Nellsoft GmbH, Sihleggstrasse 23, 8832 Wollerau – Switzerland
Tel. +41 44 500 16 50, Fax. +41 44 500 16 51
info@nellsoft.com, http://www.nellsoft.com

Checklist for an excellent Vulnerability Management solution

Author: Lionel Gresse
Date: 22nd July 2020

I've been privileged to install and operate Vulnerability Management (VM) solutions for more than 15 years. I had the chance to experiment with many types of deployment. My goal with this document is to share my experience with you. I want you to benefit from these long hours of work. I wish you can benefit from the right choices and the mistakes that I and many others made.

I've listed 23 points to consider when planning for a VM solution, whether for a small business or a large enterprise. Today they are all part of my methodology. Like any pilot before take-off, I go through all of them to ensure a safe and successful VM solution.

Nellsoft GmbH, Sihleggstrasse 23, 8832 Wollerau – Switzerland
Tel. +41 44 500 16 50, Fax. +41 44 500 16 51
info@nellsoft.com, http://www.nellsoft.com

Table of content

1. Vulnerability management is one of the first cybersecurity processes to implement

CSO Online defines vulnerability management as "the process of staying on top of vulnerabilities so that the fixes can be more frequent and effective." Others, such as TechTarget.com, defines it as "a comprehensive approach to the development of a system of practices and processes designed to identify, analyze, and address flaws in hardware or software that could serve as attack vectors."

In my view, vulnerability management is a Quality Assurance process for Cybersecurity. Its goal is to improve the level of security to an acceptable level according to the risk level that the organization accepts. It tells how and where weaknesses exist, what risks they represent, and triggers a change to correct the issue.

It is one of the first cybersecurity processes to implement. It comes at the same maturity level as firewalling or anti-viruses. SIEM and intrusion detection should, for instance, come later.

Vulnerability management comes at level 2 out of 6 in the structured approach of the PCI standard. Firewalling is at level 2, while level 4 includes intrusion detection or file integrity monitoring. Level 1 and 2 items of the structured approach are the first things to implement. Level 4 to 6 items are less pressing than them.

A new CISO should, therefore, implement a vulnerability management process as a priority.

2. Vulnerability assessment, vulnerability management, and penetration testing

Many people use these three words for one another, but they describe three very different things.

A vulnerability assessment is an action to look for vulnerabilities in IT systems, discover them, and assign them a severity level. Most practitioners use automated vulnerability scanners to be more productive and have high-quality results.

Vulnerability management is the process of detecting, sorting, and fixing IT systems' vulnerabilities regularly. It is a standard practice of computer and network security to maintain a constant security level in IT systems. A typical vulnerability management process has four steps:

- Identify: The first step is to discover vulnerabilities through a vulnerability assessment or a penetration test by measuring compliance with specific security standards or getting alerts from information services. The outcome of this step is a list of vulnerabilities on a set of IT systems.
- Analyze: The next step is to measure the severity of vulnerabilities and prioritize them through various factors, such as the criticality of the vulnerable system, the easiness to exploit the vulnerability, and the consequences of an exploit.
- Mitigate: This step fixes vulnerabilities. Usually, IT administrators install patches or change the configuration of the systems. Unfortunately, security patches are not always available or installable. In this case, the solution is to apply a workaround to prevent the exploitation of the vulnerability.
- Learn: This step is
- here to avoid repeating past mistakes and that the organization can learn from them. Should the organization patch systematically? Are there systems with vulnerabilities

lasting longer? Is the know-how of systems administrators a limiting factor? Is it possible to automate part of the process? The goal of the last step is to make changes to improve the first three steps. The identification should become more thorough, the analysis more straightforward, and the mitigation faster.

Vulnerability management is a never-ending loop because cybersecurity researchers discover new critical vulnerabilities daily.

A penetration test is a vulnerability assessment to exploit vulnerabilities to demonstrate their consequences. If a vulnerability assessment could show three critical vulnerabilities on a server, a penetration test would indicate that a malicious person could exploit one of them to download the client database from the Internet. Penetration tests are much more expensive than a vulnerability assessment and take much more time. Their outcome is very dependent on the penetration tester skills. They are done less frequently than vulnerability assessments. A company usually does it once a year, while it performs vulnerability assessments on a monthly or weekly basis.

