Inhalt

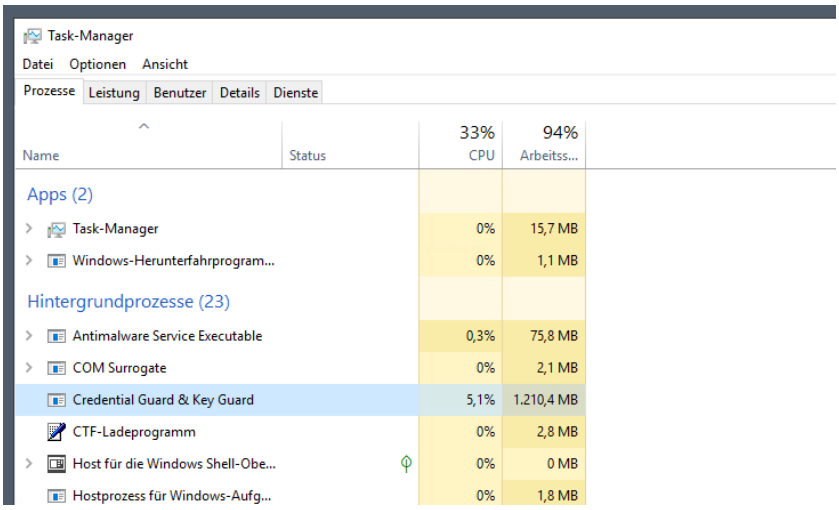
Der Einstieg

Das Problem

Ich verwende in meiner Infrastruktur einen Windows Server 2019 für mein Monitoring. Dort laufen verschiedene Dienste, welche Informationen sammeln und aufbereiten. Bei Bedarf informiert mich das System per Mail oder mit Push-Notifications über Anomalien. Und natürlich kann ich die Informationen auch historisch analysieren, um beispielsweise Trends zu erkennen.

Heute habe ich festgestellt, dass es schon länger keine Warnmeldungen mehr vom Server gab. Das ist durchaus wünschenswert, aber in meinem Fall nicht normal. Denn den Server habe ich so eingestellt, dass er sich mindestens einmal am Tag per Mail meldet und eine Zusammenfassung sendet. Und diese blieb aus.

Das Problem war schnell gefunden: Der Server ist komplett ausgelastet. Im Taskmanager ist der verantwortliche Prozess deutlich erkennbar:



Hier möchte ich aufzeigen, wie man das Problem im Detail auf seine Ursache untersuchen kann.

Ausgangsinformationen

Für jedes TroubleShooting sind zusätzliche Informationen erforderlich.

System-Informationen

- Der Server heißt WS-MON und ist Mitglied meiner Active Directory Domain.
- Er läuft mit Windows Server 2019 und der Desktop Experience.
- Das System ist eine VM in einem Hyper-V-Server.

Welche Dienste laufen auf dem Server?

- Der Server sammelt von allen anderen Servern die weitergeleiteten Ereignisse in einem zentralen Eventlog. Dafür nutze ich das Windows Eventlog Forwarding (**WEF**). Dieses habe ich über eine Gruppenrichtlinie „quell-initiiert“ eingerichtet. Die anderen Server senden also die gefilterten Informationen an meinen WS-MON. Die Last sollte daher recht gering sein.
- Dazu ist ein KIWI-SYSLOG-Server installiert. Hier senden meine PFSense-Systeme (Firewall, IPS) ihre Logfiles her. Der SYSLOG-Server speichert die Daten in Textdateien.
- Auf WS-MON habe ich eine PRTG-Instanz installiert. Diese überwacht mit der freien Edition bis zu 100 Sensoren. In meinem Fall habe ich damit diverse Dienste und Komponenten meiner Infrastruktur agentfrei im Blick.
- Zudem laufen einige Scripte über geplante Aufgaben, mit denen ich Informationen sammle und analysiere.

Was wurde zuvor verändert?

- Das System installiert Windows Updates vollautomatisch.
- PRTG darf sich ebenfalls automatisch Updates herunterladen und installieren.

- Die Belastungen auf dem System sollten sich eigentlich nicht verändern: Weder hat sich das Volumen der Eventlogs noch das der PFSense-Logfiles verändert.

Systemabsicherung

- Der Server wird durch verschiedene Gruppenrichtlinien für die Sicherheit gehärtet. Dazu gehört auch die Absicherung durch den Credential Guard.

Sonstiges

- Das System lief fehlerfrei seit der Installation vor einigen Monaten.

Die Fehlersuche

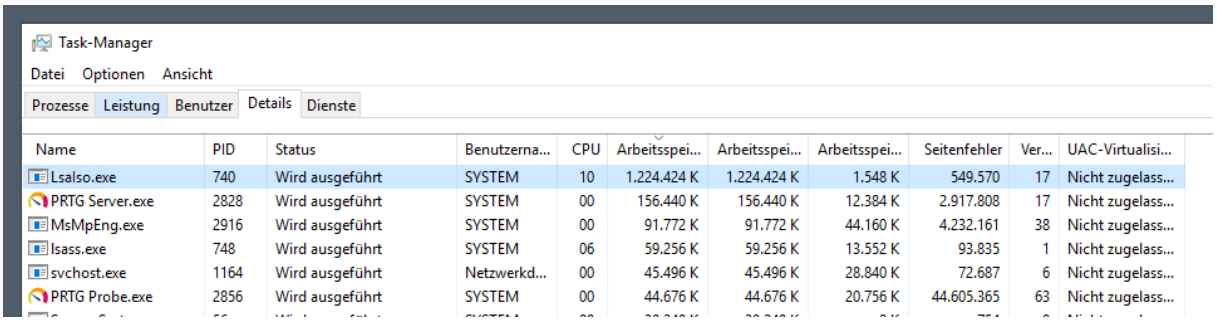
Sichtung von Informationen

Wie geht man nun an die Thematik heran? Zu den Ausgangsinformationen benötige ich Daten des aktuellen Systems. Glücklicherweise konnte ich mich auf dem Server noch anmelden. Alternativ wäre auch ein PowerShell-Remoting denkbar, denn diese Form der Verbindung benötigt nur sehr wenige Ressourcen.

Sollte eine Anmeldung nicht (mehr) möglich sein, dann könnte ein Neustart helfen. Ebenso könnte ich die ausgelastete Ressource (in meinem Fall der Arbeitsspeicher) nach oben skalieren. Aber auch ein Start im guten, alten abgesicherten Modus ist denkbar. Dann bleiben die ganzen Zusatzdienste aus – potentiell also auch die Problem-Komponente.

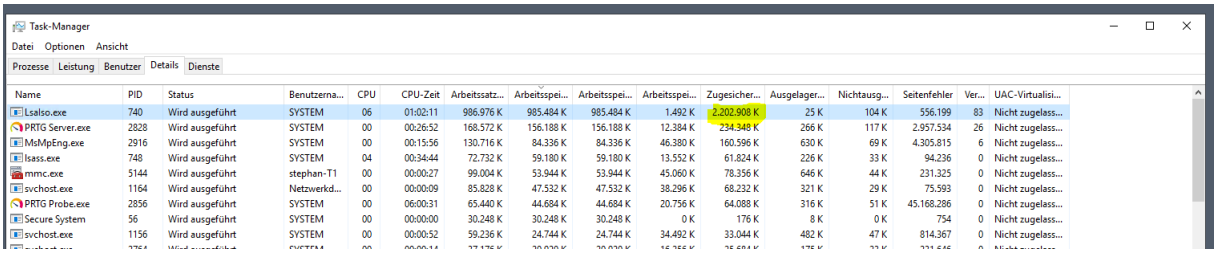
Einen Neustart hatte ich vor einigen Tagen schon durchgeführt. Das Problem konnte damit also nicht gelöst werden. Bei mir starten die Tools gerade noch, daher verzichte ich auf einen weiteren Neustart. Dieser könnte wertvolle Spuren verwischen. Ich erkenne deutlich den Prozess, der den Arbeitsspeicher bindet. LSAISO ist der „Isolated Local Security Authority“-Prozess. In diesem sind zur Laufzeit die „Geheimnisse“ des Betriebssystems – also Passwörter, Hashes und dergleichen – gespeichert. Normalerweise übernimmt diese Aufgabe der LSASS-Prozess. Dieser stellt die eigentliche „Local Security Authority“ dar. Durch den Credential Guard, der seit Windows Server 2016 und Windows 10 verwendet werden kann, werden die sensiblen Informationen noch einmal zusätzlich abgesichert.

Ich arbeite schon sehr lange mit dem Credential Guard. Er ist auch allen von meinen Servern und Clients aktiv. Doch diese Auslastung kenne ich so nicht:



Name	PID	Status	Benutzerna...	CPU	Arbeitsspei...	Arbeitsspei...	Arbeitsspei...	Seitenfehler	Ver...	UAC-Virtualisi...
Lsalso.exe	740	Wird ausgeführt	SYSTEM	10	1.224.424 K	1.224.424 K	1.548 K	549.570	17	Nicht zugelass...
PRTG Server.exe	2828	Wird ausgeführt	SYSTEM	00	156.440 K	156.440 K	12.384 K	2.917.808	17	Nicht zugelass...
MsMpEng.exe	2916	Wird ausgeführt	SYSTEM	00	91.772 K	91.772 K	44.160 K	4.232.161	38	Nicht zugelass...
lsass.exe	748	Wird ausgeführt	SYSTEM	06	59.256 K	59.256 K	13.552 K	93.835	1	Nicht zugelass...
svchost.exe	1164	Wird ausgeführt	Netzwerkd...	00	45.496 K	45.496 K	28.840 K	72.687	6	Nicht zugelass...
PRTG Probe.exe	2856	Wird ausgeführt	SYSTEM	00	44.676 K	44.676 K	20.756 K	44.605.365	63	Nicht zugelass...

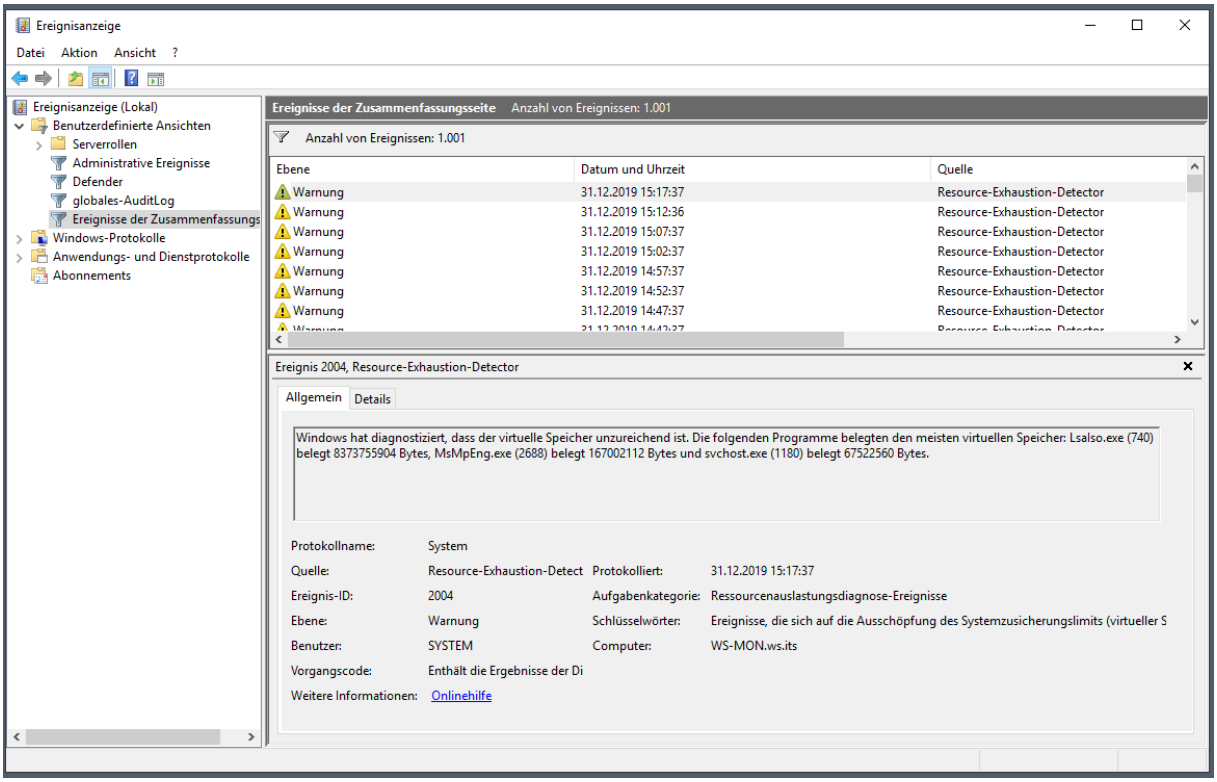
Und wie zu erwarten war, begnügt sich der Prozess nicht mit dem physikalisch zugesicherten Speicher. Er lässt auch großzügig in die Auslagerungsdatei schreiben:

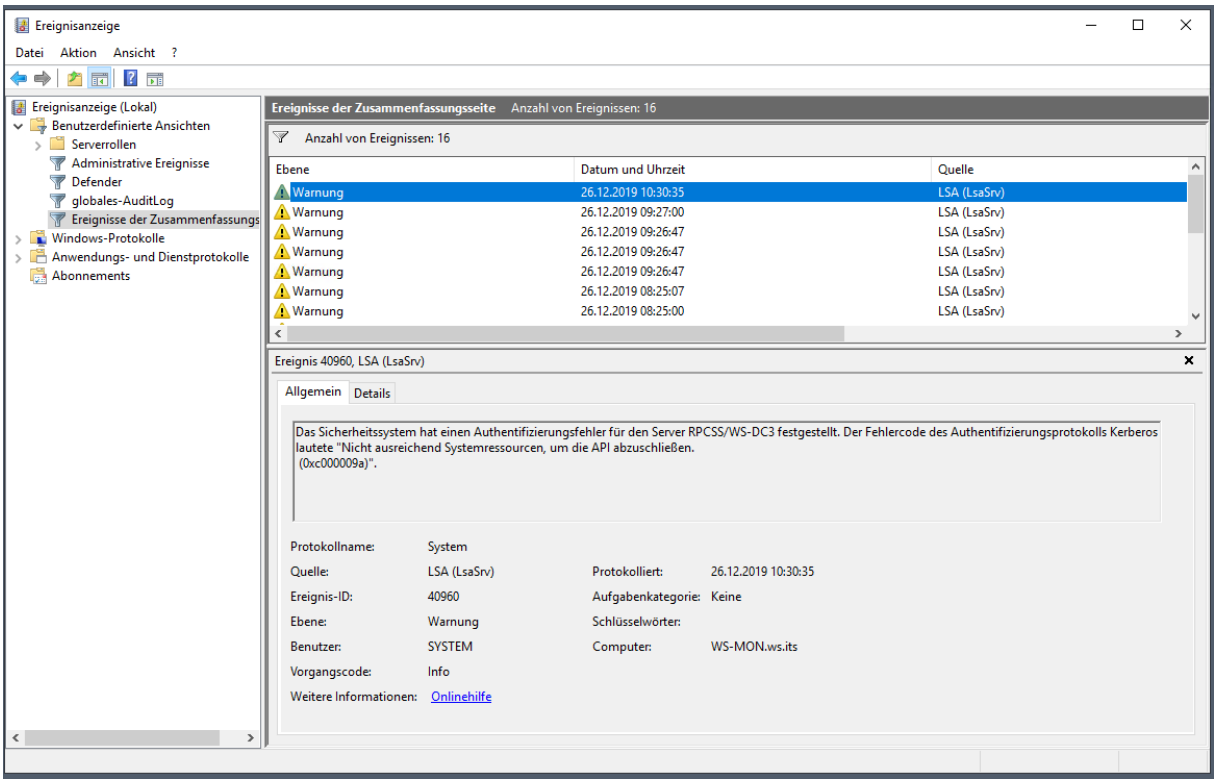


Name	PID	Status	Benutzerna...	CPU	CPU-Zeit	Arbeitssatz...	Arbeitsspei...	Arbeitsspei...	Arbeitsspei...	Zugesicher...	Ausgelager...	Nichtausg...	Seitenfehler	Ver...	UAC-Virtualisi...
Lsalso.exe	740	Wird ausgeführt	SYSTEM	06	01:02:11	986.976 K	985.484 K	985.484 K	1.492 K	2.202.908 K	25 K	104 K	556.199	83	Nicht zugelass...
PRTG Server.exe	2828	Wird ausgeführt	SYSTEM	00	00:26:52	168.572 K	156.188 K	156.188 K	12.384 K	234.348 K	266 K	117 K	2.957.534	26	Nicht zugelass...
MsMpEng.exe	2916	Wird ausgeführt	SYSTEM	00	00:15:56	130.716 K	84.336 K	84.336 K	46.380 K	160.596 K	630 K	69 K	4.305.815	6	Nicht zugelass...
lsass.exe	748	Wird ausgeführt	SYSTEM	04	00:34:44	72.732 K	59.180 K	59.180 K	13.552 K	61.824 K	226 K	33 K	94.236	0	Nicht zugelass...
mmc.exe	5144	Wird ausgeführt	stephan-T1	00	00:00:27	99.004 K	53.944 K	53.944 K	45.060 K	78.356 K	646 K	44 K	231.325	0	Nicht zugelass...
svchost.exe	1164	Wird ausgeführt	Netzwerkd...	00	00:00:09	85.828 K	47.532 K	47.532 K	38.296 K	68.232 K	321 K	29 K	75.593	0	Nicht zugelass...
PRTG Probe.exe	2856	Wird ausgeführt	SYSTEM	00	06:00:31	65.440 K	44.684 K	44.684 K	20.756 K	64.088 K	316 K	51 K	45.168.286	0	Nicht zugelass...
Secure System	56	Wird ausgeführt	SYSTEM	00	00:00:00	30.248 K	30.248 K	30.248 K	0 K	176 K	8 K	0 K	754	0	Nicht zugelass...
svchost.exe	1156	Wird ausgeführt	SYSTEM	00	00:00:52	59.236 K	24.744 K	24.744 K	34.492 K	33.044 K	482 K	47 K	814.367	0	Nicht zugelass...

