Secure Coding.
Practical steps to defend your web apps.

Vulnerability Remediation

GIAC (GSSP-JAVA) Gold Certification

Author: Chad Butler, chad.butler@gmail.com
Advisor: Rodney Caudle

Accepted: November 18, 2013

Abstract

In today's era of rapid release development projects, finding vulnerabilities is not difficult. The WhiteHat Website Security Statistics report, released in May of 2013, states that 86% of websites tested by the company had at least one serious vulnerability which would allow an attacker to compromise all or part of that website. These vulnerabilities required an average of 193 days for remediation. (WhiteHat Security, 2013)

Many security professionals have spent countless hours finding critical vulnerabilities only to have their reports treated with disdain and apathy by those whose skills and assistance is required to remediate the risk. The security landscape is replete with individuals and tools that can quickly find security vulnerabilities. Unfortunately, there are comparatively fewer individuals who can effectively manage the mitigation process to reduce business risk.

This paper explores methods for managing the vulnerability mitigation process and winning others to the cause. Specifically, it explores how to apply principles of effective human relations that have stood the test of time.

These principles, if applied consistently, will lead to healthier relationships between security and development teams, shorter vulnerability remediation times and ultimately reduced risk for the business.

1. Introduction

In today's era of rapid release development projects, finding vulnerabilities is not difficult. Most web application vulnerability scanners will produce long lists of vulnerabilities in short order. Finding vulnerabilities has become relatively easy, but fixing those vulnerabilities can be inordinately difficult. The security professional has to navigate tricky political and interdepartmental relationship issues to align the resources necessary to bring effective remediation efforts to bear.

2. The Challenge

A penetration test from a reputable security firm can be an expensive endeavor. Often times these penetration tests deliver reports that identify serious security deficiencies. This is often exciting to security teams. They conclude that the findings contained in the report are so compelling that they will finally receive the funding and recognition they deserve. Perhaps shock and a little dismay follow when the report is treated with apathy. The security team that was previously celebrating victory is left wondering what went wrong.

Unfortunately, the preceding scenario is a common occurrence. The following sections explore some of the reasons that this challenging phenomenon occurs. This report will address each of these challenges with practical solutions and recommendations.

2.1. Throw it Over the Wall

The term, “Throw it Over the Wall” is one that many security engineers are familiar with and have come to despise. In the software engineering sense it has come to be synonymous with the practice of promoting or throwing applications from development into production with little documentation and communication. In this practice, developers are focused on features and deadlines and they leave other aspects like, performance, maintainability and security to the operations team. This frequently results in finger pointing when problems arise.

Security teams despise this practice because of the negative impact it has on security and operations. However, those same teams often throw security vulnerabilities over the wall with little explanation and documentation. Often times this is caused by the fact that most security teams are overburdened and cannot spend the time required to properly communicate vulnerability findings and the required remediation work. Regardless the reason for the lack of communication, the end result is developers who feel that they have been given a task that was not important enough to be properly documented and communicated. At best, these development teams will provide a solution that meets their limited understanding of the problem and solution if they work on it at all. These solutions rarely meet security requirements and often result in rework.

2.2. R.O.U.S

"That's the fire swamp! We'll never survive!"
"Nonsense! You're only saying that because no one ever has."
-The Princess Bride (Goldman, 1987)

Vulnerability scanning tools tout their ability to generate pre-built reports that document findings, provide remediation advice, and simultaneously satsify compliance requirements. However, these reports consistently miss the mark. The summary reports do not have the necessary context to help executives understand the risk. Furthermore, the detailed reports are typically high in content and low in value. Web application vulnerability scan reports from automated scan tools often include full HTTP requests and responses along with every javascript file and HTML comment discovered. This results in Reports Of Unusual Size (R.O.U.S), typically in the form of 500+ page PDF files. They can be the impenitrible barrier that prevents vulnerability remediation due to the fact that the responsible team does not have time to dig into the report and pull out the vulnerability findings.

2.3. False Positives

False positives are a significant issue in the remediation process. It is tempting to discount false positives as a simple mistake. However, false positives show a lack of quality and lead the report recipient to justifiably question all findings in the report. False

positives can be extremely harmful to the remediation workflow. As an analogy, consider the emotional response that is brought about when finding a single strand of human hair in your food. The plate may have been masterfully prepared and the presentation immaculate but a single hair taints the plate and makes it virtually impossible for the customer to enjoy the meal. False positives create a similar emotional response for developers. The entire report is potentially called into question until it has been further validated.

