Università di Perugia
Dipartimento di Matematica e Informatica

Tesi triennale in ...

Vulnerabilità DDS OMG

Relatore
Prof. Francesco Santini

Candidato
Federico Ranocchia

Anno accademico 2024/2025

Quì la dedica...

Indice

1 Introduction **4**
1.1 Attacchi DDoS 4
1.1.1 DDoS blocco ricezione da parte del datareader Foglio 2 5
1.1.2 DDoS sfruttando estensione DDS security 6
1.2 Attacchi di enumerazione e di sniffing 7
1.2.1 Enumeration sniff foglio 2 e foglio 5 7
1.3 QoS Exploitation Attack foglio 1 9
1.3.1 Foglio 4-B Modifica maligna di ownership strength 10
1.3.2 Foglio 4-D Modifica maligna di LIFESPAN QoS 11

2 Background **13**
2.1 Sezione 13
2.1.1 Sottosezione 13

Capitolo 1

Introduction

In questo capitolo ci occuperemo di analizzare e comprendere delle vulnerabilità del protocollo DDS standard OMG (Object Management Group). In particolare verrà analizzato il vettore d'attacco, il protocollo utilizzato, il bersaglio dell'attacco e infine verrà proposta una soluzione applicabile per mitigare possibili attacchi non autorizzati. Nel prossimo capitolo grazie all'aiuto del software –inserire software– riusciremo a capire come queste vulnerabilità possono essere ricreate in un ambiente simulato. Queste vulnerabilità hanno una base di appoggio solida per l'attaccante. In molti casi un dispositivo ha già la possibilità di controllo di un partecipante all'interno della rete o ha la possibilità di modificare dei file di configurazione all'interno del network stesso.

Queste vulnerabilità riguardano la versione del DDS 1.4 con le specifiche dello standard OMG.

1.1 Attacchi DDoS

Questi attacchi consistono nel sovraccaricare uno o più dispositivi collegati alla rete DDS in modo tale da renderli non responsivi. Infatti molti sono di tipo I O T – e la potenza di calcolo nella maggior parte dei casi è ridotta. Per di più in molti casi vengono utilizzati dispositivi che non possono permettersi –delay– nell'analisi di certi dati, specialmente in ambiti RealTime in cui bisogna avere delle risposte rapide, come ad esempio nel campo della medicina e nel campo militare.

1.1.1 DDoS blocco ricezione da parte del datareader Foglio 2

Citazioni da foglio 2 a gogo Il vettore di attacco si trova nel protocollo RTPS che si occupa di scambiare pacchetti tra i DataReader (coloro che si iscrivono ai vari ai vari topic) e i DataWriter (di solito sono sensori che inviano dati). Questo protocollo utilizza il messaggio HEARTBEAT che viene mandato da un DataWriter a un DataReader per specificare il sequence number del DataWriter. Il sequence number serve al DataReader per sincronizzarsi con il DataWriter durante la ricezione dei pacchetto. Infatti il DataReader quando riceve il sequence number all'interno di un HEARTBEAT può identificare se ci sono o no dei pacchetti mancanti e in caso segnalarli al DataWriter.[7]

Un DataWriter inoltre può richiedere un messaggio ACKNACK da un DataReader se nell'HEARTBEAT inviato in precedenza dal DataWriter il parametro FINAL è attivo. Il messaggio ACKNACK consente di far rimanere sempre sincronizzato il DataReader al DataWriter che non potrà spedire nuovi pacchetti HEARTBEAT fino a quando non avrà ricevuto la conferma di ricezione con un messaggio ACKNACK. I controlli del sequence number all'interno dell'HEARTBEAT non sono sufficienti per coprire la rete da questo tipo di attacco:

- un primo controllo viene effettuato per verificare che non ci siano valori negativi
- un altro controllo serve a determinare se l'ultimo sequence number appena ricevuto non ha un valore più alto di quello ricevuto in precedenza

[7]

Dettagli attacco AFTER

Per sfruttare questa vulnerabilità l'attaccante deve utilizzare qualche strumento per sniffare la comunicazione tra il DataReader e il DataWriter, intercettando i messaggi HEARTBEAT. Dopo aver catturato un pacchetto di tipo HEARTBEAT e modificato il suo sequence number assegnandogli un valore molto alto, l'attaccante lo invia al DataReader. Una volta ricevuto il pacchetto il DataReader si metterà in attesa di un HEARTBEAT con un sequence number superiore a quello appena ricevuto. Di conseguenza il DataReader non elaborerà più i messaggi legittimi mandati dal DataWriter,

dato che hanno sequence numember più piccoli, bloccando così la sua esecuzione indefinitamente. Solo un messaggio HEARTBEAT con un sequence number maggiore a quello del DataReader farà ripristinare la sua esecuzione.[7]