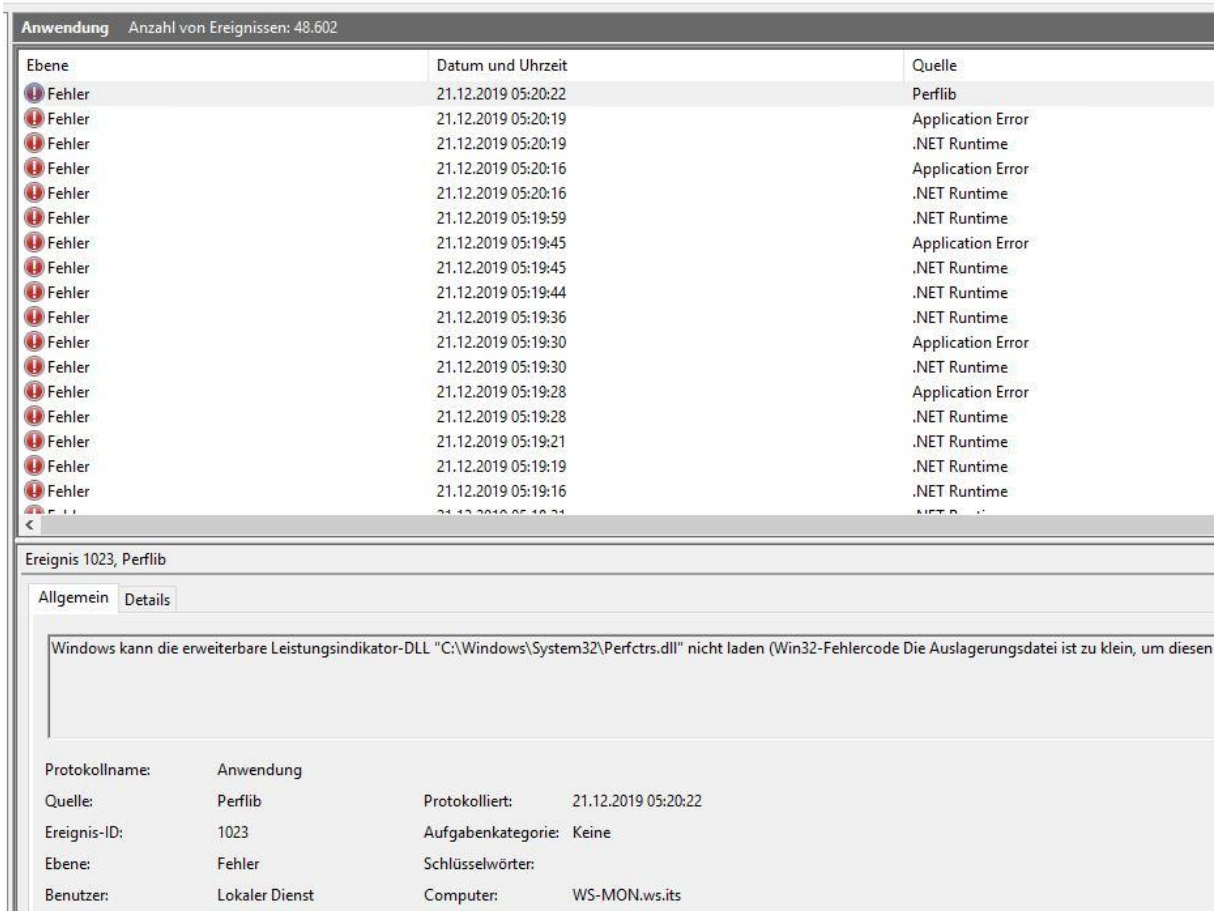
Dabei verdrängt er als systemkritischer Prozess andere Dienste und Anwendungen. In der Folge erhalte ich z.B. keine Informationen mehr vom Monitoring. Denn dieses hat selber keine Ressourcen mehr. Im Eventlog schreit der Server um

Hilfe. Leider wäre für die Weiterleitung dieses Hilferufs mein Monitoring zuständig gewesen, das auf dem betroffenen Server läuft …

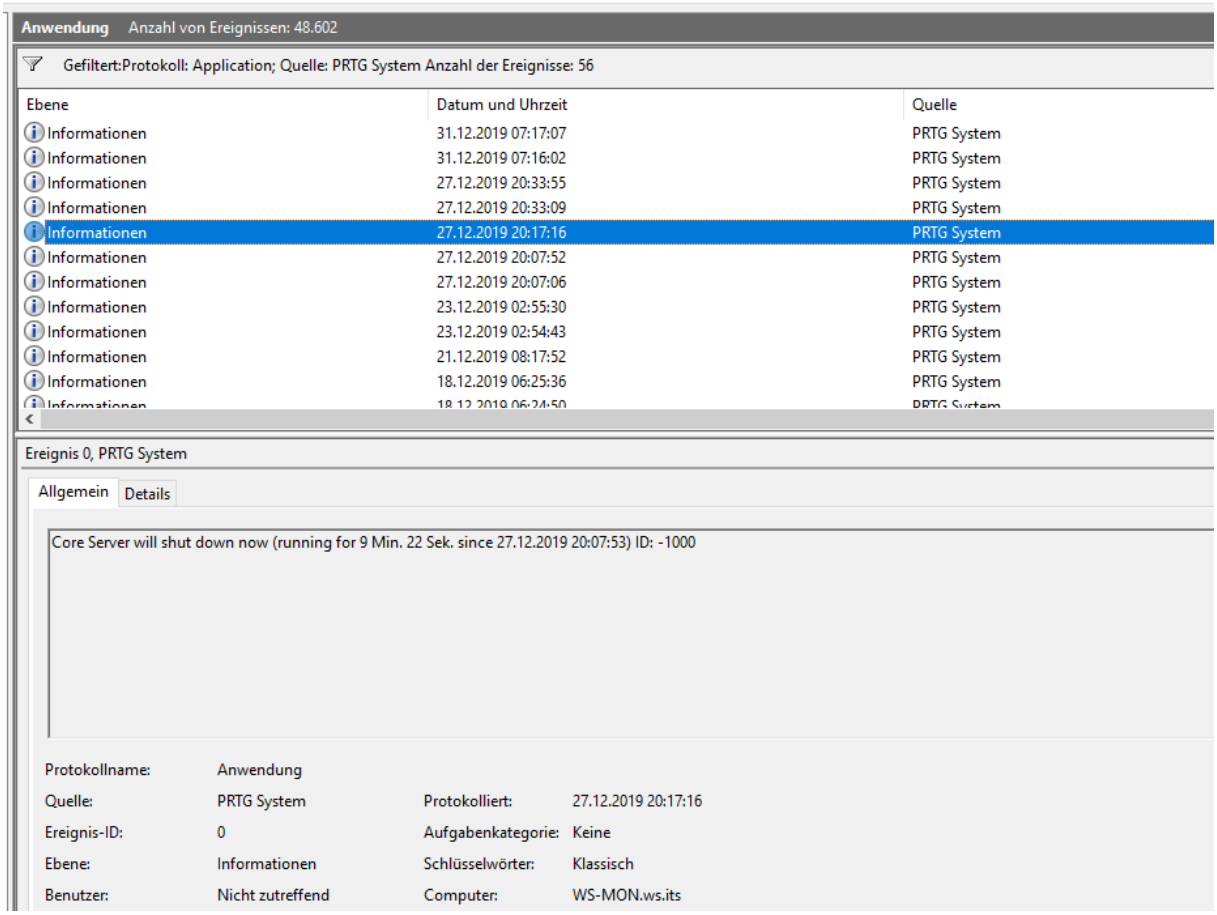




Im Anwendungs-Eventlog finde ich weitere Events, die der Auslastung vorangehen:



Und hier finde ich auch durch eine Filterung die Eventlogs, die mein PRTG-Monitoring kurz vor der Abschaltung schreibt:



TroubleShooting-Methodik

Ich vergleiche gerne das betroffene System mit anderen, um nach Gemeinsamkeiten und Unterschieden zu suchen. Ist ein anderes System nicht betroffen und hat dieses beispielsweise die gleichen Updates installiert, dann liegt die Vermutung nahe, dass es nicht an den Updates liegt. Wobei das keinesfalls ausgeschlossen werden sollte. Vergleicht hier bitte immer artgleiche Server. Einen Domain Controller mit einem SQL-Server zu vergleichen ist sinnbefreit.
In meiner Infrastruktur sind alle anderen Server im Normalbetrieb. Daher schließe ich vorsichtig das Betriebssystem aus.

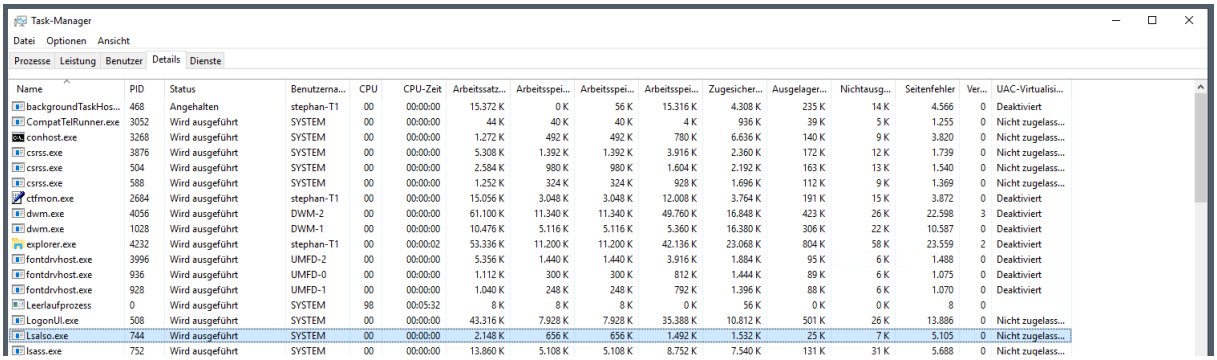
Vielleicht hat sich die Belastung schrittweise erhöht und der Server benötigt einfach mehr Power? Dazu ist eine Trendanalyse sinnvoll. Leider ist diese in meinem Fall im PRTG-Server gespeichert. Und dieser läuft auf dem ausgelasteten Server. Die Information werde ich daher später analysieren. Aus der Erfahrung mit meinem Server kann ich aber bestätigen, dass die konfigurierte Menge an Arbeitsspeicher immer ausgereicht hat. Wäre diese langsam auf dieses Maß gestiegen, dann hätte PRTG einen Hilferuf abgesetzt. Die Belastung muss also recht schnell aufgebaut worden sein.

So fällt mein Verdacht auf die installierten Anwendungen und Dienste. Mit dem Ausschlussverfahren kann ich recht schnell die Ursache eingrenzen.

Ausschlussverfahren: Deaktivierung aller Funktionen des Servers

Für den Beweis meiner These „Es ist eine installierte Anwendung bzw. ein Dienst“ deaktiviere ich den PRTG, den SYSLOG-Server und das Eventlog-Forwarding. Anschließend starte ich den Server neu.

Nach der Anmeldung kontrolliere ich die Systemauslastung. Der LSAISO-Prozess nimmt sich den gewohnten Anteil am Arbeitsspeicher. Die Zahl verändert sich nur im KB-Bereich. Die Ursache ist also eine der installierten Komponenten!

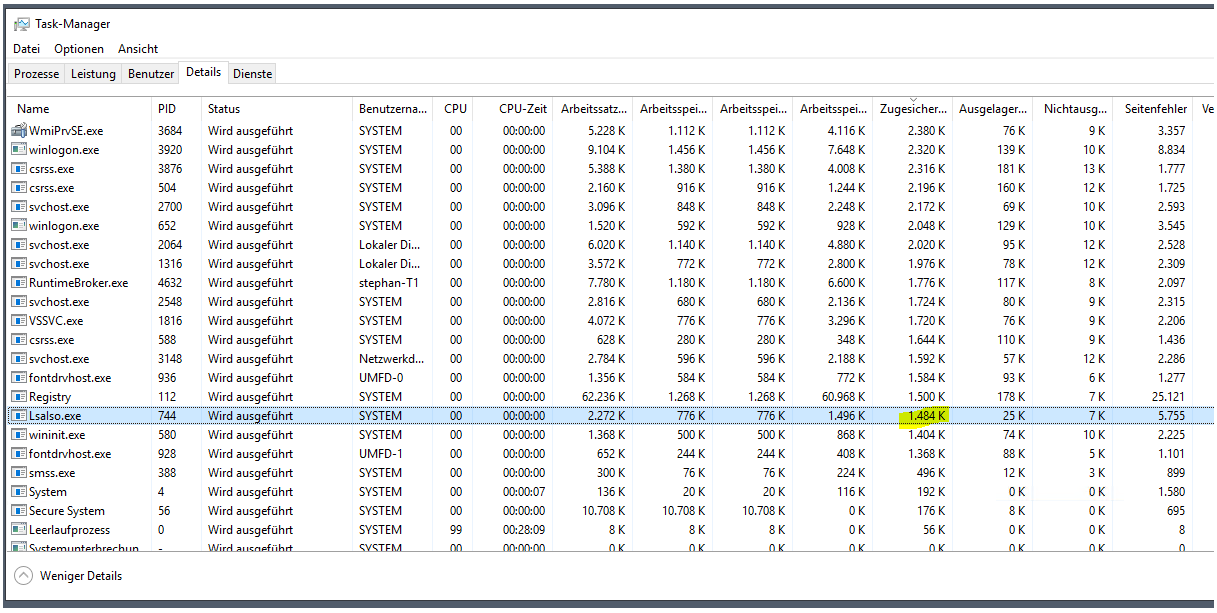


Name	PID	Status	Benutzerna...	CPU	CPU-Zeit	Arbeitssatz...	Arbeitsspei...	Arbeitsspei...	Arbeitsspei...	Zugesicher...	Ausgelager...	Nichtausg...	Seitenfehler	Ver...	UAC-Virtualisi...
backgroundTaskHos...	468	Angehalten	stephan-T1	00	00:00:00	15.372 K	0 K	56 K	15.316 K	4.308 K	235 K	14 K	4.566	0	Deaktiviert
CompatTelRunner.exe	3052	Wird ausgeführt	SYSTEM	00	00:00:00	44 K	40 K	40 K	4 K	936 K	39 K	5 K	1.255	0	Nicht zugelass...
conhost.exe	3268	Wird ausgeführt	SYSTEM	00	00:00:00	1.272 K	492 K	492 K	780 K	6.636 K	140 K	9 K	3.820	0	Nicht zugelass...
csrss.exe	3876	Wird ausgeführt	SYSTEM	00	00:00:00	5.308 K	1.392 K	1.392 K	3.916 K	2.360 K	172 K	12 K	1.739	0	Nicht zugelass...
csrss.exe	504	Wird ausgeführt	SYSTEM	00	00:00:00	2.584 K	980 K	980 K	1.604 K	2.192 K	163 K	13 K	1.540	0	Nicht zugelass...
csrss.exe	588	Wird ausgeführt	SYSTEM	00	00:00:00	1.252 K	324 K	324 K	928 K	1.696 K	112 K	9 K	1.369	0	Nicht zugelass...
ctfmon.exe	2684	Wird ausgeführt	stephan-T1	00	00:00:00	15.056 K	3.048 K	3.048 K	12.008 K	3.764 K	191 K	15 K	3.872	0	Deaktiviert
dwm.exe	4056	Wird ausgeführt	DWM-2	00	00:00:00	61.100 K	11.340 K	11.340 K	49.760 K	16.848 K	423 K	26 K	22.598	3	Deaktiviert
dwm.exe	1028	Wird ausgeführt	DWM-1	00	00:00:00	10.476 K	5.116 K	5.116 K	5.360 K	16.380 K	306 K	22 K	10.587	0	Deaktiviert
explorer.exe	4232	Wird ausgeführt	stephan-T1	00	00:00:02	53.336 K	11.200 K	11.200 K	42.136 K	23.068 K	804 K	58 K	23.559	2	Deaktiviert
fontdrvhost.exe	3996	Wird ausgeführt	UMFD-2	00	00:00:00	5.356 K	1.440 K	1.440 K	3.916 K	1.884 K	95 K	6 K	1.488	0	Deaktiviert
fontdrvhost.exe	936	Wird ausgeführt	UMFD-0	00	00:00:00	1.112 K	300 K	300 K	812 K	1.444 K	89 K	6 K	1.075	0	Deaktiviert
fontdrvhost.exe	928	Wird ausgeführt	UMFD-1	00	00:00:00	1.040 K	248 K	248 K	792 K	1.396 K	88 K	6 K	1.070	0	Deaktiviert
Leerlaufprozess	0	Wird ausgeführt	SYSTEM	98	00:05:32	8 K	8 K	8 K	0 K	56 K	0 K	0 K	8	0	
LogonUI.exe	508	Wird ausgeführt	SYSTEM	00	00:00:00	43.316 K	7.928 K	7.928 K	35.388 K	10.812 K	501 K	26 K	13.886	0	Nicht zugelass...
Lsalso.exe	744	Wird ausgeführt	SYSTEM	00	00:00:00	2.148 K	656 K	656 K	1.492 K	1.532 K	25 K	7 K	5.105	0	Nicht zugelass...
lsass.exe	752	Wird ausgeführt	SYSTEM	00	00:00:00	13.860 K	5.108 K	5.108 K	8.752 K	7.540 K	131 K	31 K	5.688	0	Nicht zugelass...

