Now on Now: Our Vulnerability Response Journey

How ServiceNow® Vulnerability Response shortens time to resolution and improves productivity and the user experience

Table of Contents

Introduction 2
Previous labor-intensive process 3
Challenges 4
ServiceNow Vulnerability Response 5
CI matching 6
ServiceNow Vulnerability Response benefits 7
Learnings and best practices 8
To learn more 8
About ServiceNow 8

Introduction

Data breaches not only put company data at risk, but also the operations and reputation of the business itself. According to the Ponemon Institute, 48% of businesses have had a breach in the last two years[1]. Of those breached, more than half said it was due to a vulnerability for which a patch was available. The cost and potential damage caused by a breach grow quickly with time as the number of impacted records increases.

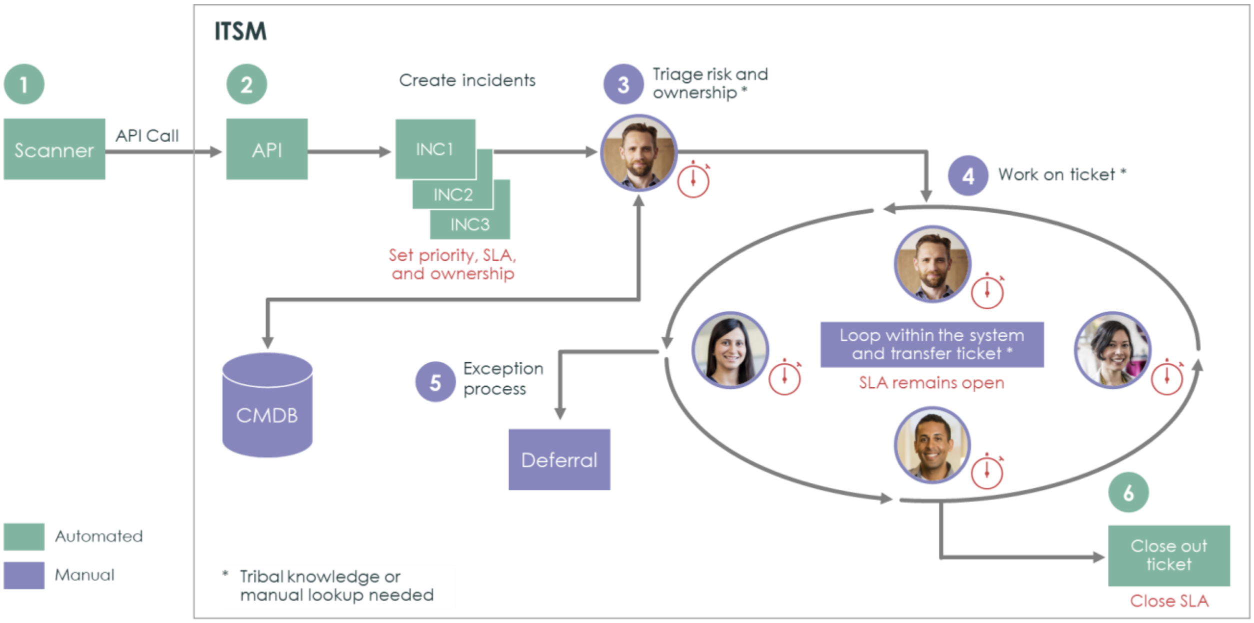
One of the best ways to prevent a breach is through an effective vulnerability management program. But many organizations struggle to keep up with patching as IT environments continue to expand, and vulnerabilities show linear growth. Prioritization and automation are key to helping security and IT teams work together faster to remediate vulnerabilities.

The ServiceNow IT Security team turned to our company's own Vulnerability Response solution to respond faster and more efficiently. Built on the Now Platform®, ServiceNow Vulnerability Response automates workflows, prioritizes and accelerates triage, and provides actionable insight for continual improvement.

[1] Ponemon Institute 2019: Costs and Consequences of Gaps in Vulnerability Response

Previous labor-intensive process

Prior to implementing ServiceNow Vulnerability Response, we used an ITSM-based approach, which involved both automated and manual steps, shown in green and purple in the following diagram:



1. **Scan** – We used a third-party scanner to detect vulnerabilities.
2. **Create incidents** – Custom API calls pulled detection data from the scanner and created incident tickets in ServiceNow IT Service Management (ITSM). Each ticket included information on all systems affected by that vulnerability. We prioritized each incident based on its severity from the scanner and set the SLA. Our systems team was the default owner for all tickets.
3. **Triage** – A systems engineer reviewed the tickets and reassigned them to another group if appropriate. Depending on the system criticality, the priority might also be adjusted. This required a lot of tribal knowledge, such as familiarity with the naming and IP addressing schemes of our environment. If the engineer did not have that tribal knowledge, they manually accessed the ServiceNow Configuration Management Database (CMDB) or a spreadsheet to get that information.
4. **Work on ticket** – The team who owned the ticket set up their patching programs. If other systems were listed in the ticket, the team passed the ticket to the next group. This would continue until all patching was complete. The SLA remained open throughout the process.
5. **Handle exceptions** – A manual deferral process was used to manage exceptions, such as false positives or risk acceptance.
6. **Close ticket** – When the next scan detected that the vulnerability was patched, the ticket and the SLA were automatically closed. Usually, this response cycle took days, if not weeks, to resolve a given vulnerability.

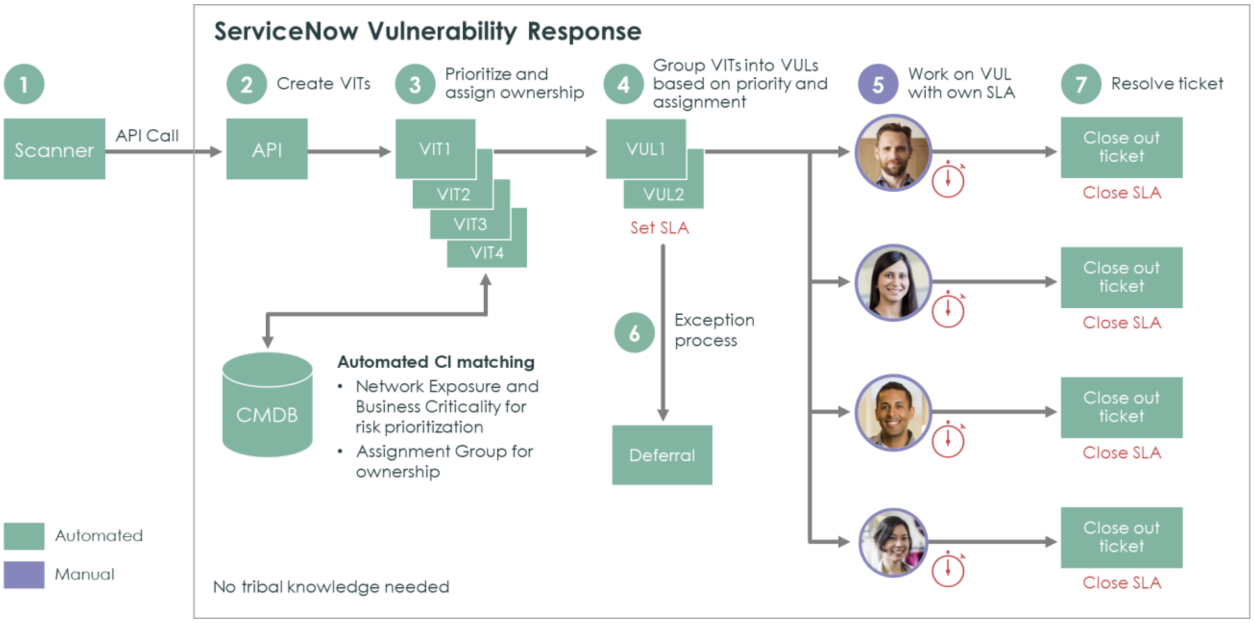
Challenges

The previous approach had multiple shortcomings:

- **Cumbersome process** – The overall process was highly manual and inefficient. Prior to deploying Vulnerability Response, the scanners only evaluated servers, not all endpoints. Including all endpoints would have increased the number of items listed in the ITSM ticket notes 30-fold and quadrupled staff workloads, which would have been unmanageable.
- **Time-consuming triage** – Host names or IP addresses had to be manually assessed and assigned, and tribal knowledge was often required to decipher which system belonged to which device and which group. Because each ticket could have multiple systems owned by different groups, some tickets had to transfer between multiple owners. Patching was done sequentially rather than in parallel, which extended the time to resolution.
- **SLA inaccuracies** – With tickets containing systems owned by multiple groups but only one SLA, it was difficult to track SLAs for a single group. In addition, an incident remained open until all servers in that ticket had been patched, which could artificially extend resolution time.
- **Manual tracking and reporting** – Spreadsheets were used to track and report progress. This process was both inefficient and error prone. In addition, having several teams working on the same ticket made it difficult to track ticket activity for an individual group.
- **Negative user experience** – All administrative work, such as manually assigning and routing tickets, took time away from interesting and value-added work for employees. It was also frustrating and time-consuming to gain the tribal knowledge engineers needed to perform a task, such as identifying servers and owners.

