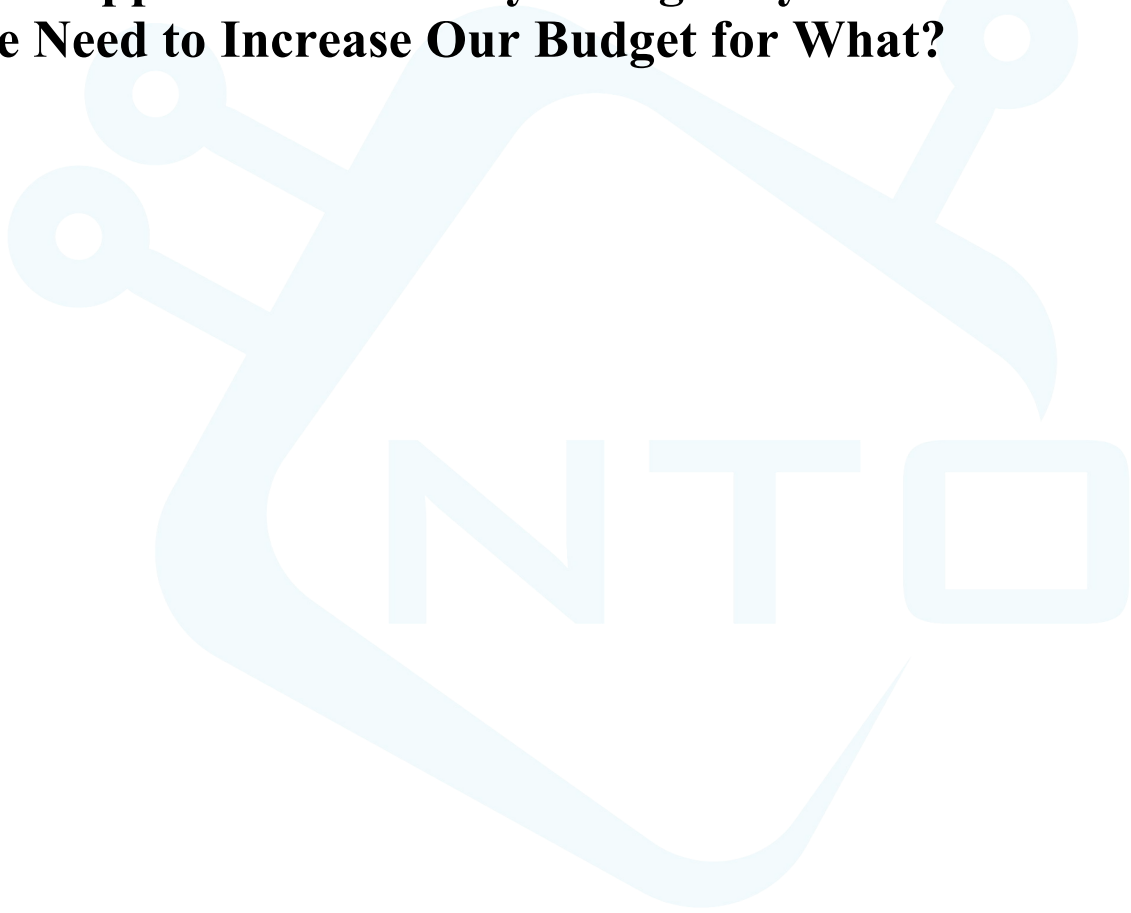


Web Application Security: Budgetary Considerations. We Need to Increase Our Budget for What?



NT OBJECTIVES, Inc.
Business White Paper
Authored by Matthew
Cohen

Web Application Security: Budgetary Considerations. We Need to Increase Our Budget for What?

It seems that every week there is another instance of identity theft, and stolen data. Industry experts report that these crimes are netting more than \$50 billion annually for the criminals and leaving the victims with endless problems. MasterCard International and VISA announced that at least 40 million credit card numbers were stolen by a third-party payment processor. DSW Shoe Warehouse reported that private customer data was stolen from 108 stores, and transaction information involving 1.4 million credit cards was obtained, and the list goes on and on.