Für einen echten Beweis und für die Feindiagnose schalte ich nun eine Funktion nach der nächsten wieder ein. Wenn die Belastung mit dem Hochfahren eines Service wieder steigt, dann kenne ich den Verursacher.

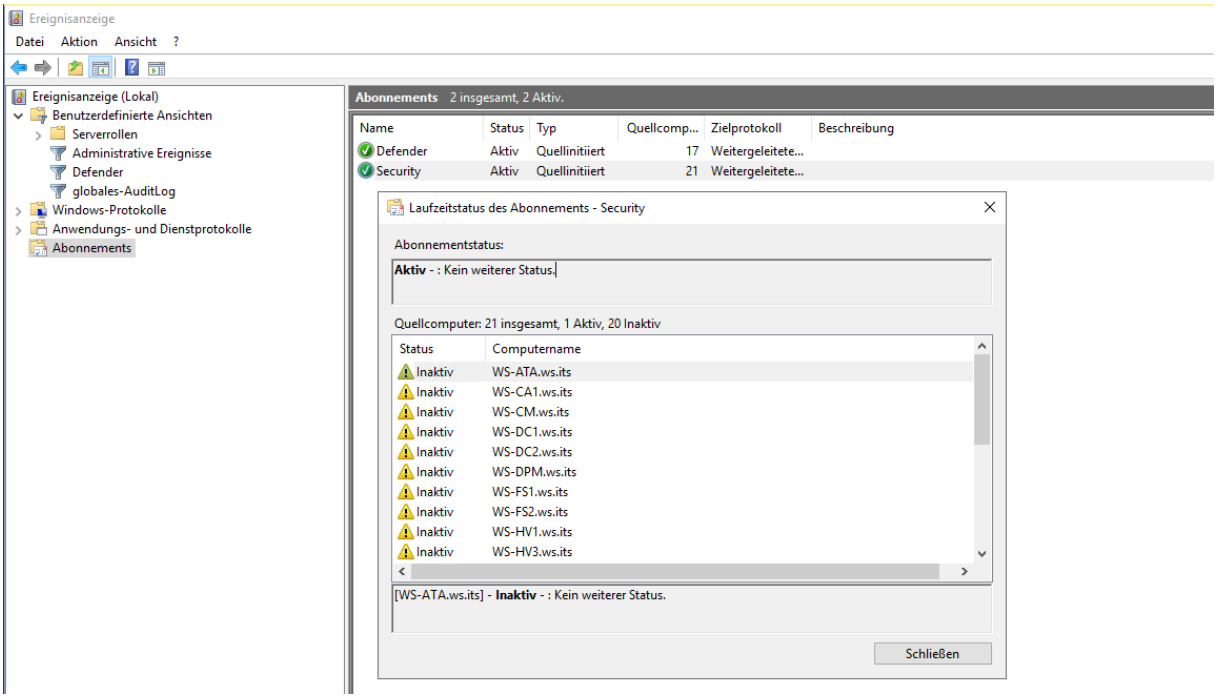
Reaktivierung des SYSLOG-Services und des Eventlog-Forwardings

Ich beginne mit dem SYSLOG-Service. Auch nach mehreren Minuten Betrieb verändert sich die Belastung des LSAISO-Prozesses nicht. Der KIWI-SYSLOG-Service ist es wohl nicht:

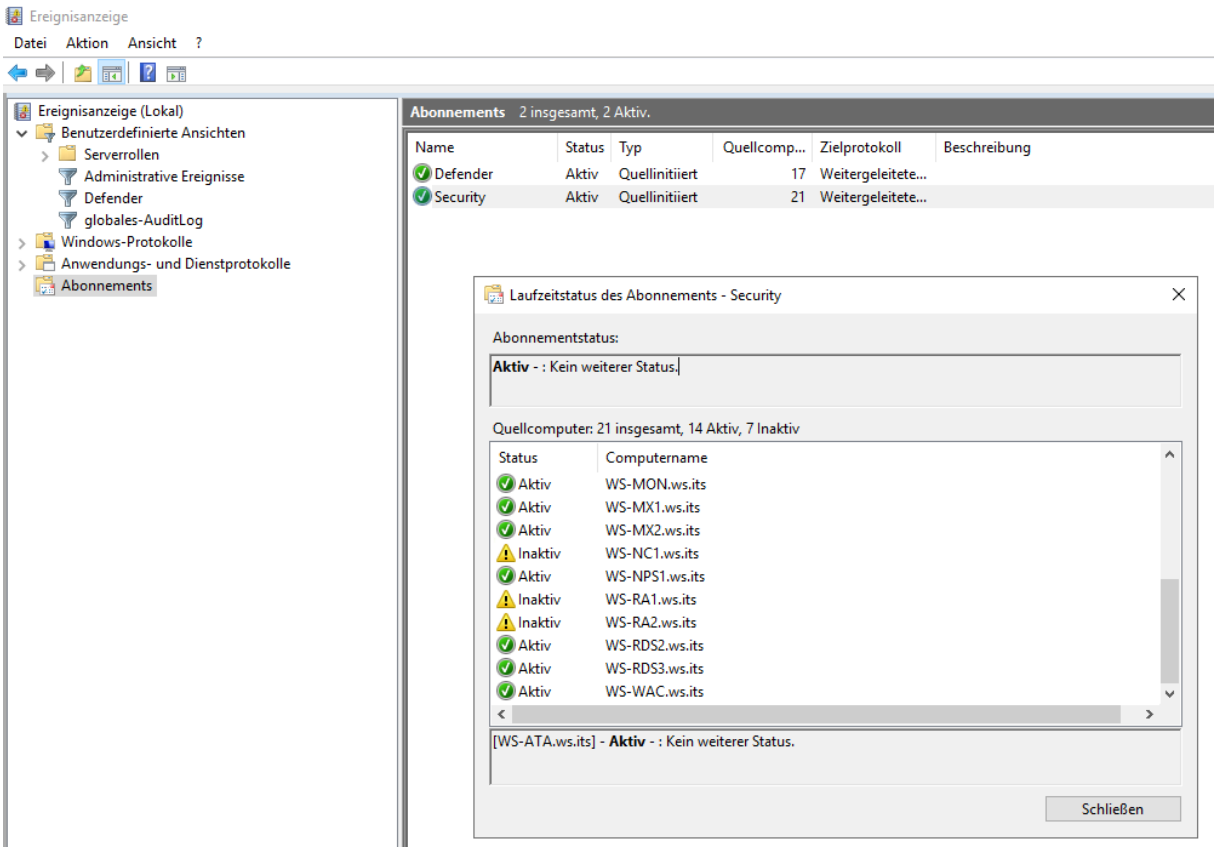


Name	PID	Status	Benutzerna...	CPU	CPU-Zeit	Arbeitssatz...	Arbeitsspei...	Arbeitsspei...	Arbeitsspei...	Zugesicher...	Ausgelager...	Nichtausg...	Seitenfehler	Ve
WmiPrvSE.exe	3684	Wird ausgeführt	SYSTEM	00	00:00:00	5.228 K	1.112 K	1.112 K	4.116 K	2.380 K	76 K	9 K	3.357	
winlogon.exe	3920	Wird ausgeführt	SYSTEM	00	00:00:00	9.104 K	1.456 K	1.456 K	7.648 K	2.320 K	139 K	10 K	8.834	
csrss.exe	3876	Wird ausgeführt	SYSTEM	00	00:00:00	5.388 K	1.380 K	1.380 K	4.008 K	2.316 K	181 K	13 K	1.777	
csrss.exe	504	Wird ausgeführt	SYSTEM	00	00:00:00	2.160 K	916 K	916 K	1.244 K	2.196 K	160 K	12 K	1.725	
svchost.exe	2700	Wird ausgeführt	SYSTEM	00	00:00:00	3.096 K	848 K	848 K	2.248 K	2.172 K	69 K	10 K	2.593	
winlogon.exe	652	Wird ausgeführt	SYSTEM	00	00:00:00	1.520 K	592 K	592 K	928 K	2.048 K	129 K	10 K	3.545	
svchost.exe	2064	Wird ausgeführt	Lokaler Di...	00	00:00:00	6.020 K	1.140 K	1.140 K	4.880 K	2.020 K	95 K	12 K	2.528	
svchost.exe	1316	Wird ausgeführt	Lokaler Di...	00	00:00:00	3.572 K	772 K	772 K	2.800 K	1.976 K	78 K	12 K	2.309	
RuntimeBroker.exe	4632	Wird ausgeführt	stephan-T1	00	00:00:00	7.780 K	1.180 K	1.180 K	6.600 K	1.776 K	117 K	8 K	2.097	
svchost.exe	2548	Wird ausgeführt	SYSTEM	00	00:00:00	2.816 K	680 K	680 K	2.136 K	1.724 K	80 K	9 K	2.315	
VSSVC.exe	1816	Wird ausgeführt	SYSTEM	00	00:00:00	4.072 K	776 K	776 K	3.296 K	1.720 K	76 K	9 K	2.206	
csrss.exe	588	Wird ausgeführt	SYSTEM	00	00:00:00	628 K	280 K	280 K	348 K	1.644 K	110 K	9 K	1.436	
svchost.exe	3148	Wird ausgeführt	Netzwerkd...	00	00:00:00	2.784 K	596 K	596 K	2.188 K	1.592 K	57 K	12 K	2.286	
fontdrvhost.exe	936	Wird ausgeführt	UMFD-0	00	00:00:00	1.356 K	584 K	584 K	772 K	1.584 K	93 K	6 K	1.277	
Registry	112	Wird ausgeführt	SYSTEM	00	00:00:00	62.236 K	1.268 K	1.268 K	60.968 K	1.500 K	178 K	7 K	25.121	
Lsalso.exe	744	Wird ausgeführt	SYSTEM	00	00:00:00	2.272 K	776 K	776 K	1.496 K	1.484 K	25 K	7 K	5.755	
wininit.exe	580	Wird ausgeführt	SYSTEM	00	00:00:00	1.368 K	500 K	500 K	868 K	1.404 K	74 K	10 K	2.225	
fontdrvhost.exe	928	Wird ausgeführt	UMFD-1	00	00:00:00	652 K	244 K	244 K	408 K	1.368 K	88 K	5 K	1.101	
smss.exe	388	Wird ausgeführt	SYSTEM	00	00:00:00	300 K	76 K	76 K	224 K	496 K	12 K	3 K	899	
System	4	Wird ausgeführt	SYSTEM	00	00:00:07	136 K	20 K	20 K	116 K	192 K	0 K	0 K	1.580	
Secure System	56	Wird ausgeführt	SYSTEM	00	00:00:00	10.708 K	10.708 K	10.708 K	0 K	176 K	8 K	0 K	695	
Leerlaufprozess	0	Wird ausgeführt	SYSTEM	99	00:28:09	8 K	8 K	8 K	0 K	56 K	0 K	0 K	8	
Systemunterbrechun...	-	Wird ausgeführt	SYSTEM	00	00:00:00	0 K	0 K	0 K	0 K	0 K	0 K	0 K	0	

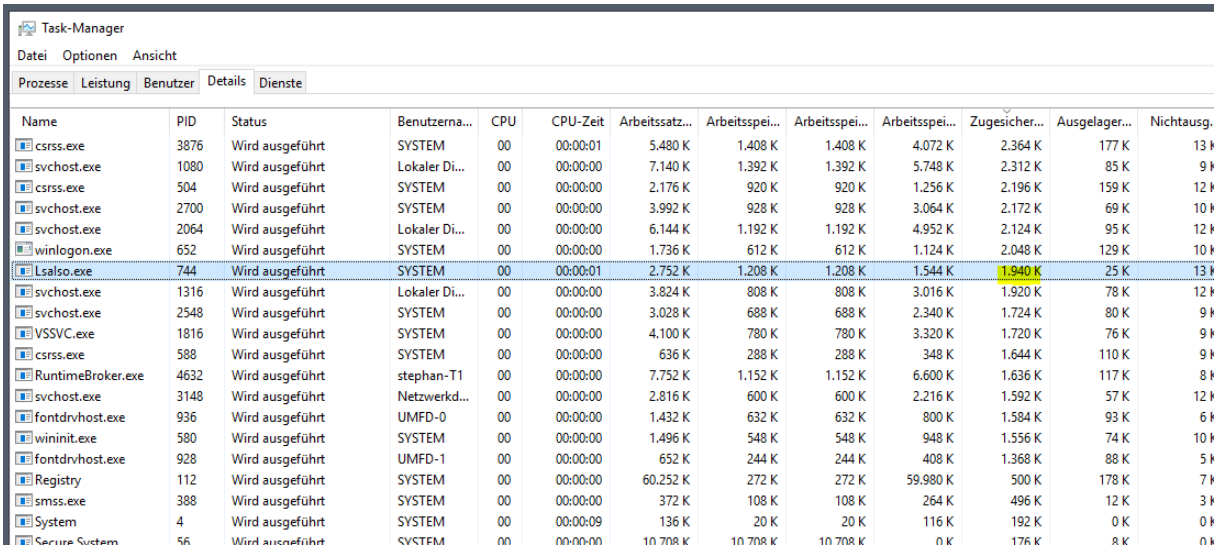
Jetzt aktiviere ich das Eventlog-Forwarding wieder. Die Aktion dauert etwas, da die anderen Server erst verzögert mit der Zustellung der Events beginnen. Daher warte ich einige Minuten ab und kontrolliere den Status der WEF-Clients:



Jetzt sind fast aller Server wieder online:



Und der Problem-Prozess? Der begnügt sich mit seinen Ressourcen:

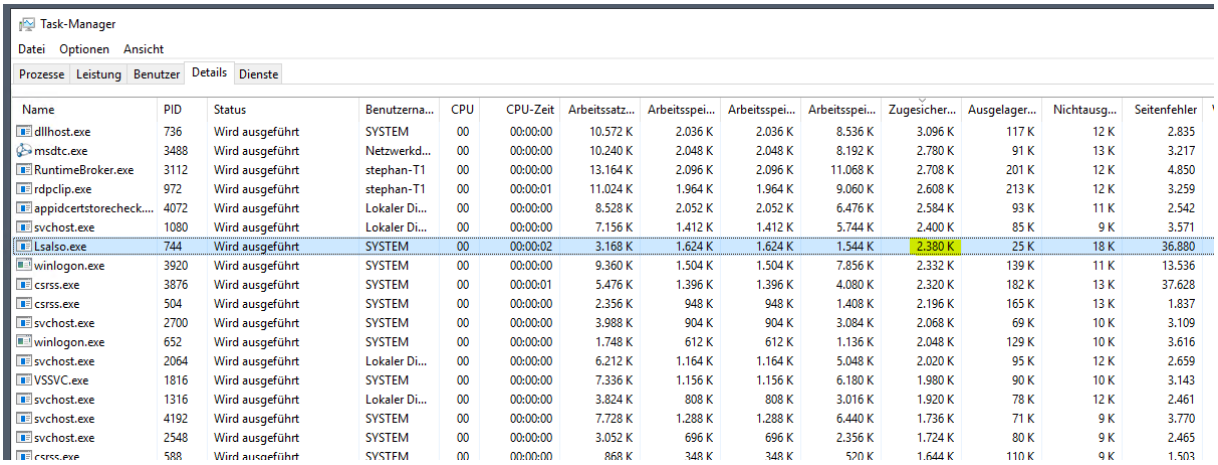


Name	PID	Status	Benutzerna...	CPU	CPU-Zeit	Arbeitssatz...	Arbeitsspei...	Arbeitsspei...	Arbeitsspei...	Zugesicher...	Ausgelager...	Nichtausg.
csrss.exe	3876	Wird ausgeführt	SYSTEM	00	00:00:01	5.480 K	1.408 K	1.408 K	4.072 K	2.364 K	177 K	13 K
svchost.exe	1080	Wird ausgeführt	Lokaler Di...	00	00:00:00	7.140 K	1.392 K	1.392 K	5.748 K	2.312 K	85 K	9 K
csrss.exe	504	Wird ausgeführt	SYSTEM	00	00:00:00	2.176 K	920 K	920 K	1.256 K	2.196 K	159 K	12 K
svchost.exe	2700	Wird ausgeführt	SYSTEM	00	00:00:00	3.992 K	928 K	928 K	3.064 K	2.172 K	69 K	10 K
svchost.exe	2064	Wird ausgeführt	Lokaler Di...	00	00:00:00	6.144 K	1.192 K	1.192 K	4.952 K	2.124 K	95 K	12 K
winlogon.exe	652	Wird ausgeführt	SYSTEM	00	00:00:00	1.736 K	612 K	612 K	1.124 K	2.048 K	129 K	10 K
Lsalso.exe	744	Wird ausgeführt	SYSTEM	00	00:00:01	2.752 K	1.208 K	1.208 K	1.544 K	1.940 K	25 K	13 K
svchost.exe	1316	Wird ausgeführt	Lokaler Di...	00	00:00:00	3.824 K	808 K	808 K	3.016 K	1.920 K	78 K	12 K
svchost.exe	2548	Wird ausgeführt	SYSTEM	00	00:00:00	3.028 K	688 K	688 K	2.340 K	1.724 K	80 K	9 K
VSSVC.exe	1816	Wird ausgeführt	SYSTEM	00	00:00:00	4.100 K	780 K	780 K	3.320 K	1.720 K	76 K	9 K
csrss.exe	588	Wird ausgeführt	SYSTEM	00	00:00:00	636 K	288 K	288 K	348 K	1.644 K	110 K	9 K
RuntimeBroker.exe	4632	Wird ausgeführt	stephan-T1	00	00:00:00	7.752 K	1.152 K	1.152 K	6.600 K	1.636 K	117 K	8 K
svchost.exe	3148	Wird ausgeführt	Netzwerkd...	00	00:00:00	2.816 K	600 K	600 K	2.216 K	1.592 K	57 K	12 K
fontdrvhost.exe	936	Wird ausgeführt	UMFD-0	00	00:00:00	1.432 K	632 K	632 K	800 K	1.584 K	93 K	6 K
wininit.exe	580	Wird ausgeführt	SYSTEM	00	00:00:00	1.496 K	548 K	548 K	948 K	1.556 K	74 K	10 K
fontdrvhost.exe	928	Wird ausgeführt	UMFD-1	00	00:00:00	652 K	244 K	244 K	408 K	1.368 K	88 K	5 K
Registry	112	Wird ausgeführt	SYSTEM	00	00:00:00	60.252 K	272 K	272 K	59.980 K	500 K	178 K	7 K
smss.exe	388	Wird ausgeführt	SYSTEM	00	00:00:00	372 K	108 K	108 K	264 K	496 K	12 K	3 K
System	4	Wird ausgeführt	SYSTEM	00	00:00:09	136 K	20 K	20 K	116 K	192 K	0 K	0 K
Secure System	56	Wird ausgeführt	SYSTEM	00	00:00:00	10.708 K	10.708 K	10.708 K	0 K	176 K	8 K	0 K

Daher kann ich das Eventlog-Forwarding auch ausschließen.

