



UNIVERSITÀ DI PERUGIA
Dipartimento di Matematica e Informatica



TESI TRIENNALE IN ...

Titolo Tesi

Relatore

Dott. Francesco Santini

Candidato

Federico Ranocchia

Anno accademico 2024/2025

Quì la dedica...

Indice

| | | |
|----------|---|-----------|
| 1 | Introduction | 4 |
| 1.1 | Attacchi di tipo DDoS | 4 |
| 1.1.1 | DDoS blocco ricezione da parte del data-reader Foglio 2 | 5 |
| 1.1.2 | DDoS sfruttando estensione DDS security | 6 |
| 1.2 | Attacchi di tipo Discovery | 7 |
| 1.2.1 | Enumeration sniff foglio 2 e foglio 5 | 7 |
| 1.3 | Attacchi sfruttando le policy QoS foglio 1 | 8 |
| 1.3.1 | Foglio 4-B Modifica maligna di ownership strength | 9 |
| 1.3.2 | Foglio 4-D Modifica maligna di LIFESPAN QoS | 10 |
| 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. Successivamente grazie all'aiuto di un software riusciremo a capire come queste vulnerabilità possono influire sul funzionamento degli –host– collegati alla rete DDS. In molti casi l'attaccante ha a disposizione un dispositivo all'interno della rete sotto il suo controllo.

1.1 Attacchi di tipo DDoS

Questo attacco consiste nel sovraccaricare un dispositivo collegato alla rete DDS in modo tale da renderlo inutilizzabile. Infatti dato che i dispositivi collegati sono di –tipo I O T– la potenza di calcolo nella maggior parte dei casi sarà molto ridotta. Inoltre in molti casi ci possiamo ritrovare ad utilizzare dispositivi che non possono permettersi –delay– nell'analisi di certi dati, specialmente in ambiti dove bisogna avere una risposta sempre rapida e disponibile, come ad esempio nel campo della medicina e nel campo militare.

1.1.1 DDoS blocco ricezione da parte del data-reader Foglio 2

Citazioni da foglio 2 a gogo Il vettore di attacco si trova nell'implementazione del DDS chiamata DDSI-RTPS che si occupa di scambiare messaggi tra i data-reader (coloro che si iscrivono ai vari ai vari topic) e i data-subscribers (di solito sono sensori che mandano dati). Per comunicare questi dispositivi utilizzano il protocollo RTPS. Questo protocollo utilizza il messaggio HEARTBEAT che viene mandato da un data-writer a un data-reader per specificare il sequence number nel data-writer. All'interno del messaggio HEARTBEAT troviamo il sequence number che serve al data-reader per sincronizzarsi con il data-writer durante la ricezione dei messaggi. Infatti il data-reader quando riceve il sequence number all'interno di un HEARTBEAT può identificare se ci sono dei pacchetti mancanti e segnalarli al data-writer.

Un data-writer inoltre può richiedere un messaggio ACKNACK da un data-reader se nel messaggio HEARTBEAT inviato dal data-writer viene specificata la flag FINAL. In casi in cui bisogna essere certi che il data-reader riceve tutti i dati del data-writer, quest'ultimo manda un HEARTBEAT con la flag FINAL impostata, al data-reader che successivamente deve rispondere necessariamente con un messaggio ACKNACK per confermare la ricezione nel messaggio HEARTBEAT. I controlli HEARTBEAT effettuati dal data-reader, infatti non sono sufficienti a coprire questo tipo di attacco dato che quest'ultimo:

- esegue un check per verificare che non vi siano numeri negativi
- controlla che l'ultimo sequence number arrivato non ha un valore più alto del sequence number ricevuto in precedenza

Dettagli attacco AFTER

Per sfruttare questa vulnerabilità l'attaccante deve utilizzare qualche strumento per sniffare la comunicazione tra data-reader e data-writer e intercettare un messaggio di tipo HEARTBEAT. Successivamente l'attaccante modifica il valore del sequence number del messaggio HEARTBEAT. Il messaggio poi viene mandato verso il data-writer così facendolo rimanere in attesa di un messaggio HEARTBEAT con Un sequence number superiore a quello appena ricevuto. Facendo così il data-reader non legge più i messaggi mandati dal data-writer e bloccando così l'esecuzione del data-reader finchè il sequence number non sarà superiore a quello ricevuto dall'attaccante.

