Jornadas Nacionales de Robótica
Spanish Robotics Conference
14-15 Junio 2018 http://jnr2018.eii.uva.es

Aprendizaje automático aplicado a la ciberseguridad en robots autónomos

Ángel Manuel Guerrero-Higueras[b,*], Vicente Matellán[a], Francisco Javier Rodríguez-Lera[c], Miguel Á. Conde[b], Camino Fernández-Llamas[a]

[a] *Grupo de Robótica. Depto. de Ingenierías Mecánica, Informática y Aero-espacial. Universidad de León, España.*
[b] *Instituto de Ciencias Aplicadas a la Ciberseguridad (RIASC), Universidad de León, España.*
[c] *AI Robolab. University of Luxembourg, Luxemburgo.*

Resumen

El problema que aborda el proyecto es la ciberseguridad de los robots autónomos. En primer lugar, se caracterizarán las vulnerabilidades conocidas en robots mediante un análisis sistemático de la literatura. También se instalará un honeypot que simule un robot controlado por el middleware estándar en desarrollo robótico ROS (Robotic Operating System). Con los datos extraídos del honeypot se espera generar un extenso conjunto de datos (dataset) que se analizará utilizando técnicas existentes de análisis de datos y de aprendizaje automático. A partir de los resultados de ese análisis se desarrollarán nuevos algoritmos de detección y prevención de ciber-ataques específicos para robots. En paralelo, se diseñarán e implementarán componentes para ROS que permitan integrar los sistemas de ciberseguridad en diferentes tipos de robots. Se desarrollarán tres prototipos para la validación de las ideas propuestas. Para ello, se utilizarán dos robots de los que dispone actualmente el grupo: un bi-manipulador Baxter, el robot más popular para los nuevos entornos de fabricación puesto que permite la colaboración entre robots y humanos en las líneas de producción (industria 4.0), y un manipulador móvil de servicios RB1 (fabricado por la empresa española Robotnik). Adicionalmente, se solicita financiación para disponer de un androide (Pepper, fabricado por Aldebaran) orientado a la interacción social, que complementará los robots de que dispone el grupo. *Copyright © 2018 CEA.*

Palabras Clave:
Ciber-seguridad, robótica, sistemas autónomos, aprendizaje automático, sistemas de localización.
Datos del Proyecto:
Denominación del proyecto: Desarrollo de componentes software reutilizables basados en aprendizaje automático para la ciberseguridad de robots autónomos.
Referencia: LE028P17
Investigador/es responsable/es: Camino Fernández-Llamas
Tipo de proyecto (internacional, nacional, autonómico, transferencia): Autonómico
Entidad/es financiadora/s: Fondo Europeo de Desarrollo Regional y Consejería de Educación de la Junta de Castilla y León
Fecha de inicio/fin: Noviembre 2017 / Octubre 2019

1. Introducción

Cada día, millones de robots perciben, actúan y toman decisiones en entornos donde tienen que interactuar con humanos. Dichas decisiones varían desde simples ajustes internos, como puede ser la disminución del brillo de la pantalla, a tareas complejas como interactuar con un cliente en un sitio público. Además, las nuevas plataformas son capaces de auto-ajustar en tiempo real las reglas con las que toman dichas decisiones, permitiendo resolver problemas que inicialmente no contemplaban. Esta situación despliega una serie de problemas de ciberseguridad, como muestra Morante et al. (2015), y que es necesario afrontar para continuar con el desarrollo y despliegue de plataformas robóticas en entornos con humanos.

El problema de la ciberseguridad en sistemas robóticos se aborda en la literatura científica desde dos perspectivas: *pasiva*, en la que se trabaja analizando un producto ya existente; y *activa*, en la que se plantean una serie de contra-medidas para sobrellevar los problemas asociados a una o múltiples vulnerabilidades. En el caso pasivo contemplamos el análisis de vulnerabilidades o auditado de plataformas existentes, ver Heelan (2011). Por ejemplo, en Bonaci y Chizeck (2012) se analizan los diferentes problemas de seguridad de los robots de rescate, o en Bonaci et al. (2015) diferentes tipos de ataques en robots cirujanos. Desde la perspectiva activa, se plantea la seguridad desde su concepción, ver Mouratidis (2004); Su et al. (2004); o, como se plantea en el proyecto SAFE, DeHon et al. (2011), desarrollando plataformas verificables. En nuestro proyecto, se encara la ciberseguridad desde ambas perspectivas, pasiva, revisando las vulnerabilidades de dos tipos de robots, industriales y de servicios; y activa, planteando el uso de *honeypots*.

