THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 18, Special Issue 2017, pp. 403-412

ON DHCP SECURITY

Dumitru Daniel DINU, Mihai TOGAN, Ion BICA

Military Technical Academy, Computer Science Dept., Bucharest, Romania
E-mail: mihai.togan@gmail.com

Abstract. Despite the security issues it has, DHCP is one of the most used protocols because it facilitates the automatic allocation of the configuration information in a network. While the number of mobile devices connected to Internet is increasing exponentially, the widespread adoption of IPv6 will take a lot of time. In this context, the need for good security mechanisms that prevents the known attacks against DHCP raises. In this paper we present an overview of the previous solutions to secure the protocol and, in the same time, we identify the reasons why each of these attempts failed. Based on the previous work missteps, we define a set of requirements for a practical and efficient authentication module for DHCP. Then we introduce a simple and flexible module that allows authentication of DHCP messages using two different trust models: PKI and PGP. We implemented and evaluated the proposed authentication module using different key types and sizes in the two trust models. The comprehensive results show that the proposed authentication module does not affect the protocol operation, but provides the so necessary security that DHCP lacks.

Key words: DHCP, authentication module, authentication option, trust model, digital signature, replay detection, PKI, PGP.

1. INTRODUCTION

DHCP is one of the most used network protocols thanks to automatic distribution of network configuration parameters, although it has many security vulnerabilities. The security of the protocol was not considered as an important criteria when designed and the simplicity and ease of use prevailed [1, 2]. Once the number of computers connected to TCP/IP networks increased, the DHCP weaknesses begun to show. Nowadays, there are billions of devices connected to Internet and the number is expected to grow exponential with the Internet of Things wave [3]. The root cause of the DHCP security issues is the lack of an authentication mechanism [4, 5]. Thus a malicious DHCP client can get unauthorized access into a network. He can use the network services without being allowed to or he can create a DHCP starvation attack to exhaust the available IP addresses in the network [6]. An attacker can also configure a rogue DHCP server in the network to provide wrong configuration information to clients. Using the rogue server, the attacker can create DoS, MiTM or phishing attacks [7, 8, 9]. Although the new IPv6 stack will solve almost all the security issues of the old IPv4, the transition to the new protocols will take long. Despite new devices are manufactured with IPv6 support, there are millions of older devices able to communicate using only IPv4. For these reasons, a security mechanism for DHCP is absolutely necessary.

Our work is motivated by the lack of an authentication mechanism for DHCP messages on one hand, and by the long transition process from IPv4 to IPv6 on the other hand. Our contribution is fourfold. Firstly, we analyze the DHCP security issues and the attempts to secure the protocol. Then we show why each of the previous attempts failed to be adopted in practice. Secondly, we formulate a set of design principles for a secure, flexible and practical DHCP authentication module. We introduce DHCPAuth, an authentication module for DHCP, designed in accordance with the formulated principles, to be secure, flexible, and easy to integrate in existing DHCP implementations. Thirdly, the proposed authentication module is evaluated with respect to packet length and processing time in two different trust models, using different key pairs and sizes.

The results can be used by network administrator to select the best trust model, key sizes and types for their needs. Fourthly, we provide the evaluated implementation of DHCPAuth for further analysis and testing.

2. DHCP SECURITY

Since the first attacks against DHCP until now a lot of solutions to secure DHCP were proposed. In this section we briefly describe the most important ones. For each solution, we identify the strengths and weaknesses and we explain why it failed to be widely used. The DHCP authentication option, RFC 3118, defines two authentication methods based on a shared secret between the client and server [10]. The possibility of authenticating DHCP messages using RSA signatures was considered computationally infeasible. The main issue of the proposal is given by the complicated key management for a large number of clients. In [11] is described a way to encode DHCP options larger than 255 bytes using the concept of aggregate buffer.

The authentication method based on certificates for DHCP [12] uses the RFC 3118 authentication option format [10], but with digital signatures. Although the MTU limitation of 1500 bytes per packet is acknowledged, the paper proposes to split larger DHCP messages in several packets. It also introduces additional communication between DHCP entities and a trusted server. The certificate based authentication solution [13] tries to solve the problem using the same authentication option format [10] and digital signatures, but it does not provide a concrete way to transmit the public key certificate to verifying entity. Other approaches to secure DHCP using digital certificates are described in [14, 15], but they also avoid to give a concrete solution for certificate transmission.

The unified approach to intra-domain security [16] describes a framework for securing network protocols within a single domain. It introduces new DHCP options and additional communication between DHCP entities and an authentication server. It also assumes that DHCP messages can be split and sent in several packets. The Challenge-Handshake Authentication Protocol (CHAP) [17] uses an Authentication, Authorization and Accounting (AAA) server, namely a RADIUS server, to secure DHCP. It introduces additional communication between the DHCP entities and the authentication server. A similar approach to secure the protocol using a Kerberos server is described in [18], but it also introduces additional communication. The DHCP authentication with an effective key management [19] assumes that a different session key is used to encrypt each DHCP message. It follows the authentication option structure defined in RFC 3118 [10], but fails to provide a reliable way for computing the session key. Even more complex and tight coupled mechanisms are proposed in [20, 21, 22]. For example, the secure DHCP system with user authentication uses no more than 8 different units to secure the protocol by introducing additional message flows between the units.

