RTDS-RSCAD LOGIC BOMB: A PROOF OF CONCEPT

Daniele Ricciardelli
Department of Computer Engineering, California State Polytechnic University, Pomona
Advisor: Dr. Sean Monemi

Abstract

This project demonstrates the potential cybersecurity risks posed to engineers working on smart grid systems, specifically those using the Real-Time Digital Simulator (RTDS) and its associated RSCAD software at Southern California Edison. We developed a script capable of subtly manipulating the simulation data within an RTDS project without the user's awareness.

By introducing undetected modifications to a simulation's initial conditions, this exploit creates a plausible scenario in which a smart grid configuration is compromised before deployment, potentially resulting in widespread operational failures and significant financial losses.

This work highlights the need for stronger validation and monitoring tools within simulation environments to safeguard critical infrastructure.

Introduction

The objective of this project is to raise awareness among the electrical engineering team at Southern California Edison (SCE) about a potential cybersecurity threat that may exist within their work environment. Through informal interviews with students currently interning at SCE, we found that many have access to critical RTDS machinery and simulation data with minimal security protocols in place to prevent cyberattacks originating from infected machines.

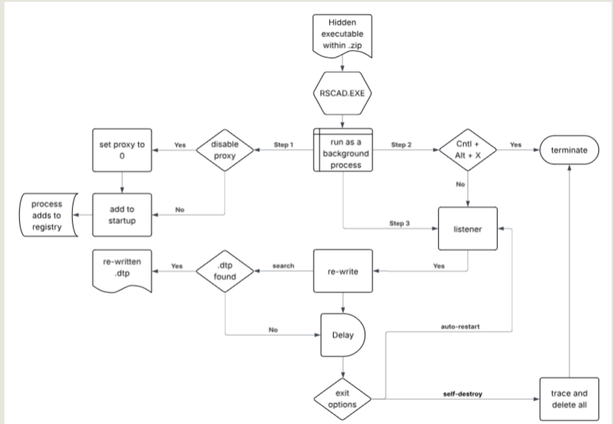
If such access remains unregulated, the integrity of smart grid demo simulations, especially those intended for real-world deployment – could be compromised. This presents a serious risk not only to SCE's operations and financial stability, but also to the safety of the public and the reliability of the state's infrastructure.

While our project centers on a self-replicating logic bomb embedded within an infected device, it serves as a proof-of-concept for how even a simple script, created by students with limited cybersecurity experience, could potentially bypass safeguards. If a similar attack were to be executed by a more advanced threat actor, the consequences could be far more severe.

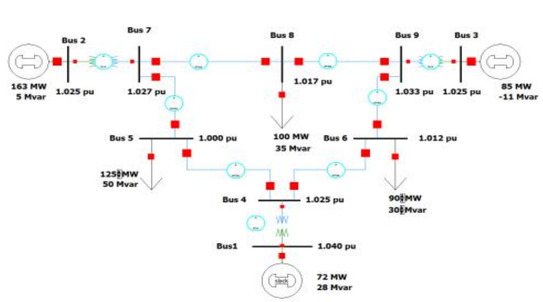
Script Features

- Stealth Operation: Functions as a fileless process by embedding into system memory and registering in system startup for persistence.
- Adaptive Behavior: If terminated, it reappears under a different name, icon, and location
- Hardware Monitoring: Activates upon detecting RTDS connections via Windows' device manager protocols.
- Targeted File Manipulation: Searches for .dtp files, which are used in RSCAD to store simulation setups. Injects subtle, damaging changes to simulate faults.
- Resilient Execution: Survives up to two manual termination attempts by relaunching itself from memory.
- Secure Removal: Can only be fully stopped using memory inspection, antivirus tools, or a specific keyboard shortcut (for testing).

