International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Aug 2024

Confirming Trust: A Reliability Framework for Distributed Systems

[1]Dr.S.Kondalarao,[2] Mr.HIMAMBASHA SHAIK ,[3] Pallapothu Yamuna,

[1]Professor and Head, Department of Master of Computer Applications, QIS College of Engineering & Technology, Ongole, Andhra Pradesh, India
[2]Assistant Professor, Department of Master of Computer Applications, QIS College of Engineering & Technology, Ongole, Andhra Pradesh, India
[3]PG Scholar, Department of Master of Computer Applications,
QIS College of Engineering &Technology, Ongole, AndhraPradesh, India

ABSTRACT_A distributed software architecture is used by many real-time process control and industrial control systems, including Supervisory Control and Data Acquisition (SCADA), and they rely on reliable message exchanges between software components. The Trust but Verify (TBV) middleware, which promotes the notion that software components shouldn't blindly trust each other, is presented in this work. This study makes the following contributions: (1) TBV middleware design.

(2) A water treatment facility is used as a cyberphysical system for the TBV's proof-of-concept implementation. (3) An experimental validation of the TBV using many attack scenarios wherein erroneous signals are sent at random by compromised or defective components.

1.INTRODUCTION

A lot of real-time industrial and process control systems, like Supervisory Control and Data Acquisition (SCADA), rely on trusted message exchanges between software components and a distributed software architecture. The engineering of these frameworks can incorporate any blend of client-server, n-level, or shared designs. The more general peer-to-peer model [1], in which components can be both producers and consumers of messages, is the model that is the focus of this paper for distributed applications. In addition, these systems must meet stringent performance, security, and reliability standards, which may prove challenging in a distribution setting. Software components must have confidence in one another in order for real- time and SCADA systems to function properly. This implies that parts should be guaranteed that different parts don't disturb

the activity of a framework when they (a) glitch or (b) act in a vindictive way. In addition, there must be assurances that (c) components can be independently verified;

d) messages between components are not altered while they are in transit, and e) data at rest is not altered without permission. Properties (c)- (e) above can be authorized with the utilization of notable cryptographic methods and conventions [2]. However, these methods do not take into account the scenario in which an authenticated attacker controls a component maliciously and forces it to send incorrect messages to other components. In the following, a message or command that differs from the one that would have been sent if the system state and the sender's component logic were not compromised and had not experienced any malfunction is referred to as an improper message (command). It is important to note that a critical system's operation could be disrupted by faulty components sending the wrong messages to other components. As a result, a component of the recipient system might respond incorrectly to a message and harm a crucial system, like a SCADA system or an aircraft control system. As a result, a receiving component must be able to confirm that a sending component has not been inconsistent. A framework for mitigating the effects of defective or malicious components is presented in this paper. Realtime systems and other critical distributed application systems have emerged as popular targets for hackers. Attacks on these kinds of systems increased by more than 110% in 2016 compared to the previous year, according to IBM's X-Force Threat Research report [3]. This paper proposes the middleware framework Trust But Verify (TBV), which promotes the idea that components should not blindly trust each other. It is inspired by the well-known proverb "Trust but Verify," which President Ronald Reagan used in his nuclear discussions with the Soviet Union. Instead, they ought to check each message for consistency before acting on it.

2.LITERATURE SURVEY

2.1 TC-Aloha: A novel access scheme for wireless networks with transmit-only nodes

AUTHORS: L. Galluccio, G. Morabito, and S. Palazzo,

In the recent past several network scenarios have emerged where transmit-only nodes - i.e., nodes without receiving capabilities - are deployed. Such nodes cannot perform carrier sensing and cannot be synchronized. Therefore, they have to apply an Aloha-like medium access control. However, it is well known that Aloha achieves low good put due to the possibility to incur in collisions, and this

results in poor energy efficiency too. In order to achieve better performance, in this paper a scheme called Timing-Channel Aloha (TC-Aloha) is introduced which exploits the timing channel. The timing channel is the logical communication channel established between a transmitter and a receiver in which the information is transferred by means of the timing of events. Another feature of TC-Aloha is that it enables multiple transmissions of the same information to improve the communication reliability. In this paper the TC-Aloha scheme is described in detail and an analytical framework is derived for the evaluation of its performance. The numerical results assess the advantages of TC-Aloha over traditional solutions.