Some of the proposed solutions to secure DHCP failed because they didn’t consider the real constraints of the protocol [13, 14, 15, 19, 20, 21], while the others were too complex and required many changes in the protocol operation [12, 16, 17, 18, 22]. We note that a detailed and careful analysis of the proposed solutions to secure DHCP can be found in [23], while in [23, 24] are introduced for the first time the ideas that led to the proposed solution described next.

3. PROPOSED SOLUTION

Based on the strengths and weaknesses of previous attempts to secure DHCP, in this section we formulate a set of design principles for a practical DHCP authentication module that is easy to integrate in existing DHCP implementations. Then we describe the proposed DHCP authentication module, DHCPAuth.

3.1. Design principles

Previous failed attempts to secure DHCP showed that before designing a solution for authenticating the DHCP messages, the limitations and constraints imposed by the protocol should be clearly understood. Because a DHCP message can’t be split into several packets, the main limitation that should be considered is that the maximum DHCP packet size is 1500 bytes, which is equal with the MTU in an Ethernet network. If we subtract the IP packet header of 20 bytes and UDP packet header of 8 bytes, we get 1472 bytes which can be used for the DHCP message, including the options. Considering that DHCP message header has 236 bytes, then it results 1236 bytes available for DHCP options. In the best case, if the *file* and *sname* fields are used for DHCP message options through DHCP overloading option [11], there are 1428 bytes available for all DHCP message options. Thus the transmission of a digital certificate or a long public key in a single DHCP message is out of question. Another constraint considered in the design process is to limit and reduce as much as possible the dependency and communication between DHCP entities and other entities. The DHCP authentication module must be easy to integrate in existing DHCP implementations and it must not modify the DHCP client state machine. It is desirable to use the authentication option format defined in RFC 3118 [10] and to avoid introducing new DHCP messages or options. The module must be able to authenticate the received DHCP messages and to prevent replay attacks. A DHCP entity without the authentication module must be able process the messages received from a DHCP entity with the authentication module, but a DHCP entity with the authentication module should not accept DHCP messages which are not authenticated if configured so. Although these design principles may seem very restrictive, they were formulated to guarantee that a module which complies with them is practical and it is likely to be deployed in real world.

3.2. DCHP message authentication module

DHCPAuth, the DHCP message authentication module proposed to secure DHCP communication by authenticating the messages sent between DHCP servers and clients, is designed to work as a firewall for DHCP entities and follows the previously formulated design principles. Each DHCP message sent by a DHCP entity passes through the DHCPAuth module before going into the network. Depending on the configuration of the module, it may add the authentication option for the message and then send it through the network or it may send it directly through the network. When receiving a DHCP message, the DHCPAuth module processes the received message before the DHCP engine. If the configuration of the module requires authentication of the received message, DHCPAuth verifies the received message authentication option. If the authentication option is valid, the received message is passed to the DHCP engine, else it is discarded. If no authentication is required, the received DHCP message is passed directly to the DHCP engine. The module is able to create and verify authentication options for DHCP messages using public key signatures in two trust models: PKI [25, 26] and PGP [27, 28, 29]. The supported trust models provide different levels of security and flexibility. The PKI model is more rigid than the web of trust model, but in the same time provides an increased level of security compared with PGP trust model. On the other hand the web of trust model allows for much more flexibility at the cost of a lower security level [30, 31]. The key aspect of the web of trust model is that each verifier can choose how to adjust the security level depending on his own needs [32, 33]. We recommend using PKI authentication for messages sent be DHCP servers and PGP authentication for messages sent by DHCP clients.

The authentication option added by the DHCPAuth module to the sent DHCP messages follows, with minor changes, the format defined in [10]. To be able to add a digital signature to the authentication option, the *Authentication Information* field is split into *Signature Length* and *Message Signature* fields. The *Signature Length* field allows the identification of the correct signature bytes in the *Message Signature* field. The other fields of the authentication option stay the same, but new values are introduced to correctly identify the new signature types. Namely, the *Protocol* field will have value 2 for PKI authentication and value 3 for PGP authentication. The *Algorithm* field will have value 2 if SHA-1 message digest is used for the digital signature and value 3 if SHA-2 message digest is used for the digital signature. The *RDM* field keeps the same value as in the original specifications and thus the *Replay Detection* field must contain strictly increasing values to avoid replay attacks. The strictly increasing value is given by the time of the machine on which the DHCPAuth module is running on. The time value is expressed in seconds and microseconds of the remaining second from the start of Unix time.