Conclusioni AFTER

Di solito questo tipo di attacco è difficile da identificare. Un messaggio HEARTBEAT riguarda un solo topic, quindi il resto delle comunicazioni che avvengono su topic differenti o anche sullo stesso topic, ma con un DataReader diverso, non subiranno cambiamenti.

1.1.2 DDoS sfruttando estensione DDS security

Per l'attacco sopra citato dobbiamo considerare il modulo del DDS chiamato DDS security versione 1.1.(fonti ora da foglio 6) Questo si occupa di stabilire una connessione sicura tra i vari dispositivi della rete, infatti verranno utilizzati dei plugin da parte dei partecipanti che servono a:

- autenticazione
- controllo accesso
- crittografia
- login
- data tagging

[5] (foglio 3 pag 718)Per effettuare l'autenticazione un partecipante deve risolvere una challenge crittografica richiesta dal sistema di autenticazione della rete. Effettuato poi questo calcolo crittografico, il risultato viene controllato dal sistema di autenticazione per verificare se esso corrisponda all'hash del risultato della challenge crittografica.[6]

Questo attacco è stato scoperto con Proverif, un tool che viene usato per individuare vulnerabilità nei protocolli crittografici. È stato utilizzato in molti studi, come ad esempio nell'analisi della posta elettronica certificata e nell'analisi del TLS 1.3.[1]

Dettagli attacco AFTER foglio 3

L'attacco DDoS avviene durante la fase di autenticazione del protocollo DDS security 1.1, in particolare quando un nuovo dispositivo tenta di collegarsi alla rete e manda una richiesta di autenticazione all'ente di controllo. La richiesta del partecipante viene poi intercettata dall'attaccante che modifica i valori della challenge crittografica all'interno del pacchetto. Modificando ripetutamente questi valori, l'attaccante inizia a inviare molteplici richieste crittografiche alla sua vittima. Il partecipante comincerà a calcolare queste challenge per effettuare l'autenticazione, consumando tutte le sue risorse. Dato che, la vittima è probabilmente un dispositivo IoT che non dispone di una potenza di calcolo molto elevata, si ritroverà occupata per tutto il tempo necessario a risolvere le challenge crittografiche ricevute dall'attaccante, bloccando così il suo funzionamento.[6]

Conclusioni AFTER foglio 3

Una raccomandazione per mitigare questo attacco può essere quello di cambiare delle policy QoS impostando un tempo limite massimo per effettuare l'autenticazione. Queste policy possono fare in modo che i partecipanti non si ritrovino sopraffatti dalle troppe richieste di autenticazione. Un allarme potrebbe essere anche utile per identificare possibili tentativi DDoS di questo tipo, allertando così un amministratore. [6]

1.2 Attacchi di enumerazione e di sniffing

Dal foglio 2 Prendere informazioni DDS senza effettuare veri e propri attacchi di tipo attivo può essere molto utile per un attaccante che prova a penetrare una rete DDS. In molti casi tutto quello che deve fare l'attaccante è osservare i messaggi che vengono scambiati all'interno del network. Successivamente quando si ottengono informazioni a sufficienza sarà più facile per l'attaccante trovare un vettore di attacco.[7]

1.2.1 Enumeration sniff foglio 2 e foglio 5

Prendendo in considerazione, il protocollo RTPS e il suo modulo discovery, possiamo notare che di default sono molto "verbose", cioè scambiano informazioni in chiaro

durante le comunicazioni tra i vari dispositivi.[7] Il modulo discovery del protocollo RTPS a sua volta si suddivide in altri 2 protocolli fondamentali, che sono necessari per le specifiche DDS:

- Simple Participant Discovery Protocol (SPDP)
- Simple Endpoint Discovery Protocol (SEDP)

Per questo attacco ci focalizzeremo in particolare su SPDP che serve ad individuare la presenza dei partecipanti alla rete. In particolar modo il funzionamento si basa su un messaggi di tipo multicast e unicast che vengono mandati a tutti i dispositivi del network per informare chi è presente attualmente. [4]

