DOI: 10.18372/2410-7840.19.12218
УДК 004.056.5

ПРОБЛЕМИ ВИКОРИСТАННЯ SSL/TLS

Тетяна Бабенко, Сергій Толюпа, Вікторія Гречко

Одним із засобів створення захищеного сеансу зв'язку є використання криптографічного протоколу SSL / TLS. Однак він не гарантує повну захищеність і має свої уразливості і недоліки, які повинні бути проаналізовані і усунені в подальшому. У цій роботі проаналізовано базова термінологія, наведені аспекти, за якими стає можливою атака типу «людина посередині», її варіації, проблема підміни сертифікатів і самоподпісанного сертифікатів, також недоліки аутентифікації, уразливості прикладних бібліотек, проблема обміну ключами, зокрема досліджена вразливість Блейхенбахера, також згадано про проблеми інфраструктури відкритих ключів, проблема інтероперабельності в Україні та недавні уразливості даного протоколу (SWEET32, DROWN, ROBOT). Результатом дослідження є сформований перелік невирішених проблем і рекомендацій щодо підвищення рівня криптостійкості протоколу.

Ключові слова: *захищений сеанс зв'язку, SSL/TLS, інфраструктура відкритих ключів, сертифікати X.509, вразливості, MITM, обмін ключами, SWEET32, DROWN, ROBOT, прикладні бібліотеки SSL/TLS.*

Актуальність. Бурхливе зростання мережі Internet разом з істотним набором нових можливостей і послуг приносить і ряд нових проблем, найбільш неприємною з яких, безумовно є проблема безпеки.

Саме для вирішення цієї проблеми було розроблено розширення протоколу HTTP (HyperText Transfer Protocol) – HTTPS (HyperText Transfer Protocol Secure), відмінною рисою якого є транспортні криптографічні протоколи SSL (Secure Sockets Layer) та TLS (Transport Layer Security). Протокол SSL забезпечує захищений обмін даними за рахунок поєднання трьох наступних елементів: автентифікація (відбувається тільки при установці сеансу і до відправки першого блоку даних), конфіденційність (забезпечується симетричними алгоритмами шифрування з розміром ключа від 40 до 128 біт), цілісність (забезпечується за допомогою хеш-функцій; алгоритми SHA (Secure Hash Algorithm) або MD5[1]).

Обидва протоколи використовують так званий сертифікат відкритого ключа стандарту X.509 [2], який не цілком коректно прийнято називати зараз «SSL сертифікатом». Безпека SSL-з'єднань в основному залежить від коректної перевірки сертифіката відкритого ключа, представленого під час встановлення з'єднання. Повний алгоритм перевірки сертифікатів X.509 можна знайти в документах RFC 5280 та RFC 2818.

Метою статті є комплексний аналіз сучасних вразливостей протоколу SSL/TLS, зокрема реалізації можливих атак, дослідження і узагальнення переліку невирішених проблем.

Основна частина дослідження. За час свого існування сертифікат встиг продемонструвати свої сильні і слабкі сторони. По-перше, найважливіша та найслабша ланка будь-яких схем безпеки — люди, тож надійність, безпека та інші достоїнства використання сертифікатів X. 509 визначаються конкретними людьми. Треба зазначити, що вагомою причиною багатьох вразливостей є нерозуміння розробниками численних опцій, параметрів та повернених значень функцій бібліотек SSL через складну архітектуру прикладного програмного інтерфейсу для основних бібліотек SSL [3].

По-друге, хоча SSL був розроблений з оглядом на забезпечення безпеки від атак типу «людина посередині» (MITM), атака буде успішною, якщо клієнт не перевіряє чи видано сертифікат дійсним органом сертифікації [4], його статус [5], термін дії, та ін.

Знаходячись на кордоні мережі організації зловмисник може підмінити оригінальний сертифікат своїм. Таким метод може використовуватися як для контролю за інформацією, що передається, так і з метою викрадення особистих даних, що передаються за допомогою захищеного з'єднання HTTPS. Крім того, якщо оригінальний сертифікат не буде переданий користувачеві, а файервол може бути налаштований на прийом і наступну підміну самопідписаних або відкликаних сертифікатів, тим самим зникає впевненість у тому, що обмін даними залишається конфіденційним. В основі цієї проблеми лежить один з недоліків SSL-сертифікатів: немає механізму, що гарантує швидке відкликання того чи іншого сертифіката[5]. По ряду

причин далеко не всі ОС і програмні продукти завжди використовують останні, найбільш актуальні списки довірених кореневих серверів сертифікатів та в цілому останні версії програмного забезпечення. В цілях забезпечення безпеки від атак даного типу, рекомендується повністю заборонити роботу з веб-серверами, чиї сертифікати містять будь-які помилки.