2.4. Exploitability Trap

Another problem that seems to plague the remediation process is the exploitability trap, a term introduced by the founders of Fortify, Brian Chess and Adam West. (Chess & West, 2007) This trap exists when developers refuse to fix problems unless it can be proven exploitable. There are several flaws with this mentality. First, developing exploits takes a considerable amount of time and effort. That time would be better spent explaining the issue, managing the remediation process, and finding more vulnerabilities. Second, it requires specialized skills to develop functioning exploits. Just because the security team cannot prove that a particular vulnerability is exploitable does not prove that it is safe. The security team has a finite amount of time and resources. The organization's adversaries are not necessarily limited by these traditional constraints. Finally, the conditions that would make a particular vulnerability exploitable may not be in place at that point in time. As business needs evolve and technical controls are modified and new features introduced, the organization may inadvertently expose vulnerabilities that were previously not exploitable. (Chess & West, 2007)

There is certainly value in demonstrating exploitability, particularly when it can be used to show that the "worst case" scenario is possible. These scenarios should be used selectively to avoid getting caught in the exploitability trap.

2.5. Lack of Accountability

The problem of lack of accountability is one that impacts all aspects of security. If a developer can check in shoddy and insecure code that cannot or will not be attributed to him/her, what extrinsic motivation remains to help ensure code quality? Unfortunately, intrinsic sources of motivation can be quickly compromised when crunch time comes and developers are encouraged to do whatever it takes to meet deadlines. This is compounded by the fact that many development teams reward the ability to ship software. Depending on the maturity of the SDLC, there may be few, if any requirements for quality and security. Regardless, security testing takes time and resources which is seen by many as a direct conflict to a development team's ability to ship software quickly.

2.6. Lack of Understanding

Probably the most significant issue that impedes the ability to quickly remediate security vulnerabilities is a lack of understanding, both on the part of development teams and security teams.

Development teams are often unaware of the risks posed by security vulnerabilities in their software. This is, in large measure, caused by the prevailing attitude that security is an optional activity that comes after all features have been developed. Many computer science programs fail to teach security as a part of the curriculum. In fact, many programming books and manuals place the security chapters at the end of the book, which serves to further solidify the mentality that security comes last. It isn't that developers set out to write vulnerable code; they simply don't understand the risks. Here is a somewhat typical example of a conversation that illustrates this point:

> ***Developer***: We created this cool form that allows the user to upload a photo.
> ***Security Engineer***: That's great. What have you done to secure this form?
> ***Developer***: What do you mean? What could go wrong?
> ***Security Engineer***: Well, for starter's, someone could upload a huge file and fill up the filesystem. Or, perhaps someone could upload a shell and take control of the server.

Developer: Why would anyone do that? Besides, that isn't in the project requirements.

This is admittedly a very simplistic example, but it illustrates the fact that many developers simply do not consider security implications during their work.

It is also common for security teams to lack understanding of the development process and the level of effort required by the solutions they suggest. Unfortunately, the remediation advice provided by vulnerability scan reports, penetration test reports and best practice guides rarely include the context that would help the security engineer understand the impact and level of effort required to implement the fix. When security teams rush in with guns blazing and assuming they know how best to fix the issue, relationships are damaged. It also places the security team in the unenviable position of being the scapegoat for any and all issues that can conveniently be blamed on the changes they were forced to implement. Consider the following exchange:

CEO: Why did the application go down? We lost significant revenue during the outage and our largest customer is outraged.
Development Manager: The security team forced us to implement technology XYZ, which caused significant performance issues and caused the application to crash.
Security Manager: <Diligently working on the next 500 page PDF report and absent from the conversation>

There are several significant problems that detract from an efficient mitigation workflow. This paper includes details on some of the most prevalent and significant factors. The following sections will provide practical solutions and recommendations that security teams can implement in order to facilitate a more efficient vulnerability remediation program.

3. Solution – Vulnerability Management Plan

To have a successful outcome, it is critical that security teams document the vulnerability management plan and get executive support. It is true that you cannot wait for the perfect plan and proper executive support to begin remediation. However, it is important to understand that this is a crucial and strategic step that must be prioritized.

NIST has created in-depth guidance for organizations that are creating a vulnerability management plan (SP 800-40). Many of the recommendations in SP 800-40 are targeted at IT systems (e.g. workstations, servers, and network devices). However, many of the concepts and principles can be applied to a vulnerability management program designed for applications written in-house. (NIST, 2005) At a minimum, an organization's vulnerability management plan should include a designation of roles and responsibilities, remediation goals, a process for risk ranking and a process for risk acceptance. These elements answer the who (roles), what (risk ranking), and when (remediation goals) questions of vulnerability remediation. They help facilitate accountability in the remediation process. It is also important to establish a metrics program to ensure that the effectiveness of the program can be measured and demonstrated. NIST states that all organizations should measure the effectiveness of the vulnerability management program and apply corrective actions. (NIST, 2005)