Conclusioni AFTER

Di solito questi tipo di attacco sono difficili da identificare.

1.1.2 DDoS sfruttando estensione DDS security

In questo attacco dobbiamo considerare il modulo del DDS chiamato DDS security versione 1.1.(fonti ora da foglio 6) Questo modulo si occupa di stabilire una connessione sicura tra i vari dispositivi della rete. Infatti verranno utilizzate delle api da parte dei partecipanti per effettuare le varie azioni, come ad esempio iscriversi ad un topic e pubblicare un messaggio del topic. (foglio 3 pag 718) Per effettuare l'autenticazione un partecipante deve prima risolvere una challenge criptografica richiesta dal sistema di autenticazione della rete. Effettuato poi questo calcolo criptografico il risultato viene controllato dal sistema di autenticazione, controllando che se il risultato inviato corrisponde all'hash della challenge criptografica.

Dettagli attacco AFTER foglio 3

L'attacco DDoS si svolge proprio durante la fase di autenticazione del protocollo DDS security 1.1, in particolare quando un nuovo dispositivo legittimo si vuole collegare alla rete e comincia a mandare una richiesta di autenticazione all'ente di controllo. La richiesta del partecipante poi viene intercettata dall'attaccante che modifica i valori all'interno del pacchetto delle challenge criptografiche. Modificando ogni volta questi valori, l'attaccante comincia a mandare tante richieste criptografiche alla sua vittima. Il partecipante comincerà a calcolare queste challenge per effettuare l'autenticazione investendo tutte le risorse necessarie. Dato che molto probabilmente chi riceve questo attacco è un dispositivo IOT che non dispone di una potenza di calcolo molto elevata e si ritroverà occupato per tutto il tempo necessario per risolvere le challenge criptografiche ricevute dall'attaccante, bloccando così il suo funzionamento.

Conclusioni AFTER foglio 3

Una raccomandazione per mitigare questo attacco può essere quello di cambiare delle policy QoS in modo tale da impostare un tempo limite massimo per effettuare l'autenticazione. Impostando un limite simile queste policy possono prevenire che i

partecipanti si ritrovino sopraffatti dalle troppe richieste di autenticazione. Un allarme potrebbe essere anche utile per identificare possibili tentativi DDoS di questo tipo così allertando un amministratore. (parlare di proverif)

1.2 Attacchi di tipo Discovery

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.

1.2.1 Enumeration sniff foglio 2 e foglio 5

Prendendo in considerazione il protocollo DDSI-RTPS, possiamo notare che di default quest'ultimo è molto "verbose", cioè scambia molte informazioni in chiaro durante le comunicazioni tra i vari dispositivi. In particolare il modulo discovery del protocollo RTPS che a sua volta si suddivide in altri 2 protocolli fondamentali che sono necessari:(foglio 5 pag 123)

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

Per questo attacco ci focalizzeremo in particolare nel SPDP che serve a individuare la presenza dei partecipanti alla rete. In particolar modo il funzionamento si basa su un messaggio di tipo multicast e unicast che viene mandato a tutti i dispositivi della rete per informare chi è presente attualmente. (foglio 5 pag 125)

Dettagli attacco AFTER

Utilizzando un qualche 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 e infine il contenuto dei submessages.

Conclusioni AFTER

Di solito questi tipo di attacco sono difficili da identificare e possono essere effettuati anche non avendo un dispositivo autenticato all'interno della rete.

