ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Security Issues in High Performance Transfer Protocol-UDT

[I]Nidhi Gupta, [II]Shanu Garg
[I]M.Tech, CSE Dept., Manav Rachna International University, Faridabad, Haryana, India
[II]Assistant Professor, CSE Dept, Manav Rachna International University, Faridabad, Haryana, India

Abstract
The present paper is to study the various security issues in Transfer Protocol-UDT. Next-generation e-Science applications will require the ability to transfer information at high data rates between distributed computing centers and data repositories. The development of next generation protocols, such as UDT (UDP-based data transfer), promptly addresses various infrastructure requirements for transmitting data in high speed networks. However, this development creates new vulnerabilities when these protocols are designed to solely rely on existing security solutions of existing protocols such as TCP and UDP. It is clear that not all security protocols (such as TLS) can be used to protect UDT, just as security solutions devised for wired networks cannot be used to protect the unwired ones. The proposed work is all about to study the security issues and different attacks happens in such protocol and to analysis the performance for the network. The work can be simulated using NS2 or MATLAB software's for the quantitative approach.

Keywords
TCP, GSS-API, High Speed Bandwidth, UDT, HIP, CGA, SASL, DTLS, AO, TLC etc

I. Introduction

UDT introduces a new three-layer protocol architecture that is composed of a connection flow multiplexer, enhanced congestion control, and resource management. The new design allows protocol to be shared by parallel connections and to be used by future connections. It improves congestion control and reduces connection set-up time. Recent developments in network research introduced an enhanced version of UDT, considered to be one of the next generations of high performance data transfer protocols.
UDT is a connection-oriented duplex protocol [17], which supports data streaming and partial reliable messaging. It also uses rate-based congestion control (rate control) and window-based flow control to regulate outgoing traffic. This was designed such that rate control updates the packet sending period every constant interval, whereas flow control updates the flow window size each time an acknowledgment packet is received. It was expanded to satisfy more requirements for both network research and applications development. This expansion is called Compassable UDT and is designed to complement the kernel space network stacks. However, this feature is intended for:

1. Implementation and deployment of new control algorithms. Data transfer through private links can be implemented using Compassable UDT.
2. Composable UDT supports application-aware algorithms.
3. Ease of testing new algorithms for kernel space when using Composable UDT compared to modifying an OS kernel.

The UDP-based data transfer (UDT) protocol proposes rate-based congestion control that is implemented as application-level processes running atop UDP. Though UDT performs better than TCP over large BDP networks, UDT's potential is not fully realized as it does not model the end-system interactions between the operating system (OS) and network that contribute to congestion. The lack of such a model then forces UDT to rely on intuitive, but theoretically unfounded, heuristics.

A. UDT Fast Data Transfer Protocol

UDT is a UDP-based approach [12] and is considered to be the only UDP-based protocol that employs a congestion control algorithm targeting shared networks. It is a new application level protocol with support for user-configurable control algorithms and more powerful APIs. It has no security mechanism to protect itself from adversaries.

1. Packet Structures

UDT is designed to have two packet structures: data packets and control packets. The packets are distinguished by the first bit (flag bit) of the packet header. The data packet header starts with 0, while the control packet header starts with 1 as shown in figure1

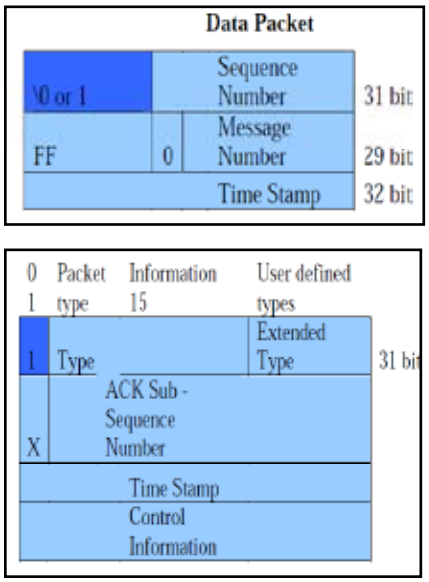

Fig. 1: Data Packet & Control packets

II. Literature Work

- Jing Liu et all concluded, UDT was designed to effectively utilize the rapidly emerging high-speed wide area optical networks. It is built on top of UDP with reliability control and congestion control, which makes it quite easy to install. The congestion control algorithm is the major internal functionality to enable UDT to effectively utilize high bandwidth. Meanwhile, we also implemented a set of APIs to support easy application implementation, including both reliable data streaming and partial reliable messaging.
- Yunhong Gu summarized work on the UDT high performance data transport protocol. UDT was designed to effectively utilize the rapidly emerging high speed wide area optical networks. It is built on top of UDP with reliability and Congestion Control. UDT is used in distributed data intensive application.