3.3. Sent message

If the DHCPAuth module is enabled for a DHCP entity, then each DHCP message passes through the module before being sent through the network. If the module is configured to authenticate the sent message, then it adds the authentication option to the message to be sent. The authentication option fields are filled with the correct values and then the DHCP message signature is computed. The signature is computed over the whole DHCP message including the newly added authentication option. The *hops* and *giaddr* fields of the DHCP message are set to zero to allow the validation of the signature even if these fields are modified by a relay agent. The *Signature Length* and *Message Signature* fields of the authentication option are also set to zero because the signature length and the signature are not known prior to signature creation. These field values will be filled in after the signature computation. The *Signature Length* field allows the verifier to extract the signature byte stream from the *Message Signature* field of the received DHCP message authentication option. If the DHCPAuth module is not configured to authenticate the sent message, the module just sends the message through the network without modification. The DHCPAuth module operation is controlled through a configuration file. The DHCP user can configure which of the sent DHCP messages must be authenticated, how they must be authenticated and what private key must be used for the signature creation. The module supports authentication in two trust models: PKI and PGP.

3.4. Received message

All DHCP messages received by a DHCP client or server with the DHCPAuth module enabled are processed first by the module. If the module is configured to authenticate the received message, then it first checks if the authentication option is present. If the authentication option is not present in the received DHCP message, the message is dropped. If the authentication option is present, the module verifies the *Replay Detection* field value. To be able to detect replay attacks, the DHCPAuth module maintains a list of last received *Replay Detection* field values for each transaction identifier *xid*. If the received *Replay Detection* field value is greater than the stored value, the stored value is updated and the authentication option verification continues. If the received value of the *Replay Detection* field is less than the stored value, then the replay detection check fails and the received message is dropped. If there is no entry in the list for the received message *xid*, the module considers the received message as valid and a new entry is added in the list. If the replay detection check is passed, the module verifies the digital signature of the received message. The signature stream of *Signature Length* bytes is extracted from the *Message Signature* field and verified with the received DHCP message. For the verification of the signature, the *hops*, *giaddr*, *Signature Length* and *Message Signature* are set to zero as in the case of the signature creation. If the signature of the received DHCP message is valid, the DHCP message is passed to the DHCP engine to be processed further, else the DHCP message is dropped. The DHCP user can configure the DHCPAuth module to authenticate or not the received DHCP messages. The verifier has to be able to acquire the public key or certificate of the sender to be able to verify the received message signature. For the experiments we conducted, the public keys and certificates required for the signature verification were available in the local store of the verifier. We recommend that the DHCP clients download the DHCP server certificate before connecting to it from a trusted third party share server. The DHCP server can acquire the public keys of the DHCP clients from a public key server by updating the local key ring at least once a day.

3.5. Security considerations

The proposed DHCP authentication module prevents all known possible vulnerabilities of DHCP entities. Thanks to authentication option, an illegitimate client can't get access into a network without being authorized to. The DHCP starvation attack launched by a malicious client is not effective, because the server first checks the authentication option before allocating the configuration information. There is the possibility that a legitimate DHCP client to send a large amount of DHCP requests using different MAC addresses and thus to exhaust the available IP addresses. This insider treat can be countered by limiting the number of requests signed with a given private key in a time frame or changing the trust of a private key if a certain threshold is reached. Attacks like phishing and MITM will fail because the DHCP client is able to authenticate the configuration information sent by server. Replay attacks are not possible because of the *Replay Detection* of the authentication option. If someone tries to change this field value, he will have to resign the DHCP message or to find a collision in the hash function, which is almost impossible. On the other hand, the possibility of a successful DoS attack increases because the signature verification operation is time consuming. The only solution to protect against this attack is to limit the number of requests processed in a time frame and speed the cryptographic operations using hardware accelerators. Hence, the module prevent all the major DHCP security issues. More, if someone wants to exploit a DHCP entity, he has first to bypass the authentication module by compromising the secret keys of the DHCP users. Thus if simple security rules are followed when manipulating and storing the secret keys, the DHCP entities are safe.

3.6. Implementation

We provide a patch with DHCPAuth module implementation [34] for ISC DHCP software [35]. The ISC implementation of DHCP is used in all major Linux distributions. The implementation done in C introduces minimal changes in the DHCP source code to integrate the authentication module. It uses the open source cryptographic library OpenSSL [36] for PKI authentication and the GPGME library for PGP authentication. The GPGME library runs over GnuPG implementation [37] of OpenPGP standard. Together with the DHCPAuth patch we provide all the scripts necessary to reproduce the experiments we conducted.

4. EVALUATION

We have evaluated the proposed authentication module using different signature types to understand the impact of the authentication module on the protocol operation. The evaluation was performed on two virtual machines running Ubuntu with one core clocked at 2.1 GHz and 1 GB of RAM each. We measured the time to generate and verify the signatures, the overhead introduced by the authentication option and the worst case processing time for the DHCPREQUEST and DHCPACK messages on both client and server. We chose to evaluate these DHCP messages given the high occurrence of the two messages between clients and servers in a network that uses DHCP for configuration information allocation.

