Patch Management ist keine Raketen Wissenschaft mehr

Meine Erfahrungen und einfache Schritte

Jürgen Rinelli

Jürgen Rinelli
Am Eichet 6a
85302 Alberzell
info@success.eu.com

Inhaltsverzeichnis

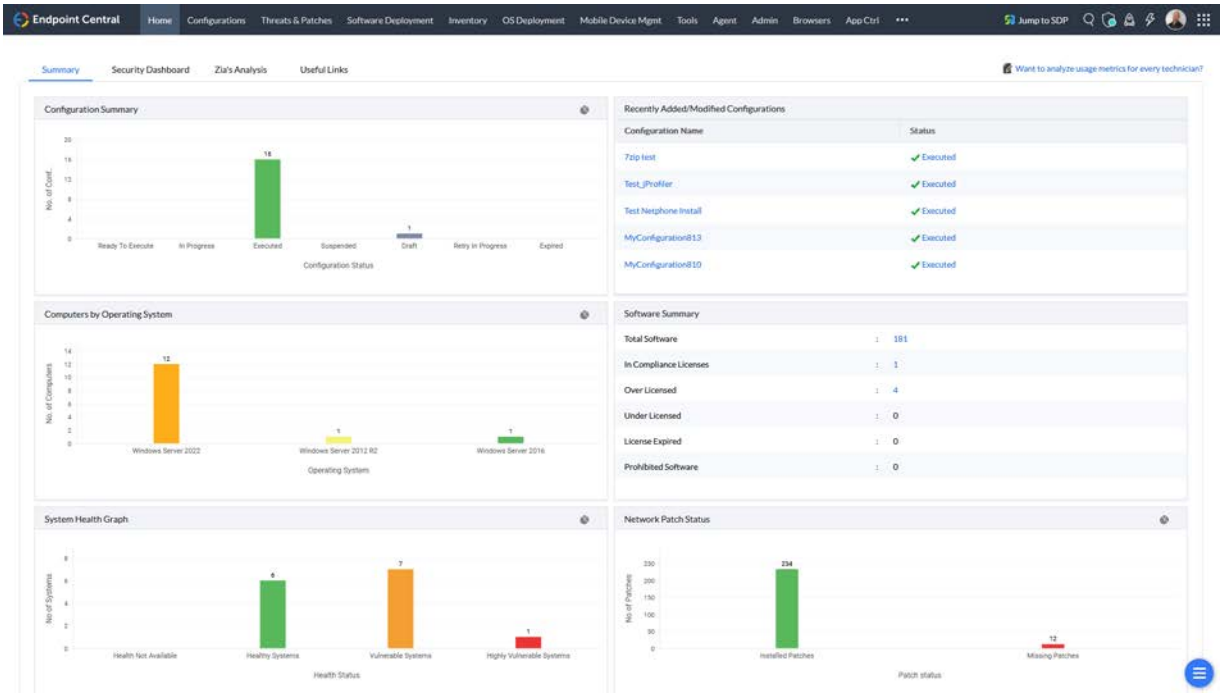
1. Ein paar Worte

Als Senior Consultant sehe ich immer wieder Firmen, die gehackt wurden, und unterstütze diese beim Neuaufbau. Meist ist der Grund eine unzureichende oder nicht vorhandene Endpoint-Management-Strategie. Was früher noch mit viel Aufwand einherging, ist heute keine Raketenwissenschaft mehr.

Bei der Vielzahl an Cyberthreads und durch Nutzung von KI Technologien der Angreifer, werden Attacken immer intelligenter, schneller und anpassungsfähiger.

In diesem kleinen E-Book teile ich meine Best Practices für das Endpoint-Patch-Management und zeige wie einfach eine umfangreiche Patch-Strategie eingerichtet werden kann. Meine persönlich bevorzugte Lösung und Grundlage auf welcher die Screenshots und Einstellungen basieren, ist von ManageEngine.

Ich nutze und empfehle die EndpointCentral Security Edition, wobei meine Best Practices bezogen auf das reine Patchmanagement dieses mini E-Books, auch mit dem VulnerabilityManager Plus angewendet werden können.



„Patch-Management“ und „Endpoint Security“ sind Begriffe, welche sich heute keinesfalls mehr trennen lassen. Ich erzähle auch nichts Neues, wenn ich sage, dass es die Endgeräte sind, welche am anfälligsten für Cyberattacken sind. **Alle** Endgeräte – worunter ich auch die User zähle!

Während ich bei Usern ein „Patch-Management“ nur über konsequente Schulungen durch gefakte E-Mails, Links, etc. durchführen kann – und sollte, lässt sich Hardware mit deutlich geringerem Widerstand und Aufwand patchen.

Was ist für mich wichtig und welche Schritte gehe ich beim Patch-Management?

1. Ich richte ein automatisches Test und Freigabeverfahren ein.
2. Ich richte **zwei** automatische Verteilungen der freigegebenen Patches ein.
3. Ich prüfe, welche Schwachstellen noch offen sind im Threat-Management und behebe diese.
4. Ich prüfe regelmäßig die Dashboards bei einer guten Tasse Cappuccino.

Nach diesem ersten einfachen Schritt der Patch-Automatisierung durch automatisches Testen und anschließender automatischer Verteilung, ist eine Grundsicherheit der Geräte gegeben. Das damit die Endpoints nicht komplett gesichert sind, habe ich schon erwähnt. Es gilt auch biologische Endpoints (User) an die Hand zu nehmen. Auf diesen Punkt gehe ich in einem separaten E-Book mit den Themen Device-Control, Applikation-Control, Data-Loss-Prevention, Anti-Ransomware und ganz wichtig – Browser Security ein.

Sollen wir loslegen?

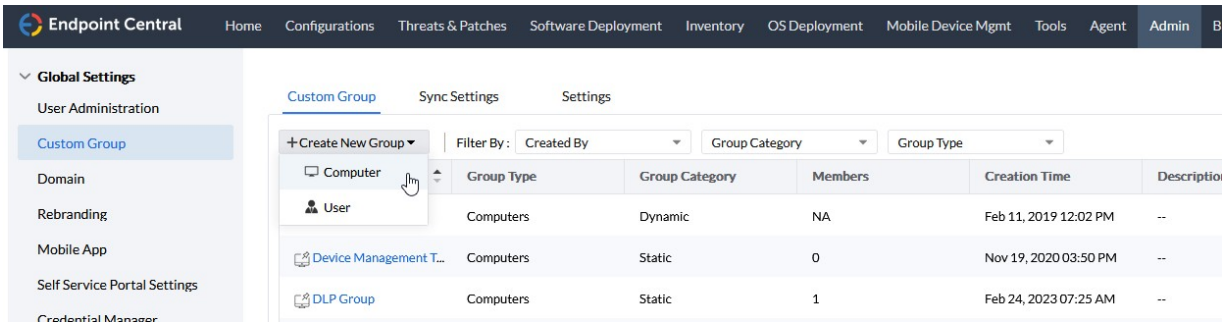
2. Schritt für Schritt

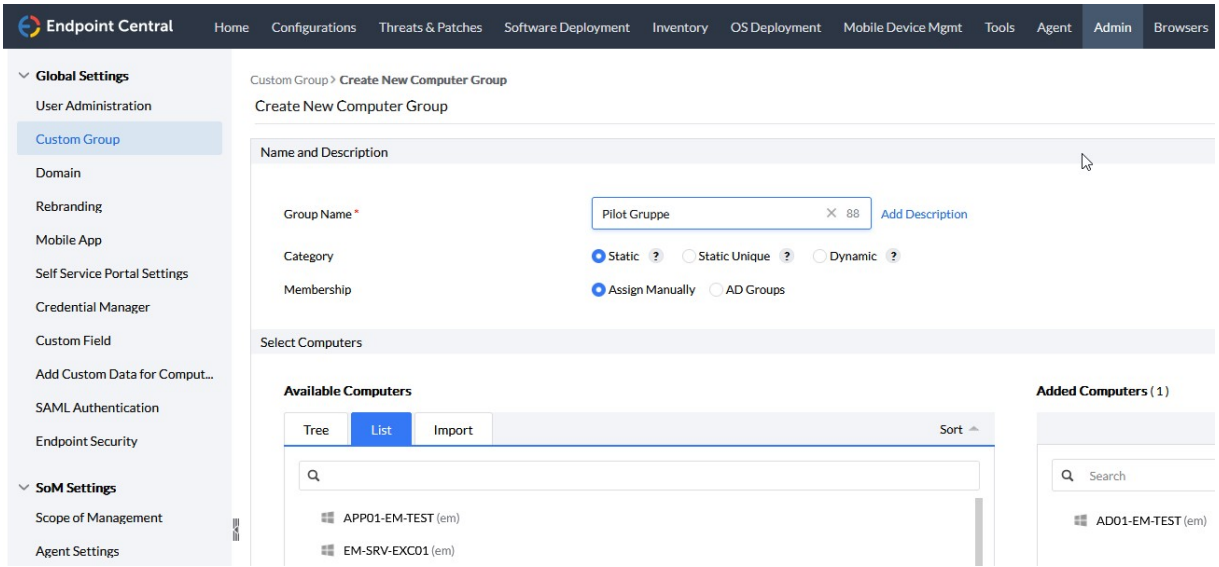
2.1 Automatisiertes Test und Freigabeverfahren:

Da ein Patch auch zu unerwünschtem Verhalten führen kann, gebe ich Patches nur in den seltensten Fällen ungeprüft raus. Ja – besser wäre es, alles zu testen!
Um mir die Arbeit einfacher zu machen, gebe ich das Testen ab. Hierzu erstelle ich zuerst eine Testgruppe mit meinen eigenen Testgeräten sowie auch Geräten von Key-Usern. Also User aus jeder Abteilung, welche mit der Software täglich arbeiten. Mir ist es oft aus Lizenz und Berechtigungsgründen nicht möglich, Aktionen in der Software auszuführen. Oft kann ich die Software überhaupt nicht auf meinen Testgeräten installieren, wegen fehlender Lizenz. Also benötige ich User mit dieser Software welche auch Aktionen in dieser ausführen. Nur so kann sichergestellt werden das ein weiterführender Link z.B. zu einer Reader-Applikation, auch funktioniert.
Bei den Key-Usern binde ich vor allem die User ein welche sich bisher auch schon häufig mit Fehlern oder langsamen Systemen gemeldet haben. – Ich will ja Feedback und den möglichst schnell!

