Please read all sections of this specification carefully before starting to work.

After reading this coursework specification, I strongly suggest that you make yourself a detailed check-list of the submission requirements.

Definition

Remediation; noun. The action of remedying something, of reversing or stopping environmental damage.

Background

Security vulnerabilities are often difficult to spot and therefore they can remain dormant for a very long time. Security researchers will spend their time trying to uncover such vulnerabilities, and their task is to both identify the vulnerability and test whether it may be exploitable. Once a researcher has found an exploitable vulnerability, responsible disclosure requires them to contact the vendor and request a patch for the problem, thereby removing the exploitable situation for their population of users. In some cases, a vendor (for one reason or another) may not respond, and it is the security researcher's responsibility to bring the vulnerability to the attention of the security industry. In order to do this, the researcher will often publish the full vulnerability along with a sample exploit so that others can test the situation. The side effect of this is that the vendor is shamed into producing a patch as quickly as possible.

During the time between the researcher revealing the vulnerability in a public forum, and the vendor supplying a patch, all the users with that specific software running on their computer systems will be in a critically vulnerable position.

In this coursework, you will work with two known vulnerabilities. Assuming that patches are not available, you will research and implement remediation strategies for these vulnerabilities. You will demonstrate your mitigations on your CTEC2912 Ubuntu VM, but keep in mind that Ubuntu 18 and above are not actually vulnerable.

Each remediation strategy will both:

A) Reduce the likelihood of the vulnerability being exploited, and/or the likelihood of an exploit achieving its goals.
B) Trigger an alarm in the event of an exploit for the vulnerability being activated.

Vulnerabilities

Vulnerability 1: Heartbleed

Heartbleed (CVE-2014-0160) is a vulnerability in openssl, a library that enables webservers to serve websites over an SSL/TLS encrypted channel. Heartbleed allows remote attackers to read the memory of the webserver, possibly compromising the server's secret keys.

Vulnerability 2: ShellShock

ShellShock (CVE-2014-6271) is a vulnerability in GNU Bash, which allows remote attackers to execute arbitrary code.

Prerequisites

Before you start on the coursework, make sure that the necessary software is installed on your VM. You will need, at the very least, apache, ssl, and python. The following commands will take care of the installation:

```
sudo apt install apache2
sudo apt install ssl-cert
sudo a2enmod ssl sudo
a2ensite default-ssl sudo
systemctl reload apache2
sudo apt install python
```

Note that you may have to install further software to complete some of the tasks below.

Exploits

To test whether your remediation strategy works, you are given a proof-of-concept exploit for each vulnerability[1]. Download the Python scripts from Blackboard and save them in your home directory.

1. **Heartbleed**: run the following command from a terminal in your home directory.

   ```
   python heartbleed-exploit.py localhost
   ```

2. **ShellShock**: run the following command from a terminal in your home directory.

   ```
   python shellshock-exploit.py rhost=localhost payload=reverse
   lport=1234 lhost=localhost
   ```

Tasks

There are four tasks in this coursework. You need to complete each task **twice**, once for Heartbleed and once for ShellShock.

Task 1

For each vulnerability, familiarize yourself with the vulnerability. You will need to conduct your own research to find out how it works, why it might exist, why the vulnerability may result in unintended behaviour of the vulnerable software, and how common exploits work.

Document your findings in your report, including technical detail and explanations.

[1] Sources for the exploits:
https://www.exploit-db.com/exploits/32745
https://www.exploit-db.com/exploits/34900

Task 2

For each vulnerability, write firewall rules for `iptables` that mitigate the vulnerability. Your iptables rules need to:

- reduce the likelihood of the vulnerability being exploited, and/or reduce the likelihood of an exploit achieving its goals, and
- create log entries each time an exploit is attempted.
- You can use the proof-of-concept exploits to show that your firewall rules work.

Document your firewall rules in your report.

- Include the commands used to create the firewall rules
- Include detailed explanations for each part of the iptables rules
- Include screenshots of the relevant portions of your /var/log/syslog and /var/log/kern.log

Task 3

For each vulnerability, design a technique that will trigger an alarm in the event of an exploit for the vulnerability being activated. Implement your alarm trigger technique and use the proof-of-concept exploits to trigger the alarm.

You will succinctly write up the technique in the report.

- Use screenshots to demonstrate that your alarm trigger technique works.
- Document all commands and/or configuration files that you have used to implement your alarm trigger.
- Include in the report your reasoning for choosing this technique as well as a critical evaluation of your alarm trigger technique and the processes which you have implemented. You can use your research from Task 1 to support your reasoning.

Task 4

For each vulnerability, research one additional remediation strategy that reduces the likelihood of the vulnerability being exploited, and/or the likelihood of an exploit achieving its goals. Critically compare this additional remediation with the iptables remediation you have implemented above.

Note that the official patch is NOT an acceptable remediation strategy. Your remediation strategy must work in the hypothetical situation that no patch is available yet.

You will succinctly write up the strategy in the report. Include in the report

- detailed explanations how the additional remediation strategy works,
- to what extent it mitigates against exploits,
- your reasoning for choosing this strategy, and
- a critical evaluation of the remediation strategy, including a comparison with the iptables based remediation you have implemented in task 2.

Submission

You must submit a single document with a word count of 3000 words, excluding references/bibliography and appendices. You must display the word count figure on your title page.

Your submitted document must be in PDF format. If the file is too large, then the appendices should be submitted separately using a cloud storage account, such as Dropbox, Google Docs/Drive, or Microsoft OneDrive. In this case, a link to the appendices should be included in the submitted document.

Your report will include (as a minimum) a title page, introduction, summary and references/bibliography. Ensure all imported/referenced material is properly crossreferenced.

- The report will contain, **for each vulnerability**:
  - A description of the underlying vulnerability, including technical explanations for why and how it can be exploited
  - A description of the iptables firewall rules o A description of the alarm trigger technique
  - A description of one additional remediation strategy which prevents the exploit in systems that do not have the official patch, including detailed reasoning for choosing this strategy and a critical evaluation of the remediation strategy.

Bonus marks

- Quite brilliant submissions may attract more marks than the documented maxima for each stage.

- You can achieve up to 10 bonus marks by implementing the additional remediation strategy from task 4 in your Linux machine. Clearly document your implementation, including all commands/configuration files used and screenshots as appropriate.

Notes

- Read this specification in conjunction with the marking scheme, available as a separate document.
- Make copious notes of everything that you do. It will make the report writing much easier.
- Take screenshots as you progress. Use these to illustrate your report.
- After reading this coursework specification, I suggest you make yourself a check-list of the submission requirements.