Penetration Tester's Suitcase – International

https://grayhatacademy.wordpress.com/

Section 16, Lecture 50

My articals on “international penetest magazine”, on the topic of how to keep web applications safe by performing penetration tests with well known and best of the free tools to scan, analyze, code review and protect web applications on your own.

                                  Pentester’s Suitcase

                                  Everything You Need to Keep Web Applications Safe

Front facing web segments are always the target of malicious hackers. This article explains about how to save web applications by using various tools and techniques. Pentesting web applications from hacker’s perspective reveal the pesky applications to the web that could be targeted by bad guys.

The most common web application security weakness is the failure to properly validate input coming from the client or environment before using it. This weakness leads to almost all the major vulnerabilities in web applications, like cross site scripting, SQL injection, interpreter injection, locale/Unicode attacks, file system attacks, and buffer overflows. Data from an external entity or client should never be trusted, since it can be arbitrarily tampered with by an attacker. **“Accept known good and reject known bad, this technique must be followed.”** That is rule number one. Unfortunately, complex applications often have a large number of entry points, which makes it difficult for a developer to enforce this rule. I will describe latest tools and techniques that will evaluate the security issues into web applications.

• There are lots of open source and paid web application auditing frameworks.

• Top 5 tools will be discussed here that I personally use for pentesting. At first, one of my favorite tools for auditing web applications is Burp Suite from Port swigger. Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application’s attack surface, through to finding and exploiting security vulnerabilities.

Burp Suite contains many key features:

• An intercepting proxy, which lets you inspect and modify traffic between your browser and the target application.

• An application-aware spider, for crawling content and functionality.

• An advanced web application scanner, for automating the detection of numerous types of vulnerability.

• An intruder tool, for performing powerful customized attacks to find and exploit unusual vulnerabilities.

• A repeater tool, for manipulating and resending individual requests.

• A sequencer tool, for testing the randomness of session tokens.

• The ability to save your work and resume working later.

• Extensibility, allowing you to easily write your own plug-ins, to perform complex and highly customized tasks    within Burp.

**Setting Up Intercepting Proxy for Nonproxy- aware Client**

 Sometimes when testing web applications, you may find yourself in such position that you need to use a thick client that runs outside of the browser. And hence, many of these clients do not let you configure an HTTP proxy, because it connects directly to the web server hosting the web application. At this stage, it will stop you from using an interception proxy to modify the requests.

Figure 1. Burp Suite – Editing request through intercepting proxy on the fly

**Burp Suite gives you some features that will let you continue at this stage. To do this, you need to follow these steps:**

Just modify your OS’s host file to resolve the address used by application to the localhost (127.0.0.1), for example 127.0.0.1 [http://www.grayhat.in](http://www.grayhat.in/). This will tell the thick client to redirect the traffic to your system. Now configure the Burp Proxy listener on port 80 or 443 (according to the port used by the application) of your loopback interface, and set the listener to invisible proxying. The invisible proxying means that the listener can accept the non-proxy requests sent by the thick client, which have been redirected to your loopback address. Invisible mode supports both HTTP and HTTPS. You will get certificate issue with this kind of features. It becomes necessary to configure invisible proxy listener to give an SSL certificate with a specific hostname which matches what the thick client applications expects.

You can find these settings under: Connections -> Hostname Resolution. It will let you define mappings for domain names to IP addresses to override your computer’s own DNS resolution. this causes the outgoing requests from Burp to be directed to the correct destination server. (if you not follow these steps, requests would be redirected to your localhost in an infinite loop).

Arachni

During a pentest of web applications I have used this tool that supports large amount of features. Arachni is a full- featured, high-performance Ruby framework that helps penetration testers and administrators evaluate the security of web applications. Unlike other scanners, Arachni supports the dynamic nature of web applications and can detect changes made while passing through the paths of a web application’s cyclomatic complexity. This way attack/input vectors that would otherwise be undetectable by non-humans are seamlessly handled by Arachni. Arachni can do a huge amount of jobs in pentesting web applications like:

• For forms, links, and cookies auditing.

• A wide range of injection strings/input combinations.

• Writing RFI, SQL injection, XSS, and others happens in fraction of seconds.

• Some of the more advanced Recons supported by Arachni are:

• Allowed HTTP methods • Back-up files

• Common directories

• Common files

• HTTP PUT

• Insufficient Transport Layer Protection for password forms

• WebDAV detection

• HTTP TRACE detection

• Credit Card number disclosure

• CVS/SVN user disclosure, Private IP address disclosure

• Common backdoors

• .htaccess LIMIT misconfiguration

• Interesting responses

• HTML object grepper

• E-mail address disclosure

• US Social Security Number disclosure

**OWASP Zed Attack Proxy Project**

 Auditing web applications becomes easy when a lot of tools with a lot of features are in the toolbox. One of those full featured tools is OWASP project Zed Attack Proxy. The Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications. ZAP provides automated scanners as well as a set of tools that allows you to find security vulnerabilities manually. Figure 3. Zed Attack Proxy It is designed in such a fashion by people with a wide range of security experience, and as such is ideal for developers and functional testers who are new to penetration testing. ZAP supported features are:

• Intercepting Proxy

• Automated scanner

• Passive scanner

• Brute Force scanner

• Spider

• Fuzzer

• Port scanner

• Dynamic SSL certificates

• API

• Beanshell integration (*Contd. to pdf resources added in this section*)

**Conclusion Web applications are always a main target of malicious hackers. In this era, almost everyone is going to be more dependent on the web from netbanking, social networking, online shopping, and so on. That attracts web application security experts to focus on security of that frontend web. Reputation plays a critical role mostly in business. Obviously, there is no doubt that web application security is a current and critical subject. For businesses that collect increasing revenue from Ecommerce, for users who trust web applications with sensitive information, and criminals who can make big money by stealing payment details or compromising your bank accounts. Some of them want to do business with an insecure website, so few organizations want to disclose details about their own security vulnerabilities or breaches. Hence, it is not 100% secure and sure that web application you are relying on is not vulnerable. With a simple and powerful tool like burp suite attackers can manipulate the entire validation scheme that resides at client side.**

*ATUL TIWARI – Penetration tester, EThical hacker* Atul Tiwari is the founder of an ethical hacking and information security training and service provider company “gray hat (P) Ltd Ranchi (India).” He is currently working for his own company and has been working for innobuzz knowledge solutions. He is has been information security and web app pentesting for 5 years and holds many certifications including “Diploma in Cyber Laws, GLC Mumbai,” Certified Ethical Hacker, CCNA, CISSP, web application security analysts. Atul is reachable at atul@grayhat.in.

Resources for this lecture

* More on my blog