or virtual Windows system. Once installed, uberAgent Endpoint can act on its own and communicate directly with the Splunk server. Or, as an alternative, uberAgent can use a Splunk Universal Forwarder that may already be used by the client to send data to the server.

One benefit of the latter option is that Splunk also collects data that uberAgent would not gather if it were left to its own devices. This applies to all sources that the Splunk Universal Forwarder is already integrating with the Splunk add-on for Windows anyway, including event logs, log files and changes in the file system and registry as well as information about the active directory. Another strength of the Universal Forwarder is that it can monitor the state of the uberAgent when it is teamed with a Splunk app called uberAgent Log Collector.

Simple launch

In our test environment, we had already deployed the Universal Forwarder. For

vast limits uberAgent for Splunk 3.6

The product
Splunk add-on for Windows user experience monitoring.

The vendor
vast limits
https://uberagent.com

The price
Costs at request

System requirements
Server: All operating systems supported by Splunk, Splunk Enterprise 6.2 or higher for uberAgent 3.x, Splunk Free or Enterprise 5.x or 6.x for the older uberAgent 2.x.

Clients: Windows Vista/7/8.x/10 and Windows Server 2008 (R2)/2012 (R2), both 32 bit (if available) and 64 bit.

Technical data
www.it-administrator.de/downloads/datenblaetter

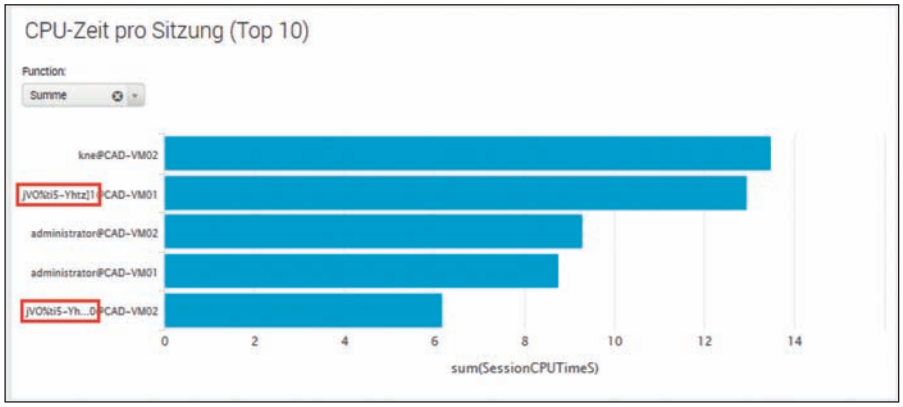


Image 1: For data-protection purposes, uberAgent can turn user names into cryptic identifiers.

this reason, we also selected to use this data-transmission option with uberAgent. Accordingly, we had previously opened the normal port 9997/TCP to receive the Forwarder data on our Splunk server. This meant that the work on the server involved nothing more than installing the necessary packages, which we had downloaded earlier from the uberAgent website and the Splunk base.

We opened "Manage apps / Install app from file." In the process, we added the packages "uberAgent_indexer.tgz" and "uberAgent_searchhead.tgz" that we had previously extracted from the ZIP archives with the uberAgent components. We then relaunched Splunk selecting "Settings / Server controls / Restart Splunk." For good measure, we also added the option "uberAgent Log Collector App" and the related add-on "uberAgent Log Collector SA." Within a few short minutes, our server was ready to receive and evaluate data. In the Splunk Web interface, we discovered our new apps "uberAgent" and "uberAgent Log Collector" on the left-hand menu bar of the home page.

Endpoints installed quickly

The endpoints to be monitored by uberAgent comprised several virtual Windows 7 desktops running under Citrix XenDesktop 7.8 that were equipped with NVIDIA-GRID-K1 and -K2 graphic cards. In addition, we wanted to integrate a few Windows Server 2008 R2 remote desktop session hosts under Citrix XenApp 6.5. One special characteristic of these servers was that they were started through Citrix Provisioning Services from a joint golden image. We therefore started this image as read/write in maintenance mode.

We also installed the Splunk Universal Forwarder on the endpoints where it was not already installed and tested it using event logs that the forwarder sent to our Splunk server. We then opened a local port in the configuration of the forwarder to enable uberAgent to transmit its data. We also added a section in the file "C:\Program Files\SplunkUniversal Forwarder\etc\system\local\inputs.conf" to the endpoints and then restarted the forwarder:

```
[tcp://127.0.0.1:19500]
connection_host = dns
sourcetype = dummy
listenOnIPv6 = no
acceptFrom = 127.0.0.1
```

We also found the setup for the uberAgent Endpoint in the ZIP archive. We launched the installation by using the script ".\uberAgent components\uberAgent_endpoint\bin\manual-install.cmd". As an alternative, you can initiate an unattended installation using central software deployment.

