PTA: Practical Threat Analysis

By Russ McRee – ISSA member, Puget Sound (Seattle), WA, USA chapter

Practical Threat Analysis

Prerequisites

Windows OS
MS Access 2003 runtime (included in installer)

Similar Projects

Threat Modeling Tool[1]
SOMAP[2]

It gives me comfort just saying it out loud: "practical threat analysis." Short, sweet, to the point, and extremely useful when conducting exactly what you might imagine: threat analysis, or more generically, risk assessment. In October 2007 we covered SOMAP, a tool set useful for similar endeavors, and I promised myself then that I'd feature PTA as soon as reasonably possible, given its rich feature set. Project lead Zeev Solomonik refers to Practical Threat Analysis (PTA) as a *risk assessment methodology and a suite of software tools that enable users to find the most beneficial and cost-effective way to secure systems and applications according to their specific functionality and environment. PTA's role is to identify system vulnerabilities, map system assets, assess the risk of the threats and define an effective risk mitigation plan for a specific system architecture, functionality and configuration.*[3]

PTA can assist in PCI DSS self-assessment; for vendors with less than 1,000,000 Visa e-commerce transactions per year, the PCI DSS threat model template will serve you well. If you're interested in performing an ISO 27001 risk assessment audit you can also take advantage of the PTA ISO 27001 library. These freeware libraries, and other useful articles regarding methodology, case studies, tools and technology, reporting, and sample projects, can be downloaded at the "Practical Threat Analysis Documents & Samples" page.[4]

I queried Zeev regarding current project details and he offered the following:

1. PTA Professional Edition is now used by many thousands of risk analysts and security professionals worldwide for their threat analysis projects. PTA threat models have been found productive in risk assessment missions in a variety of domains such as IT, finance, communication, healthcare as well as academic and research projects.
2. The PTA calculative method for assessing risks in $ values and prioritizing implementation of countermeasures became the standard de facto for justifying mitigation investments. A detailed description of the methodology and the calculative method is available.[5] The dynamic nature of the PTA database allows the analyst to constantly adjust the threat model to the changing circumstances.
3. The PTA plug-in libraries of predefined security entities has proven to be the community's favorite for transforming compliance knowledge and data into effective mitigation actions. More material on how to convert security compliance standards to PTA threat models and use them as a dynamic baseline for quantitative risk analysis is available.[6]
4. The product's development is driven by user feedback and requests. In the last two years more than 20 builds have been released with many usability improvements, bug fixes and additional features – thanks to the community of PTA users for their constant feedback and continuous support; PTA versions history.[7]
5. The next release (planned for late October) will include additional reports which improve the ability to aggregate and present the threat model data and results according to useful parameters. Future releases will include improvements of export/import features to support easier integration with existing risk management systems as well improved documentation and on-line tutorial.[8]

In order to run PTA through its paces while creating an example project, I compiled what I hope will be a useful PTA library for threat modeling web applications (details below).

Web application threat modeling highlights

Before diving into PTA, let's take a quick look at some relevant content from relevant modeling resources, to more easily map them to our use of PTA.

1 http://www.microsoft.com/downloads/details.aspx?familyid=62830f95-0e61-4f87-88a6-e7c663444ac1&displaylang=en.
2 http://somap.org.
3 http://www.ptatechnologies.com.
4 http://www.ptatechnologies.com/DocumentsFrameset.htm.
5 http://www.ptatechnologies.com/pta.htm.
6 http://www.ptatechnologies.com/comments.htm.
7 http://www.ptatechnologies.com/latestupdate.htm.
8 Zeev Solmonik, interview.

ISSA Information Systems Security Association The Global Voice of Information Security ©2008 Information Systems Security Association • www.issa.org • editor@issa.org • Permission for author use only.

A good threat model typically includes a check list, and I appreciate how succinctly Kevin Beaver pulled his together in *The Essentials of Web Application Threat Modeling*. His list, quickly summarized here, includes seven key steps:

1. **Determine your security goals** by outlining what's in scope, what's absolutely critical, as well as what's being mandated by management, security policy, or even customers or business associates.
2. **Document the general architecture of your application** and outline information flows from user to Web server, Web server to application server, application server to database server, and so on.
3. **Outline what really needs to be protected** such as user login credentials, session information, source code, application logic, and (especially) customer information.
4. **Pinpoint the various entry points and "trust" zones** that need protection, including user authentication, user management, system logging, server/application maintenance, and critical application and database interfaces.
5. **Discover what can be exploited** using a malicious mindset – from both the perspective of an untrusted outsider and a trusted user.
6. **Determine what's urgent and important** based on the likelihood and impact of each weakness.
7. **Determine what can be done about each weakness** and when it can/will be resolved.[9]

