ЛИТЕРАТУРА

1. *Schlegel R., Zhang K., Zhou X., et al.* Soundcomber: a stealthy and context-aware sound Trojan for smartphones // Proc. 18th Annual Network and Distributed System Security Symposium (NDSS '11), San Diego, CA, February 6–9, 2011. P. 17–33.

УДК 004.94

ИСПОЛЬЗОВАНИЕ ЭЛЕКТРОННЫХ СЕРТИФИКАТОВ ДЛЯ АВТОРИЗАЦИИ ПО ДОВЕРЕННОСТИ В ОС LINUX

В. И. Рыжков

Предлагается решение для делегирования некоторого набора прав от одного пользователя (доверителя) операционной системы другому (доверенному лицу) на определённый промежуток времени. Для этого предложено использовать «доверенности» — объекты, содержащие в себе такие поля, как идентификатор доверителя, идентификатор доверенного лица, время действия доверенности, а также набор прав, делегируемых доверенному лицу доверителем. Доверенность должна содержать также цифровую подпись на закрытом ключе доверителя под всеми вышеперечисленными полями. Предложенное решение реализовано для операционной системы Linux с помощью криптографического инструмента OpenSSL и подключаемых модулей аутентификации (PAM). В качестве доверенностей здесь выступают цифровые сертификаты стандарта X.509 v3, а делегируемые полномочия указываются по определённому формату в поле «Расширения» этих сертификатов. Сам функционал авторизации по доверенности реализован в виде модуля PAM.

Ключевые слова: *электронные сертификаты, X.509, Linux, PAM, OpenSSL.*

В операционных системах привилегии пользователя можно задавать, используя группы, в которые он входит.

Пусть некоторый пользователь (доверитель) хочет передать некоторые свои права другому пользователю (доверенному лицу), который изначально этими правами не обладает. Такая схема полезна в случае, когда доверителю приходится отсутствовать по той или иной причине и он хочет передать свои полномочия своему доверенному лицу. Самое очевидное решение: доверитель может добавить доверенное лицо в некоторую группу, которая обладает этими правами. При таком подходе возникают следующие проблемы:

1) Право переводить пользователя из группы в группу есть, как правило, далеко не у каждого.
2) Пусть право переводить пользователей из группы в группу у доверителя есть. Допустим, доверитель будет отсутствовать в течение месяца, но эти права необходимо делегировать на неделю. Следующие три недели доверенное лицо будет находиться в привилегированной группе, не имея в этом потребности.

Таким образом, возникает задача построения системы делегирования некоторого набора прав доступа, которыми обладает некий пользователь-доверитель (не обязательно «привилегированный» в системе), другому пользователю — доверенному лицу, который ими изначально может не обладать, на некоторый (определённый пользователем-доверителем) промежуток времени.

При этом, очевидно, должны выполняться следующие требования:

1) любой пользователь системы может быть доверителем для любого другого пользователя;
2) пользователь может делегировать только права, принадлежащие ему, и никакие другие;
3) только доверенное лицо может авторизоваться на делегируемые ему права;
4) доверитель может определять промежуток времени, в течение которого доверенное лицо наделяется делегируемыми ему правами.

Дадим формальное описание решения, удовлетворяющее перечисленным требованиям.

Пусть

— $U = \{u_1, u_2, \ldots, u_n\}$ — множество пользователей, и каждый пользователь $u_i \in U$ имеет пару (x_{u_i}, y_{u_i}) — закрытый/открытый ключ. Очевидно, должно выполняться условие $x_{u_i} \neq x_{u_j}$, $y_{u_i} \neq y_{u_j}$ для всех $i \neq j$;
— $G = \{g_1, g_2, \ldots, g_k\}$ — множество групп;
— $P = \{p_1, \ldots, p_m\}$ — множество всех возможных прав доступа к объектам;
— $T = \{0, 1, 2, \ldots\}$ — множество целых неотрицательных чисел (время);
— $PA : G \to 2^P$ — функция, задающая соответствие прав доступа для конкретной группы;
— $UA : U \to 2^G$ — функция, задающая множество групп, на которые может быть авторизован пользователь;
— $PROXY$ — множество доверенностей, объектов вида $proxy_{u_i}^{u_j}(g, t_{\text{start}}, t_{\text{end}}) = (B, Sig_{x_{u_i}}(B))$, где $B = (u_i, u_j, g, t_{\text{start}}, t_{\text{end}})$; $u_i, u_j \in U$; $g \subseteq UA(u_i)$; $t_{\text{start}}, t_{\text{end}} \in T$; $Sig_{x_{u_i}}(B)$ — цифровая подпись B на закрытом ключе пользователя u_i. Другими словами, это сертификат, который подтверждает факт делегирования на промежуток времени $[t_{\text{start}}, t_{\text{end}}]$ некоторого набора групп g от пользователя u_i, который изначально им обладает, пользователю u_j, который им изначально может и не обладать.