3.1. Risk Ranking Process

There are many risk ranking methodologies and processes. At the end of the day, they all have the same goal, to provide a simplified understanding of risk that can be used to make business decisions and prioritize remediation efforts. The methodology must produce results that are relevant to the business. The following is a simplistic formula used to calculate risk:

Risk = Threat Likelihood (Probability) x Threat Impact (NIST, 2013)

Quantitative risk analysis certainly has its place, but it is difficult to do in a defensible manner. The problem is that determining the chance of attack is very difficult. Professional actuaries have devised complex mathematical formulae and models to help understand the probability of events that cause insurance claims. They use these models to set insurance premiums and to help ensure that the insurance company remains profitable. (Bureau of Labor Statistics, 2012)

Likewise, casino mathematicians use statistical calculations to ensure that casinos are profitable. Because of these mathematical calculations, the casino can modify variables such as the payout percentage to ensure that the casino maintains their "house edge" and, in the long run, always wins. (Hannum, 2012)

The form of quantitative risk analysis that both actuaries and casino mathematicians effectively use to help protect their respective business from risk relies on large populations of customers. They are not trying to predict a single event, but rather ensure that a disaster or large payout does not cause financial insolvency.

The problem with applying similar models to security risk is that we simply don't have sufficient data to make those accurate calculations. Determining the probability of achieving a certain number with a single six-sided dice roll is a much different equation than determining whether or not an adversary will be sufficiently motivated to attempt to exploit a vulnerability and whether or not that attack will be successful.

A penetration test can be a great validation of the risk analysis. The goal of a penetration test is to prove that vulnerabilities are exploitable and to explore the full damage potential. This activity helps validate and communicate the actual risk associated with a threat. The adage, "a picture is worth ten thousand words", certainly holds true in risk assessment as well. (University of Regina) Nothing communicates that a specific vulnerability could lead to system compromise like producing a screenshot that shows the penetration tester was able to get shell on the system.

The purpose of this paper is not to provide an exhaustive treatment of the various risk ranking methodologies available. However, it seems relevant to the topic at hand to provide a brief overview of the risk ranking methodologies that seem to be most applicable.

3.1.1. DREAD

The DREAD system is part of a risk determination methodology that was once used at Microsoft and which has been included in several Microsoft publications regarding the Security Development Lifecycle and threat modeling. DREAD is an acronym that contains several categories used to help quantify risk. The following example comes from OWASP's Threat Risk Modeling page:

- **Damage Potential** - If a threat exploit occurs, how much damage will be caused?
    - 0 = Nothing
    - 5 = Individual user data is compromised or affected.
    - 10 = Complete system or data destruction
- **Reproducibility** – How easy is it to reproduce the threat exploit?
    - 0 = Very hard or impossible, even for administrators of the application.
    - 5 = One or two steps required, may need to be an authorized user.
    - 10 = Just a web browser and the address is sufficient, without authentication.
- **Exploitability** – What is needed to exploit this threat?
    - 0 = Advanced programming and networking knowledge, with custom or advanced attack tools.
    - 5 = Malware exists on the Internet, or an exploit is easily performed, using available attack tools.
    - 10 = Just a web browser
- **Affected Users** – How many users will be affected?

  - 0 = None
  - 5 = Some users, but not all
  - 10 = All users

- **Discoverability** – How easy is it to discover this threat?
  - 0 = Very hard to impossible; requires source or administrative access.
  - 5 = Can figure it out by guessing or by monitoring network traces.
  - 9 = Details of faults like this are already in the public domain and can be easily discovered using a search engine.
  - 10 = The information is visible in the web browser address bar or in a form.

In the case of DREAD, the numerical risk rating is derived via the following formula:

Risk = (Damage + Reproducibility + Exploitability + Affected Users + Discoverability)/5 (OWASP, 2010)

Certainly, DREAD seems better than blindly accepting the High/Medium/Low rating that is churned out by your scan tool or penetration testing vendor since it forces the security analyst to consider business factors like system criticality and impact to the business. However, DREAD has also received a fair amount of criticism. One of the initial authors of the STRIDE/DREAD methodology, David Lipner, admits that it was designed without a lot of academic rigor and that it doesn't stand up well from a scientific perspective. (LeBlanc, 2007) According to a presentation given in 2005 by a former Microsoft security team member, Microsoft has revised their threat modeling process and no longer uses DREAD. The presentation cites the fact that DREAD is highly subjective, a criticism that is true of many risk assessment methodologies. (Seller, 2005) It is worth mentioning that despite the criticisms that exist there is still a great deal of value to be derived from DREAD.