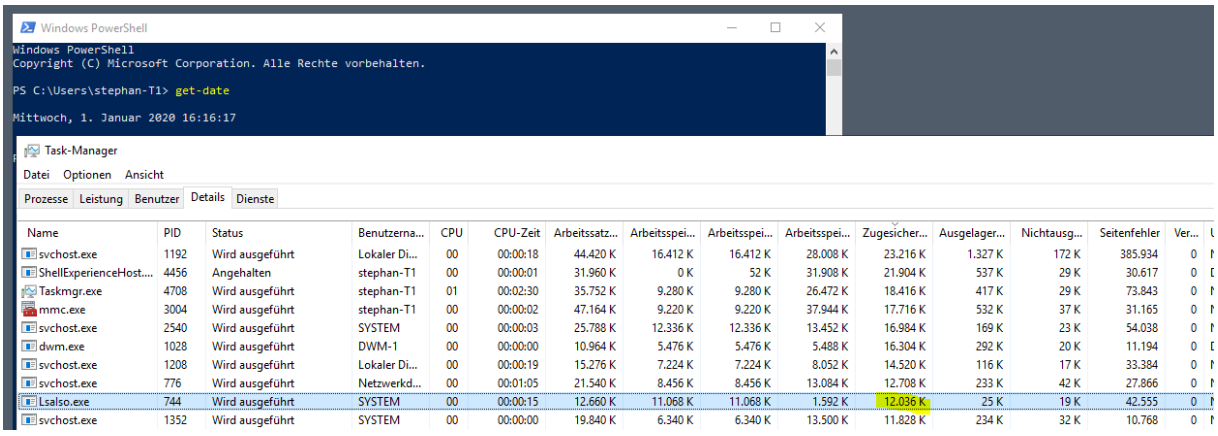
Reaktivierung des PRTG-Services

Es bleibt der Monitoring-Service PRTG über. Dieser besteht aus 2 Diensten. Zuerst starte ich den Core-Service. Dabei beobachte ich wieder die Auslastung des Prozesses LSAISO. Der Wert bleibt nahezu statisch.



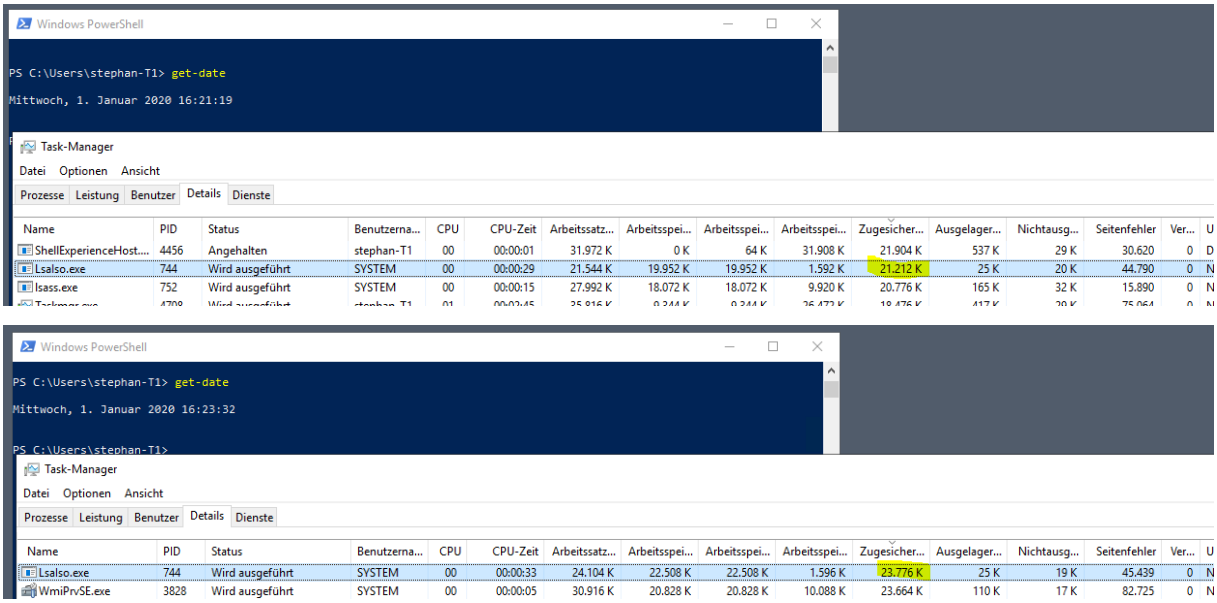
Name	PID	Status	Benutzerna...	CPU	CPU-Zeit	Arbeitssatz...	Arbeitsspei...	Arbeitsspei...	Arbeitsspei...	Zugesicher...	Ausgelager...	Nichtausg...	Seitenfehler
dllhost.exe	736	Wird ausgeführt	SYSTEM	00	00:00:00	10.572 K	2.036 K	2.036 K	8.536 K	3.096 K	117 K	12 K	2.835
msdtc.exe	3488	Wird ausgeführt	Netzwerkd...	00	00:00:00	10.240 K	2.048 K	2.048 K	8.192 K	2.780 K	91 K	13 K	3.217
RuntimeBroker.exe	3112	Wird ausgeführt	stephan-T1	00	00:00:00	13.164 K	2.096 K	2.096 K	11.068 K	2.708 K	201 K	12 K	4.850
rdpclip.exe	972	Wird ausgeführt	stephan-T1	00	00:00:01	11.024 K	1.964 K	1.964 K	9.060 K	2.608 K	213 K	12 K	3.259
appidcertstorecheck....	4072	Wird ausgeführt	Lokaler Di...	00	00:00:00	8.528 K	2.052 K	2.052 K	6.476 K	2.584 K	93 K	11 K	2.542
svchost.exe	1080	Wird ausgeführt	Lokaler Di...	00	00:00:00	7.156 K	1.412 K	1.412 K	5.744 K	2.400 K	85 K	9 K	3.571
Lsalso.exe	744	Wird ausgeführt	SYSTEM	00	00:00:02	3.168 K	1.624 K	1.624 K	1.544 K	2.380 K	25 K	18 K	36.880
winlogon.exe	3920	Wird ausgeführt	SYSTEM	00	00:00:00	9.360 K	1.504 K	1.504 K	7.856 K	2.332 K	139 K	11 K	13.536
csrss.exe	3876	Wird ausgeführt	SYSTEM	00	00:00:01	5.476 K	1.396 K	1.396 K	4.080 K	2.320 K	182 K	13 K	37.628
csrss.exe	504	Wird ausgeführt	SYSTEM	00	00:00:00	2.356 K	948 K	948 K	1.408 K	2.196 K	165 K	13 K	1.837
svchost.exe	2700	Wird ausgeführt	SYSTEM	00	00:00:00	3.988 K	904 K	904 K	3.084 K	2.068 K	69 K	10 K	3.109
winlogon.exe	652	Wird ausgeführt	SYSTEM	00	00:00:00	1.748 K	612 K	612 K	1.136 K	2.048 K	129 K	10 K	3.616
svchost.exe	2064	Wird ausgeführt	Lokaler Di...	00	00:00:00	6.212 K	1.164 K	1.164 K	5.048 K	2.020 K	95 K	12 K	2.659
VSSVC.exe	1816	Wird ausgeführt	SYSTEM	00	00:00:00	7.336 K	1.156 K	1.156 K	6.180 K	1.980 K	90 K	10 K	3.143
svchost.exe	1316	Wird ausgeführt	Lokaler Di...	00	00:00:00	3.824 K	808 K	808 K	3.016 K	1.920 K	78 K	12 K	2.461
svchost.exe	4192	Wird ausgeführt	SYSTEM	00	00:00:00	7.728 K	1.288 K	1.288 K	6.440 K	1.736 K	71 K	9 K	3.770
svchost.exe	2548	Wird ausgeführt	SYSTEM	00	00:00:00	3.052 K	696 K	696 K	2.356 K	1.724 K	80 K	9 K	2.465
csrss.exe	588	Wird ausgeführt	SYSTEM	00	00:00:00	868 K	348 K	348 K	520 K	1.644 K	110 K	9 K	1.503

Zuletzt starte ich den zweiten Service „PRTG-Probes“. Jetzt belegt der LSAISO-Prozess mehr Speicher. Ich merke mir dazu die aktuelle Uhrzeit:

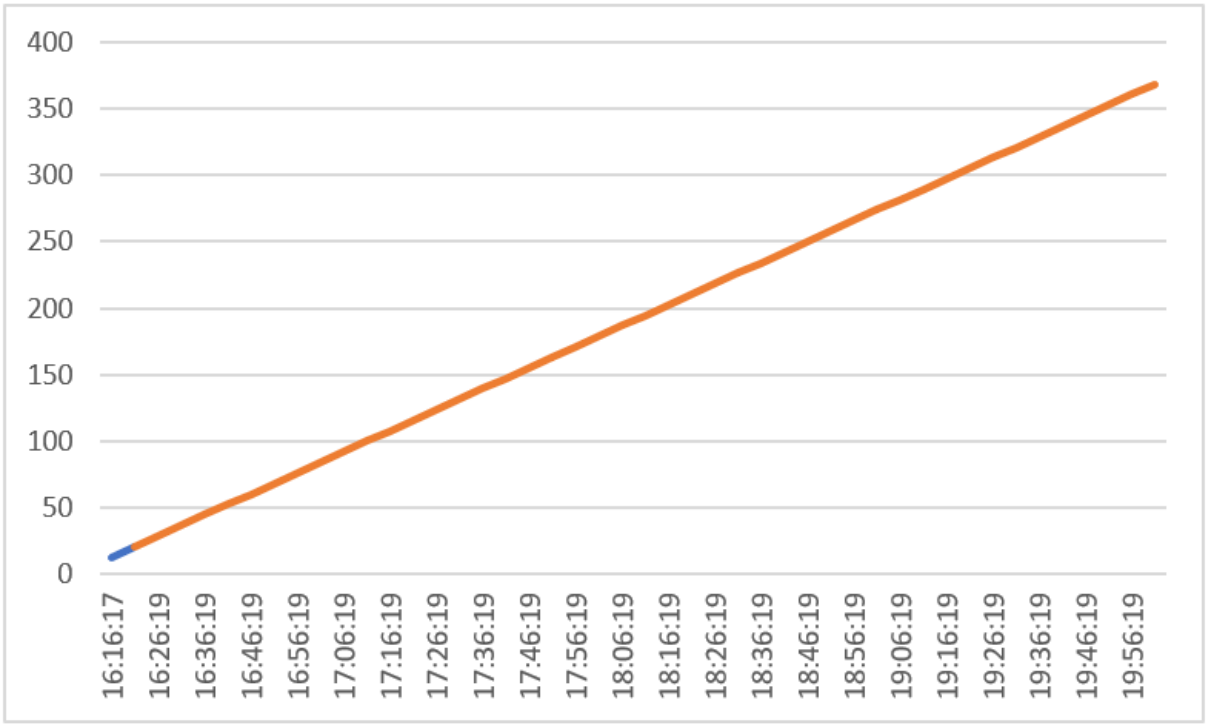


Name	PID	Status	Benutzerna...	CPU	CPU-Zeit	Arbeitssatz...	Arbeitsspei...	Arbeitsspei...	Arbeitsspei...	Zugesicher...	Ausgelager...	Nichtausg...	Seitenfehler	Ver...
svchost.exe	1192	Wird ausgeführt	Lokaler Di...	00	00:00:18	44.420 K	16.412 K	16.412 K	28.008 K	23.216 K	1.327 K	172 K	385.934	0
ShellExperienceHost....	4456	Angehalten	stephan-T1	00	00:00:01	31.960 K	0 K	52 K	31.908 K	21.904 K	537 K	29 K	30.617	0
Taskmgr.exe	4708	Wird ausgeführt	stephan-T1	01	00:02:30	35.752 K	9.280 K	9.280 K	26.472 K	18.416 K	417 K	29 K	73.843	0
mmc.exe	3004	Wird ausgeführt	stephan-T1	00	00:00:02	47.164 K	9.220 K	9.220 K	37.944 K	17.716 K	532 K	37 K	31.165	0
svchost.exe	2540	Wird ausgeführt	SYSTEM	00	00:00:03	25.788 K	12.336 K	12.336 K	13.452 K	16.984 K	169 K	23 K	54.038	0
dwm.exe	1028	Wird ausgeführt	DWM-1	00	00:00:00	10.964 K	5.476 K	5.476 K	5.488 K	16.304 K	292 K	20 K	11.194	0
svchost.exe	1208	Wird ausgeführt	Lokaler Di...	00	00:00:19	15.276 K	7.224 K	7.224 K	8.052 K	14.520 K	116 K	17 K	33.384	0
svchost.exe	776	Wird ausgeführt	Netzwerkd...	00	00:01:05	21.540 K	8.456 K	8.456 K	13.084 K	12.708 K	233 K	42 K	27.866	0
Lsalso.exe	744	Wird ausgeführt	SYSTEM	00	00:00:15	12.660 K	11.068 K	11.068 K	1.592 K	12.036 K	25 K	19 K	42.555	0
svchost.exe	1352	Wird ausgeführt	SYSTEM	00	00:00:00	19.840 K	6.340 K	6.340 K	13.500 K	11.828 K	234 K	32 K	10.768	0

Nach 5 Minuten sind weitere 9 MB allokiert:



Das klingt nach nichts. Aber der Server soll 24/7 laufen. Und rein rechnerisch sind das unter der Annahme der linearen Steigerung 1,6MB je Minute. Das würde dann von jetzt bis 20:00 so aussehen:



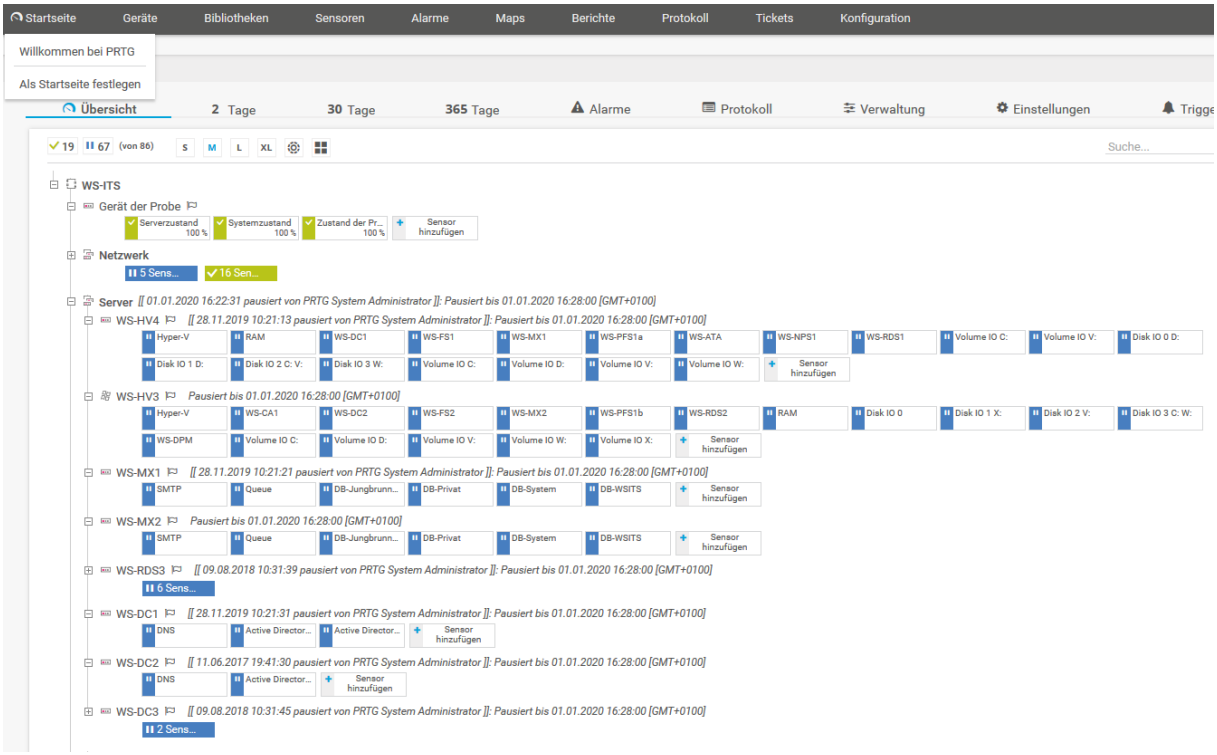
In 24 Stunden wären mehr als 2GB Arbeitsspeicher belegt…

Detailsuche im Service PRTG

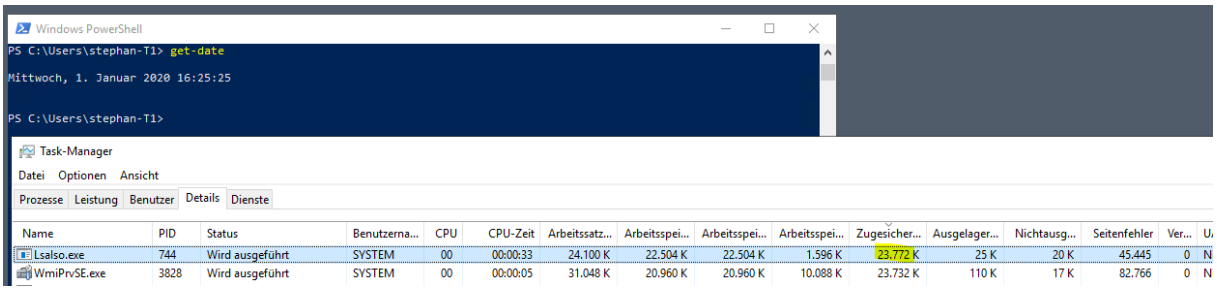
Jetzt kenne ich die Komponenten, die das Problem verursachen: ein PRTG-Service auf einem Windows Server 2019 mit aktivem Credential Guard.

