Lab 5: Web Application Security

CS 460 Spring 2015: Security Lab

Due 11:59 02/26/15

Overview

This lab will take a brief tangent to discuss secure messaging, and then focus on web penetration testing and web application exploitation. The VMs are not necessary at any point for this week's labs. You can check out your SVN respository for this lab using the following command:

```
svn checkout --username <YOUR NETID HERE> https://subversion.ews.illinois.
    edu/svn/sp15-cs460/<YOUR NETID HERE>/lab5
```

Alternatively, you could use git to manage your subversion repository, and checkout using

```
git svn clone https://subversion.ews.illinois.edu/svn/sp15-cs460/<YOUR
    NETID HERE>/lab5
```

You will have to do your own research about git svn should you decide to use it, course staff will not be officially supporting it.

1. Part 1 - Secure Messaging
2. Part 2 - WebGoat

Part 1 - Secure Messaging

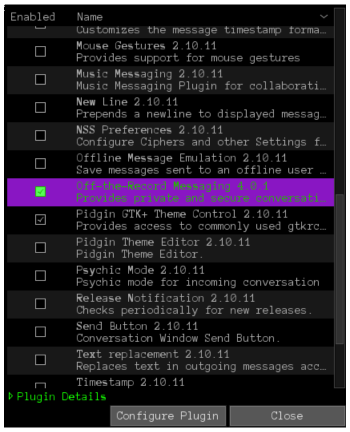
In this part of the lab, you will learn one convenient method to encrypt your communications to ensure that individuals sniffing on the network cannot deduce what you are talking about, be it communication between bots on your botnet, sources leaking information to journalists about human rights abuses, or simply you chatting with your friends. We will continue our use of pidgin to cover installing and using the Off-The-Record encryption plugin.

On Windows, go to the OTR site at:

`https://otr.cypherpunks.ca/`

Grab the .exe installer, and you should be good to go. Most Linux distributions have a package called "pidgin-otr" in their package repositories. Adium for Mac OS X will have OTR preinstalled. To check, go to tools → plugins and look for the OTR plugin in the list of preinstalled plugins. If you choose to use your ubuntu VM, the packages you need would be:

```
sudo apt-get install pidgin-otr
```



Enabled	Name
	Customizes the message timestamp forma...
☐	Mouse Gestures 2.10.11 Provides support for mouse gestures
☐	Music Messaging 2.10.11 Music Messaging Plugin for collaborati...
☐	New Line 2.10.11 Prepends a newline to displayed messag...
☐	NSS Preferences 2.10.11 Configure Ciphers and other Settings f...
☐	Offline Message Emulation 2.10.11 Save messages sent to an offline user ...
✓	Off-the-Record Messaging 4.0.1 Provides private and secure conversati...
✓	Pidgin GTK+ Theme Control 2.10.11 Provides access to commonly used gtkrc...
☐	Pidgin Theme Editor 2.10.11 Pidgin Theme Editor.
☐	Psychic Mode 2.10.11 Psychic mode for incoming conversation
☐	Release Notification 2.10.11 Checks periodically for new releases.
☐	Send Button 2.10.11 Conversation Window Send Button.
☐	Text replacement 2.10.11 Replaces text in outgoing messages acc...
☐	Timestamp 2.10.11

Now, you must chat staff@54.191.94.255. When you send a message, you will se an option in the message window under OTR → Start private conversation. You must enable that. Send the admin a message like hello. Course staff monitors the admin account, and when you send a properly encrypted message, will send you back a flag. It may take some time, so if you can just leave pidgin running in the background and Course Staff will get back to you eventually.