- Robert L. Grossman concluded that UDT is often used for data intensive applications for bulk data transfer. However it is difficult to implement a new protocol in a fast and efficient way. It plans to provide more built-in congestion control classes (eg Fast TCP) as a part of composable UDT releases. More experiment will be performed based on security issues.

III. Unsecured UDT

UDT is at the application layer or on top of UDP. Subsequently data that it carries are required to be transmitted securely and correctly by each application. The contention, therefore, for the need of security mechanisms of the new UDT is moreover derived from four important observations

- Absence of an inherent security mechanism, such as checksum for UDT
- Dependencies on user preferences and implementation on the layer on which it is Implemented
- Dependencies on existing security mechanisms of other layers on the stack
- Dependencies on TCP/UDP which are dependent on nodes and their addresses for high speed data transfer protocol leading to a number of attacks such as neighborhood, Sybil and DoS (Denial of Service) attacks

There are other attacks which are described in brief

1. Routing Attacks

When bulk packets are sent over a network using UDT, can lead to flooding attacks which consumes large amount of Bandwidth.

- S wants to send a packet to D
- S does not have a route to D
- S sends a routing request RREQ to all its neighbors
- They will either return a fresh path to D or forward the RREQ to their neighbor
- Once the RREQ reaches D, it returns a route reply RREP

2. Session Hijacking

- Session is created between a sender and a receiver using UDT Protocol
- Attacker Exploits valid session of sending and receiving packets
- Data can be snooped by stealing valid session ID Involves 2 Levels:

1. **Network level:** it involves transfer of Packets
2. **Application level:** it obtains session ID Preventions:

- Encryption
- Anti Virus Software

3. Authentication Attacks

- Problem
- other receivers of the data are not trusted
- Solve scheme
- authentication packet
- Packet authentication is concerned with:
- protecting the integrity of a packet
- validating identity of originator
- Alternative functions used:
- message encryption
- Digital Signatures

4. Defense against Attacks

- **Packet filtering:-** one defense attacks
- **Ingress filtering:-** blocking of packets from outside the network with a source address inside the network
- **Egress filtering: -** blocking outgoing packets from inside the network source address.
- **Using Encryption:-** Plain text is encrypted and Cipher text is decrypted by using keys
- **Firewall:-** Solid Solution to Security lies between local network and Internet filters harmful traffic.

IV. Advantages of UDT over TCP protocol

- Reliability
- Packet-based sequencing
- Acknowledgment and loss report from receiver
- ACK sub-sequencing
- Retransmission (based on loss report and timeout)
- Streaming and Messaging
- Buffer/memory management
- Connection maintenance
- Handshake, keep-alive message, teardown message
- Duplex
- Each UDT instance contains both a sender and a receiver

V. Conclusion and Future Work

In the present paper we have discussed the various issues in the UDT protocol. We have reviewed the attacks and compare the advantages of UDT over TCP Protocol. Protecting UDT can be achieved by introducing approaches related to self-certifying address generation and verification. A technique which can be applied without major modifications in practice is Cryptographically Generated Addresses (CGA). This technique is standardized in a protocol for IPv6. Similarly, HIP solves the problem of address generation in a different way by removing the functionality of IP addresses as both host identifiers and topological locations. However in order to achieve this, a new network layer, called Host Identity (HI), is introduced, which makes HIP incompatible with current network protocols.

In future work the same work can be shown with the result analysis. The work can be implemented on the simulators. Protecting UDT by using GSS-API in UDT is another approach; however, this needs to be thoroughly evaluated by application vendors. The use of the GSS-API interface does not in itself provide an absolute security service or assurance; instead, these attributes are dependent on the underlying mechanism(s) of UDT which support a GSS-API implementation to achieve adequate security mechanism. Another way of protecting UDT is by introducing Authentication-Option (AO). The work can be extended in future by changing the structure of UDT and to enhance the protocol with authenticity considereations.

