Network Security

Assignment 6, Tuesday, June 13, 2018, version 1.0

Handing in your answers: Submission via Blackboard (http://blackboard.ru.nl)

Deadline: Wednesday, June 20, 23:59:59 (midnight)

Teaching assistants. Please email *all* of us if you have a question.

- Pol Van Aubel <pol.vanaubel@cs.ru.nl>
- Daan Sprenkels <dsprenkels@science.ru.nl>
- Wouter Kuhnen <w.j.a.kuhnen@student.ru.nl>

This final assignment uses the homework WiFi network again.
In this assignment you will be using the following tools:

- aircrack-ng: http://www.aircrack-ng.org/
- arpspoof: http://www.monkey.org/~dugsong/dsniff/
- nmap: http://nmap.org/
- sslstrip: http://www.thoughtcrime.org/software/sslstrip/, https://pypi.python.org/pypi/sslstrip/0.9.2
- wireshark, tshark or tcpdump for packet capturing: https://www.wireshark.org/

Again, do not compile these programs from source, but install them using your distribution's package manager.

The assignment consists of one long practical exercise. Please turn in all your work in plain text files (program source files are also plain text), unless specified otherwise. If you prefer a document with formatting for whatever reason, like including images, use the PDF format to turn in your work (most editors allow you to export to PDF). Note that it's okay to include images separately and then refer to them from within the text files.

Commands that need to be run with root rights are denoted by a prefix `#`. When a command should be run without root rights, it will be prefixed with `$`. Do not include the prefix when typing the command.

1. Create a folder called `exercise1`.

   This exercise is a multi-stage attack. Somewhere, there's a website containing your grades for this exercise. Everybody starts out with an O. It is up to you to give yourself the grade you want. (Note that you can still receive an NSI if we feel you have not put in sufficient effort.)

   You are *not* allowed to sniff the general network traffic in order to eavesdrop on other groups performing the attack. Also, please do not change other people's grades while performing this exercise, and don't do anything else on the target website.

   (a) Although WPA2 is more secure than WEP, just like any other good cryptographic system it is only as strong as the key material in use. To demonstrate this, you will use aircrack-ng to crack the passphrase of the wireless network where the course administrator is working. We have already taken care of capturing the WPA2 connection handshake, you can download it at `http://www.cs.ru.nl/~paubel/netsec/2018/handshake.cap`.

   To crack WPA2 passphrases, you need wordlists. A tutorial on how to crack WPA is on `http://www.aircrack-ng.org/doku.php?id=cracking_wpa`. Ignore the stuff about injecting packets, capturing the handshake etc. We've already taken care of that. The interesting part is section 4. Pointers on where to find wordlists are on `http://www.aircrack-ng.org/doku.php?id=faq#how_can_i_crack_a_wpa-psk_network`. Since it is not our intention to have you spend hours on WPA cracking, use the wordlist at `http://www.cs.ru.nl/~paubel/netsec/2018/GDict_v2.txt.tar.gz`. Note that you have to unzip it first (`tar -xzvf GDict_v2.txt.tar.gz`).

   The bssid of the network is 00:0f:c9:0c:f7:93. If you want to decrypt the capture to see whether you have the correct key, you also need the essid. This is "netsec-wpa". The capture should contain a single DHCP packet. Beware of the Ubuntu decryption bug, however: if you see other stuff you may still have the correct key. The best way to check is to try to connect to the network.

   Keep in mind that the network may not have a running DHCP server so if you fail to connect, try to set a static IP address in the 192.168.84.200–249 range, with netmask /24 and gateway 192.168.84.1.

   Write the passphrase you found to a file called `exercise1a`.

   (b) Connect to the network. There should be a DHCP server running. If not, use an IP address in the range of 192.168.84.200–249, with netmask /24 and gateway 192.168.84.1.

   Use nmap to scan this network. Find the hosts in the range 192.168.84.1–80. Disable reverse DNS lookup to speed up things. There should be many hosts, apart from the gateways (192.168.84.1–20). Write which hosts you find to `exercise1b`.

   (c) From this point onwards you will need to coordinate with other groups, since there is only a limited number of hosts to arpspoof. Do not get in each others way.

   Pick one of the hosts that are not the gateways (192.168.84.1–20). Its gateway is matched modulo 20 (so 192.168.84.32 and 192.168.84.52 both have gateway 192.168.84.12, whereas 192.168.84.23 has gateway 192.168.84.3).[1]

   Using arpspoofing and wireshark, figure out which websites this host is contacting. Save the network capture in `exercise1c.cap`. Write the URLs to `exercise1c`. Note that you may need to also arpspoof its gateway.

   NOTE: There is some delay between requests in order to not abuse the target website. This delay is approximately 300 seconds as of this writing.

[1] This somewhat weird network configuration is required to enable you to arpspoof in parallel with other groups. Without going into too much detail, the problem is that if we only had one gateway IP address, we would need as many hardware devices as IP addresses for you to spoof. In most normal situations, a network only has one gateway which all clients will use.

(d) Now, use sslstrip (`http://www.thoughtcrime.org/software/sslstrip/`, `https://pypi.python.org/pypi/sslstrip/0.9.2`) to strip out SSL from its web traffic. The documentation and explanation on the websites should be enough to get it to work.

Look at the traffic in wireshark and figure out the login credentials to use. Save the network capture in `exercise1d.cap` and write the login credentials you found to `exercise1d.creds`.

(e) Now, log in to the website, find your grades, and edit them to your desired result. After that, write your student numbers and the result you set to `exercise1e`.

(f) Give a brief description of the process of SSLstripping: what does it do? How does it receive the traffic it should strip SSL from? Does it simply forward traffic, or does it rewrite outgoing traffic as well? Write your answer to `exercise1f`.

(g) Can you think of countermeasures that clients and servers can take to alleviate the threat of SSLstripping? Write your suggestions to `exercise1g`.

2. Place the directory `exercise1` and all its contents in a folder called `netsec-assignment6-SNR1-SNR2`. Replace `SNR1` and `SNR2` by your respective student numbers, and accomodate for extra / fewer student numbers.

   Make a `tar.gz` archive of the whole directory `netsec-assignment6-SNR1-SNR2` and submit this archive in Blackboard.