Patch management is no longer rocket science

My experience and easy steps

Jürgen Rinelli

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

Table of contents

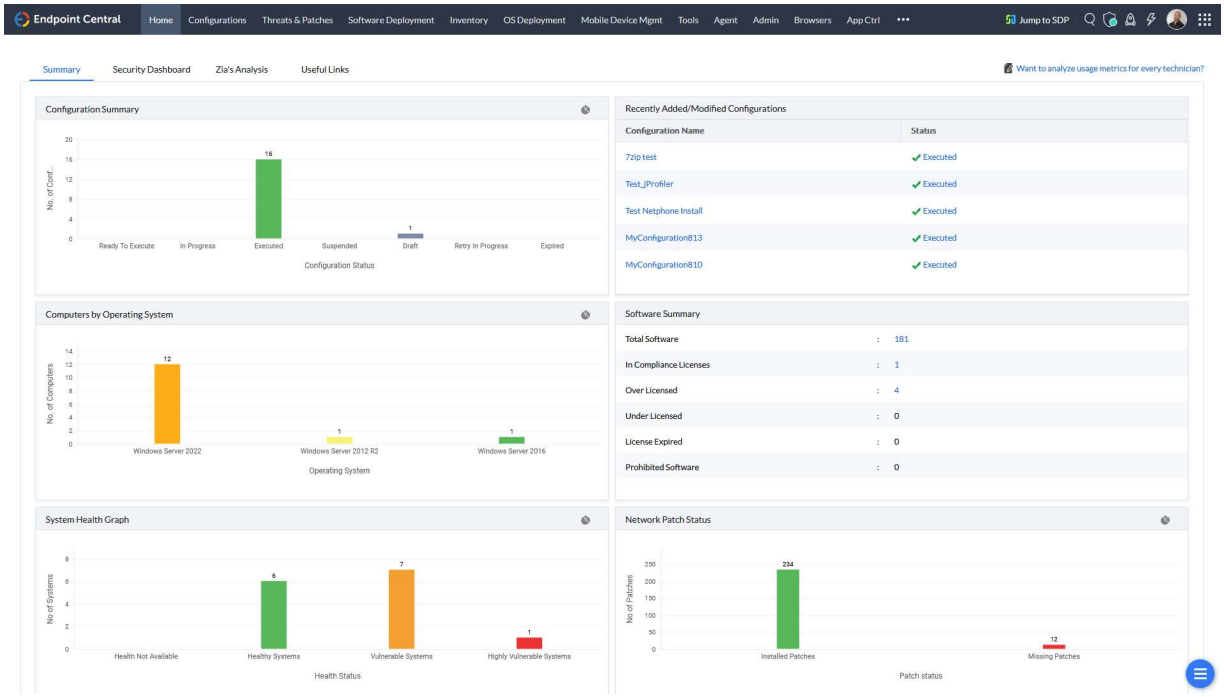
1. A few words

As a senior consultant, I regularly see companies that have been hacked and help them rebuild. The reason is usually an inadequate or non-existent endpoint management strategy. But - what used to involve a lot of effort is no longer rocket science.

With the multitude of cyberthreats and the use of AI technologies by attackers, attacks are becoming smarter, faster and more adaptable.

In this short e-book, I share my best practices for endpoint patch management and show how easy it is to set up a comprehensive patch strategy. My personal favourite solution and foundation on which the screenshots and settings are based is from ManageEngine.

I use and recommend the EndpointCentral Security Edition, although my best practices for pure patch management in this mini e-book can also be used with VulnerabilityManager Plus.



"Patch management" and "endpoint security" are terms that can no longer be separated. I am not telling anything new when I say that it is the end devices that are most vulnerable to cyberattacks. All end devices – where I am including even users!

While I can - and should - only carry out "patch management" for users through consistent training with fake emails, links, etc., hardware can be patched with much less resistance and effort.

What is important for me and what steps do I take in patch management?

1. set up an automatic test and approve procedure
2. set up two automatic distributions of the released patches
3. check which vulnerabilities are still open in threat management and fix them
4. regularly check the dashboards over a good cup of cappuccino

After this first simple step of patch automation through automatic testing and subsequent automatic distribution, the basic security of the devices is in place. I have already mentioned that this does not completely secure the endpoints. It is also important to take biological endpoints (users) by the hand. I address this point in a separate e-book with the topics of device control, application control, data loss prevention, anti-ransomware and, very importantly - browser security.

Shall we get started?

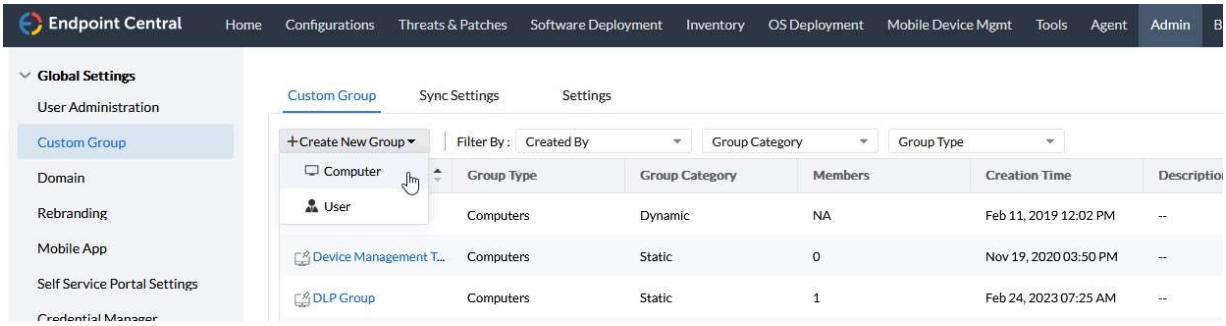
2. Step by Step

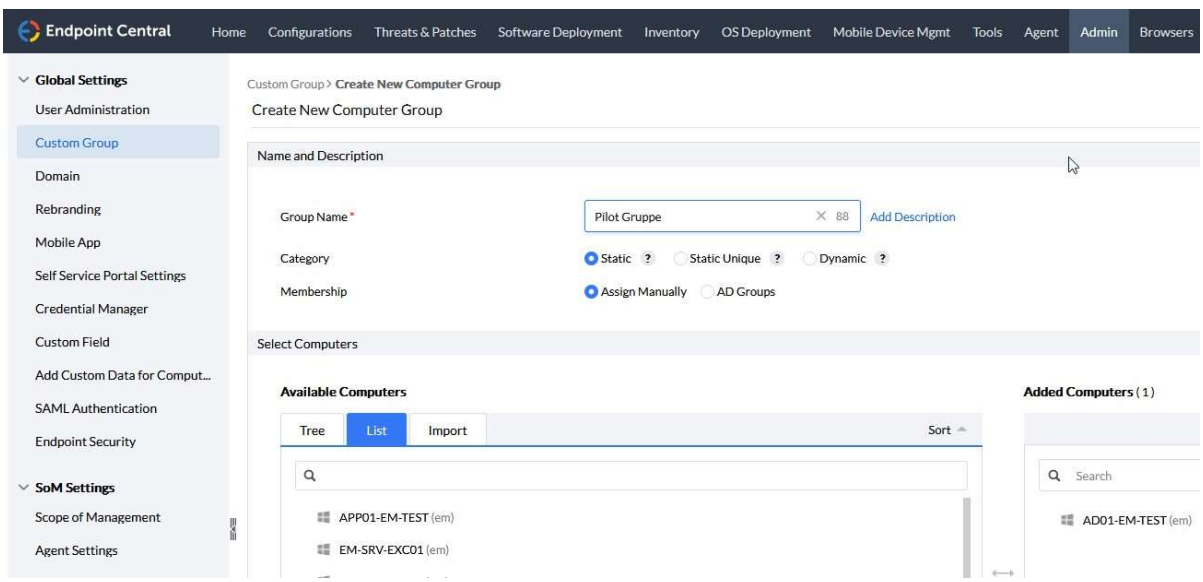
2.1 Automatic test and approve procedure:

Since a patch can also lead to undesired behaviour, I rarely release patches unchecked. Yes - it would be better to test everything!
To make my work easier, I outsource the testing. To do this, I first create a test group with my own test devices as well as devices from key users. In other words, users from each department who work with the software on a daily basis. I am often unable to perform actions in the software for license and authorization reasons. Often I cannot install the software on my test devices at all due to a lack of license. So I need users with this software who also execute actions in it. This is the only way to ensure that a link to a reader application, for example, works.
For the key users, I mainly include the users who have already frequently reported errors or slow systems. - I want feedback and I want it as quickly as possible!

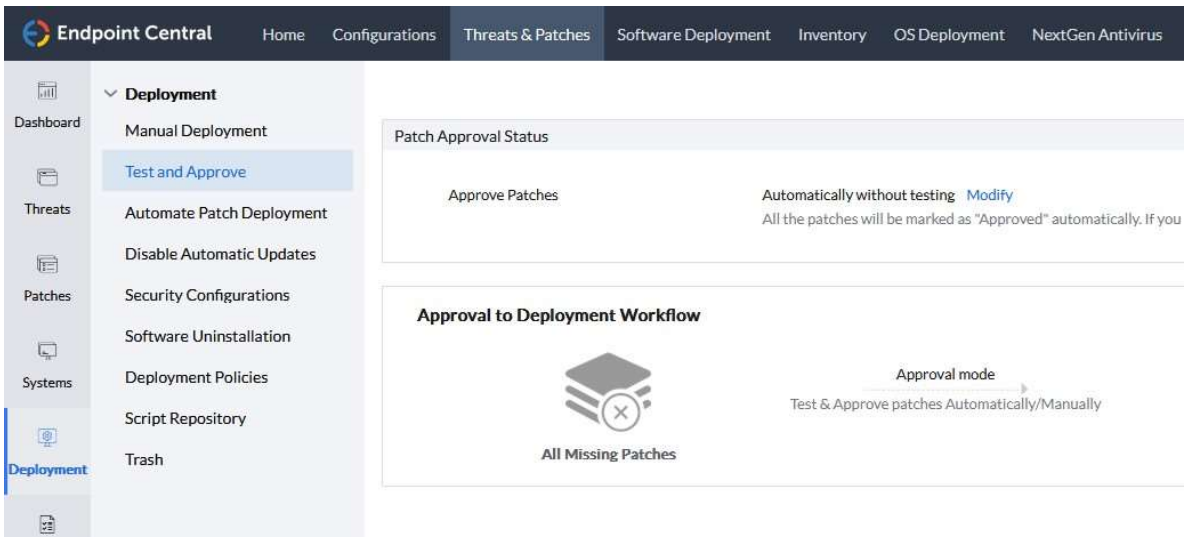
As I mentioned at the beginning, I will show you the setup steps using my favourite "EndpointCentral UEMS" from ManageEngine as an example, because I have never found it easier and more intuitive in any other endpoint management solution on the market - and I have worked with many over the last 25 years.

