Case Study

ASTAM

MWR
an F-Secure company

Assessment of the Attack Surface Detector Burp Plugin

The Challenge

In the realm of application security, penetration testing organizations are faced with many serious testing challenges. Organizations developing software are continuously creating larger, more complex software systems, and, as a result, increasingly rely upon penetration testers to devise comprehensive testing strategies and detailed testing scenarios to ensure the security of the web applications they are responsible for testing.

Unlike functional testing, penetration testing requires a complete understanding of the security threats associated with the target web applications, and identification of all possible avenues by which an adversary may attack. Pen testers must ensure that every test is comprehensive, clearly covering the vulnerable attack surface of an application, while meeting high quality standards, and operating within a specified testing time box and budget. Consequently, pen testing is an increasingly expensive and time-consuming undertaking. Pen testing companies are faced with the challenge of reducing their costs and testing time while still providing valuable, consistent and reliable services to their customers.

The Solution

To meet these penetration testing needs, Secure Decisions developed the Attack Surface Detector (ASD) tool, a new application security testing technology, developed under the Department of Homeland Security Science & Technology Directorate multi-year funded ASTAM (Application Security Technologies and Metrics) program, that enhances and streamlines software penetration testing. The ASD tool was developed to provide a complete picture of a web application's exposed attack surface by automating the examination of a web application's source code via static analysis, discovering the application's hidden endpoints, optional parameters, and data types, and identifying gaps in an application's visible attack surface.

Such endpoints, parameters, and data types can often be missed by DAST scanners, and consequently, penetration testing firms must resort to manual code inspection to be confident that all endpoints, parameters, and data types are identified before penetration testing begins. The ASD tool greatly accelerates the attack surface identification process, saving substantial time for penetration testers. These discovered endpoints and parameters are then used to pre-seed penetration testing tools such as Burp Suite and OWASP ZAP, providing more thorough testing of a web application.

Evaluating the Solution

To better understand the value and effectiveness that the Attack Surface Detector brings to penetration testing, Secure Decisions commissioned an independent assessment of the Attack Surface Detector tool and invited MWR InfoSecurity, a CREST certified independent cyber security consultancy firm, to perform a unique evaluation.

To evaluate the ASD tool in a realistic setting, MWR used ASD in their assessment of a web application used in cybersecurity education. The web app, which was developed by another team at Secure Decisions, allows users to develop, publish, score, and manage educational comics. The application is also used for training exercises and can include branching storylines, quizzes, and the ability to publish comics. With a number of different inputs, user roles, and elements of business logic, this application has a significant attack surface. Two MWR testers performed two tests in parallel of the application—one test using the ASD tool, and the other using MWR's traditional testing methods. Without the use of the ASD tool, a tester was required to perform manual code review and interactive testing—both of which are potentially time-consuming and error-prone processes. This allowed MWR to understand how the ASD tool affects the assessment process.

Results and Recommendations

The use of the Attack Surface Detector tool provided an efficient and useful overview of the attack surface for the application under test. It offered considerable convenience and assurance and allowed for a faster and more reliable assessment of the application's attack surface than was available without the ASD tool. The manual testing process involved searching the source code for relevant keywords and patterns; this time-consuming process was automated and performed in minutes by testers utilizing the Attack Surface Detector. In addition, the tool's ability to reveal URL parameters that affect the behavior of endpoints was helpful in accelerating exploit development.

Penetration testers using the ASD tool realized substantial time savings simply by automating the endpoint discovery process. Analysts reported an estimated time savings of 4-6 hours in identifying application endpoints and parameters compared to the time it would take to perform the task manually. Additional time savings was achieved when using the tool to assist in exploit development.

Also noteworthy is that the application under test was deemed to be of moderate size and complexity, and had high-quality code style, which was very well-commented and organized. This facilitated manual analysis and made endpoints relatively easy to find using traditional methods. In an ideal world, all applications would meet this standard. However, in actual practice, lower quality, poorly-documented source code is more common. The ASD tool provides its greatest value when evaluating low quality, poorly documented code bases. In this situation, a tool that discovers endpoints by parsing code could prove more effective than manual testing where endpoints might be missed due to a mismatch with tester's search terms. In cases where larger, more complex applications are tested, the time and cost savings could prove even greater.

The MWR conclusion stemming from this evaluation was that the use of the Attack Surface Detector tool is of significant help when performing application penetration testing when source code is available.

Availability

The Attack Surface Detector tool has been developed as both a standalone Command Line Interface (CLI) tool and as a plugin for the widely adopted security testing tools, Portswigger Burp Dynamic Application Security Testing (DAST) tool Suite and OWASP ZAP. It is open source and freely available for download:

- https://github.com/zaproxy/zap-extensions/releases
- https://portswigger.net/bappstore