PATCH MANAGEMENT IN A CONNECTED WORLD

AXONIUS

You Can Only Patch What You Can See: Patch Management In A Connected World

OVERVIEW

At its core, patch management is a discipline that combines both knowledge and action. It requires IT and security teams to understand which devices are known and unknown, the version and subsequent vulnerabilities of software being used, and the impact of change. In an environment complicated by always-on, smart devices, a new approach is needed to address what's known as well as the unknowns that *should* be known.

Table of Contents

The Patch Management Problem

Traditionally, IT and Security teams had to know 7 things to enact a viable patch management process:

1. Which Devices Are Connected to the Network?
2. What Version of Software Applications are Resident?
3. What Security Vulnerabilities are Present?
4. How Critical are the Vulnerabilities?
5. Is a New Version Available?
6. What's the Impact of Upgrading?
7. What's the Urgency of Upgrading?

Answering each of these basic questions are critical in determining whether a patch should be applied. And while these may seem obvious and straightforward, we can see that there are several challenges along the way.

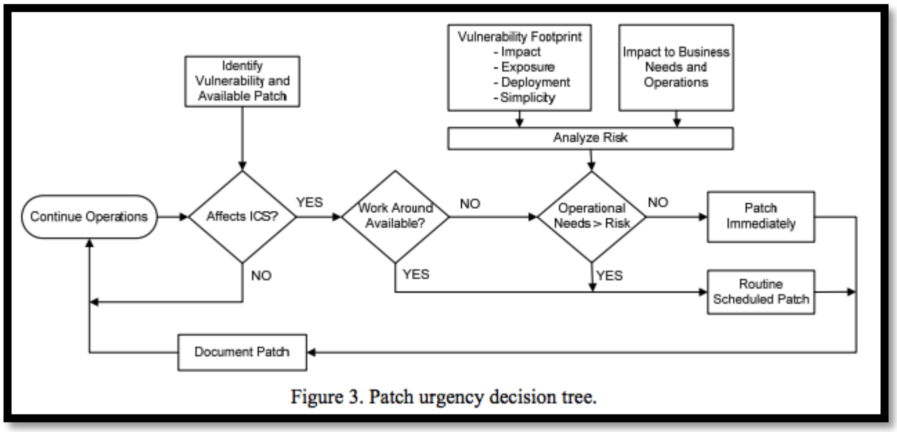


Figure 3. Patch urgency decision tree.

Patch Management Decision Tree[1]

[1] Recommended Practice for Patch Management of Control Systems

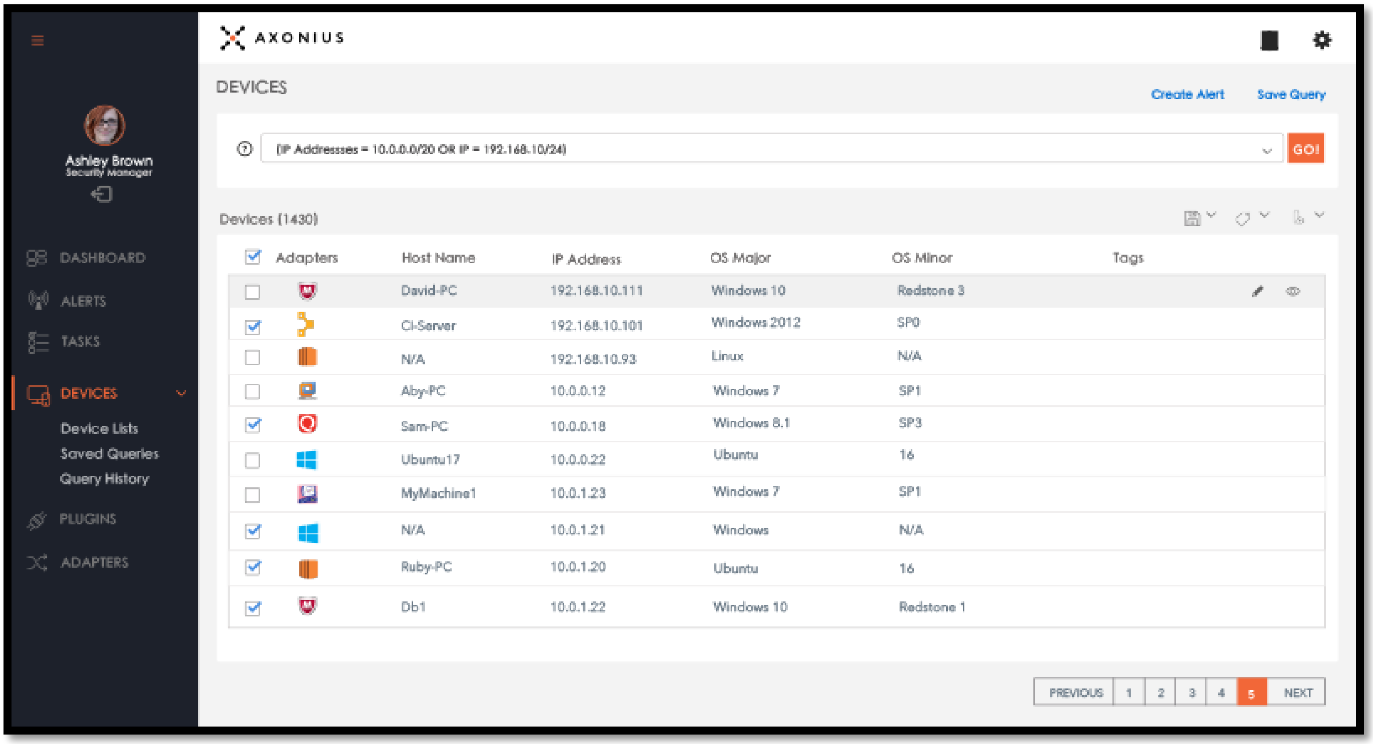
Which Devices Are Connected to the Network?