"The fact that it isn't a rigorous classification doesn't mean that it isn't useful – it is useful in helping people focus the debate about what to do about some specific problem. Outside of the ivory towers of academia, there's a lot of very useful things that don't pass muster with a strict academic standard." (LeBlanc, 2007)

3.1.2. CVSS

The Common Vulnerability Scoring System or CVSS was created by the NIAC Vulnerability Disclosure Working Group, which was started by the US Department of Homeland Security. One of the primary benefits of the CVSS system is that it provides a normalized severity rating which is generally understood in the industry. Most public vulnerability disclosures also include a CVSSv2 rating. This system provides a convenient method of communicating risk across organizations.

CVSS is considered more complex than DREAD and could therefore be considered unsuitable for an application security program where hundreds of vulnerabilities may need to be ranked. (OWASP, 2010) Fortunately, NIST has provided the CVSS Calculator to simplify the process. (NIST) The following outline provides a brief overview of the components required to produce a CVSS score:

- Base Score
    - Access Vector – How the vulnerability is exploited
        - Local – requires local access (e.g. physical access to system or local shell account)
        - Adjacent Network – requires access to local network (i.e. broadcast/collision domain)
        - Network – requires any network access. This is often referred to as "remotely exploitable".
    - Access Complexity – What is required to access the vulnerability
        - High – specialized access conditions exist (e.g. race condition)
        - Medium – Some additional access requirements (e.g. source IP address limitations)

        - Low – No special access requirements
    - Authentication –
        - Multiple – Attacker is required to authenticate two or more times, even when the same credentials are used.
        - Single – Attacker must authenticate once
        - None – No authentication requirement
- Temporal Score
    - Exploitability – The current state of exploitation techniques
        - Unproven – No exploit code is available (theoretical)
        - Proof-of-concept – Proof-of-concept code is available but not practical for widespread use
        - Functional – Exploit code available which works in most situations
        - High – Automated exploits are possible
    - Remediation Level
        - Official Fix – A complete vendor-provided solution is available
        - Temporary Fix – An official, but temporary, fix is available
        - Workaround – An unofficial, non-vendor solution is available
        - Unavailable – No solution available
    - Report Confidence
        - Unconfirmed – A single unconfirmed source, or multiple conflicting sources
        - Uncorroborated – Multiple sources agree broadly, but some uncertainty about the vulnerability remains
        - Confirmed – Acknowledged and confirmed by the vendor
- Environmental Score

- Collateral Damage Potential
  - None – No potential for loss of property, revenue, or productivity
  - Low – Slight damage to assets, or minor loss of revenue or productivity
  - Low-Medium – Moderate damage or loss
  - Medium-High – Significant damage or loss
  - High – Catastrophic damage or loss
- Target Distribution
  - None – No target systems exist
  - Low – 1%-25% of systems at risk
  - Medium – 26%-75% of systems at risk
  - High – 76%-100% of systems at risk
- Security Requirements
  - Low – Loss of security is expected to have a limited effect on the organization
  - Medium - Loss of security is expected to have a serious effect on the organization
  - High – Loss of security is expected to have a catastrophic effect on the organization

Whatever methodology you choose make sure it is well thought out. It needs to be defensible and able to withstand the subjectivity that naturally creeps into this process. It also needs to add value to the business and aide business leaders in making informed decisions.

As organizations consider which methodology to use, a good rule of thumb is to consider whether or not the risk ratings need to be shared outside of the organization. If the

organization does not need to share those ratings with outside parties, a lighter weight methodology, like STRIDE/DREAD may be perfectly acceptable. If ratings need to be shared outside of the organization, CVSS will likely be the better choice.

3.2. Remediation Goals

Remediation goals help communicate the process to stakeholders and outline expectations. Like other types of goals, these should also be S.M.A.R.T. (Doran, 1981)

Specific - The goals should be specific about what must be remediated. Preferably, these goals should also include some component that ties the remediation goals back to an organizational goal. For example, if the organization has a goal to maintain superior levels of customer service, the remediation goal may reference this goal and the fact that keeping the customer's data safe is critical in maintaining superior customer service.

Measurable – The goals also need to be measurable. If you cannot measure remediation progress, it is impossible to know when the vulnerability has been closed. At a minimum, an organization should be able to measure the time between vulnerability discovery and closure. It is also beneficial to be able to measure the risk of the vulnerabilities discovered.

Attainable – The goals need to be attainable. When determining whether the goal is attainable, the organization needs to consider factors such as release cycles and available testing resources. If the security team asks for an emergency hot fix for every vulnerability finding, they may find that they are treated like Chicken Little.

Relevant – Remediation goals should be relevant in terms of reducing risk. The goals should focus the quickest response on those areas of the system or application that represent the highest risk.