3. Prepare and maintain an inventory of your IT assets

A vulnerability management process requires a comprehensive inventory of IT systems to be reliable. I've seen many organizations getting compromised despite a vulnerability management process in place. But each time, the vulnerability management process did not include the compromised system. Never scanned, and of course, never hardened, such a server was an easy prey on the Internet.

Inventory management is not part of this document, and I won't talk any longer about it. However, every security practitioner should be aware of the importance of the inventory to manage a vulnerability management process successfully.

4. Prepare a layer 3 topology

The topology has two influences on the vulnerability management processes: during the scanning and for the risk calculation.

Firewalls and IPS can affect network scanners by blocking their network tests. Closed ports or proactive measures such as rogue IP blacklisting or virtual firewalling can prevent network traffic. The scanner cannot test open ports properly, and it can not send its vulnerability tests. It gives a false sense of security. Vulnerabilities are not detected, but they are present. It generates false negatives.

A layer 3 topology is very useful during the analysis of vulnerability data. The network connectivity of a vulnerable computer has an impact on security risk. If the vulnerable device is challenging to reach or has little reach on the IT landscape, even a critical vulnerability creates almost no risk. On the other hand, the same computer on a flat network can provide an entry point giving a boulevard to attack the entire corporate network.

The importance of an up-to-date layer 3 topology is obvious. However, it is generally a real challenge for the security team to get up-to-date and accurate layer 3 topologies. Silos among different groups can be an issue. The network is sometimes not well documented. Frequent changes in the IT architecture give a moving target and an unstable layer 3 topology.

In any case, I always recommend trying to get this information at the beginning of any vulnerability management initiative. It is an essential piece of information to perform vulnerability management.

5. Do you need to keep your vulnerability data on your systems?

When looking at the vulnerability management solutions market for over the past 20 years, two camps separated vendors: cloud-based solutions and on-premise solutions. Interestingly, I have clients on both sides, and both are right in their choice. Why do I agree with both? Because choosing to keep vulnerability data on-premise or in a cloud-based environment is a business and organizational decision. So, let me describe the pros and cons of both approaches.

IT security teams are small compared to most organizations' IT departments. Outside the financial sector, 5000 employees companies often have an IT security team with only 2 or 3 full-time persons. Such a small group cannot spend much time operating its systems. So a cloud-based solution can make sense for them. They will save time. However, cloud-based solutions are more expensive. On the other hand, I've seen security teams using on-premise free solutions because they had no budget for anything else. I always support such a decision because it is better than nothing and improves their IT security.

I have clients with more stringent operational requirements or enough resources to effectively maintain their IT security systems. For them, an on-premise solution makes sense. License costs are cheaper most of the time. They can guarantee their vulnerability management process even if the network connectivity with their vendor is lost. They also have full control over their vulnerability data.

Sometimes, external legal requirements mandate on-premise solutions. For instance, the PCI standard for credit card manufacturers forbids storing vulnerability data in the cloud.

6. One limiting factor: How fast can your team fix the discovered issues?

Clients often ask me: "Lionel, how often should I scan, because I want to know as soon as possible that we have new vulnerabilities." The hard truth is that scanning every day, the same vulnerability for weeks, is a waste of time and resources.

The speed at which your organization will fix a vulnerability is an essential factor to consider. At the time of writing this document, it usually takes a few weeks to patch a critical server. Why so much time? Server managers must analyze the solution. They must verify the compatibility with all software on the server. The business side must agree on a maintenance window. All this work takes 2-3 weeks to do all this in a typical organization.

The only way to get around this is to design systems and applications with maintenance in mind. For instance, car manufacturers create modern automobiles like this: simple maintenance operations such as changing a light bulb or adding brakes fluid are very fast. My vision is that patching servers should be as simple.