Dettagli attacco AFTER

Utilizzando un software in grado di "sniffare" i vari pacchetti della rete, come un semplice script python è stato possibile analizzare il loro contenuto. I pacchetti analizzati sono quelli di tipo multicast RTPS-SPDP. All'interno di un pacchetto di questo tipo possiamo trovare: (nel foglio 2 non viene specificato bene di quale pacchetto si parla, ma guardando la documentazione da pag 125 del foglio 5, stiamo analizzando il pacchetto SPDPdiscoveredParticipandData) (da scrivere in corsivo) l'indirizzo ip dell'host, il prefisso GUID dell'RTPS, la versione dell RTPS, L'ID del venditore, informazioni riguardanti la sincronizzazione ed infine il contenuto dei submessages.[7]

Conclusioni AFTER

Di solito questo tipo di attacco è difficile da identificare e possono essere effettuati anche non avendo un dispositivo autenticato all'interno della rete.

Una soluzione potrebbe essere usare l'estensione del protocollo DDS security o eseguire la connessione tra i nodi tramite un tunnel con WireGuard per criptare le comunicazioni.

1.3 QoS Exploitation Attack foglio 1

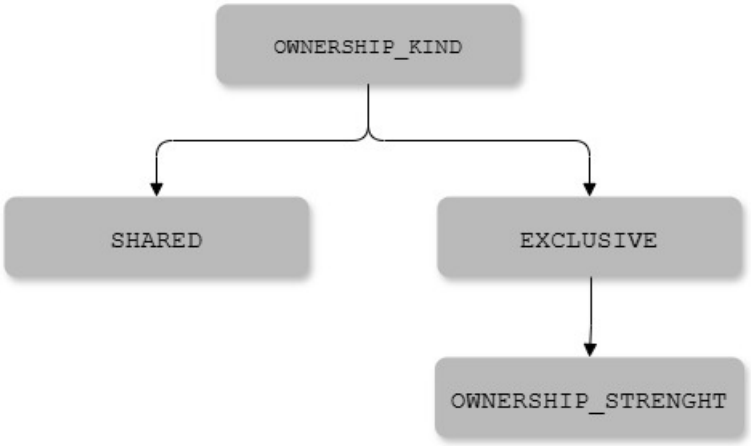

Figura 1.1: Illustrazione policy QoS del DDS

Queste tipologie di attacco sono possibili solo se certe policy QoS vengono modificate durante l'esecuzione della rete, specialmente il parametro OWNERSHIP-KIND che gestisce quanti DataWriter possono scrivere per un determinato Topic. Questo parametro può essere impostato in due modi diversi:

- SHARED: in questo modo più di un DataWriter possono aggiornare le informazioni di un topic. Inoltre un DataReader si può iscrivere a qualsiasi scrittore dello stesso topic.
- EXCLUSIVE: solo un DataWriter può aggiornare le informazioni di un topic. Il DataWriter che ha il permesso di scrittura per il topic è quello che dispone di un OWNERSHIP-strength con valore più alto.

Un'altra policy QoS che può essere usata come vettore di attacco è quella che regola il parametro LIFESPAN. Questa corrisponde al tempo limite massimo per la lettura da parte di un DataReader di un dato di un topic, che viene inserito all'interno del pacchetto inviato dal DataWriter. Per determinare se un pacchetto di un determinato topic è scaduto viene utilizzato il timestamp di creazione aggiungendo il LIFESPAN impostato; se questo "expiration time" risulta superiore all'orario durante la ricezione del DataReader allora l'informazione ricevuta è ancora valida. Per

funzionare gli orologi del DataWriter e del DataReader devono essere sincronizzati tra di loro.

Un'altra importante policy da considerare è quella riguardo all'affidabilità (RELIABILITY) dei dati riguardanti un topic che può essere impostata in due modi:

- RELIABLE: questa impostazione costringe il DataReader a farsi ritrasmettere dal DataWriter i pacchetti mancanti o ricevuti in maniera errata. In questo modo le informazioni del DataReader saranno sempre corrette anche se non sempre saranno aggiornate in tempo reale.
- BEST-EFFORT: l'impostazione predefinita non consente il recupero dei pacchetti mancanti o corrotti del DataReader, quindi, quest'ultimo potrebbe anche perdere dei pacchetti che gli sono stati inviati.

