

Vulnerability Management: From Conception to Execution

By Aurobindo Sundaram

Introduction

In today's security world, where companies are forced to expose hundreds of systems to attackers on the Internet, it is important to have a cohesive, well thought-out vulnerability management implementation. In this article, we discuss how a network vulnerability assessment program should be conceived and executed. In particular, we give the reader tips on what requirements are important to consider, how to obtain support from business management, and how to execute a program to maximize ROI and minimize enterprise risk.

Most companies would like to implement a vulnerability management program to:

- ▼ Have an immediate and real-time sense of the risk exposure of external facing systems (DMZ).
- ▼ To gain an understanding of risk on their internal network.
- ▼ To report, measure, and analyze security risk from a vulnerability assessment angle.

However, most programs are not successful for several reasons:

- ▼ Lack of understanding/articulation of actual business requirements.
- ▼ Trying to do too much too soon.
- ▼ Not selling the program to business executives, so funding is lost after the initial problems are cleaned up.

We will attempt to give the reader a blueprint on how to nurture a vulnerability management system from conception to execution. Although this article covers network vulnerability assessment, it is important to note that the principles hold for other types of vulnerability assessment (e.g. physical, telephony, Web services, etc.).

Why Perform Vulnerability Management?

Although technical staff know that vulnerability management needs to be performed, there are often issues in trying to obtain funding for a program. Questions asked by executives go along the lines of:

1. I just spent \$500K on firewalls, are you telling me that was money thrown away?
2. Why do we need a vulnerability management system? We've never been hacked.
3. We're setting up our systems securely, right? Why do we need vulnerability management then?

4. You mean you're going to try and hack our machines? That's not acceptable.
5. I've heard that these systems are full of errors. What's the point?

It is extremely important to have a solid business case prepared before you approach senior management with requests for funding. Here are some ideas:

1. Vulnerability management allows us to detect and fix our problems before our attackers do.
2. Compliance with best practice is required by our customers. We can use VM to demonstrate this. We can use our program as a competitive advantage in RFP situations. (e.g. if you're a Visa customer and you're hacked and non-compliant with the PCI standard, you face fines up to \$5,000,000).
3. Today, all of our locations do their own (inconsistent) patch/vulnerability management. This solution allows consistency.
4. Firewall and IDS do not prevent attacks. Neither do they stop newly discovered attacks. VM can help us identify our greatest risks, so we can prioritize our response.
5. Mistakes happen. The vulnerability management system will quickly alert us when someone mistakenly fat-fingers a computer setup.

It also behooves you to put numbers (cost savings, efficiencies gained, cost of non-compliance, risk mitigated) into your business case; executives are much more receptive to grant funding if there is a clear ROI.

Understanding Your Vulnerability Management Requirements

We believe there are some basic requirements that everyone should be looking at. There are also supplemental requirements that are really driven by you, the customer, and are "nice to have" rather than required. There are no easy answers for these questions; your own situation will drive your decision. We state them here so you can use them in your RFP creation.

Basic Requirements

1. Must be able to assess a wide variety of products (ChoicePoint, Linux, routers and firewalls, etc.).
2. User authentication implementation (integration with LDAP, certificate based, username/password only)
3. Data retention implementation—are there tools to help manage the amount of data that is generated? Do you want to keep the data in-house, or would you prefer to outsource these operations?

4. Workflow—how can ticketing and case assignment be handled? There must be ways to customize and automatically perform this assignment, as manual assignment is not an option for most enterprises.
5. Role-based access controls—can access to the system be separated by location? By user type? By other methods?
6. User interface requirements—the most convenient interface is Web-based. Fat clients are support nightmares, in general.
7. Asset classification is important to help prioritize response.
8. Enterprise-level scaling is important to understand. Is scaling done horizontally, vertically, or a combination of both? Can network effects (e.g. scan configuration, bandwidth use, etc.) be customized?
9. Scheduling is critical (hands-off operation).
10. Reporting implementation—what types of reports are available, and are they useful to you? We have found that technical and high-level trend reports are most useful, the former for system administrators, and the latter for executives.
11. Product/signature updates should typically be automatic and non-intrusive.

