My name is Paul Ionescu and I lead the IBM Security Ethical Hacking team. Today I will discuss number 9 in the OWASP Top 10 list of the most common web application flaws: Using Components with Known Vulnerabilities

There is a wealth of reusable software components available. Many of these components are Open Source developed with voluntary contributions and available for free. Developers can quickly build feature-rich applications by using third party components.

While the benefit of taking such an approach is obvious, companies need to account for the cost of security bugs if 3rd party components are used. In the case of Open Source Libraries, some of the code is 20 years old and the people maintaining the software are volunteers.

While security flaws in third party components can also be part of any of the other OWASP Top 10 categories, this category outlines the failure to update these components in a timely manner.

In 2014, two vulnerabilities raised huge media awareness. In the spring, Heartbleed, a vulnerability in Open SSL allowed attackers to siphon private information from public servers by leveraging a buffer overflow weakness. A notable case was the Heartbleed attack on the Government of Canada Revenue Agency by a university student. The attack occurred while Canadians were in the process of submitting their tax returns and exposed thousands of Social Insurance Numbers.

In the fall, a Shell Command Injection vulnerability was uncovered in the Linux command line and was dubbed Shellshock. The vulnerability existed for 20 years before being uncovered. The Linux command line is used in major operating systems such as Max OS X or Red Hat although not all systems or configurations are impacted.

Both vulnerabilities have had a wide impact and have sent companies scrambling to deploy security updates.

Here we have a vulnerable web application running on Apache and Tomcat. The OpenSSL component was not updated.

Hereâ€™s how we can exploit Shellshock. I am using a custom Python script which sends malformed heartbeat requests to the web server in order to leak information from the web server memory.

As you can see, we are able to observe portions from the last HTTP request sent to the server.

I am going to capture a browser request using a HTTP Tamper Proxy tool and append a header containing user credentials to it. We want to see if this confidential information can be obtained using the python script.

Indeed we can observe that the output of the Python script displays the sensitive header.

Now a demonstration of the ShellShock vulnerability. Looking at the HTTP capture of the browser traffic that we performed earlier, we observe that one of the requests goes to a CGI script.

Here it is, the request to cgi-bin.

A CGI script can be written using Linux commands, and is vulnerable to the type of Shell Command Injection vulnerability that is Shellshock.

The attack is done using the HTTP headers, because the web server will populate header values in environment variables used by the command line interpreter.

Here we will modify the User-Agent header to inject three additional commands. The string construct at the beginning is a function definition that causes the vulnerability. The first two commands serve the purpose to complete the HTTP header syntax that the Web-Server requires from the output of the shell script. The last command is the arbitrary command that we want to execute, in this case we simply want to output SUCCESS.

Besides echoing strings, we want to take control of the system and execute commands as if we were sitting in front of a terminal, so hereâ€™s a few examples of how we can do that.

Here we are using the â€œlsâ€ command to list the directory structure. The string at the end redirects standard input to this CGI script output.

Hereâ€™s another example of using the â€œfindâ€ command to locate a password file. To escalate privileges on the system, an attacker would look for password files.

Now we are going to use the â€œcatâ€ command to list the contents of the password file. Luckily the password is in clear text.

People reuse passwords all the time, so letâ€™s see if this password allows us to login as the root user.

You can use a vulnerability scanner such as AppScan to detect outdated 3rd party components such as Shellshock and Heartbleed. You can see that the AppScan 3rd party components category contains tests for both the vulnerabilities along with many other tests for other 3rd party vulnerabilities. Simply point AppScan to the site youâ€™d like to test, trigger a scan using the 3rd party policy, and find out if you are using any of these outdated libraries.

Thank you for watching and I wish you best of luck in developing and maintaining secure applications!