4.1. Processing time

First we report the result of the evaluation using different RSA key pairs (Fig. 1 and Fig. 2), then we give results for different DSA key pairs (Fig. 3 and Fig. 4). The processing time is an important metric since if the processing of a DHCP message takes too much, the DHCP client may enter in a new state until the response is received. Thus it is important that the processing of the authentication option to be fast enough to avoid endless loops. We chose to measure the time spent on processing the authentication options as well as the total time to process the DHCP message in order to be able to clearly evaluate the impact of the authentication option processing. For the client we measured the time to process the sent DHCPREQUEST message and the received DHCPACK message separately, while for the server we measured the time to process the received DHCPREQUEST and create the DHCPACK message sent to client.

The time required for creating the DHCPREQUEST message on the client is strongly influenced by the time spent on creating the authentication option as can be seen in Fig. 1 and Fig. 3. The most time consuming operation is the signature creation. We also note that the processing time increases as the RSA or DSA key size increases. Even though almost all the time spent on creating the DHCPREQUEST is given by the signature creation, the introduced overhead is acceptable and is not influencing the communication. For example, the time required to create DHCPREQUEST is around 20 milliseconds when using a RSA key of 2048 bits and around 25 milliseconds when using a DSA key of the same size. For a RSA key size of 4096 bits, the time spent on creating the DHCPREQUEST is around 75 milliseconds.

The time required by server to process the received DHCPREQUEST and create the DHCPACK message is given in Fig. 2 for RSA key pairs and in Fig. 4 for DSA key pairs. If we look at the processing time of DHCPREQUEST authentication option and DHCPACK authentication option for RSA keys (Fig. 2), we can see the verification of the authentication option for the received message takes more than the creation of the authentication option for the sent message for key sizes less or equal with 2400 bits. This means that for key sizes less than 2400 bits, the PGP signature verification is slower than the PKI signature creation. For RSA key sizes greater than 2400 bits, the DHCPREQUEST authentication option processing takes less time than the DHCPACK authentication option creation. For DSA keys, the time required to process the DHCPREQUEST message is always higher than the time required to create the DHCPACK message. The total time required to process the received message and send the response to the client is around 20 milliseconds for 2048 bits keys, whether RSA or DSA keys are used.

The time spent by the client to verify the authentication option of the received message is negligible compared with the time to process the DHCPACK message for RSA keys (Fig. 1) and DSA keys (Fig. 3). By analysing Fig. 1 and Fig. 2 (or Fig. 3 and Fig. 4), one can notice that the authentication option creation on client takes considerable more time than authentication option creation on server. This difference is explained by the different trust models used to create the signatures. The PGP signature creation on client takes more time than the PKI signature creation on server. The verification process of the authentication option takes always more time for the PGP signatures than for the PKI signatures. The difference between the time required for PGP and PKI to create and verify the signatures is given by the additional fields included in the PGP signatures and the interaction with the key chains from the local store.

The creation of the authentication option is around 25% faster for 2048 bits RSA key than for 2048 bits DSA key when using PGP trust model. When using the PKI trust model the signature creation for 2048 bits DSA keys is around 50% faster than the signature creation for the 2048 bits RSA key. On the other hand, the verification of the authentication option that contains a DSA signature generated with a 2048 bits key in the PKI trust model takes 6.70 milliseconds, while the verification of the authentication option that contains a RSA signature generated with a 2048 bits key is 0.57 milliseconds. The time required to verify the authentication option generated with a 2048 bits DSA key in PGP takes 16.47 milliseconds, while for the same RSA key size in the same trust model it takes 11.52 milliseconds.

The key conclusion of the processing time evaluation is that the DHCP operation is not affected by the trust model, key type or key size. Consequently the user is free to choose what trust model to use with what key types and sizes without worrying about performance. The only concern should be selecting a trust model that fits the concrete needs of the user.

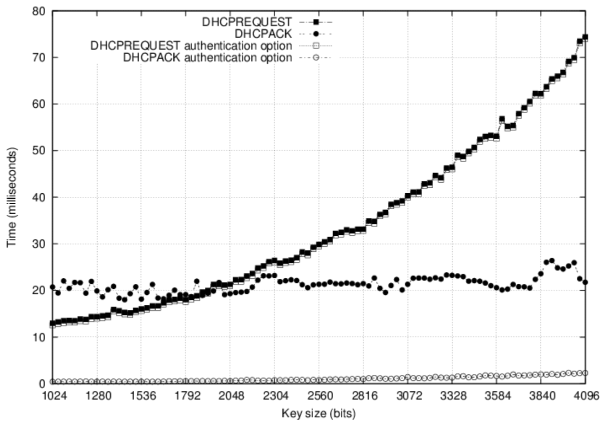

Fig. 1 – Client processing time for RSA signatures.

