Improving Computer System Integrity Resilience With Agent-managed Decoupled Components

William Horsthemke
Argonne National Laboratory

Nathanial Evans
Argonne National Laboratory

Dan Harkness
Argonne National Laboratory

ABSTRACT

We depend upon continuous operation of trustworthy, resilient computer systems, but ensuring their integrity and availability poses substantial challenges. These challenges come from ever-increasing cyber attacks by dedicated, sophisticated adversaries as well as unintentional faults caused by the complexity of the software and hardware operating these systems.

This paper proposes reducing the complexity of traditional multi-function computer systems by separating the functions and deploying those functions on simpler, dedicated-function components. Simple dedicated-function components are easier to verify and replace. To ensure high-availability, this paper proposes to deploy multiple instances of each component in redundant arrays, and use autonomous agents to continuously verify their integrity and availability and replace them as necessary.

The motivation for decoupling complex, multi-function systems into dedicated-function components appeared in two fields: nuclear verification and electricity grid protection. The nuclear verification community wants trustworthy measurement systems than can be confidently verified to ensure that they correctly perform their functions and only the required functions. The complexity of systems increases the opportunities for adversaries to tamper with the system and decreases the ability of inspectors to verify the integrity of the system. To decrease overall system complexity, researchers are developing modular systems and deploying each function on a dedicated component and interconnecting them with function-specific communication. These dedicated-function components permit inspectors to independently test and verify each functional component and its interfaces, before inspecting and verifying the combined system. This builds mutual trust in the verification process.

The manufacturer of protective relays used to protect the electricity grid proposes a similar approach. They attribute many of the faults of their relays to the complexity of multi-function, integrated systems, where a fault in a support function cascades into a failure of their control system. They observe that the software required to perform support functions becomes more complex and requires more frequent updates than the software used for control and protective functions. They propose separating their system into dedicated function components so that faults in one function do not cause faults in other functions.

INTRODUCTION

Recent research and development efforts have focused on reducing the complexity of traditional multi-function computer systems by separating the functions into individual dedicated-function components. This effort has two related goals: decrease risk and improve trust in component and system verification. The control system community aims to decrease the risk that the control function suffers from a fault or compromise in support functions. When the control function operates on the same device as the support functions, a problem with a support function can cause a problem with the control function.

The nuclear verification community aims to increase trust in the computer systems used to measure nuclear material and support the verification of treaty compliance. This requires inspecting a multi-function computer system to ensure that it performs as expected and only as expected. By separating functions onto dedicated-function components, the community expects to create a system that is easier to inspect and test. This design should foster mutual trust in the computer system used to verify nuclear treaty compliance.

In this paper, we focus on control-system design. Control systems require continuously available high-integrity operations. We propose to deploy redundant arrays of components to provide a pool of operationally ready standby components and use autonomous agents to verify and replace components which fail integrity or availability tests. Agents will actively manage all aspects of the control system, including the components, networks, security, and situational awareness. The use of autonomous agents will increase the overall resilience and help maintain the integrity and availability of the control system.

In the next section, we discuss recent research and development on reducing complexity and risk. First, we introduce the goal of decoupling complex, multi-function devices into separate dedicated-function components which are connected together using function-specific interfaces. Second, we describe how to improve the resilience of these systems by deploying them in redundant arrays. Third, we provide an overview of autonomous agents. Fourth, we describe how autonomous agents can improve the resilience of these systems by managing their integrity and availability.

DECOUPLING MULTI-FUNCTION DEVICES TO REDUCE COMPLEXITY AND RISK

With the goal of reducing risk and improving trust, research has started to focus on reducing the complexity of systems by decoupling multi-function devices into separate dedicated-function components with function-specific interfaces between components. This effort emerges from both the control-system community and the radiation-detection community who design verification systems that require trust by nuclear safeguards inspectors.

