ThreadFix: You Found It, Now Fix It

By Russ McRee – ISSA Senior Member, Puget Sound (Seattle) Chapter

Prerequisites

ThreadFix is self-contained and as such runs on Windows, Mac, and Linux systems

JEE based, Java 7 needed

As an incident responder, penetration tester, and web application security assessor, I have long participated in vulnerability findings and reporting. What wasn't always a big part of my job duties was seeing the issues remediated, particularly on the process side. Sure, some time later, we'd retest the reported issue to ensure that it had been fixed properly, but none of the process in between was in scope for me. Now, as part of a threat intelligence and engineering team I've been enabled to take a much more active role in remediation, often even providing direct solutions to discovered problems. I'm reminded of a London underground (Tube) analogy for information security gap analysis (that space between find and fix) taken whilst stepping on the train. Mind the gap!

But with new responsibilities comes new challenges. How best to organize all those discovered issues to see them through to repaired nirvana? As is often the case, I keep an eye on some of my favorite tool sources, and NJ Ouchn's Toolswatch[1] came through as it often does. There I discovered ThreadFix,[2] developed by Denim Group, a team I was already familiar thanks to my work with ISSA. When in 2011 I presented "Incident Response in Increasingly Complex Environments" to the ISSA Alamo Chapter in San Antonio, TX, I met Lee Carsten and Dan Cornell of Denim Group. They've had continued growth and success in the three years since and ThreadFix is part of that success. After pinging Lee regarding ThreadFix for *toolsmith* he turned me over to Dan who has been project lead for ThreadFix from its inception and provided me ample insight. Dan indicated that while working with their clients, they saw two common scenarios—teams just getting started with their software security programs and teams trying to find a way to scale their programs—and that ThreadFix is geared toward helping these groups. They'd seen lots of teams that had just purchased a desktop scanning tool who'd run some scans and the results would end up stored on a shared drive or in a Sharepoint Document Repository. Dan pointed out that these results though were just blobs of data such as PDFs being emailed around to development teams with no action being taken. ThreadFix gives organizations in this situation an opportunity to start treating the results of their scanning as managed data so they can lay out their application portfolio, track the results of scanning over time, and start looking at their software security programs in a much more quantitative manner. Per Dan, this lets them have much more "grown up" conversations with management about application and software risk. A natural byproduct of managed data leads to conversations that evolve from "Cross-site scripting is scary" to "We've only remediated 50% of the XSS vulnerabilities we've found and on average it takes us 120 days, which is twice as slow as what others in our industry are doing." WHAT!? An informed conversation is more effective than a FUD conversation? Sweet! Dan described more sophisticated organizations who are tracking this "mean time to fix" metric as better managing their window of exposure, and that public data sets, such as those released by Veracode and WhiteHat Security, can provide a basis for benchmarking. Amen, brother. Mean time to remediate is one of my favorite metrics.

Dan and the Denim team, while working with bigger organizations, saw huge struggles with teams getting bogged down trying to deal with different technologies across huge portfolios of applications. He cites the example of the information security group buying scanner X while the IT audit group purchased scanning service Y and the QA team was starting to roll out static analysis engine Z. He summed this challenge up best with "The only thing worse than approaching a development team with a 300 page PDF report with a color graph on the front page is approaching them with two or three PDFs and expecting them to take action." Everyone familiar with Excel hell? That's where these teams and many like them languish, trying to track mountains of vulnerabili-

1 http://www.toolswatch.org/2014/05/threadfix-v2-1m1-released/.

2 http://www.threadfix.org/.