1.3 Attacchi sfruttando le policy QoS foglio 1

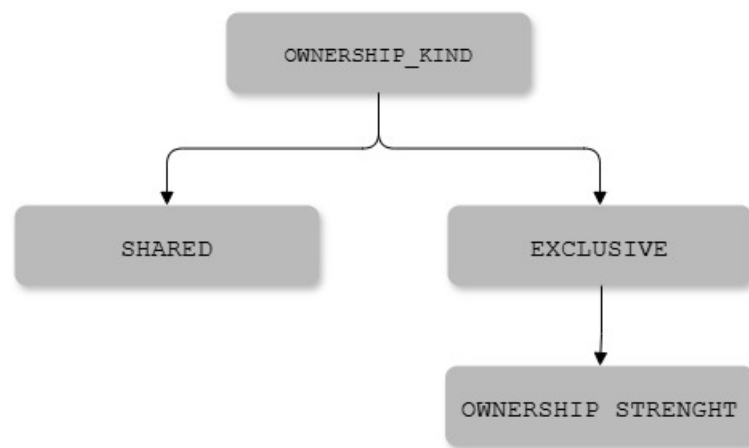


Figura 1.1: Illustrazione policy QoS del DDS

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`. 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 che consente il recupero dei pacchetti mancanti del `DataReader`, quindi quest'ultimo potrebbe anche perdere dei pacchetti che gli sono stati inviati.

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.

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 `DataWri-`

ter 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 quella utilizzata dal DataWriter originario che aggiorna il topic. Ora i DataReader che sono iscritti al topic bersaglio ricevono le informazioni dal DataWriter dell'attaccante.

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, può causare in certi casi molti danni a seconda della rete DDS, specialmente se il DataWriter dell'attaccante riesce a mandare degli aggiornamenti del topic senza che venga scoperto.

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 in questo attacco deve poter modificare le policy QoS riguardanti LIFESPAN e se necessario il parametro RELIABLE. Infatti il tempo di limite di scadenza dei pacchetti (LIFESPAN) può essere impostato a valori molto piccoli creando problemi di comunicazione tra un DataWriter e un DataReader. Utilizzando un'affidabilità di tipo RELIABLE si riesce a mitigare l'attaccante deve utilizzare valori più estremi per compromettere la comunicazione. Questo test è stato dimostrando RTI Shapes Demo che implementa una soluzione DDS di RTI corrisponde a quella dello standard OMG.

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 essere utilizzati dai DataReader, in certi casi il pacchetto che deve essere inviato viene distrutto dallo stesso DataWriter all'interno della coda prima dell'invio del pacchetto. In questo caso il test è stato effettuato impostando il valore di LIFESPAN $< 80\text{ms}$ 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 $\geq 120\text{ms}$ 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 $< 80\text{ms}$ utilizzando la RELIABILITY di tipo BEST-EFFORT, per ottenere lo stesso risultato con RELIABILITY di tipo RELIABLE dobbiamo impostare un LIFESPAN $< 0.8\text{ms}$.

Foglio 4-D Conclusioni

Inizialmente molte reti DDS hanno impostato come parametro RELIABILITY quella di tipo BEST-EFFORT che è l'impostazione di default. 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 perchè arrivati con un LIFESPAN scaduto. Questo controllo potrebbe essere anche utile, nel caso in cui il DataWriter e il DataReader si trovano distanti fisicamente tra di loro, per verificare la qualità del collegamento.

Questa è una sottosezione

La teoria dell'attacco ci dice che se

| Tipo di attacco | Vettore attacco | Protoc./ Estens. | Bersaglio nella rete | Software | Soluzione |
|----------------------|---------------------------|------------------------------------|----------------------|----------------|--------------------------------------|
| Discovery devices[2] | Verbose nature of RTPS | DDSI-RTPS | Tutti i partecipanti | Sniffer python | - |
| DDos[2] | Heartbeat sequence number | DDSI-RTPS | Data-reader | Sniffer python | - |
| DDoS[3] | Authentication challenge | DDS security 1.1 Discovery protoc. | Tutti i partecipanti | Proverif | Scadenza richieste di autenticazione |
| QoS policy[4] | ownership-strength | DDSI-RTPS | DataReader | RTI shapes | DDS security |

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

Capitolo 2

Background

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullamco laboriosam, nisi ut aliquid ex ea commodo consequat. 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 adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullamco laboriosam, nisi ut aliquid ex ea commodo consequat. 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 adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullamco laboriosam, nisi ut aliquid ex ea commodo consequat. 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[1]

- 1
- 2



Figura 2.1: didascalia figura.

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

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

Bibliografia

- [1] 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.

Ringraziamenti

Lorem ipsum dolor sit amet, consectetur adipisci elit, sed do eiusmod tempor incididunt 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