The control system proposal comes from a major provider of protective relays which trip circuit breakers to protect electricity transmission and generation equipment from faults and other abnormal operating conditions (Edmund O. Schweitzer 2020). The protective relay example identifies three functions for its control system: protection, automation, and communication. The

goal of their design is to ensure that the protection function can reliably operate even if faults occur in the automation and communication functions.

The radiation detection system proposal comes from the research community who develop radiation detection systems to support the verification of international treaties on nuclear material. Treaty verification requires the use of mutually-accepted radiation detection systems. Acceptance depends upon verifying and trusting that the radiation detection system functions as designed and only as designed. Recent research proposes a modular, open-architecture approach to developing transparent, easy-to-test components and well-defined interconnections (Polack 2020). The goal of this approach is to simplify the inspection and testing process to increase the mutual trust of the proposed system.

The overall goal of both communities aims to reduce the complexity by creating simplified components dedicated to specific functions. By performing a specific function only, the components can use simpler hardware, software, and interfaces. By composing a system using simple components, connected with well-defined, functional-specific protocols; the constructed system will be easier to verify, more trustworthy, more reliable, easier to protect, and, perhaps, less vulnerable to compromise.

Nuclear Verification System

One of the goals of the nuclear verification community is to create a mutually trustworthy radiation-detection system. To build mutual trust in the system, the community needs to inspect and test the system to verify that the system performs its role and only its role. The proposed design (Polack 2020) aims to create a transparent system that is easy to inspect, test, and trust. Traditional, multi-function systems are more complex and difficult to comprehensively inspect and test. This complexity presents challenges to authenticating and certifying that the equipment performs as expected and only as expected and has not been modified since it was last evaluated. Inspection challenges reduce trust in the verification process. To facilitate inspection and testing, Polack identified independent functions and built dedicated hardware and software modules to perform those functions and function-specific communication interfaces to communicate between functions. Each module contains all the necessary controls to operate independently, without the need for communication from other modules. This self-contained modularity permits inspectors to independently test and verify each functional component and its interfaces, before inspecting and verifying the combined system. This builds trust in the verification process. In addition, the verification community wants flexibility to adapt to changes in verification needs, such as what and how to measure. By using a modular design, the system can be transparently composed and verified.

Control System

The goal for the control system community is to decrease the risk of device failure and enable in-service maintenance of support functions, such as automation and communication, without requiring downtime (Edmund O. Schweitzer 2020). To decrease the risk of control failures, the

design decouples the control function to allow it to operate independently from the other functions and enables the control function to continue to operate even if other functions fail. To decrease downtime, the automation and communication functions can be shut down for software updates, without shutting down the control function. This decreases downtime, because the automation and communication functions require more frequent updates than the control function. More frequent updates are required because these functions implement evolving standards for industrial control system protocols. Implementing these standards requires software modifications that can increase the risk of software errors that cause component failures or introduce security vulnerabilities. The decoupled design helps protect the control function against cyber attacks. The design uses a sequential network: from communication to automation to control. This layered design provides defense-in-depth protection as well as protection against denial-of-service attacks against the innermost-layer where the control function resides.

Protection Against Compromise of the Control Function

The physical separation of functional components, the simplicity of their hardware and software, and the function-specific communication interface between components should reduce the risk of compromise of the control function. By physically separating functions onto different components, faults and compromises are prevented from propagating within the system as can happen within a multifunction system. If each component uses different hardware or software designs, they are less likely to suffer from common vulnerabilities. A compromise affecting the external facing communication component would need to traverse the network to the automation component then to the control component. If each component has a different design, the same compromise is less likely to compromise all components.

By designing function-specific interfaces, protective measures can restrict the ability to propagate between connected components. To reach the control component, a compromise of the external-facing communication component would need to exploit its interface to the automation component, the automation component itself, the automation interface to the control component, and finally the control component. By designing control-function components that can operate independently, protective measures can isolate the control function if faults or compromises are detected on the other components.

The design of a protective measure against control-function compromise could develop an integrity-based health check on the support functions and send the health check on the interface to the control function. The health check could include verification of the integrity of various software-based elements, including the firmware, operating system, application, and application settings.