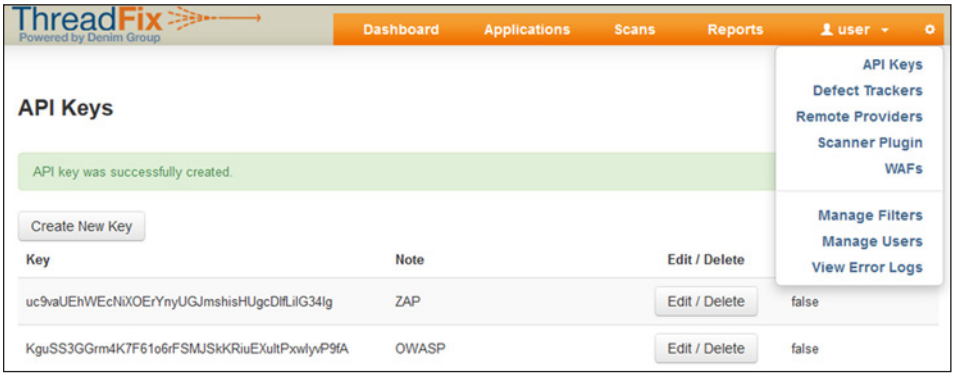

Figure 1– Create ThreadFix API keys for plugin use

ties and making no headway. Dan and Denim intended for ThreadFix to enable these teams to automatically normalize and consolidate the results of different scanning tools even across dynamic (DAST) and static (SAST) application security testing technologies. This is achieved with Hybrid Analysis Mapping as developed under a contract with the US Department of Homeland Security (DHS). According to Dan, with better data management, security teams can focus on high value tasks such as working with development teams to actually implement training and remediation programs. Security teams can take the data from ThreadFix and export it to developer defect tracking tools and IDEs that developers are already using. This reduces the friction in the remediation process and helps them fix more vulnerabilities, faster.

Great stuff from, Dan. The drive to remediate has to be the primary goal. The industry has proven its ability to find vulnerabilities; the harder challenge, and the one I'm spending the vast majority of my focus on, is the remediation work. Threat modeling, security development life cycles, and secure coding best practices are a great start, but one way to take your program to the next level is tuning your vulnerability data management efforts with ThreadFix. There is a Community Edition, free under the Mozilla Public License (MPL), which we'll focus on here, which includes a central dashboard, SAST and DAST scanner support, defect tracker integration, virtual patching via WAF/IDS/IPS options, trend analysis and reporting, and IDE integration.

If you seek an enterprise implementation you can upgrade for LDAP & Active Directory integration, role-based user management, scan orchestration, enhanced compliance reporting, and technical support.

Preparing ThreadFix

First, I tested both the 2.0.1 stable version and the 2.1M1 development version and found the bleeding edge to be perfectly viable. ThreadFix includes a number of plugins, and most importantly for our scenario, for OWASP ZAP[3] and Burp Suite Pro.[4] There is also a plugin for Eclipse[5] too, though for defect tracking and IDE I'm a Microsoft TFS/Visual Studio guy (shocker!). Under *Defect Tracking* there is support for TFS, but I can't wait until Dan and team implement a plugin for VS. ☺ To get started ThreadFix installation is a download-and-run-it scenario. ThreadFix Community Edition includes a self-contained .ZIP download containing a Tomcat web and servlet engine along with an HSQL database. That said, most production environment installations of ThreadFix use a MySQL database for scalability; if you wish to do so instructions are provided.[6] As ThreadFix uses Hibernate for data access, other database engines are also supported.

Once you've downloaded ThreadFix, navigate to your installation directory and double-click **`threadfix.bat`** on a Windows host or run **`sh threadfix.sh`** on *nix systems. Once the server has started, navigate to https://localhost:8443/threadfix/ in a web browser and log in with the username *user* and the password *password*. Then immediately proceed to change the password, please.

Click *Applications* on the ThreadFix menu and add a team, then an application you'll be assessing and managing. My team is HolisticInfoSec and my application is Mutillidae as it has obvious flaws we can experiment with for remediation tracking.

After you download the appropriate plugins, unpack each (I did so as subdirectories in my ThreadFix path) and fire up the related tool. Big note here: Burp and XAP default proxy ports conflict with ThreadFix's API interface; you'll have conten-

3 https://github.com/denimgroup/threadfix/wiki/Zap-Plugin.
4 https://github.com/denimgroup/threadfix/wiki/Burp-Plugin.
5 https://github.com/denimgroup/threadfix/wiki/Eclipse-IDE-Plugin.
6 https://github.com/denimgroup/threadfix/wiki/Using-MySQL.