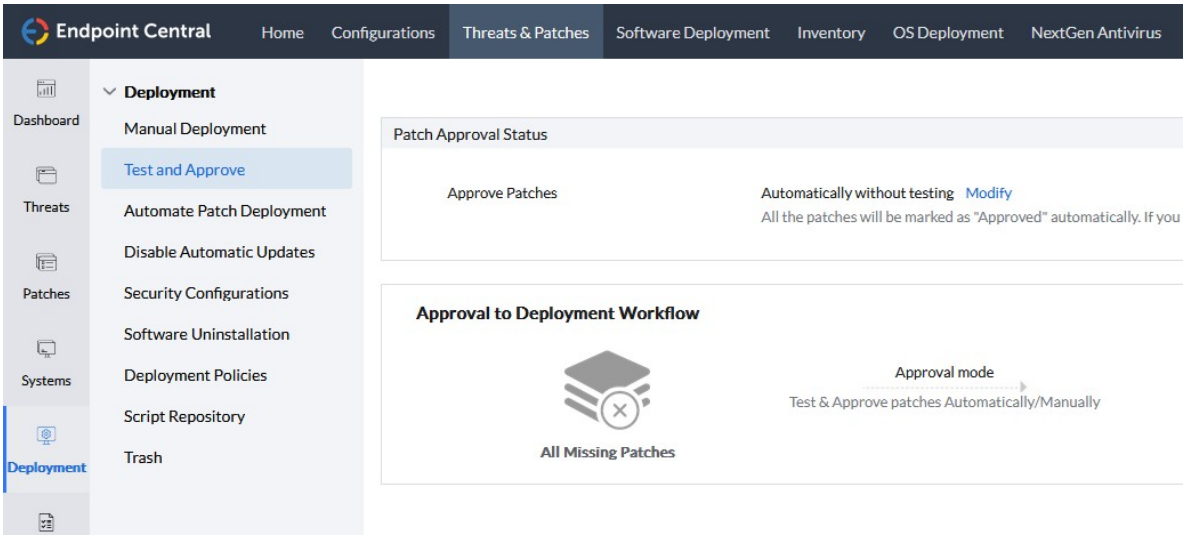
Wie eingangs bereits erwähnt zeige ich die Einrichtungsschritte exemplarisch an meinem Favoriten dem EndpointCentral UEMS von ManageEngine, denn einfacher und intuitiver habe ich es bisher in noch keiner anderen Endpoint Management Lösung auf dem Markt gefunden – und ich habe mit vielen gearbeitet in den letzten 25 Jahren.

Zuerst erstelle ich eine statische Computergruppe über Admin/Custom Group und packe alle Testsysteme und Key-User Systeme hinein.

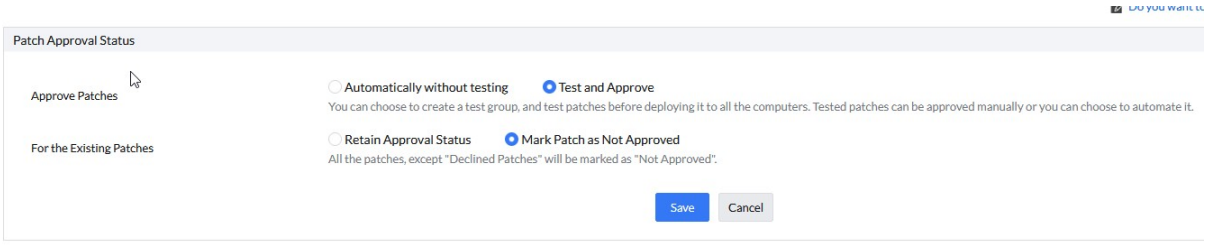




Weiter gehts in Threads & Patches/Deployment/Test and Approve, um ein automatisches Test- und Freigabeverfahren einzurichten.



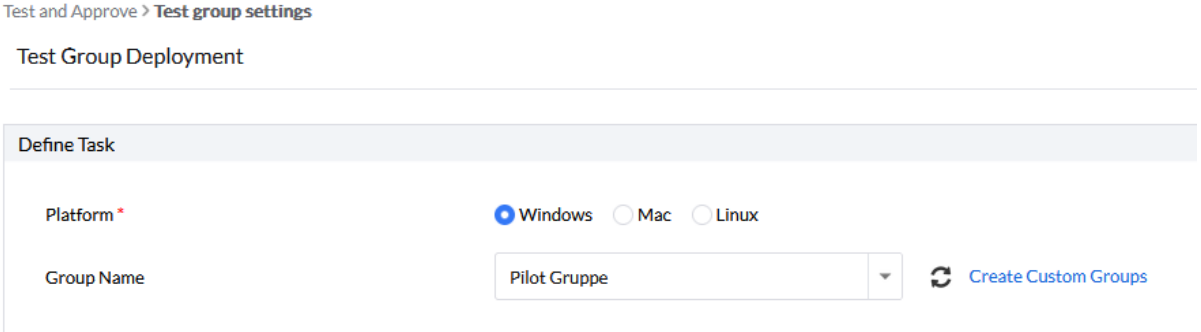
Dort stelle ich – falls noch nicht geschehen – den Patch Approval Status über den Modify-Button von „Automatically without testing“ um auf „Test and Approve” und „Mark Patch as not Approved” um die bereits automatisch freigegebenen Patches sicherheitshalber auf nicht freigegeben, zu setzen.



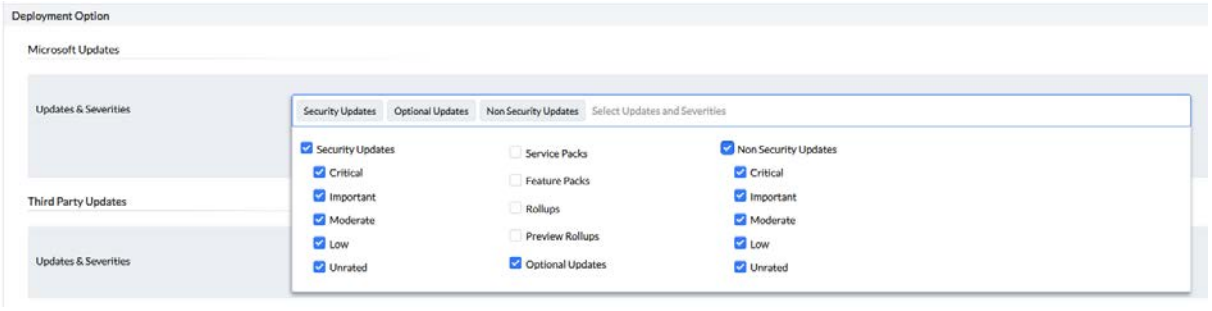
Nach dem Speichern der Einstellung kann ich über „AddGroup“ meine erste Testgruppe anlegen. Es können verschiedene erstellt werden für unterschiedliche Szenarien. Ich trenne Clients und Server auf jeden Fall voneinander. Ja – auch Server können zu einem gewissen Teil automatisch gepatcht werden.

Im ersten Schritt wähle ich die Plattform aus: Windows, Mac oder Linux

Dann wird die zuvor erstellte Gruppe benötigt – „Pilot Gruppe“ in diesem Beispiel.

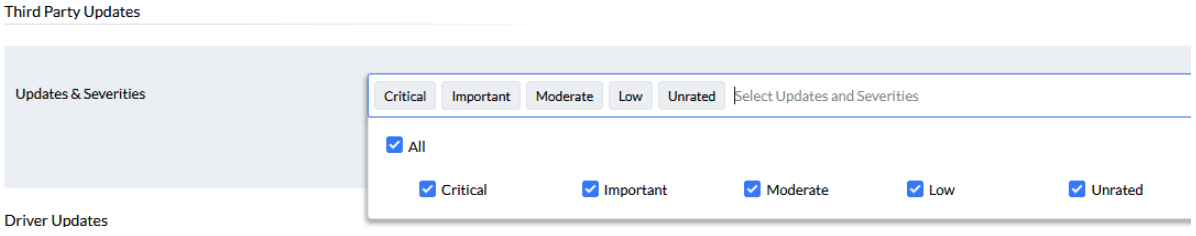


In den „Deployment Options“ wähle ich bei Microsoft Updates wie folgt:

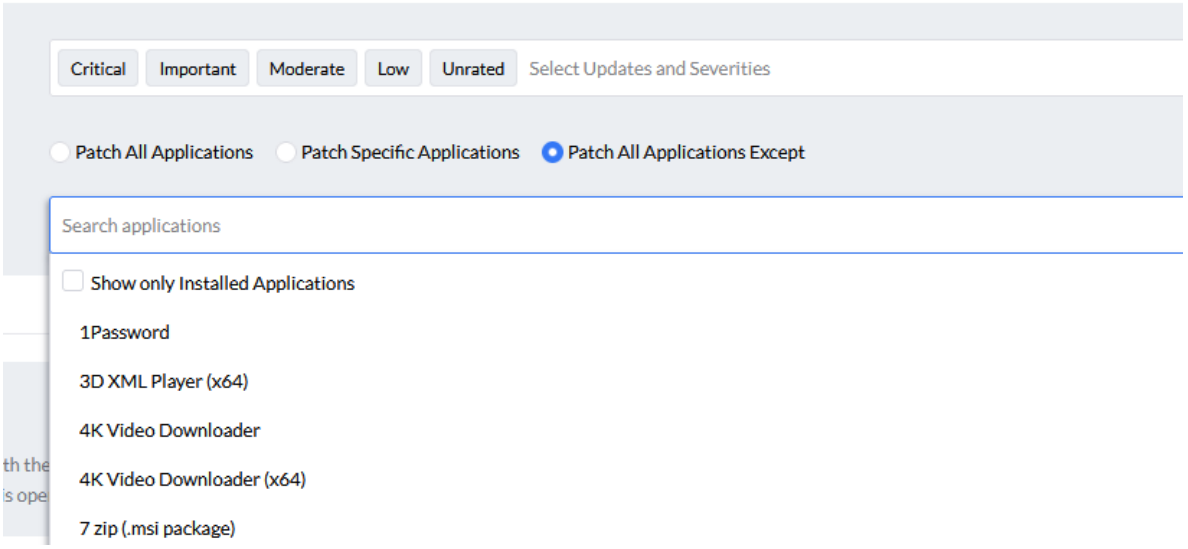


Service Packs, Feature Packs und der Gleichen teste ich weiterhin manuell oder erstelle eine spezielle Testgruppe hierfür.

Bei Third Party Updates definiere ich alles.



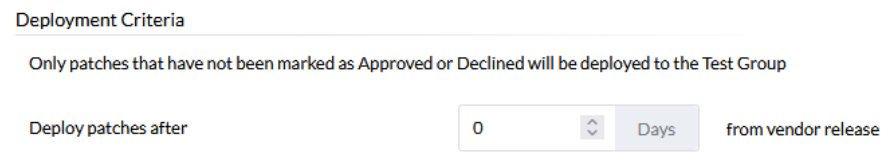
Wenn ich aus den Auswahldialogen gehe, kann ich spezielle Anwendungen bei Microsoft Updates und Third Party Updates auch vom Patchen ausschließen. Z.B. spezielle VPN Lösungen, Workspace Applikationen, SQL, Java, etc.



Treiber Updates teste ich persönlich manuell oder teste diese über eine spezielle Gruppe an Geräten in einem weiteren Automatismus.

Deployment Criteria, damit kann ich eine Verzögerung definieren, also einen Zeitpunkt ab dem Patch-Release, ab welchem das Patch ausgerollt werden soll.
Ich belasse das auf 0 Tage, da ich in eine Testgruppe verteile und ich eine Freigabe des Patches im späteren Verlauf, nicht noch länger hinausschieben möchte.