If a valid health check was not received as expected, the control function could isolate itself and operate independently, perhaps by reloading a trusted software system, or perform an orderly/graceful degradation. A similar health check could improve protection against compromise from other interconnected components. For example, the automation function could receive a health check from the communication function and perform self-protective measures if the health check fails. Note: although the design offers substantial protection against attacks that compromise

components to gain access to assert control or otherwise damage the systems, the design of the system remains vulnerable to attacks that can exploit the support components and cause them to send invalid instructions and commands through the connected system which might violate the integrity of the operation of the control function. The design of the health and status of the entire system needs to incorporate methods to verify the operational integrity of each component.

DEPLOYING REDUNDANT ARRAYS OF COMPONENTS TO IMPROVE RESILIENCE

To create high-integrity, high-availability control systems from dedicated-function components, sets of components can be deployed in redundant arrays. For example, a protective relay system can be constructed from an array of communication components, an array of automation components, and an array of control components (if possible). Control components might require physical connections to the what they control which would prevent the deployment of multiple instances. Together these arrays provide redundancy so that components can be replaced as needed to ensure continued, reliable operations and improve overall resilience.

The design of dedicated function, multi-component systems can natively benefit from network defense-in-depth protection by permitting communication only between components that need to inter-communicate. The two examples discussed in this paper use a sequential, point-to-point network. This paper proposes interconnecting the components by using software-defined networking (SDN) to control access at the component level and prevent communication by components without specific needs to communicate, such as inactive components or components that are not undergoing maintenance.

AUTONOMOUS CYBER AGENTS

Overview of Agent-based Cybersecurity

Autonomous agents offer substantial advantages for enabling cybersecurity. When embedded with the systems and networks they protect, agents have more visibility to detect problems earlier and respond more rapidly and effectively than humans. The roles of agents will vary ranging from heuristic agents performing well-defined tasks to highly-intelligent self-learning, self-planning mobile entities that can actively patrol the network and robustly react to hostile threats such as autonomous malware.

Evans in (Alexander Kott 2018) describes how to endow autonomous agents with goals and capabilities. This paper proposes autonomous cybersecurity agents who detect, resist, and respond to adversaries to achieve their goal of defending and protecting the integrity and availability of a system composed of redundant arrays of dedicated-function components. To achieve their goals, agent-based cybersecurity models must endow agents with various capabilities to enable them to detect, resist, and respond to adversaries. Agents need to sense their environment, detect and evaluate changes, and respond to suspicious changes. Agents need to communicate and cooperate

with other entities such as other agents, systems, or humans. Agents need the ability to evaluate and improve their performance and exploit opportunities for improvement. Agents need to actively hunt for threats; verify the integrity and availability of the systems and networks; assess the state of the components, systems, networks, and network traffic; detect and characterize anomalies; and distinguish between normal and abnormal states.

Agents needs stealth, deception, and denial capabilities to defend themselves and the systems and networks they manage and protect (Heckman, et al. 2015). They need to behave stealthily to conceal themselves and their resources, such as processes, data, systems, networks, investigations, and communications to prevent direct observation by adversaries (Yuill 2006). For example, in this paper, agents need to hide the standby components and the trusted reference and recovery material, such as integrity signatures for software and configurations, and backup copies for use in recovery and restoration.

To verify the integrity of the system and network elements, agents need capabilities to sense and detect changes by directly observing the elements or analyzing event logs to discover modifications of software, firmware, and configurations of systems and networks. They need the capability to check whether the changes are authorized by referencing trusted digitally-signed cryptographic hashes or by communicating with other agents or the organization to verify the authenticity of the change. If the change is unauthorized, they need to assist the organization in recovery and response or initiate a response by themselves or by cooperating with dedicated response agents.