Supplemental Requirements

1. Web vulnerability assessment may be important to you. Only some vendors do a passable job of this. There are pure-play vendors (e.g. WebInspect by SPI Dynamics) that perform only Web assessment.
2. Integration with trouble ticketing systems. This is a basic requirement for large enterprises. For smaller companies, this may be nice to have.
3. Integration with other point products (SIM, etc.)—As your program matures, you may want integration with your MSSP, your SIM product, and your patch management product. This is supplemental in that it is rarely required before phase 3.

Understanding Resource Requirements

It is tempting to think that once the system is set up with scheduling, there is little need for resources. Nothing could be further than the truth, however. In our experience, we have found that there are significant requirements before rollout, during initial rollout, and during the remediation phase. Some of these are mentioned below. A mistake many practitioners make is requesting too few resources, or not creating a collaborative atmosphere from the start. These efforts, unfortunately, are doomed to failure.

Before rollout

1. Business management: For approval of purchase as well of the program itself.
2. Change management: To schedule the initial scans in conjunction with operations.
3. IT operations: To install the system, as required, and to monitor the applications during pilot scanning.
4. Database operations: To install the database (if necessary), set up adequate data storage and data management scripts, etc.
5. Network operations: To assess network traffic during scans, and help tune the system for optimum efficiency without impacting operations.
6. Other: You will need to collaborate with various parties to define the exact reports they require, to obtain the list of hosts to be scanned, to define the initial scan options, to define system access rights, to define asset risk weights, etc.

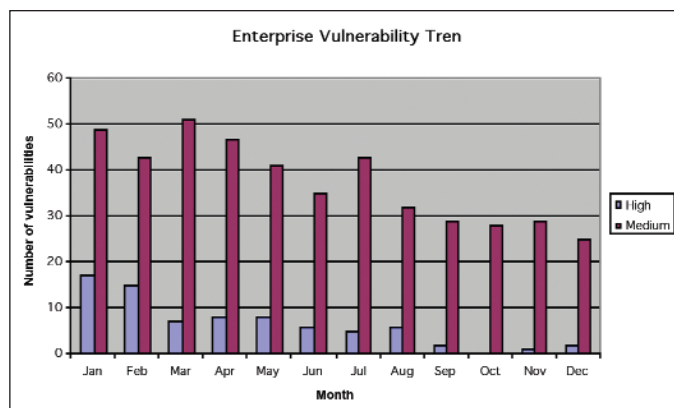


Figure 1: Risk management trend report

Pre-VM:	
Average number of IDS logs analyzed:	3,000,000/mon
Average number of trouble tickets opened:	750/mon
Average time to close:	30 minutes
Cost:	\$11250/mon (at \$20/hour)
Post-VM	
Average number of IDS logs analyzed:	3,000,000/mon
Average number of trouble tickets opened:	250/mon
Average time to close:	30 minutes
Cost:	\$3750/mon (at \$20/hour)
Cost savings/mon = \$7500/mon	

Figure 2: Synergies on integration with SIM system

Initial rollout

(2), (3), (4), and, on occasion (5) from the previous section will be required.

Remediation phase

This phase is possibly the most resource intensive of the three phases, depending on the numbers of vulnerabilities discovered. Much of the resources required here will be coordination and system administration related. As with most programs, the hard part is not generating the reports and assigning them to people; it is getting already overworked system administrators to stop, test proposed fixes thoroughly, and install them in production in a timely fashion.

1. System administrators: In general, you will need to help system administrators, many of whom are not necessarily security savvy, through the process. You can expect a lot of time to be spent by administrators testing fixes in development, and applying them in production.
2. QA: In enterprise-class systems, due to the need for regression testing, the QA department will have to pitch in substantially, often on short notice, and with other projects already on hand.
3. Project management: This is critical. You will need a dedicated resource to create tickets, follow-up on vulnerabilities that have not been fixed, and co-ordinate meetings to discuss findings.
4. Program management: It is very important that you carefully ensure that the program manager does not try to do too much too soon, is sensitive to the needs of the business, sets achievable goals, and motivates the virtual team to perform at its peak.
5. Other: Apart from the explicit resources above, various soft resources such as IT operations and change management are needed in this phase as well.

Step by Step: Implementing Vulnerability Management

This is not a comprehensive step by step, but it gives the reader some important steps to follow while conceiving a Vulnerability Management solution.