7. The two phases of a VM solution: debt payback and cruise flight

When a vulnerability management process starts, the scanners discover hundreds of vulnerabilities. The CISO ends up with hundreds of pages of vulnerability data. It can shock the beginning practitioner.

As IT operation teams have a finite amount of time they can allocate to fixing vulnerabilities, the number of exposures can only disappear at a given speed. Fixing the first batch of vulnerabilities usually takes months. It is what I call the debt payback. During the payback

phase, the critical metric to watch is the difference between newly discovered vulnerabilities and fixed one. It is like paying back a credit: income should be higher than expenses. Recently discovered vulnerabilities should be smaller than what the organization resolves over the same period.
Once you have paid back the vulnerability debt, the cruise flight can start. It is the second phase of a vulnerability management process. During cruise flight, the IT operation team fixes mostly recent vulnerabilities. There are almost no issues older than three months.

8. What are the typical side effects of VM solutions?

In my 20 years of experience with vulnerability management, I had to deal with many side effects. I list them here to help you plan on how to deal with them.

- The first side effect is the human reaction among the IT department.

Some people will blame anything on it because it changes the status quo, and they feel threatened. What is needed here is communication and social skills to make people understand that vulnerability scanning is standard practice. Reliability issues should not stop them. It is the other way around. The vulnerability management process can help detect and reproduce reliability issues. I'll be blunt but honest. A web application that cannot withstand a vulnerability scan made with an out-of-the-shelf solution is not ready to be deployed on the Internet. It will face there a much harsher environment.

- Firewall performance issues.

Firewalls have extensive session tables. However, ending and removing records uses CPU and memory resources. I had the issue mostly with older firewalls (released before 2010) and entry-level ones. When scanning through a firewall, I always recommend getting the brand and model. I always check the session table's size and, more importantly, the speed at which it can enter new entries. I compare it to the number of ports per second that my scan settings will generate. The port scan should only account for a small percentage so that production traffic is unaffected. Again, I encountered only this issue with older or entry-level devices.

- Logs on SIEM and log servers.

Network scans, whether authenticated or not, can generate logs pikes. Firewall logs will increase. Authentication logs will grow. It can become an issue depending on the log settings and the performance of the log server.
The most common issue with firewall logs pikes is the additional amount of firewall logs. The first thing to do is to have special rules on firewalls to limit logging from the scanners' network traffic. I usually recommend setting it up as a preventive measure.
Another common issue is the additional amount of logs on authentication servers such as Active Directories or LDAP servers. It is more challenging to solve. Active Directory does not offer the necessary granularity to whitelist an IP address, for instance. Here, I recommend not to perform large scale network password brute-forcing. Network-based password testing is a great tool to verify default passwords or past corporate-wide administrative ones. However, it is not adapted to verify passwords against an extensive password database. It may generate a significant amount of logs, lockout production accounts, and is less efficient than offline password brute-forcing.

- Account locked out.

Testing passwords can lock out production accounts. If you perform five password verification on an account locked out after three failed attempts, it will lockout during the scanning

process. To prevent this, I recommend using limited lock durations. Most of my clients use something between 15 and 30 minutes. You can also perform such a test at night or during the weekend so that users are not affected.

9. Vulnerability management requires education within your organization

The vulnerability management process will impact two kinds of persons. The obvious ones belong to the IT operations team. They are the ones who will receive vulnerability data and will have to fix it. For them, the vulnerability management process is just additional work and operational risk.
But it will also impact another kind of person: the business owner of the system with vulnerability. This person will have to agree on the maintenance window. He will have to decide on additional expenses such as third-party software upgrades or further external development. And they will have no immediate tangible benefit. They don't understand the large number of issues to fix, the unpredictable costs, and the loss of production loss because of the maintenance windows.
Any CISO will have to win acceptance from both groups. At one point, interpersonal skills replace technical ones.