First I create a static computer group via Admin/Custom Group and put all test systems and key user systems into it.

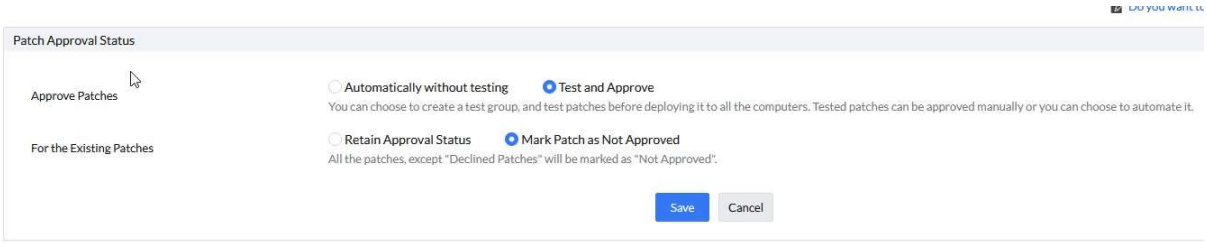




Continue in Threads & Patches/Deploy/Test and Approve to set up an automatic test and approval procedure.



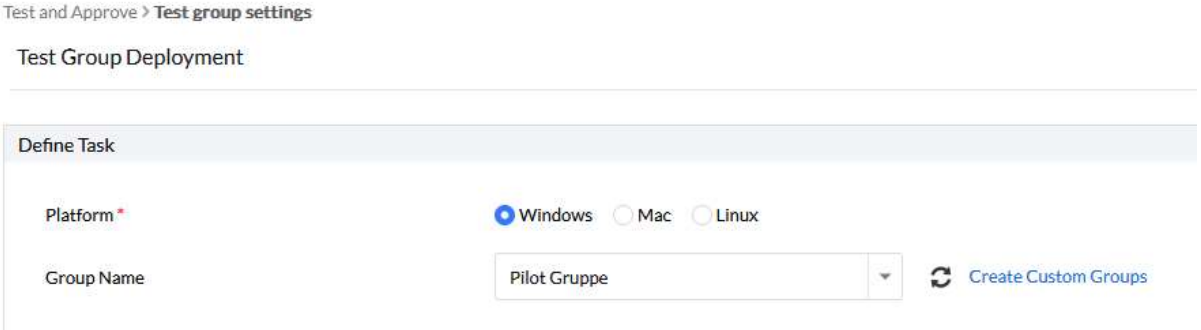
There I set the patch approval status - if not already done - via the modify button from "Automatically without testing" to "Test and Approve" and "Mark Patch as not Approved" to set the already automatically approved patches to not approved, just to be on the safe side.



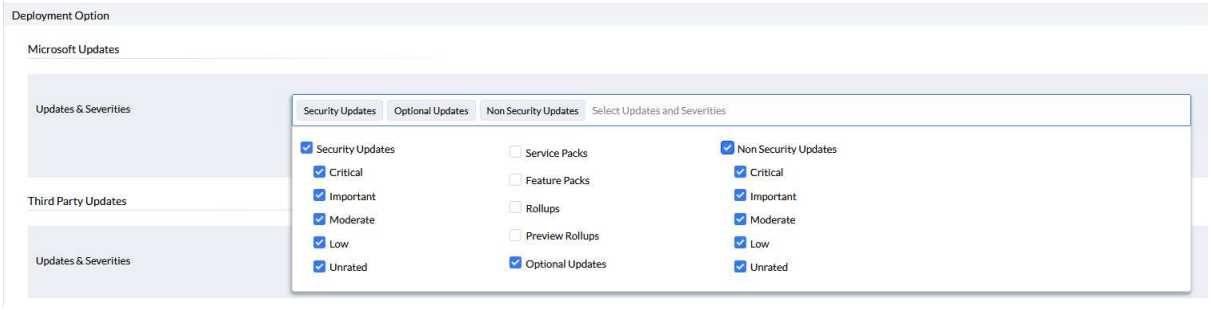
After saving the settings, I can create my first test group via "AddGroup". You can create different ones for different scenarios. I always separate clients and servers from each other. Yes - servers can also be patched automatically to a certain extent!

In the first step, I select the platform: Windows, Mac or Linux

Then the previously created group is required - "Pilot Group" in this example.

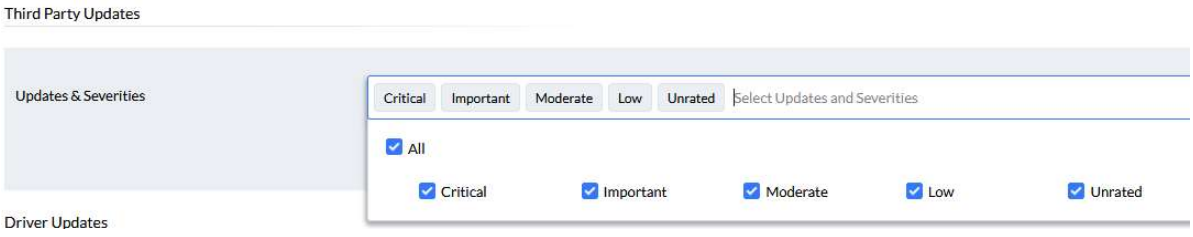


In the "Deployment Options" I select Microsoft Updates as follows:

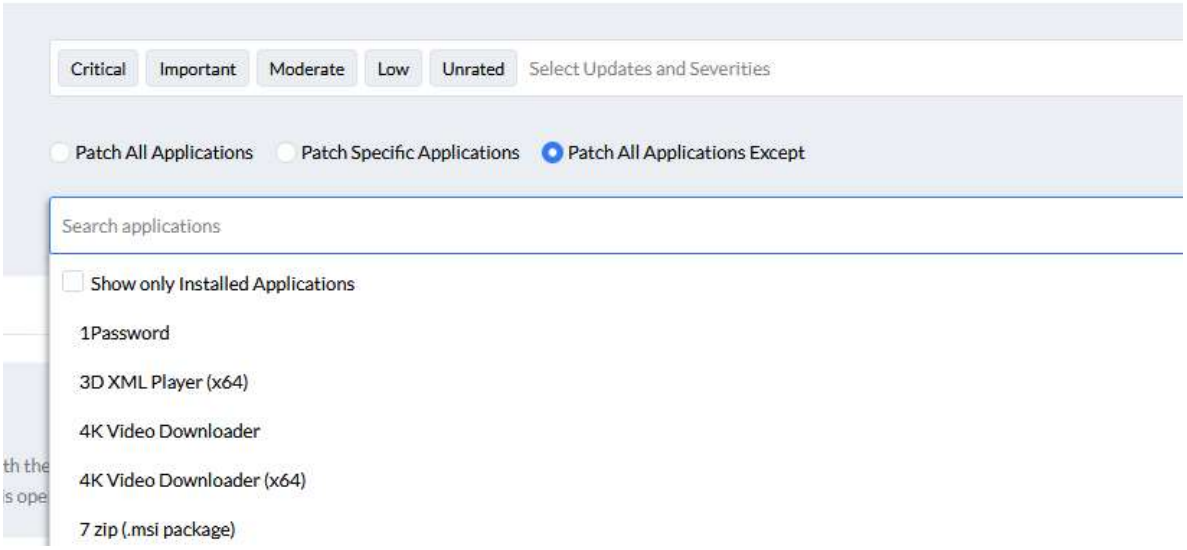


I continue to test service packs, feature packs and such, manually or create a special test group for them.

For third-party updates, I define everything.



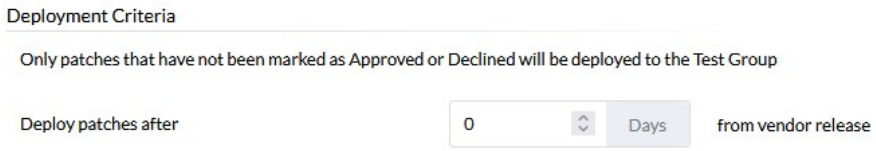
When I go out of the selection dialogs, I can also exclude special applications from patching for Microsoft updates and third-party updates. E.g. special VPN solutions, workspace applications, SQL, Java, etc.



I personally test driver updates manually or test them via a special group of devices in a further automated process.

With "Deployment Criteria" I can define a delay, i.e. a time from the patch release at which the patch should be rolled out.
I leave this at 0 days, as I am deploying to a test group and I don't want to delay the release of the patch any longer.

Yes - even if it means that a system may fail at some point! Then I simply re-install the system, but don't compromise my overall security.

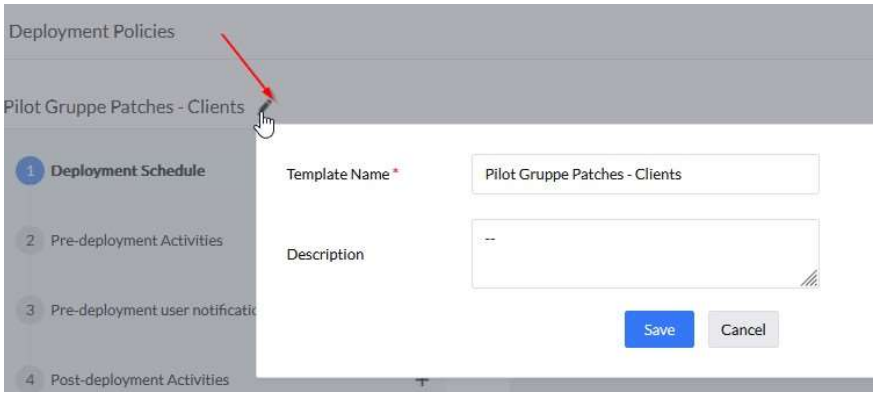


In the next step, I select the deployment policy - the "when and how" of the deployment.