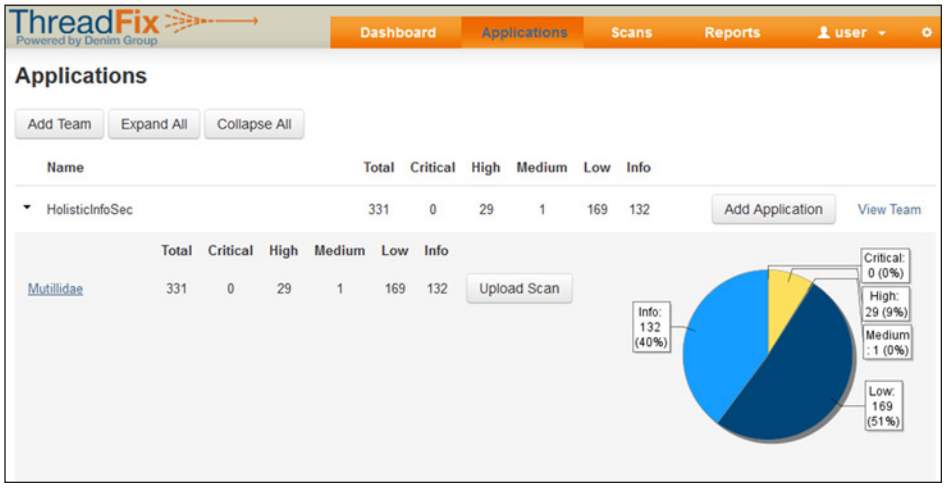

Figure 2 – Scan results uploaded into ThreadFix

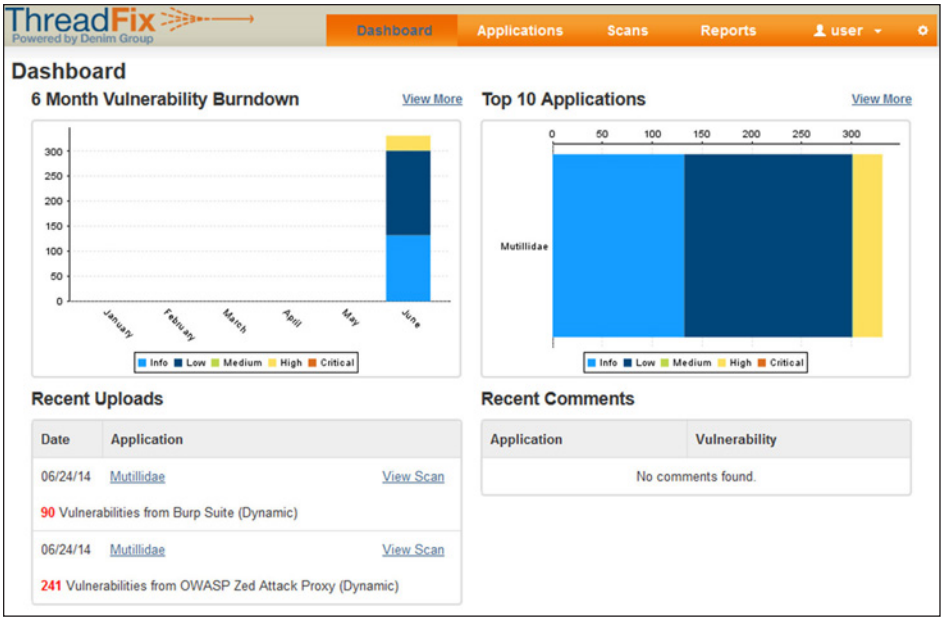

Figure 3 – ThreadFix dashboard provides application vulnerability status

In order to enable for the like of Eclipse, you'll need to take a few steps from here.

- Have team/application setup in Threadfix
- Have source code for an application linked in Threadfix
- Have a scan for the application in Threadfix
- Have the applications scan linked to a Defect Tracker

Once you have it configured, you can select specific vulnerabilities and submit them directly to your preferred Defect Tracker under the Application view, then click *Action*. This is vital if you're pushing repairs to the development team via the likes of Jira or TFS.

tion for port 8080 if you don't configure Burp and ZAP to run on different ports. For Burp, click the *Extender* tab, choose *Add*, navigate to the Burp plugin path and select **`threadfix-release-2.jar`**. You'll then see a new *ThreadFix* tab in your Burp UI which will include *Import Endpoints* and *Export Scan*. You'll need to generate API keys as follows: click the settings gear in the upper right hand of the menu bar and select *API keys* as seen in figure 1.

Click *Export Scan* and paste in the API key you created as mentioned above. Similarly in ZAP, choose *File* then *Load Add-On File* and choose **`threadfix-release-1.zap`**. After restarting ZAP you'll see *ThreadFix: Import Endpoints* and *ThreadFix: Export Scan* under *Tools*.

You may find it just as easy to save scan results from Burp and ZAP in an .xml format and upload them via the ThreadFix UI. Go to *Applications*, then *Expand All*, select your *Application*, and click *Upload Scan*. You'll benefit from immediate results as seen from incomplete Burp and ZAP scans of Mutillidae in figure 2.

The ThreadFix dashboard then updated to give me a status overview per figure 3.

Drilling into your target via the *Application* menu will provide even more nuance and detail with the ability to dig into each vulnerability as seem in figure 4.