10. Changes in the inventory should automatically be reflected in the VM solution

Inventory information is key to a successful vulnerability management process. If you don't know what is on your network, you will not scan it; you will analyze its vulnerabilities properly. You will not measure the security risk it may create. You will never take any measures to fix your security issues.
Security teams usually have difficulties in getting accurate and up-to-date inventory information. If they know if an IP address is in use, they don't have the critical pieces of information needed to perform vulnerability prioritization and risk analysis. This information is difficult to gather. Most organizations don't track it. Security teams need to spend time with other IT teams, such as operations and networks, to collect the information they need. Sometimes, they also need to engage with business teams or even outsourcers.
In any vulnerability management initiative, inventory management and data collection must have time and resources allocated. Without it, the vulnerability management will sub perform.

11. Be aware that for many IT systems, maintenance is an afterthought

Most IT systems are designed for functionalities and performance. They sometimes are designed for high-availability. But maintenance is usually an afterthought. It is something that the operational teams are expected to deal with on their own. What is the consequence for vulnerability management? Improvements become limited by an organization's ability to fix issues. Detection is fast and straightforward in comparison.
The IT industry came up with excellent technical solutions to automatically deploy patches or even patch virtually by modifying network traffic. But even with them patching and fixing security issues takes weeks and not minutes.

There are a couple of reasons for this. Let's review them.
Dependencies between components make it impossible to deploy patches or workarounds without breaking things. When security researchers discovered the POODLE vulnerability in SSL v3 and TLS v1.0, many organizations decided to stop using these cryptographic protocols completely. Many of my clients could not do it because database software XYZ or whatever component would not support it. I encounter the same today with web applications build around javascript libraries. They usually are programmed for a specific version of the library with no possibility of upgrading it without modifying the web application. Fixing a critical web application issue such as a SQL injection or XSS vulnerability can take weeks or months in such a case.
Another issue is the infamous "maintenance window with reboot possibility." Many critical applications can't withstand the reboot of one of their components. Unfortunately, the maintenance windows allowing a system reboot can be many months away. The patch installation must be successful during that maintenance windows, and IT teams enjoy the pleasure to work during night-shifts or weekends and public holidays. This inefficiency always surprises me in 2020.

12. Involve the business early on

A vulnerability management process impacts the business team, especially when it comes to application security. Fixing a vulnerability requires maintenance windows and changes that can be costly, especially with web applications. Correcting an issue requires expensive development resources. Installing them will require a maintenance window with programmed downtime.
The business managers who own the impacted systems and applications must be involved early on and understand the exercise's benefits. A non-technical person is not always aware of the risks related to application security. She may perceive the frequent deployment of security patches every as an overkill.
The business owner will have the final word on the deployment of changes and on the maintenance window. It often limits the agility of my clients. They simply cannot quickly get maintenance windows to perform security changes.
Another issue is the budget allocation, especially for application security. When a security issue exists in a web application, the fix involves new development work. It is never free, even if the organization has internal developers. Some of my clients are sometimes postponing changes due to a lack of resources and budget. Working closely with business managers is, therefore, critical when starting a VM initiative targeting web applications. There are important questions to ask them related to maintenance contracts. Do they have a budget allocated? Did they define a warranty related to bugs, where for the first 6 or 12 months of production, fixing security bugs is free-of-charge? Ten years ago, none of my clients had such conditions in their contracts with developers. But time has changed, and recent contracts usually contain provisions for this issue.

13. Delegate access to data

All major vulnerability solutions today have role-based access control. Or to put it in clear-text, users can have customized access to data and functionalities. For instance, a system administrator can scan and analyze his security posture before going live on the Internet. The

internal auditor can track systems' compliance without asking any data to anyone in the IT department. The possibilities are endless.
The main advantage that I see in delegating access to data is that it changes the IT security team's role. They become a trusted advisor instead of being the bad cop with a hard stick. Any system administrator wants to do a great job: he wants to build secure systems that provide the best performance possible. But they lack visibility and advice on security. Having access to the vulnerability data can change this. It makes vulnerability management more collaborative and much less confrontational.

14. Linking vulnerability management and incident management systems

It is a feature that all leading products propose. Interestingly, I've never seen it used in production in 20 years. There are a couple of reasons why my clients never used it. But it also shows where you can benefit from it.