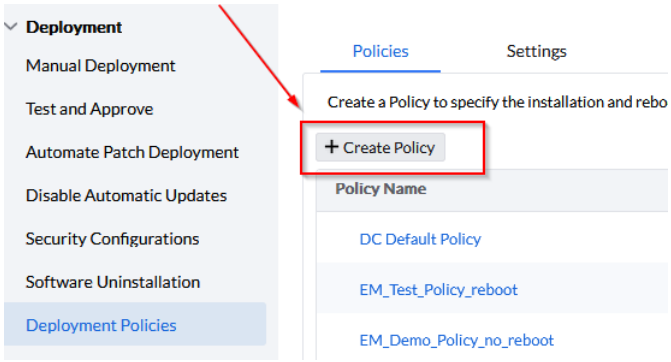
Ja – auch wenn es bedeutet, dass ein System u.U. einmal ausfallen kann! Dann setze ich das System einfach neu auf, kompromittiere damit aber nicht meine Gesamtsicherheit.



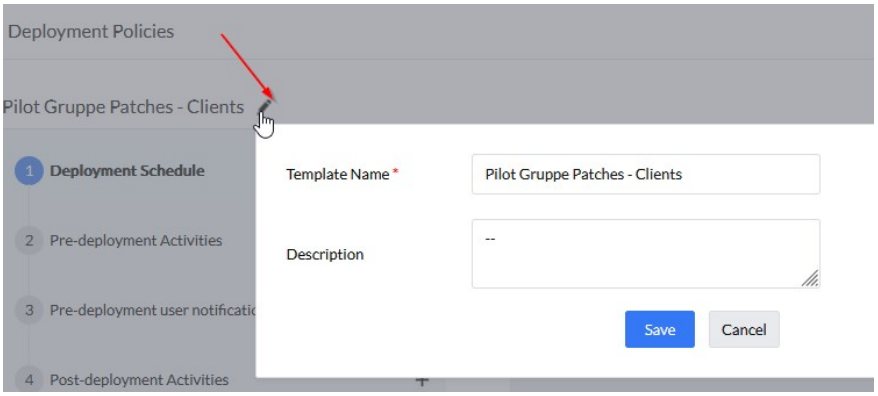
Im nächsten Schritt wähle ich die Deployment Policy – also das „Wann und Wie“ der Verteilung.

Hier erstelle ich eine eigene Policy, um das Verteilungsfenster auch unabhängig anpassen zu können.

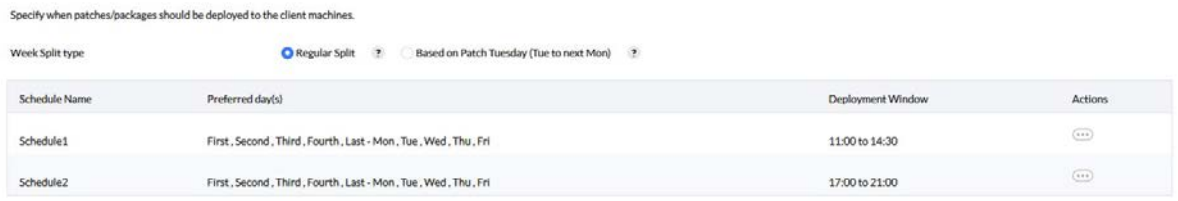
Also rauf auf „Create/Modify/Save As Policy”, was mich in ein neues Tab, zu den Deployment Policies führt. Hier auf „Create Policy“ gehen.



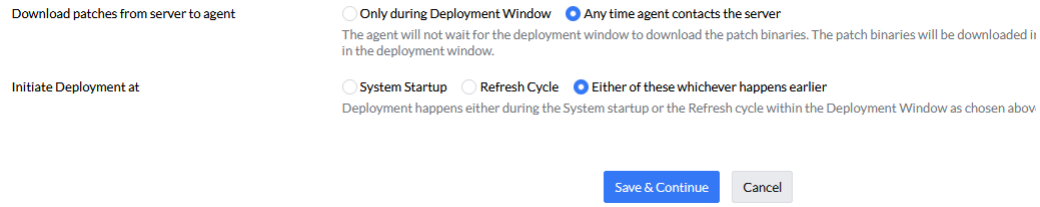
Aussagekräftigen Namen geben.



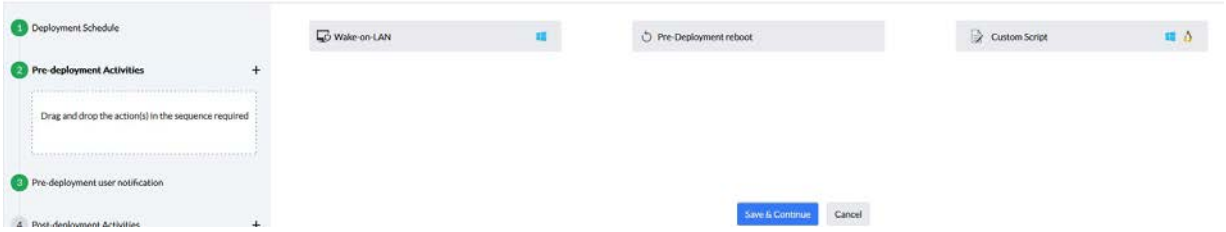
Im Schritt 1 gebe ich das Zeitfenster an. Ich bevorzuge hier täglich wochentags (wenn die IT am Wochenende besetzt ist, dann auch Samstag und Sonntag), sowie zwei Zeitfenster, um auch auf flexiblere Arbeitszeiten zu reagieren. Man könnte jetzt fragen warum nicht gleich 24/7 patchen oder von 07:00 – 21:00 Uhr. Mein Gedanke bei einer festen Patch-Zeit ist, das ich auch die freundlichen Key-User nicht den ganzen Tag stressen möchte. Ich kann so den freundlichen Helfern eine fest Zeit geben in der die Patches installiert werden und sie können sich darauf einstellen. Die Mittagszeit bietet sich dazu auch am besten an.



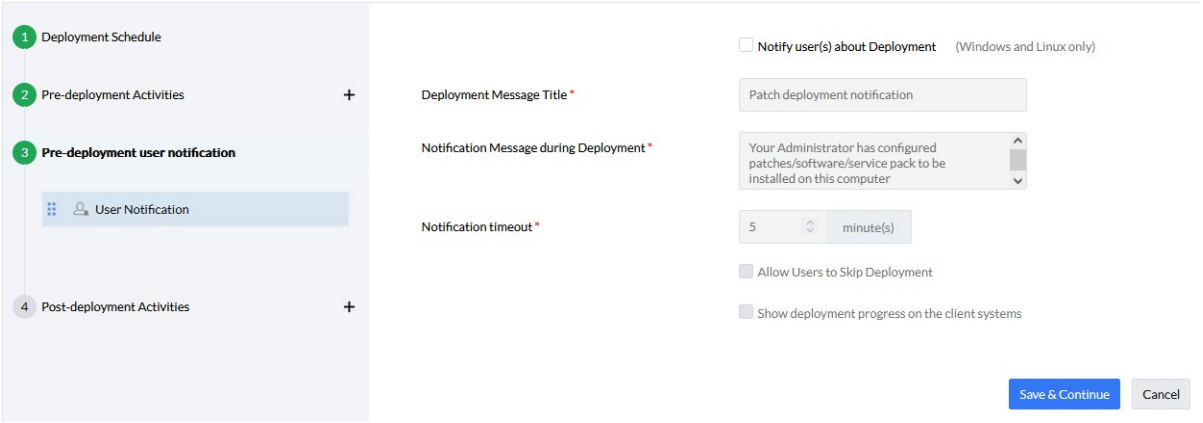
Patches lasse ich bereits Cashen wenn diese bereitstehen und verteilt werden soll zu jeder Zeit innerhalb des definierten Zeitfensters. So spare ich Bandbreite und Zeit.



Nach „Save & Continue“ kann ich Pre-deployment Activities definieren. Wake-on-Lan sollte ich eine Verteilung nachts durchführen wollen, Pre-Deployment reboot oder ein Custom Skript laufen lassen. In meinem Beispiel definiere ich nichts und gehe gleich auf „Save & Continue“.

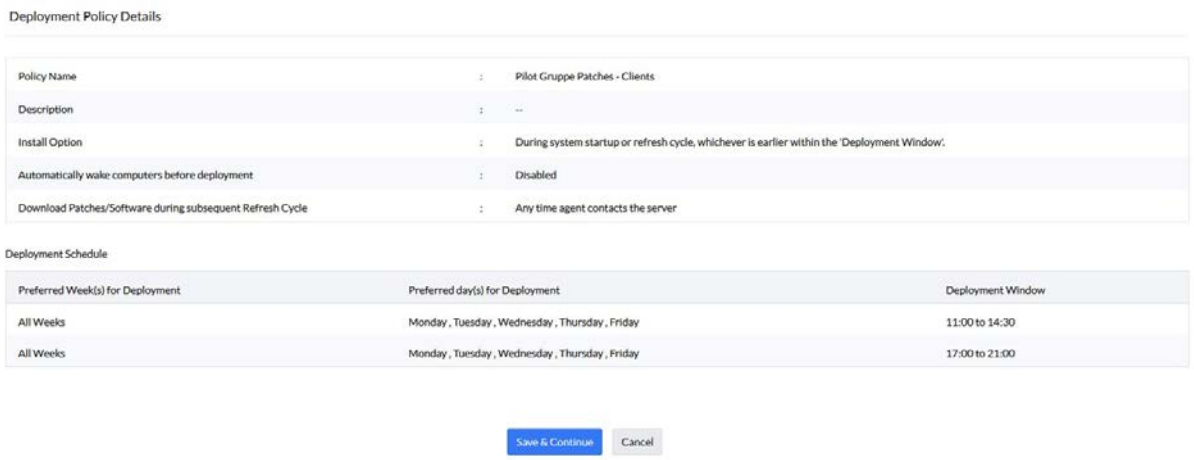


Schritt 3, bietet mir die Möglichkeit, die User vor der Verteilung zu benachrichtigen und ihnen ein Überspringen zu ermöglichen. Auch hier definiere ich nichts und gehe mit „Save & Continue“ weiter. Wenn ich ein Überspringen der Installation erlauben würde, kompromittiere ich meine Sicherheit. Außerdem werden die User dann wieder mit einem Dialog genervt. Wir sind aus dem digitalen Mittelalter heraus, Patches laufen zu 95% ohne eine Unterbrechung im Hintergrund.



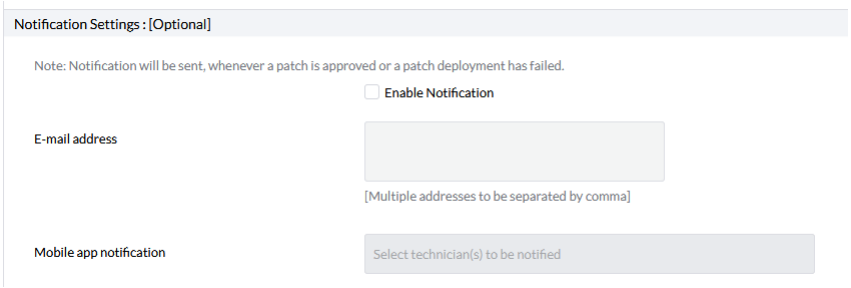
Schritt 4 und damit der Letzte bei der Definition der Deployment Policy, bietet mir die Möglichkeit einen Reboot oder Shutdown nach dem Deployment zu initiieren. Auch hiervon mache ich für den Fall der Key-User Pilotverteilung, keinen Gebrauch und gehe auf weiter mit „Save & Continue“. Bei Feature und Server Packs und dergleichen, erstelle ich eine manuelle Verteilung, welche dann auch mit einem Hinweis auf einen Reboot versehen ist. Ich halte diese Art der Patches aus dem täglichen automatischen Testverfahren heraus.

