Audrey Long

JHU Web Security

Module Assignment: ToR Onion Service

Introduction

We talked a bit about ToR in the lectures so we'll use this assignment as an opportunity to explore some of ToR's inner-workings more deeply, namely by creating a ToR hidden (Onion) service.

Assignment Objectives: You will install a VM, Web Server, and create a ToR hidden service, document your progress, and answer three topical questions demonstrating your application of the knowledge.

1. Virtual Machine

a. If you have not already please download this clean Ubuntu 18.04 (Bionic Beaver) VM and import it into VirtualBox by clicking 'Machine' -> 'Add' and selecting the *.ova file you downloaded.

b. Install the ToR Client

c. Install the ToR client (not the browser) on your VM by doing 'sudo apt-get install tor' in a Terminal window. Verify the install was successful by issuing the 'sudo tor' command

d. Artifact for deliverable: Create a deliverable document, add a screenshot of the output of your Tor command results.

```
student@student:~$ sudo tor
Apr 20 16:35:10.717 [notice] Tor 0.3.2.10 (git-0edaa32732ec8930) running on Linux with Libevent 2.1.8-st
able, OpenSSL 1.1.0g, Zlib 1.2.11, Liblzma 5.2.2, and Libzstd 1.3.3.
Apr 20 16:35:10.719 [notice] Tor can't help you if you use it wrong! Learn how to be safe at https://www
.torproject.org/download/download#warning
Apr 20 16:35:10.719 [notice] Read configuration file "/etc/tor/torrc".
Apr 20 16:35:10.761 [notice] Scheduler type KIST has been enabled.
Apr 20 16:35:10.761 [notice] Opening Socks listener on 127.0.0.1:9050
Apr 20 16:35:10.761 [warn] Could not bind to 127.0.0.1:9050: Address already in use. Is Tor already runn
ing?
Apr 20 16:35:10.761 [warn] Failed to parse/validate config: Failed to bind one of the listener ports.
Apr 20 16:35:10.762 [err] Reading config failed--see warnings above.
student@student:~$
```

Figure 1: Output from tor command

2. Install a Webserver

a. For this step you're welcome to use any web server you're familiar with or interested in working with for the first time. Good candidates include Apache Tomcat, lighttpd, and nginx. Whichever webserver you choose, be sure to make note of which default port it uses to accept incoming connections (usually 80 or 8080).

b. Use Firefox, 'curl', or 'wget' to verify your webserver is running correctly

c. Artifact for deliverable: Add to your deliverable document a screenshot of your web server welcome page.

