Lecture Network Security
Assignment Sheet 3

Jens Tölle, Wolfgang Moll
Jonathan Chapman, Martin Clauß, Martin Lambertz, Christian Meier

Summer Term 2016

Publication date of this sheet:	12.05.2016
Submission deadline (via email):	24.05.2016, 23:59:59
Discussion of results:	25.05.2016

Results must be submitted via email to
network-security-worksheet@lists.iai.uni-bonn.de
in one archive file named after the scheme
`sheet3_lastname1_lastname2[_lastname3].{tar|tar.gz|tgz|zip}`

Task 3.1 (theoretical): Authentication Beyond Passwords

Research and briefly explain three alternatives for password-based authentication.

ps -Af

Part (a)

For each method, describe at least one advantage and one disadvantage in comparison to password-based authentication and name a popular service, site or product using this method.

Possessing something / inherent features - Biometry, key in hardware and key in software
Knowing something
Asking another authority

Part (b)

Due to the weaknesses of purely password-based authentication, more and more service providers offer a so-called two-factor authentication. Find an example for such a service and describe the advantages and disadvantages of the two authentication methods combined in comparison to purely password-based authentication.

Task 3.2 (theoretical): Reconnaissance in the SecLab

You already know that there are a bunch of computers in the SecLab. When hackers try to break into a network, they start with reconnaissance of their target and try to collect as much information as they can.

Log in to the SecLab with your personal account that you should have received per email. Find out as much as you can about the lab: IP addresses, running services, operating systems, etc. When gathering information, please do not use tools or methods that may disturb or kill services in the lab.

Submit:

uname -r
lsb_release -a
cat /etc/issue
lspci -v
ifconfig
ps aux
sudo netstat -tap | grep "LISTEN"

users
who
last
cat /etc/passwd
sudo cat /etc/shadow
ifconfig -a
nmap -sV 10.0.0.10
nmap -sV -Pn 10.0.0.13
route

- a list of tools you used,
- source code of self-made programs/scripts, and
- all information you gathered.

Feel free to discuss tools and methods with your fellow students on the mailing list (however, do not post your findings).

Bonus: Candy Hunt

We hid some candy in form of recognizable strings in the SecLab for you. For example

- a file named `bonbon` or
- a network service returning `lollipop` on correct input

would be such candy. Can you find the candy that we hid in the SecLab for you?

Task 3.3 (practical): DNS sniffing

You have heard about weaknesses in the Domain Name System (DNS) in the lecture. You will be developing a DNS spoofer during this and the next assignment sheet. To start with, this task covers the sniffer part, which you will extend in the next assignment sheet to actually send fake responses.

Write a program that monitors all DNS related UDP traffic and prints out the relevant protocol fields of requests and responses you see.

DNS - the seventh layer - application layer
QNAME QTYPE QCLASS
TTL RDLENGTH RDATA

Submit:

- a list of the DNS protocol fields required to spoof a response as well as a brief description of why they are important,
- the source code of your tool, and
- a sample output for both requests and responses *recorded in the SecLab*!

Task 3.4 (practical): Hash Collisions

Let's build our own super-collider!

Write a small program that generates two independent random byte sequences with length at least 64 bytes and calculates the SHA256 digest on these sequences. Repeat this step until you found a collision on the first four bits, i. e. the 4-bit prefix, of the hash values.

Now run this program five or more times and note how many tries you needed until a collision occurred. Repeat this step comparing the first 8, 12, 16 and 20 bits and create a plot indicating how many counts you needed until a collision occurred for each of these bit lengths. Add the average count needed for creating a collision on each of the prefixes and connect the consecutive averages with a line.

Your hand-in must include:

- the documented source code of the collider

int.from_bytes

hash collision wikipedia

- the data file(s) generated by the collider
- the documented plotting script
- the output of the plotting script as PDF or PNG file

Task 3.5 (theoretical): Designing Asymmetric Encryption Schemes

Confidentiality
Integrity
Availability

MitM

Part (a):

Bob has the idea to use the following asymmetric method to secure the transport of objects: To be able to securely transmit objects (e. g. chocolate or money), he uses a lockable box. Since it is not possible to exchange keys with the partner, Bob locks the box with a padlock. Only Bob has a key to this padlock. The locked box is sent to Alice by a parcel service. The lock provides integrity and privacy, i. e., nobody is able to see or change the content of the box. Alice cannot open the box, since she does not own a key to the padlock. Thus, she adds a second padlock to the box. Alice has the only key to this second padlock. The parcel service returns the box – secured with two padlocks now – to Bob. Bob removes the first padlock (the one he has a key for) and sends the box back to his partner. Alice receives the box, removes the second padlock and opens the box.

Is this method secure? Does it also work with cryptographic means? Which problems could arise?

Part (b):

Bob believes in his method. Thus, he wants to send digital data using the same approach. Instead of padlocks and keys, Bob chooses a secret key and XORs every character of the clear text with one character of the key (key length is at minimum the length of the clear text). The partner also chooses his own, different secret key. The method is now performed as in Part (a).

Does it work? Can confidentiality and integrity be assured? Would choosing different random keys for each message have an impact?