*Autor en correspondencia.
Correos electrónicos: am.guerrero@unileon.es (Ángel Manuel Guerrero-Higueras), vicente.matellan@unileon.es (Vicente Matellán), francisco.lera@uni.lu (Francisco Javier Rodríguez-Lera), mcong@unileon.es (Miguel Á. Conde), camino.fernandez@unileon.es (Camino Fernández-Llamas)
URL: http://robotica.unileon.es/~vmo (Vicente Matellán)

Inicialmente, este proyecto plantea la evaluación y auditoría de los problemas de ciberseguridad en las dos categorías básicas de robots, industriales y de servicios. Hasta ahora, los estudios dedicados a esta tarea son escasos. Desde el punto de vista de la seguridad y las vulnerabilidades existen trabajos como el de Roosa y Decker (2013) analizando robots en el área de salud, o en el trabajo de nuestro grupo, Guerrero-Higueras et al. (2017), en el que se audita las situaciones de ataque a un sistema de localización de un robot. Desde el punto de vista de la seguridad e integridad del usuario, Fosch-Villaronga (2015) desarrollan el modelo Care Robot Impact Assessment (CRIA) basado en la ISO 13482:2014, pero está más dedicado a la seguridad física en la interacción humano-robot. Nuestro objetivo en este caso es aunar y actualizar la literatura actual realizando un estudio que permita caracterizar los problemas de ciberseguridad existentes en plataformas comerciales actuales tanto industriales como de servicios.

Dada dicha caracterización, y utilizando modelados previos realizados por el grupo, como el presentado en Lera et al. (2016), desarrollaremos un honeypot robótico. Un honeypot, Zhang et al. (2003), es un recurso de seguridad aplicado a un entorno ciber-físico, que se espera que interactúe cuando el sistema se vea comprometido, ver Litchfield et al. (2016). Los honeypot plantean tres posibilidades de funcionamiento según Baykara y Das (2015): de baja interacción, de alta interacción, e híbridos. La primera plantea sistemas donde se emulan los dispositivos en el que se despliega el honeypot, Durairajan et al. (2016); la segunda utiliza entornos reales de trabajo, Guarnizo et al. (2017); finalmente, los sistemas híbridos, mezclan ambas soluciones. En nuestra propuesta, implementaremos las tres posibilidades y analizaremos los resultados obtenidos mediante el uso de técnicas de aprendizaje automático para encontrar la mejor solución a aplicar en cada uno de los entornos estudiados industriales y de servicio.

Finalmente, resaltar que el número de trabajos científicos que utilizan honeypot con robots es muy reducido. Este año el Georgia Institute of Technology, en Irvene et al. (2018), ha presentado un trabajo "pionero" donde se describe un framework específico para robótica. Desafortunadamente el modelo está descrito a alto nivel, sin implementaciones y únicamente con resultados iniciales. Es complicado encontrar en la literatura muchas más aproximaciones generales de este concepto. Sí que es posible encontrar algún planteamiento o solución particular como la realizada en 2013 por McClean et al. (2013), que implementó un primer ejemplo de honeypot sobre un robot que utilizaba ROS (Robotic Operating System) en una de las DEFCON, o como la realizada en ROSRV, Huang et al. (2014b), cuyas características de análisis de red y contra-medidas se engloban dentro de las tareas que se le asocian clásicamente a un honeypot.

Esta situación nos motiva de doble manera: por un lado a plantear el uso de honeypot en el ámbito de la robótica dada la escasez de este tipo de aproximaciones experimentales, y por otro, a hacerlo sobre ROS ya que es el middleware para control de robots más extendido, ver Quigley et al. (2009), y es por defecto el elemento más importante de un robot. Dicha motivación va unida con la experiencia previa del grupo, que ha realizado ya trabajos preliminares sobre la seguridad de ROS, ver Lera et al. (2017), lo que nos permite afrontar y especificar el proyecto de forma más precisa.

El resto del artículo se organiza como sigue: La sección 2 muestra los avances realizados en lo que al bastionado de ROS se refiere. La sección 3, muestra las primeras aportaciones en detección de ciber-ataques en tiempo real utilizando algoritmos de aprendizaje automático. La sección 4 describe un dataset construido específicamente para entrenar y evaluar modelos de detección y seguimiento de personas basados en redes neuronales. Por último, se presentan unas breves conclusiones del trabajo realizado hasta la fecha en el marco del proyecto.