We can only patch those things we know exist, so we must first start by knowing what is already managed. A plethora of IT and Security tools exist to manage things like:

1. Authentication
2. Identity and Access Management
3. Endpoint Protection
4. Vulnerability Assessment

And although these tools do a fantastic job of managing specific devices or data types, they create more fragmentation through silos. Taken together, they can answer the question "which of the devices managed through my many solutions are connected to my network?"

However, the problem of the unknown, unmanaged device persists. How can we get to the unknowns that **should be known and managed** when we can't see them?



(IP Addresses = 10.0.0.0/20 OR IP = 192.168.10/24)

Devices (1430)

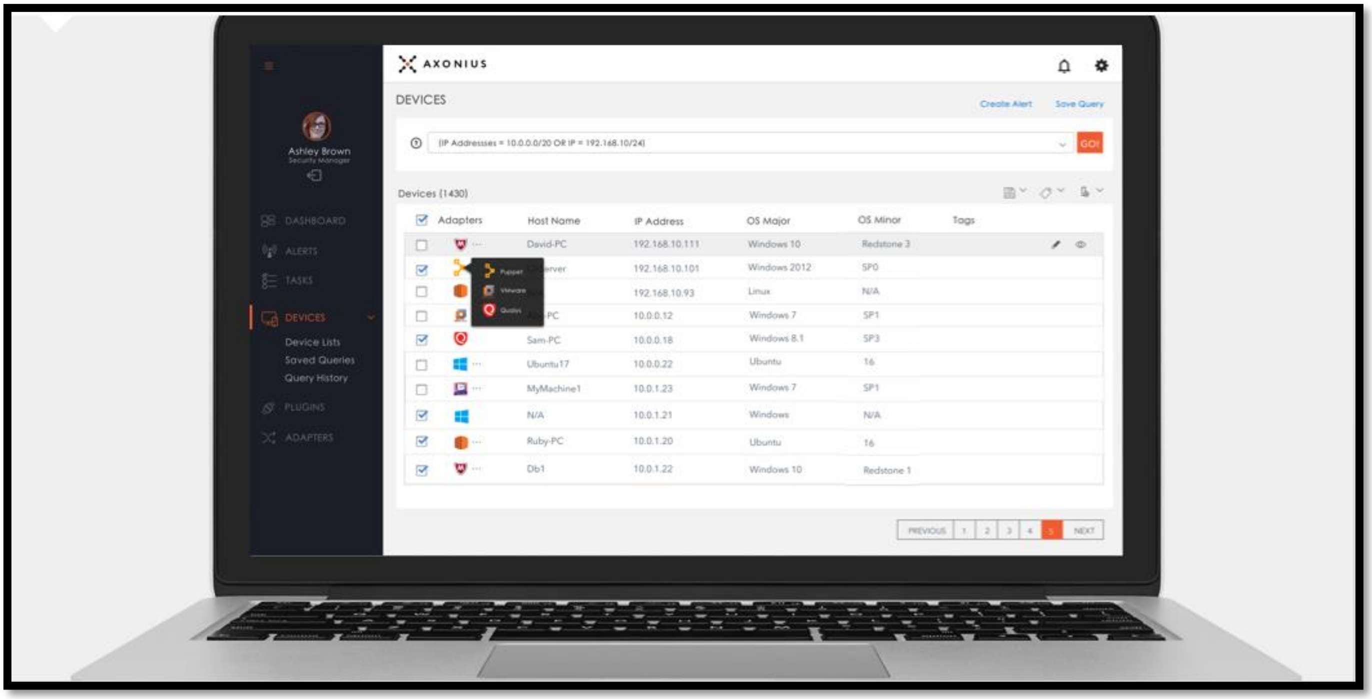
☑	Adapters	Host Name	IP Address	OS Major	OS Minor	Tags
☐		David-PC	192.168.10.111	Windows 10	Redstone 3	
☑		CI-Server	192.168.10.101	Windows 2012	SP0	
☐		N/A	192.168.10.93	Linux	N/A	
☐		Aby-PC	10.0.0.12	Windows 7	SP1	
☑		Sam-PC	10.0.0.18	Windows 8.1	SP3	
☐		Ubuntu17	10.0.0.22	Ubuntu	16	
☐		MyMachine1	10.0.1.23	Windows 7	SP1	
☑		N/A	10.0.1.21	Windows	N/A	
☑		Ruby-PC	10.0.1.20	Ubuntu	16	
☑		Db1	10.0.1.22	Windows 10	Redstone 1	