Configuration locally or by Group Policy

In addition to the acceptance of license terms and conditions, the interactive setup only needed the target server and port to be entered. We let the default "localhost:19500" take over here, and the setup was done in no time flat. The agent can then be immediately run without the need to make any further configuration adjustments. Any fine-tuning can be done locally using the text file "C:\Program Files\vast limits\uberAgent\ uberAgent.conf" or centrally by Group Policy.

For this purpose, you will find the template "uberAgent.admx" in the archive of uberAgent components. We copied these components along with the appropriate language file into the SYSVOL share of our domain. We could then create a Group Policy object for uberAgent. The backup of a GPO with sensible basic settings is part of the import template included in the uberAgent package.

All options are located in the branch "Computer Configuration / Policies / Administrative Templates.../ uberAgent." If the global switch "Configuration through Group Policy" is activated here, computers covered by the policy will ignore their local configuration and respond only to the Group Policy.

Anonymization of users possible

Data protection officers and members of company works' councils in Germany will be particularly pleased with the policy "Computer Configuration / Policies / Administrative Templates.../ uberAgent / Miscellaneous / Encrypt user names." This option ensures uberAgent anonymizes user names as it collects data so that no natural persons can be identified during the analysis process. The dashboards will simply display cryptic identifiers in the place of explicit names.

But certain issues must be considered in regard to providing effective support. If a particular user calls and asks for help because his or her session hangs, administrators cannot systematically search for the cause of the problem because the anonymization option is activated.

Subtleties in provisioning

We then copied our test license "uberAgent.lic" to the path "C:\ Program Files\vast limits\uberAgent," restarted the uberAgent service and checked the log "C:\Windows\Temp\uberAgent.log" to reassure ourselves that uberAgent had indeed read the license.

To be able to centrally analyze this log from all endpoints in the future, we extracted the uberAgent log collector technology add-on from the archive "uberagent-log-collector-ta_110.tgz" in the path "C:\Program Files\Splunk UniversalForwarder\etc\apps."

At this point we only had to use four commands to generalize the provisioning server image:

```
net stop SplunkForwarder /yes
"C:\Program Files\SplunkUniversal-
  Forwarder\bin\splunk.exe"
  clone-prep-clear-config
net stop uberAgent /yes
reg delete "HKLM\SOFTWARE\vast
  limits\uberAgent" /f /reg:64
```

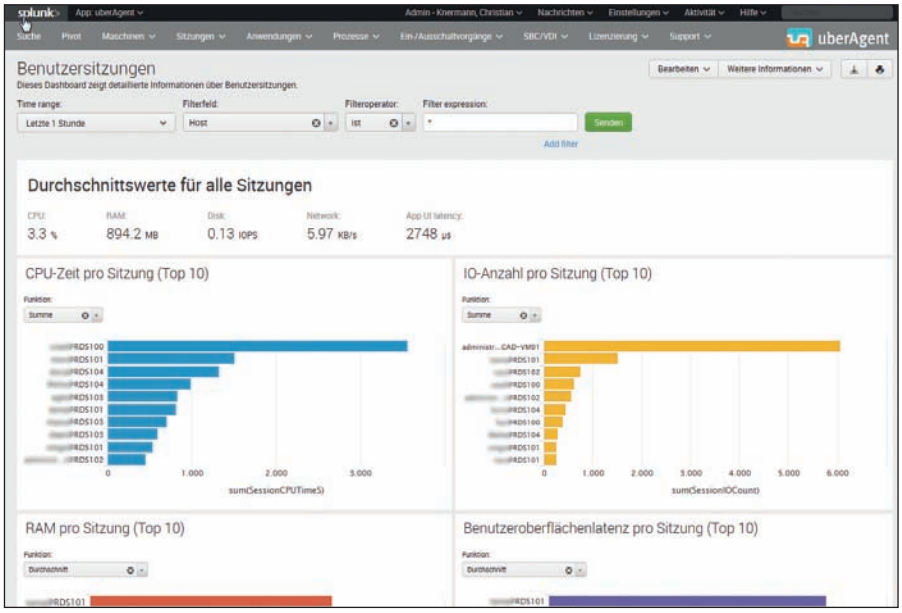


Image 2: uberAgent collects data about the end-user experience and displays this in a clear manner.