2.2 Efficiency analysis of jamming-based countermeasures against malicious timing channel in tactical communications

AUTHORS:S. D'Oro, L. Galluccio, G. Morabito, and S. Palazzo

A covert channel is a communication channel that creates a capability to transfer information between entities that are not supposed to communicate. A relevant instance of covert channels is represented by timing channels, where information is encoded in timing between events. Timing channels may result very critical in tactical scenarios where even malicious nodes can communicate in an undisclosed way. Jamming is commonly used to disrupt this kind of threatening wireless covert communications. However jamming, to be effective, should guarantee limited energy consumption. In this paper, an analysis of energy-constrained jamming systems used to attack malicious timing channels is presented. Continuous and reactive jamming systems are discussed in terms of their effect on the achievable covert channel capacity and jammer energy consumption. Also, a simple experimental set up is illustrated and used to identify proper operating points where jamming against malicious timing channels is effective while achieving limited energy consumption.

2.3 Jamming sensor networks: Attack and defense strategies

AUTHORS:W. Xu, K. Ma, W. Trappe, and Y. Zhang

Wireless sensor networks are built upon a shared medium that makes it easy for adversaries to conduct radio interference, or jamming, attacks that effectively cause a denial of service of either transmission or reception functionalities. These attacks can easily be accomplished by an adversary by either bypassing MAC-layer protocols or emitting a radio signal targeted at jamming

a particular channel. In this article we survey different jamming attacks that may be employed against a sensor network. In order to cope with the problem of jamming, we discuss a two-phase strategy involving the diagnosis of the attack, followed by a suitable defense strategy. We highlight the challenges associated with detecting jamming. To cope with jamming, we propose two different but complementary approaches. One approach is to simply retreat from the interferer which may be accomplished by either spectral evasion (channel surfing) or spatial evasion (spatial retreats). The second approach aims to compete more actively with the interferer by adjusting resources, such as power levels and communication coding, to achieve communication in the presence of the jammer.

3 PROPOSED SYSTEM

(1) The design of the TBV middleware, which intercepts all message communications between sending and receiving components and compares them to message type-specific rules to ensure message consistency; this verification takes into account the system state, which is updated via a state distribution mechanism like the gossiping protocol. This is one of the contributions of this paper. A message is either delivered to the recipient or dropped based on this verification.

(2) An application of the TBV framework on a cyberphysical system (CPS) as a proof of concept

(3) An experimental validation of the TBV using many attack scenarios wherein erroneous signals are sent at random by compromised or defective components. In these trials, the attack probability and the state database update delay are used to determine the TBV's attack detection rate.

(4) An assessment of the TBV middleware's costs and performance impact

3.2 METHODOLOGY

Service Provider

In this module, the Service Provider browses the required file, initializes nodes with digital signature and uploads to the end user (node a, node b, node c, node d, node e, node f) via Router.

Router

The Router is responsible for forwarding the data file in shortest distance to the destination; the Router consists of Group of nodes, the each and every node (n1, n2, n3,n4,n5,n6,n7,n8,n8,n10,n11,n12, n13) consist of Bandwidth and Digital Signature. If router had found any malicious or traffic node in the router then

it forwards to the IDS Manager. In Router we can assign the Sleeping time for the nodes and can view the node details with their tags Node Name, Sender IP, Injected data, Digital Signature, Sleeping time and status.

End User

In this module, the End user can receive the data file from the Service Provider which is sent via Router, if malicious or traffic node is found in the router then it never forwards to the end user to filter the content and adds to the attacker profile.

3.2 ALGORITHM

Algorithm for Trust But Verify (TBV) Middleware

The TBV middleware's core functionality involves intercepting, verifying, and either forwarding or blocking messages based on their consistency with predefined rules and the system's state. Below are the detailed working steps along with relevant formulas and pseudocode.