Which Versions of Software Applications Are Resident on the Devices?

Assuming we are able to see all devices, we then move on to the task of creating an inventory of all applications along with the version of each. In an ideal world, it would be easy to simply query one management system to return a list of devices with:

1. All installed applications on every device
2. The major and minor versions of each installed application
3. Whether or not the application is being used (last time the application ran)

With both a list of all devices and a list of installed software and version information, we can start to make informed decisions. We now have a baseline to work with, but now we'll need to gather some external information.



DEVICES

(IP Addresses = 10.0.0.0/20 OR IP = 192.168.10/24)

Devices (1430)

☑	Adapters	Host Name	IP Address	OS Major	OS Minor	Tags
☐		David-PC	192.168.10.111	Windows 10	Redstone 3	
☑		erver	192.168.10.101	Windows 2012	SP0	
☐			192.168.10.93	Linux	N/A	
☐		-PC	10.0.0.12	Windows 7	SP1	
☑		Sam-PC	10.0.0.18	Windows 8.1	SP3	
☐		Ubuntu17	10.0.0.22	Ubuntu	16	
☐		MyMachine1	10.0.1.23	Windows 7	SP1	
☑		N/A	10.0.1.21	Windows	N/A	
☑		Ruby-PC	10.0.1.20	Ubuntu	16	
☑		Db1	10.0.1.22	Windows 10	Redstone 1	

What Security Vulnerabilities Are Present?

We now turn the corner from “what do we have?” to “what does it mean?”. To derive meaning, we’ll need to find up-to-date vulnerability data on each application version to allow us to cross-reference and determine which devices are subject to which vulnerability.

Real World Example: Vulnerability Assessment Questions to Answer

The vulnerability assessment process is wide-ranging, starting from the simple to the more detailed. For example:

1. What vulnerabilities target Windows 10 machines that are running version 1607?
2. Are there any zero days that impact Macs running High Sierra with an outdated version of Chrome?
3. What are the risks associated with Windows Phones now that Microsoft has discontinued support for updates?
4. Are the IoT devices connected to my network likely to become part of a DDOS army?
5. Which of my AWS instances have SSL or Apache Struts vulnerabilities that are likely to be found by crawlers?

How Critical Are the Vulnerabilities Present?