As a side note, we would like to point out that these commands are not just limited to Citrix Provisioning Services. They are also designed for every type of cloning or imaging. They should be applied every time a change to the image has to be made before the image is resealed. Having carried out these commands, we were able to use the XenApp Server Role Manager to prepare the image for provisioning, shut it down, put it in "production" mode and boot as many terminal servers as were necessary.

Many dashboards included

We were then able to use the uberAgent Log Collector in the Splunk Web interface to determine that all endpoints had reported back and no serious errors had occurred.

We reached uberAgent through the icon on the home page of the Web interface or directly with the URL "https://splunk-server:8000/en-US/app/uberAgent." uberAgent also comes with dashboards to evaluate the state of a Windows infrastructure on the basis of machines, user sessions and individual applications and processes.

As is typical for Splunk, the dashboards are included in a horizontal bar containing a number of drop-down menus. When launched, the filter focuses on the last hour. It is easy to change the period under review to any relative or absolute point in the past. One or more filter rules can be used to limit a search to such areas as particular hosts or user names.

User sessions clearly displayed

The view "Sessions / User Sessions" is linked as the home page. It provides a global overview of the performance levels of all user sessions in the past hour and displays four bar diagrams that show the average utilization rate of the CPU, RAM, disk I/O and user interface latency in all sessions. This last indicator is uberAgent's very own metric, measuring the time that transpires between keyboard or mouse inputs and the reaction time of the application.

Detailed information about individual sessions is also provided by the data table located at the very bottom of the page. This information includes the start and end times for each session, the log used, the status of the session and various performance indicators. A number of options can be used to measure the last metric, including average, minimum and maximum. Nearly everywhere on the uberAgent dashboards, a click on a detail initiates a drilldown, that is, a more detailed analysis of each selected object.

A dashboard designed in virtually the same way is provided by "Sessions / Session 0." But this view offers no performance data about actual users. Rather, it focuses on "Session 0" on all machines. This involves the session used by the operating system for background services. As a result, this view offers no details about UI latency.

The dashboard "Sessions / User Session Overview" also provides a global overview of the entire environment. Here is where we found graphics showing the absolute number of user sessions over time as well as average logon duration. The latter value is based on a period of time selected for review. If you select a one-week period for review, the system will provide average figures per day. If you choose to evaluate a day, it will provide the average per hour.

Two tables of data displayed at the bottom of the dashboard provide information about simultaneous sessions per host and per user. A click on an individual host or user will initiate a drilldown with the particular focus. Depending on the perspective desired, the drilldown shows the sessions of all users on a host or all sessions of a particular user on all hosts. Two diagrams display network data volume and latency as well as CPU and RAM usage over a period of time. A data table located below these diagrams provides detailed performance data per user. If the host is being examined, another data table will display all of the system's network connections.

Automatic grouping of processes

uberAgent distinguishes between applications and individual processes, although it automatically summarizes these processes in the dashboards according to the application. It thereby draws on information from the MSI packages stored in Windows. In our test, uberAgent determined on its own that the processes "outlook.exe" and "winword.exe" were part of the Microsoft Office Professional Plus 2010. The processes "AcroRd32.exe" and "reader_sl.exe" were correctly grouped to the application of "Adobe Reader XI (11.0.15)." Should individual processes not be assigned as desired, the link can be manually modified by using the local configuration or Group Policy.

A click on the application will initiate another drilldown concerning the performance of this particular application. The respective dashboard lists all processes that belong to the application and displays its performance for all users and hosts. This makes it possible to quickly determine whether the performance levels of a certain instance of the application are within the expected average range or represent an outlier that must be examined more closely. The dashboards in the "Applications" and "Processes" menu provide assistance with follow-up analysis.

Detailed examination of logon processes

Let's turn our attention back to sessions once again. One of uberAgent's real strengths is its ability to not only determine the average logon duration of all sessions, but also to provide granular information about how the time was spent. The key to this is found in the dashboards "Sessions / User Logon Duration" and "Sessions / User Logon Duration - Group Policy."

The first dashboard displays all logons on a time line and uses stacked bars to visualize which phase of the logon process lasts for how long. This information is broken down into specific areas: loading the user profile, processing Group Policy, initializing the shell and running Active Directory and Group Policy logon scripts.