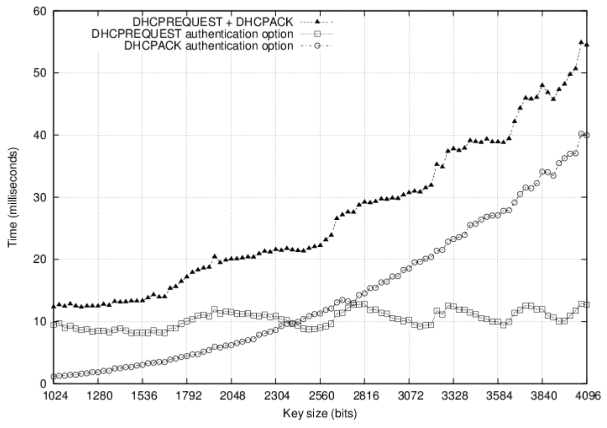

Fig. 2 – Server processing time for RSA signatures.

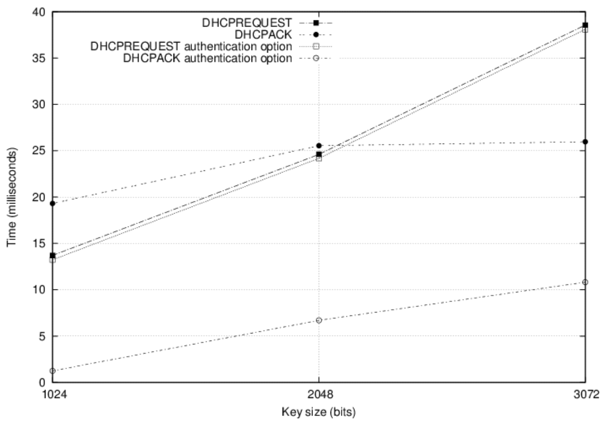

Fig. 3 – Client processing time for DSA signatures.

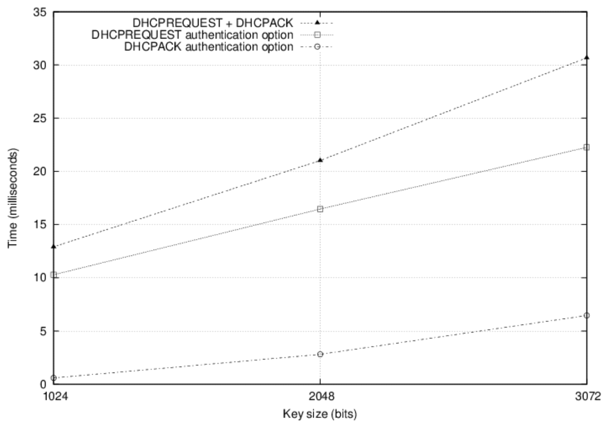

Fig. 4 – Server processing time for DSA signatures.

4.2. DHCP packet length

To evaluate the impact of the authentication option on DHCP messages we measured the length of the DHCP packets sent between the client and server in the key renewing phase. Fig. 5 gives the length of the DHCPREQUEST authentication option, the length of the DHCPREQUEST packet sent by client as well as the length of the DHCPACK authentication option and the length of the DHCPACK packet sent by server when using different RSA keys. Similarly, Fig. 6 shows the length of the DHCPREQUEST and DHCPACK packets and the length of the authentication option for the two DHCP messages when using DSA keys. We choose to use the length of the DHCPREQUEST and DHCPACK packets instead of the length of the DHCPREQUEST and DHCPACK messages, to clearly emphasize that the authentication option does not force the DHCP message to use all available bytes of an IP packet.

For a RSA key size of 2048 bits, the DHCPREQUEST authentication option length is 300 bytes, while when using a DSA signature it is 109 bytes. The DHCPACK authentication option for the same key size is 269 bytes for RSA signatures and 85 bytes for DSA signatures. Considering the overhead introduced in the DHCP messages, the DSA signatures are about 3 times lighter than the RSA signatures. However, the total length of the DHCP packets even when using RSA signatures with keys of 4096 bits is less than 900 bytes. The conclusion of the evaluation is that the authentication option does not introduce an overhead that makes impossible the transmission of a DHCP message in a single packet. Moreover, the packet overhead is far below the 1500 bytes MTU limit. This leaves enough space for other DHCP message options.

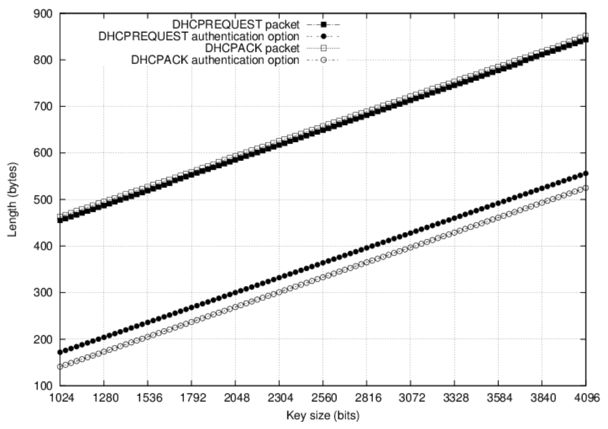

Fig. 5 – Packet length overhead for RSA signatures.