In an ideal world, we wouldn’t need to prioritize vulnerabilities. If any vulnerability was found on any device, we’d want to immediately patch and/or upgrade the software on the device to ensure optimal security. Unfortunately, the world is not ideal (we’re working on it) due to some real challenges:

1. **Too Many Devices, Too Much Software** – In many organizations, the sheer number of devices, coupled with the number of installed applications makes being always up-to-date impossible.
2. **Not Enough Staff** – Even with perfectly accurate and current information, most IT and Security teams simply do not have the resources to devote full-time staff to patching and upgrading on-demand.

This leads to the need to prioritize. But unfortunately, prioritization is just a conscious decision about what we’re willing to ignore.

Is a New Version Available?

Like we did when we needed to understand which vulnerabilities exist, we must again use external sources to find out whether a new version of software is available. This seems like it should be easy. As consumers with smart phones, everything is managed through our app store, and we see any time an update is available. That’s not the case in enterprise software because we have a very important next question to answer.

Real World Example: Upgrading Linux Servers

When it comes to performing updates between management software, distributions and the broader ecosystem, Linux differs in many aspects (for e.g. Docker has its own way of patching). Some examples of questions that arise include:

1. How can I make sure to install only security updates in order to minimize impact? Depending on the distribution, you can do that by configuring the update repositories or by a specific command.
2. What should the correct update repositories be and where are they present on the distribution? For example, the repo for patches in Ubuntu is */etc/apt/sources.list* and in Redhat it is */etc/yum.repos.d/*
3. What is the right command to execute on that instance? Like the previous example, in Ubuntu, the example would be: *”apt-get update && apt-get -y upgrade”* and in Redhat: *”yum -y update”*
4. Through which management solution can I execute the command? A machine managed through Puppet would be different then a machine in ESX managed through VMWare Tools.
5. As a simple example, this is what you would need to execute on the Puppet managed machines, running Redhat to only install security patches: *class yum::update { exec { ”yum-update”: command => ” yum install yum-plugin-security; yum update-minimal --security -y; ”, timeout => 1800 }}*

Axonius plugins consolidate all of these details into the same policy with many different options and are enforced within the customer’s environment with little to no effort.

What's the Impact of Upgrading?

Dependencies: the bane of patch management.

In the smartphone example above, dependencies are all managed through the device. But in enterprise software, there are so many interdependencies that patching something could break another. It happens all the time.

So apart from prioritizing which patches and upgrades to apply, IT and Security teams need to decide whether applying a patch will adversely impact something else.

Real World Example: Standard Upgrade Takes Laptops Out of Commission

Right now, this paper is being written on my personal laptop. Why? After seeing a notification that a new update for macOS was available for my MacBook Pro, I decided to go for it (it was a security update, and we are a security company). Four calls and 6 hours later, that MacBook Pro is at an Apple Store under the watchful eye of a Genius. Fingers crossed.

Luckily, I have backups and another machine to use. However, imagine this kind of issue on an enterprise level: hundreds of MacBooks being inaccessible with users unable to work and potential data loss.

There's always tension between wanting to stay up-to-date with the latest OS version for security reasons and wanting to stay with what works.

What’s the Urgency of Upgrading?

If we’re successful in all the steps above, we’ve got the following:

1. A list of all devices both managed and unmanaged
2. A list of all software resident on those devices
3. A prioritized list of vulnerabilities for all software
4. A list of available upgrades and patches for all installed software
5. An understanding of the impact of upgrading/patching

You’d think that would be enough. Problem solved. But even then, it’s likely that we have too many devices and applications to reasonably patch everything.

Just as we did when we addressed which vulnerabilities are most critical, we must prioritize what we’re going to upgrade. Only then can we start.

Real World Example: Where Were You on WannaCry Day?

We’ve heard many examples from CISOs about what they had to do in May of 2017 when the WannaCry ransomware attack locked hundreds of thousands of computers in over 150 countries. In most large multinational companies with distributed networks, a war room was created with a team working across the globe to identify those machines that had been infected, and which were vulnerable.

In hearing this story many times, one theme was common: most organizations scrambled to manually identify and upgrade those devices that were at immediate risk.

IoT Devices: A New Challenge

We doubt that any reader will get this far thinking "okay, everything sounds easy. Give me a brand-new set of challenges to make my easy job harder." But if anyone is feeling the need to make patch management more difficult, look no further than IoT devices.