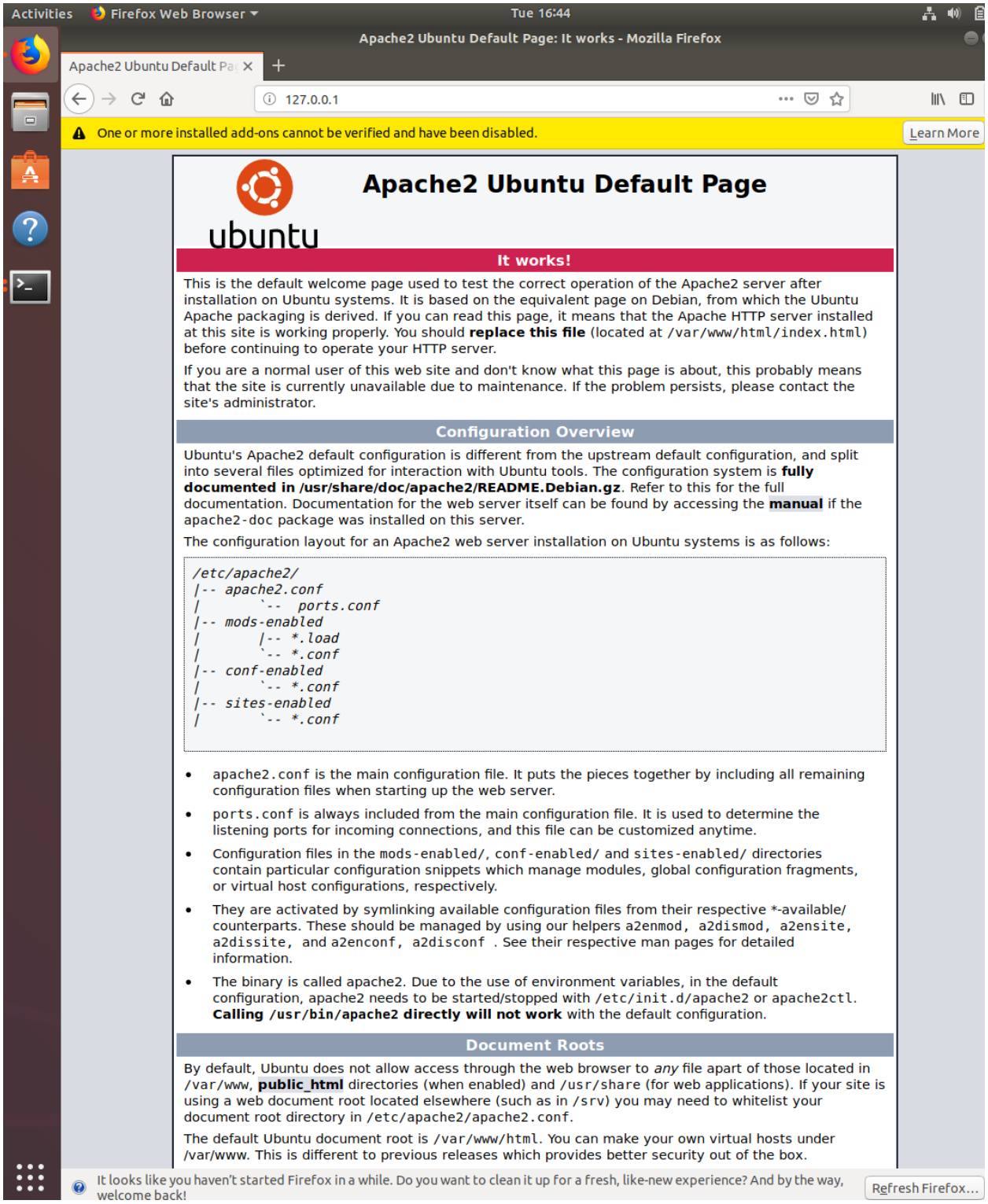

Figure 2: apache welcome page

3. Configure a ToR Hidden Service

a. Use the following guide to create a ToR hidden (onion) service by beginning at the “Prepping the actual service (web, SSH)” step.

b. Artifact for deliverable: Add to your deliverable document a screenshot for each configuration change you make while creating your *.onion service.