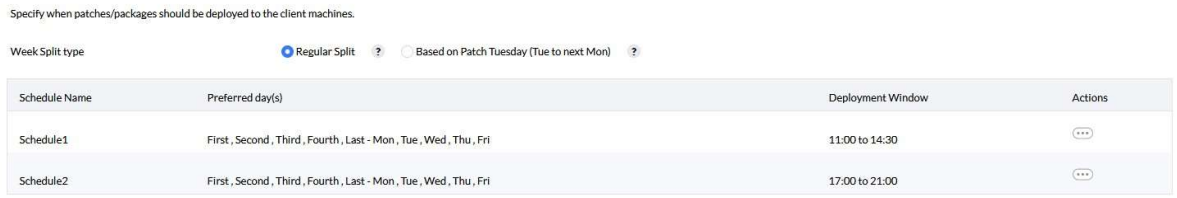
Here I create my own policy, to be able to customize the deployment window independently.

So I click on "Create/Modify/Save As Policy", which takes me to a new tab for the deployment policies.
Click on "Create Policy".

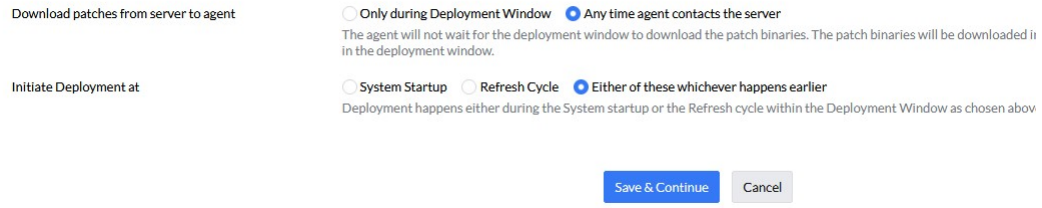
Give it a meaningful name.



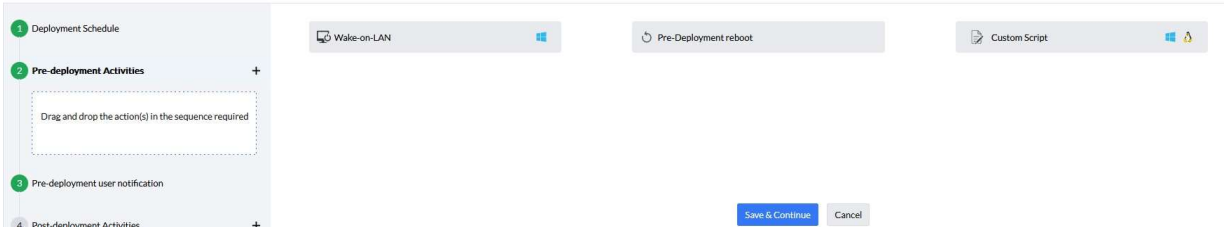
In step 1, I specify the time window. I prefer daily weekdays (if the IT Dept. is manned at the weekend, then also Saturday and Sunday), as well as two time windows to respond to more flexible working hours. You could now ask why not patch 24/7 or from 07:00 - 21:00. My thought with a fixed patch time is that I don't want to stress the friendly key users all day long. I can give the friendly helpers a fixed time during which the patches are installed and they can adjust to this. Lunchtime is also the best time for it.



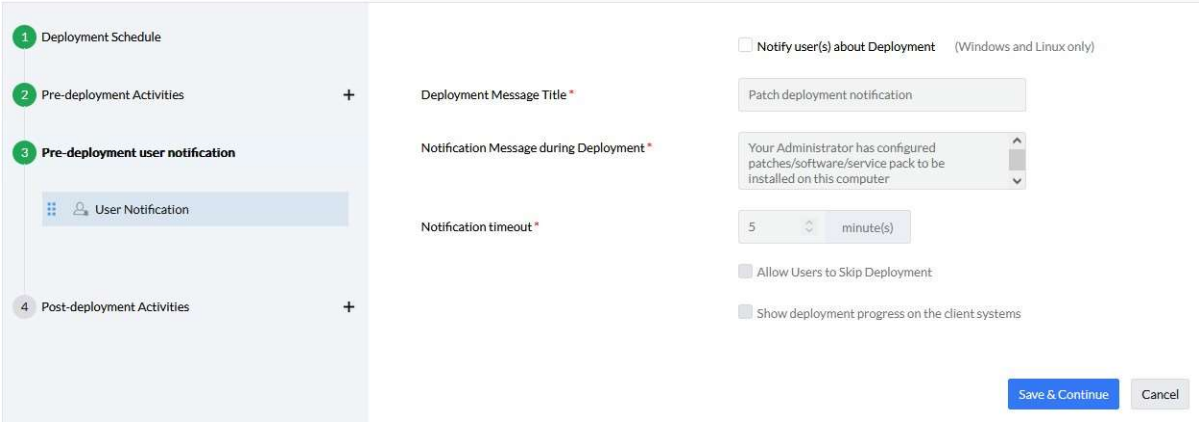
I want to have patches already cashed when they are ready to be distributed at any time within the defined time window. This saves bandwidth and time.



After "Save & Continue" I can define pre-deployment activities. Wake-on-Lan if I want to carry out a deployment at night, Pre-Deployment reboot or run a custom script. In my example, I don't define anything and go straight to "Save & Continue".

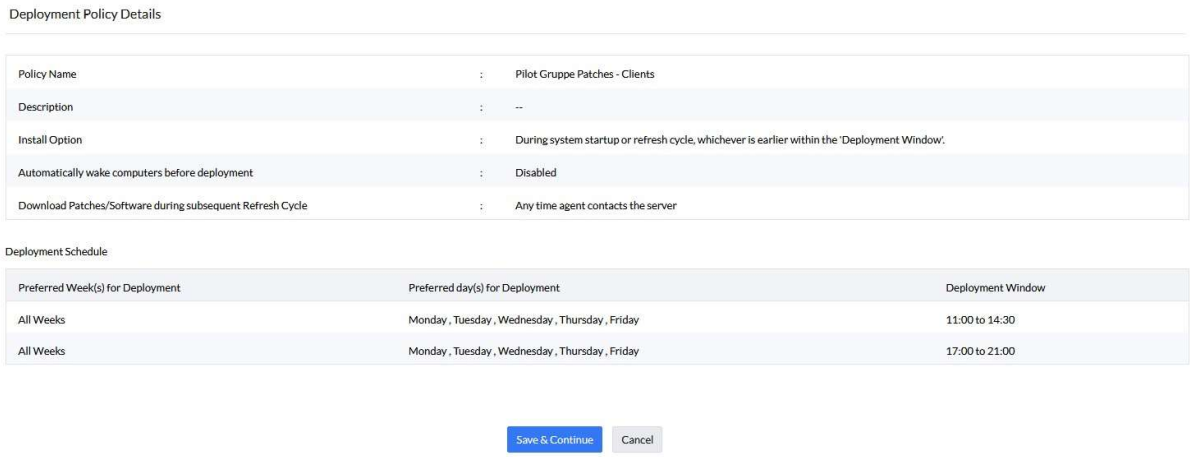


Step 3 gives me the option of notifying users prior distribution and allowing them to skip. Again, I do not define anything and continue with "Save & Continue". If I were to allow the installation to be skipped, I would compromise my security. In addition, users are then annoyed again with a dialog. We are out of the digital middle ages, patches run 95% of the time without interruption in the background!



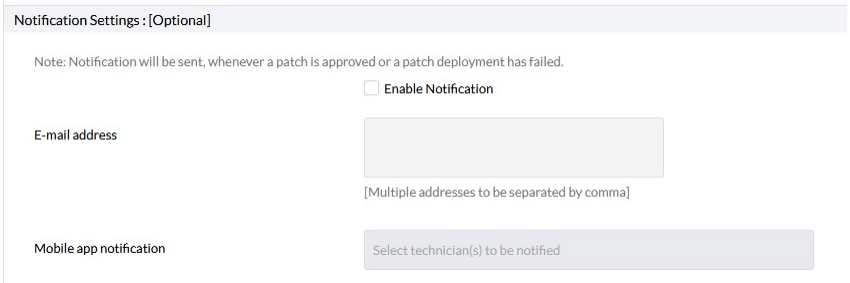
Step 4, which is the last step in defining the deployment policy, gives me the option of initiating a reboot or shut down after deployment. In the case of the key user pilot deployment, I do not make use of this either and continue with "Save & Continue". For feature and service packs and the like, I create a manual deployment which is then also provided with a note about a reboot. I keep these types of patches out of the daily automatic test procedure.

Now I get another overview and confirm it again with "Save & Continue".

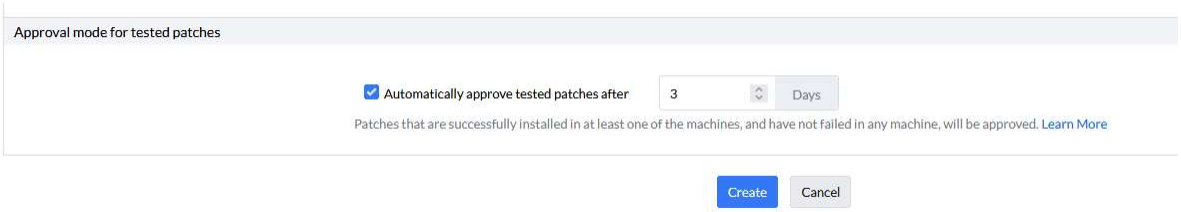


Back in the tab with the test group we started creating, I select the policy.