Time-bound – Remediation goals need to have deadlines. They should also define who gets notified and the penalties that are incurred when remediation deadlines are missed.

3.3. Secure Development Standards and Requirements

Finally, it is essential that the organization produce a documented list of secure development standards and security requirements. These standards and requirements should be developed in collaboration with the developers who will be required to follow them. Ideally, security should specify the requirements and allow development teams to identify the solutions, technologies and practices that satisfy these requirements. A development team who views these requirements as edicts will be much less inclined to follow them than a team that is involved in the initial design process. As vulnerabilities are discovered and reported, the remediation advice should reference these standards and requirements. This practice helps train developers and reinforces the organizations standards.

3.4. Risk Acceptance Process

As time passes a situation will arise that inevitably will require a discussion about how exceptions to the process should be handled. If the fix causes more business risk than it solves, this is a good candidate for risk acceptance.

When the risk acceptance process is informal or undocumented, the organization may find that technical contributors and individual project teams accept large amounts of risk because the remediation effort is unfunded and causes problems in their current work. This needs to be avoided. Formalizing the risk acceptance or exception process helps ensure that the risk decisions are handled by the individuals who actually own the risk. It also helps ensure that the reasons for accepting the risk are evaluated by those business owners so that they can choose to either accept the risk or allocate more resources.

3.4.1. ASVS

The OWASP Application Security Verification Standard (ASVS) is a very interesting approach to the process of developing standards and requirements. The project states that the key benefit is using a common yardstick to measure an applications' trust, ensuring that results are repeatable, and ensuring that expectations are clearly set. (OWASP, 2009) Additional benefits of ASVS include the fact that OWASP provides the following great resources, which cross-reference and support each other. The following list includes several notable examples:

- Top 10
- Developer's Guide
- Code Review Guide
- Testing Guide
- Enterprise Security API
- Open Software Assurance Maturity Model
- Prevention Cheat Sheets

OWASP has provided a wealth of resources that help explain and simplify the process of implementing application security. Due to the quality of the resources OWASP provides, they have become recognized as a leader in the application security space. They are frequently referred to as a standard de jure for securing applications. One notable example of this is the inclusion of the OWASP Top 10 in the PCI-DSS requirement 6.5. (PCI Security Council, 2010)

3.4.2. ISO/IEC 27034-1

The ISO/IEC 27034-1 is a standard developed by the same ISO/IEC subcommittee (SC27), which published other well-known information security standards such as ISO/IEC 27001 and 27005. (ISO) Like other ISO standards published, ISO/IEC 27034-1 is a process based approach. It is implemented by defining an application's

target level of trust (TLT) and then assuring that the design, development, test, and operational policies exist to ensure that the TLT is met. (Pescatore) Microsoft has been an early and vocal proponent of ISO/IEC 27034-1 and has encouraged other organizations to begin aligning their software development and acquisition policies to this standard. As ISO releases additional parts of ISO 27034 and visibility is increased it is reasonable to expect that these requirements will begin to be included in more RFPs. (Pescatore)

3.4.3. Microsoft SDL

Microsoft's SDL was a direct outcome of the Trustworthy Computing Memo sent by Bill Gates in the wake of the Code Red and Nimda worms. In this memo, Bill Gates identified security vulnerabilities as a serious threat to Microsoft's image and reputation. He directed the organization to give security highest priority. (eWeek, 2002) Microsoft has published the SDL as well as a number of tools to help organizations implement the SDL within their development processes. Among these resources is a simplified implementation guide, intended to allow companies to adopt essential elements of the SDL more quickly. Finally, Microsoft announced in May 2013 that SDL meets or exceeds the requirements for ISO/IEC 27034-1 compliance. (Lipner, 2013)

It is hard to overstate the importance of documenting software security policies and processes and gaining executive support. This is a key initial battle that must be won for subsequent software security initiatives to be effective. Unfortunately, there are no "decision tree" checklists to help an organization select the right secure development standard. The security team will need to understand regulatory requirements as well as factors that are significant to the organization and its customers and partners in order to select the right standard for the organization.

4. Solution – Knowledge

In order to effectively manage an application security and vulnerability remediation program the security team must be knowledgeable and competent regarding multiple issues. A team member who attempts to bluff his or her way through an explanation of the vulnerability and how to fix it destroys credibility and trust. Furthermore, a team member who does not understand his or her audience or how the technical risk translates

to business risk may in fact harm the organization's ability to operate a viable business. There are several key areas of knowledge that must be gained and maintained in order for the security team to be effective in this endeavor.

4.1. Know Your Audience

Much of the success of security initiatives is based on the security team's ability to sell the benefits of security. The ability to understand your audience is crucial in helping the security team tailor the message to appeal to the needs and goals of executives and stakeholders. Security professionals who charge ahead without considering their audience run the risk of communicating to executives that the security team feels that they know, better than the executive leadership team, what is best for the organization.

