Assignment 3

Abbas Khan , Mariia Rybalka , Linara Adilova

May 24, 2016

Task 3.1 (theoretical): Authentication Beyond Passwords

As passwords based authentication showed itself to be not very sufficient, taking into account all the possible attacks and users choosing weak passwords, some alternatives are finding their way.

Biometrics measures

One of the ways to identify user is to check his(her) biological unique data, such as iris, heartbeat, ear form, fingerprint, etc. There are already a lot of examples of using this technique, among most well known can be named iPhone, that uses fingerprint in order to unlock the device. Main advantage of this way of authentication, that user does not have to remember any information - he just uses his body as a password. Disadvantage - this method is not 100 percent reliable. For example, already in cold weather iPhone unlocker sometimes does not work, because of slight changes in fingerprint.

A personal USB key

One more way of authentication without password is using USB stick as devise with unique information, that identifies user. In order to use service or software user has to plug in his personal USB and then everything will be activated.
[1] As an example Google can be named. They are providing users with an USB stick, that will intercat with Chrome and load all the personal data - no more typing in passwords. According to their information, communication between the browser and the key generate no information that could be used to impersonate the user if intercepted.
Advantage once again the same - no need to remember anything. Disadvantage - USB stick can be lost, stolen and there will be no possibility to login (like passwords have function ”restore”).

A virtual ’token’

[1] This method is close to the personal USB stick, but here it is an information, personally generated for user. In order to login user has to use his smartphone with special application, that will perform the authentication. Example of implementation is in app Clef (`https://getclef.com/`), that logs users in by displaying a temporarily-generated, unique image on the phone screen. Users simply hold the image up to webcam to authenticate it. Advantage - no need to generate or remember passwords. Also, not like USB stick, the image can’t be stolen, as each one is randomly generated and lasts for less than thirty seconds. Disadvantage - user has to have smartphone with camera, that is not always hold.

Two-factor authentication

The idea of two-factor authentication is to use additional channel for authorizing user, besides simple password. In order to be logged in, user has to type in password and, for example, enter a code, that he receives on his phone or use some additional application to finish identification process. Twitter, Google, LinkedIn and Dropbox, as well as many others now offer the service, as an optional ’extra’ security add-on. For example in Google user provides his phone number and when he logins additional code is sent on his phone. So if somebody will steal the password he still will not be able to login without the phone.
As an advantage, but also disadvantage can be seen the necessity of additional actions. Like for user, who wants to log in, he will need a smartphone or he will have to use some additional service, browser, email, etc. For attackers the same thing will harden possibility of attack a lot. It is not enough now only to break a password - one has also to find a way to go through the second stage. For instance, a recent attack against World of Warcraft involved criminals building a fake replica of the popular add-on site Curse, where every download was laced with malware. [2]

Task 3.2 (theoretical): Reconnaissance in the SecLab

File ”Reconnaissance in the SecLab”
Used tool is Nmap, with command **sudo nmap 10.0.0.0-255 -A**
Found bonus:

```
$ wget -qO - 10.0.0.12:4242
HELO
201 OK
This is a beautiful red-yellow-green-white-black-hat bonbon!
```

Task 3.3 (practical): DNS sniffing

DNS protocol fields required to spoof a response:

1. fields, taken from the DNS request:

   - **id** - 2-bytes identifier. Id fields of DNS request and corresponding DNS response should match
   - **qd**, question section - contains DNS question records, each include 3 fields: *qname*(human-readable name of the server, e.g. "google.com"), *qtype*(type of query, e.g. *A* for host records), *qclass*(class of query, e.g. IN for internet addresses). DNS response should contain the same values of **qd** field, as corresponding DNS request. Also, in **an** (answer section) field of the response, value of *rrname* should be set to the value of *qname* from **qd** field of the request.

2. fields, also required for DNS response spoofing, that are not taken from DNS request:

   - **qr** - 1 bit field, should be set to 1, because value 0 corresponds to DNS request, value 1 to DNS response
   - **aa** - authoritative answer - this bit should be set to 1 in DNS response, and specifies that the responding name server is an authority for the domain name in question section
   - **qdcount** - (2 byte integer) number of DNS question records in question section, no need to copy it from the DNS request, but it will match anyway if we have corresponding **qd** field from the DNS request.
   - **ancount** - (2 byte integer) number of DNS resource records in answer section
   - **type** and **rclass** in answer section should be adjusted according with type and class of data in **rdata** field (which is provided by us)

Note: for DNS response spoofing, also source and destination IPs needed (**src** and **dst** fields of IP header), and source and destination ports (**sport** and **dport** fields from UDP header).
Based on materials from [3].

Task 3.5 (theoretical): Designing Asymmetric Encryption Schemes

Part (a)

This method is very similar to the Diffie-Hellman scheme for key generation. It is not secure against man-in-the-middle attack - in this case for example postman. He can simply add his padlock on Bob's box and send it back to Bob and

make a new box with his padlock and send it to Alice. Bob then will unlock his and send it back - postman has access to the content of the box.
From the point of view of the three questions of security - as there is no Authentication, algorithm can be broken. But from the other side, man-in-the-middle does not get the keys - he just can get access to all the packages, that Bob sends. Also, he can change the content and send it to Alice with his padlock.
For cryptographic means this method will have same problems. Also it is possible, if for example both Bob and Alice will use ROT13 as their method for crypting and the plain text will be in English, after second application, there will be plain text already.

Part (b)

No confidentiality and no integrity can be assured - as man-in-the-middle gets the plaintext on the second step (apply his own key, send back to Bob and get plaintext XORed only with his key). Even more - Alice can get the Bob's key and use it later: simply XOR plaintext with the original message.
Here password can be stolen, as if the same man-in-the-middle will apply plaintext to the Bob's message, he will get his password. That is why different random keys can help. But it will change only that the process has to be performed every time from the beginning, attacker cannot just get plaintext on the first step.

References

[1] What are the alternatives to passwords? *http://www.welivesecurity.com/2015/02/05/alternatives-passwords/* WeLiveSecurity by ESET

[2] Two-factor authentication: What is it ? and why do I need it? *http://www.welivesecurity.com/2015/02/05/alternatives-passwords/* WeLiveSecurity by ESET

[3] Fundamentals of Computer Networking: Project 1 Primer: DNS Overview CS4700/CS5700 Fall 2009, 17 September 2009 *http://www.ccs.neu.edu/home/amislove/teaching/cs4700/fall09/handouts/project1-primer.pdf*