ServiceNow Vulnerability Response

Vulnerability Response went live internally in August 2018 and has automated virtually every workflow in the process.


ServiceNow Vulnerability Response
1
2 Create VITs
3 Prioritize and assign ownership
4 Group VITs into VULs based on priority and assignment
5 Work on VUL with own SLA
7 Resolve ticket
Scanner
API Call
API
VIT1
VIT2
VIT3
VIT4
VUL1
VUL2
Set SLA
Close out ticket
Close SLA
Close out ticket
Close SLA
Close out ticket
Close SLA
Close out ticket
Close SLA
6 Exception process
Deferral
CMDB
Automated CI matching
• Network Exposure and Business Criticality for risk prioritization
• Assignment Group for ownership
Automated
Manual
No tribal knowledge needed

1. **Scan** – We use the same third-party scanner to detect vulnerabilities.
2. **Create Vulnerable Items** – Rather than creating a single incident ticket with all impacted systems, the built-in API workflow pulls detections from the scanner and creates separate Vulnerable Items (VITs) for each vulnerability as it is applied to a single system.
3. **Prioritize and assign ownership** – Automated Configuration Item (CI) matching accesses the CMDB and matches the configuration item to the system listed in the VIT. We then use information in the CI to help prioritize and assign ownership. Other parameters in the CI, such as business criticality, can be used to calculate the risk score. We can then take the severity of the vulnerability from the scanner, run the risk score calculator, and get a true priority for our environment. Because CIs also include owner information, the system can automatically assign the VIT to the right team; no tribal knowledge is needed.
4. **Group Vulnerable Items and set SLAs** – In a large organization, the above process could create hundreds of thousands or even millions of VITs. To make this more manageable, the system consolidates them into Vulnerability Groups (VULs). We have configured the system to group VITs by vulnerability, priority, and assignment group. SLAs are set on the Vulnerability Group at this point.
5. **Resolve ticket** – Teams begin working on the tickets that are relevant to their systems. Because there's only one owner per ticket, patching can be done in parallel. Teams have their own SLAs; the work of other teams does not impact their SLAs or metrics.
6. **Manage exceptions** – Exception handling is fully automated based on the parameters we set for the Vulnerability Group. For exceptions such as false positives, risk acceptance, won't fix, etc., the system creates an automated task that is sent to a team for approval—in our case to a security group, then to a compliance group.

7. **Close ticket** – When the scanner detects that a VIT is resolved, it becomes closed. Once all the VITs in a group are closed, the Vulnerability Group is closed, and the SLA is stopped.

CI matching

The CMDB and CI matching are keys to the effectiveness of our Vulnerability Response process. We use multiple sources to create CIs:

- **ServiceNow Discovery** is our primary discovery tool.
- System configuration software such as **Microsoft Systems Center** pulls rich information for servers and endpoints.

When the scanner finds a system that is not yet in the CMDB, an *extended CI* is created to store its information while we process tickets. The extended CI must then be reconciled and brought into the CMDB as a permanent CI. We consider the following:

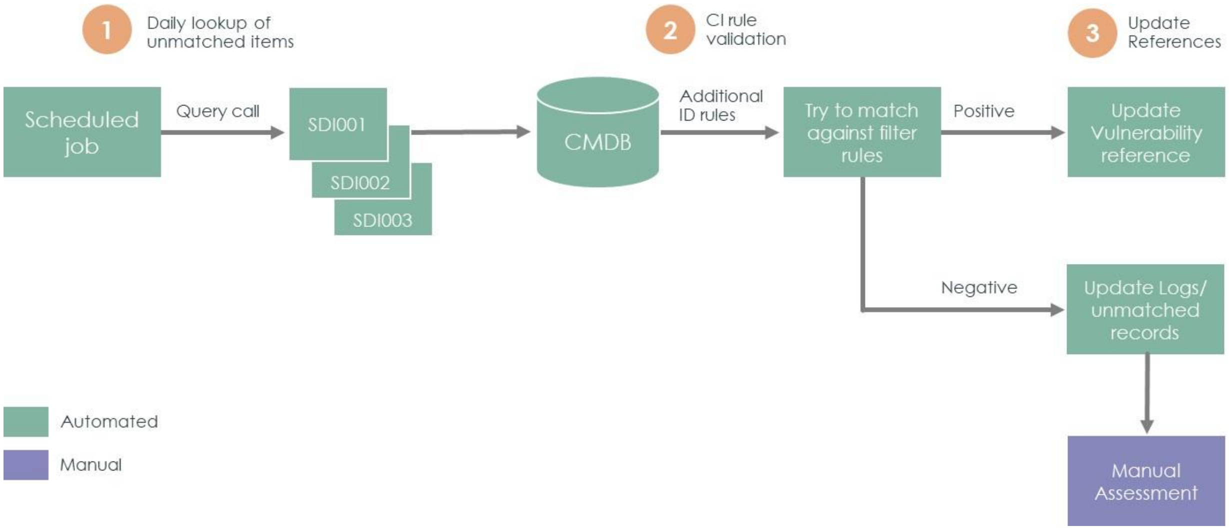
- **The CI records that hold the needed information** – As an example, a virtualized server in our CMDB had a virtual machine CI, a hardware CI, and a DNS-named CI. We have to choose which has the best information for our matching process and workflow.
- **The attributes that tie the system to the vulnerability** – Our scanner creates a host ID that uniquely identifies that system—or an agent ID if it's an endpoint with an agent—that we can add to the CI and use for future matching.
- **The attributes that are unique** – Initially we used IP addresses, which did not work well for DHCP endpoints, because the IP addresses would change frequently. We now focus only on hardware CI classes that have the unique information we need.