Ich benötige aber für die Fehlerbehebung weitere Informationen. Daher prüfe ich nun verschiedene Einstellungen im PRTG und beobachte deren Auswirkungen auf den LSAISO-Prozess.

Mein PRTG überwacht neben den Windows-Servern auch andere Geräte und Dienste. Ich habe einen Verdach, dass mein Problem durch die Windows Sensoren verursacht wird. Daher pausiere ich alle Sensoren, die für meine Windows Services gedacht sind:



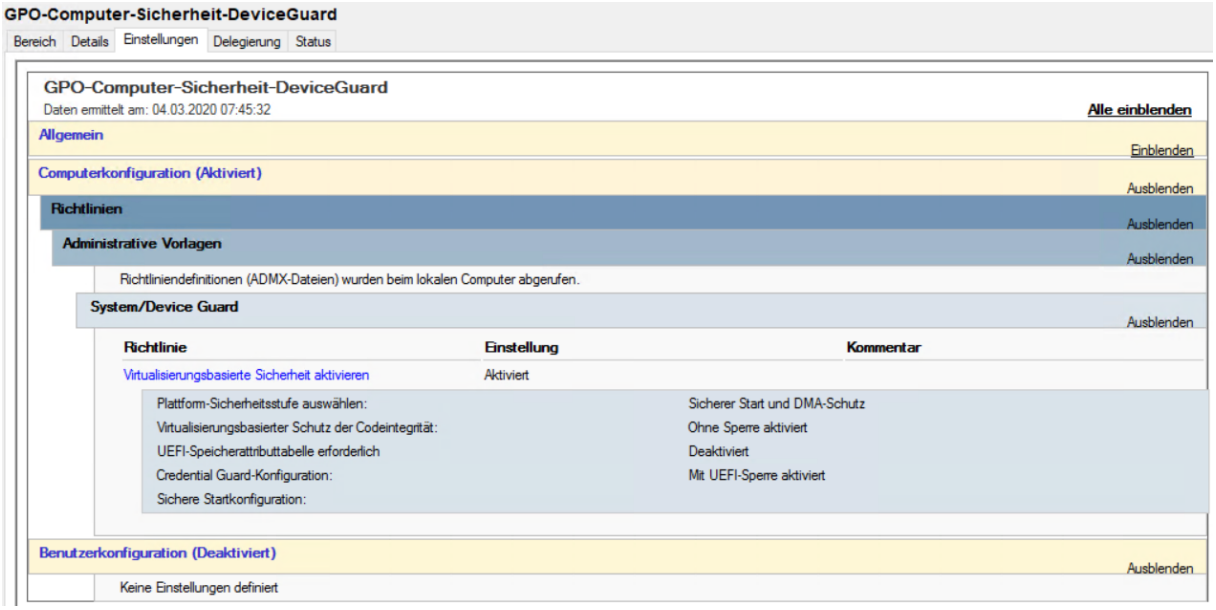
Interessant: Seit der Pausierung stagniert der Hunger nach mehr Arbeitsspeicher vom LSAISO-Prozess wieder. Die Vermutung scheint richtig zu sein:



Ich schalte einzelne Sensoren wieder ein. Mit jedem aktiven Windows-Sensor steigt der Arbeitsspeicherkonsum leicht an. Da habe ich mit meiner freien PRTG-Lizenz und max. 100 Sensoren durchaus Glück gehabt.

Vielleicht liegt es an der Kennung, die mein PRTG für den Remotezugriff verwendet? Diese ist ein Active Directory Benutzer, den ich als Service Account verwende. Ich konfiguriere einen anderen Benutzer-Account in den Sensoren. Durch die Vererbung der Einstellungen geht das recht schnell. Dann starte ich die Sensoren wieder. Leider nimmt sich der Credential Guard wieder mehr Speicher. Das Problem wird nicht durch den Service Account verursacht.

Vielleicht ist es gar nicht die PRTG-Konfiguration? Ich probiere die Deaktivierung des Device Guards. Dieser stellt auch den Credential Guard bereit. Die Aktivierung dieser Schutzkomponente übernimmt eine Gruppenrichtlinie:



Ich nehme den Server aus deren Wirkungsbereich heraus, aktualisiere die Gruppenrichtlinienverarbeitung durch ein gpupdate und starte ihn neu. Da ich die Einstellung mit UEFI-Sperre aktiviert habe, wird das aber nicht genügen. Dazu ist etwas Hintergrundwissen hilfreich:

> Der Device Guard soll das System auch vor Schadprogrammen schützen, die selber im Systemkontext laufen. Der LSAISO-Prozess kann im laufenden Betrieb nicht beendet werden. Zur Laufzeit ist das System also sicher. Aber ein Angreifer könnte die Einstellung für den Start des Device Guards verändern und das System neu starten. Dann wäre der Prozess nach dem Neustart aus und die Geheimnisse des Betriebssystems könnten abgegriffen werden.
>
> Und genau hier greift die UEFI-Sperre. Die GPO erzeugt im UEFI eine Variable für den Device Guard Startmodus. Diese Variable wird auf ReadOnly konfiguriert. Sie kann also auch vom System selber nicht mehr zur Laufzeit verändert werden. Bei jedem Neustart prüft der LSASS-Prozess den Inhalt der Variable. Ist sie vorhanden und auf einen aktiven Device Guard konfiguriert, dann wird der LSAISO-Prozess gestartet – unabhängig davon, was die lokale Registry dazu sagt. Damit bleibt ein Device Guard auch nach dem Neustart aktiv – egal, welche Rechte ein Angreifer erbeutet hat.

Dieses hohe Schutzniveau behindert aber nicht nur Angreifer. Ich als Administrator muss jetzt zusätzliche Schritte unternehmen, um den Device Guard zu deaktivieren. Das einfache Entfernen der GPO wird nicht genügen, denn darin wird ja auch nur der Registry-Key verändert. Die UEFI-Variable kann das System ja selber nicht mehr verändern.

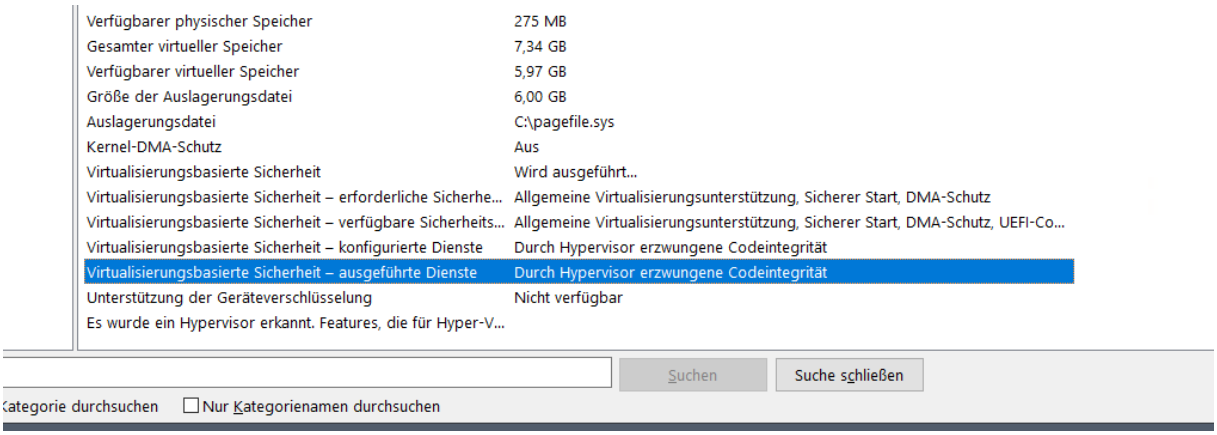
Für die Deaktivierung gibt es eine offizielle Anleitung von Microsoft. Dabei wird für den nächsten Neustart der UEFI-Boot um eine Sequenz erweitert. In dieser muss dann vor dem Start des Betriebssystems an der Konsole des Servers die Deaktivierung vom Device Guard explizit bestätigt werden. Man benötigt also einen physikalischen Zugang.

Für die Abschaltung stellt Microsoft eine efi-Datei bereit. Diese habe ich in der passenden x64-Variante direkt auf die C-Partition gespeichert. In meinem Server führe ich diese Zeilen in einer administrativen cmd aus:

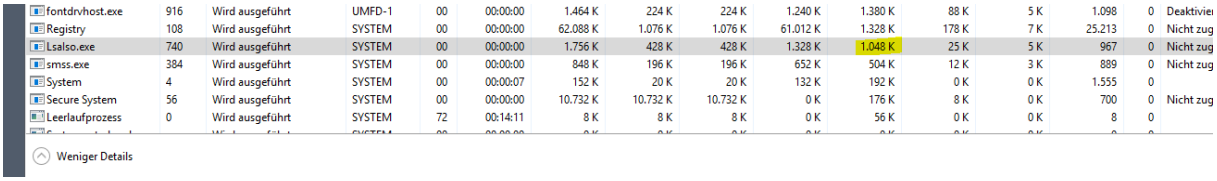
```
mountvol X: /s
copy C:\LSAPPLConfig.efi X:\EFI\Microsoft\Boot\LSAPPLConfig.efi /Y
bcdedit /create {0cb3b571-2f2e-4343-a879-d86a476d7215} /d "DebugTool" /application osloader
bcdedit /set {0cb3b571-2f2e-4343-a879-d86a476d7215} path "\EFI\Microsoft\Boot\LSAPPLConfig.efi"
bcdedit /set {bootmgr} bootsequence {0cb3b571-2f2e-4343-a879-d86a476d7215}
bcdedit /set {0cb3b571-2f2e-4343-a879-d86a476d7215} loadoptions %1
bcdedit /set {0cb3b571-2f2e-4343-a879-d86a476d7215} device partition=X:
mountvol X: /d

shutdown -r -t 0
```

Der Server wird jetzt neugestartet. Auf der Konsole im Hyper-V stehen nun wenige Sekunden für die Abschaltung zur Verfügung:

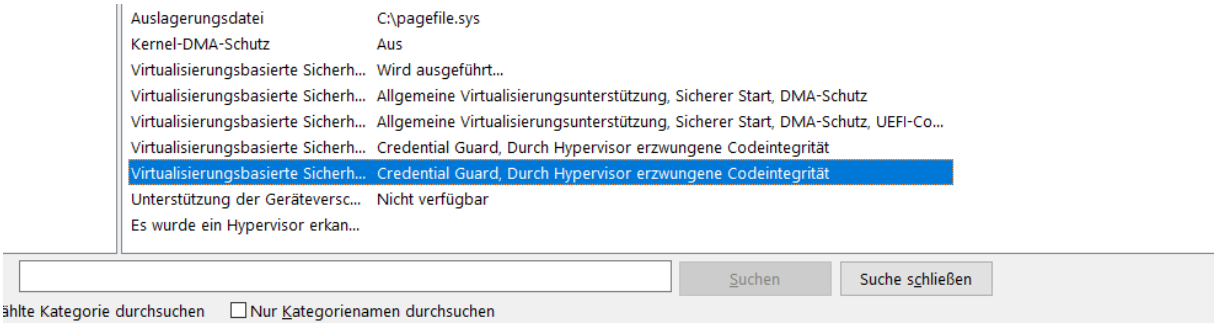
Anschließend startet der Server wie gewohnt. Nur eben ohne den LSAISO-Prozess des Device Guards. Mit einem Blick ins msinfo kann ich die Abschaltung überprüfen:



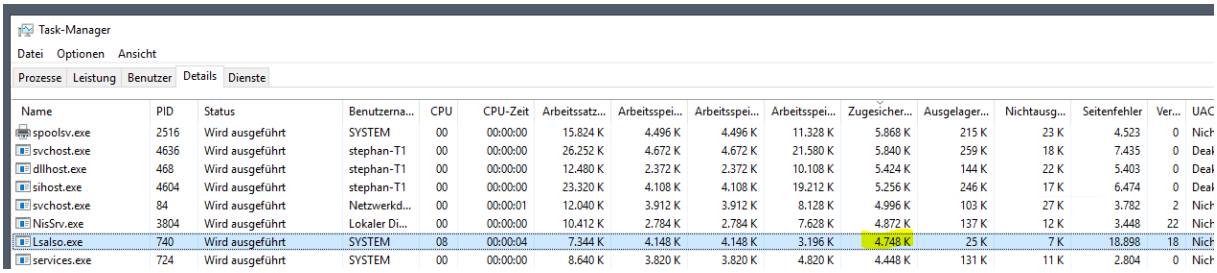
Die PRTG-Services starten automatisch. Den LSAISO-Prozess gibt es weiter, denn dieser wird auch für das CodeIntegrity-Enforcement benötigt. Aber er begnügt sich mit wenig Arbeitsspeicher:



Für einen echten Beweis verschiebe ich den Server wieder in den Einflussbereich der DeviceGuard GPO. Ein gpupdate und einen Neustart ist er wieder online. Msinfo zeigt die Präsens:



Und der Taskmanager zeigt den LSAISO-Prozess mit seinem Memory Leak…



Der Beweis ist erbracht: Der Credential Guard des Device Guards kann nicht mit dem PRTG Probe Service umgehen, wenn im PRTG Windows Server remote überwacht werden.

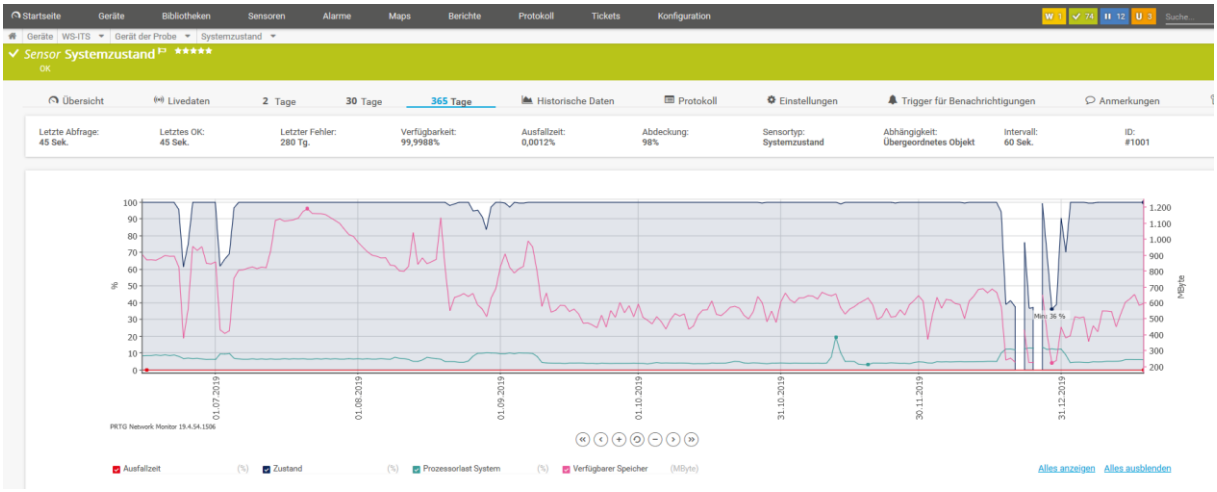
Der Auslöser

Die Ursache für das direkte Problem ist identifiziert. Aber der Auslöser ist noch nicht bekannt. Den Server betreibe ich seit Monaten problemfrei.