Kevin refers to STRIDE as a subset of step 5 in this useful article. STRIDE refers to Microsoft's threat model methodology, specifically an acronym for:

1. **S**poofing identity
2. **T**ampering with data
3. **R**epudiation
4. **I**nformation disclosure
5. **D**enial of service
6. **E**levation of privilege

If employing STRIDE as a referential starting point for step 5, you'll be well on your way to covering the exploit vectors. The DREAD model can be used in step 6, aiding you in what may be more generically referred to as data classification, useful in ranking assets and data in your version of *urgent*, *critical*, *high*, *medium*, and *low* (or some derivative thereof). According to OWASP, DREAD is an "algorithm used to compute a risk value, which is an average of all five categories."

Risk_DREAD = (DAMAGE + REPRODUCIBILITY + EXPLOITABILITY + AFFECTED USERS + DISCOVERABILITY) / 5

An example of quantifying a DREAD category, from the OWASP description, includes *Exploitability*.

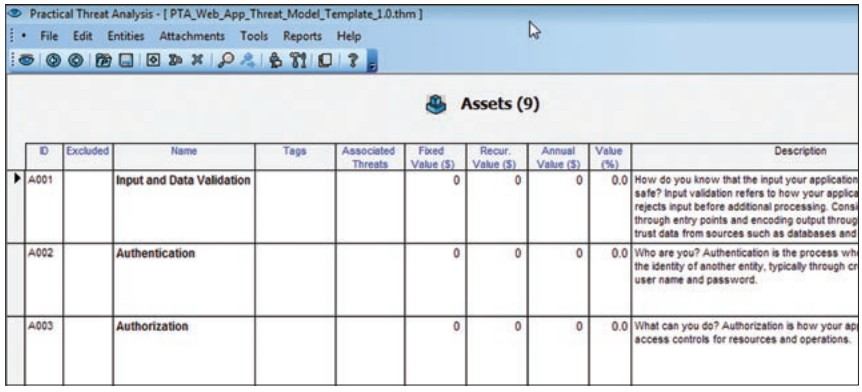

Figure 1 – Populating Assets

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[10]

Finally, give *Template Sample: Web Application Threat Model on MSDN*[11] a good long read. Although a bit dated (2005), it served as an excellent reference template for me to populate the above mentioned `PTA_Web_App_Threat_Model_Library_1.0.thl`,[12] a library that is intended to aid you in shaping your web application assessment process.

Enough process work, let's explore PTA from a hands-on perspective.

Using PTA

PTA includes such a straightforward installation process, I'll tell you only that following the default process will have you up and running in less than five minutes. There is a dependency on Microsoft Office Access 2003 Runtime, but it's included in the 39MB install download.

The UI for PTA is largely browser-like complete with drop-down menu, forward and back buttons, and the all important System's Status button (the eye) leading you to the summary page.

Using the MSDN Cheat Sheet: Web Application Security Frame[13] as a reference, I began by mapping the Web Application Security Frame Categories to PTA Assets. These categories include:

- Input and data validation
- Authentication
- Authorization
- Configuration management
- Sensitive data

9 http://searchsoftwarequality.techtarget.com/tip/0,289483,sid92_gci1306902,00.html.

10 http://www.owasp.org/index.php/Threat_Risk_Modeling#DREAD.

11 http://msdn.microsoft.com/en-us/library/ms978516.aspx.

12 http://holisticinfosec.org/toolsmith/files/PTA/PTA_Web_App_Threat_Model_Library_1.0.thl.

13 http://msdn.microsoft.com/en-us/library/ms978518.aspx.

©2008 Information Systems Security Association • www.issa.org • editor@issa.org • Permission for author use only.

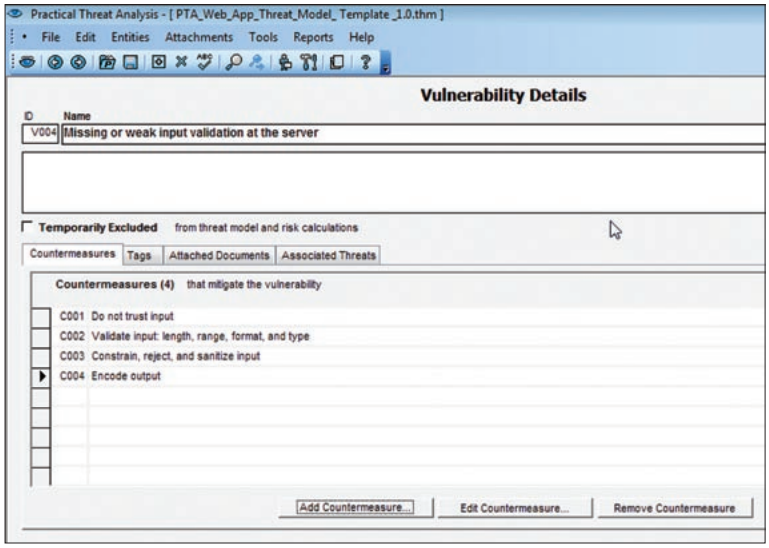

Figure 2 – Vulnerability Details

- Session management
- Cryptography
- Exception management
- Auditing and logging