A wise security professional will take time to understand his or her audience before crafting and communicating their message. It is important to understand what goals, priorities and concerns your audience has. It takes time to do the due diligence and research necessary to understand these motives. A security professional that is more tactically focused may find it difficult to take the time and do the research when they have the assumption that they already know the best answer.

People have the intrinsic need to be heard and validated. When communication parties fail to listen, arguments and misunderstandings inevitably ensue. One example of this is what occurs on political talk shows. When individuals with opposing political views take their platforms it becomes clear that there is very little listening going on. When the opponent does listen, it is generally only long enough to find a point that can be used for counter argument. The rest of the time is consumed by formulating a response rather than listening and understanding. At the end of the debate, both sides have done at least twice as much talking as listening and have left the debate more strongly convinced of their position and more vehemently against the opposing position. Taking the time to understand, hear and validate the concerns of your audience will open the way to more effective communication of your message, even when your message contradicts your audience's viewpoint. (Covey, 1989)

4.1.1. Appeal to Nobler Motives

In the book, How to Win Friends and Influence People, the author introduced a communication technique that he refers to as appealing to nobler motives. The concept is actually quite simple. The author explains that people who do things that society classifies as "bad" actually feel justified in their actions. Whether they have justified their actions based on the injustices dealt to them or because of some other reason, these individuals are operating under a rationale that allows them to do what they do and escape with a clear conscience. (Carnegie, 1936)

This concept applies very well to security. If you were to ask the worst security offender in your organization if they felt security was important you may be surprised by the answer, particularly if that conversation occurs in front of the CEO, board of directors or a customer. Security has gained sufficient visibility in the news media, that nobody wants to be identified as the source of a breach. So why do these individual who, in public, claim to be ardent supporters of security act inconsistently with security policy and best practice? Perhaps they feel justified by the fact that their budget was cut earlier that year and they no longer have the resources to effectively do their jobs. Or, perhaps they feel that the organization does not place critical emphasis on the importance of security as evidenced by the fact that they have never been reprimanded for security failures in the past.

Often times when security professionals speak amongst themselves, they verbally attack the motives, credibility and work ethic of those members of the organization who cause security vulnerabilities. When security professionals hold this viewpoint, it becomes extremely difficult to communicate without also sending nonverbal messages of arrogance and disdain. These nonverbal messages can destroy the effectiveness of the message before it is even communicated.

It is much better for the security professional to acknowledge this fact and appeal to nobler motives. Start by stating their noble motives and acknowledging the challenges they face. For example, "I know that you are committed to protecting this organization and our customers. I also know that it is difficult to find the time and resources to perform all of the security tasks that need to be done." You can be confident and

completely honest in stating these things because of two safe assumptions. First, people generally want to do the right thing and protect their employer and customers. If they are not committed to protecting their employer and customers, their employment contract should be terminated immediately. Second, most employees do not receive all of the resources they need to do their jobs completely. The reality of limited resources means that prioritizing work and doing more with less are a reality.

Demonstrating that you understand your audience gives them the validation they seek and will, in most cases, lead to a more productive discussion.

4.1.2. Understand the Business Risk

Another reality of limited resources is demonstrated by the risk-based approach to vulnerability remediation and security management programs in general. Organizations have to focus their efforts on the issues that cause the most risk. The problem lies in the fact that many security teams blindly accept the risk ratings assigned by vulnerability assessment tools and penetration test reports. They are prepared to provide a list of all discovered vulnerabilities but they often cannot provide justification for those ratings and therefore cannot give business decision makers the business context information they need to make an accurate decision.

How does a security professional understand the business risk well enough to communicate in terms decision makers can understand? If the company is publicly traded, you can start by reading the 10-K report with the SEC filing. This report describes key factors that are important in the company's growth. The 10-K report also highlights key risks that would substantially harm the company. Another option is to simply ask. Schedule time with management and decision makers and ask them what security concerns they have. Be prepared to start quick discussions in the elevator. The most crucial element of these conversations is listening. Send them a thank you note after the discussion to let them know that you heard their concern and intend to act. If you have a quick solution, offer your help in implementing the solution.

Another source of business intelligence is the organization's compliance requirements. Understand what penalties may be assessed if your organization fails to

meet compliance requirements and be prepared to give examples of other organizations in your industry that have suffered similar penalties or losses as a result of security failures.

A security professional should also understand the business benefits of security and be able to communicate them. If your customers require assurance of your security practices, be prepared to show how your security program has reduced the length of the sales cycle and improved customer trust and satisfaction.