Іншим прикладом атаки MITM є атака при встановленні TCP з'єднання (handshake). «Рукостискання» починається тоді, коли клієнт підключається до SSL-сервера. Зловмисник може вплинути на обмін повідомленнями так, щоб сторони обрали різні алгоритми шифрування, а не ті, що вони вибирають зазвичай. Через те, що багато реалізацій підтримують експортоване шифрування, а деякі навіть 0-шифрування або MAC-алгоритм, ці атаки представляють великий інтерес.

По-третє, однією з проблем забезпечення кібернетичної безпеки інформації, що існує при обміні зашифрованими даними є проблема надійності ключів сеансу обміну. Як відомо, обмін ключами є обов'язковим і відбувається до обміну даними. При цьому обмін сеансовими ключами може відбутися, як з цільовою обчислювальною системою, так і зі зловмисником. З багатьох історичних і комерційних причин найчастіше в TLS використовується обмін ключами по алгоритму RSA. Даний алгоритм має один недолік: ця ж пара відритого і закритого ключів використовується і для автентифікації сервера. Відповідно, якщо зловмисник отримує доступ до закритого ключа сервера, він може розшифрувати весь сеанс зв'язку (можливі атаки відклику, відкритого тексту, атака за словником, BEAST, однак в таких випадках SSL відбиває атаки за допомогою особливого унікального ідентифікатора з'єднання MAC). Більш того, зловмисник може здійснювати атаки на криптографічні алгоритми (RC4 [6], атака Барда і Воденея [7], стійкими до описаних типів атак є версії протоколу TLS 1.2).

Одним із прикладів сучасної атаки на криптографічні алгоритми є SWEET32[13]. Власне, атака на шифри 3DES і Blowfish. Криптографічні протоколи TLS, SSH, IPsec і OpenVPN використовують алгоритми блокового шифрування (AES, Triple-DES, Blowfish). Таким чином забезпечується шифрування переданих між клієнтом і сервером даних. Коротка довжина блоку робить шифр уразливим до «атак дня народження». Дослідники заявляють про те, що подібні атаки зустрічаються для 64-бітних шифрів протоколів TLS і OpenVPN. Подібні алгоритми шифрування використовуються величезною кількістю ресурсів в інтернеті. SWEET32 - це атака пошуку колізій в режимі зчеплення блоків з використанням зворотного зв'язку CBC (Cipher block chaining). На практиці це дозволяє розшифровувати HTTPS-з'єднання за умови, що токен аутентифікації передається в кожному запиті. Завдяки тому, що можливо передбачити зміст заголовків повідомлень (або наявності можливості їх контролю) зловмисник може згенерувати велику кількість запитів з деякою кількістю заздалегідь передбачуваних даних у відповідях і, в результаті, спробувати розшифрувати потрібні сесії і дізнатися токен.

Детальніше атаку Блейхенбахера. За алгоритмом RSA перед установкою зашифрованого зв'язку, клієнт випадковим чином вибирає ключ сесії, що шифрується згодом відкритим ключем і відправляється на сервер. Сервер розшифровує це «повідомлення», зберігає копію ключа сесії, а після використовує для ідентифікації клієнта. Таким чином клієнт проходить валідацію, і встановлюється захищене HTTPS-з'єднання. Основна проблема була виявлена у використанні паддінгу PKCS # 1 1.5 при шифруванні ключів сесій за допомогою RSA. Виявилося, що зловмисник може просто направляти TLS-серверу випадкові ключі, запитуючи, чи правильні вони. Таким чином можна підібрати справжній ключ.

Обмін ключами Діффі-Хеллмана є більш захищеним, оскільки встановлений симетричний ключ ніколи не залишає клієнта або сервера і, відповідно, не може бути перехоплений зловмисником, навіть якщо той знає закритий ключ сервера[8]. На цьому заснований принцип зниження ризику компрометації минулих сеансів зв'язку: для кожного нового сеансу зв'язку створюється новий, так званий «тимчасовий» ключ. Відповідно, навіть в найгіршому випадку (якщо зловмисникові відомий закритий ключ сервера), зловмисник може лише отримати ключі від майбутніх сесій, але не розшифрувати записані раніше **сесії**.

Однак навіть враховуючи вищевказані вразливості, від використання алгоритму RSA ще не відмовились, а старі вразливості набули нових форм - атака Блейхенбахера була реалізована й іншими способами, наприклад, атака DROWN та ROBOT [14,15].