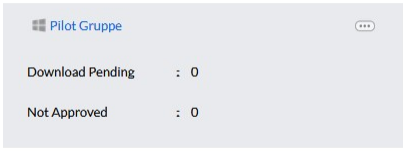
That leaves the question of notifications in the next step. Personally, I don't think much of the flood of e-mails. They end up in a folder which is deleted at some point. I rely on the dashboards which give me a clear status at all times. I can follow up on the faulty patches there if necessary.



The last step is to check the approval mode. I define three days here. **No more than that** as I don't want to delay a further rollout of the patches. In these three days after the successfully installed patches are automatically released, I can identify any unwanted program behaviour either via my own program calls or via the key users in the test group and then specifically reject these patches. With such a solution, a patch should only be released automatically if the installation was successful on all participating computers. This is the case with EndpointCentral. If, for example, a patch responds with an error on only one of 20 or even 100 computers, then the patch is not approved.



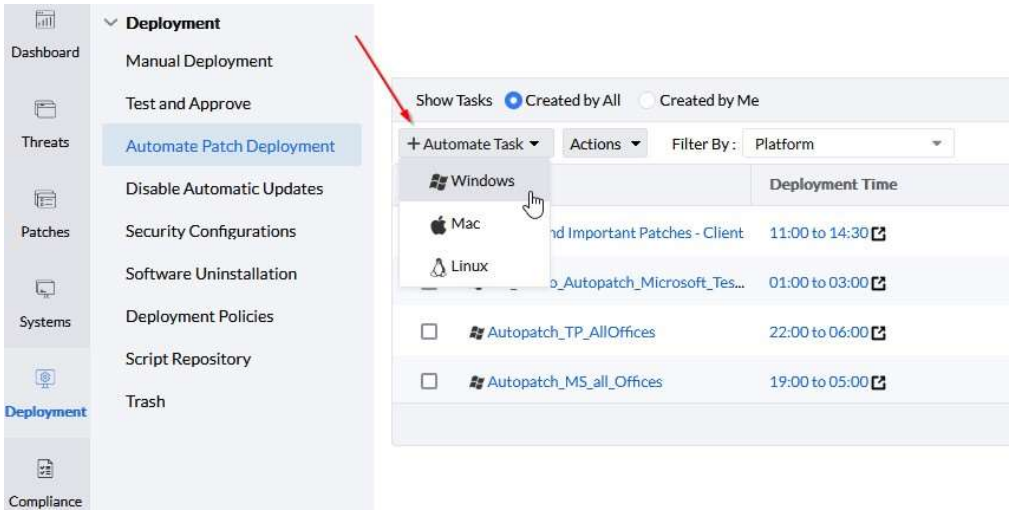
Click on "Create" to create the test group.



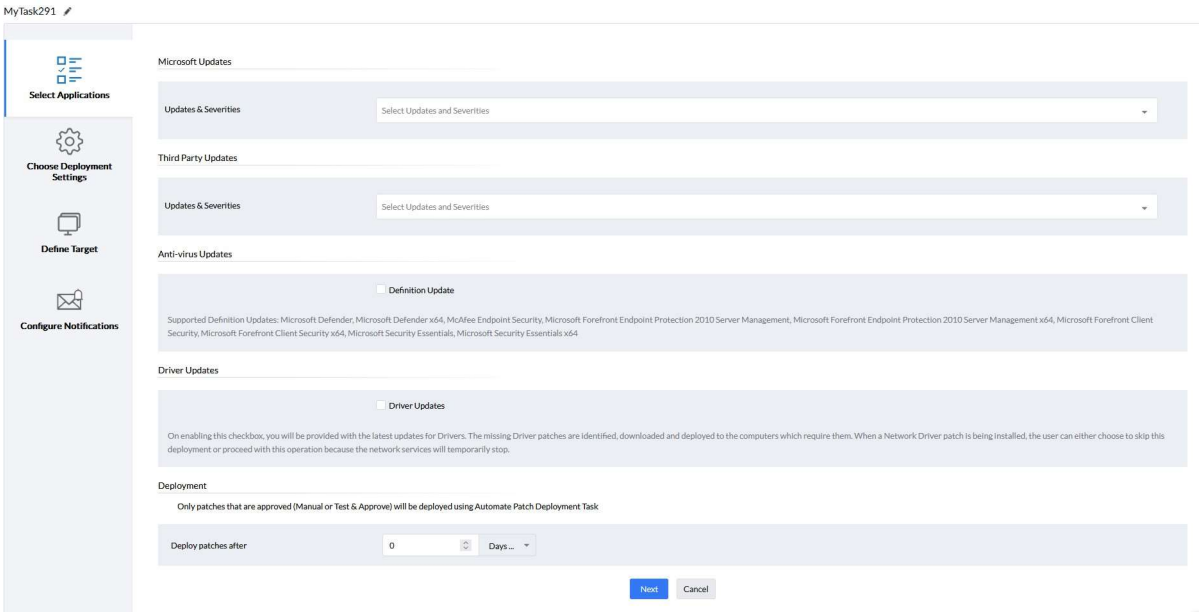
2.2 Set up automatic deployments:

Now that I have a test and approve procedure, I create the first automatic deployment for the rest of the endpoints in the organization. I start with the Critical and Important patches.

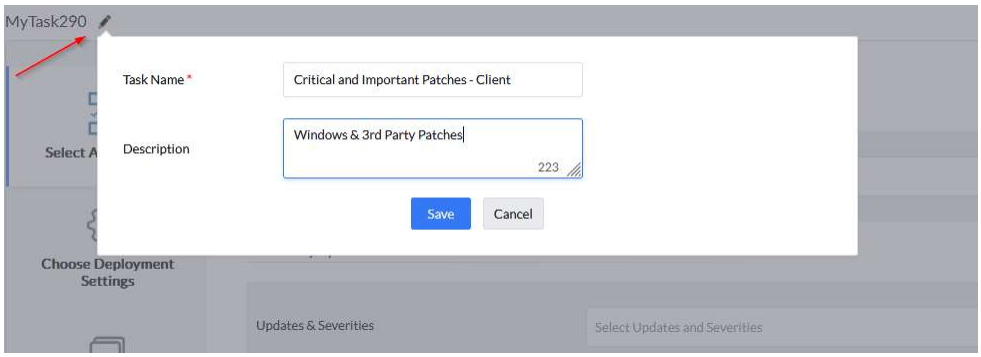
To do this, switch to "Automate Patch Deployment" and "Automate Task/Windows".



The dialog has a similar structure to that of the test group.



We give the new baby a name and a description.

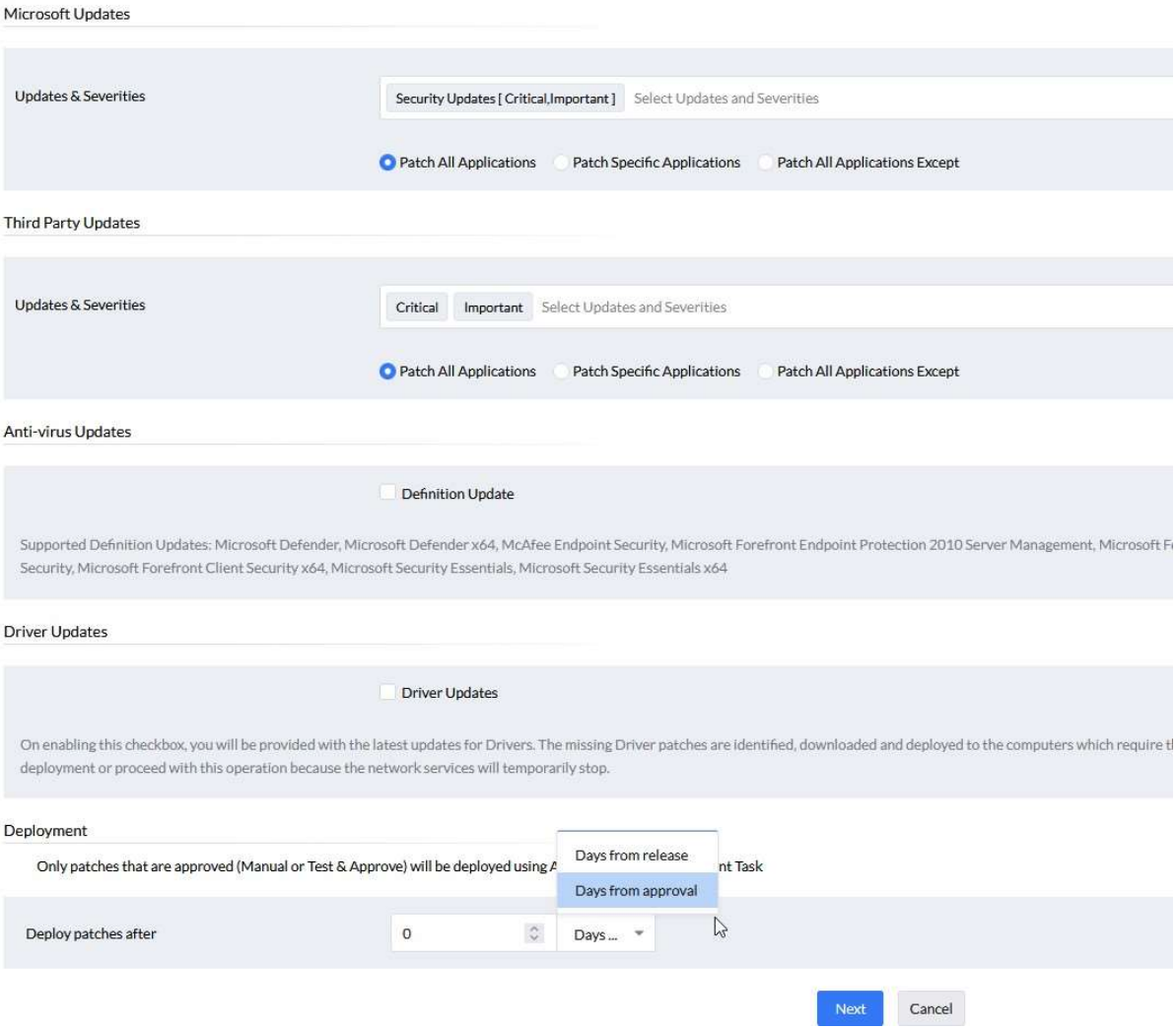


For the Microsoft and third party patches, I select Critical and Important and define any exceptions for software that should not be patched automatically.