Step 1: Interception of Messages

When a message is sent from a sender to a receiver, the TBV middleware intercepts it before it reaches the receiver.

Step 2: Authentication Verification

Ensure that both sender and receiver are authenticated using mutual authentication mechanisms.

Step 3: Message Consistency Verification

Check the message against predefined rules and the current system state. This involves verifying:

1. **Message Format:** Ensure the message adheres to the expected format.
2. **Message Type Rules:** Verify the message content against the rules specific to its type.
3. **System State:** Ensure the message is consistent with the current state of the system.**Formulas:**

Let MMM be the message, RtR_tRt be the set of rules for message type ttt, and SSS be the current system state.

1. **Format Verification:**
   - Use a regular expression regext\text{regex}_tregext for message type ttt.

$$\text{isValidFormat}(M, \text{regex}_t) = \text{True or False}$$

2.Rule Verification:

- Each rule $r \in R_t$ is a function $r(M, S)$.

$$\text{isValidContent}(M, R_t, S) = \forall r \in R_t, r(M, S) = \text{True}$$

Step 4: Decision Making

Based on the verification, decide whether to forward the message to the receiver or block it.

Experimental Validation

To validate the effectiveness of the TBV

middleware, conduct experiments involving various attack scenarios where compromised or faulty components send erroneous messages. Measure the TBV's detection rate and the overhead it introduces.

1.Detection Rate:

Ratio of correctly detected and blocked erroneous messages to the total number of erroneous messages sent.

$$\text{Detection Rate} = \frac{\text{Number of blocked erroneous messages}}{\text{Total number of erroneous messages sent}}$$

2.Overhead:

Measure the additional time and resources consumed by the TBV middleware.

$$\text{Overhead} = \text{Time with TBV} - \text{Time without TBV}$$

```
while (true)
                {
                con777 = server909.accept();

DataInputStream dis = new
DataInputStream(con777.getInputStream());

                String node1  = dis.readUTF();

                String attack="Yes";
                SimpleDateFormat dateFormat = new
SimpleDateFormat();
            Date date = new Date();

            String dt=dateFormat.format(date);
                String energy="";

            if(node1.equalsIgnoreCase("Node8"))
            {
                ResultSet rs=
connect.createStatement().executeQuery("select * from NodesInfo where
node='"+node1+"'");

                while(rs.next())
                {
                    energy=rs.getString(3);
                }
            }
            if(node1.equalsIgnoreCase("Node12"))
            {
                ResultSet rs=
connect.createStatement().executeQuery("select * from NodesInfo where
node='"+node1+"'");

                while(rs.next())
                {
                    energy=rs.getString(3);
                }
            }
            if(node1.equalsIgnoreCase("Node13"))
            {
                ResultSet rs=
connect.createStatement().executeQuery("select * from NodesInfo where
node='"+node1+"'");

                while(rs.next())
                {
                    energy=rs.getString(3);
                }
            }
            if(node1.equalsIgnoreCase("Node14"))
            {
                ResultSet rs=
connect.createStatement().executeQuery("select * from NodesInfo where
node='"+node1+"'");

                while(rs.next())
                {
                    energy=rs.getString(3);
                }
            }
```

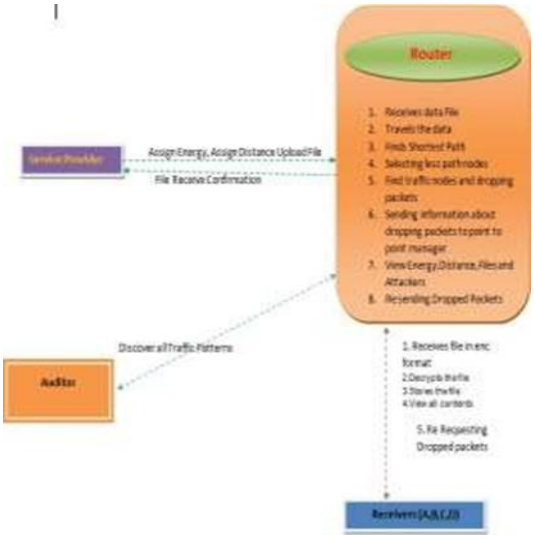
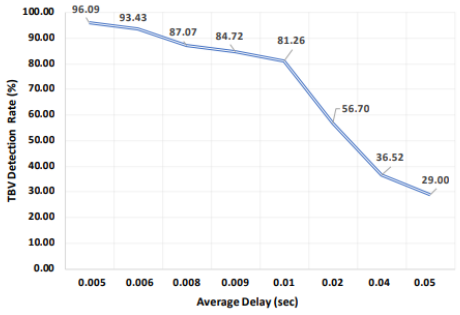