[4]

1.3.1 Foglio 4-B Modifica maligna di ownership strength

In una rete dove si utilizza un OWNERSHIP-kind di tipo EXCLUSIVE è possibile utilizzare l'OWNERSHIP-strength a favore dell'attaccante. Infatti è possibile far ricevere informazioni a un DataWriter in maniera errata, dato che quest'ultimo non riceverà più informazioni da una fonte affidabile.[3]

Foglio 4-B Dettagli attacco

L'attaccante, con un DataWriter in suo possesso all'interno di una rete DDS, può sfruttare il fatto che il topic preso di mira può essere aggiornato solo dal DataWriter con l'OWNERSHIP-strength più alta. Per effettuare questo attacco, tutto quello che serve, è sapere il topic che si vuole modificare, le policy QoS in uso e il valore dell ownership-strength. L'ultimo passo è quello di impostare il topic scelto nel DataWriter dell'attaccante con OWNERSHIP-strength superiore a quello utilizzato dal DataWriter originario. Ora i DataReader che sono iscritti al topic bersaglio ricevono le informazioni dal DataWriter dell'attaccante. [3]

Foglio 4-B Conclusioni

L'OWNERSHIP-kind di tipo EXCLUSIVE è utilizzata in contesti dove le informazioni ricevute dal DataReader devono essere accurate dato che un singolo scrittore (in molti casi si tratta di un sensore) può mandare nuovi aggiornamenti del topic. Se l'attaccante, dovesse riuscire a modificare i valori del topic con questo attacco, potrebbe causare in certi casi molti danni, specialmente se il DataWriter dell'attaccante riuscisse a mandare degli aggiornamenti del topic senza essere scoperto. [3]

Una soluzione utile a risolvere questo vettore di attacco potrebbe essere l'utilizzo dell'estensione DDS security che rende impossibile capire qual è il topic bersaglio perchè i messaggi scambiati tra DataReader e DataWriter sono criptati.

1.3.2 Foglio 4-D Modifica maligna di LIFESPAN QoS

L'attaccante a volte potrebbe modificare le policy QoS riguardanti il LIFESPAN e se necessario il parametro RELIABLE. Infatti il tempo limite di scadenza dei pacchetti, contenenti i dati del topic, può essere impostato a valori molto piccoli creando problemi di comunicazione tra un DataWriter e un DataReader. Utilizzando un'affidabilità di tipo RELIABLE si a riesce mitigare l'attaccante che così deve utilizzare valori più estremi per compromettere la comunicazione. Questo test è stato dimostrato con RTI Shapes Demo che implementa una soluzione DDS di RTI corrispondente alle specifiche dello standard OMG. [3]

Foglio 4-D Dettagli attacco

Avendo sotto controllo questi due parametri policy, l'attaccante può modificare la policy dei DataWriter in modo tale da avere un LIFESPAN molto piccolo. Così facendo, i pacchetti spediti dal publisher arriveranno già scaduti e non potranno più essere utilizzati dai DataReader. In certi casi il pacchetto che deve essere inviato viene distrutto dallo stesso DataWriter all'interno della sua coda prima dell'invio. In questo caso il test è stato effettuato impostando il valore di LIFESPAN $<$ 80ms dove si è visto che nessun pacchetto raggiunge il DataReader. Se si aumenta il valore tra gli 80ms e i 100ms già si può notare che dei pacchetti vengono letti con successo dal DataReader, mentre altri vengono eliminati prima della lettura. Infine impostando

un valore LIFESPAN >= 120ms si può notare che la comunicazione tra publisher e subscriber avviene senza nessun problema.

Un dettaglio da aggiungere è che se su RTI Shapes veniva impostata la policy dell'affidabilità (RELIABILITY) di tipo RELIABLE i millisecondi utilizzati dal LIFESPAN per compromettere le comunicazioni tra DataReader e DataWriter devono essere moltiplicati per un fattore di 0.01. Quindi, ad esempio se si ottiene un completo annullamento delle comunicazioni con un LIFESPAN < 80ms utilizzando il la RELIABILITY di tipo BEST-EFFORT, per ottenere lo stesso risultato con RELIABILITY di tipo RELIABLE dobbiamo impostare un LIFESPAN < 0.8ms. [3]

Foglio 4-D Conclusioni

Inizialmente molte reti DDS hanno impostato la RELIABILITY di tipo BEST-EFFORT che è l'impostazione predefinita. Quindi nella maggior parte dei casi l'attaccante non si deve preoccupare di questo parametro.