DROWN[12] - Decrypting RSA using Obsolete and Weakened eNcryption, дозволяє дешифрувати TLS-трафік клієнта, якщо на серверній стороні не відключена підтримка протоколу SSLv2 у всіх серверах, що оперують одним і тим же приватним ключем. Загальний варіант атаки експлуатує уразливість в експортних шифри SSLv2, що використовують 40-бітові ключі RSA. Зловмисникові необхідно пасивно прослуховувати сотні TLS-з'єднань жертви і відправляти спеціальним чином сформовані пакети на сервер з SSLv2, що використовує такий же ключ. Дослідникам вдалося відновити TLS-сесію клієнта протягом 8 годин, використовуючи 200 машин Amazon EC2: 150 типу g2.2xlarge с nVidia GPU, і 50 g2.8xlarge з 4 nVidia GPU.

ROBOT (Return Of Bleichenbacher's Oracle Threat). Основна проблема в тому, що більшість провайдерів серверного обладнання не можуть реалізувати розділ 7.4.7.1 стандарту TLS (RFC 5246), в якому й описується протидія атаці Блейхенбахера. У зловмисників з'явився спосіб розшифровувати дані, публікувати і підписувати інформацію приватними ключами легітимного сайту. І, хоча ROBOT базується на проблемах, описаних ще 19 років тому, перед нею залишаються вразливими дуже популярні веб-сайти, зокрема Facebook і PayPal (27 із 100 найпопулярніших доменів за версією Alexa).

По-четверте, SSL має ряд вразливостей в реалізації і конфігурації протокола та прикладних бібліотек. В більшості випадків за допомогою SSL-проксі контролюються такі потенційні загрози безпеці, як: застарілі версії протоколу SSL, криптонестійкі хеш-функції в підписах сертифікатів, крипто-нестійкі алгоритми шифрування[9], однак атака на переузгодження TLS-сеансу принципово невиявна з боку клієнта (повторне підтвердження дозволяє серверу створити новий секретний ключ на вже існуючому SSL-з'єднанні)[10].

На практиці більшість додатків залежать від каркасів передачі даних для встановлення HTTPS-з'єднань, які у загальному випадку використовують внутрішні бібліотеки SSL неявно для додатків. Наприклад, Apache HttpClient, клієнтська бібліотека HTTP(S) на базі JDK, повинна виконувати перевірку власного імені хосту, що призводить до численних вразливостей програмного забезпечення на основі старих версій HttpClient, які не підтверджують імена хостів. Крім того, Apache HttpClient використовує структуру даних HttpHost для опису HTTP(S) з'єднань. HttpHost не має перевірки внутрішньої послідовності, що дозволяє, наприклад, при з'єднанні з портом 443 використовувати HTTP, замість HTTPS[3].

По-п'яте, загальні проблеми інфраструктури відкритих ключів. Фундаментальною проблемою в даному випадку є те, що, якщо будь-який значимий компонент публічної мережі довіри, заснованої на загальновизнаних головних уповноважених центрах, скомпрометований – це призводить до **компрометації** всієї системи. При цьому, щоб скомпрометувати систему, не обов'язково втручання зловмисника в функціонування уповноваженого центру. Досить, щоб слабкою ланкою в ланцюжку довіри виявився, наприклад, криптографічно-нестійкий хеш-алгоритм (MD5, [11]).

В Україні також існує проблема наявності та функціонування репрезентативних центрів сертифікації. Нерідко різні використовувані додатки та інформаційні сервіси державного рівня вимагають застосування різних алгоритмів шифрування, а отже і різних пар ключів від різних центрів сертифікації. Відсутність єдиного надійного центру сертифікації породжує проблему інтероперабельності – ключі одного центру нерідко не підтримуються іншими центрами.

Висновок. Незважаючи на те, що технологія HTTPS існує майже стільки ж, скільки протокол HTTP, який вона доповнює, певні класи проблем її реалізації не вирішені до сих пір. А саме:

1. Проблема наявності репрезентативних центрів сертифікації та управління сертифікатами, що реалізовано без урахування реальних потреб користувачів (включаючи зручність користування).

2. Наявність помилок в реалізації і/або конфігурації протоколу SSL/TLS і його прикладних бібліотек.

3. Сучасні стандарти ІВК не підтримуються в належному обсязі.

4. Можливість підміни або фальсифікації сертифікатів, що призводить до компрометації всієї ІВК.