It may be a bit disconcerting declaring these categories as assets, but you'll find that PTA will allow you to further map vulnerabilities and countermeasures to each of the assets. I contend that these categories as assets are, in fact, worthy of protection as unique entities, given the litany of possible vulnerabilities, threats, and attacks that can be exploited and perpetrated against them. You can always add specific physical assets as you wish when you create a project using this library. Using the PTA library we're building here, you can narrow or expand your selections as needed (see Figure 1).

I then moved to *Vulnerabilities* and utilized the MSDN Template Sample again (Figure 2). As you flesh out *Vulnerability Details*, you have the option to immediately populate *Countermeasures*, *Tags* and *Associated Threats*. Tags are applicable where you choose to use them, in that they are "free-text descriptive attribute associated with the threat model entities (assets, threats, vulnerabilities and countermeasures). Tags help the analyst classify the various model entities and improve their comprehensibility."[14]

From Vulnerabilities I moved to *Threats*, which allows you to immediately map a threat to assets, vulnerabilities, entry points, attackers, tags, and attached documents.

After Threats are populated it's an obvious next step to ensure you have Countermeasures for those threats, and the library I built during this exercise includes all countermeasures recommended from the MSDN cheat sheet.

I finished the process with *Entry Points* and *Attacker Types* to ensure a comprehensive and useful library for your projects (Figure 3). With the library complete, Iet's conduct a simple web application threat model to see the benefits of using PTA to complete the task.

We'll make the following entirely arbitrary assumptions. No giggling, please. You'll likely reach the conclusion that I don't do much risk-threat analysis-modeling (Damn it, Jim… I'm an engineer, not a miracle worker!). Let me repeat, the values and assumptions are arbitrary.

The web application is a fundamentally simple social forum, with no PII other than simple account information, ASP, an MS-SQL backend, and a lot of content posted by authenticated users. Guests have read only access. Loss of confidentiality, integrity, and availability have a direct cost associated with them, as reputation damage and down time could lead to lost ad revenue or sponsorship.

Following are the assumed components for our small exercise:

Assets

- Input and data validation
- Authentication
- Authorization
- Sensitive data
- Session management

Vulnerabilities

- Using application-only filters for malicious input
- Failing to validate input from all sources including cookies, query string parameters, HTTP headers, databases, and network resources
- Permitting prolonged session lifetime
- Failing to limit database access to specific stored procedures
- Using insecure administration interfaces.

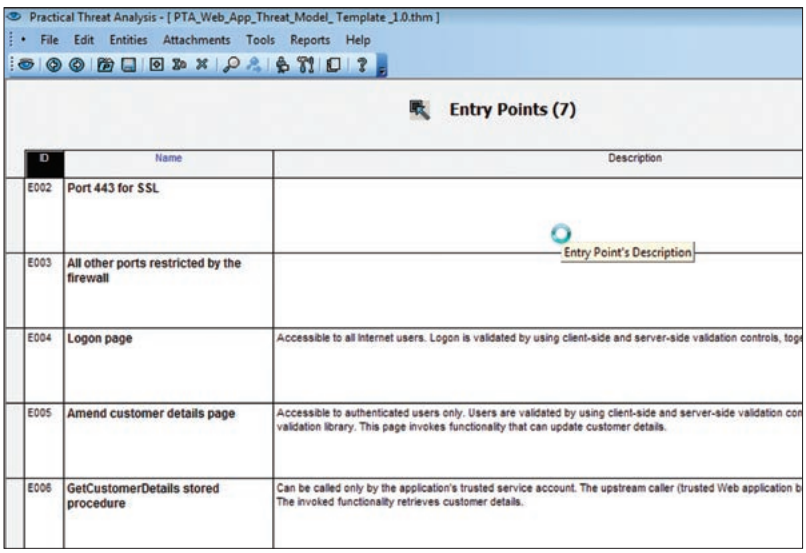

Figure 3 – Entry Points

14 PTA Help files, PTA-PTA Screens-Tags.

©2008 Information Systems Security Association • www.issa.org • editor@issa.org • Permission for author use only.