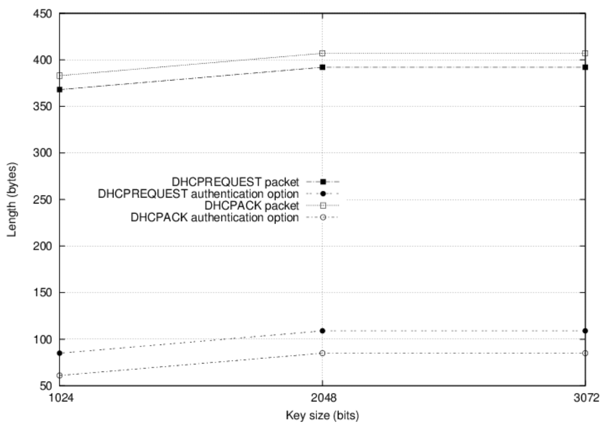

Fig. 6 – Packet length overhead for DSA signatures.

4.3. Worst case processing time

The time spent to verify the authentication option of a received DHCP message depends on the key type and size as was shown in previous subsections. It may also depend on the number of certificates available in the certificate store of the verifier or the number of public keys available in the verifier key ring. The worst case processing time evaluation is conducted to determine how the number of trusted communicating entities influences the processing time of the received DHCP messages. We measure the time required by the DHCP verifier to identify the correct public key and verify the received message authentication option as well as the total time to process the received DHCP message. For the public key certificates of the server, we ordered the certificates in the client store such that the correct certificate to be checked last. We could not apply a similar approach for the PGP public keys because the correct key is identified in the public key ring by the associated identifier. The number of PKI certificates/PGP public keys in Fig. 7, Fig. 8, Fig. 9, and Fig. 10 indicates the number of incorrect PKI certificates/PGP public keys tried before the correct key pair is used. For these experiments we used RSA and DSA keys of 2048 bits.

The time spent by the DHCP server to verify the received DHCPREQUEST is not considerably influenced by the number of RSA keys (Fig. 8) or DSA keys (Fig. 10) available in the public key ring thanks to the fast identification of the correct key using the key identifier embedded in the PGP signature. However, the total processing time for RSA signature verification increases from 11.84 milliseconds when there is just one public key in the verifier key ring to 13.58 milliseconds when there are 100 public keys in the verifier key ring. For DSA signatures the required time varies between 16.49 milliseconds to 18.91 milliseconds. We can conclude that identifying the right public key of the signer in the worst case takes 1.74 milliseconds more for 100 RSA public keys and 2.42 milliseconds more for 100 DSA public keys.

The time required to process the received DHCPACK message and the time required to verify the received authentication option is not directly proportional with the number of available certificates in the store of the verifier. In fact the time variations that can be observed in Fig. 7 and Fig. 9 are given by the time required to identify the correct public key in the certificate store. Although the time variation with the number of available keys to check is greather in the client case than in the server case, the maximum values are around 45 milliseconds for RSA signatures and 300 milliseconds for DSA signatures. These peak values are small enough and thus they do not affect the client state machine. The much higher peak values for DSA keys than for RSA keys are a consequence of the higher time required to verify a DSA signature created with a 2048 bits key than a RSA signature created with the same key size.

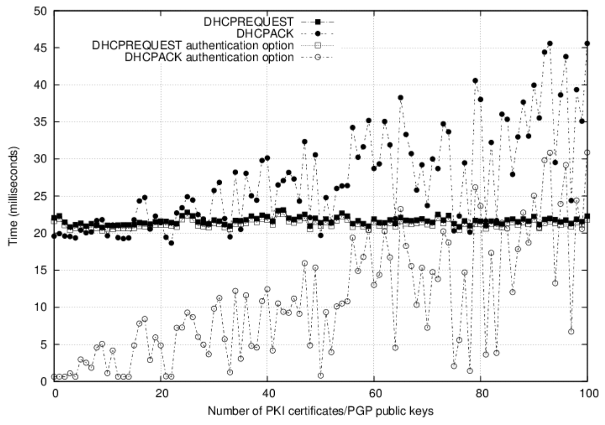

Fig. 7 – Client worst case processing time for RSA keys.

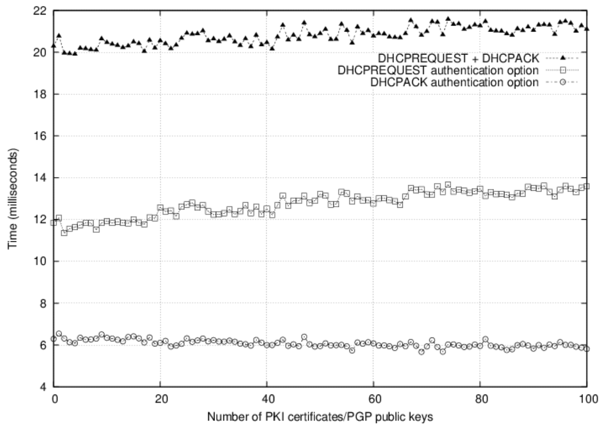

Fig. 8 – Server worst case processing time for RSA keys.