2. Bastionado de ROS mediante comunicaciones cifradas y reglas semánticas

ROS es básicamente un conjunto de bibliotecas que proporcionan una serie de servicios para sistemas robóticos. Estos servicios son similares a los que proporciona un sistema operativo: abstracción de hardware para sensores y actuadores, control de dispositivos de bajo nivel y comunicación entre procesos. El framework ROS es un sistema distribuido de transmisión de mensajes. El procesamiento se lleva a cabo en procesos denominados *nodos* que envían y reciben *mensajes* a través de unas estructuras de comunicación denominadas *topics*.

Los nodos ROS se pueden ejecutar en la misma o en diferentes máquinas. La configuración habitual está compuesta por al menos un ROS Master y varios clientes. ROS Master es el elemento clave en el sistema ROS. Se ejecuta como un servicio de nombres y gestiona la información de registro sobre los topics utilizados por los diferentes nodos. El ROS Master se actualiza en tiempo real con datos de los nodos en ejecución, que proporcionan información sobre los topics que publican o a los que se suscriben y el tipo de mensaje utilizado en cada topic. La figura 1 describe el modelo conceptual de ROS, mostrando el ROS Master con dos nodos, Listener y Talker. Talker publica mensajes de tipo *foo* en el topic *bar*, mientras que el nodo Listener está suscrito al topic *bar* para recibir sus mensajes.

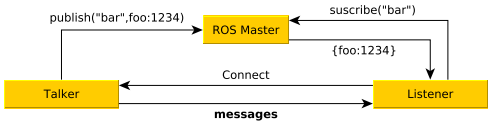

Figura 1: Modelo conceptual de ROS.

Desafortunadamente, no se consideró seguridad en el diseño de este mecanismo de comunicación distribuida. Los principales problemas de seguridad en la versión actual de ROS son las comunicaciones en claro, puertos TCP no protegidos y almacenamiento de datos sin cifrar. Por lo tanto, ROS plantea riesgos de privacidad para centros de investigación, empresas y particulares. Las comunicaciones en claro de ROS pueden sufrir escuchas, suplantación de identidad o denegación de servicio (DoS). Un intruso puede comprometer la privacidad de los usuarios finales accediendo a las lecturas de cámaras u otros sensores, o monitorizando a una persona con un robot comprometido. En particular, al desarrollar robots de asistencia, la privacidad de los pacientes y cuidadores es una gran preocupación.

Nuestra propuesta para mejorar la seguridad de ROS, presentada en Balsa-Comerón et al. (2017), incluye dos etapas. Primero, aplicamos algoritmos de cifrado para proporcionar una

capa de seguridad a los datos de ROS. En segundo lugar, definimos reglas semánticas que proporcionan cierto metacontrol para comportamientos específicos. Por ejemplo, podríamos definir una regla para mejorar la seguridad ignorando decisiones relativas al movimiento del robot cuando una persona está demasiado cerca.

En Lera et al. (2016) se evaluó el rendimiento de bastionar ROS mediante el cifrado de sus comunicaciones con 3DES, ver Smid y Branstad (1988). En este proyecto, aplicamos AES para cifrar los mensajes que publican algunos nodos en topics concretos. Los nodos suscritos a estos topics cifrados tienen que descifrar los mensajes antes de poder utilizar los datos.

Asegurar las comunicaciones es solo una dimensión en la ciberseguridad en sistemas autónomos. Si queremos que los robots trabajen en entornos cambiantes, como puede ser una vivienda, necesitamos asegurar las capacidades de navegación y los mecanismos de interacción, para evitar comportamientos manipulados o maliciosos y hacer que los robots sean asistentes confiables. Con este objetivo en mente, incluimos reglas semánticas que controlan los mensajes que se pasan entre nodos ROS. Usamos el framework de ROSRV, ver Huang et al. (2014a), que permite aplicar políticas de seguridad. Definimos dos reglas, *monitores* siguiendo la nomenclatura ROSRV, como prueba de concepto. El primer monitor tiene como objetivo rechazar cualquier mensaje que no esté cifrado. Este monitor proporciona confidencialidad, pero también puede proporcionar integridad y autenticación. El segundo, permite detectar ciberataques contra sistemas de localización en tiempo real (RTLSS), dispositivos comúnmente utilizados por robots autónomos. Ambos se explican en profundidad a continuación.

2.1. *Monitor 1: detección de mensajes cifrados*

La figura 2 representa el modelo conceptual para el intercambio de mensajes a través del topic */chatter*. El monitor 1 analiza los mensajes que el nodo *Simple-talker* publica en este topic descartando aquellos que no estén cifrados. *Simple-listener* consume los mensajes publicados por *Simple-talker*.