We use a three-tier matching process:

- **Scanner ID** – If we can find the scanner ID—either a host ID or agent ID—we know we have an accurate match.
- **Name** – If we cannot match the scanner ID, we look at the name (FQDN, hostname, or DNS).
- **IP Address** – We then look at the IP address, though only if it's an infrastructure-type device, not an endpoint.

If the CI does match any of the above, we create an extended CI.

We then perform a reconciliation process, as shown in the following diagram.



1. Each day the system does a scheduled search for unmatched CIs.
2. We then re-compare the unmatched CIs against the CMDB. If needed, we create additional identification rules and scenarios to enhance the matching process.
3. If we get a positive hit, we update the reference, delete the extended CI, and create a permanent CI. If the result is negative, we update our logs and track it as an unmatched record. Periodically we perform a manual assessment to identify the root cause and fix the issue.

ServiceNow Vulnerability Response benefits

Using ServiceNow Vulnerability Response has resolved the challenges with our previous labor-intensive ITSM-based approach and delivered measurable business value outcomes.

- **90% assignment accuracy** – Nine out of ten times, tickets are assigned to the right team the first time.
- **75% reduction in time to close tickets** – Because tickets are automatically assigned correctly, no time is wasted passing tickets between teams.
- **30X productivity gain** – We can now address vulnerabilities not only for servers, but for all endpoints such as laptops and tablets—a four-fold increase. Because the vulnerability management process is so highly automated, there is no increase in workload for our security team.
- **Improved user experience** – The security team now spends less time on unfulfilling administrative tasks and more time patching. Tribal knowledge is no longer needed, so the learning curve for new employees is shortened dramatically. This has improved job satisfaction and employee retention, which is critical in the tight security labor market.

Learnings and best practices

We learned a great deal in our journey from a manual ITSM-based approach to an automated process using ServiceNow Vulnerability Response:

- **Start with the CMDB** – The CMDB contains all the critical CI information needed for fast, effective vulnerability response, so it's essential to keep the CMDB up to date with accurate data.
- **Optimize CI matching rules** – Establish the right rules and match them to the right CI rules for the environment. It takes a fair amount of time to achieve the proper matching with high accuracy.
- **Create a feedback loop** – Vulnerabilities are a moving target, so it's important to ensure that accuracy remains high. Establish a feedback loop to enable continuous process improvement.
- **Minimize customizations** – Customizations create complexity and make maintenance and upgrades challenging, so stay as close to out-of-the-box as possible. This will also improve accuracy.
- **Automate the exception workflow** – Patching teams don't want to deal with exceptions. Get exceptions approved quickly and off their dashboards, so they can focus on value-added work.

To learn more

- Check out our demo of Vulnerability Response on DemoCenter.
- Read our datasheet and find out more about how our solution connects security and IT.
- Browse the Community Forum for Security Operations to get tutorials and insight on how to setup Vulnerability Response and the latest features.

About ServiceNow

ServiceNow is making the world of work, work better for people. Our cloud-based platform and solutions deliver digital workflows that create great experiences and unlock productivity for employees and the enterprise. For more information, visit: www.servicenow.com.

Now on Now is about how we use our own ServiceNow solutions to work faster, smarter, and better. With Now on Now, we're achieving true end-to-end digital transformation. To learn more, go to www.servicenow.com/nowonnow.