Una possibile soluzione sarebbe quella di impostare qualche tipo di controllo in modo tale da avvertire un operatore umano se molti pacchetti vengono scartati perché arrivati con un LIFESPAN scaduto. Questo controllo potrebbe essere anche utile, nel caso in cui il DataWriter e il DataReader si trovassero distanti fisicamente tra di loro, per verificare la qualità del collegamento. [3]

Tipo di attacco	**Vettore attacco**	**Protoc./ Estens.**	**Bersaglio nella rete**	**Software**	**Soluzione**
Discovery devices[7]	Verbose nature of RTPS	DDSI-RTPS	Tutti i partecipanti	Sniffer python	WireGuard
DDos[7]	Heartbeat sequence number	DDSI-RTPS	DataReader	Sniffer python	-
DDoS[6]	Authentication challenge	DDS security 1.1 Discovery protoc.	Tutti i partecipanti	Proverif	Scadenza richieste di autenticazione
QoS policy[3]	ownership-strength	DDSI-RTPS	DataReader	RTI shapes	DDS security
QoS policy[3]	LIFESPAN	DDSI-RTPS	DataReader	RTI shapes	Controllo per LIFESPAN scartati

Tabella 1.1: La versione DDS in tutti i casi è la 1.4

Capitolo 2

Background

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed do eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullamco laboriosam, nisi ut aliquid ex ea commodi consequatur. Duis aute irure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

2.1 Sezione

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed do eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullamco laboriosam, nisi ut aliquid ex ea commodi consequatur. Duis aute irure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

2.1.1 Sottosezione

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed do eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullamco laboriosam, nisi ut aliquid ex ea commodi consequatur. Duis aute irure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Citazione[2]

- 1
- 2

Figura 2.1: didascalia figura.

Riferimento immagine (o tabella... o sezione...) 2.1

$$\pi_i(v) = \sum \frac{x_{S_i}}{\mathrm{N}} \tag{2.1}$$

Elenco delle figure

1.1 Illustrazione policy QoS del DDS 9

2.1 didascalia figura. 14

Elenco delle tabelle

1.1 La versione DDS in tutti i casi è la 1.4 12

Bibliografia

[1] Bruno Blanchet, Ben Smyth, Vincent Cheval, and Marc Sylvestre. *ProVerif 2.05: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial*, 2023. URL `https://bblanche.gitlabpages.inria.fr/proverif/manual.pdf`. [Accesso: 2 febbraio 2025].

[2] Edsger W. Dijkstra. A bagatelle on euclid's algorithm. In Manfred Broy, editor, *Proceedings of the NATO Advanced Study Institute on Deductive Program Design, Marktoberdorf, Germany*, pages 21–23, 1996. ISBN 3-540-60947-4.

[3] Michael James Michaud, Thomas R. Dean, and Sylvain P. Leblanc. Attacking OMG data distribution service (DDS) based real-time mission critical distributed systems. In *13th International Conference on Malicious and Unwanted Software, MALWARE 2018, Nantucket, MA, USA, October 22-24, 2018*, pages 68–77. IEEE, 2018. doi: 10.1109/MALWARE.2018.8659368. URL `https://doi.org/10.1109/MALWARE.2018.8659368`.

[4] Object Management Group. *OMG Data Distribution Service*. 2015. URL `http://www.omg.org/spec/DDS/1.4/PDF`. [Accesso: 2 febbraio 2025].

[5] Object Management Group. *DDS Security*. 2018. URL `http://www.omg.org/spec/DDS/1.4/PDF`. [Accesso: 2 febbraio 2025].

[6] Binghan Wang, Hui Li, and Jingjing Guan. A formal analysis of data distribution service security. In Jianying Zhou, Tony Q. S. Quek, Debin Gao, and Alvaro A. Cárdenas, editors, *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, ASIA CCS 2024, Singapore, July 1-5, 2024*. ACM, 2024. doi: 10.1145/3634737.3656288. URL `https://doi.org/10.1145/3634737.3656288`.

[7] Thomas White, Michael N. Johnstone, and Matthew Peacock. An investigation into some security issues in the dds messaging protocol, 2017. URL https://api.semanticscholar.org/CorpusID:52840449.

Ringraziamenti

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed do eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullamco laboriosam, nisi ut aliquid ex ea commodi consequatur. Duis aute irure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Caio