Threats

- Cross-site scripting
- SQL injection
- Cookie manipulation
- Cookie replay attacks
- Credential theft
- Session hijacking
- Session replay

Countermeasures

- Validate input: length, range, format, and type
- Constrain, reject, and sanitize input
- Encode output
- Encrypt communication channels to secure authentication tokens
- Use HTTPS only with forms authentication cookies
- Use strong authentication and authorization on administrative interfaces

Entry points

- Port 80 for web requests
- Port 443 for SSL
- All other ports restricted by the firewall
- Logon page

Attacker types

- Mass SQL injection attacks seeking to embed malicious JavaScript
- Script kiddies

To begin our project, I selected *New Project*, named it `toolsmith_web_app_threat_model.thm`, then clicked *Tools – Load Entities from Library*. Using the library created above I selected each of the components detailed in our lists.

To apply value of my assets, I made a blanket assumption that revenue is $1000 a day for the forum, and that each asset is of equivalent value. Thus at $365, 000 revenue total, each asset is valued at $73,000 annually.

Under Countermeasures, again costs were applied as recurring, per year, assuming annual code updates, certificate renewals, resource expenses, etc.

All cooked up in the risk management stew pot that is PTA, we find a comprehensive snapshot of our web application that would make any risk manager proud (Figure 4).

We've included all countermeasures as part of the mitigation plan, but as you can see, we have some work to do implementing them as only half our countermeasures are currently in play.

Don't forget to take a closer look at the *Report* features and the *Threat Builder*, as I have neglected them.

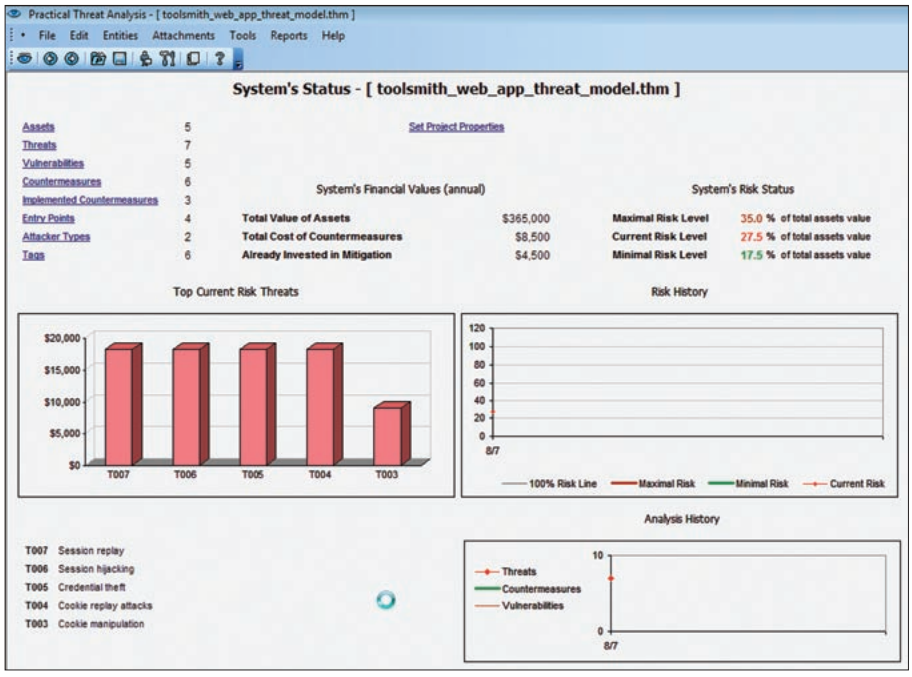

Figure 4 – PTA sums it up

Remember, you can download both this project and library from my website and experiment for yourself.[15]

Benefits and drawbacks

A well-conducted threat model is invaluable, regardless of the subject. Although we chose a small web application, this methodology is relevant to any technology you value and whose protection is your charter.

Time and fiscal considerations can be a deterrent to accuracy and commitment. Applying value, be it qualitative or quantitative, can be burdensome.

In conclusion

Make use of this excellent threat analysis tool, consider contributing to the project, and reap the benefits of a concrete understanding of your assets. Go forth and analyze! Cheers, until next month…

Acknowledgments

Zeev Solomonik, for his contributions to this article as PTA project lead.

About the Author

Russ McRee, GCIH, GCFA, CISSP, is a security analyst working in the Seattle area. As an advocate of a holistic approach to information security, Russ' website is holisticinfosec.org. Contact him at russ@holisticinfosec.org.

15 http://holisticinfosec.org/toolsmith/files/PTA/toolsmith_web_app_threat_model.thm, http://holisticinfosec.org/toolsmith/files/PTA/PTA_Web_App_Threat_Model_Library_1.0.thl.

ISSA Information Systems Security Association The Global Voice of Information Security ©2008 Information Systems Security Association • www.issa.org • editor@issa.org • Permission for author use only.