ICRA 2020 Pre-Conference Half-Day Workshop on Security and Privacy in Robotics

Sunday 31 May 2020 - Sunday 31 May 2020

Paris (France)

Book of Abstracts

Contents

2

A Dynamic Game Framework for Robot Deception with an Application to Deceptive Pursuit-Evasion

Authors: Linan Huang[1]; Quanyan Zhu[2]

[1] *NYU*

[2] *Tandon School of Engineering, New York University*

Corresponding Authors: quanyan.zhu@nyu.edu, lh2328@nyu.edu

Recent advances in automation and adaptive control strategies in multi-agent systems enable robots to use deception to accomplish their objectives. We study rational and persistent deception among intelligent robots to enhance the security and operation efficiency of autonomous vehicles. We present an N-person K-stage nonzero-sum game with an asymmetric information structure where each robot's private information is modeled as a random variable or its type. The deception is persistent as each robot's private type remains unknown to other robots for all stages. The deception is rational as robots aim to achieve their deception goals at minimum cost. Each robot forms a belief on others' types based on state observations and updates it via the Bayesian rule. The level-t perfect Bayesian Nash equilibrium is a natural solution concept of the dynamic game. It demonstrates the sequential rationality of the agents, maintains the belief consistency with the observations and strategies, and provides a reliable prediction of the outcome of the deception game. In particular, in the linear-quadratic setting, we derive a set of extended Riccati equations, obtain the explicit form of the affine state-feedback control, and develop an online computational algorithm. We define the concepts of deceivability and the price of deception to evaluate the strategy design and assess the deception outcome.

The proposed model has wide applications including cooperative robots, pursuit and evasion, and human-robot teaming. The pursuit-evasion games are used as case studies where the evader aims to deceptively reach the target and the pursuer keeps her maneuverability as private information. The pursuer has the lowest cumulative cost under the proposed policy than the direct-following and conservative policies. We have proposed multi-dimensional metrics such as the stage of truth revelation, the endpoint distance, and the cumulative cost to measure the deception impact throughout stages. We have concluded that Bayesian learning can largely reduce the impact of initial belief manipulation and sometimes result in a win-win situation. The increase of the pursuer's maneuverability can also reduce the endpoint distance and her cumulative cost yet has a marginal effect. A robot is more deceivable, i.e., less learnable when his/her potential types are less distinguishable. Finally, we have found that the idea of using deception to counter deception is not always effective. In particular, it is beneficial for the low-maneuverability pursuer to disguise as a high-maneuverability pursuer but not vice versa.

3

Modeling a Honeypot Architecture for a Robotic Scenario

Author: Francisco J. Rodriguez Lera[1]

Co-authors: Angel Manuel Guerrero Higueras [1]; Camino Fernández Llamas [1]; Vicente Matellán Olivera [1]

[1] *Universidad de León*

Corresponding Authors: fjrodl@unileon.es, am.guerrero@unileon.es, vmato@unileon.es, cferll@unileon.es

There is a massive number of honeypot frameworks focused on detection and information gathering from attackers that believe they are working on authentic machines. However, there is a lack of honeypots devoted to robotics environments, and it is not easy to identify cybercriminals' behaviors once they have control over a robotic platform.

This research proposes a passive honeypot architecture that aims to monitor a set of simulated and real robotics platforms running under supervised environments. The architecture is inspired in current honeypots for critical systems and provides information about robot and attacker behavior for cybersecurity researchers. This abstract overviews our prof-of-concept, its operative modes, and the monitorization system.

Two pages abstract: https://drive.google.com/file/d/1L-lXefbo8lplOQSX_MrjR-VNpDhBuIMe/view?usp=sharing

I. INTRODUCTION
\section{Introduction}

The aim of a honeypot is to be compromised, attacked and invaded by cybercriminals and other malicious users. Under the appearance of a real system, a honeypot is a
security resource, that provides different sources of information for the security researcher. Using different monitoring mechanism and analyzing attacker behavior it is possible to understand which parts of the system that can be compromised.

This approach, which is quite fruitful in common computer systems, deals with loss of information and system compromise. However, when thinking in a robotic platform, we add two dimensions: 1) the physical effects on the robot, it is expected to interact not only with data but also with real environment elements such as humans; 2) the high cost that characterizes a robotic platform.

This research proposes a honeypot architecture that aims to monitor a set of simulated and real robotics platforms running under supervised environments.

The design and deployment of honeypots is been running in security research during the last 20 years. It is possible to classify honeypots (hps) attending to four different characteristics~\cite{nawrocki2016survey}: field of operation (fully realistic or bounded), direction of interaction, interaction level, and physicality.

The remainder of this abstract provides an insight of the proposed architecture and discusses future research directions we identified along with a concluding remarks.