Für die Suche nach dem Auslöser gibt es verschiedene Strategien, die mit unterschiedlichen Fragen verfolgt werden:

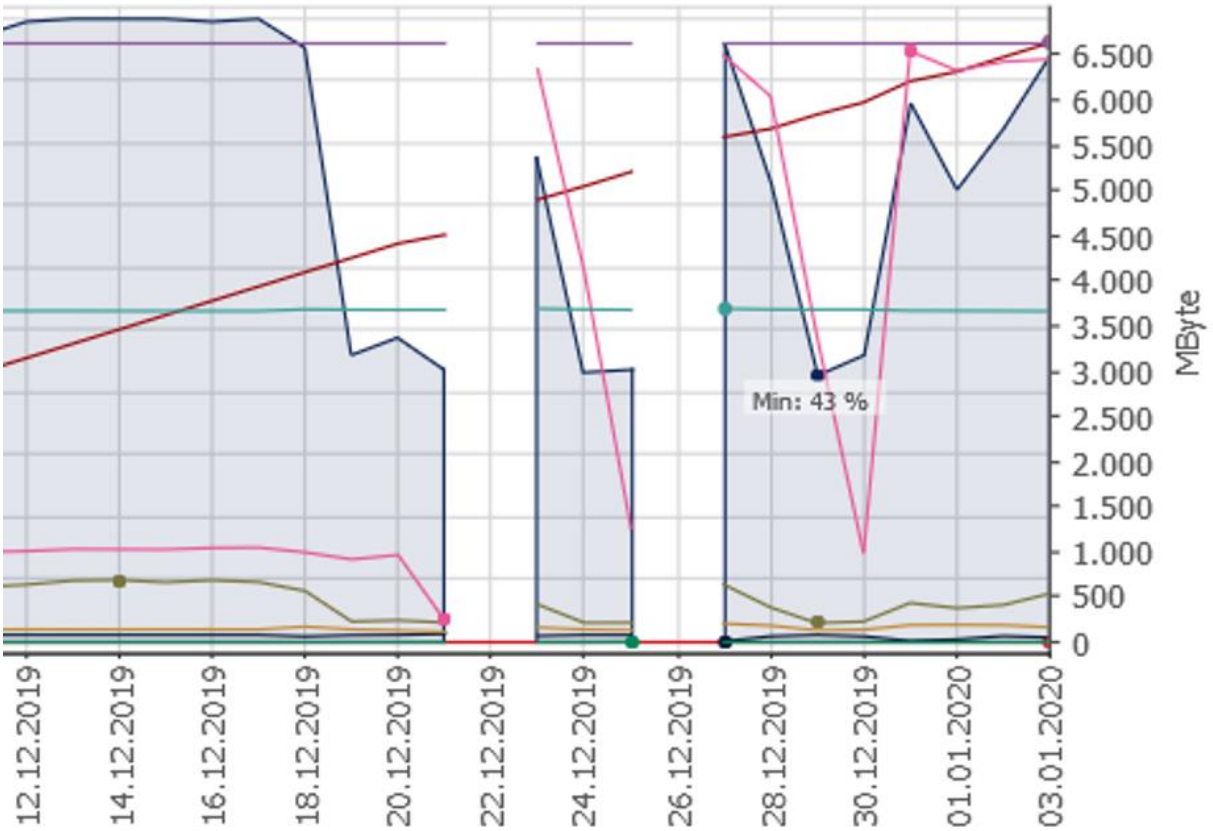
- Wann trat das Problem das erste Mal auf? Ggf. wurden zu dieser Zeit Konfigurationen oder Komponenten angepasst…
- Gibt es ein Muster im Problemverlauf? Vielleicht tritt ein Problem immer zu einer bestimmten Zeit auf? Eventuell kann aber auch ein zeitunabhängiger Trigger identifiziert werden, z.B. eine Anmeldung…

Die erste Frage ist meist einfacher zu beantworten. Interessanter Weise hilft mir hier mein jetzt wieder funktionales Monitoring. Der PRTG-Service überwacht sich selber. Und die Daten werden für Trendanalysen und Blicke in die Vergangenheit gespeichert. Also rufe ich den passenden Sensor auf:



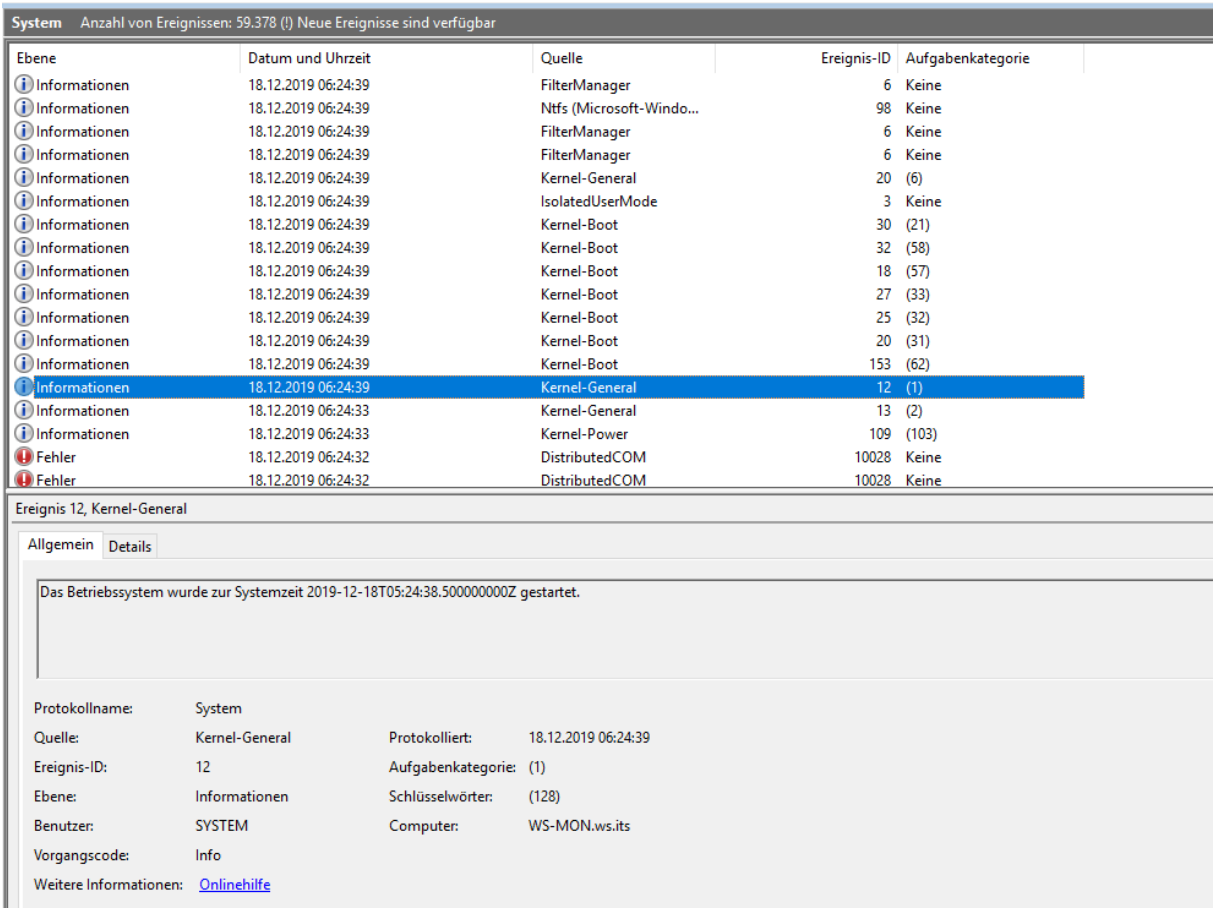
Der Text im Bild ist recht klein, aber man erkennt sehr gut den Ausfallzeitraum. Auch kann man meinen Versuch „Reboot tut gut“ nach der ersten Downtime erkennen. Und dass dieser nicht sehr lange hergehalten hat. Ebenso erkennt man, wie lange mein TroubleShooting gedauert hat. Und sehr wichtig ist auch die Aussage: Vorher gab es das Problem nicht – bis auf wenige Klicke ist der Statusgraph bei 100%.

Ich zoome etwas weiter rein:

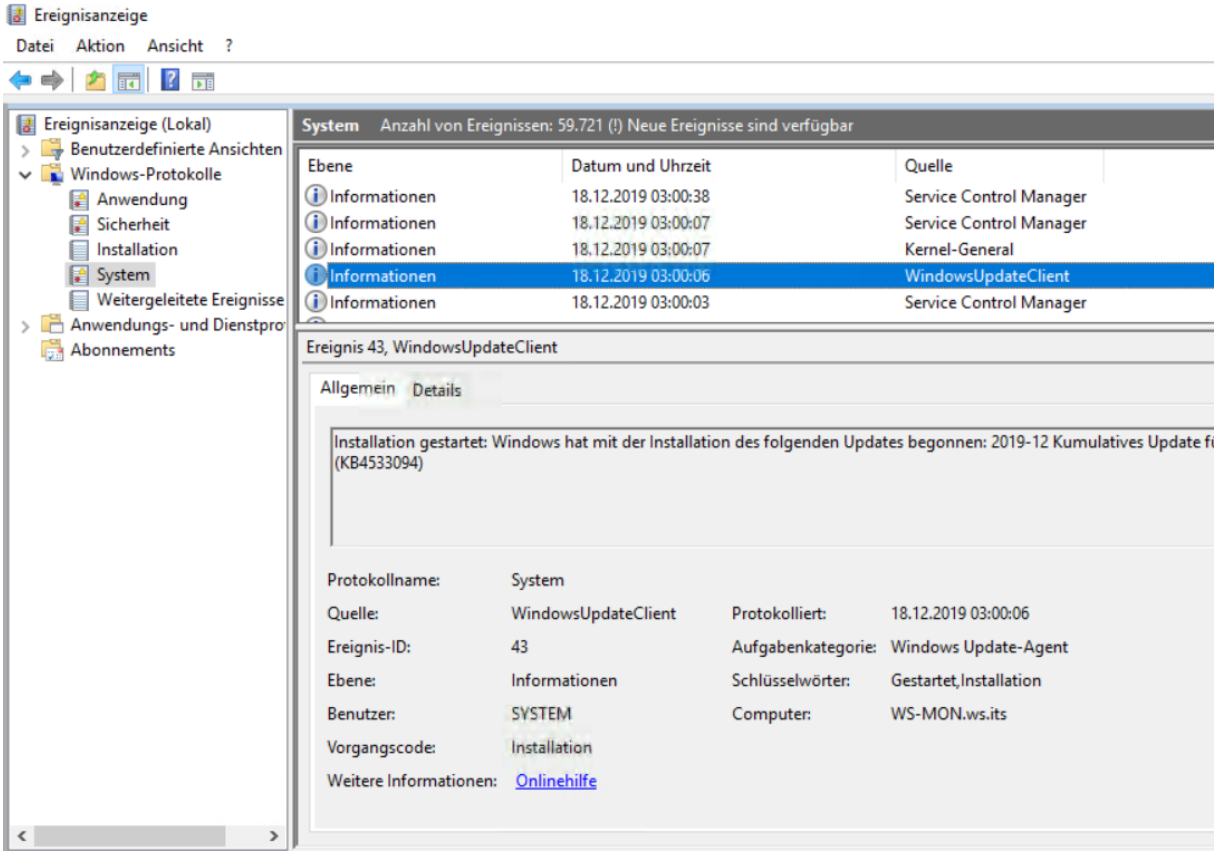


Jetzt erkennt man deutlich den 18.12.2019 als Startzeitpunkt. Und vielleicht erklären sich jetzt auch die langen Unterbrechungen ohne meine Anteilnahme.

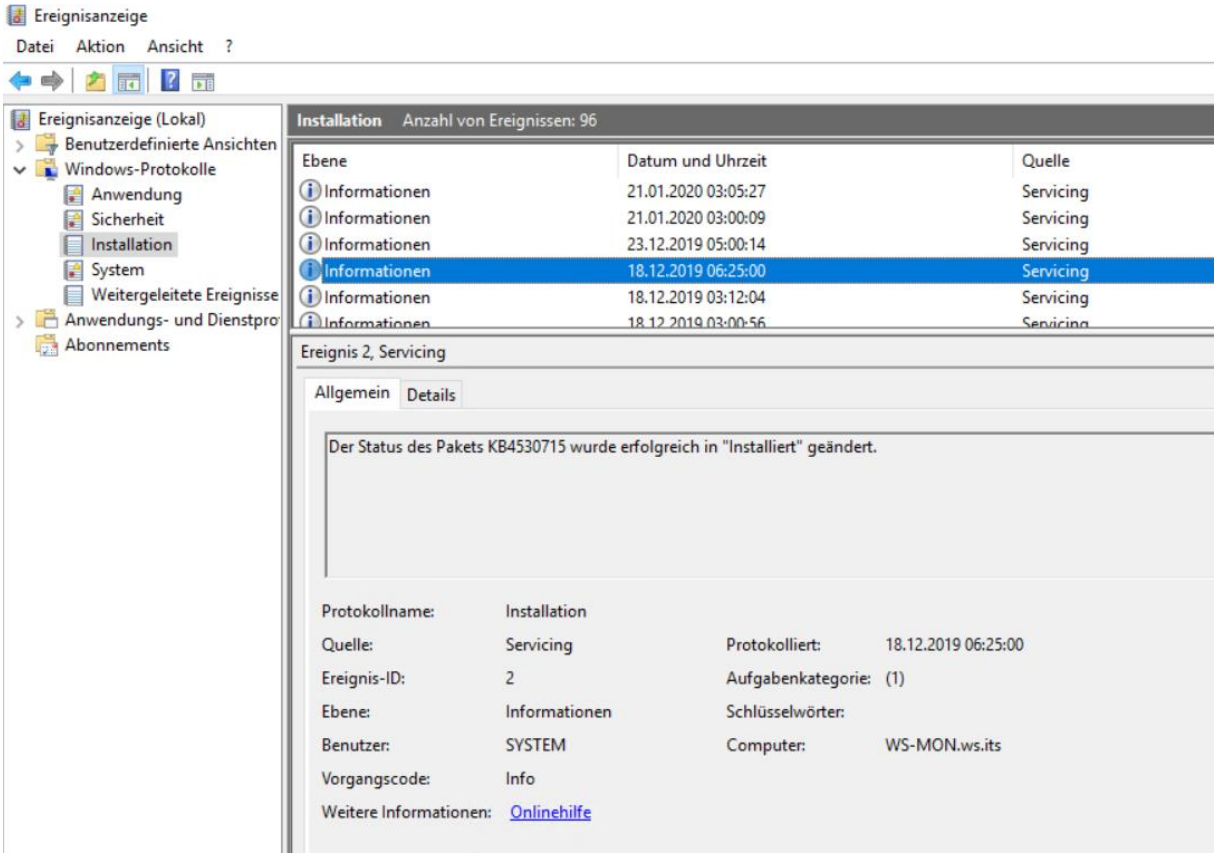
Mit dieser Information geht es weiter im Eventlog. Was passierte in der Früh des 18.12.? Es gab einen Neustart:



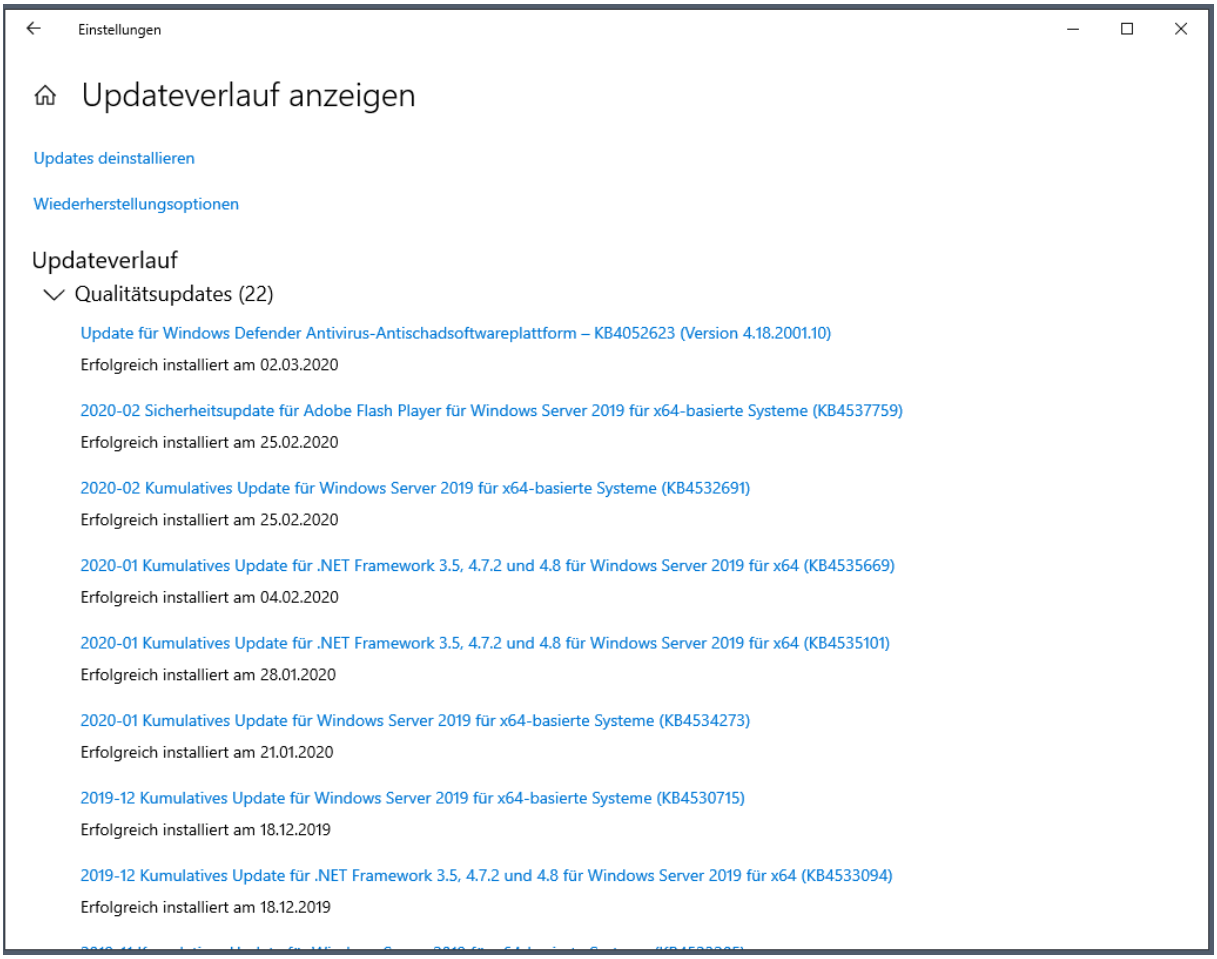
Aber warum hat sich der Server neugestartet? So viele Optionen gibt es da nicht. Ich war es nicht. Andere Administratoren gibt es bei mir noch nicht. Also war es der Server selber. Und das macht er nur bei Windows Updates. Die passenden Events finde ich ein paar Einträge weiter unten:



Und nach dem Neustart wurden die Updates als installiert markiert:



Ein Blick in den Update-Verlauf zeigt die Installation mit dem Datum an (Das Bild habe ich später neu erstellen müssen):



Es muss also in mit einem dieser beiden Updates eine Änderung vorgenommen worden sein. Im Update KB4530715 (https://support.microsoft.com/en-us/help/4530715/windows-10-update-kb4530715) sieht es in der Übersicht schon recht treffend aus. Der Device Guard ist schließlich ein Sicherheitsfeature:

Improvements and fixes

This security update includes quality improvements. Key changes include:

- Addresses an issue with diagnostic data processing when a device has the Diagnostic data setting enabled and set to Basic.
- Addresses an issue in which the Microsoft Store might fail to open on Windows on Arm.
- Security updates to Windows Virtualization, Windows Kernel, Windows Peripherals, the Microsoft Scripting Engine, and Windows Server.