A security professional who understands the business implications of security will be much more successful in winning others to his or her cause and obtaining executive support and funding in the future.

4.1.3. Slow Down

Stephen R. Covey, in his bestseller book, 7 Habits of Highly Effective People, repeats the oft-quoted phrase, "slow is fast and fast is slow". This phrase refers to the need to slow down and listen when working with people. At the heart of this phrase is the notion that if an individual tries to force a quick "fix" change with another individual, he or she may find that the time to implement that change is drawn out indefinitely. An individual who takes time to listen to and understand the other individual will find a much more receptive audience that is more willing to change. (Covey, 1989)

This principal of effective human relations has direct applicability to vulnerability remediation. It usually takes a person with significant technical abilities to discover security vulnerabilities. At times, it is difficult for intelligent, technically gifted people to slow down long enough for people to catch up. If the security team approaches the remediation process with the attitude that they already know what is best and how to resolve the issue, they will miss an opportunity to learn. This contributes to the "us vs. them" mentality that seems to exist between many security teams and the rest of the organization. The "slow is fast" half of the phrase quoted previously means that the security team that presents the vulnerability and then takes time to listen to and answer the questions of the affected teams will find that the remediation process progresses much more quickly. The team that attempts to force the issue with a "my way or the highway" mentality may in fact find themselves slowed by the mire of personal power struggles and

political turf wars that always seem to arise when people feel threatened or belittled. When delivering vulnerabilities to teams, schedule time to meet and discuss so that the problem can be explained and questions answered. You may not end up making any concessions to the affected team, but at least they will not feel that the issue has been "thrown over the wall".

4.2. Understand Technical Risk

In addition to understanding the business risk it is also crucial that security professionals understand the technical risk of the vulnerability. Using a risk ranking process similar to DREAD or CVSS may derive part of this understanding. It also entails understanding the technical details of the vulnerability well enough to explain the damage potential, remembering to avoid the exploitability trap. A security professional doesn't need to demonstrate that a particular vulnerability is exploitable in order to explain the technical implications of the vulnerability.

4.3. Validate

It can be difficult to eliminate all false positives from a report. However, a developer or IT team will quickly recognize a report that has not been scrubbed. The security team should be eager to generate a quality work product, which means eliminating false positives so as to provide an accurate and clean report. The security team should also be able to help the team with remediation responsibilities understand how to reproduce the test results and validate when it has been fixed. This is nearly impossible if the security team has not reviewed and validated the results.

This step can take considerable time and effort. However, it is an imperative step. The report of vulnerability findings is the longest lasting artifact of a test. Many security tests have been performed with great care and professionalism only to be documented by a report that is full of spelling and grammatical mistakes and false positives that were not validated. A low quality report also reflects poorly on the professionalism and competence of the team performing the test. Schedule time to ensure that the reporting phase is performed properly.

4.4. Understand Solutions

When meeting with teams to discuss vulnerability remediation, it is wise to come armed with potential solutions to recommend. Fortunately, the security team doesn't have to be an expert on every potential solution to the problem. Many resources are available to help understand the potential solutions without mandating the technical approach taken. For example, OWASP produces many cheat sheets that describe the steps that need to be taken. This allows the security team to specify requirements (e.g. encode all characters, except for alphanumeric, with ASCII values less than 256 with the \xHH format) rather than the actual framework or coding technique. This gives the development team the flexibility to choose tools that meet performance, architectural and functional requirements while satisfying security requirements.

5. Conclusion

The vulnerability remediation process can be prolonged, difficult and fraught with unpleasant experiences. Some of the difficulties encountered by security teams can be ascribed to the approach taken by these teams. Security teams often fail to understand business objectives and create significant obstacles for development teams by haphazardly delivering scan reports that are several hundred pages long and filled with false positives and inaccuracies. This sort of behavior damages working relationships and ultimately leads to business risk, in the form of unclosed vulnerabilities.

Security teams don't have to take an "us versus them" approach. By applying time-honored principles of effective human communication and relationship skills, many of these obstacles can be overcome. Security teams also need to create a vulnerability management plan, which includes a defensible method for ranking and prioritizing risks. They also need to set realistic remediation goals, and secure development standards that support business objectives.

If security teams will apply the principles described here, they will be able to achieve better working relationships between their team and the rest of the organization. This in

turn leads to shorter remediation times and reduced risk for the organization. It also helps the security team demonstrate the value they provide in terms that are important to the business. In the end everyone wins.

6. References

Bureau of Labor Statistics. (2012, April 05). *Occupational Outlook Handbook*. Retrieved September 27, 2013, from US Department of Labor Bureau of Labor Statistics: http://www.bls.gov/ooh/Math/Actuaries.htm

Carnegie, D. (1936). *How to Win Friends and Influence People.* New York, NY: Simon & Schuster.