Hand-in

1. Put the flag in flag1.txt and commit it to subversion.

Part 2 - Web Goat

This week's lab will be performed on WebGoat, a web application tool that was built intentionally vulnerable for students to learn to break into web applications. Course Staff have prepared such a vulnerable server for you to audit. In order to access the WebGoat server, you must use a technique called SSH tunneling, which allows for you to connect a port on your local computer to a port on a remote computer through ssh. On Mac OS/Linux, you can use this command to open up a port on your local computer:

```
ssh -f -N -L 8080:localhost:8080 class@ec2-54-68-11-241.us-west-2.compute.
    amazonaws.com
```

The tool PuTTY for Windows should theoretically be able to perform ssh tunneling, however when testing PuTTY with windows, Course Staff was unable to make the ssh tunnel work, so should you be using windows, Course Staff recommends either using a linux VM or going to an EWS lab to complete the lab.
The password you will have to provide is shown below:

```
thisisaverysecurepassword_cs460isaclass535975359@@@@@@####$$$$++++
```

Once you have the tunnel open, leave it running in another window, and then navigate to the URL shown below in your web browser:

```
http://127.0.0.1:8080/WebGoat/start.mvc
```

This navigates to localhost on your computer, which because of the ssh tunnel, redirects to port 8080 on the remote machine (which you wouldn't have typically been able to get to because firewalls were blocking them). To log in, use the credentials provided to you in your svn repository in the file `password.txt`.
WebGoat is a pretty famous Web Application learning tool, so many write-ups with answers and discussion of techniques exist on youtube and you can google around for answers. Course staff is hoping that you are a self-motivated person who will not cheat, and will actually try to learn

how to exploit Web Applications, although the guides and writeups are there in case you need a crutch to help.

The following tools may come in handy when learning WebGoat:

1. Zed Attack Proxy
   (a) A Proxy will allow you to "catch" a web request before it is actually made to the server and manually inspect it or modify it before sending it out to the world. ZAP is one such proxy. Useful Tutorial Video Here.
2. Firebug for Firefox
   (a) Firebug is a debugging tool for Web Developers that allows you to make changes to what the browser sees in HTML, JavaScript, CSS on the client side only. A pdf you can read to learn how to use it is Here.
3. SQL Inject Me & XSS Inject Me
   (a) These are automated tools that will allow you to test a site for SQL and XSS. We will not be using them for WebGoat, however they are fun to have around.
4. REST Easy for Firefox
   (a) REST Easy will give you some of the Proxying features of ZAP in-browser. More Info Here.
5. User Agent Switcher for Firefox
   (a) This will allow you to pretend to have a different web browser when surfing the web. Not strictly necessary for this lab, but useful: Developer's Website.
6. Cookies Manager+ for Firefox
   (a) This tool adds features to Firefox's default Cookie managing, and lets you edit the cookies your browser has. Link.

Hand-in

1. In Webgoat, under General:
2. Under "Access Control Flaws:"
   (a) "Bypass a Path Based Access Control Scheme"
3. Under "AJAX Security:"
   (a) "DOM Injection"
4. Under "Authentication Flaws:"
   (a) "Password Strength"
   (b) "Forgot Password"
5. Under "Cross-Site Scripting (XSS)", perform the following modules:
   (a) Stage 1 XSS
   (b) Stage 2 XSS
   (c) Stage 3 XSS
   (d) Stored XSS Attacks
   (e) Reflected XSS Attacks
   (f) Cross Site Request Forgery
   (g) CSRF Prompt By-Pass
   (h) CSRF Token By-Pass
6. Under "Improper Error Handling:"
   (a) "Fail Open Authentication Scheme"
7. Under "Injection Flaws:"

   (a) "Numeric SQL Injection"
   (b) "XPATH Injection"
   (c) "String SQL Injection"
   (d) "Blind String SQL Injection"

8. Under "Denial of Service:"
   (a) "ZipBomb"
   (b) "Denial of Service from Multiple Logins"
9. Under "Insecure Configuration"
   (a) "Forced Browsing"
10. Under "Insecure Storage"
    (a) "Encoding Basics"
11. Under "Parameter Tampering"
    (a) "Bypass HTML Field Restrictions"
    (b) "Exploit Hidden Fields"
    (c) "Exploit Unchecked Email"
    (d) "Bypass Client Side JavaScript Validation"

You do not need to specifically turn anything in for this part of the lab. Course Staff will know which modules you performed in WebGoat automagically.

Deliverables:

Turn in this assignment on SVN. Autograder scripts will run every night around 11PM starting on Saturday night, putting grade.txt files in your SVN repository. The following should be committed to your SVN repo:

1. The flag you got from part 1.
2. WebGoat will be graded automagically.