Nun erhalte ich noch einmal eine Übersicht und bestätige diese wieder mit „Save & Continue“.

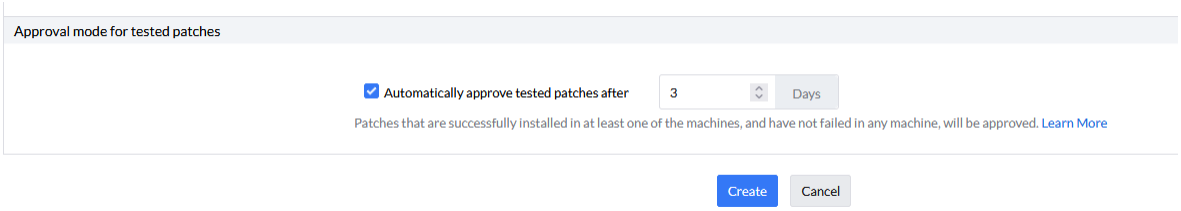


Zurück im Tab mit der Testgruppe welche wir erstellen, wähle ich die Policy aus.

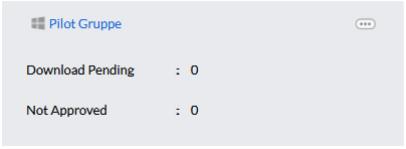
Bleibt die Frage nach Benachrichtigungen im nächsten Schritt. Ich persönlich halte von der E-Mail Flut nicht viel. Diese landen in einem Ordner, welcher irgendwann gelöscht wird. Ich baue auf die Dashboards, welche mir übersichtlich zu jeder Zeit einen Status geben. Dort kann ich ggfs. den fehlerhaften Patches nachgehen.



Letzter Schritt ist der Haken beim Approval Mode. Hier definiere ich drei Tage. **Mehr nicht** da ich ein weiteres Rollout der Patches nicht verzögern will. In den drei Tagen bis zur automatischen Freigabe der erfolgreich installierten Patches kann ich entweder über meine eigenen Programmaufrufe oder durch die Key-User in der Testgruppe eventuell ungewünschtes Programmverhalten identifizieren und diese Patches dann speziell ablehnen. Ein Patch sollte bei einer solchen Lösung nur dann automatisch freigegeben werden, wenn auf allen beteiligten Rechnern die Installation erfolgreich war. Bei EndpointCentral ist das gegeben. Wenn z.B. nur auf einem von 20 oder auch 100 Rechnern ein Patch mit einem Fehler antwortet, dann wird das Patch nicht freigegeben.



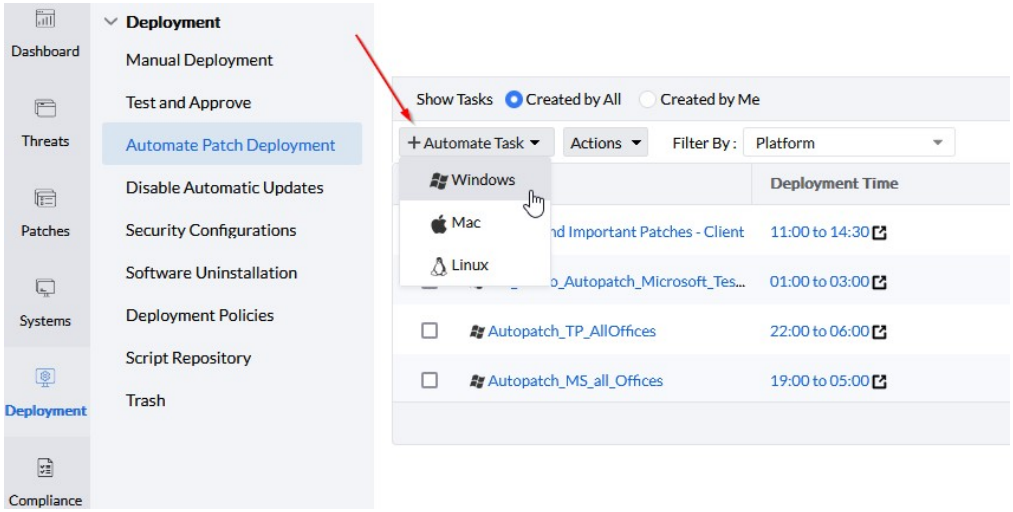
Ein Klick auf „Create“ erstellt mir dann meine Testgruppe.



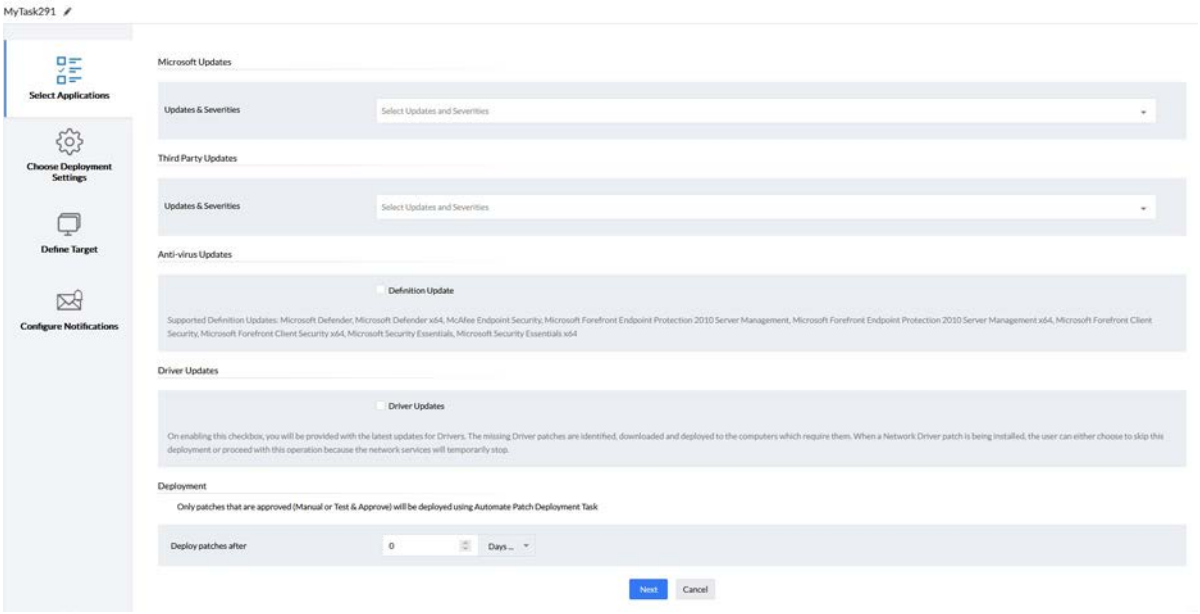
2.2 Automatische Verteilungen einrichten:

Nun, da ich ein Test und Freigabeverfahren eingerichtet habe, erstelle ich das erste automatische Deployment für die restlichen Endpoints im Unternehmen. Ich beginne mit den Critical and Important Patches.

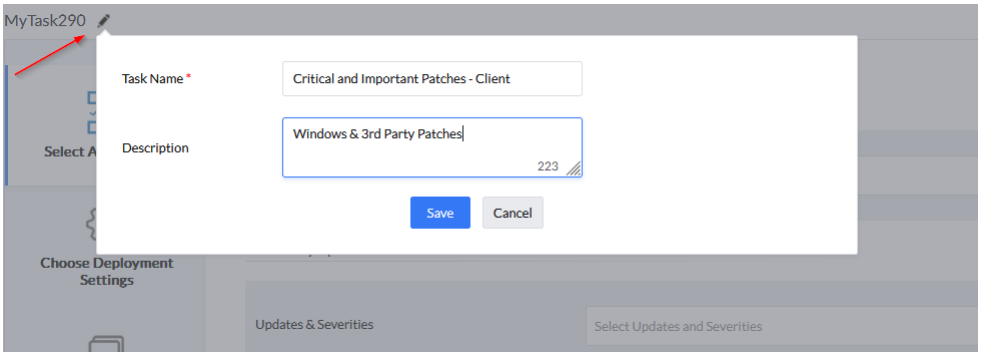
Hierzu in „Automate Patch Deployment“ wechseln und „Automate Task/Windows“.



Der Dialog ist ähnlich aufgebaut wie der aus der Testgruppe.



Wir geben dem Kind einen Namen und eine Beschreibung.

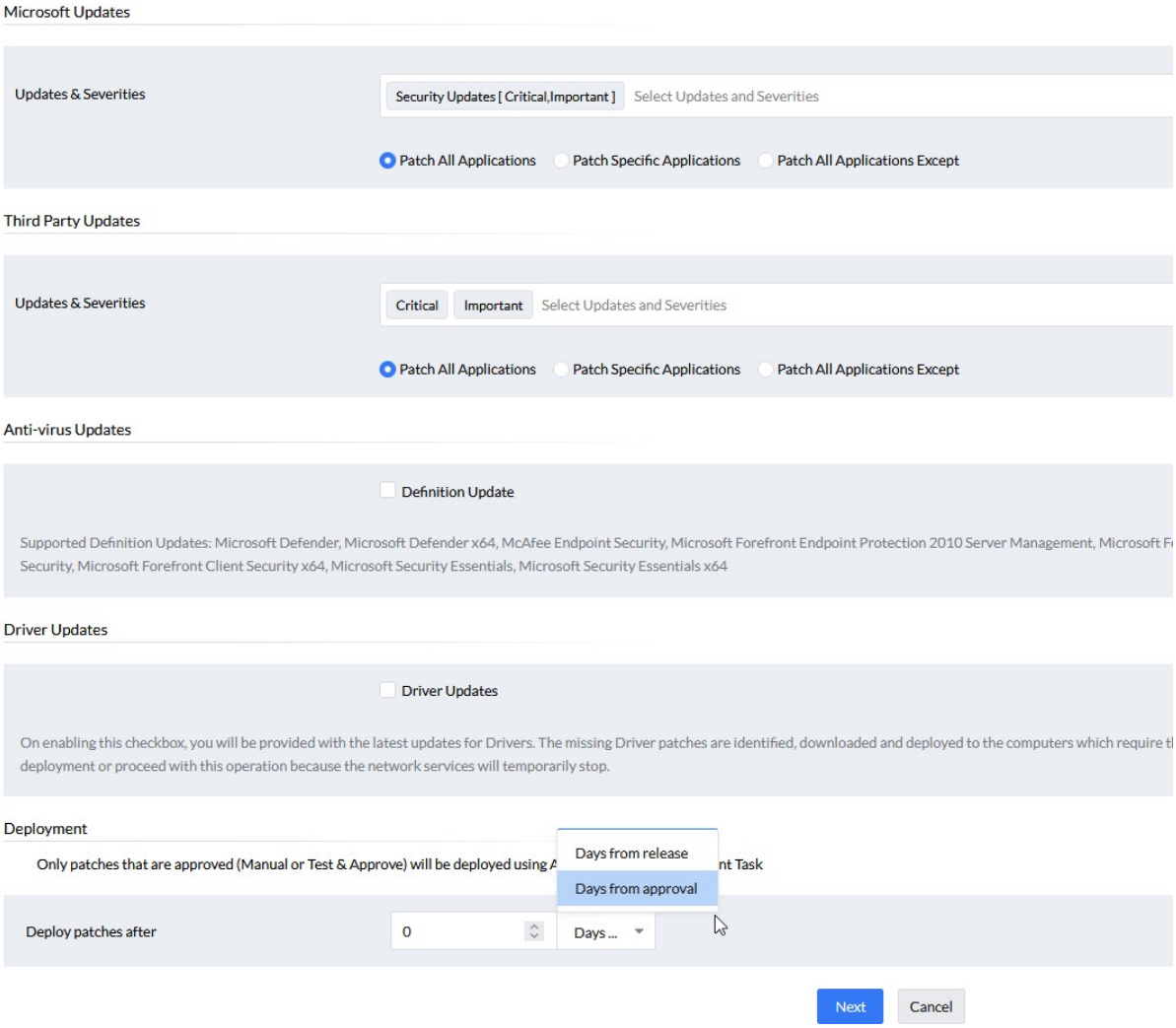


Bei den Microsoft und Third Party Patches wähle ich jeweils die Critcal und Important aus und definiere eventuelle Ausnahmen für Software, welche nicht automatisch gepatcht werden soll.