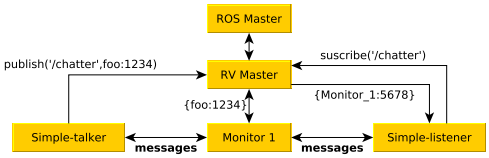

Figura 2: Detección de mensajes cifrados a través del Monitor 1.

El algoritmo 1 describe el comportamiento del primer monitor. Se utiliza la herramienta *ent*[1] para comprobar si un mensaje está cifrado. Ent realiza pruebas estadísticas en la secuencia de bytes recibida como entrada, en concreto: entropía ($\mathcal{S}$), prueba de Chi-cuadrado (χ y χ_{err}), media aritmética (μ), valor Monte-Carlo para Pi (π) y coeficiente de correlación lineal (r). El primero, $\mathcal{S}$, es un indicador de la densidad de información del contenido del mensaje expresada como un número de bits por carácter. Los valores cercanos a 8.00 indican una alta entropía en los datos, como ocurre en los mensajes cifrados, ver Roman (1992). Por tanto, $\mathcal{S}$ es el mejor indicador para detectar mensajes cifrados, ver Clifford y Cosma (2013). Sin embargo, no funciona bien con mensajes cortos, como es el caso con la mayoría de los que se publican en ROS. Con mensajes de menos de 200 bytes, no hay datos suficientes para obtener un valor fiable para la entropía. En ese caso, tenemos que verificar el valor de μ, que es simplemente el resultado de sumar todos los bytes en el mensaje y dividir por la longitud del mensaje. Si los datos son casi aleatorios, μ se aproximará a 127.5. En nuestras pruebas, valores de μ superiores a 100 se obtienen solo para mensajes cifrados, lo que nos permite identificarlos inequívocamente. En los casos en que μ es inferior a 100, debemos considerar los resultados de la prueba Chi-cuadrado, comúnmente usada para verificar la aleatoriedad de los datos. Según nuestras pruebas, valores de χ mayores que 290 y de χ_{err} mayores del 10 % permiten identificar mensajes cifrados cuando μ es menor que 100.

Algorithm 1 Detección de mensajes cifrados

Input: *msg* ▷ Encrypted message
Output: *IsEncrypted* ▷ Boolean value
1: $\mathcal{S}, \chi, \chi_{err}, \mu, \pi \leftarrow$ ent *msg* ▷ Analyse *msg* with *ent*
2: **if** $\mu > 100$ **then**
3: **return** TRUE
4: **else**
5: **if** $\chi < 290$ **and** $\chi_{err} > 10\,\%$ **then**
6: **return** TRUE
7: **else**
8: **return** FALSE
9: **end if**
10: **end if**

2.2. *Monitor 2: detección de ataques de DoS*

La figura 3 representa el modelo conceptual para el intercambio de mensajes a través del topic */kio_rtls_talker/stdout*. El monitor 2 analiza los mensajes que el nodo *kio-rtls-talker* publica en este topic con las salidas del dispositivo KIO. *kio-rtls-listener* consume los mensajes publicados por *kio-rtls-talker*.

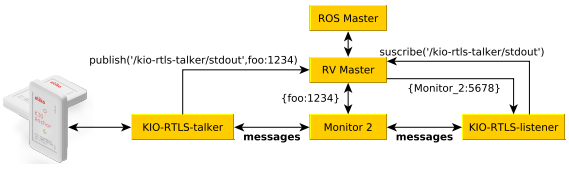

Figura 3: Ddetección de ataques de DoS a través del Monitor 2.

El listado 1 muestra datos recopilados por el transceptor móvil del dispositivo KIO, un RTLS basado en radiofrecuencia. Los valores de las columnas 2, 4, 6 y 8 muestran el identificador de las radio-balizas cuyos datos se han recibido. Las columnas 3, 4, 7 y 9 muestran la distancia desde estas balizas al transceptor móvil (anclado al robot). Las columnas 10 y 11 muestran el identificador del transceptor móvil y el timestamp de la medida. Las últimas tres columnas muestran las coordenadas 3D de la estimación de posición.

```
1 003c 408a 684 408b 351 501c 263 401d 659 4062 1214180
      00000000 00000000 00000000 00000000 3.40 1.34
    1.54
```

Listing 1: Data gathered by the KIO tag.

[1] http://www.fourmilab.ch/random/

Se puede detectar un ataque DoS cuando no se reciben datos de ninguna baliza. Cuando esto sucede, la etiqueta recibe "0000" como identificador de baliza y "0" como distancia al transceptor móvil. El listado 2 muestra los datos recopilados por el transceptor móvil cuando la baliza 408A (columnas 2 y 3) sufre una denegación de servicio.

```
003c 0000 0 408b 351 501c 265 401d 664 4062 6419526
     00000000 00000000 00000000 00000000 3.47 1.41
     1.57
```

Listing 2: Data gathered by the KIO tag when suffering a DoS attack.

