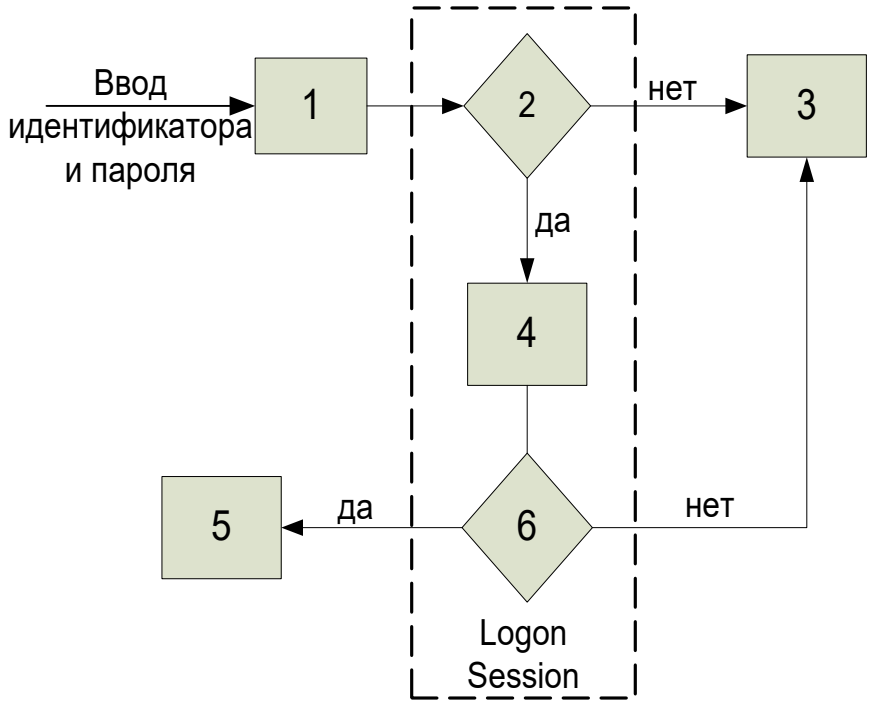
Защита программного обеспечения

Защита ОС Windows

1. Защита паролем



1 – прием си

– проверка идентификатора и пароля на

Windows в базе SAM (Security Account

3 – если ID и пароль не найдены, т

с

4 – устанавливаются

п

6

р

Ес

Е

5 – разрешение в

Рис.1. Процедура идентификации, аутентификации

и установления полномочий пользователя

Эффективность использования пароля

- **A = {a_i}** – алфавит, состоящий из фиксированного набора символов, **i** ∈ **[1, N]**, **N** – мощность алфавита
- **s** - длина пароля **H**; при **H** = 12AAa!!* **s** = 8
- Кол-во комбинаций пароля при фиксир **N** : $I_H = N^s$;

Пример1. **A** = **{a,b,c,d,…,z}** , N=26; при **s** = 8 $N^s = 26^8 = 208827064576$

- Безопасное время использования пароля

$$t_H = ½ (I_H \cdot t), \quad (1)$$

$$t = E/R,\ E = S + S_{sl};$$

Пример2. **N** = 5 симв, S = 6 симв, скорость передачи **R** = 3 [Кбит/с]; принимаем S_{sl}=4 симв, тогда **E**= 6+4=10 симв (либо 80 бит) и

$t_H = ½ (I_H \cdot t) = 1/2(5^6 * 80/(3*1024)) = 203$ с

Пример3. **N** = 26 симв, **S** = 6 симв, скорость передачи **R** = 32 [Кбит/с]; принимаем S_{sl}=14 симв, тогда **E**= 6+14=20 симв (либо 160 бит) и