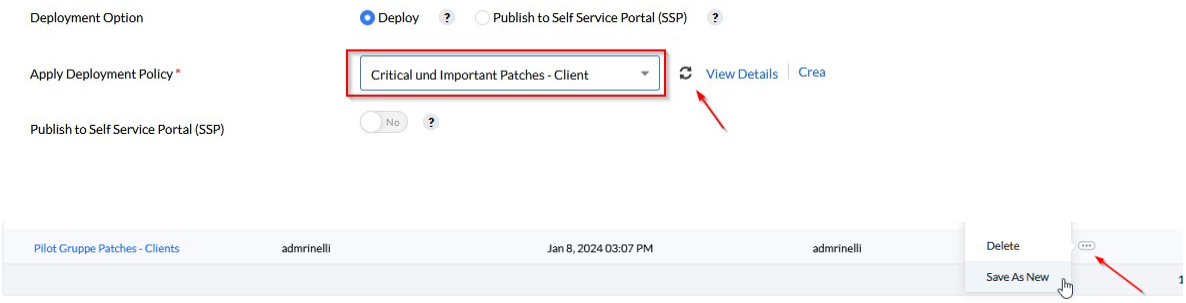
Anti-Virus Updates überlasse ich der Virensoftware. Meiner Überzeugung nach sollte dies das Einzige sein, welches die Clients von selbst herunterladen können dürfen. So kann ich sichergehen, das Virenpatterns aktuell sind – auch in dem Falle der Agent auf dem System nicht mit dem EndpointCentral-Server kommunizieren kann und damit keine Updates gepusht werden.

Treiber Updates verteile ich persönlich manuell oder in einem separaten Automatismus.

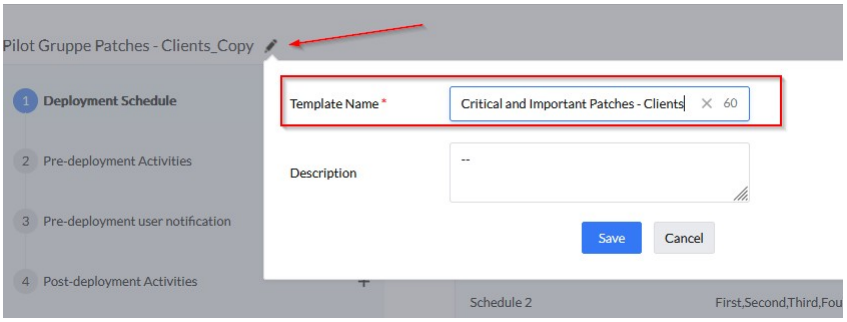
Letzter Punkt auf dieser Seite ist der Zeitpunkt, wann das Patch verteilt werden soll. Da ich eine Verteilung nicht weiter verzögern will – wir haben bereits drei Tage aus dem Testverfahren – gebe ich hier null Tage ab Freigabe an. Außerdem kann ich auf diese Weise auch gerade erschienene High Critical Patches manuell testen und Freigeben. Durch meine manuelle Freigabe werden diese dann im automatischen Deployment sofort berücksichtigt. Ich muss hierfür dann keine separate Konfiguration erstellen – wieder Zeit gespart.



Weiter mit Next und ich gelange auf die Deployment Settings. Hier wieder über Create eine Neue erstellen. Keine Angst … wir können die Deployment Policy, welche wir zuvor für die Pilot Gruppe erstellt haben, duplizieren. Hierzu einfach auf den Action Button der „Pilot Gruppe Patches – Clients“ Policy gehen und „Save As New“ klicken.



Wieder umbenennen und speichern.

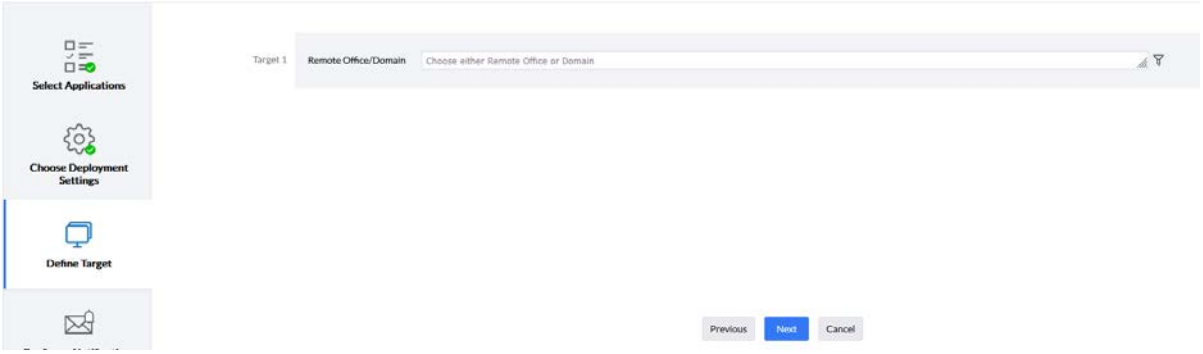


Anschließend jeden Dialog mit „Safe & Continue“ abspeichern. Wir behalten alle Zeiten und Einstellungen bei.

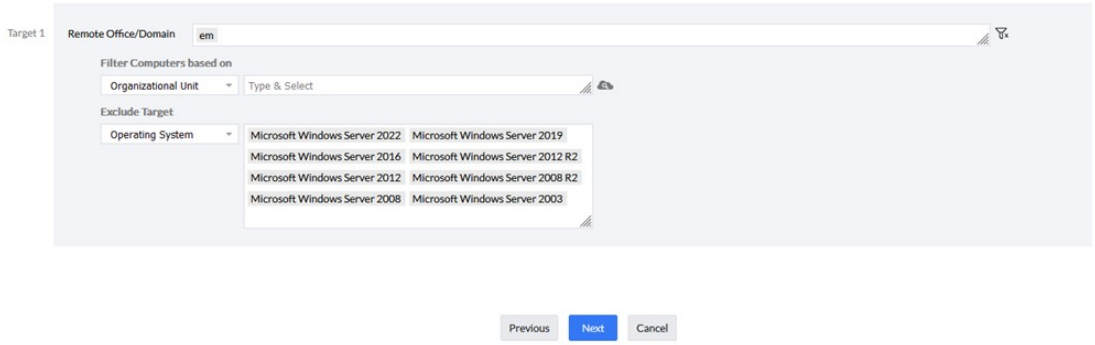
Zurück in das Tab mit der Automatischen Deployment Policy, können wir nach einem Klick auf den Refresh Button die eben erstellte Policy auswählen.

Patches können auch im Self-Service-Portal bereitgestellt werden. In meinem Fall belasse ich es beim normalen Deployment. Das Self-Service-Portal bietet sich vor allem dann an, wenn ich Feature oder Servicepacks verteilen möchte. Auch bei OS-Upgrades nutze ich das SB Portal und gebe den Mitarbeitern so die Möglichkeit ein zeitintensiveres Patchen, selbst zu planen.
Selbstverständlich gebe ich eine Deadline, zu welcher das Patchen dann erzwungen wird. Mir ist es wichtig, die User mit einzubinden.

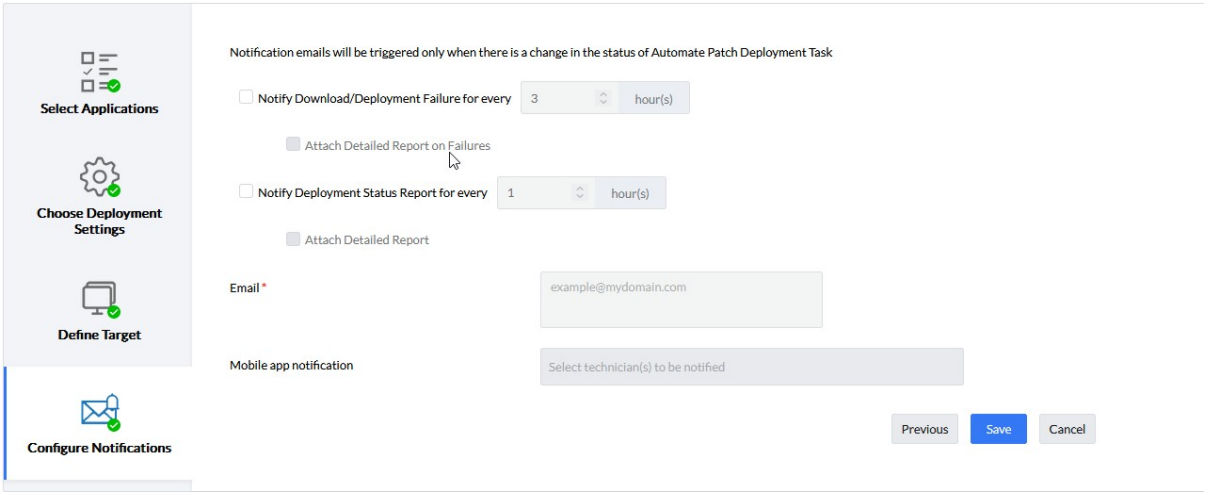
Über „Next“ gelange ich in die Ziele Definition.



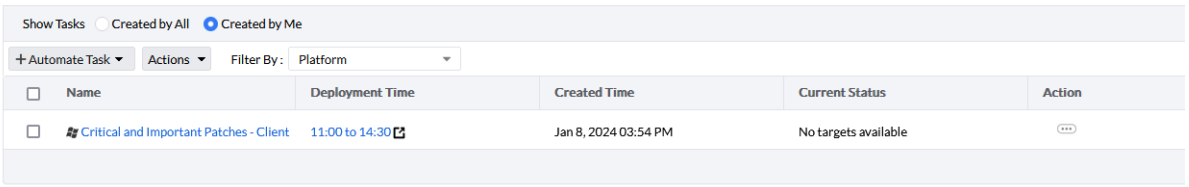
Über das Targeting wähle ich ein Remote Office oder die Domäne, filtere nach den Systemen und exkludiere bestimmte Gruppen wie z.B. die Server Betriebssysteme oder Custom Groups, in welchen ich Systeme zusammengefasst habe, welche nichts **automatisiert** erhalten dürfen.