Flow Diagram



RSCAD



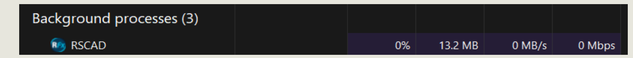
RSCAD is the software platform used to build and run real-time power system simulations on the RTDS hardware. One of the most essential files generated during a project is the .dtp file, which stores the schematic layout, configuration of components, and key simulation parameters. These files define how the system operates and responds under different conditions. If a .dtp file is corrupted or subtly manipulated, the simulated behavior may no longer reflect the intended design, which could potentially cause engineers to approve flawed smart grid configurations. Such undetected tampering could result in operational failures, financial losses, or widespread system vulnerabilities once deployed. This highlights the critical importance of securing simulation files and maintaining the integrity of RSCAD environments.

RTDS

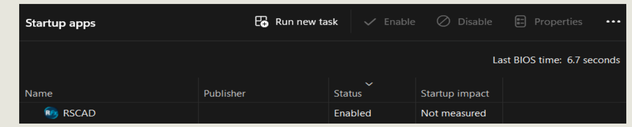
The Real-Time Digital Simulator (RTDS) is a specialized hardware platform used for modeling and testing power systems in real time. Its companion software, RSCAD, provides the graphical interface for designing and executing these simulations. Together, they enable engineers to analyze grid behavior, test protection schemes, and develop control strategies with high fidelity. The RTDS system supports both digital and analog I/O, allowing for hardware-in-the-loop (HIL) testing with physical components. However, due to its real-time architecture, manipulating data during an active simulation presents a significant challenge. Additionally, remote live-attacks are impractical, as the RTDS system requires a direct Ethernet connection. Once a user connects to the RTDS, that computer is typically isolated from the internet, including through Wi-Fi, preventing further remote access during operation.

Results

Once the script is tested, the first flag we will obtain will be a background process as shown in the task manager:



Following it up, a startup app will be automatically enabled and implemented



Which can be seen and modified only manually through the Registry Editor

Once the computer is corrupted and the script activated, it starts looking for .dtp files. In this case it will look for files that have the most basic data that all .dtp files will have, a voltage base, also denoted in those files as "VBASE".

```
SECTION: NODE_SECTION
   NUMBER_OF_NODES= 39
   INITIAL_NODE_VOLTAGES= 0.0
      NODE= 1 VBASE= 345 IBASE= 1.0 NODENAME= "VBSSNOut"
      NODE= 2 VBASE= 345 IBASE= 1.0 NODENAME= "N10"
      NODE= 3 VBASE= 345 IBASE= 1.0 NODENAME= "N11"
      NODE= 4 VBASE= 345 IBASE= 1.0 NODENAME= "N12"
      NODE= 5 VBASE= 120 IBASE= 1.0 NODENAME= "WFAVAOut"
      NODE= 6 VBASE= 345 IBASE= 1.0 NODENAME= "WFBVAOut"
      NODE= 7 VBASE= 240 IBASE= 1.0 NODENAME= "N25"
      NODE= 8 VBASE= 345 IBASE= 1.0 NODENAME= "N13"
      NODE= 9 VBASE= 120 IBASE= 1.0 NODENAME= "WFAVBOut"
      NODE= 10 VBASE= 345 IBASE= 1.0 NODENAME= "WFBVBOut"
```

In order to keep it simple, to assure RSCAD FX will compile, and to don't make any major change to prevent the infected user to notice the script, it will change a random node, adding voltage, in this case we chose to add a random value of 60V → `NODE= 5 VBASE= 180 IBASE= 1.0 NODENAME= "WFAVAOut"`

After modifying the target file, the script returns to a dormant state, remaining inactive until it detects another RTDS communication event via the Ethernet port. For testing and demonstration purposes, two additional operational modes were implemented. The first is a manual shutdown feature that allows the script to be terminated by a specific keyboard combination (Ctrl + Alt + X). The second is a self-destruct mode, which initiates a cleanup routine that retraces the infection steps and removes all traces of the script using PowerShell commands. This deletion process bypasses the Recycle Bin, ensuring no visible remnants remain on the system—making post-infection analysis by casual users extremely difficult.

Although the script was developed using only foundational coding skills and a basic understanding of OS libraries, it highlights a serious vulnerability in environments where background processes and new installations go unchecked. One of the most concerning aspects is its potential for propagation through simple social engineering tactics, such as bundling the script with a seemingly harmless RTDS project file. This underscores how easily critical data can be compromised when proper cybersecurity measures are not enforced.