References

[1] *Bernardo, D.V and Hoang, D. B "Empirical Survey: Experimentation and Implementations of High Speed Protocol Data Transfer for GRID" Proceedings of IEEE 25th International Conference on Advance Information Networking and Application Workshops, March 2011, Singapore*

[2] *Bernardo, D.V., Hoang, D.B. "A Security Framework and its Implementation in Fast Data Transfer Next Generation Protocol UDT", Journal of Information Assurance and Security 4(354-360) (2009) ISN 1554-1010, 2009.*

[3] *Mathis, M., Mahdavi, J., Floyd, S., Romanow, A, "TCP*

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

selective acknowledgment options" IETF RFC 2018 (April 1996)

[4] Rescorla, E., Modadugu, N. "Datagram Transport Layer Security", RFC 4347, IETF (April 2006)

[5] Xinwei Hong, and Robert L. Grossman, "Experiences in the Design and Implementation of a High Performance Transport Protocol", SC ', Pittsburgh, PA, 2011.

[6] Yunhong Gu and Robert L. Grossman, "UDT: UDP-based Data Transfer for High-Speed Wide Area Networks", Computer Networks (Elsevier). Volume 51, Issue 7. May 2007.

[7]. Hamill, J., Deckro, R., and Kloeber, J., "Evaluating information assurance strategies" in Decision Support Systems, Vol. 39, Issue 3 (May 2005), 463- 484.

[8]. H. I. for Information Technology, H. U. of Technology, et al. Infrastructure for HIP, 2008.

[9]. Harrison, D., RPI NS2 Graphing and Statistics Package, http://networks.ecse.rpi.edu/~harrisod/graph.html

[10]. Jokela, P., Moskowitz, R.,and Nikander, P., Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP). RFC 5202, IETF, April 2008.

[11]. Joubert, P., King, R., Neves, R., Russinovich, M., and Tracey, J.: Highperformance memory-based web servers: Kernel and user-space performance. USENIX '01, Boston, Massachusetts, June 2001.

[12]. Jray, W., "Generic Security Service API Version 2: C-bindings", RFC 2744, January 2000.

[13]. Kent, S., Atkinson, R., "Security Architecture for the Internet Protocol", RFC 2401, 1998.

[14]. Laganier, J. and Eggert, L., Host Identity Protocol (HIP) Rendezvous Extension. RFC 5204, IETF, April 2008.

[15]. Laganier, J., Koponen, T., and Eggert, L., Host Identity Protocol (HIP) Registration Extension. RFC 5203, IETF, April 2008.

[16]. Leon-Garcia, A., Widjaja, I., Communication Networks, McGraw Hill, 2000

[17]. Linn, J. "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January 2000.

[18] Linn, J., "The Kerberos Version 5 GSS-API Mechanism", IETF, RFC 1964, June 1996.

[19]. Mathis, M., Mahdavi, J., Floyd,S. and Romanow, A.: TCP selective acknowledgment options. IETF RFC 2018, April 1996.

[20]. Melnikov, A., Zeilenga,, K., Simple Authentication and Security Layer (SASL) IETF, RFC 4422, June 2006

[21]. Menezes, A.J., Oorschot, van P.C. and Vanstone, S. A., Handbook of Applied Cryptography, CRC Press, 1997

[22]. Moskowitz, R. and Nikander, P., RFC 4423: Host identity protocol (HIP) architecture, May 2006.

[23]. R. Moskowitz, R., Nikander, P. Jokela, P., and Henderson, T., Host Identity Protocol. RFC 5201, IETF, April 2008.

[24]. Neuman, C., Yu, T., Hartman, S., Raeburn, K., Kerberos Network Authentication Service (V5), IETF, RFC 1964, June 1996

[25]. NIST SP 800-37, Guide for the Security Certification and Accreditation of Federal Information Systems, May 2004.

[26]. NS2. http://isi.edu/nsna/ns.

[27]. PSU Evaluation Methods for Internet Security Technology (EMIST), 2004, http://emist.ist.psu.edu visited December 2009

[28]. Rabin, M., "Digitized signatures and public-key functions as intractable as Factorization" MIT/LCS Technical Report, TR-212 (1979).

[29]. Rescorla, E., Modadugu, N., "Datagram Transport Layer Security" RFC 4347, IETF, April 2006

[30]. Rivest, R.L., Shamir,A., and Adleman, L.M.,"A method for obtaining digital signature and publickeycryptosystems", Communication of ACM, 21, (1978),120-126