When deployed to proactively manage systems that permit replacing active systems with redundant standby replacements, agents need the capability to replicate the state of an active system to a redundant, standby system. This replication capability enables agents to ensure continuity of operations when employing proactive defense technologies such as moving target defense (MTD). The resilience of MTD depends upon the operational readiness of a standby system to quickly recover operations with minimum downtime. The ability of the standby system to resume service depends upon how closely its operational state matches the system it replaces.

Restoration requires the capability to retrieve and install trusted copies of software, firmware, and configurations. Agents need access to secure storage locations (which might be proactively hidden) and the ability to communicate with the organization to obtain new versions if available.

Agents need to segment, isolate, and hide a variety of infrastructure, data, and processes. They need to control and protect systems and networks by restricting, blocking, or throttling communications by direction (inbound versus outbound), by location (IP address, network, etc.), or by characteristics (latency, bandwidth, speed, and amount).

Throughout these activities, agents need to communicate with other agents and the organization and ensure that agents and the organization remain aware of the progress and results of the response.

Information sharing forms the basis of most communication and collaboration among agents and humans. To work together, agents and humans need a common understanding of the state of the

system and the actions taken in response to adversaries or other issues. This includes agent-initiated and organization-initiated changes to configurations to enable or block activities as well as the reasons for these actions such as signatures or indicators of compromise. Other agents and the organization can use this information to protect their systems and networks.

Agents and humans within the organizations who plan to take actions need to inform other agents and humans so they can expect an authorized change, prepare for the effects of the change, and manage the expected alarms or other notifications. Agents deployed to monitor system and network configurations and state need to detect changes. If the agent is not informed about the authenticity of the change, they can alert the organization and respond on its behalf if requested.

USING AUTONOMOUS AGENTS TO ENSURE SYSTEM INTEGRITY AND AVAILABILITY

To ensure their primary goals of system integrity and availability, agents will serve several roles and reside in various locations within the system, the network, and, where possible, the components of the system. The agents will serve various roles to achieve their primary goals of ensuring integrity and availability. Agents will monitor the availability and integrity of the component, its software, firmware, and communications. If monitoring detects problems, the agents will replace components with spare components from their redundant array.

This paper proposes employing autonomous agents to manage the software-defined networking multiple operating system rotational environment (SMORE) approach invented by Lyle and Evans (Lyle 2020). SMORE provides the capability to enforce strict network security by managing the movement of the target at the networking layer. Agents use SMORE to switch between active and hot-standby components; and use software-defined networking to enable or restrict access to inactive components to allow replicating operating state information or receiving updates from the organization.

Availability-monitoring agents will reside within the network on separate components dedicated to supporting agents and use a component-supported method, such as a dedicated interface or a network-based protocol. Integrity-monitoring agents will be situated within the component if possible. Agents will serve to monitor other agents and vote to replace and restore agents which fail availability or integrity verification tests.

Integrity Verification of Components

The methods of verifying the integrity of components will depend upon the design and possibly the function of the component, and typically rely upon a separate trusted reference, such as a digitally-signed trusted cryptographic hash. The agents will obtain trustworthy updates of copies or signatures from the organization then manage these trustworthy references for use in verifying component integrity.

Agents will also monitor and verify that the component continuously performs as expected. For example, agents will monitor control signals from the control component or data from

instruments in the physical system under control. Agents will replace components which fail integrity tests or otherwise appear untrustworthy. To continually manage the information requires to verify component integrity, agents will communicate with the organization by submitting status reports, configurations, settings, or other relevant information to the organization for verification. If these differ from the expected information, the organization might request removal, replacement, and restoration of the component to a trustworthy state.

Managing the Operational Readiness of the Components

The resilience of the overall system depends upon the operational readiness of the redundant standby components which will replace components that fail integrity or availability tests. Maintaining operational readiness requires two processes: 1) replicate application configuration updates of the operational component to the redundant components and 2) replicate application state of the operational component to at least one redundant component.