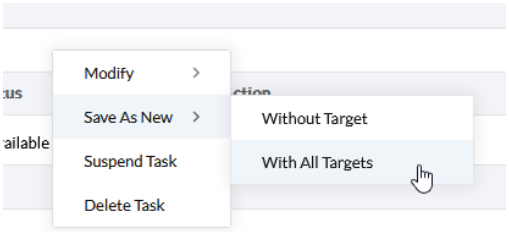
Beim Klick auf „Next“ gelange ich in den letzten Dialog mit den Benachrichtigungen. Auch hier sind E-Mail und Mobile-App Benachrichtigungen möglich. Ich selbst belasse es auch hier bei den Dash-Boards und klicke auf „Save“ ohne weitere Auswahl.



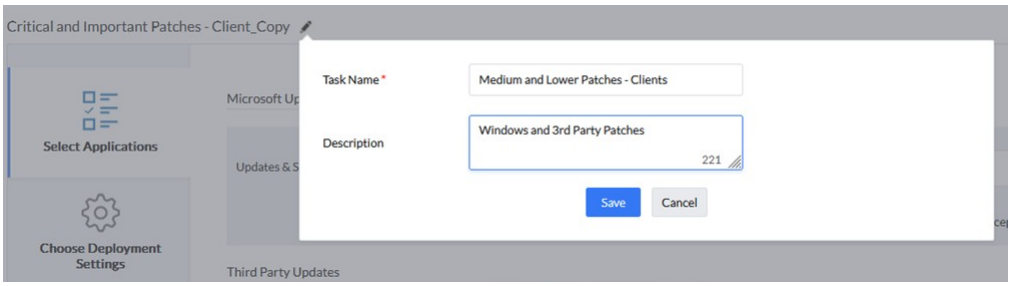
Trommelwirbel bitte … fertig ist unsere erste von zwei automatischen Verteilungen.



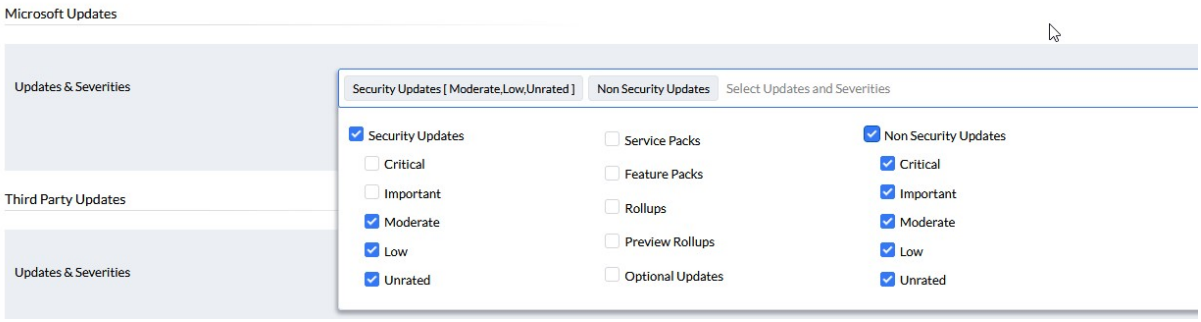
Nun dupliziere ich diese über den Action Button mit allen Zielen um auch die restlichen Patches von Medium abwärts, zu verteilen.



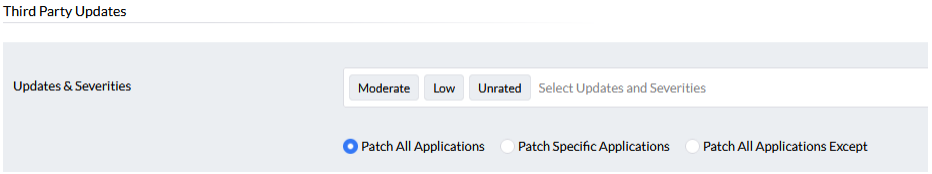
Kurz umbenennen.



Patches auswählen für Microsoft Updates. Optionale Updates verteile ich wie die Service und Feature Packs separat. Entweder in einem speziellen Automatismus oder über eine manuelle Konfiguration. Hier könnte sich ein monatlicher Zyklus anbieten.

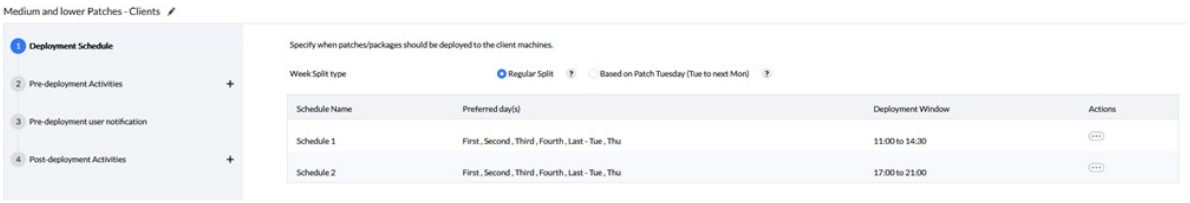


Gleiches gilt für die Third Party Patches.

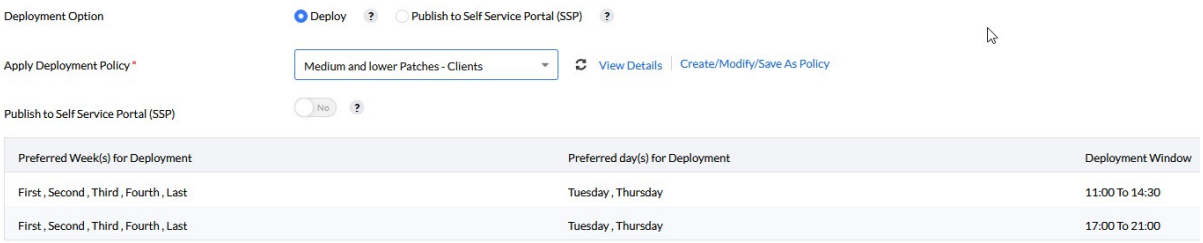


Nach dem Klick auf Next gelangen wir zu den Deployment Settings. Hier erstellen wir wieder eine Neue, da ich diese zum Zeitpunkt dieses Buches nur zwei Mal pro Woche verteile. Also wieder auf „Create/Modify/Save As Policy“ gehen.

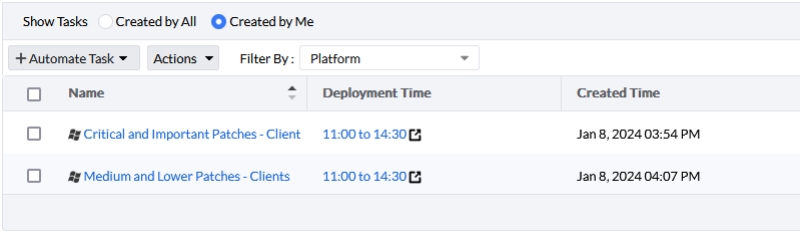
Ich dupliziere die Critical and Important Patches – Client Policy, benenne diese um und definiere zu denselben Uhrzeiten lediglich Dienstag und Donnerstag als Verteilungstage. Der Rest bleibt wie gehabt und ich klicke durch alle Dialoge mit „Save & Continue“.



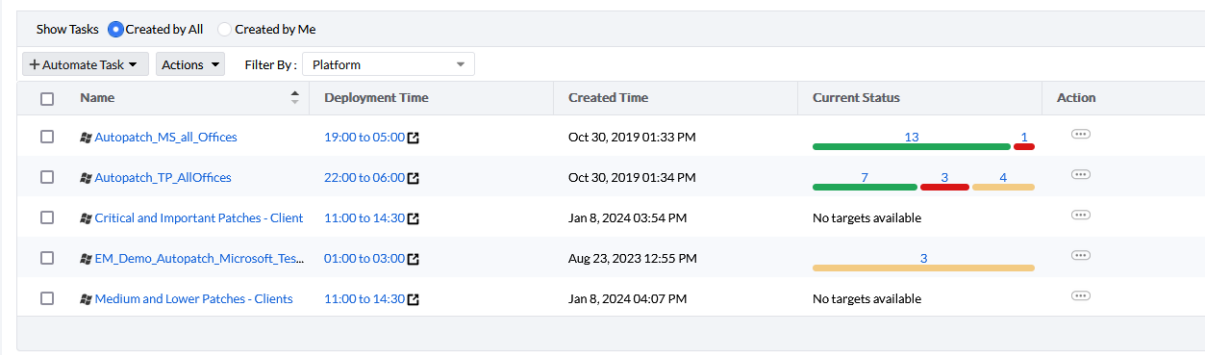
Zurück in unseren Automatismus Tab kann ich nach einem Refresh die neue Policy auswählen und bestätige alle weiteren Dialoge ohne Änderungen.



Im Anschluss habe ich meine zwei Automatismen, welche mir alle freigegebenen Patches im Unternehmen ausrollen.



Über die Statusanzeigen kann ich den Verlauf kontrollieren und ggfs. den fehlerhaften Verteilungen nachgehen.



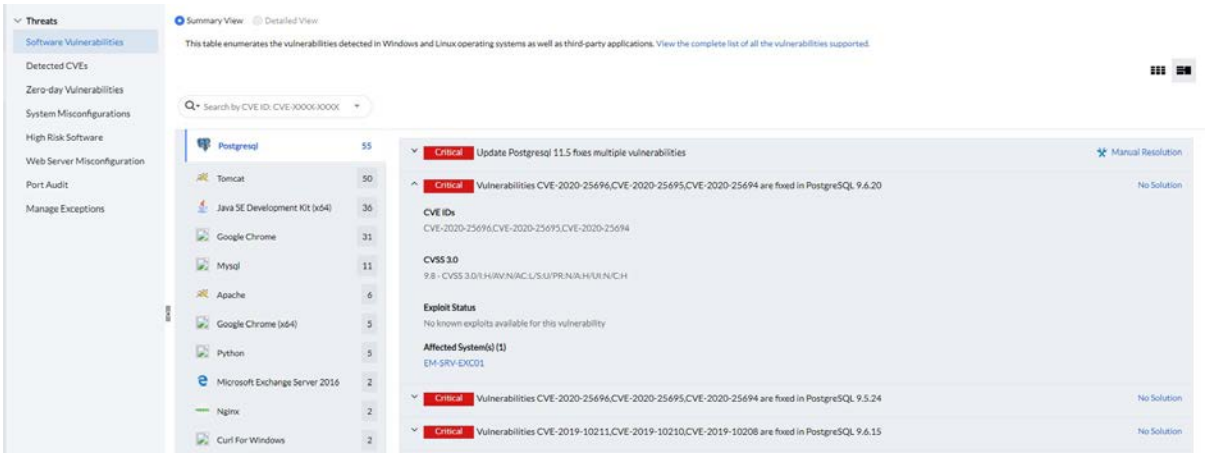
Noch ein Tipp! Wenn ich in Kundenumgebungen komme, in welchen es bisher die Speedy Gonzales Turnschuh-Administration gab, sehe ich mir zuerst die Menge an fehlenden Patches in der Übersicht an. Sind es zu viele, erstelle ich zwar beide Automatismen wie oben beschrieben, suspendiere jedoch die für Medium and Lower Patches. Ich lasse die ersten zwei Wochen **allein** die Critical and Important Patches ausrollen. Auf diese Art werden die Systeme und mein Netzwerk nicht überstrapaziert. Ich lasse sozusagen etwas Druck ab.