$t_H = ½ (I_H \cdot t) = 1/2(26^6 * 160/(32*1024)) = 7.5* 10^5$ с = **3.5** ч

Безопасное время использования пароля

Принимаем **P** – это вероятность того, что пароль будет сломан за **M** мес,

P_0 – нижняя граница **P**; $P_0 = n1/n2$; **n1** – число попыток взлома пароля за **M** мес; **n2 –** число всех возможных паролей при определенных **N** и **s;**

n1 = **n11/ n12; n11 –** число символов, которые можно передать по сети за **M** мес, **n12** – число символов, передаваемых в одной попытке;

$$n1 = (R*M*24(ч/д)*60(мин/ч)*30(д/мес))/E, \quad (2)$$

$$n2 = N^s,$$

тогда $$P_0 = (R*M*24*60*30)/(E*N^s) . \quad (3)$$

Так как $P > P_0$, $P > (R*M*24*60*30)/(E*N^s)$ или иначе

$$N^s >= (4.32 * 10^4 *R*M)/(E *P)$$ - ф-ла *Андерсена* (4)

Пример. P = 10^{-3}, M = 3;R = 10 (сим/сек);E = 20 (сим);N = 26 (сим);s = 6 (сим);

$(4.32 * 10^4 * R * M)/(E * P) =$ $(4.32 * 10^4 * 10 * 3)/(20 * 10^{-3}) = 3.9 * 10^9$;

$\mathbf{N^s} = 26^6 \approx 3.089 \cdot 10^8 \leq 3.9 \cdot 10^9$.

Это означает, что при выбранном размере алфавита и длине пароля, необходимое условие неравенства не выполняется.

При s = 7 (сим):

$26^7 \approx 8.03 \cdot 10^9 \geq 3.9 \cdot 10^9$. *Выполнение условия означает, что для выбранного алфавита, с вероятностью P = 10^{-3} пароль с длиной 7 символов не будет сломан за 3 месяца.*

Протокол Kerberos

- **Назначение** - для пересылки зашифрованного сообщения (A → B) по открытым каналам на платформе ОС **Windows** при взаимод с **T;**
- **Опирается** на протокол **Нидхэма-Шрёдера (R. Needham-M. Schröder)** и базируется на симметричном шифровании данных

Протокол Нидхэма-Шрёдера

Обозначения: **A, B, T** – имена участников, E_A - ключ, общий для **A и T**, E_B – ключ, общий для **B и T**

1. **A** → **T**: **A, B,** R_A ; R_A – случ число, сгенерир-е **A**
2. **T** генерир-т случ сеансовый ключ **K;** затем щифрует :

C = $E_A(R_A,B,K; E_B(K,A))$; **T: C** → **A**

3. **A** извлекает из **C: K** и убеждается, что R_A равно R_A для 1-го этапа;

извлекает $E_B(K,A) = C_3$; **A:** C_3 → **B**

4. **B,** используя E_B, извлекает **K** из C_3; **B** генерирует случ число R_B, создает шифртекст $C_4 = K(R_B)$ и **B**: C_4 → **A**
5. **A** расш-т C_4 ключом **K,** создает шифртекст $C_5 = K(R_B - 1)$; **A:** C_5 **B**
6. **B** расш-т C_5 ключом **K** и убеждается, что **известное ему** R_B **уменьшено на 1;** ***Т. о. создан секретный сеансовый ключ K для A и B***

- Установленная в сети TCP/IP служба Kerberos, является доверенной стороной (**T**)
- Основой Kerberos является БД Клиентов и их секретных ключей
- Сетевые службы, которые требуют аутентификацию, должны зарегистрировать в Kerberos свои секретные ключи
- Так как Kerberos знает все секретные ключи, он может убеждать одни объекты в подлинности других. Керберос создает сеансовые ключи, которые выдаются Клиенту и Серверу, и никому больше
- Для шифрования используется алгоритм DES
- Для организации канала связи Клиент запрашивает у Kerberos разрешение на обращение к службе организации таких сообщений, эта служба называется **Ticket Grating Service** (TGS) — служба выделения мандата