Focus on group policies

The second dashboard really dives deep in terms of Group Policy. uberAgent displays the processing duration of the policies, broken down by individual client side extensions (CSEs). As a result, we could view in detail what share each of our policies on folder redirection, registry, start menu and Internet settings had in the logon processes. In the case of our Citrix infrastructure, the Citrix Group Policy and Citrix profile management were displayed separately.

Both dashboards also include two diagrams showing the top 10 users and hosts. This is actually a negative hit list that displays the users and hosts with the longest logon times. Outliers easily stand out in this information.

At the very bottom, the dashboards provide a data table that lists each particular value for all sessions in the particular period. A drilldown opens the dashboard for individual logons with a focus on the desired session. All performance data previously discussed about the logon process is once again displayed particularly well in tables here.

The information shown here includes which domain controller authenticated the user and how long it took to find this domain controller. Regarding Group Policy, we were informed not only about the processing duration for individual CSEs, but also about which of our Group Policy objects were individually processed. A table also presented a visual image of the performance levels for all processes during the session. This information included their running time and the CPU, RAM and disk I/O load. When more time is required for logons, uberAgent serves as a valuable tool in the effort to track down the cause.

An overview of applications and processes

In a manner similar to the view of sessions, the dashboards evaluate the performance of servers and desktops in the "Machines" area. The dashboard "Machine Performance" includes the usual suspects: CPU, RAM, disk I/O and network. The dashboard "Machine GPU" takes a special look at graphics performance. We used it to check on the usage of our NVIDIA cards. The dashboards in the menus "Applications" and "Processes" showed details about just where the load shown there was coming from.

In addition to general performance levels, uberAgent determined UI latency and waiting time per application. As a result, conclusions can be quickly drawn about whether certain programs are reacting slowly or are getting stuck.

In addition, uberAgent delivers details about start behavior and network communication of applications. It also monitors separate metrics regarding the browser performance of Chrome and Internet Explorer. The related dashboards show which websites consume the most resources. Another dashboard exclusively tracks the loading speed of Outlook plugins and clearly shows which plugins are particularly time consuming during the start process. No dashboard in the area "On/Off Transitions" collects data on terminal servers because servers are designed to run continuously and as a rule are rarely restarted. For physical endpoints and virtual desktops, two dashboards displaying duration and delays in processing have been created for each of the three process steps startup, standby/resume and shutdown.

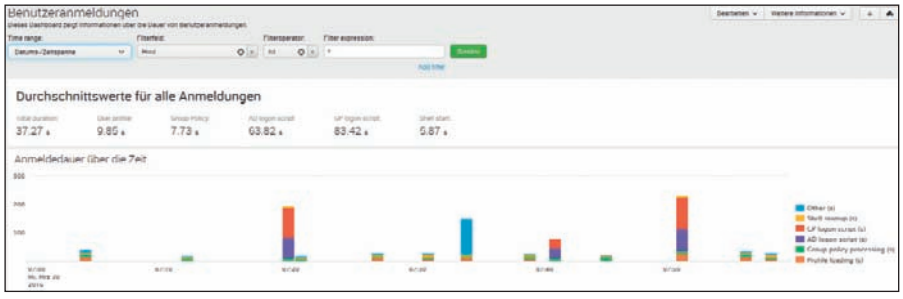


Image 3: Why does it take so long to log onto Windows? uberAgent helps track down the culprit.

Finally, the area "SBC/VDI" displays special metrics regarding sessions in Citrix or VMware environments. For our Citrix sites, we could learn about such things as the client versions being used, their monitor resolution and the most popular released applications.

Summary

uberAgent is a very interesting new extension for the Splunk infrastructure that simplifies debugging and, in general, optimization in the area of terminal server and VDI. The agent can be implemented quickly and supplies detailed information about sessions, applications and computers. The dashboards included in the package present the collected data in a sensible and well-organized manner. Individuals interested in greater detail can expand the dashboards and queries. Through uberAgent, they find a good reason to take a deeper look into Splunk. *(ln)*

Judgement of IT-Administrator

Placed in service	7
Integration into Splunk	7
Metrics for user logons	8
Metrics for applications	7
Integration of Citrix/VMware	8

This product is

ideal for companies that would like to monitor their physical and virtual Windows desktops and are already using Splunk.

of limited interest to organizations that are not yet using Splunk because of the additional costs incurred for further licenses.

not for heterogeneous environments in which Linux is used on desktops