Fig 1:Architecture

4.RESULTS AND DISCUSSION



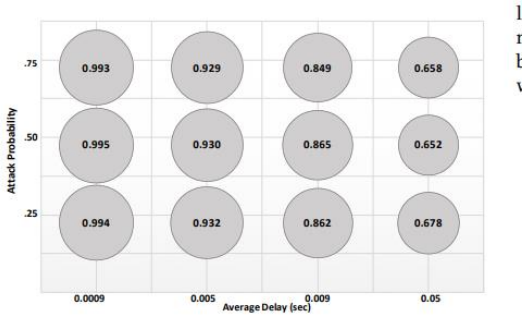


TABLE 1
Two-way ANOVA for TBV detection rate for tank overflow attack in stage 4

Source of Variation	F	p-value	Fcrit
delay	13662.093	0.000	2.631
attack probability	0.637	0.529	3.022
Interaction	0.110	0.995	2.125

TABLE 2
Two-way ANOVA for TBV detection rate for water quality attack in stage 4

Source of Variation	F	p-value	Fcrit
delay	2066.945	2.4216E-221	2.631
attack probability	2.938	0.054	3.022
Interaction	1.831	0.092	2.125

TABLE 3
Two-way ANOVA for TBV detection rate for tank overflow attack in the entire system

Source of Variation	F	p-value	Fcrit
delay	61150.824	0.000	2.631
attack probability	5.902	0.003	3.022
Interaction	1.222	0.294	2.125

TABLE 4
Two-way ANOVA for the TBV detection rate for water quality attack in the entire system

Source of Variation	F	p-value	Fcrit
delay	2932.713	1.3954E-246	2.631
attack probability	2.931	0.055	3.022
Interaction	1.366	0.228	2.125

4.CONCLUSION

This paper presented a system that confirms in the event that order messages sent by a part (e.g., a PLC) to another product part (e.g., an actuator) are reliable with a bunch of space explicit principles. Application data and constraints can be used in the verification process with the help of middleware like the TBV. A system is against a rules DB, making it unnecessary to modify the application code in order to use the middleware. This works with the coordination of the TBV structure with existing dispersed applications. The fact that the current arrangement separates shielded from the effects of malicious and defective components thanks to the verification. While the TBV is appropriate to a disseminated framework, it is the most ideal for crucial frameworks, for example, cyberphysical frameworks. TBV components are deployed between message senders and receivers and use a system state DB to verify message consistency concerns is a nice feature: Because the TBV takes care of this issue, a CPS need not be concerned about the effects of faulty or malicious components.

Additionally, adding a TBV meets the level of rigor and assurance required by critical systems,

supports layered security, improves system resilience, and more. The TBV is dependable because it reduces the likelihood of attacks or failures by utilizing self-protection capabilities like redundancy, diversity, and software protection techniques. The TBV code is simple, application-independent, easy to maintain, and can be adapted to any software protections because it is not proprietary, in contrast to application component code (a PLC in our prototype). On the other hand, many CPS components use proprietary methods [75] that restrict the use of software security controls like software/hardware attestation [64]. We gave a nitty gritty illustration of how the TBV can be incorporated to Smack, a water treatment office model, and examined the location pace of assaults and estimated the normal execution above of utilizing the TBV. The TBV allows for commands that can take any range of parameters, as discussed in Section 2's Rule Verification Engine steps, despite the fact that SWaT commands are binary. We intend to investigate methods that automatically extend the base rule templates by learning from system states and constructing dynamic rules using either physics-based models or machine learning techniques in light of the emergence of unknown attacks. Amazon EC2 VM instances were used in our experiments. The commotion presented by other VMs sharing Amazon servers could influence the TBV execution. However, the TBV average overhead appears to be worse than it would in

REFERENCES

[1] A. Belapurkar, A. Chakrabarti, H. Ponnapalli, and N. Varadarajan,Distributed Systems Security: Issues, Processes and Solutions. JohnWiley and Sons, 2009.