The explosion of always-on, smart, connected devices represents a monumental shift in how IT and Security teams think about managing devices. While we could easily give examples of the casino that was brought down by a hacked smart aquarium, or go on about how Krebs was DDOSd off the internet by an army of zombie baby monitors, but we'll save you the trouble. Examples are everywhere.

The simple economics of IoT devices require them to be made very inexpensively, and when cost is inflexible, security is often the first tradeoff.

And setting aside the inherent security risks that come along with IoT devices, there's a more fundamental issue: it's hard to know which of these devices are on our networks. It's even harder to know which of these devices should be there and managed.

Real World Example: Amazon Alexa for Business

In November, 2017, Alexa for Business was announced as Amazon's IoT move into the enterprise. It's easy to see that businesses want to use IoT devices that can drive productivity. In this case, companies are using Alexa to synch calendars, manage conference rooms, and coordinate meeting assets.

However, organizations don't want yet another device that needs another system to manage it. The software that manages these devices can't be a speed bump that takes away the potential productivity gains.

We've also heard that in some cases just knowing an IoT device is present is enough to take action. While you may not know if there's an upgrade available when it comes to IoT, knowing what you have is crucial.

Seeing All Devices for What They Are

Regardless of the type of device, mobile, laptop, desktop, server, IoT, etc., it is difficult to understand all devices for what they are. There's a difference between an employee's laptop, a smart thermostat, a production Apache server, and an AWS instance.

We believe there are 6 fundamental questions that must be answered about every device:

1. Is the device "known" and managed?
2. What is it?
3. Where is the device?
4. Is the core software up to date?
5. What additional software is installed?
6. What credentials are applicable?

When it comes to patch management, each question must be answered. But without a unified view to start with, we can't be sure that all devices that should be patched will be.

Addressing the Unknown in the Network

Many customers cite usage of SCCM and agent-based patch management solutions as being central to their patching processes. Where deployed, these work well. But we've found several scenarios where SCCM combined with an agent-based system leave unknowns:

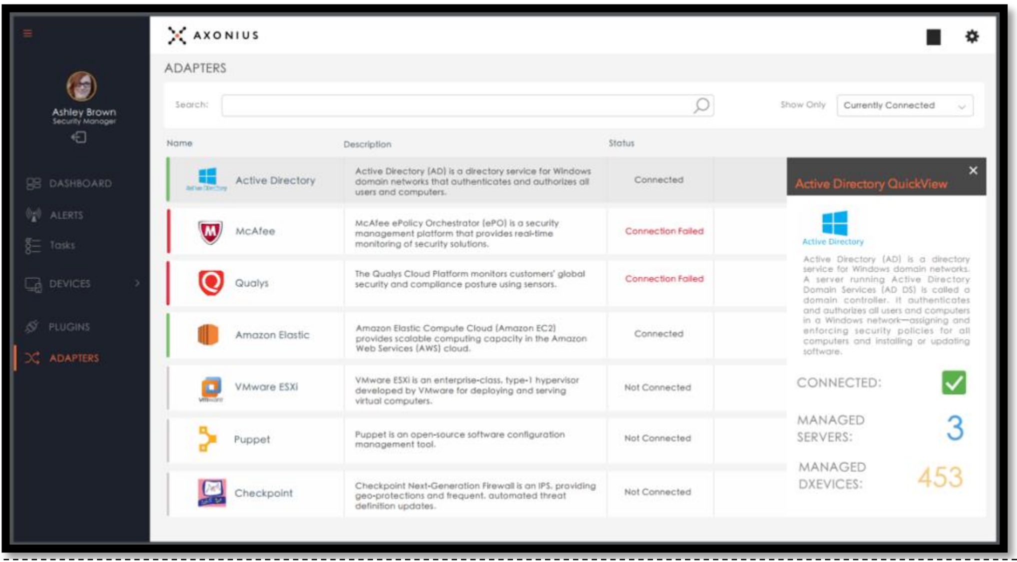
1. **Non-Windows Devices:** While SCCM works well on Windows devices, we no longer live in a Windows-only world. Smart devices, mobile phones, Macs, Linux machines, and virtual environments create a fragmented environment that leaves devices unmanaged by OS and device-specific management systems.
2. **Organizations with Subsidiaries:** Many global or distributed organizations have a structure that is not centralized, often with different groups having different policies. The office in Seattle may have different policies and security tools than the office in Shanghai. Without a centrally managed and controlled environment, getting to a unified view relies on gathering data bottom-up, requiring manual and error-prone work. [with Axonius, there's value with just read-only info and correlation]
3. **Research and Development Machines –** The devices used for research and development are treated differently from normal end-user devices, as they are by definition "exceptions". These devices are (and should be) considered different, and therefore are treated separately by both security and patching solutions.
4. **Production Servers –** The risk of taking down a critical production server is often considered too high to warrant aggressive patching. This creates a risky scenario. [emphasize the danger and the approach to not be intrusive]