El algoritmo 2 muestra el comportamiento del Monitor 2. Es capaz de detectar un ataque DoS, mediante el análisis de los datos obtenidos por el tranceptor móvil del dispositivo KIO.

Algorithm 2 Detección de ataques de DoS

Input: *msg* ▷ KIO RTLS message
Output: *DoS Detected* ▷ Boolean value
1: *regex* ← (.*␣0000␣0␣.*)
2: **if** *regex* **in** *msg* **then**
3: **return** TRUE
4: **else**
5: **return** FALSE
6: **end if**

3. Detección de ataques en sistemas de localización mediante algoritmos de aprendizaje autómatico

El objetivo principal de esta línea de trabajo, cuyos resultados se presentan en Guerrero-Higueras et al. (2018), es crear modelos que analicen las estimaciones de posición recibidas por un RTLS y detectar, si es el caso, dos tipos de ciber-ataques: DoS y Spoofing. Estos modelos permitirán crear, en futuros trabajos, un sistema de alerta que pueda proporcionar a los robots una instrucción prescriptiva para omitir las estimaciones de posición recibidas cuando se detecte un ciberataque.

La propuesta es similar a la presentada en Van Phuong et al. (2006), dónde se buscan anomalías en las mediciones recibidas por el transceptor móvil. En nuestro caso es un proceso centralizado, no distribuido, y no estamos analizando la secuencia de paquetes, sino su contenido. La primera pregunta específica que se presenta en este estudio es si sería posible detectar algunos tipos de ataques analizando solo las mediciones recibidas por el transceptor móvil a bordo del robot. Esto puede no ser fácil, ya que los errores de precisión de un RTLS podrían confundirse fácilmente con un ciber-ataque. Los métodos de aprendizaje supervisado son los más apropiados para este problema. Podemos construir un honeypot con datos de entrenamiento y test que contengan ejemplos explícitamente etiquetados para puntos conocidos y, a partir del mismo, crear modelos que generen predicciones para puntos que hayan sido visitados (validación). Dado que la clasificación está limitada a dos categorías ("ataque" y "no ataque") y debido a la naturaleza de las características elegidas, para evaluar nuestra propuesta hemos utilizando ocho clasificadores y algoritmos de predicción bien conocidos y comúnmente utilizados en problemas de clasificación y predicción: Adaptive Boosting (AB), Classification And Regression Tree (CART), K-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), Logistic Regression (LR), Multi-Layer Perceptron (MLP), Naive Bayes(NB) y Random Forest (RF).

La decisión final con respecto al mejor modelo para nuestro objetivo se basa en la evaluación de diferentes métricas: precisión con datos de test (capacidad de clasificar utilizando datos de test), precisión con datos de de validación (capacidad de clasificar con datos obtenidos en otros entornos) y matriz de confusión. El último permite medir la sensibilidad del modelo, que es esencial en un contexto de ciberseguridad debido a las consecuencias de los ataques clasificados erróneamente.

La tabla 1 muestra la precisión para los conjuntos de test y validación, considerando ataques de DoS. Los scores más altos para el conjunto de validación se resaltan en negrita. La tabla 2 muestra la precisión para los conjuntos de test y validación, considerando ataques de Spoofing. Los scores más altos para el conjunto de validación también se resaltan en negrita.

Tabla 1: Precisión para la detección de ataques DoS.

Classifier	Test score	Validation score
MLP	0.955357	**0.947559**
RF	0.984375	**0.766727**
KNN	0.966518	**0.742315**
CART	0.986607	0.478300
LR	0.964286	0.443038
LDA	0.975446	0.429476
NB	0.872768	0.429476
AB	0.979911	0.429476

Tabla 2: Precisión para la detección de ataques de Spoofing.

Classifier	Test score	Validation score
MLP	0.882222	**0.768260**
LR	0.848889	**0.663661**
LDA	0.846667	**0.587917**
NB	0.802222	0.567178
KNN	0.931111	0.537421
RF	0.977778	0.496844
CART	0.955556	0.486925
AB	0.928889	0.467087

4. Conjunto de datos de referencia para la evaluación de modelos de seguimiento de personas basados en LIDAR

Detectar y seguir personas es una habilidad muy útil para mejorar la navegación en robots móviles, o para facilitar robots socialmente más aceptables. Muchas soluciones en la literatura intentan resolver este problema utilizando un enfoque multimodal, típicamente mediante sensores de visión y distancia, como se ve en Arras et al. (2012). Los sensores de visión son más caros y por esta razón, los sistemas basados sólo en sensores de distancia son preferibles. Con respecto a los clasificadores para procesar los datos recopilados por diferentes sensores, las redes neuronales convolucionales se están convirtiendo en una solución muy popular según Long et al. (2015).