Replicating updates to the application configuration requires intervention by the agents. In the protective relay control system example, the updates to the application configuration are communicated through the network to the communication component, forwarded to the automation component, which sends them to the control component. To replicate these updates, the agents need to intercept or otherwise obtain these updates and install them on the redundant components.

Replicating the application state of the operational component requires methods which can extract the application state or component-based methods which communicate state to one or more redundant components. The acceptable delay between when state changes and when redundant components receive updates depends upon the time-criticality of the process performed by the component. For processes that can tolerate momentary outage, the replacement component needs only the most current configuration of the system and application and not the current state of application. For time-critical processes, the replacement component needs hot-standby replication of the state of the application.

Note: although operational readiness might include updates to component software and firmware, the designers of the protective relay example expect to perform those updates manually. The use of redundant arrays of components, managed by autonomous agents, should enable automated updates to the component software and firmware, assuming methods to verify proper installation and operation are developed. This depends upon the criticality of the component and trust of agents. In the protective relay example, automated verification of the updates to the communication and automation components poses minimal risk but methods to update and verify the control component might pose unacceptable risk and require human intervention.

CONCLUSION

Simplifying computers by separating them into dedicated, single function components promises to improve the reliability of critical functions and substantially reduce the risk of cascading failure caused by faults in support functions. In addition to improving the resilience of critical functions,

the simplification and separation of functions onto dedicated components will improve the ability to verify their integrity.

The overall resilience of systems composed of dedicated-function components can be improved by deploying redundant arrays of components, continually monitoring their integrity and availability, replacing faulty or suspicious components, and managing the security of the network.

Although scheduled software routines can manage these activities, autonomous agents offer substantial advantages. Agents can more easily adapt to and recover from unanticipated problems, including faults that might disable the routines. Agents can be deployed in redundant groups on different components to tolerate faults and ensure their continuous availability. Groups of agents can collaborate; vote on decisions to ensure trustworthy decisions; and verify that their decisions are properly executed.

Agents can collaborate with each other and with their organization to increase situational awareness. Because agents situated within the systems and networks might observe activities or changes not observed or observable by the organization, agents can actively alert organizations about suspicious behavior and follow standard operating procedures which might require the agent to take immediate action or wait for guidance from the organization.

ACKNOWLEDGMENTS

The work presented in this paper was partially supported by the U.S. Department of Energy, Office of Science under DOE contract number DE-AC02-06CH11357. The submitted manuscript has been created by UChicago Argonne, LLC, operator of Argonne National Laboratory. Argonne, a DOE Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

Disclaimer: This report was prepared as an account of work sponsored by an agency of the United States Government. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government.

BIBLIOGRAPHY

Alexander Kott, Ryan Thomas, Martin Dražar, Markus Kont, Alex Poylisher, Benjamin Blakely, Paul Theron, Nathaniel Evans, Nandi Leslie, Rajdeep Singh, Maria Rigaki, S Jay Yang, Benoit

LeBlanc, Paul Losiewicz, Sylvain Hourlier, Misty Blowers, Hugh Harney,. 2018. "Toward Intelligent Autonomous Agents for Cyber Defense." *2017 Workshop by the North Atlantic Treaty Organization (NATO) Research Group.*

Edmund O. Schweitzer, III. 2020. "Resetting Protection System Complexity." *2020 Western Protective Relay Conference.*

Heckman, Kristin E., Frank J. Stech, Roshan K. Thomas, Ben Schmoker, and Alexander W. 2015. *Cyber Denial, Deception and Counter Deception.* Springer International Publishing.

Lyle, Joshua A. and Evans, Nathaniel. 2020. Software defined networking multiple operating system rotational environment. USA Patent 20200059434.

Polack, Kyle. 2020 . "A Modular Approach to Trusted System Design for Arms Control Treaty Verification." *Institute of Nuclear Materials Management Conference.*

Yuill, Jim, Dorothy Denning, and Fred Feer. 2006. "Using Deception to Hide Things from Hackers: Processes, Principles, and Techniques." *Journal of Information Warfare 5, no. 3* 26-40.