Additionally, with increased regulations, organizations that are not compliant, and allow these breaches to occur, risk large fines, prosecution and loss of reputation. However, many organizations are still extremely vulnerable. Perimeter protection and network security provide some protection, but the majority of these threats occur at the Web application layer, and it is critical for organizations to put comprehensive programs in place that include application security. Many organizations wonder why the security investments they have already made are no longer sufficient. This article provides guidance for including Web application security in your overall strategy, and best practices for planning and budgeting for it appropriately.

What is the Risk of This Happening?

It could be tragic if something like this happened to your organization, but how likely is it really? Well, last year 95% of companies reported Web breach incidents, so it is certainly a real concern. It is a critical time for companies to take Web application seriously, and to implement comprehensive security programs to protect private data.

How Do We Secure Our Applications?

The best way to stop a problem is before it starts. The QA group is supposed to find any security vulnerabilities before the application goes live. Unfortunately, 1) they are focused on many other things, 2) are likely not security experts, 3) do not have automated tools that work effectively and 4) based on our experience, if an application is late, it is unlikely that a manager is going to stop the show to fix a security vulnerability.

There are a large number of vulnerabilities in production applications. Which means that the overworked security audit teams are responsible for finding them. Once they find them, they likely do not have the skills or authority to fix them and must ask the development team to stop what they are doing (they have likely moved on to a new project) and fix an old problem. In my experience, the success and progress in fixing old vulnerabilities is almost nil.

It is important for the security team and the development team to address issues of Web vulnerability and security, while supported by the executive staff. In performing comprehensive assessment of an application, they may find that there are many security holes and they do not have the resources to fix them (or perhaps even identify them). Without the proper tools, there is really no way of telling 1) whether this is a real problem, 2) what the extent of the problem is at your company, 3) how to fix it in the most cost effective manner, 4) how much it will cost or 5) how long it will take.

Why This Won't Get Solved In the Normal Budget Process

Given the unique nature of this problem, you will not be able to fix this problem using the normal budget process. Here are a few reasons:

1. The selection of Web application security tools is done by the security team. Typically, they have limited budget. This means that they have to go hat in hand and explain to the other teams (QA and development) that this is a serious problem and that the other teams need to allocate some of their budgets to a security problem. This is because there are no traditional vendor fixes to apply in order to remediate

Web Application Security: Budgetary Considerations. We Need to Increase Our Budget for What?

the problem in the manner they are trained. The application must be corrected by fixing in-house code, and there is an inherent disconnect between the security group and the development teams who are trying to develop new applications, not remediate legacy ones.

2. While the QA and development teams are not unsympathetic, they are typically less knowledgeable about Web application security than the security team and are largely concerned with completing new Web applications (or revisions of old ones) on deadline. Even if they are sympathetic and knowledgeable, the communication and explanation process can result in months of delays; simply put security teams and development teams speak different languages.

3. There will almost certainly be almost no line item in the budget for Web application security tools. This means that there will need to be a series of meetings and presentations to get approval (likely C level) to add to the budget or move around the budget to address this problem.

Business and Budgeting Best Practices For Addressing Application Security

1. Be Proactive

With all due respect to the Ostrich, proactivity is generally a preferred strategy. The Year 2000 problem has acclimated decision makers to the notion that advances in technology can come with the occasional hiccup that results in an unforeseen charge.

2. Understand the Scope of the Problem

You will undoubtedly hear from the security team that there are hundreds or thousands of Web applications, millions of lines of code and tens of thousands of vulnerabilities. All of this is, of course, mostly meaningless. Most of the code has nothing to do with potential vulnerabilities. Not all vulnerabilities are equally serious. Most of the vulnerabilities can be traced to a finite number of coding errors. Get your security team to define a) the total number of unique Web interfaces that could be vulnerable (e.g. an html form to submit a credit card number) and b) the number of these that actually are vulnerable. These are useful, trackable metrics that you can use to 1) track the effectiveness of your development and QA teams and 2) to track progress towards remediating vulnerabilities and the cost of doing so.

3. Stop the Problem at Its Inception

The most cost-effective place to address security in the application lifecycle is during development. Once it is out of development, security has to identify the problem, communicate it back to development, cajole them to fix it, retest, potentially go back for a second round, and additional cycles. It will cost roughly 5-10 times as much to fix a vulnerability once the application goes into production.