The Axonius Approach: See All to Secure All

At Axonius, we empower customers to take action with a consolidated device management platform that provides a single point of view for all devices – both managed and unmanaged. By being able to see all devices as well as the patch status of all software, IT and Security teams are able to intelligently and efficiently apply patch management policies.

Adapters

Using the platform’s extensible framework, customers are able to use adapters to connect to their existing systems via APIs. This creates an abstraction layer to devices and provides a unified point of view for visibility and control.



ADAPTERS

Search:

Show Only: Currently Connected

Name	Description	Status
Active Directory	Active Directory (AD) is a directory service for Windows domain networks that authenticates and authorizes all users and computers.	Connected
McAfee	McAfee ePolicy Orchestrator (ePO) is a security management platform that provides real-time monitoring of security solutions.	Connection Failed
Qualys	The Qualys Cloud Platform monitors customers' global security and compliance posture using sensors.	Connection Failed
Amazon Elastic	Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud.	Connected
VMware ESXi	VMware ESXi is an enterprise-class, type-1 hypervisor developed by VMware for deploying and serving virtual computers.	Not Connected
Puppet	Puppet is an open-source software configuration management tool.	Not Connected
Checkpoint	Checkpoint Next-Generation Firewall is an IPS, providing geo-protections and frequent, automated threat definition updates.	Not Connected

Active Directory QuickView

Active Directory

Active Directory (AD) is a directory service for Windows domain networks. A server running Active Directory Domain Services (AD DS) is called a domain controller. It authenticates and authorizes all users and computers in a Windows network—assigning and enforcing security policies for all computers and installing or updating software.

CONNECTED: ✓

MANAGED SERVERS: 3

MANAGED DXEVICES: 453

Real World Example: Active Directory Adapter

Different properties of each device are present in different silos within the organization. What do we need to see the device for what it is? What information would we need to determine whether a device needs updating?

Firstly, in order to understand its network positioning (Is it in production or the corporate net? Is it internet facing or not?) we would go to the solutions that look at the device from a network perspective: the firewall, the vulnerability assessment tool, and/or the NAC.

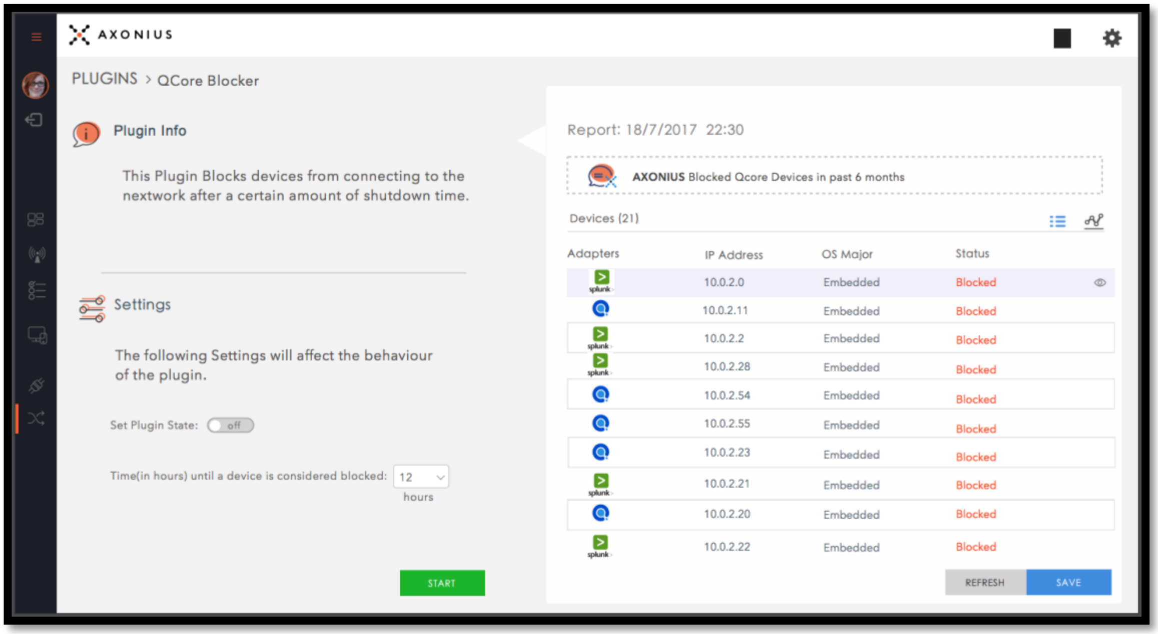
Next, in order to get a precise and accurate reading of the OS version, we would need to use an asset management system, such as SCCM/Active Directory. This will enable us to also understand the software installed on the device and the respective version and requirements for updating.