Chess, B., & West, J. (2007). *Secure Programming With Static Analysis.* Pearson Education.

Covey, S. R. (1989). *The 7 Habits of Highly Effective People.* Ney York, NY: Simon & Schuster.

Doran, G. T. (1981). There's a S.M.A.R.T. way to write management's goals and objectives. *Management Review , 70*, 35-36.

eWeek. (2002, January 21). *Gates: Security Over Features*. Retrieved November 11, 2013, from eWeek: http://www.eweek.com/c/a/Security/Gates-Security-Over-Features/

Goldman, W. (Writer), & Reiner, R. (Director). (1987). *The Princess Bride* [Motion Picture].

Hannum, R. (2012, June 05). *UNLV Center for Gaming Research*. Retrieved September 27, 2013, from Casino Mathematics: http://gaming.unlv.edu/casinomath.html

ISO. (n.d.). *ISO/IEC 27034-1:2011*. Retrieved from http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=44378

LeBlanc, D. (2007, August 14). *MSDN Blogs*. Retrieved September 27, 2013, from David LeBlanc's Web Log: http://blogs.msdn.com/b/david_leblanc/archive/2007/08/13/dreadful.aspx

Lipner, S. (2013, May 14). *Microsoft SDL Conforms to ISO/IEC 27034-1:2011*. Retrieved November 11, 2013, from The Security Development Lifecycle Blog:

http://blogs.msdn.com/b/sdl/archive/2013/05/14/microsoft-sdl-conforms-to-iso-iec-27034-1-2011.aspx

NIST. (n.d.). *Common Vulnerability Scoring System Version 2 Calculator*. Retrieved Jan 25, 2014, from National Vulnerability Database: https://nvd.nist.gov/cvss.cfm?calculator&version=2

NIST. (2005, November). *Creating a Patch and Vulnerability Management Program.* Retrieved September 19, 2013, from Computer Security Resource Center: http://csrc.nist.gov/publications/nistpubs/800-40-Ver2/SP800-40v2.pdf

NIST. (2013, September). SP800-30r1 Guide for Conducting Risk Assessments. Gaithersburg, MD, USA.

OWASP. (2009). *Getting Started Using ASVS.* Retrieved 10 17, 2013, from OWASP: https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project

OWASP. (2010, September 29). *Threat Risk Modeling.* Retrieved September 29, 2013, from OWASP: https://www.owasp.org/index.php/Threat_Risk_Modeling

PCI Security Council. (2010, October). *Requirements and Security Assessment Procedures Version 2.0.* Retrieved October 17, 2013, from PCI Data Security Standard: https://www.pcisecuritystandards.org/documents/pci_dss_v2.pdf

Pescatore, J. *Application Security: Tools for Getting Management Support and Funding.* SANS Institute.

Seller, D. (2005, November 17). *New Microsoft Threatmodeling Slidedeck.* Retrieved September 27, 2013, from SilverStr's Blog: http://silverstr.ufies.org/blog/archives/000877.html

University of Regina. (n.d.). *The History of a Picture's Worth*. Retrieved Jan 25, 2014, from http://www2.cs.uregina.ca/~hepting/research/web/words/history.html

Vick, P. (2005, February 9). *Why is this man smiling?* Retrieved September 18, 2013, from Panopticon Central: http://www.panopticoncentral.net/2005/02/09/why-is-this-man-smiling/

WhiteHat Security. (2013). *Website Security Statistics Report.* Santa Clara: WhiteHat Security.

Last Updated: May 22nd, 2017

Upcoming SANS App Sec Training

SANS Stockholm 2017	**Stockholm, Sweden**	**May 29, 2017 - Jun 03, 2017**	**Live Event**
SANS Minneapolis 2017	**Minneapolis, MN**	**Jun 19, 2017 - Jun 24, 2017**	**Live Event**
SANS London July 2017	**London, United Kingdom**	**Jul 03, 2017 - Jul 08, 2017**	**Live Event**
SANSFIRE 2017	**Washington, DC**	**Jul 22, 2017 - Jul 29, 2017**	**Live Event**
Community SANS San Francisco DEV541	**San Francisco, CA**	**Aug 28, 2017 - Aug 31, 2017**	**Community SANS**
SANS Network Security 2017	**Las Vegas, NV**	**Sep 10, 2017 - Sep 17, 2017**	**Live Event**
Secure DevOps Summit & Training	**Denver, CO**	**Oct 10, 2017 - Oct 17, 2017**	**Live Event**
SANS Seattle 2017	**Seattle, WA**	**Oct 30, 2017 - Nov 04, 2017**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**