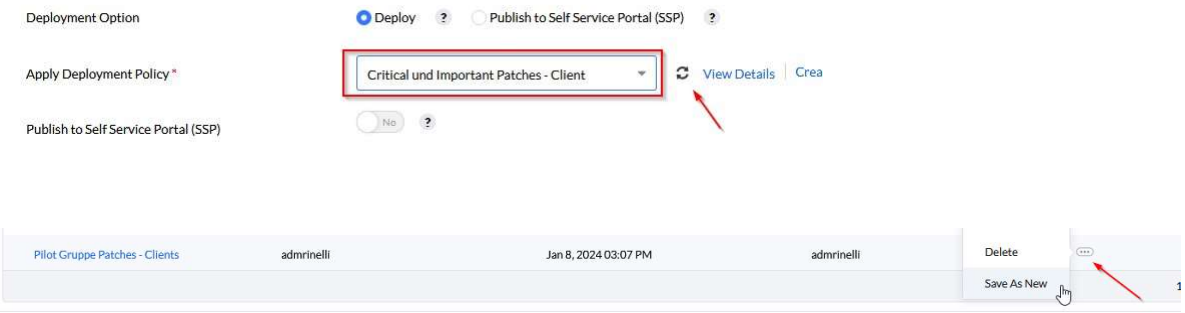
I leave Anti-Virus updates to the Virus software. In my opinion, this should be the only thing that the clients can download themselves. This way I can be sure that virus patterns are up-to-date - even if the agent on the system cannot communicate with the EndpointCentral server and therefore no updates are pushed.

I personally distribute driver updates manually or in a separate automated process.

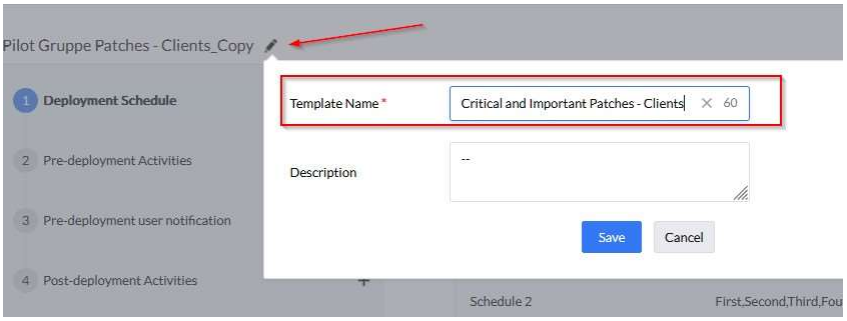
The last point on this page is the time when the patch should be distributed. As I don't want to delay distribution any further - we already have three days from the test procedure - I specify zero days from approval here. This also allows me to manually test and approve High Critical patches that have just been released. Thanks to my manual release, these are then immediately taken into account in the automatic deployment. I don't have to create a separate configuration for this - another time saver.



Continue with Next and I get to the Deployment Settings. Create a new one here again via Create. Don't worry ... we can duplicate the deployment policy that we previously created for the pilot group. To do this, simply click on the action button of the "Pilot Group Patches - Clients" policy and click "Save As New".



Rename and save again.

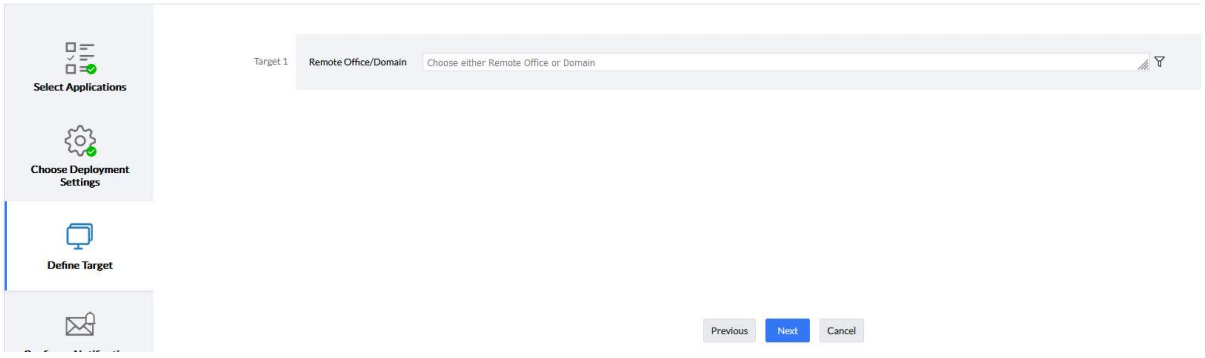


Then save each dialog with "Safe & Continue". We retain all times and settings.

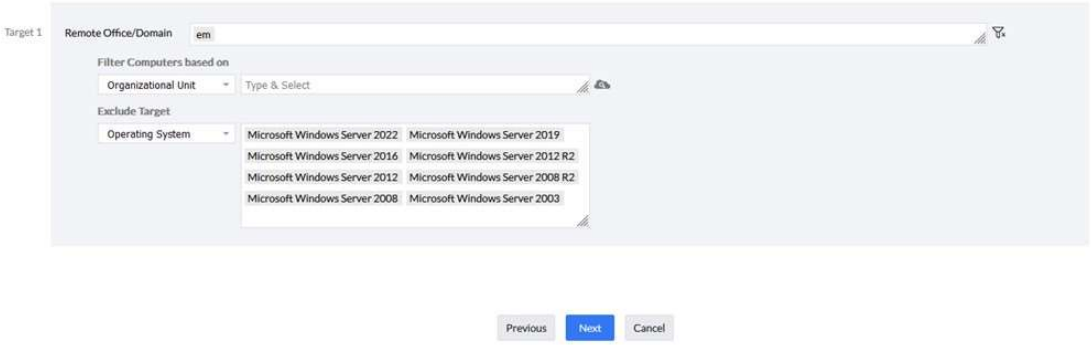
Back in the tab with the Automatic Deployment Policy, we can select the policy we have just created by clicking on the Refresh button.

Patches can also be deployed in the Self Service Portal. In my case, I leave it at normal deployment. The Self Service Portal is particularly useful when I want to distribute features or service packs. I also use the Self Service Portal for OS upgrades, giving employees the opportunity to plan more time-consuming patching themselves. Of course, I set a deadline by which the patching is then enforced. It is important to me to involve the users.

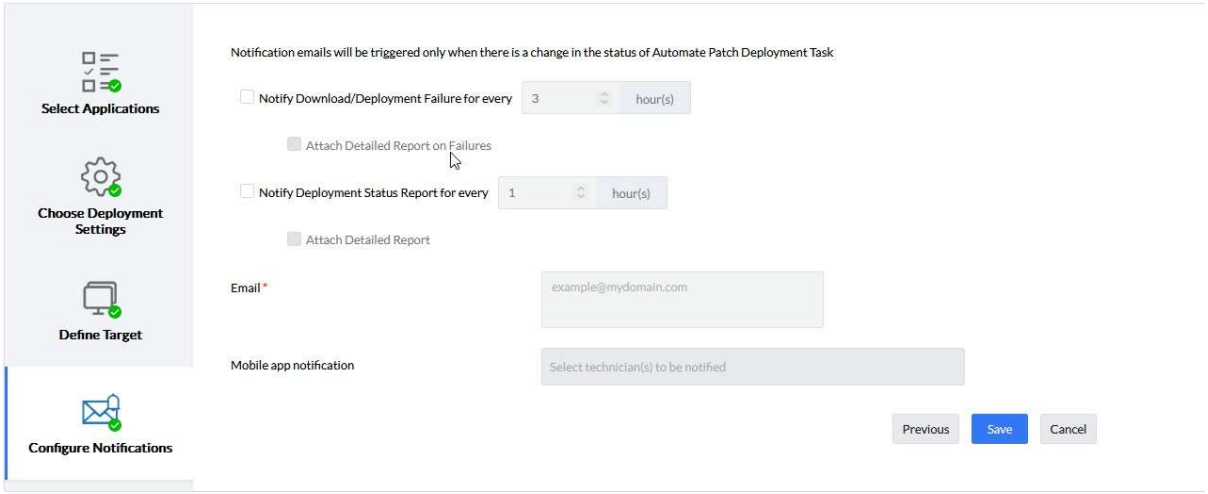
Via "Next" I get to the target definition.



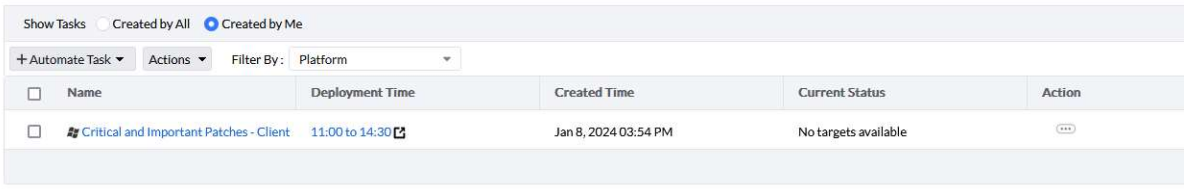
I select a remote office or the domain, filter according to the systems and exclude certain groups such as the server operating systems or custom groups in which I have summed systems that are not allowed to receive anything automatically.



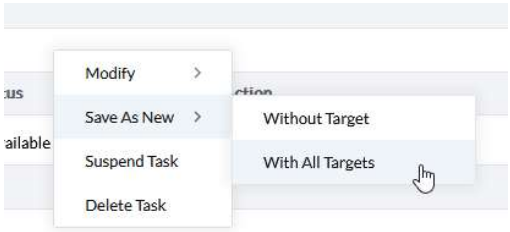
Clicking on "Next" takes me to the last dialog with the notifications. Email and mobile app notifications are also possible here. I leave it to the dashboards and so the settings as they are and click on "Save" without making any further selections.