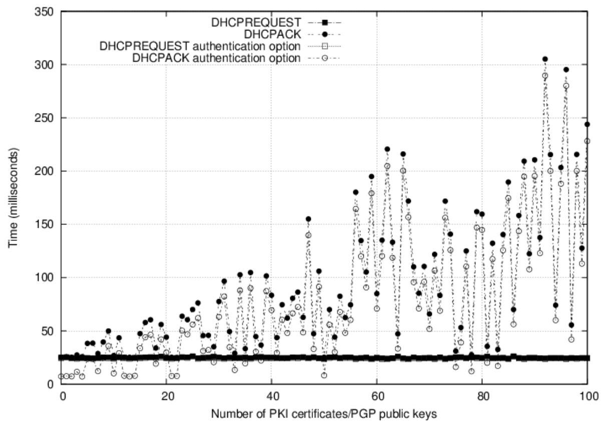

Fig. 9 – Client worst case processing time for DSA keys.

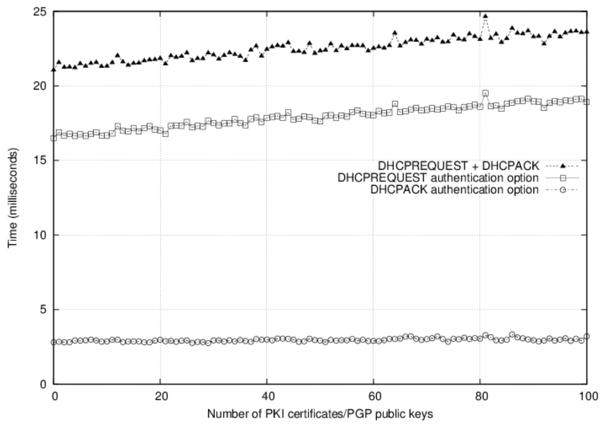

Fig. 10 – Server worst case processing time for DSA keys.

5. CONCLUSION

In this paper we described the DHCP security issues and then we analyzed the previous attempts to secure the protocol. Using the knowledge gained from the previous failed attempts, we formulated a set of design principles for a secure and flexible authentication module for DHCP. Considering the real limitations and constraints of DHCP, we proposed a practical authentication module, DHCPAuth, to secure the protocol using two different trust models: PKI and PGP. Then we presented the result of the proposed module evaluation in different use cases. Firstly, the processing time evaluation showed that the overhead introduced by the processing done in the authentication module is insignificant with respect to the total processing time. Secondly, we demonstrated that the size of authenticated DHCP packets is below the MTU limit and thus does not requires splitting DHCP messages into several packets. Thirdly, the worst case processing time measurements showed that the processing time of the authentication option is not influenced in a dangerous way by the number of available trusted keys in the certificate store. The detailed results can help network administrators and DHCP clients in selecting the trust model, key sizes and types to use. To the best of our knowledge, this is the first practical solution to secure DHCP using two different trust models: PKI and PGP. We provided the source code of the proposed module together with the evaluation scripts and detailed evaluation results for different use cases to prove the practicality of the solution.