4. Tie Compensation to Achievement of Metrics

Once you have agreed on how to track vulnerabilities, tie compensation to progress on their reduction. Money talks and given the loud voices demanding new features, the only way to get results is to tie achievement to compensation.

5. Create a Working Group

Given the interdepartmental nature of the problem (development, QA, security) you will need a working group chaired by a neutral, senior manager to assess the problem and make recommendations.

6. Treat This like any other Business Process

Creating secure Web applications is a business process, like any other. It needs to be analyzed, measured and improved if you want to have secure applications.

7. Create Separate Budgets

Web Application Security: Budgetary Considerations. We Need to Increase Our Budget for What?

Unfortunately, like most business problems, the only way to find a solution is to create a budget. Security, Development and QA teams do not have the tools or resources to prevent new vulnerabilities or to begin to address old ones. There have to be budgets for both the facilitation of remediation efforts in addition to the discovery of vulnerabilities.

8. **Create a Separate Application Security Development Team**
Although this suggestion has the potential to spark political turmoil within an organization, (“you can’t touch my code”), consider creating budget for a separate application security team involving 1) developers that can fix code and 2) dedicated QA teams responsible for remediation of Web application security problems. Development teams do not have the time or the enthusiasm to fix prior errors. It may be helpful to have the security “developers” have a dotted line report to the head of development. Getting dedicated professionals to fix network problems is the only way that the network security problem is ultimately addressed.
9. **Create Top Level, Third Party Management Reports**
You will need quantitative, plain English reports that enable senior managers to track progress. Having third party reporting will be helpful, particularly if compensation is tied to progress.
10. **Trust, but Verify**
Create a budget to have regular third party audits of a random sampling of internal reports.
11. **Beware the Quick Fix**
Web Application Security illustrates the old maxim, “there are no simple solutions to complex problems.” Web application hacks occur through the URL. There is no bright line delineating what is acceptable input versus what is a potential hack. Technologies like ISAPI filters, Web application firewalls and intrusion detection/prevention systems all can play a role but they should not be viewed as cure-alls. They all require expensive and intelligent customization to make them effective and non-destructive (i.e. they do not stop desired user behavior). Given the constant upgrades and changes to Web applications, repeated updates of these technologies can create prohibitive TCO. Additionally, there are classes of Web application vulnerabilities that will not be addressed by these technologies (like session hijacking).

Conclusion

The Web application security threat is a real one. A failure to respond to this threat will result in real risk to any enterprise that stores financial or customer data. While the problem is a serious one, it is not something that cannot be fixed so long as proper attention and budget are allocated to it. Unfortunately, given the unique nature of the problem and its impact on the budgetary process, it will likely require direct intervention by the financial staff.

Web Application Security: Budgetary Considerations. We Need to Increase Our Budget for What?

About NT OBJECTives, Inc.

NT OBJECTives, based in Orange County, California, brings together an unprecedented collection of this industry's top experts to offer a comprehensive suite of industry-leading technology and services to solve the application security problems of today's global business leaders. Through the synergy of the top security software developers and some of the industry's best consultants and researchers, NTO has created the first next-generation, automated technology capable of performing accurate application security audits. Coupled with a comprehensive service offering, including security training services, NTO is uniquely positioned to provide complete application security solutions to today's businesses.

About Matthew Cohen

Matthew Cohen is Chief Operating Officer at NT OBJECTives, where responsible for quality assurance, professional services and marketing. Mr. Cohen brings an extensive background in business strategy, finance, marketing and operations to NT OBJECTives.

Mr. Cohen has extensive experience in the investment banking industry and executive management to NTO. Previous to NTO, Matthew was CFO of publicly traded TTR Technologies. Prior to TTR, Mr. Cohen was the founding CFO at APB Online, Inc., where he raised \$27 million in three private placements and built a financial infrastructure to support a 140 person company. Matthew has held positions at The Blackstone Group, Rothschild, Inc., and Kidder, Peabody & Co.; where he worked on restructurings, capital financings, and mergers and acquisitions. In one of these restructurings, New Dartmouth Bank, the investor group acquired 3 insolvent banks from the FDIC, invested \$40 million to re-capitalize them and after an operating restructuring, sold the bank for \$160 million two and a half years later. He holds a degree in economics from Princeton University.