\label{sec:architecture}

This section presents the framework proposed for a robotic honeypot. It is mainly divided in two sections: Honeypot Core System (HoCoSys) and the Monitoring System (MonSys). The monitoring system is involved in the the study and analysis of network and system behavior. The HoCoSys is the element that reassembles the robot platform. It presents four different approaches for a robotic honeypot and it is supported on other state of the art works~\cite{irvene2019evaluating}, \cite{irvene2017honeybot}, \cite{jicha2016scada}, \cite{baykara2018novel}.

\subsection{Honeypot Core System (HoCoSys)}

This research proposes four types of Honeypot Core System attending the manner in which the robot honeypot is deployed (Fig.~\ref{sec:architecture} illustrates them): A) HoCoSys-Simulated, it provides a robot environment supported on simulation, B) HoCoSys-Real, an environment supported on real robots; C) HoCoSys-Emulated, it provides an environment supported on information from real sensors played on simulated environment; and D) HoCoSys-Service, a lightweight access to a simulated or real robot. \\

\textit{A) HoCoSys-Simulated} It serves a full machine with a robotic environment as those used in development and research scenarios. The system mirrors the a approach of a remote control machine deployed for an employee. This user will navigate the directory tree freely and will be able to access to run the simulator or other components installed.

\textit{B) HoCoSys-Real} This method provides fully access to a robot. The system offers all robot functionalities on top of it. However, some of the robot functionalities are blocked and appears with issues.

\textit{C) HoCoSys-Emulated} It serves a full machine with a robotic environment as those used in development and research scenarios, in the same manner that simulated but offering information gathered from the real sensor.

\textit{D) HoCoSys-Service} This approach provides a set of predefined services ready for being used by users. This approach uses a webservice of teleoperation and sensor discovery. The service will

provide information about the robot status, and will show real or semi-real information. This type of component corresponds to the classic low-interaction honeypot

B. Monitoring System
An attacker exploiting a robot has several points of incursion among hardware elements, software pieces or at network level exploiting traditional existing implementation vulnerabilities. This scenario suggests a three level approach for detect, defend and inform about the attack situation and attacker behavior.

\subsubsection{Network Tracking}

Connections that have exchanged lots of information are potentially more valuable for detecting matches with new traffic. The system must prevent aggressive port scans from overflowing the connection hash tables which would cause the valuable connections to be dropped.

The system will be ready for accepting any outside connection. It will limit the number of open different connections. We currently perform protocol analysis at the network layer and transport layers for IP, TCP and UDP packet headers.

\subsubsection{System Performance Tracking}

The system needs to provide a set of mechanisms for tracking the system performance. This is monitoring changes in components such as network, CPU, RAM, and the file system volume.
This information is used to model the regular system behavior, allowing us the selection individual decision in the honeypot system.

\subsubsection{Robot Hardware Tracking}

It is necessary to introduce a monitor system supported on the robot middleware. This system aims to record those log messages from all hardware sensors or software nodes which are present in the robotic platform. The log messages should be presented in a human-readable way and should be able to apply minimal accountability process.

\subsubsection{User Interaction Tracking}
This approach provides a mechanism that hides a command line monitor system that reads, labels and stores every command that a user writes in the terminal.

\section{Conclusions}

This abstract introduced the first iteration of a novel architecture for a robotic honeypot. A honeypot for a robotic scenario will allow security researchers to understand the users behavior in this kind of platform.

Supported on two components, the Honeypot Core System and the Monitoring System, our framework presents four different honeypot models for deploying in real scenarios. Each model introduces a range of interesting advantages and disadvantages. Within the former we can find flexibility, containment, focus on particular services and system control, however, we find disadvantages such as limited field of view, limited platform information, or being fingerprinted.

Future research directions point out to how to avoid being fingerprinted when the robot performance is fully emulated and how to allow phisics reusability between robotic honeypots. There are also a range of issues attending the hardware and software involved. Although ROS is the \textit{de facto} standard for most newer platforms, there are many software alternatives for robot control, not only by component but also by middleware and it is hard to replicate a realistic honeypot scenario between similar robots using different software.

REFERENCES
[1] Muhammet Baykara and Resul Das. A novel honeypot based security approach for real-time intrusion detection and prevention systems. Journal of Information Security and Applications, 41:103–116, 2018.
[2] Celine Irvene, David Formby, and Raheem Beyah. On evaluating the effectiveness of the honeybot: A case study. arXiv preprint arXiv:1905.12061, 2019.
[3] Celine Irvene, David Formby, Samuel Litchfield, and Raheem Beyah. Honeybot: A honeypot for robotic systems. Proceedings of the IEEE, 106(1):61–70, 2017.
[4] Arthur Jicha, Mark Patton, and Hsinchun Chen. Scada honeypots: An

in-depth analysis of conpot. In 2016 IEEE conference on intelligence and security informatics (ISI), pages 196–198. IEEE, 2016.
[5] Marcin Nawrocki, Matthias Wählisch, Thomas C Schmidt, Christian Keil, and Jochen Schönfelder. A survey on honeypot software and data analysis. arXiv preprint arXiv:1608.06249, 2016.