Определим условия корректности доверенности $proxy_{u_i}^{u_j}(g, t_{\text{start}}, t_{\text{end}}) \in PROXY$ для пользователя $u_k \in U$, предъявляющего данную доверенность, в момент времени t_{current}:

1) $t_{\text{current}} \in [t_{\text{start}}, t_{\text{end}}]$ — условие актуальности доверенности;
2) $g \subseteq UA(u_i)$ — условие обладания доверителя делегируемыми правами;
3) $k = j$ — условие предъявления доверенности доверенным лицом;
4) $V(proxy_{u_i}^{u_j}(g, t_{\text{start}}, t_{\text{end}})) = \texttt{true}$ — условие корректности подписи.

Здесь $V : PROXY \to \{\texttt{true}, \texttt{false}\}$ — функция проверки цифровой подписи.

Доверенность, удовлетворяющую условиям 1–4, далее будем называть корректной, в противном случае — некорректной.

Используем следующее обозначение: пусть A — некоторое непустое множество, тогда $A^{\varnothing} = A \cup \{\varnothing\}$.

Определим функцию $assign : U \times PROXY^{\varnothing} \times T \to (2^G)^{\varnothing}$, которая осуществляет авторизацию на делегированные группы при предъявлении корректной доверенности:

$$assign(u_j, z, t_{\text{current}}) = \begin{cases} \varnothing, \text{ если } z = \varnothing \text{ или } z \text{ некорректная;} \\ g, \text{ если } z = proxy_{u_i}^{u_j}(g, t_{\text{start}}, t_{\text{end}}) \text{ и } z \text{ корректная.} \end{cases}$$

Функция $groups : U \times PROXY^{\varnothing} \times T \to (2^G)$ осуществляет авторизацию пользователей в системе. Определим её следующим образом:

$$groups(u_j, z, t_{\text{current}}) = UA(u_j) \cup assign(u_j, z, t_{\text{current}}).$$

Таким образом, если пользователь авторизуется без доверенности или авторизуется с некорректной доверенностью, то он получает ровно те права, которые дают ему группы, в которых он изначально состоит (посредством функции UA). Однако при предъявлении корректной доверенности он может авторизоваться на некий набор групп, которым он изначально не обладал, но который делегировал ему пользователь-доверитель. Требования 1–4 выполняются благодаря использованию доверенностей, в которых явно указаны доверенное лицо, доверитель, определён срок действия, и все эти поля подписаны на закрытом ключе доверителя.

В реализации данного решения для операционной системы Linux в качестве доверенностей используются сертификаты X.509 версии 3 [1]. Сертификаты данного стандарта содержат следующие ключевые поля:

— имя эмитента (кто выдал сертификат);
— имя субъекта (кому выдан сертификат);
— период действия;
— расширения;
— подпись сертификата (с указанием алгоритма хэширования и подписи).

Поле «Расширения» представляет собой набор троек $(OID, criticalityFlag, Value)$, где OID (Object Identifier) используется для именования расширения; $criticalityFlag$ — флаг критичности; $Value$ — значение расширения. Расширения предоставляют возможность внедрения в сертификат произвольной информации до его создания.

Таким образом, сертификаты стандарта X.509 v3 могут использоваться в качестве доверенностей. Для этого в поле «Имя эмитента» необходимо указать имя пользователя-доверителя, в поле «Имя субъекта» — имя доверенного лица, в поле «Расширения» — набор делегируемых прав в системе. Доверителю необходимо также указать период действия доверенности в поле «Период действия» и подписать сертификат на своём закрытом ключе.

Создание доверенностей осуществляется при помощи криптографического инструмента OpenSSL [2]. Функция *assign*, авторизующая пользователя на делегированные группы (при предъявлении корректной доверенности), реализована в виде модуля PAM [3] — элемента ядра Linux.

ЛИТЕРАТУРА

1. `https://www.ietf.org/rfc/rfc5280.txt` — RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.
2. `http://www.openssl.org/` — OpenSSL: The Open Source toolkit for SSL/TLS.
3. `http://www.linux-pam.org/` — A Linux-PAM page.

УДК 004.94

ФОРМИРОВАНИЕ ВЕКТОРОВ ПОКАЗАТЕЛЕЙ ДЛЯ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ ПРИ ОБНАРУЖЕНИИ АТАК НА WEB-ПРИЛОЖЕНИЯ

С. Н. Сорокин

Представлен подход к оценке качества и выбора наиболее подходящих показателей для обучения нейронных сетей при решении задач обнаружения атак на web-приложения, предложена методика формирования векторов показателей для классов атак, позволяющая уменьшить количество нейронных сетей, используемых для обнаружения различных атакующих воздействий.