Los sensores LIDAR son sensores de alcance fiables y asequibles actualmente que. Los datos recopilados son fáciles de

procesar en tiempo real porque cada sondeo consiste en una matriz de solo unos pocos enteros. Por lo general, los robots móviles montan sensores LIDAR en una posición baja ($\sim 30 - 50$ cm desde el suelo) para detectar obstáculos. También se usan para construir mapas de ocupación y navegar. La información proporcionada permite estimar la distancia en ángulos precisos (resolución de 0.5 grados).

Diferentes objetos, como patas de mesas o sillas, troncos de plantas, etc. pueden confundirse fácilmente con las piernas de las personas. Las redes neuronales que permiten trabajar en estas situaciones requieren un buen conjunto de datos de entrenamiento. Recopilar y organizar un conjunto de entrenamiento completo requiere tiempo, así como conocimiento específico del dominio. Existe una gran colección de datasets disponibles para robots móviles, como el *Repository of Robotics and Computer Vision Datasets*.Sin embargo, la mayoría de estos datasets no son adecuados para el entrenamiento de redes neuronales. En Álvarez-Aparicio et al. (2018), presentamos un dataset de referencia que se puede utilizar para evaluar el rendimiento de diferentes enfoques para detectar y seguir personas mediante el uso de sensores LIDAR. La información contenida en este dataset es especialmente adecuada para su uso como datos de entrenamiento en clasificadores basados en redes neuronales.

Los datos contenidos en el dataset permiten evaluar dos sistemas de seguimiento de personas, ambos basados en redes neuronales: Leg Detector (LD), una solución ampliamente utilizada por la comunidad ROS; y una herramienta de seguimiento desarrollada por el Grupo de Robótica en la Universidad de León, conocida como PeTra. La figura 4 muestra los diferentes elementos y dispositicos utilizados para recopilar los datos, que incluyen: un área de estudio certificada conocida como *Leon@Home Testbed*[2], un robot asistente fabricado por Robotnik[3] (utilizado a modo de honeypot) con un sensor LIDAR incorporado y un dispositico KIO, un RTLS basado en radiofrecuencia utilizado para obtener datos de la verdad del terreno sobre la ubicación real de la/s persona/s. Los datos grabados incluyen estimaciones de posición LD y PeTra.

Los datos se han recopilado en diferentes situaciones donde una o más personas, llevando un transceptor móvil del dispositivo KIO, se mueven alrededor de Orbi-One. Estos escenarios se han elegido de acuerdo con diferentes situaciones que pueden darse en concursos de robótica, como la European Robotics League (ERL)[4] o la RoboCup[5].

5. Conclusión

La primera conclusión que se puede extraer del trabajo realizado en este proyecto hasta la fecha es que la preocupación por la seguridad de los sistemas robóticos está creciendo y hay razones para ello. Cada vez va a ser más frecuente encontrarse con robots autónomos fuera de los laboratorios de investigación y dichos dispositivos deben ser seguros en la interacción con los usuarios y cíber-seguros en su interacción con otros dispositivos, especialmente si están conectados a Internet.

(A) (B) (C)

(D) (E) (F)

Figura 4: Elmentos utilizados para la recopilación de datos. **A**: Robot Orbi-One. **B**: plano de Leon@Home Testbed. **C**: plano del apartamente. Los puntos rojos muestran la ubicación de las balizas del dispositivo KIO. Los puntos negros numerados muestran la posición de Orbi-One durante la recolección de datos. **D**: balizas del dispositivo KIO **E**: vista general del apartamento. **F**: Mobiliario del apartamento.

Un elemento primordial de la ciberseguridad de los robots de servicio son los frameworks de desarrollo. En el caso particular de ROS son bien conocidos sus problemas de seguridad, que se están tratando de resolver en la versión 2 con el cambio de TCPROS por DDS. Sin embargo, la disponibilidad de esta nueva versión parece lejana en el tiempo y además es probable que numerosos sistemas ya desplegados que no sean capaces de implementarla. Por ello, hemos propuesto soluciones alternativas basadas en cifrado y en la incorporación de reglas semánticas que permitirán mejorar la seguridad de los robots que usan actualmente ROS. Estos resultados se presentan en Balsa-Comerón et al. (2017).