[2] J. Katz and Y. Lindell, Introduction to modern cryptography, 2nd. ed.Chapman& Hall, 2014.

[3] M. Alvarez and S. Craig, “At risk: The energy andutility sector infrastructure,” 2017, accessed: 2018-05-20.[Online]. Available: https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=SEL03135USEN

[4] M. Berndtsson and J. Mellin, ECA Rules. Boston, MA: SpringerUS, 2009, pp. 959–960.

[5] A. Kermarrec and M. van Steen, “Gossiping in distributed systems,” SIGOPS Oper. Syst. Rev., vol. 41, pp. 2–7, Oct 2007.

[6] G. Miklau and D. Suciu, “Implementing a tamper-evidentdatabase system,” in Advances in Computer Science – ASIAN 2005.Data Management on the Web. Berlin Heidelberg: Springer, 2005,pp. 28–48.

[7] IAB and G. Camarillo, “Peer-to-Peer (P2P) Architecture: Definition, Taxonomies, Examples, and Applicability,” RFC 5694, Nov.2009. [Online]. Available: https://rfc-editor.org/rfc/rfc5694.txt

[8] S. Adepu and A. Mathur, “Generalized attacker and attack modelsfor cyber physical systems,” in 2016 IEEE 40th Annual ComputerSoftware and Applications Conf. (COMPSAC), vol. 1, June 2016, pp.283–292.

[9] T. Bartman and K. Carson, “Securing communications for SCADAand critical

industrial systems," in 69th Annual Conf. ProtectiveRelay Engineers (CPRE), April 2016, pp. 1–10.
[10] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defensemechanisms against distributed denial of service (DDoS) floodingattacks," IEEE Communications Surveys Tutorials, vol. 15, pp. 2046–2069, Fourth 2013.

[11] E. N. Ylmaz, B. Ciylan, S. Gnen, E. Sindiren, and G. Karacaylmaz,"Cyber

security in industrial control systems: Analysis of DoSattacks against PLCs and the insider effect," in 2018 6th Intl.Istanbul Smart Grids and Cities Congress and Fair (ICSG), April 2018,pp. 81–85.
[12] M. Permann, K. Lee, J. Hammer, and
K. Rhode, "Mitigations forsecurity vulnerabilities found in control systems networks," inProc.16th Annual Joint ISA POWID/EPRI Controls and Instrumentation Conference, 2006.
[13] R. Radvanovsky and J. Brodsk, Handbook of SCADA/Control Systems Security. CRC Press, 2016.
[14] NIST, "Guide to industrial control systems (ICS) security,"2015. [Online]. Available: ttps://csrc.nist.gov/publications/ detail/sp/800-82/rev-2/final

AUTHOUR'S PROFILE

Dr.S. KONDALARAO

MPhil. D, MLSE, MIST currently working as an head of department and in charge of PG programs in the department of MCA,QIS college of engineering and technology, Andhra Pradesh, Ongole. Dr. Kondalarao brings a wealth of knowledge and expertise to his role. His research interests are broad and impactful, encompassing areas such as software integrity, control systems for plants with variable structure, and tamper-resistant software.

Mr.HIMAMBASHA SHAIK
ASST.PROFESSOR DEPT.OF M.C.A
S.N.PADU.
AREAS OF INTREST
CLOUD COMPUTING,DEVOPS AND DBMS..

MS.PALLAPOTHU YAMUNA,
Currently pursuing Master of Computer Applications at QIS College of engineering and Technology Autonomous), Ongole, Andhra Pradesh. She Completed BSC(C), from TRR. GOVT. Degree College, Kandukur, Andhra Pradesh. Her areas of interests are Cloud Computing & java