But what happens if that device doesn't have an agent and we're required to get the data only from Active Directory? Should we access each one through RPC/WMI just to get a reading of the software version? All of these different siloed properties would be able to be gathered and used through Axonius adapters.

AD is just a single example of an adapter. Many other adapters are available to integrate with tools like firewalls, endpoint protection solutions, asset management systems, and more.

Plugins

Once connected to the devices and management systems in a customer's environment via adapters, plugins allow for cross-device functionality using logic. For example, a sample plugin could connect to Active Directory and continuously query for any new OU created.



PLUGINS > QCore Blocker

Plugin Info

This Plugin Blocks devices from connecting to the nextwork after a certain amount of shutdown time.

Settings

The following Settings will affect the behaviour of the plugin.

Set Plugin State: off

Time(in hours) until a device is considered blocked: 12 hours

Report: 18/7/2017 22:30

AXONIUS Blocked Qcore Devices in past 6 months

Devices (21)

Adapters	IP Address	OS Major	Status
splunk>	10.0.2.0	Embedded	Blocked
Q	10.0.2.11	Embedded	Blocked
splunk>	10.0.2.2	Embedded	Blocked
splunk>	10.0.2.28	Embedded	Blocked
Q	10.0.2.54	Embedded	Blocked
Q	10.0.2.55	Embedded	Blocked
Q	10.0.2.23	Embedded	Blocked
splunk>	10.0.2.21	Embedded	Blocked
Q	10.0.2.20	Embedded	Blocked
splunk>	10.0.2.22	Embedded	Blocked

Real World Example: WannaCry plugin

When WannaCry surfaced, it presented an immediate threat to any network running windows and using file sharing. It required immediate action from security teams in order to safeguard their organization. How would we answer the question "which of my devices are vulnerable to WannaCry?":

1. First, we would need a list of all machines running Windows within the organization.
2. We would then need to understand if they are at the patch level that makes them specifically vulnerable to WannaCry.
3. Since this would require being able to run a command on the devices, and different Windows instances are managed differently (for example some are only managed through SCCM, some have endpoint protection, and others can only be accessed using RDP creds), we would need to go through all these different silos and systems to get the answer.

Axonius plugins answer the question by pulling the relevant data on each device from the respective solution that manages it.

About Axonius

For organizations that see opportunity in today's always-on and always-connected reality, Axonius is the consolidated device management platform that lets IT and Security teams see devices for what they are in order to manage and secure all. By easily integrating with customers' existing management and security technologies, and using an extensible plugin infrastructure to add custom logic, customers are able to get a unified view of all devices – both known and unknown. Axonius aims to be IT's favorite Security tool and Security's favorite IT tool. For more information and to see what's possible with a universal view of all devices, visit Axonius.com.

Support and Questions

We are committed to helping our customers deploy, configure, and start seeing value immediately. The POC deployment process will be hands-on, with any and all support services available to get up and running. Should you have any questions, concerns, or product feedback, please do not hesitate to contact your Axonius account representative at any time.

Thank You

Finally, we want to thank you for considering working with Axonius. As IT and Security professionals ourselves, we understand the time and effort it takes to consider a new product. Thank you for trusting us to help you.