- The separation between production incidents and security events

I've seen IT departments who wanted a clear separation between operational incidents and tasks and security-related ones. In such a situation, the IT department decides not to have any ticket related to vulnerabilities in its central ticketing system. Since the security team is too small to operate a ticketing system, this feature is disabled. Vulnerability tracking is done manually with an Excel spreadsheet, for instance. Such a separation is extremely inefficient because the security team's productivity would benefit from a proper ticketing system.

- Vendors do not provide a sophisticated enough triage mechanism, and it can lead to an avalanche of tickets.

When Microsoft or Adobe publish new patches batches, scanner vendors introduce new detections within a few days. As it takes time to patch and fix them, scans performed shortly after that often show a long list of vulnerabilities in new environments. As all triage mechanisms are built from a risk management perspective, and not from a fixing process one, dozens of tickets of redundant tickets may be open for issues corrected by the same service pack. Such a situation leads to manual work on the ticketing system to close or link together redundant tickets.

15. Define a vulnerability correction policy

I often get the questions, "How fast should we fix critical vulnerabilities?" and "how should it be defined in my policy?". These simple questions are more complex than they seem. The first one is what do I have to do from a regulatory perspective. The second dimension is that all organizations fix as fast as they can. If your operational team needs, on average, three weeks to fix critical vulnerabilities, enforcing one week is unrealistic.
There are many obstacles to fix vulnerabilities. First, patches do not correct every vulnerability. More often than not, a configuration change or a workaround is needed. The "simple" patch may also require a complete upgrade of the system. Many questions can arise at this point. Is the change compatible with the software stack on the server? Is it compatible with the OS version? How about custom code? Finally, change needs planning. A maintenance window is usually required and can be difficult to become. When can the business afford to take down its operation? Can the IT department work at night or during public holidays?

All these questions and many others that come with the correction process makes it essential to define how critical vulnerabilities must be corrected. This policy must balance the need to secure system, to be compliant with external regulations, and at the same time to be doable by the IT team.

16. Follow a “security per design” approach

When I started in the vulnerability management space 20 years ago, the approach was always to install "as is" because of time constraints or to optimize performances. Hardening and security came after, almost on an "as needed" basis. Unfortunately, this approach created systems challenging to secure and patch because they were not designed for this.
Over time, a new approach emerged: secure by design. It had a few essential tenets:

- Patching and improving the security of systems over time is a daily routine. The system must make those changes easier.
- Security is built-in because attacks happen on a 24/7 basis. Improvement in reaction mode due to a new worm or vulnerability should be an exception.

This approach has a strong influence on the vulnerability management process. The old-school approach leads to many findings, complex triage decisions, and a long list of exceptions or pending correction. Almost all systems have issues, and the question is what to fix first.
The new approach leads to much less discovered issues, and the vulnerability management process is similar to a quality assurance one. All systems have very few vulnerabilities. Detections are defects in the hardening processing. With this approach, the IT department becomes more efficient and can fix issues much faster.

17. The particular issues of web application scanning

Scanning application vulnerabilities is very different from system scanning. But I need to explain both concepts first.
Web application scans detect issues in the bespoke layer of the application, such as the java development on tomcat, the custom extensions in a framework like WordPress.
System scanning targets out-of-the-shelf products that make the stack underneath the web applications and that delivers the network infrastructure, the server infrastructure, or the virtualization layer.
Web application scans will navigate the web application and will test out its pages and forms. It will test all the data inputs. There is no navigation phase in APIs, but likewise, it will verify all data entries. The consequence is that the scan will use the web site like a user would: it may trigger orders, post data, delete records. In a nutshell, it can have unwanted side effects. I always recommend verifying the potential effect of a program randomly using the tested website. You must configure the scan to avoid any areas that would create any unwanted effect. Another approach is to test the application in a dedicated testing environment where nothing wrong can happen.