2.3 Überprüfung weiterer Schwachstellen im Thread Management:

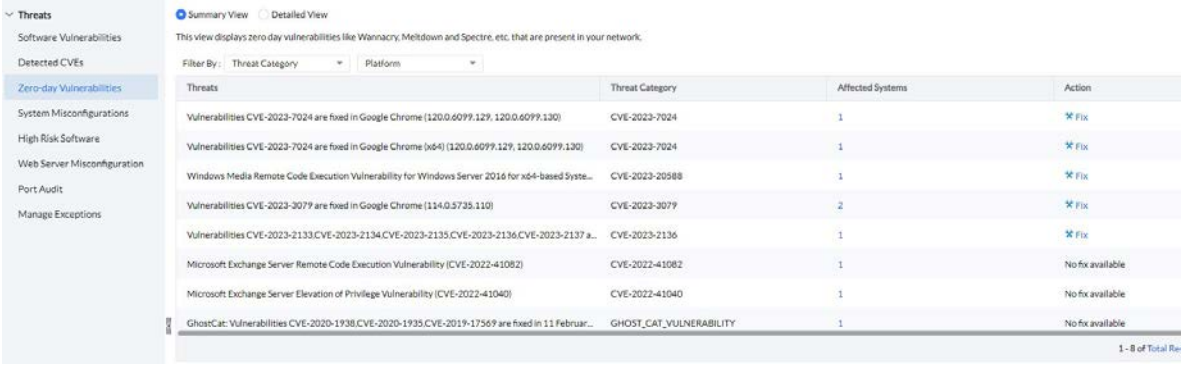
Es gibt neben der normalen Patches auch eine Menge an weiteren Schwachstellen, um welche ich mich kümmern darf. Misskonfigurationen, Zero-Day Schwachstellen, offene Ports, Web-Server Misskonfigurationen, etc.

In der EndpointCentral Security Edition sowie im VulnerabilityManager Plus, ist hierzu ein zusätzlicher Bereich „Threats“ geschaffen worden.

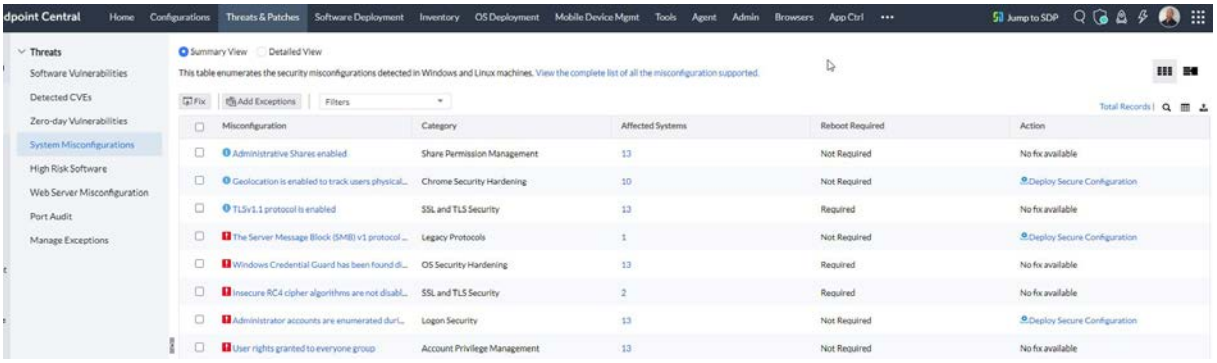
Hier finde ich noch einmal in Kategorien einteilbar eine Übersicht über meine noch offenen Schwachstellen inclusive ausführlicher Beschreibung und CVE Scores. Kann hier auch Ausnahmen definieren und/oder die Schwachstellen beheben.

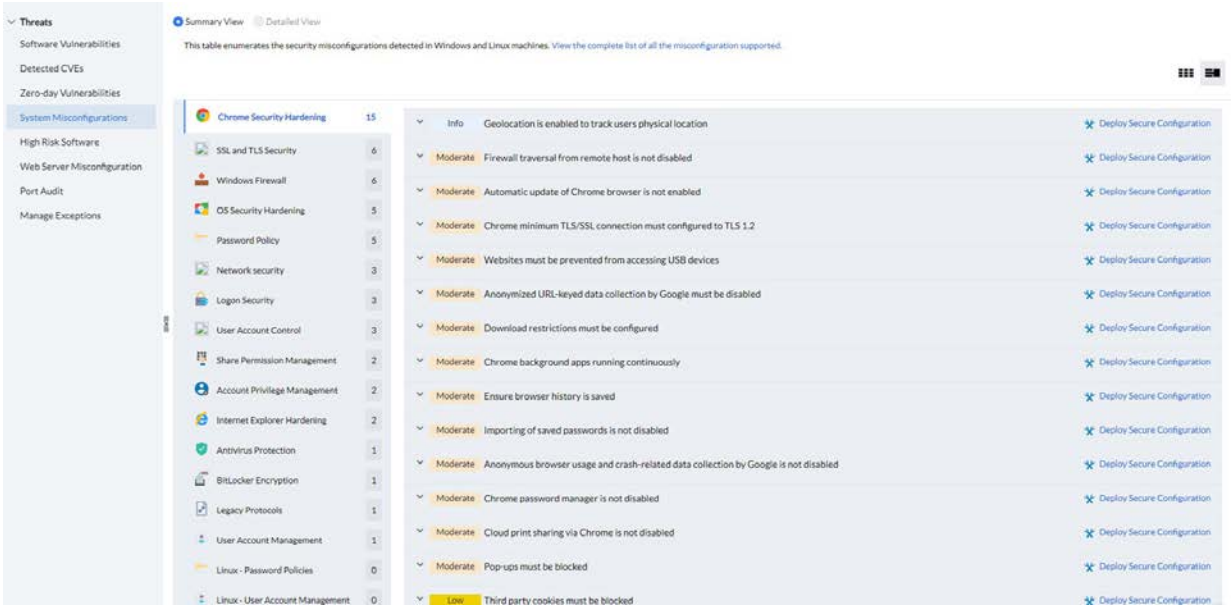


Es gibt auch einen Bereich, welcher Zero-Day Schwachstellen aufzeigt, und ich kann entsprechend handeln.

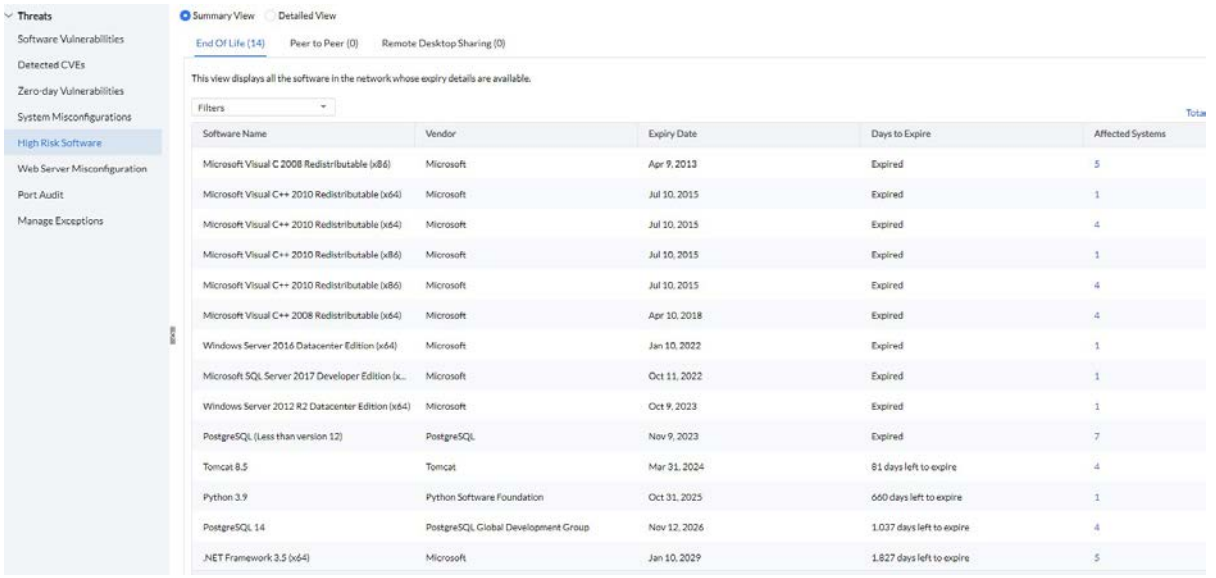


Im Bereich „System Misconfigurations“ schließe ich, was mir möglich ist oder definiere Ausnahmen. Natürlich kann vieles davon über die Active Directory konfiguriert werden. Ich kann die Ergebnisse dieser Überprüfung nutzen und ggfs. direkt in der AD konfigurieren.