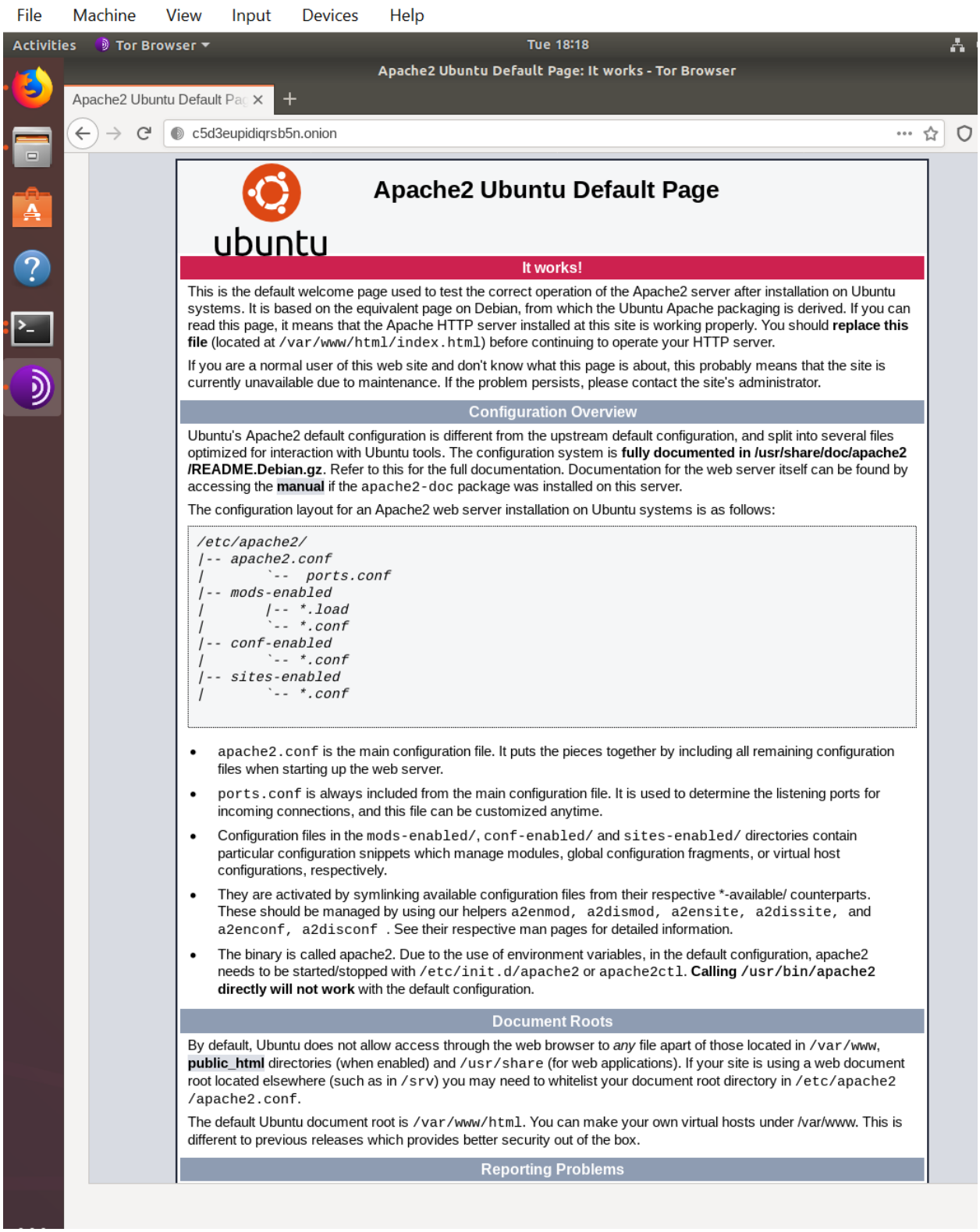

Figure 3: accessing the service on Tor

4. Deliverables

Submit a PDF containing:

1) The previous artifact screenshots
2) The URL to your onion service : **c5d3eupidiqrsb5n.onion**
3) Answers to the following questions:

a. Why isn't it necessary for clients to use HTTPS to ensure the confidentiality of their interaction with a ToR hidden service?

Even though HTTPS is the standard to ensure that traffic is traveling through an encrypted channel and the data itself is encrypted in transit, Tor uses its own source of protocols within the hidden service network which resolves down to the .onion web extension through tunneling which does not connect directly to exit nodes. This can be accomplished with two way anonymity where the server does know the IP of the client, and the client doesn't know the IP of the server which provides protection on both sides of the transaction which ensures confidentiality of the traffic because adversaries cannot see the traffic and cannot tie the traffic back to a source. Below I go over the handshake and encrypted tunneling process which shows the confidentiality has been preserved with the strenuous process. With that being said, plenty of software bugs can exist in the Tor browser, protocols, and payloads where someone out there can or has potentially exploited information for Tor networks which is inevitable no matter what the service.

b. How are ToR *.onion address resolved if they are not published to public DNS Servers?

Users of the dark web do not use public DNS to resolve .onion extensions to public IP addresses. Instead, it gets resolved in the Tor hidden service protocol which makes the extension known and redirects clients to the domain services while providing anonymity to both the client and the server. The first step is for the Tor host domain to be available for users by publishing a service descriptor which contains a public key and a list of Tor nodes which will provide an entryway to the hidden service. Tor then creates connection strings to the listed points which a client can connect to. Connection then gets established by the client querying the service descriptor and randomly selecting an entry point which is anonymously connected and transmits a message to the hidden service containing the identity of the entry point encrypted with the public key to begin the encrypted handshake process. Then the hidden services created a connection back to the chosen point and completed the handshake which establishes a private network pathway resistant to surveillance and can exchange data anonymously and confidentially.

c. Does the person running the hidden service know the identity of the client sending requests to their service or are requests made to onion services anonymously? Explain.

No, the person running the hidden services does not know the identity of the client, and inversely the client does not know the identity of the server.Tor-only service that gets its own

.onion hostname. When visitors connect to the Tor network, Tor resolves those .onion addresses and directs you to the anonymous service sitting behind that name [1]. Detailed in the previous response, it can be seen how the anonymous handshake occurs along with a list of optional entry points are provided to the client, which then sends a message to the server to generate a rendezvous point to continue the translation all under the umbrella of anonymity and encryption.

Grading Rubric for this assignment

Assessment	Points
1. Artifact Screen shots indicating successfully completing the tasks assigned.	30
2. URL of the student's onion service.	5
3. Topical Question narrative:	
a. Did the student demonstrate knowledge application of the module material in the subsections?	
a.	15
b.	15
c.	15
b. Did the student make an effort to provide something substantive and new to the topic discussion?	10
c. Does the submission look professional with spelling and grammar at master's level?	10
TOTAL	100

References

https://www.linuxjournal.com/content/tor-hidden-services [1]

https://www.icann.org/en/blogs/details/the-dark-web-the-land-of-hidden-services-27-6-2017-en [2]