18. Authenticated vs. anonymous scan vs. agent-based scan

When vulnerability management started more than 20 years ago, there are two kinds of scanners: network-based scanners or agent-based ones. The former ones would find vulnerabilities that could be exploited over the network. The later ones focused on local

vulnerabilities whose exploitation required a console or terminal access. Network scans gained in popularity because they were easier to deploy and discovered the most dangerous vulnerabilities. Local scans became a niche market and disappeared.
As the technology matured and new features added, network scanners started to perform local tests. Some vulnerabilities are local and not network-based, such as a privilege escalation. Detecting them requires a certain level of privilege and local access. VM solutions vendors stuck to the "no agent" mantra and implemented authenticated scans. The scanner authenticates with various protocols to perform local tests, improving the overall quality of its results. Some VM solutions can authenticate on more than 20 protocols.
Authenticated scans, unfortunately, have many scalability issues and operational pitfalls. Vulnerability scanners need a high privilege account to perform an authenticated scan. Saving them in such a tool can be a security issue. Over time, the password may expire, or newer equipment may lack the required account. The authenticated scans start failing, and after a couple of months, it is only successful on a minority of systems. A PAM solution (Privileged Access Management), which will serve as a trusted repository of high privilege account for the scanners, can be a workaround. It complexifies the setup and increases costs, however.
In the past two years, there has been a revival of the agent-based scan. It is the solution for the manageability of authenticated scans. They can perform their data collection very fast. They can also decrease significantly the amount of bandwidth used by scanners in large server farms. However, they participate in the "security agent inflation" on servers, a recent security trend where production servers must run an always-increasing number of security agents. I've seen operational managers refusing them because of the issues they may create: additional work to install and maintain them, potential bugs, performance decrease. But it is a new trend to count on.

19. Don’t forget the backup and recovery strategy

This advice is generic to any IT systems, but it also applies to vulnerability scanners. Backups offer protection against configuration errors, hardware failure, and unexpected issues. Even if it is not a main corporate asset, the good practice is to regularly back up and apply best practices: review the back up once a year, keep them in a trustworthy location, and develop a straightforward recovery process.
All these pieces of advice are standard backup practices. However, not all IT departments apply it to their vulnerability scanners. When trouble occurs, they have to reinstall the system from scratch, and the significant downtime they experience is avoidable.

20. Be aware of the time you'll spend on it

VM solutions differ in their ease of use. Generating ad-hoc reports tailored to your specific needs can take more or less time, depending on the solution used.
However, other areas affect the time you spend on it. Different architectures and designs create more or less ambiguous detection and manual clarification. Many VM solutions, for instance, do not provide evidence of their detection. Some, on the other hand, are transparent. With such products, you can read in clear-text the output of the scanner. You are even in the position to be able to reproduce the test by copying and pasting in a telnet session the request sent by the scanner.

Finally, generating KPI can be time-consuming. The difference between the various vendors is significant. When assisting clients with the purchase of a new product, I advise them to test the generation of a few KPIs. They will save time down the road if they chose the right product.

21. Don't forget the out-of-band management of physical appliances

Vulnerability scanners are sometimes in a remote office or difficult to access closet in an industrial building. Out-of-band access is needed to have reliable access when network connectivity is lost. My tip is not to overlook it. Traveling to a remote place to fix a scanner is time-consuming, tedious, and expensive.

22. Don't forget to give visibility to management

Top management needs to understand why they invest in IT. It is accountable just like anyone in the organization. Though information needs can vary from one organization to another as the vulnerability management use cases differ, your management typically needs the following monthly KPIs.

- Total number of endpoint scanned
- EOL components, present, and future (3 months and 6 months)
- Number of critical patches to install and number of vulnerability they will cover
- Number of vulnerabilities on critical systems
- Number of critical vulnerabilities detected for more than 6 months
- Number of new and fixed vulnerabilities over the past 6 months

23. Monitor for performance and availability

Vulnerability management systems, like any IT systems, have limited resources. Performance issues lead to erratic behaviors and never-ending vulnerability scans. Unfortunately, the monitoring of security systems is not a priority and not a widespread practice. Essential monitoring such as free memory, CPU usage, or disk activity is often not done, and the lack of resources is seldom detected.