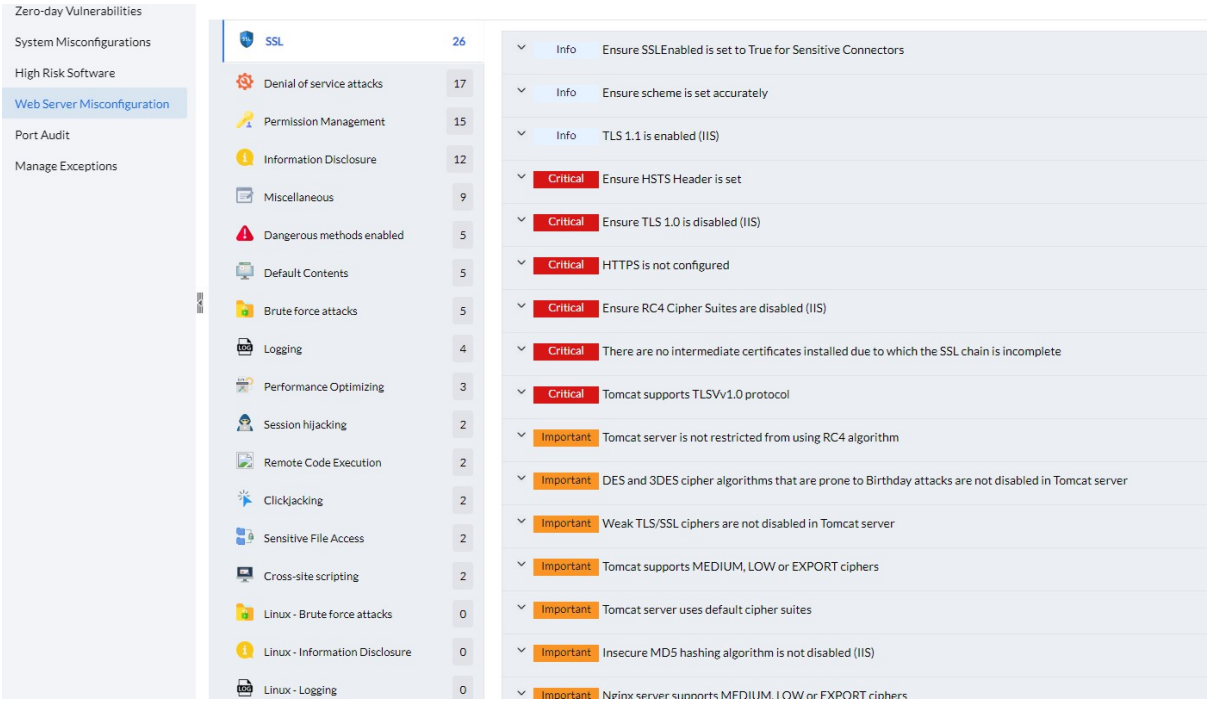




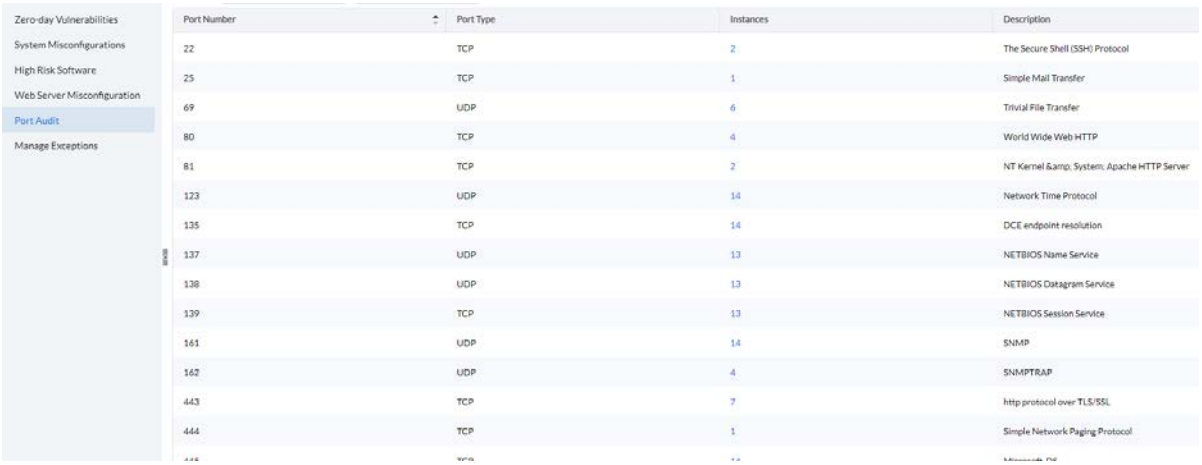
Dann sehe ich nach ob es abgelaufene Software unter „High Risk Software“ gibt bzw. wann vorhandene Software ihr Laufzeitende erreicht. Ein Thema, welches oft vernachlässigt wird und „urplötzlich“ auftaucht.



Weitere Schwachstellen betreffen die Webserver. In diesem Bereich bekomme ich eine Übersicht mit Lösungsvorschlägen, mit deren Hilfe ich Schwachstellen auf den Webservern beheben kann.

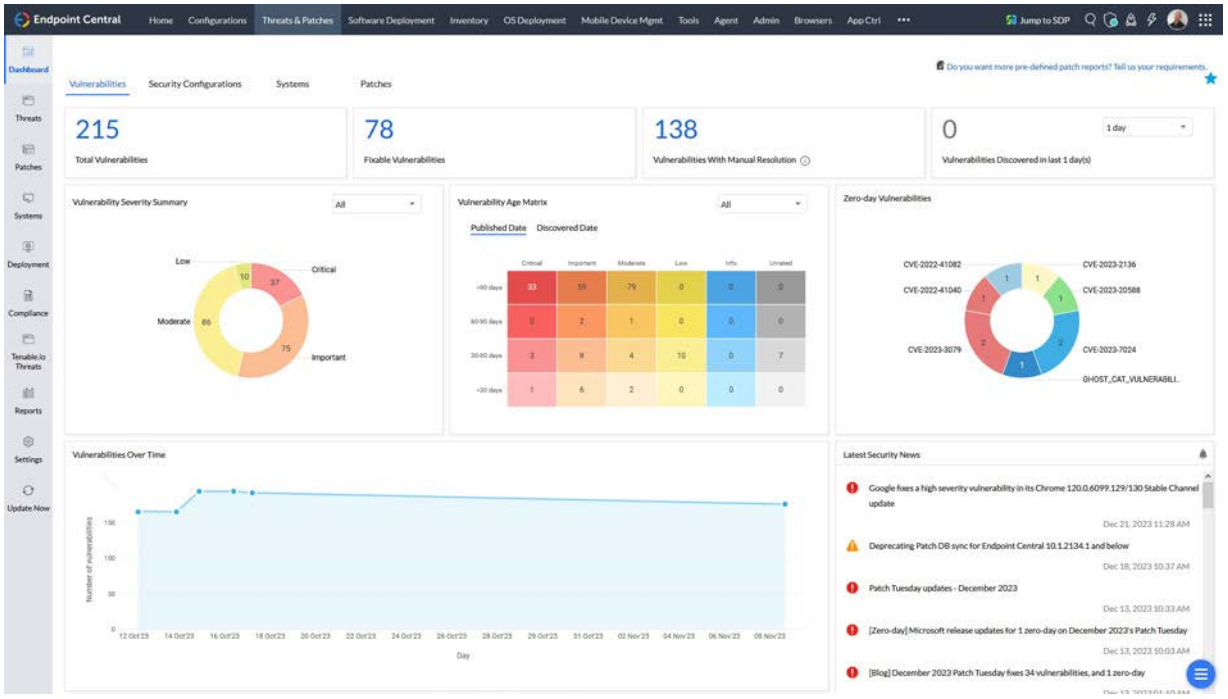


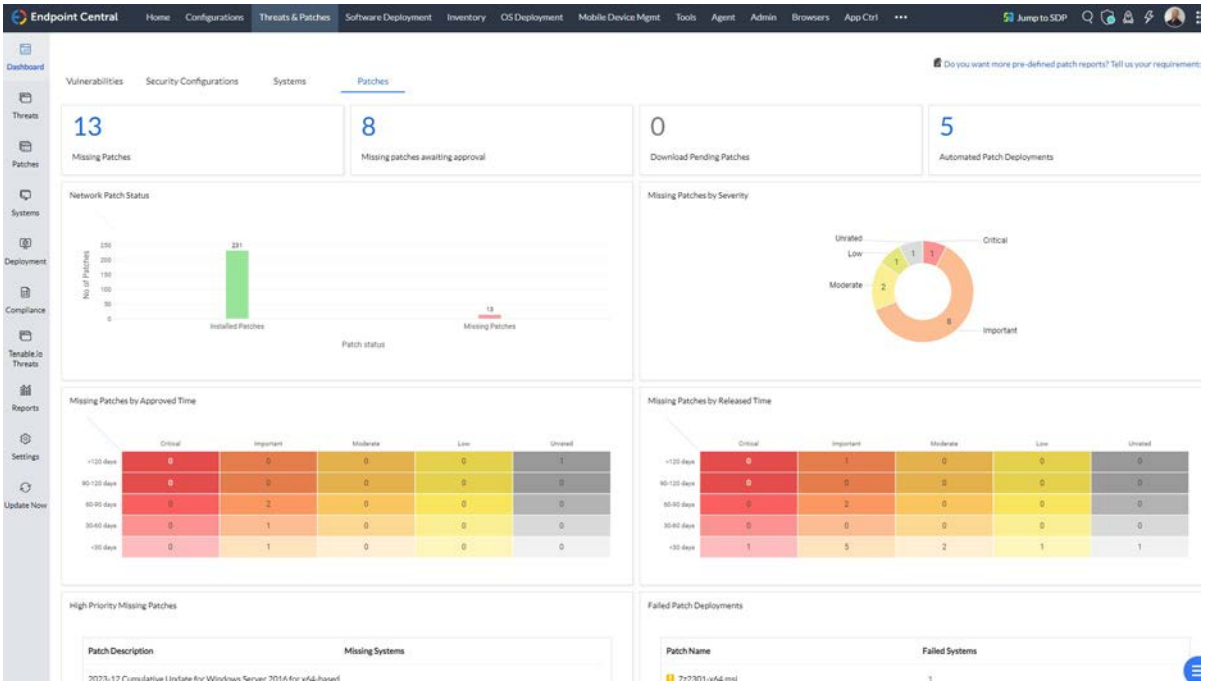
Und zum Schluss sehe ich mir an, welche Ports auf welchen Systemen offen sind.



2.4 Zeit für den Cappuccino – Dashboards prüfen:

Anstelle mich durch eine Flut an Benachrichtigungs- und Status E-Mails zu wühlen, gehe ich in die Dashboards. Das Threats & Patches Dashboard liefert mir alle nötigen Informationen auf einen Blick und ich wechsle durch das Dashboard in Bereiche welche meiner Aufmerksamkeit bedürfen. Ein übersichtliches Dashboard mit direkten Verlinkungen ist ein MUSS für mich.





Einmal eine saubere Patchstrategie definiert brauche ich mich nur noch um die noch fehlenden nicht freigegebenen Patches kümmern und ggfs. ein Troubleshooting, wenn auf einem System ein Patch nicht erfolgreich installiert werden konnte.

Zugegeben, es ist ein initialer Aufwand eine Strategie zu konfigurieren. Einmal eingerichtet jedoch, reduziert es meinen bisherigen manuellen Aufwand erheblich und bietet maximalen Schutz der Endpoints. Was mir an der Lösung von ManageEngine so gefällt ist, das ich übersichtlich und einfach Patchautomatismen anlegen kann – ohne Raketenwissenschaften studiert zu haben. Auch brauche ich mit guten Dashboards und Reports keine Glaskugel zu Rate zu ziehen und orakeln lassen wie sicher meine Clients sind.

Das ist meine Strategie beim Endpoint Patching. Durch diese schaffe ich mir freie Zeit, welche ich nutzen kann um die weiteren Schwachstellen wie USB Devices, Applikationen, Browser, Bitlocker, etc. anzugehen. Auch hierzu gibt es viele Lösungen auf dem Markt und natürlich auch in meiner bevorzugten Managementlösung EndpointCentral.

Gerade in einer Zeit in welcher ich es schwer habe neue Kollegen in der IT zu finden, benötige ich eine Lösung, welche mir Freiräume schafft. Mit EndpointCentral habe ich solch eine Lösung, bei welcher ich außerdem einen exzellenten Support von ManageEngine und dem deutschen Partner MicroNova AG, zur Verfügung habe.

Also worauf noch warten?

Viel Spaß und haltet Eure Endpoints Up-To-Date!

Jürgen Rinelli

3. About the Author:

MCITP, MCTS, MCP, MOS, Enterprise Administrator, Senior Software Consultant, SCCM-Spezialist, Autor, Coach, Reiki-Lehrer ...

Jürgen Rinelli wurde 1970 in Deutschland geboren. In seinem ereignisreichen und oft abenteuerlichen Leben hat er in vielen Ländern gelebt und gearbeitet. Ob als Geschäftsmann, Manager, Mechaniker, Trainer, Taucher oder IT-Experte, er findet immer einen Weg, seine Träume zu verfolgen.