1. Write down your requirements (both business and technical). Decide carefully what you really need.
2. Create your evaluation requirements and test cases (how you decide if a product satisfies your criteria). In this space, it is very important to be clear about whether you want a product or a service—this can affect your decision drastically.
3. Create and issue an RFP (pick the 4-5 most suited vendors). Ensure that you invite not only security, but also operations to the decision meetings.
4. After the evaluation, pick 2 vendors to bring in-house and run against your test cases. In particular, make sure you test stress conditions (data overload) against the test system.
5. At this point, ensure that you carefully study and understand licensing options in your scenario. In addition, also factor in resource requirements (e.g. service solutions do not require database resources; however, service solutions are also not as customizable)
6. As part of the pilot, also make sure you talk to external sources as well as reference customers from both vendors to judge actual level of effort in implementation.
7. Do not try to go too fast. We suggest the following plan of action
 - a. Phase 1: (First 6 months): Ensure that you scan all Internet facing systems at least weekly. Ensure that all "High" risk items are addressed within a predefined time. Do **not** consider "Medium" and below items.
 - b. Phase 2: (Months 6-12): Consider adding selected Medium risk items to be resolved. Start scanning selected high risk internal systems, where "High" risk items are addressed in a predefined time.
 - c. Phase 3: (Month 12+): Start using your system to demonstrate compliance with regulation (e.g. Sarbanes-Oxley), business (e.g. Visa/Mastercard PCI standard). Expand the number of hosts scanned as appropriate.

Some things vendors will say to you (and what they really mean in italics):

1. We have 5000+ checks. *But on many of them, we're really not sure whether our results are right or wrong.* A critical feature of these solutions is how few false positives they have. It's better to have fewer checks that are correct than more checks that are incorrect. For the most part, writing checks is easy (like writing AV signatures). Making sure each check works flawlessly without false positives/negatives is much harder. Ask your vendor for more details.
2. Our product is plug-and-play. *If you require the simplest solution possible with no additional features.* Always make sure you understand how long the simplest implementation and how long the first functional implementation will take. They're not the same.
3. Our product has 250+ reports. *Most of which you will never use.* It's important not to confuse quality and quantity. A handful of good reports are better than a hundred poor ones. Be cognizant of what you really want, and more importantly, need in terms of reporting.

The Marketplace

There are several companies in this fiercely competitive marketplace. Some of the better-known ones are:

1. Foundstone/McAfee (product and appliance based, stores data locally)
2. QualysGuard/Qualys (service and appliance based, stores data offsite)
3. nCircle (appliance, "continuous" assessment, stores data locally)
4. Retina/eEye (product, stores data locally)
5. NetIQ, BindView and others have products as well
6. TruSecure/Cybertrust provides a complete assessment service

Your requirements should drive your decision on purchase. Home-written systems (using Nessus, nmap, etc.) are hard to maintain in the long run. We recommend using a product vendor, despite the seemingly additional cost.

4. Our product is completely customizable. *But we provide no templates, so you'll have to do the entire setup yourself.* It is important to have something that works adequately out of the box. Your goal is not configuration, it's risk management.


Demonstrating Value

There are several ways to demonstrate the value of your program. Financial value can be demonstrated using metrics such as:

1. ROI: We spent \$Xk on this solution, but paying a consultant for each scan would have cost us \$Yk. We saved (\$Y - \$X)k.
2. The cost to certify our systems are Visa/Mastercard PCI compliant would have been \$Xk. Using our system, we saved \$Zk.
3. By integrating our solution with our SIM solution, we now have to respond to 50% less incidents, therefore X less tickets a month opened, therefore, savings of \$Yk monthly.

Risk reduction can be expressed in terms of qualitative metrics such as:

1. Trend reports of High severity incidents to show how the risk has decreased. This is particularly useful if a recent vulnerability has been exploited on the Internet, and your VM helped you patch it before you were exploited.
2. Using your firewall and IDS logs to show that after your systems were patched, attackers tried to exploit it. This is a case of explaining "If we hadn't had <VM>, we would have been compromised."
3. Trend reports in (1) based on asset classifications and severity are even more useful.

Some examples of value demonstration graphs are shown in Figures 1 and 2. It's important to note that none of these graphs reflect any real-life data. 

Aurobindo Sundaram, CISSP, CISM, is Director of Network Security at ChoicePoint.