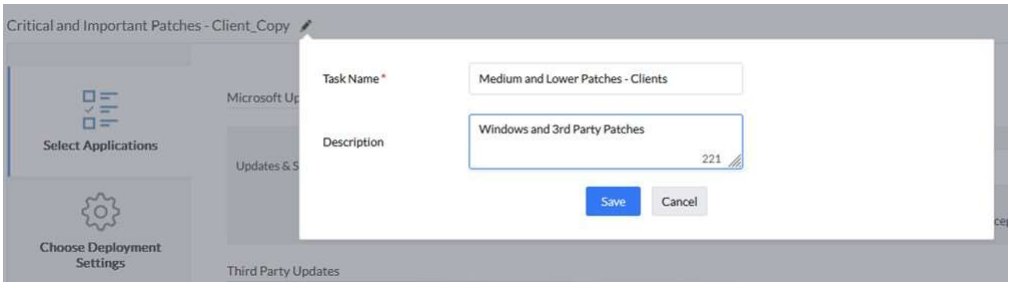
Drumroll … our first of two automatic distributions is ready.



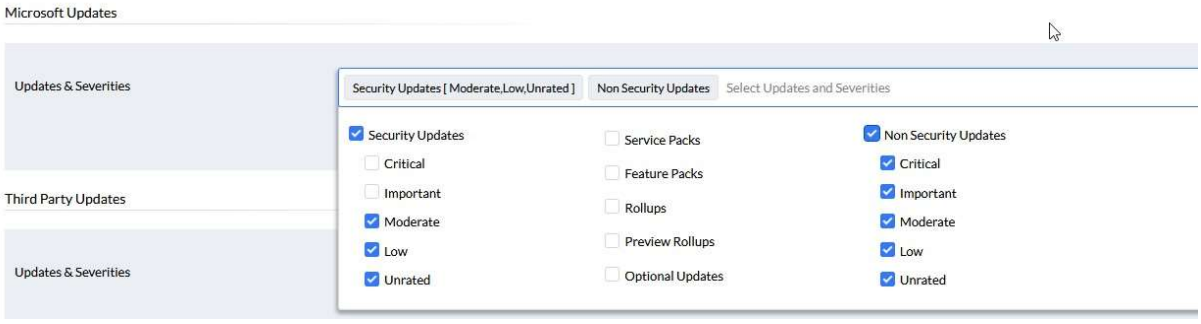
Now I duplicate these using the action button by choosing "with all targets" in order to distribute the remaining patches from medium downwards.



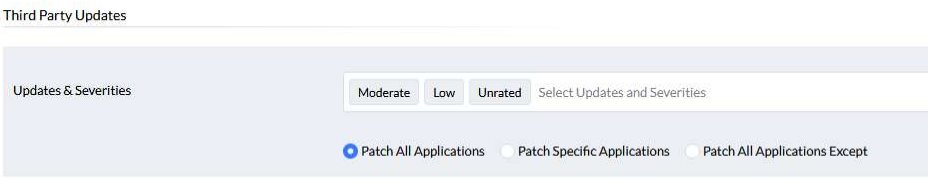
Rename briefly.



Select patches for Microsoft updates. Service and Feature Packs and optional updates I distribute separately. Either in a special automatic process or via a manual configuration. A monthly cycle could be an option here.

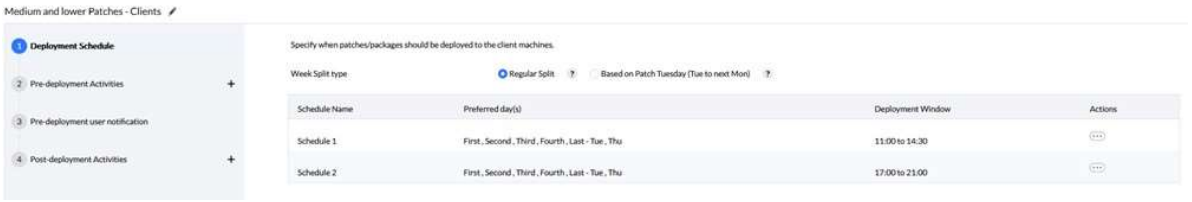


The same applies to third-party patches.

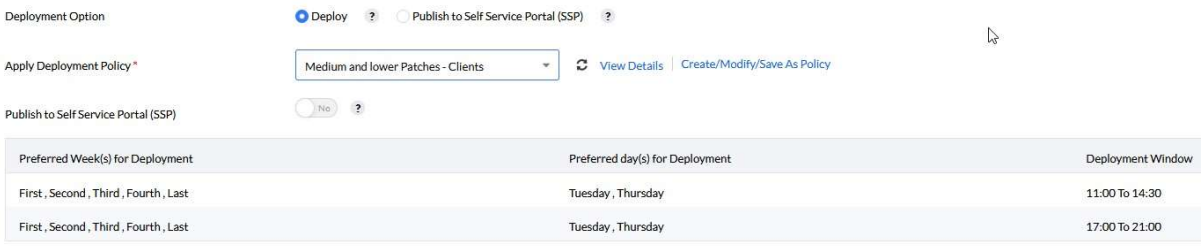


After clicking on Next, we get to the deployment settings. Here we create a new one, as I only deploy these twice a week at the time of writing this book. So go to "Create/Modify/Save As Policy" again.

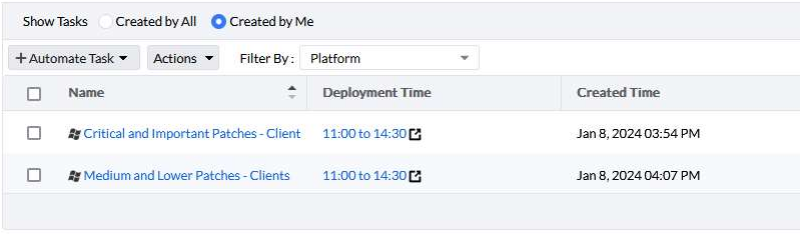
I duplicate the Critical and Important Patches - Client Policy, rename it and define only Tuesday and Thursday as deployment days at the same times. The rest remains as before and I click through all dialogs with "Save & Continue".



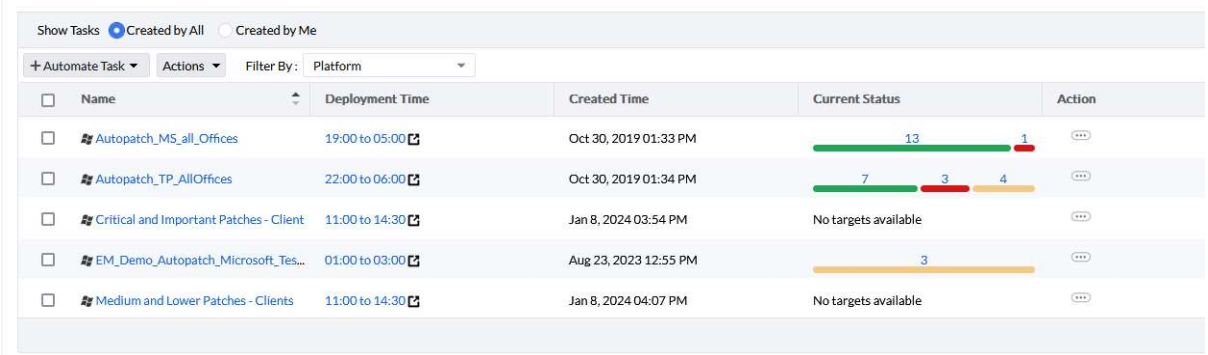
Back in our automatism tab, I can select the new policy after a refresh and confirm all further dialogs without making any changes.



Then I have my two automated deployment tasks which roll out all the approved patches in the company.



I can use the status displays to check the progress and investigate any incorrect distributions.



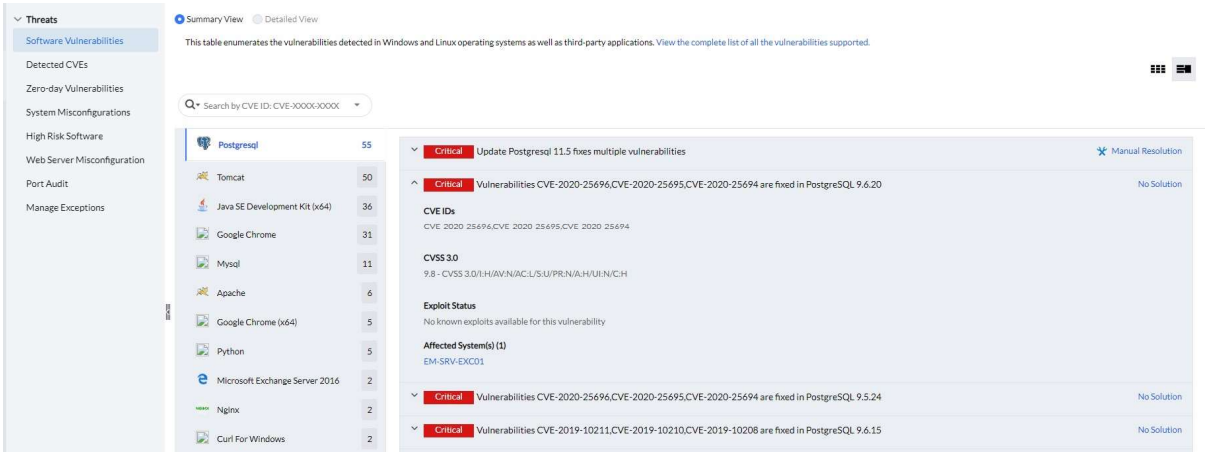
Another tip! When I enter customer environments in which the Speedy Gonzales sneaker administration has previously still been in place, I first look at the number of missing patches in the overview. If there are too many, I create both automatisms as described above, but suspend those for medium and lower patches. For the first two weeks, I only roll out the Critical and Important patches. This way, the systems and my network are not overloaded. I let some steam off, so to speak.