Otra aportación realizada en este proyecto, y presentada en Guerrero-Higueras et al. (2018), es implementación de modelos de detección de ciberataques (DoS y Spoofing en concreto) en RTLS, muy utilizados en robótica móvil. Estos módelos, basados en aprendizaje automático pueden servir de base para la implementación de sistemas de seguridad más complejos que permitan a un robot detectar cuando está siendo víctima de un ataque y actuar en consecuencia.

Por último, una primera versión del conjunto de datos denominado "Range-based people tracker classifiers Benchmark Dataset"[6] (RRID:SCR _015743) ha sido liberada y presentada en Álvarez-Aparicio et al. (2018). Se puede acceder a los datos desde el sitio web del Grupo de Robótica.

English Summary

Machine learning applied to cybersecurity in autonomous robots

Abstract

The aim of this project is cybersecurity in autonomous robots. On the first place, a systematic literature review will focus on known robots' vulnerabilities. At the same time, a honeypot simulating a robot controlled by the standard

[2] http://robotica.unileon.es/index.php/Testbed

[3] http://www.robotnik.es/manipuladores-roboticos-moviles/rb-one/

[4] https://www.eu-robotics.net/robotics_league/

[5] http://www.robocup.org/

[6] https://scicrunch.org/browse/resources/SCR_015743

middleware for robotics development ROS (Robotic Operating System) will be created and deployed. A dataset will be obtained with the results from the honeypot operation. These data will be analysed by means of machine learning and data analysis methods. Taking the results obtained from the analysis as a starting point, new cyberattacks' detection and prevention algorithms will be defined. On the second place, but at the same time, new ROS components will be designed and developed to integrate cybersecurity systems in different types of robots. Three prototypes will be developed in order to validate our proposal. The first two ones will use two robots already owned by the research group: Baxter and RB1. Baxter is the most popular bi-manipulator robot for manufacturing environments because it allows collaboration with humans in production lines (Industry 4.0). RB1 is a service mobile manipulator manufactured by Robotnik, a Spanish company. For the third prototype, the proposal includes a request to acquire an android called Pepper, manufactured by Aldebaran, in order to cover social interaction and complement the rest of the robots already owned by the group.

Keywords:

Cybersecurity robotics autonomous systems machine learning location systems.

Agradecimientos

Este artículo ha sido parcialmente financiado por la Junta de Castilla y León mediante el proyecto LE028P17, con participación de la Unión Europea a través de los fondos FEDER. Los autores agradecen igualmente al Instituto Nacional de ciberseguridad (INCIBE) el apoyo prestado a esta línea de investigación mediante la Adenda 21 çiberseguridad en sistemas autónomos.ªl convenio marco con la Universidad de León.

Referencias

Álvarez-Aparicio, C., Guerrero-Higueras, Á. M., Olivera, M. C. C., Rodríguez-Lera, F. J., Martín, F., Matellán, V., 2018. Benchmark dataset for evaluation of range-based people tracker classifiers in mobile robots. Frontiers in neurorobotics 11, 72.

Arras, K. O., Lau, B., Grzonka, S., Luber, M., Mozos, O. M., Meyer, D., Burgard, W., 2012. Towards Service Robots for Everyday Environments. STAR 76. Springer, Ch. Range-Based People Detection and Tracking for Socially Enabled Service Robots, pp. 235–280.

Balsa-Comerón, J., Guerrero-Higueras, Á. M., Rodríguez-Lera, F. J., Fernández-Llamas, C., Matellán-Olivera, V., 2017. Cybersecurity in autonomous systems: Hardening ros using encrypted communications and semantic rules. En: Iberian Robotics conference. Springer, pp. 67–78.

Baykara, M., Das, R., 2015. A survey on potential applications of honeypot technology in intrusion detection systems. International Journal of Computer Networks and Applications 2 (5), 1–9.

Bonaci, T., Chizeck, H. J., 2012. On potential security threats against rescue robotic systems. En: Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on. IEEE, pp. 1–2.

Bonaci, T., Herron, J., Yusuf, T., Yan, J., Kohno, T., Chizeck, H. J., 2015. To make a robot secure: An experimental analysis of cyber security threats against teleoperated surgical robots. arXiv preprint arXiv:1504.04339.

Clifford, P., Cosma, I., 2013. A simple sketching algorithm for entropy estimation over streaming data. En: AISTATS. pp. 196–206.

DeHon, A., Karel, B., Knight Jr, T. F., Malecha, G., Montagu, B., Morisset, R., Morrisett, G., Pierce, B. C., Pollack, R., Ray, S., et al., 2011. Preliminary design of the safe platform. En: Proceedings of the 6th Workshop on Programming Languages and Operating Systems. ACM, p. 4.