REFERENCES

1. R. DROMS, *Dynamic Host Configuration Protocol*, IETF, RFC 2131, March 1997.
2. S. ALEXANDER, R. DROMS, *DHCP Options and BOOTP Vendor Extensions*, IETF, RFC 2132, March 1997.
3. INTERNET LIVE STATS, *Internet users in the world*, [Online]. Available: http://www.internetlivestats.com/.
4. THE TCP/IP GUIDE, *DHCP Security Issues*, [Online]. Available: http://www.tcpipguide.com/free/t_DHCPSecurityIssues.htm.
5. B. GROZA, M. MINEA, M. CRISTEA, P.S. MURVAY, M. IACOB, *Protocol Vulnerabilities in Practice: Causes, Modeling and Automatic Detection*, Proceedings of the Romanian Academy, Series A, **13**, pp. 167-174.
6. THINK SECURITY, *DHCP starvation – quick and dirty*, October 2010, [Online]. Available: http://think-security.com/dhcp-starvation-quick-and-dirty/.
7. M. TULLOCH, *DHCP Server Security (Part 1)*, July 2004, [Online]. Available: http://www.windowsecurity.com/articles-tutorials/misc_network_security/DHCP-Security-Part1.html.
8. S. JONES, *DHCP man in the middle attack*, September 2002, [Online], Available: http://seclists.org/vuln-dev/2002/Sep/99.
9. F. LINDNER, *Exploiting DORA – Attacks on the DHCP protocol*, [Online]. Available: https://www.nruns.com/_downloads/Whitepaper-exploiting-D.O.R.A-Lindner.pdf.
10. R. DROMS, W. ARBAUGH, *Authentication for DHCP Messages*, IETF, RFC 3118, June 2001.
11. T. LEMON, S. CHESHIRE, *Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)*, IETF, RFC 3396, November 2002.
12. Y. XU, S. MANNING, M. WONG, *An Authentication Method based on Certificate for DHCP*, DHC Internet Draft, September 2011.
13. G. GLAZER, C. HUSSEY, R. SHEA, *Certificate-Based Authentication for DHCP*, March 2003.
14. J. DEMERJIAN, A. SERHROUCHNI, *DHCP Authentication Using Certificates*, Security and Protection in Information Processing Systems, IFIP Advances in Information and Communication Technology, **147**, pp. 457-472, Springer, 2004.
15. S. DUANGPHASUK, S. KUNGPISDAN, S. HANKLA, *Design and Implementation of Improved Security Protocols for DHCP Using Digital Certificates*, ICON, Singapore, December 2011.
16. C.A. SHUE, A.J. KALAFUT, M. GUPTA, *A Unified Approach to Intra-Domain Security*, IEEE International Symposium on Secure Computing (SecureCom), August 2009.
17. K. DeGRAAF, J. LIDDY, P. RAISON, J.C. SCANO, S. WADHWA, *Dynamic Host Configuration Protocol (DHCP) Authentication using Challenge Handshake Authentication Protocol (CHAP) Challenge*, United States Patent Application Publication, 2011.
18. K. HORNSTEIN, T. LEMON, B. ADOBA, J. TROSTLE, *DHCP Authentication Via Kerberos V*, IETF DHC Working Group, October 2001.
19. H. JU, J.W. HAN, *DHCP Message Authentication with an Effective Key Management*, World Academy of Science, Engineering and Technology, August 2005.
20. T. AURA, M. ROE, S.J. MURDOCH, *Securing Network Location Awareness with Authenticated DHCP*, 3rd International Conference on Security and Privacy in Communication Networks (SecureComm), Nice, France, September 2007.
21. N. SHANKAR, W.A. ARBAUGH, K. ZHANG, *A Transparent Key Management Scheme for Wireless LANs Using DHCP*, HP Laboratories, Palo Alto, September 2001.
22. T. KOMORI, T. SAITO, *The Secure DHCP System with User Authentication*, Proceedings of the 27th Annual IEEE Conference on Local Computer Networks, November 2002.
23. D.D. DINU, M. TOGAN, *DHCP Server Authentication using Digital Certificates*, The 10th International Conference on COMMUNICATIONS (COMM 2014), Bucharest, May 2014.
24. D.D. DINU, M. TOGAN, *DHCPAuth – A DHCP Message Authentication Module*, 10th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2015), pp. 405-410, Timișoara, May 2015.
25. P. GUTMANN, *PKI: It's not dead, just resting*, IEEE Computer, **35**, 8, pp. 41-49, 2002.
26. B. HEIN, *The Threat Landscape of PKI: System and Cryptographic Security of X.509, Algorithms, and Their Implementations*, Proceedings of the Romanian Academy, Series A, **14**, pp. 286-294.
27. D. ATKINS, W. STALLINGS, P. ZIMMERMANN, *PGP Message Exchange Formats*, IETF, RFC 1991, August 1996.
28. J. CALLAS, L. DONNERHACKE, H. FINNEY, R. THAYER, *OpenPGP Message Format*, IETF, RFC 2440, November 1998.
29. J. CALLAS, L. DONNERHACKE, H. FINNEY, D. SHAW, R. THAYER, *OpenPGP Message Format*, IETF, RFC 4880, November 2007.
30. C. ELLISON, B. SCHNEIER, *Top 10 PKI risks*. *Computer Security Journal*, **XVI**, 1, 2000.
31. SECURE SHARE, *15 reasons not to start using PGP*. [Online]. Available: http://secushare.org/PGP
32. A. JØSANG, *An Algebra for Assessing Trust in Certification Chains*, Proceedings of the NDSS'99, Network and Distributed Systems Security Symposium, 1999.

33. J. JONCZY, M. WUTHRICH, R. HAENNI, *A Probabilistic Trust Model for GnuPG*, 23rd Chaos Communication Congress, Berlin, 2006.
34. DHCPAuth, *A DHCP Message Authentication Module*. [Online]. Available: https://github.com/daniel-dinu/DHCPAuth.
35. INTERNET SYSTEM CONSORTIUM, *ISC DHCP: Enterprise Grade Solutions for Configuration Needs*. [Online]. Available: https://www.isc.org/downloads/dhcp/.
36. OPENSSL, *Cryptography and SSL/TLS Toolkit*, [Online]. Available: https://www.openssl.org/.
37. GNUPG, *The GNU Privacy Guard*, [Online]. Available: https://www.gnupg.org/.