Assignment 4

Abbas Khan , Mariia Rybalka , Linara Adilova

June 6, 2016

Task 4.1 (practical): DNS spoofing

Source code is in file **MyDNSSpoofer.py**.
Response for every group member: (add response) adilova

Task 4.2 (theoretical): Message Authentication Code (MAC)

Abbas

Task 4.3 (practical): Diffie-Hellman

Exploited - man in the middle (details)
Source code is in file **diffie_hellman.py**.
Decrypted communication:
- Done. Hey, can you tell me the password of your Dridex botnet?
- Haha, you are so evil! Ship me 42 euros! :) Just kidding, it is "s00p4doOPas3cReT".
- No no! I am a whitehat ;)... thanks and bye!
- Bla bla... ;) - bye!

Task 4.4 (theoretical): TLS Cipher Suites

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

[1]
In this suite client and server generate pre-master key using Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) and this exchange will be verified using RSA keys. So here RSA is used for Authentication.
Then package exchange will be processed with 128 bit AES in Galois Counter Mode (AES_128_GCM). So this method is used for Encryption.
And for MAC SHA256 is used. It performs both Integrity and Authentication.

TLS_RSA_WITH_RC4_128_MD5

Here pre-master key is generated with RSA - authentication part. Then packages are encrypted with 128 bit Rivest Cipher 4 (RC4_128) - encryption part. And Mac is created with MD5 - integrity and authentication.

Reasons

Three reasons why first suite is considered to be more secure than the second one:

1. First suite provides so called Prefect Forward Secrecy ([2]). When client and server use RSA to generate pre-master key, they use server's RSA and server's RSA public key is a part of server certificate and stored for a long time. So if it will be compromised all the previous connections (if somebody will record them) can be decrypted. While with ECDHE keys are generated for every connection and forgot after connection is closed. So attacker would have to break the encryption for every connection separately.
2. RC4 is proved to be insecure with many different vulnerabilities. One of them is described in [4] - statistical weakness, that alows to distinguish between short outputs of RC4 and random strings by analyzing their second bytes.
3. MD5 hash function compromised by allowing collisions in the output, i.e. there exist another message, not equal to the initial, that will give the same MD5 value. So, the authors of the paper [3] showed several practical ways of doing it.

Task 4.5 (practical): Cipher Suites in the Wild

Script is in file **ssl_suits.sh**. Used list of top sites from Alexa Top 500 (**top_sites.txt**). Checked ciphersuits are from OpenSSL 1.0.2g. Histogram of usage is depicted in Figure 1. Also attached as file **histogram-suits.png** because of bad readability. Script used for drawing is in file **ssl_suites_hyst.py**

References

[1] TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM) *https://tools.ietf.org/html/rfc5289*

[2] Perfect Forward Secrecy *https://www.perfectforwardsecrecy.com/* Sponsored by Encryption Chat

[3] A Study of the MD5 Attacks: Insights and Improvements *http://www.cs.colorado.edu/~jrblack/papers/md5e-full.pdf* J. Black, M.Cochran, T. Highlandy

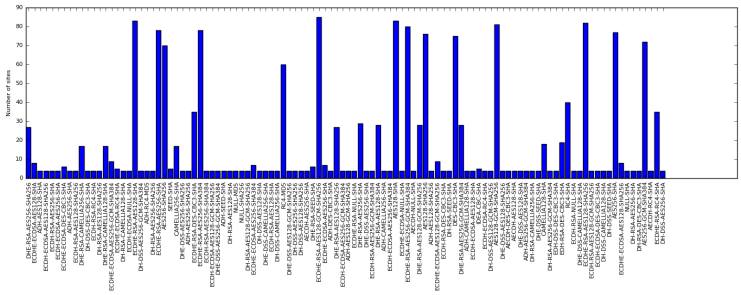

Figure 1: Histogram of usage of TLS cipher suits

[4] A Practical Attack on Broadcast RC4 *http://saluc.engr.uconn.edu/refs/stream_cipher/mantin01attackRC4.pdf* Itsik Mantin and Adi Shamir
Computer Science Department, The Weizmann Institute, Rehovot 76100, Israel