Підсумовуючи вищесказане, можна висунути наступні рекомендації:

– Веб-сервери та VPN повинні бути налаштовані на перевагу 128-бітних шифрів; Згідно з даними дослідників, близько 1,1% відсотка з 100 тисяч найпопулярніших сайтів каталогу Alexa і 0,5%

з мільйона найпопулярніших підтримують AES, але вважають за краще використовувати 3DES;

– TLS-бібліотеки і додатки повинні обмежувати довжину TLS-сесій для 64-бітних шифрів. Зробити це можна за допомогою механізму повторної установки з'єднання;

– Користувачі OpenVPN можуть змінити використовуваний шифр за замовчуванням Blowfish на AES. Якщо зробити цього не можна, то необхідно примусово ініціювати повторний випуск ключів;

– Рекомендовано встановлювати особливі правила для розриву з'єднання з клієнтом, який виконує операцію повторного підтвердження більше установленої за замовчуванням кількості раз на секунду;

– На державному рівні вирішити проблему інтероперабельності;

– В цілях забезпечення безпеки від атак типу MITM, рекомендується повністю заборонити роботу з веб-серверами, чиї сертифікати містять будь-які помилки.

ЛІТЕРАТУРА

[1]. Stephen Thomas, "SSL&TSL Essentials, securing the Web", *Wiley Computer publishing*, 2000.

[2]. Cooper, "Standards Track, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL)", *RFC 5280*, 2008.

[3]. M. Georgiev, "The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software", Proceedings of the 2012 ACM conference on Computer and communications security, 2012.

[4]. J. Sunshine, S. Egelman, "Crying Wolf: An Empirical Study of SSL Warning Efectiveness", *SSYM'09 Proceedings of the 18th conference on USENIX security symposium*, 2009.

[5]. S. Santesson, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP", *RFC 6960*, 2013.

[6]. A. Klein, "Attacks on the RC4 stream cipher", *Designs, codes and cryptography*, 2008.

[7]. С. Леонтьєв, В. Попов, С. Смишляев, "Противодействие атакам на протокол TLS", *Системи високої доступності*, 2012.

[8]. I. Grigorik, "High Performance Browser Networking", *O Reilly Media*, 2013.

[9]. A. Sotirov, M. Stevens, "MD5 considered harmful today: Creating a rogue CA certificate", *International Journal of Applied Cryptography*, 2009.

[10]. T. Zoller, G-Sec, *TLS/SSLv3 renegotiation vulnerability explained*, University of Luxembourg, 2011.

[11]. Ah. Kioon, M. Cindy, Z. Wang, Deb. Das. S., "Analysis of MD5 Algorithm in Password Storage", *Applied Mechanics and Materials Security*, 2013.

[12]. N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, "DROWN: Breaking TLS using SSLv2", *USENIX Security Symposium*, 2016.

[13]. K. Bhargavan, G. Leurent, "On the Practical (In-) Security of 64-bit Block Ciphers Collision Attacks on HTTP over TLS and OpenVPN", *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[14]. T. Jager, J. Schwenk, J. Somorovsky, "On the Security of TLS 1.3 and QUIC Against Weaknesses in PKCS#1 v1.5 Encryption", *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[15]. H. Böck, J. Somorovsky, C. Young, "Return Of Bleichenbacher's Oracle Threat (ROBOT)", *Cryptology ePrint Archive: Report 2017/1189*, 2017.

REFERENCES

[1]. Stephen Thomas, "SSL&TSL Essentials, securing the Web", *Wiley Computer publishing*, 2000.

[2]. Cooper, "Standards Track, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL)", *RFC 5280*, 2008.

[3]. M. Georgiev, "The Most Dangerous Code in the World: Validating SSL Certificates in Non-Browser Software", Proceedings of the 2012 ACM conference on Computer and communications security, 2012.

[4]. J. Sunshine, S. Egelman, "Crying Wolf: An Empirical Study of SSL Warning Efectiveness", *SSYM'09 Proceedings of the 18th conference on USENIX security symposium*, 2009.

[5]. S. Santesson, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP", *RFC 6960*, 2013.

[6]. A. Klein, "Attacks on the RC4 stream cipher", *Designs, codes and cryptography*, 2008.

[7]. S. Leontiev, V. Popov, S. Smyshlyaev, " Countering attacks on the TLS protocol", *High Availability Systems*, 2012.

[8]. I. Grigorik, "High Performance Browser Networking", *O Reilly Media*, 2013.

[9]. A. Sotirov, M. Stevens, "MD5 considered harmful today: Creating a rogue CA certificate", *International Journal of Applied Cryptography*, 2009.

[10]. T. Zoller, G-Sec, *TLS/SSLv3 renegotiation vulnerability explained*, University of Luxembourg, 2011.

[11]. Ah. Kioon, M. Cindy, Z. Wang, Deb. Das. S., "Analysis of MD5 Algorithm in Password Storage", *Applied Mechanics and Materials Security*, 2013.