Additionally, if you're interested in virtual patching, first create a WAF under *Settings* and *WAFs* where you choose from Big-IP ASM (F5), DenyAll rWeb, Imperva SecureSphere, Snort, and mod_security, which I selected and named it HolisticInfoSec. Click *Applications* again, drill into the application you've added scans for; then click *Action* and *Edit/Delete*. The *Edit* menu will allow you to *Set WAF*. I then selected *HolisticInfoSec* and click *Add WAF*. You can also simply add a new WAF here as well. Regardless, go back to *Settings*, then

Figure 4 – ThreadFix vulnerability details

WAFs, then choose *Rules*. I selected *HolisticInfoSec/Mutillidae* and *deny* then *Generate WAF* rules. The results as seen in figure 5 can then be imported directly into mod_security. Tidy!

So many other useful features with ThreadFix too. Under *Settings* and *Remote Providers* you can configure ThreadFix to integrate with QualysGuard WAS, Veracode, and WhiteHat Sentinel. There are tons of reporting options including trending, snapshots (point in time), scan comparisons (Burp versus ZAP for this scenario), and vulnerability searching. Try the scan comparisons; you'll often be surprised, amused, and angry all at the same time. That said, trending is vital for tracking mitigation performance over time and quite valuable for denoting improvement or decline.

In conclusion

Make use of the ThreadFix wiki[7] to fill you in on the plethora of detail I didn't cover here and really, really consider standing up an instance if you're at all involved in application security discovery and repair. This tool is one I am absolutely making use of in more than one venue; you should do the same. You're probably used to me saying it every few months but I'm really excited about ThreadFix as it is immediately useful in my every day role. You will likely find it equally useful in your organization as you push harder for find and fix versus find and…

Ping me via email if you have questions (russ at holisticinfosec dot org).

Cheers…until next month.

Acknowledgements

—Dan Cornell, CTO, Denim Group, ThreadFix project lead

—Lee Carsten, Senior Manager, Business Development, Denim Group

About the Author

Russ McRee manages the Threat Intelligence & Engineering team for Microsoft's Online Services Security & Compliance organization. In addition to toolsmith, he's written for numerous other publications, speaks regularly at events such as DEFCON, Black Hat, and RSA, and is a SANS Internet Storm Center handler. As an advocate for a holistic approach to the practice of information assurance Russ maintains holisticinfosec.org. He serves in the Washington State Guard as the Cybersecurity Advisor to the Washington Military Department. Reach him at russ at holisticinfosec dot org or @holisticinfosec.

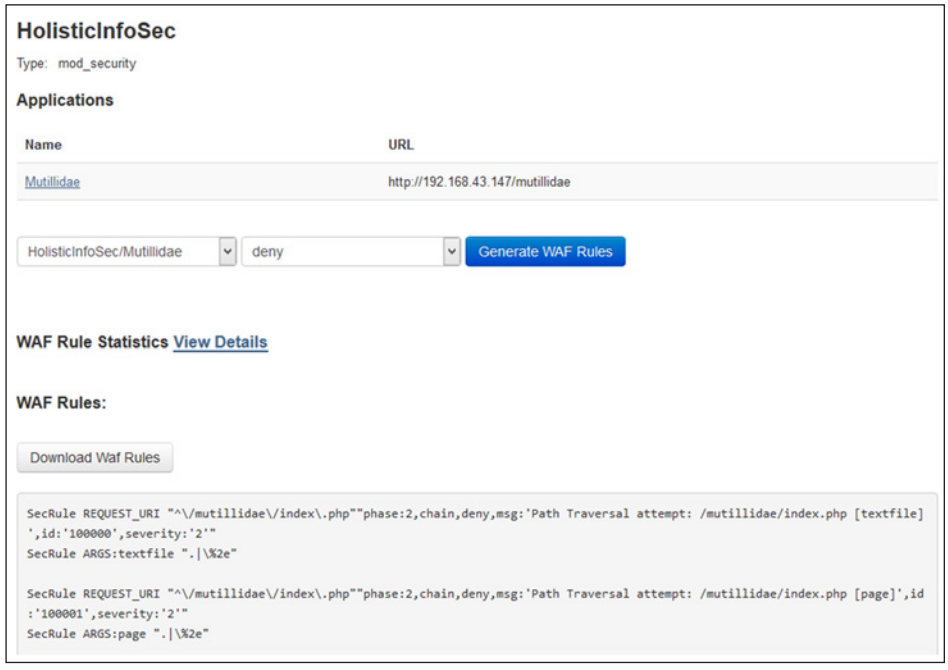

Figure 5 – ThreadFix generates mod_security rules

7 https://github.com/denimgroup/threadfix/wiki.