Message Authentication Codes for Secure Remote Non-Native Client Connections to ROS Enabled Robots

Russell Toris, Craig Shue, and Sonia Chernova
TePRA 2014
Putting Robots Online

- Remote Teleportation of Robots
- Exposure to broader user-base
  - Expert & Non-Expert Users
Tools and Resources

- ROS (Robot Operating System) tools
  - rosbridge (WebSockets and JSON)
  - roslibjs and ros3djs
  - RMS (Robot Management System)
Online Robots via RMS

The project consists of several virtual and physical worlds that allow access to online robots. This project is designed to address several critical drawbacks in traditional human-robot interaction (HRI) user studies with automatically recruited participants, which are expensive, time consuming, and confirmatory. The round-the-clock testing facility enables continuous user interaction and feedback to enhance the development and evaluation of robots with users of varying backgrounds and levels of expertise.

More information can be found at the project's website.

The project is funded by grants from the National Science Foundation and DARPA.

Further details about the project can be found on the website.
Common Use Case

- Non-expert User Study Testing
  1. User logs into website
  2. WebSocket connection made to robot using *rosbridge* protocol
  3. User interacts with robot
Common Use Case

- Non-expert User Study Testing
  1. User logs into website
  2. **WebSocket connection made to robot using *rosbridge* protocol**
  3. User interacts with robot
Problem

- `rosbridge` server listens to *any* connection
- Allows full access to all robot functions
- Common view - security through obscurity
  - Does not hold anymore!
- Potential harm to robot and its surroundings
Current “Solution”

- Use a VPN
Current “Solution”

- Use a VPN – Does this make sense anymore?
Web Robotics simply cannot continue to ignore this problem

Dangerous to robots, environments, and human researchers
Can we adopt ideas from the security community to form an efficient, simple, and secure procedure and keep the current benefits?
The Use of MACs

MACs – Message Authentication Codes
  ◦ IPSec and SSL

Main Idea:
  ◦ Client $C$ sending verified message $m$ to server $S$
  ◦ $C$ and $S$ both know secret key $k$ and hashing function $hash()$
  ◦ {
    mac : hash(k+m)
    message : m
  }
**rosauth**

- External authenticator provides MAC to remote, non-native clients
- Client passes information back to ROS server
- ROS verifies the information and accepts or rejects the connection
rosauth Token Fields

- **client** (string): The client string contains the IP of the client where this message originated.
**rosauth Token Fields**

- **client**
- **dest (string)**: The destination string contains the IP or host of the server the client is trying to connect to.
**rosauth Token Fields**

- **client**
- **dest**
- **rand** (string): A random string is added to the hash as a nonce to prevent replay attacks and cookie stealing, and allow for multiplexing.
rosauth Token Fields

- client
- dest
- rand
- t (int): A count of seconds since the start of the Unix epoch is given, indicating the time the original MAC was created.
rosauth Token Fields

- client
- dest
- rand
- t
- level (string): A user level string is provided to state what level of user is connected (e.g., admin).
**rosauth Token Fields**

- **client**
- **dest**
- **rand**
- **t**
- **level**
- **end (int)**: An end time in seconds is given stating how long the client is authorized to remain connected.
rosauth MAC

- \( MAC = \text{hash}(\text{key} + \text{token\_fields}) \)
- SHA-2 algorithm used (SHA512 used in practice)
- Connection verified once, not per-message
- **Important assumption - SSL used!**
Overview

1: procedure CHECKAUTHENTICATION(mac, data)
2:   if sha512(key + data) is not mac then
3:     return False
4:   else if data['client'] is not socket’s client IP then
5:     return False
6:   else if data['host'] is not server’s IP then
7:     return False
8:   else if data['t'] is not current time ±δ then
9:     return False
10:  else if data['end'] ≤ current time then
11:    return False
12:  else
13:    return True
14:  end if
15: end procedure
Applications and Source

- RMS – Robot Management System
  - External authenticator
- roslibjs
  - Non-native client library
- rosbridge
  - Authorization protocol & server implementation

http://robotwebtools.org/
Q&A