Протокол Kerberos

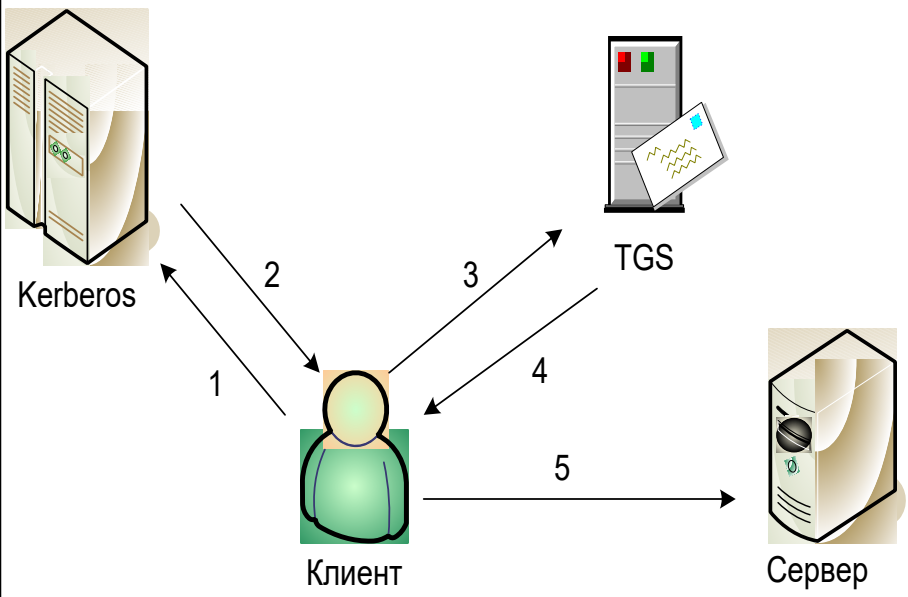


Рис.2. Общая схема взаимодействия компонент в протоколе Kerberos

1 — Клиент запрашивает Керберос разрешение на обращение к службе TGS.
2 — После анализа предоставленных документов о возможности организации сообщения между Кл и Серв Керберос выдает Кл-ту соответствующее разрешен
3 — Пользуясь разрешением службы Керберос, Кл запрашивает TGS о выделен ему мандата на организацию канала между Клиентом и Сервером.
4 — Получение такого мандата.
5 — Клиент пересылает соответствующее сообщение серверу

C— Клиент (Client),

S – Сервер (Server),

A — Сетевой адрес Клиента (Address) — имя Клиента,

v — Временная метка, содержащая начальное и конечное время действия мандата,

t — просто метка времени, соответствующая периоду времени, в течение которого действует сеансовый ключ,

Kx — секретный ключ объекта **X**,

Kx,y — сеансовый ключ для организации сеанса между **X** и **Y**,

{m}K_x — сообщение **m**, зашифрованное ключом **Kx**,

Tx,y — мандат, выданный **X** на использование **Y**,

Ax,y — аутентификатор, выданный **X** для **Y**, то есть информация, с помощью которой **Y** аутентифицирует **X**.

Операции (стрелки 1-5 на рис.2) могут быть записаны в формализованном виде:

1 — Клиент-Kerberos: **C, TGS**

2 — Kerberos-Клиенту: **{Kc, TGS}K_c; {Tc, TGS}K_{TGS}**

3 — Клиент-TGS:**{Ac,s}K_c, $_{TGS}$;**

{Tc,TGS}K_{TGS}

4 — TGS-Клиенту: **{$K_{c,s}$}$K_{c,TGS}$;**

{Tc,s}K_s

5 — Клиент-Серверу: **{Ac,s}$K_{c,\ s}$;**

{Tc,s}K_s

Kerberos использует 2 типа удостоверений:

- Мандаты (для безопасной передачи Серверу данных о личности Клиента):

$$T_{c,s} = S, \{C, A, v, K_{c,s}\}K_s$$

Клиент не может расшифровать мандат, поскольку он не знает секретный ключ Ks, но он может предъявить его Се-ру, как док-во его аутентичности, те. прочитать либо изменить мандат ни Клиент, ни кто-либо иной не может.

- Аутентификаторы (это дополнительная информация, предъявляемая вместе с мандатом):

$$A_{c,s} = \{C, t, \text{Ключ}\}\ K_{c,s}$$

Клиент создает аутентификатор на каждый сеанс, Ключ - является просто ключом и необязательным дополнительным элементом сеанса и все эти данные шифруются общим ключом, известным Клиенту и Серверу: $K_{c,s}$. В отличие от мандата, аутентификатор используется только один раз