Durairajan, M., Saravanan, R., Chakkaravarthy, S. S., 2016. Low interaction honeypot: A defense against cyber attacks. Journal of Computational and Theoretical Nanoscience 13 (8), 5446–5453.

Fosch-Villaronga, E., 2015. Creation of a care robot impact assessment. En: XVII International Conference on Social Robotics.

Guarnizo, J. D., Tambe, A., Bhunia, S. S., Ochoa, M., Tippenhauer, N. O., Shabtai, A., Elovici, Y., 2017. Siphon: Towards scalable high-interaction physical honeypots. En: Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security. ACM, pp. 57–68.

Guerrero-Higueras, Á. M., DeCastro-García, N., Matellán, V., 2018. Detection of cyber-attacks to indoor real time localization systems for autonomous robots. Robotics and Autonomous Systems 99, 75–83.

Guerrero-Higueras, Á. M., DeCastro-García, N., Rodríguez-Lera, F. J., Matellán, V., 2017. Empirical analysis of cyber-attacks to an indoor real time localization system for autonomous robots. Computers & Security 70, 422–435.

Heelan, S., 2011. Vulnerability detection systems: Think cyborg, not robot. IEEE Security & Privacy 9 (3), 74–77.

Huang, J., Erdogan, C., Zhang, Y., et al., 2014a. ROSRV: Runtime verification for robots. En: International Conference on Runtime Verification. Springer, pp. 247–254.

Huang, J., Erdogan, C., Zhang, Y., Moore, B., Luo, Q., Sundaresan, A., Rosu, G., 2014b. Rosrv: Runtime verification for robots. En: International Conference on Runtime Verification. Springer, pp. 247–254.

Irvene, C., Formby, D., Litchfield, S., Beyah, R., 2018. Honeybot: A honeypot for robotic systems. Proceedings of the IEEE 106 (1), 61–70.

Lera, F. J. R., Balsa, J., Casado, F., Fernández, C., Rico, F. M., Matellán, V., 2016. Cybersecurity in autonomous systems: Evaluating the performance of hardening ros. Málaga, Spain, 47.

Lera, F. J. R., Llamas, C. F., Guerrero, Á. M., Olivera, V. M., 2017. Cybersecurity of robotics and autonomous systems: Privacy and safety. En: Robotics-Legal, Ethical and Socioeconomic Impacts. InTech.

Litchfield, S., Formby, D., Rogers, J., Meliopoulos, S., Beyah, R., 2016. Rethinking the honeypot for cyber-physical systems. IEEE Internet Computing 20 (5), 9–17.

Long, J., Shelhamer, E., Darrell, T., Jun. 2015. Fully convolutional networks for semantic segmentation. En: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3431–3440.
DOI: 10.1109/CVPR.2015.7298965

McClean, J., Stull, C., Farrar, C., Mascareñas, D., 2013. A preliminary cyber-physical security assessment of the robot operating system (ros). En: Unmanned Systems Technology XV. Vol. 8741. International Society for Optics and Photonics, p. 874110.

Morante, S., Victores, J. G., Balaguer, C., 2015. Cryptobotics: Why robots need cyber safety. Frontiers in Robotics and AI 2, 23.

Mouratidis, H., 2004. A security oriented approach in the development of multiagent systems: applied to the management of the health and social care needs of older people in england. Ph.D. thesis, University of Sheffield.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., 2009. Ros: an open-source robot operating system. En: ICRA workshop on open source software. Vol. 3. Kobe, Japan, p. 5.

Roman, S., 1992. Coding and information theory. Vol. 134. Springer Science & Business Media.

Roosa, J., Decker, L., 2013. Security risks of utilizing robotics and medical devices in the medical profession. Viitattu 25, 2017.

Smid, M. E., Branstad, D. K., 1988. Data encryption standard: past and future. Proceedings of the IEEE 76 (5), 550–559.

Su, K. L., Chien, T. L., Guo, J. H., 2004. Design a low cost security robot applying in family. En: International Conference on Autonomous Robots and Agents. pp. 367–372.

Van Phuong, T., Hung, L. X., Cho, S. J., Lee, Y.-K., Lee, S., 2006. An anomaly detection algorithm for detecting attacks in wireless sensor networks. En: International Conference on Intelligence and Security Informatics. Springer, pp. 735–736.

Zhang, F., Zhou, S., Qin, Z., Liu, J., 2003. Honeypot: a supplemented active defense system for network security. En: Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on. IEEE, pp. 231–235.