2.3 Checking further vulnerabilities in thread management:

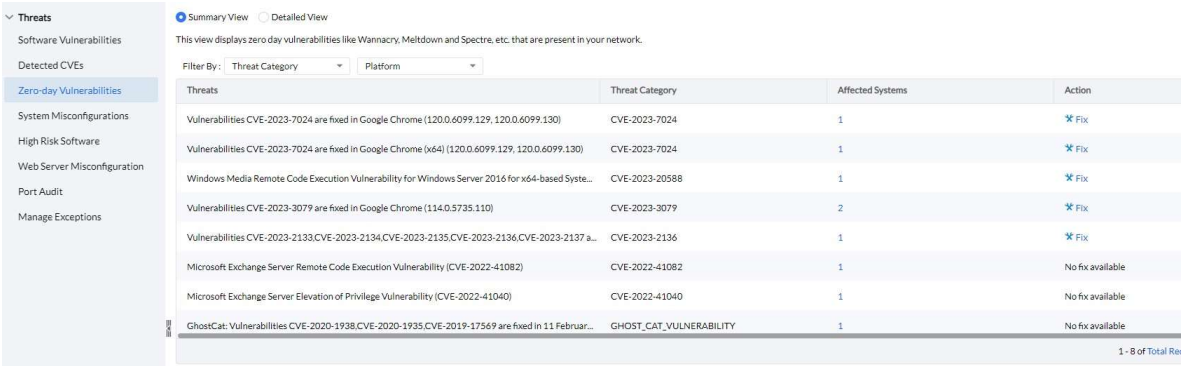
In addition to the normal patches, there are also a lot of other vulnerabilities that I can take care of. Misconfigurations, zero-day vulnerabilities, open ports, web server misconfigurations, etc.

Within "EndpointCentral Security Edition" and in "Vulnerability Manager Plus", an additional "Threats" section has been created for this purpose.

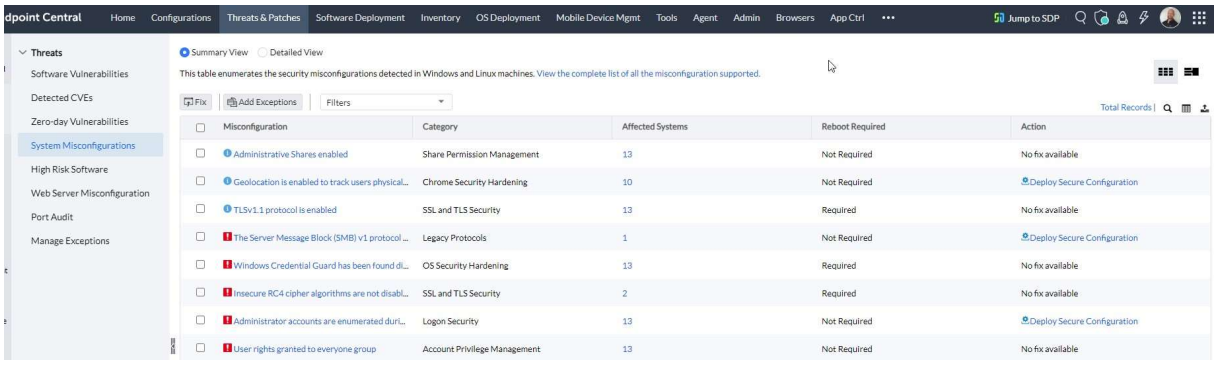
Here I find an overview of my open vulnerabilities, including a detailed description and CVE scores, divided into categories. I can also define exceptions here and/or fix the vulnerabilities.

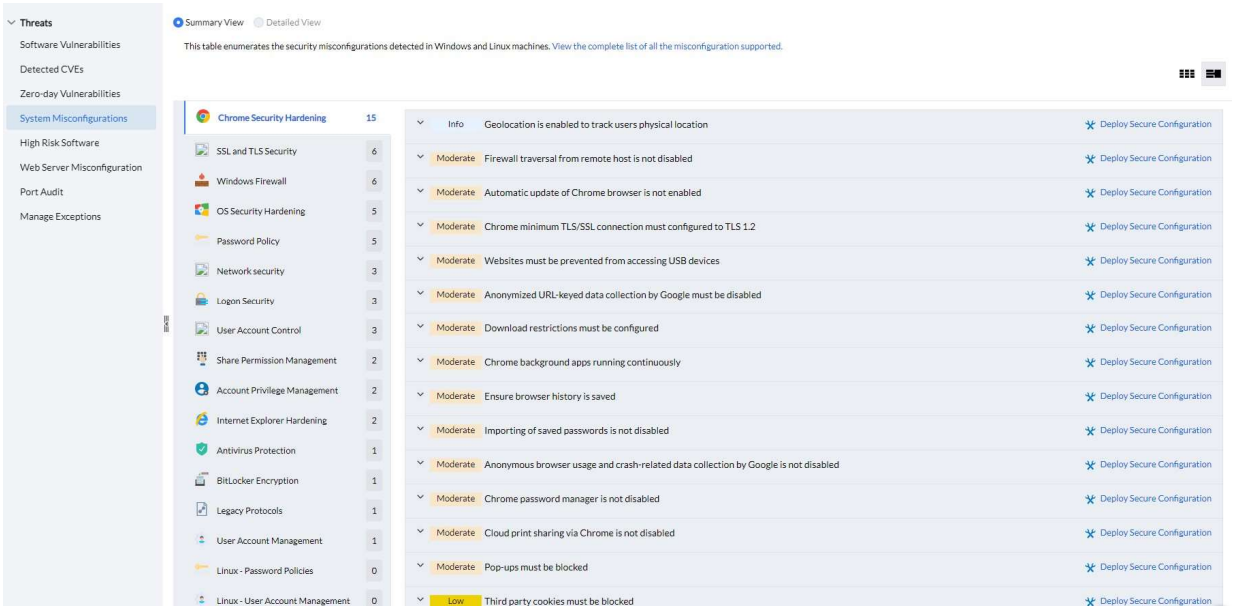


There is also an area that shows zero-day vulnerabilities and I can act accordingly.

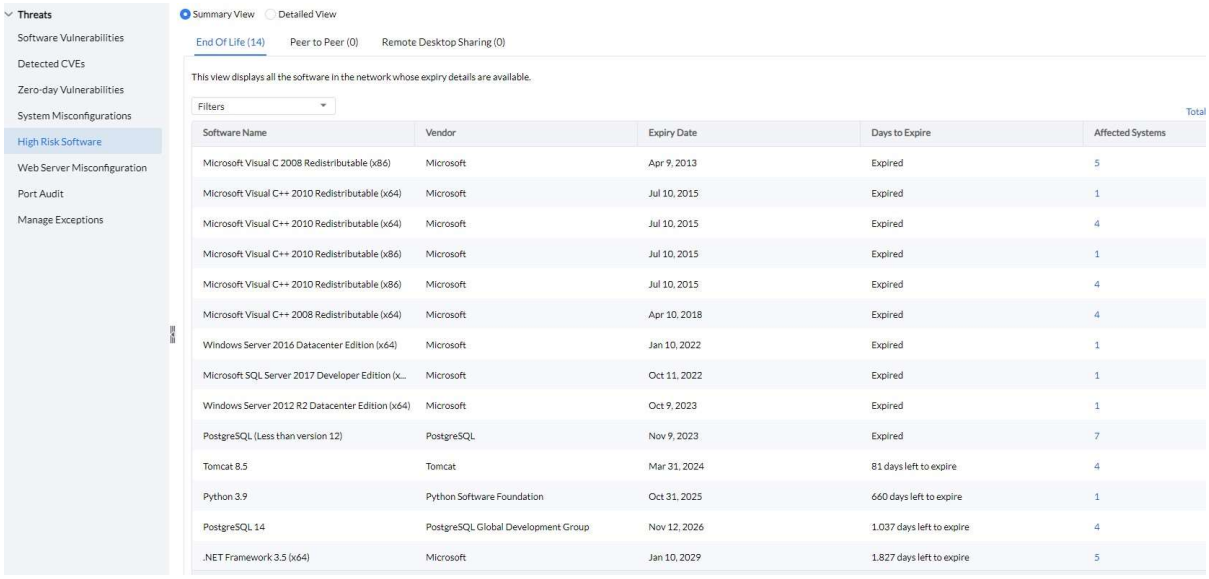


In the "System Misconfigurations" area, I solve what I can or define exceptions. Of course, much of this can be configured via the Active Directory. I can use the results of this check and, if necessary, configure them directly in AD.