[12]. N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, "DROWN: Breaking TLS using SSLv2", *USENIX Security Symposium*, 2016.

[13]. K. Bhargavan, G. Leurent, "On the Practical (In-) Security of 64-bit Block Ciphers Collision Attacks on HTTP over TLS and OpenVPN", *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[14]. T. Jager, J. Schwenk, J. Somorovsky, "On the Security of TLS 1.3 and QUIC Against Weaknesses in PKCS#1 v1.5 Encryption", *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[15]. H. Böck, J. Somorovsky, C. Young, "Return Of Bleichenbacher's Oracle Threat (ROBOT)", *Cryptology ePrint Archive: Report 2017/1189*, 2017.

ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ SSL/TLS

Одним из средств создания защищенного сеанса связи является использование криптографического протокола SSL / TLS. Однако он не гарантирует полную защищенность и имеет свои уязвимости и недостатки, которые должны быть проанализированы и устранены в дальнейшем. В этой работе проанализирована базовая терминология, приведены аспекты, по которым становится возможной атака типа «человек посередине», ее вариации, проблема подмены сертификатов и самоподписанных сертификатов, также недостатки аутентификации, уязвимости прикладных библиотек, проблема обмена ключами, в частности исследована уязвимость Блейхенбахера, также упомянуто о проблемы инфраструктуры открытых ключей, проблема интероперабельности в Украине и недавние уязвимости данного протокола (SWEET32, DROWN, ROBOT). Результатом исследования является сформированный перечень нерешенных проблем и рекомендаций по повышению уровня криптостойкости протокола.

Ключевые слова: защищенный сеанс связи, криптографический протокол, SSL/TLS, инфраструктура открытых ключей, сертификаты X.509, уязвимости, MITM, обмен ключами, SWEET32, DROWN, ROBOT, прикладные библиотеки SSL/TLS.

USAGE ISSUES OF SSL/TLS ISSUES

One of the means of creating a secure communication session is using the SSL/TLS cryptographic protocol, however it does not guarantee full protection and also has its own vulnerabilities and disadvantages, which must be analyzed and eliminated in the future. In particular, in this paper the basic terminology is analyzed, vulnerabilities of the protocol are analyzed and generalized, some aspects that make possible implementation of the "man in the middle" attack and it's variations,the problem of certificates substitution and self-signed certificates, authentication defects, application libraries vulnerabilities, key exchange problem, including the Bleichenbacher's threat, public key infrastructure problems, the problem of interoperability in Ukraine and the most recent vulnerabilities of this protocol are presented (SWEET32, DROWN, ROBOT). The result of the research is the arranged list of unsolved problems and recommendations to increase cryptoresistability level of the protocol.

Keywords: secure communication session, SSL/TLS cryptographic protocol, public key infrastructure, X.509 certificates, vulnerability, MITM attack, key exchange, SWEET32, DROWN, ROBOT, application libraries.

Бабенко Тетяна Василівна, доктор технічних наук, професор кафедри кібербезпеки та захисту інформації факультету інформаційних технологій КНУ імені Тараса Шевченка.
E-mail: babenkot@ua.fm

Бабенко Татьяна Васильевна, доктор технических наук, профессор кафедры кибербезопасности и защиты информации факультета информационных технологий КНУ имени Тараса Шевченко.

Babenko Tatyana, full professor, Professor of the Cybersecurity and Information Security Department of the Information Technology Faculty of Taras Shevchenko National University of Kyiv.

Толюпа Сергій Васильович, доктор технічних наук, професор кафедри кібербезпеки та захисту інформації факультету інформаційних технологій КНУ імені Тараса Шевченка.
E-mail: tolupa@i.ua

Толюпа Сергей Васильевич, доктор технических наук, профессор кафедры кибербезопасности и защиты информации факультета информационных технологий КНУ имени Тараса Шевченко.

Tolupa Sergiy, full professor, Professor of the Cybersecurity and Information SecurityDepartment of the Information Technology Faculty of Taras Shevchenko National University of Kyiv.

Гречко Вікторія Володимирівна, студентка кафедри кібербезпеки та захисту інформації факультету інформаційних технологій КНУ імені Тараса Шевченка.
E-mail: grechko.viktoria@gmail.com

Гречко Виктория Владимировна, студентка кафедры кафедры кибербезопасности и защиты информации факультета информационных технологий КНУ имени Тараса Шевченко.

Grechko Victoria, student of the Cybersecurity and Information Security Department of the Information Technology Faculty of Taras Shevchenko National University of Kyiv.