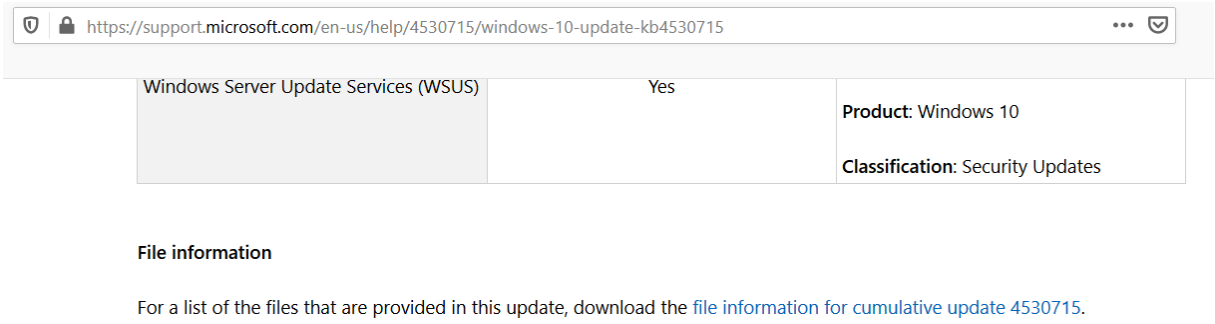
If you installed earlier updates, only the new fixes contained in this package will be downloaded and installed on your device.

For more information about the resolved security vulnerabilities, please refer to the Security Update Guide.

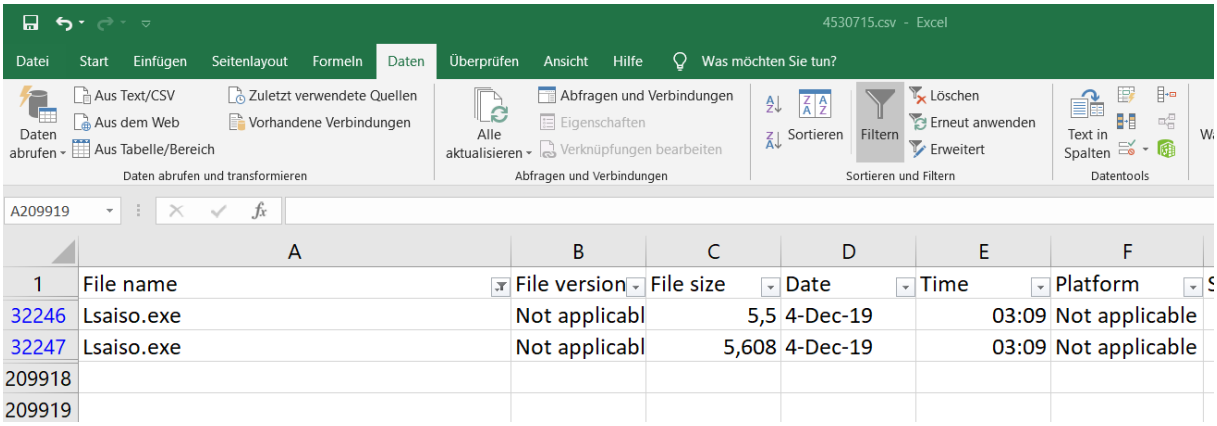
Ich folge dem Link „Security Update Guide“ (https://portal.msrc.microsoft.com/en-us/security-guidance) und finde die zum KB dazugehörige CVE-Nummer heraus

12/10/2019	Windows 10 Version 1809 for ARM64-based Systems		4530715	Security Update	CVE-2019-1469
12/10/2019	Windows Server 2019		4530715	Security Update	CVE-2019-1469
12/10/2019	Windows Server 2019 (Server Core installation)		4530715	Security Update	CVE-2019-1469
12/10/2019	Windows 10 Version 1709 for 32-bit Systems		4530714	Security Update	CVE-2019-1469

Viel interessanter ist aber der Link zu den Dateien, die vom Update geändert wurden. Diesen finde ich auf der Hauptseite des KBs.



Die Datei ist wie üblich schlecht strukturiert, aber ich finde die Lsaiso.exe gelistet.



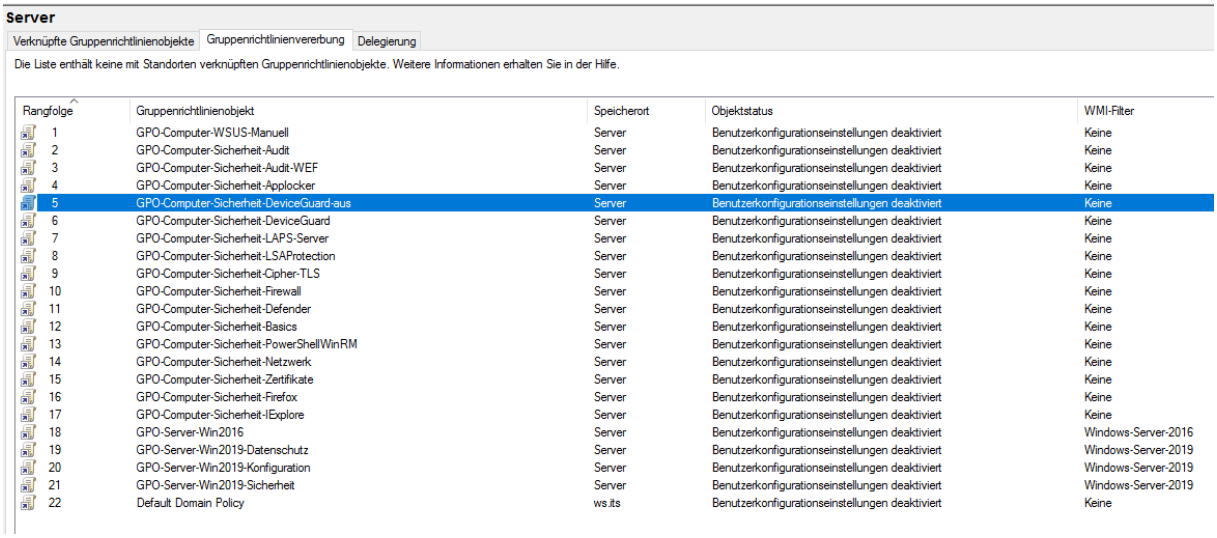
Die Lösung

Ein Workaround

Ich sehe 2 ordentliche Möglichkeiten für die Lösung meiner Problematik:

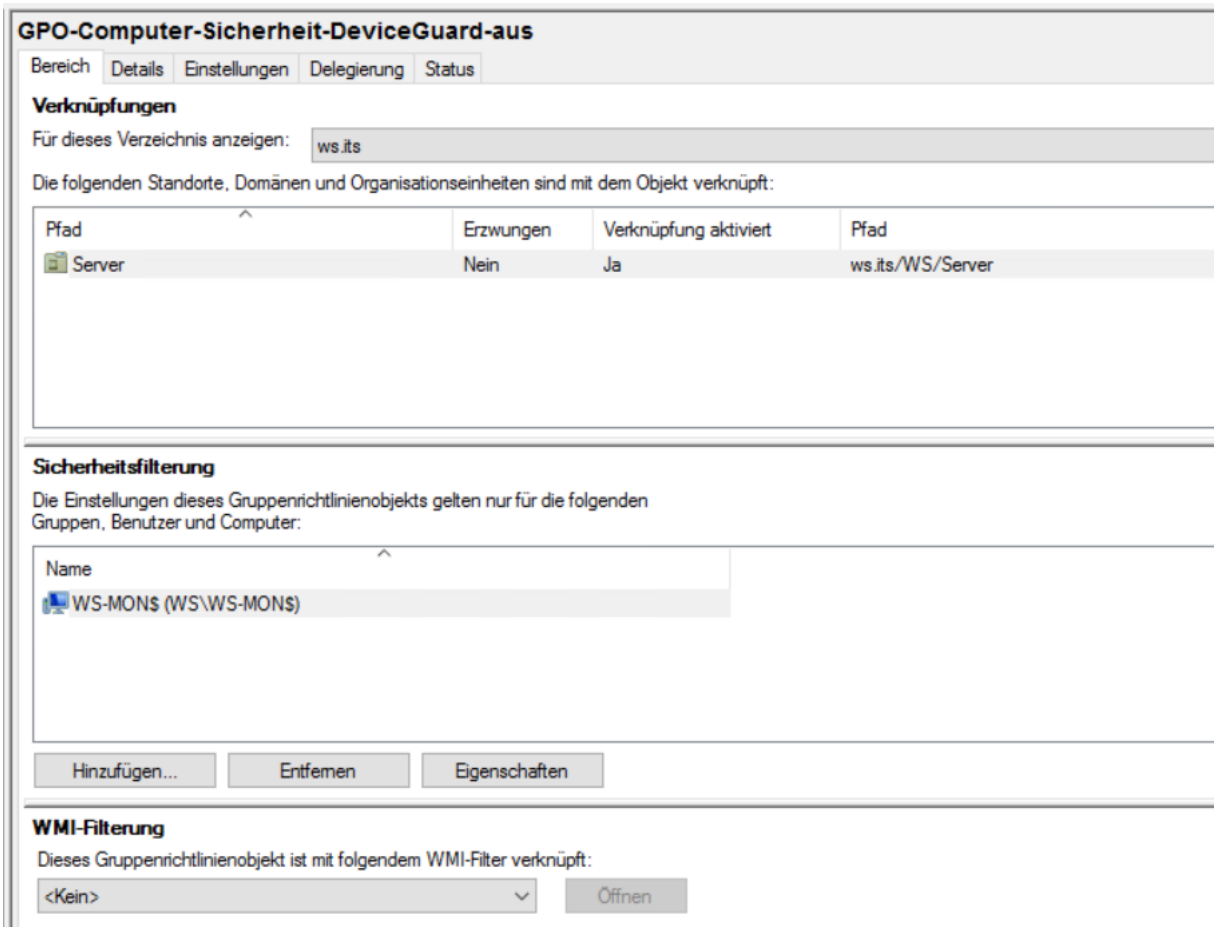
- Microsoft korrigiert den Fehler im Credential Guard mit einem späteren Update.
- PRTG passt seine Anmeldeprozesse an bzw. gibt Hinweise zum Arbeiten in Umgebungen mit aktivem Credential Guard.

Beide Lösungen werden Zeit benötigen. Diese hat mein Server aber nicht. Daher werde ich bis zur finalen Lösung den Device Guard auf diesem einen Server deaktivieren. Dafür erstelle ich eine separate GPO mit der Einstellung und wende diese gefiltert auf meinen PRTG-Server an. Hier sieht man die GPO mit der Device Guard Deaktivierung. Sie wird in der Rangfolge vor der Richtlinie mit der Aktivierung angewendet. Damit „gewinnt“ ihre Einstellung:



Rangfolge	Gruppenrichtlinienobjekt	Speicherort	Objektstatus	WMI-Filter
1	GPO-Computer-WSUS-Manuell	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
2	GPO-Computer-Sicherheit-Audit	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
3	GPO-Computer-Sicherheit-Audit-WEF	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
4	GPO-Computer-Sicherheit-Applocker	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
5	GPO-Computer-Sicherheit-DeviceGuard-aus	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
6	GPO-Computer-Sicherheit-DeviceGuard	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
7	GPO-Computer-Sicherheit-LAPS-Server	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
8	GPO-Computer-Sicherheit-LSAProtection	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
9	GPO-Computer-Sicherheit-Cipher-TLS	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
10	GPO-Computer-Sicherheit-Firewall	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
11	GPO-Computer-Sicherheit-Defender	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
12	GPO-Computer-Sicherheit-Basics	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
13	GPO-Computer-Sicherheit-PowerShellWinRM	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
14	GPO-Computer-Sicherheit-Netzwerk	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
15	GPO-Computer-Sicherheit-Zertifikate	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
16	GPO-Computer-Sicherheit-Firefox	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
17	GPO-Computer-Sicherheit-IExplore	Server	Benutzerkonfigurationseinstellungen deaktiviert	Keine
18	GPO-Server-Win2016	Server	Benutzerkonfigurationseinstellungen deaktiviert	Windows-Server-2016
19	GPO-Server-Win2019-Datenschutz	Server	Benutzerkonfigurationseinstellungen deaktiviert	Windows-Server-2019
20	GPO-Server-Win2019-Konfiguration	Server	Benutzerkonfigurationseinstellungen deaktiviert	Windows-Server-2019
21	GPO-Server-Win2019-Sicherheit	Server	Benutzerkonfigurationseinstellungen deaktiviert	Windows-Server-2019
22	Default Domain Policy	ws.its	Benutzerkonfigurationseinstellungen deaktiviert	Keine

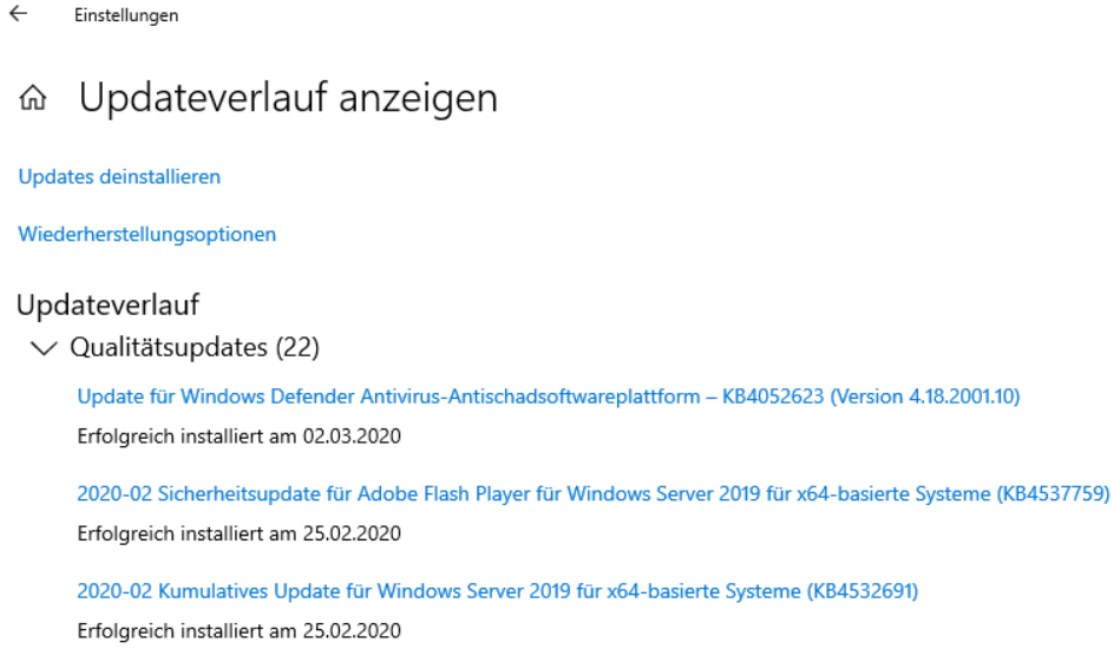
Damit aber nur der Monitor-Server editiert wird, verwende ich einen Sicherheitsfilter:



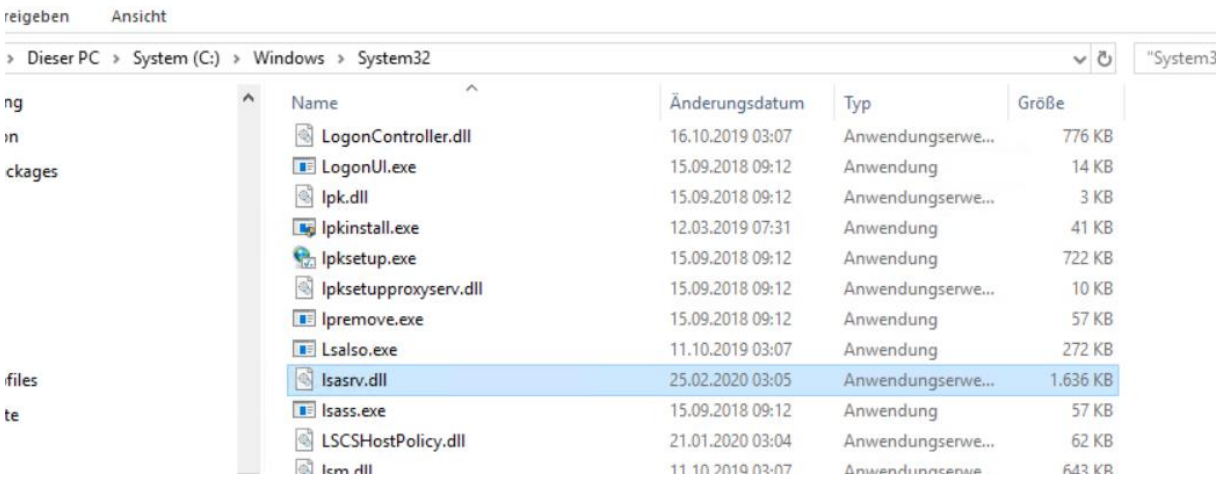
Natürlich muss ich dafür die Prozedur mit dem UEFI-Unlock erneut ausführen.

neues Update – neuer Versuch

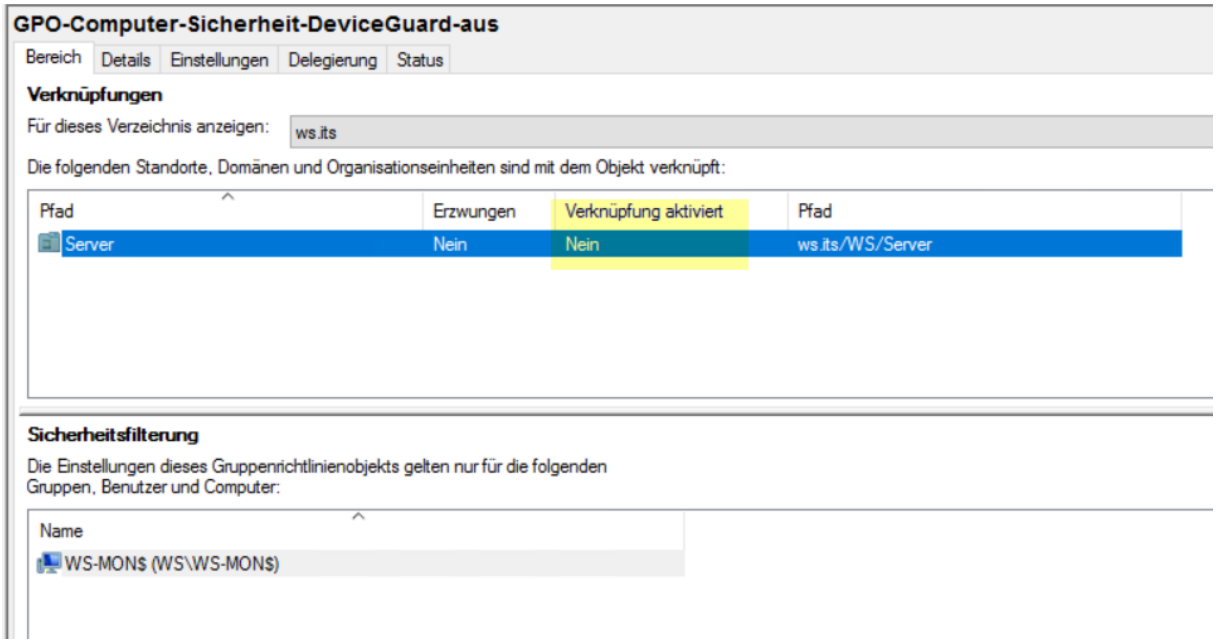
Es sind nun einige Wochen vergangen. Der Server läuft ohne den Credential Guard wieder stabil. Zwischenzeitlich hat mein Server das Update vom Februar (2020-02) installiert:



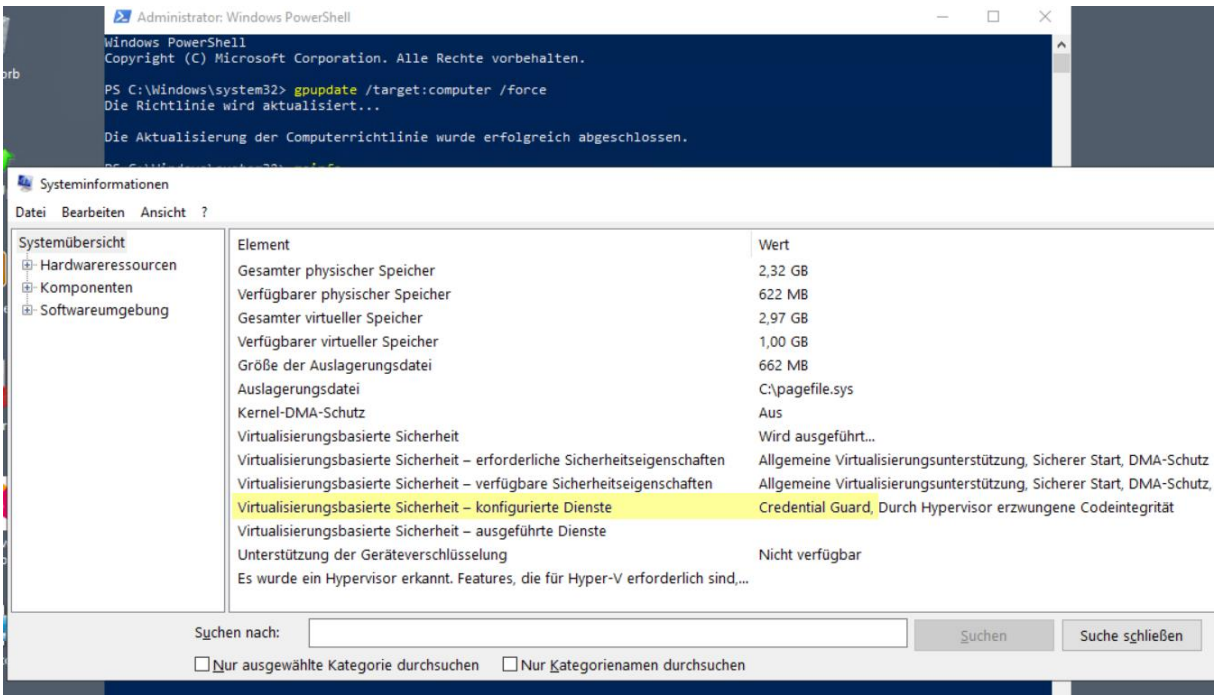
Ein Blick ins Dateisystem verrät, dass eine LSA-DLL modifiziert wurde. Das könnte eine neue Chance für die Reaktivierung des Device Guards sein:



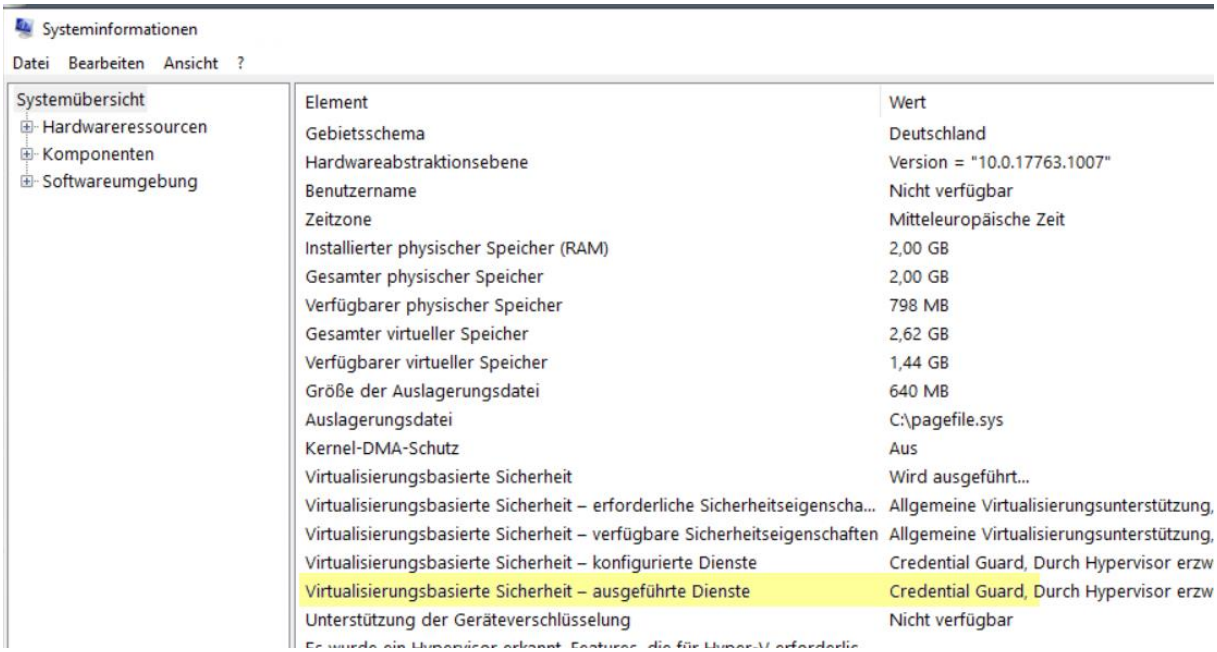
Ich deaktiviere die GPO, mit der ich für diesen einen Server die Abschaltung vorgenommen habe. Damit „gewinnt“ wieder die andere Richtlinie und der Credential Guard sollte reaktiviert werden:



Ich beschleunige den Vorgang durch ein gpupdate auf meinem Monitor-Server. Mit msinfo sehe ich bereits die Einstellung. Diese greift aber erst nach einem Neustart:



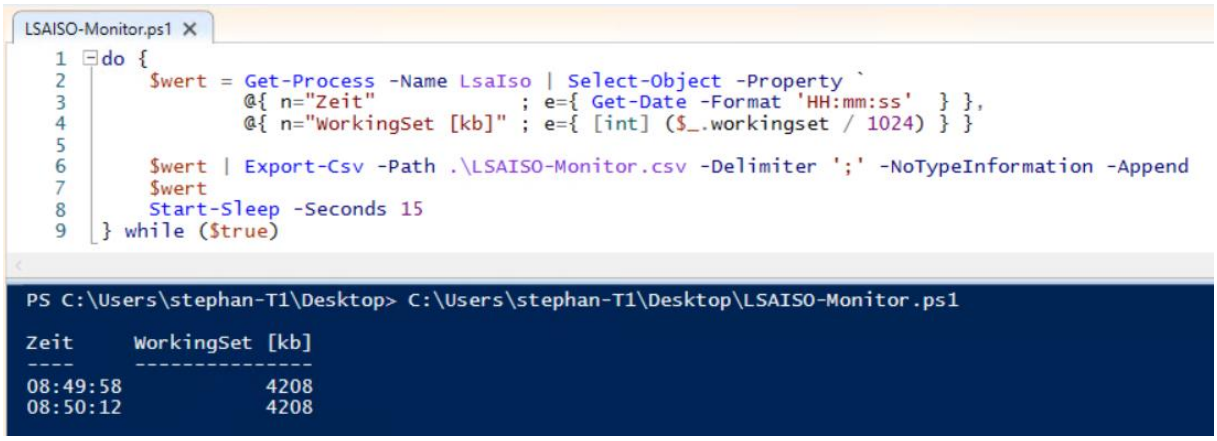
Also initialisiere ich den Reboot. MSInfo zeigt nun einen laufenden Credential Guard:



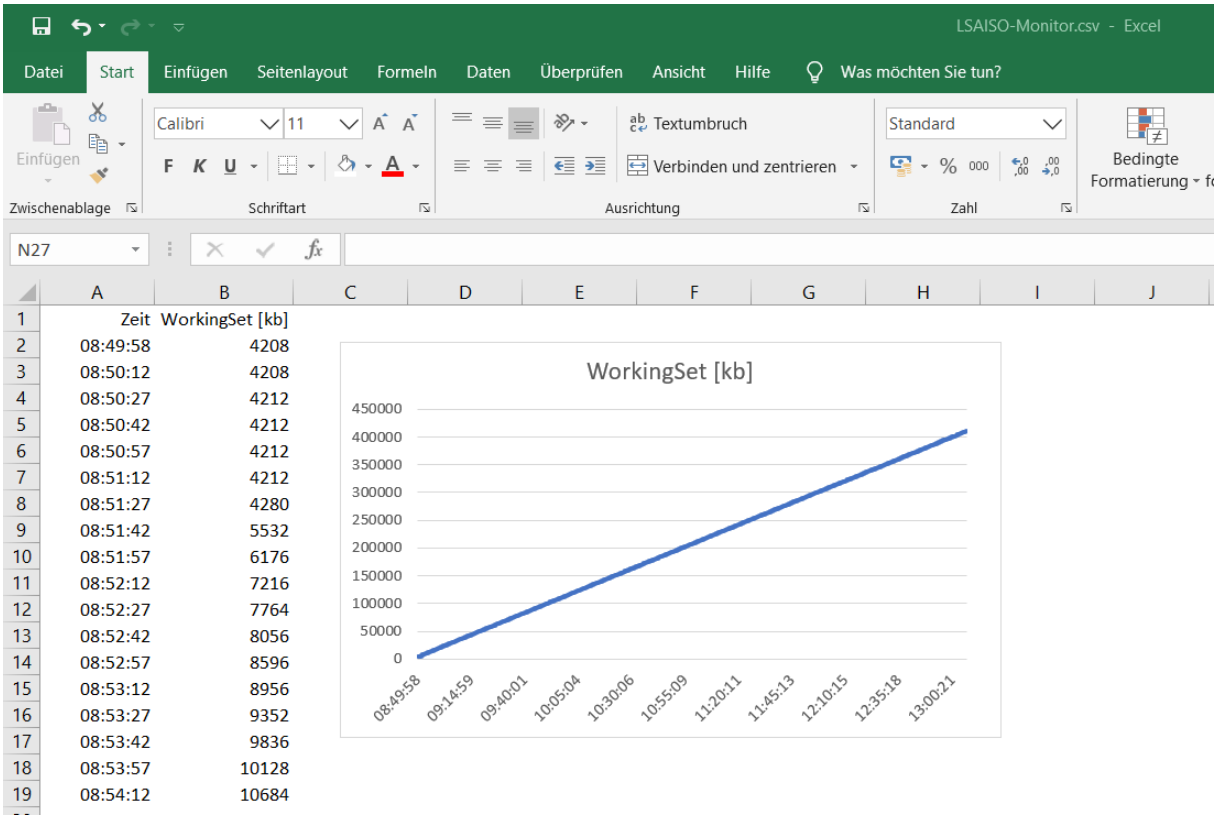
Und wie entwickelt sich dessen Hunger auf Arbeitsspeicher? Das soll ein keines PowerShell-Script aufzeigen:

```
do {
    $wert = Get-Process -Name LsaIso | Select-Object -Property `
            @{ n="Zeit"            ; e={ Get-Date -Format 'HH:mm:ss'  } },
            @{ n="WorkingSet [kb]" ; e={ [int] ($_.workingset / 1024) } }

    $wert | Export-Csv -Path .\LSAISO-Monitor.csv -Delimiter ';' -NoTypeInformation -Append
    $wert
    Start-Sleep -Seconds 15
} while ($true)
```

Ich starte das Script und lass den Server arbeiten:



Nach einiger Zeit hole ich mir die erzeugte CSV-Datei auf meinen Client und lasse Excel die Daten grafisch darstellen. Leider bläht sich LSAISO wieder auf:



	A	B
1	Zeit	WorkingSet [kb]
2	08:49:58	4208
3	08:50:12	4208
4	08:50:27	4212
5	08:50:42	4212
6	08:50:57	4212
7	08:51:12	4212
8	08:51:27	4280
9	08:51:42	5532
10	08:51:57	6176
11	08:52:12	7216
12	08:52:27	7764
13	08:52:42	8056
14	08:52:57	8596
15	08:53:12	8956
16	08:53:27	9352
17	08:53:42	9836
18	08:53:57	10128
19	08:54:12	10684

Die beiden Updates seit dem ersten Auftreten durch den Patch 2019-12 haben das Problem leider nicht gelöst. Mir bleibt nichts anderes über, als zu dem Workaround zurück zu kehren:

- Ich aktiviere die GPO mit der Device Guard Deaktivierung wieder.
- Danach wende ich die Richtlinie durch ein gpupdate an und starte den Server neu.
- Nun editiere ich den UEFI-Start und deaktiviere den UEFI-Lock des Device Guards auf der Konsole des Servers nach einem weiteren Neustart.

Es wird Zeit, den Hersteller zu informieren. Vielleicht habe ich ja einen Hinweis übersehen?

Zusammenfassung

Auch wenn ich das Problem nicht zufriedenstellend lösen kann: Ich kenne nun die Ursache. Zusätzlich habe ich einen funktionalen Workaround gefunden.