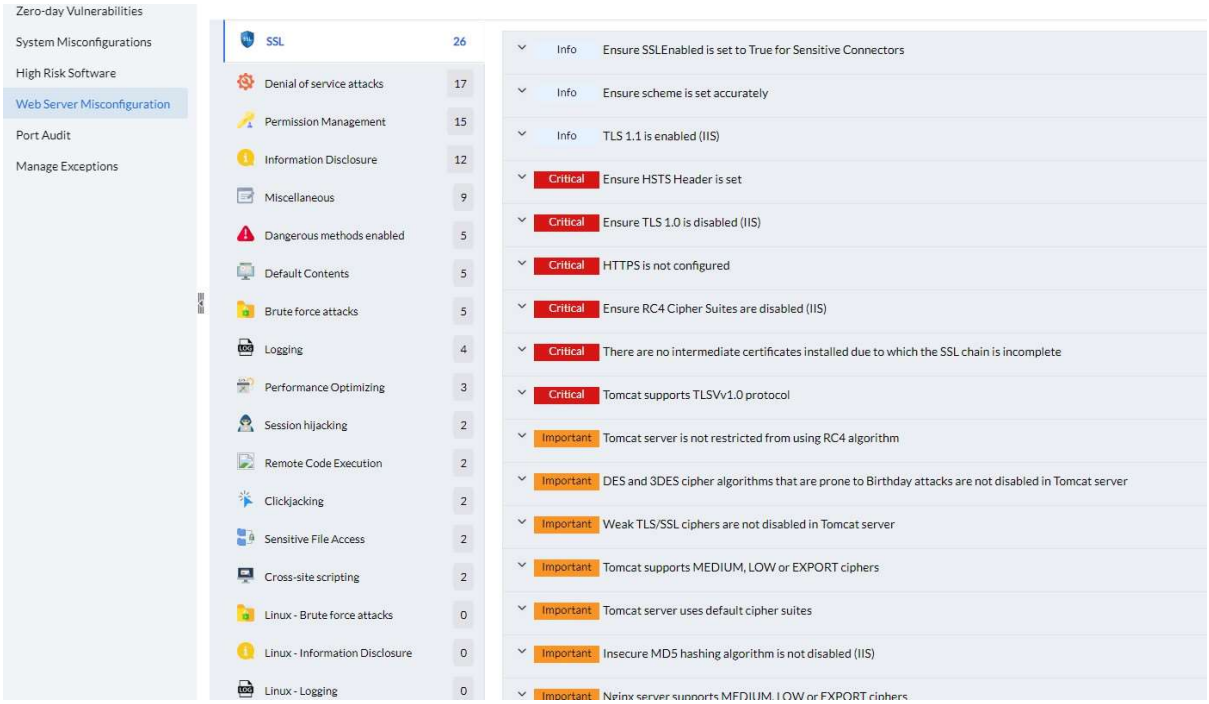




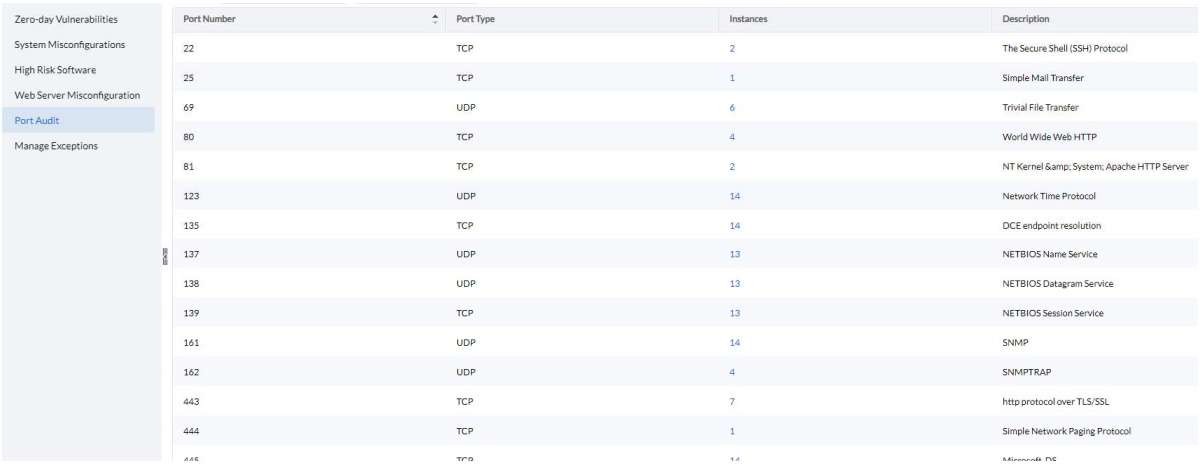
Then I check whether there is expired software under "High Risk Software" or check when existing software reaches its end of life. A topic that is often neglected and "suddenly" appears.



Other vulnerabilities affect the web servers. In this section I get an overview with suggested solutions that I can use to fix vulnerabilities on the web servers.

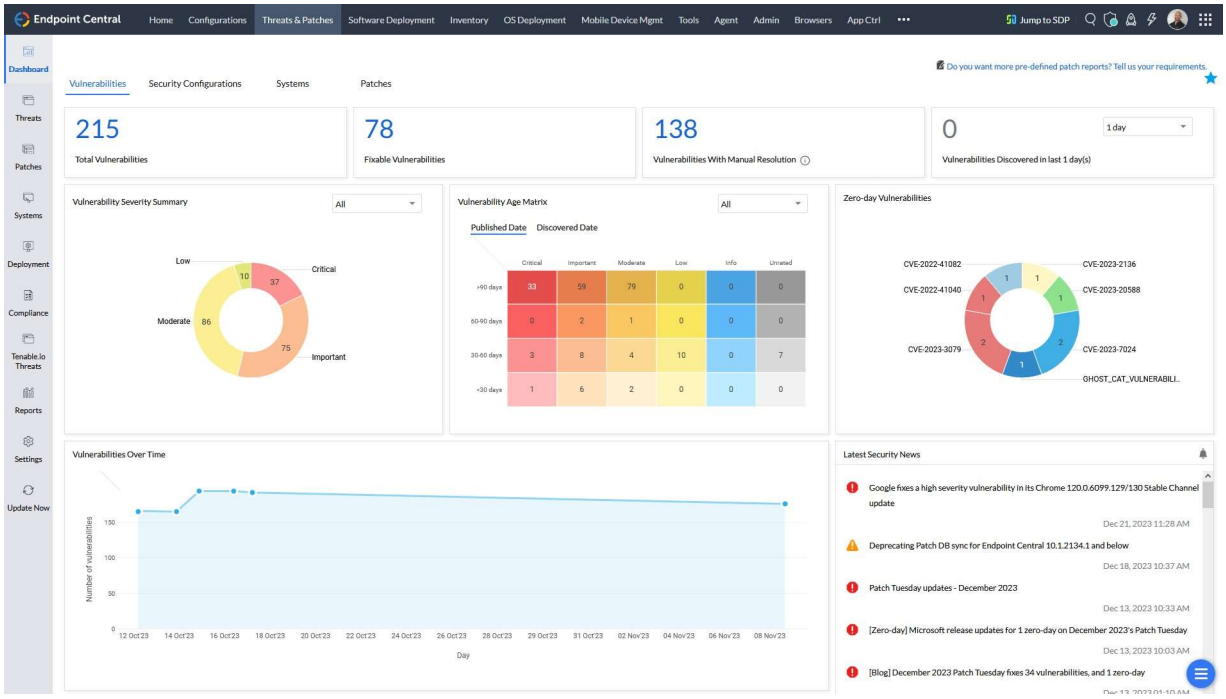


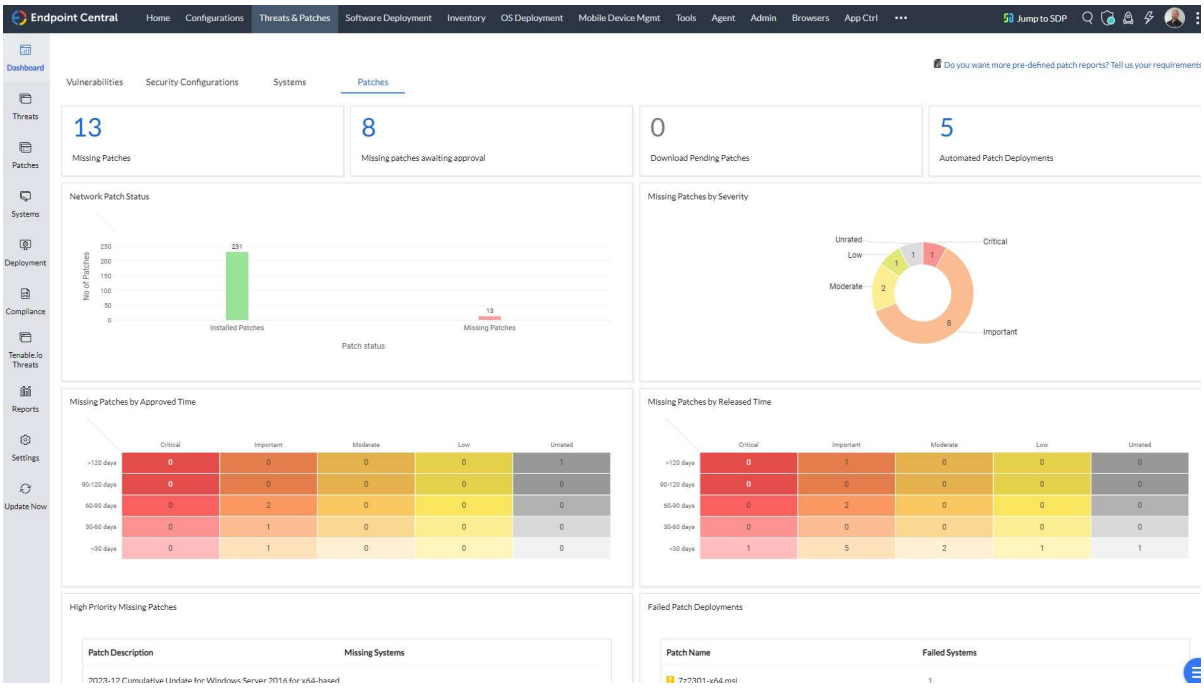
And finally, I have a look at which ports are open on which systems.



2.4 Time for the Cappuccino - check dashboards:

Instead of digging through a flood of notifications and status emails, I stick to the dashboards. The Threats & Patches dashboard provides me with all the information I need at a glance and I switch through the dashboard to areas that require my attention. A clear dashboard with direct links is a MUST for me.





Once a clean patch strategy has been defined, I only need to take care of the missing patches that have not yet been released and, if necessary, troubleshoot if a patch could not be successfully installed on a system.

Admittedly, it is an initial effort to configure a strategy. However, once set up, it significantly reduces my previous manual effort and offers maximum protection for the endpoints. What I love so much about the ManageEngine solution is, that I can create patch automatisms clearly and easily - without having studied rocket science or be an magician. And with good dashboards and reports, I don't need to consult a crystal ball to find out how secure my clients are.

That's my strategy for endpoint patching. This frees up my time, which I can use to tackle other vulnerabilities such as USB devices, applications, browsers, Bitlocker, etc.. There are also many solutions for this on the market and of course in my preferred management solution "EndpointCentral".

Especially at a time when I'm finding it difficult to find new colleagues in IT, I need a solution that gives me the freedom I need. With EndpointCentral I have such a solution with which I also have excellent support from ManageEngine and Partners around the globe.

So what waiting for?

Have fun and keep your Endoints up to date

Jürgen Rinelli

3. About the Author:

MCITP, MCTS, MCP, MOS, Enterprise Administrator, Senior Software Consultant, SCCM Specialist, Author, Coach, Reiki Teacher …

Jürgen Rinelli was born in Germany in 1970. In his eventful and often adventurous life, he has lived and worked in many countries. Whether as a businessman, manager, mechanic, trainer, diver or IT expert, he always finds a way to pursue his dreams.