Necesse 2018, Vol 3, Issue 2, 149-163 DOI: 10.21339/2464-353x.3.2.149

An Attack on an Integrated Navigation System

Mass Soldal Lund[1], Odd Sveinung Hareide[2,3], and Øyvind Jøsok[1,4]

[1] Norwegian Defence University College, Cyber Academy
[2] Norwegian Defence University College, Royal Norwegian Naval Academy, Navigation Competence Center
[3] Norwegian University of Science and Technology, Joint Research Program in Nautical Operations
[4] Inland Norway University of Applied Sciences, Faculty of Social and Health Sciences

Abstract. Maritime cyber security is emerging as a field as reports of cyber attacks against computerized maritime systems have started arriving. Modern vessels are equipped with computerized systems for navigation employing the Global Positioning System (GPS), known as Integrated Navigation Systems (INS) and Electronic Chart Display and Information Systems (ECDIS). This paper describes a proof-of-concept attack on an INS and its integrated ECDIS, and reports on a demonstration of the attack on a vessel. The attack includes malware that acts as a man-in-the-middle intercepting and manipulating GPS coordinates. Furthermore, the paper discusses the feasibility of the attack, as well as countermeasures.

Keywords: Cyber security · Integrated Navigation System · ECDIS

1 Introduction

Maritime cyber security is emerging as a potentially large concern [8,9,13,33]. Modern vessels are equipped with computerized systems for navigation using Global Navigation Satellite Systems (GNSS) such at the Global Positioning System (GPS). Lately, reports of cyber attacks against maritime systems have started arriving and have placed cyber security at sea on the agenda [4,7]. One type of attack is GPS spoofing, in which navigation systems are fooled by the transmission of false GPS signals [24,30]. In a recent incident, more than 20 vessels operating in the Black Sea reported receiving obviously wrong GPS positions in what appears to be a massive GPS spoofing [14].

Maritime navigation systems connected through onboard networks are usually referred to as Integrated Navigation Systems (INS) [20]. In an INS, operator stations with software for displaying the vessel's position in electronic charts – known as Electronic Chart Display and Information Systems (ECDIS) [19] – are integrated with the GPS and other devices such as heading sensors (gyroscope), depth sensors, Automatic Identification System (AIS), etc. [16] (see Fig. 1).

In order to build defenses, it is necessary to understand attacks. In this paper we report on a proof-of-concept attack on an INS, and the practical demonstration of the attack conducted on a vessel in cooperation with the Royal Norwegian Navy (RNoN).[1]

The main feature of the attack is to infect the ECDIS software on an operator station with malware that acts as a man-in-the-middle intercepting and manipulating incoming coordinates from the GPS. This can be seen as a kind of GPS spoofing that does not interfere with the signals of the GPS satellites, but with the internal signaling of the INS. In addition the malware can crash the operator station by provoking a bluescreen. The system on which the attack was demonstrated is air-gapped, i.e. without an Internet connection; a USB Human Interface Device (HID) attack is therefore used for delivery. The attack is rather crude, but we believe that it demonstrates the feasibility of this kind of attacks. To some degree, the attack depends on bad security of maritime systems. Several reports available indicate that there are large cyber security challenges in the maritime sector (see e.g. [4,7,10,26]). Such reports are usually produced by security companies, which may have an interest in exaggerating the situation, but on the other hand we should also expect underreporting of incidents. In the absence of more systematic studies the overall state of cyber security at sea is unknown. The attack presented in this paper nonetheless highlights the need for maritime cyber security, and also what can be gained (security wise) from implementing relatively simple security measures.

The remainder of the paper is organized as follows: We first give a brief presentation of the INS in question and some of its main features (Section 2). This is followed by a detailed presentation of the attack and its development, structured after the intrusion kill chain model of Hutchins et al. [17], including a report from the practical demonstration of the attack [15] (Section 3). After presenting the attack, we discuss possible further developments (Section 4), as well as the feasibility of the attack and possible counter-measures (Section 5). Finally, we provide conclusions (Section 6).

2 Integrated Navigation System (INS)

The INS in question, illustrated in Fig. 1, is a network that connects a number of operator stations (bridge consoles) with the sensors of the vessel. The diagram shows the sensors, such as the Global Positioning System, the Gyro System, the Automatic Identification System and so forth, connected to a sensor integrator (SINT) through serial connections (blue lines). The operator stations are connected to the same SINT through switches in a Dual LAN (red lines). The Dual LAN is two parallel Ethernets of which one is a backup, thus providing redundancy. As a second redundancy feature, the operator stations also have serial connections to the SINT (thin blue lines). The role of the

[1] The INS and ECDIS in question are anonymized per request of the provider. The owner of the vessel and the provider of its INS supported the project and had representatives participating in the planning, the reconnaissance and the practical demonstration. The provider of the INS has been offered all information concerning the attack and has approved of the publication of this paper.

SINT is to receive signals from the sensors and communicate them to the operator stations in a common format, thus providing a single source of sensor data. In addition the SINT integrates an autopilot (AP). The operator stations are regular desktop computers running ECDIS software on top of the Windows operating system. The ECDIS software interprets the signals received form the SINT and uses information such as GPS coordinates and AIS messages to render the vessel's position and heading, as well as the position of nearby vessels, in the chart.

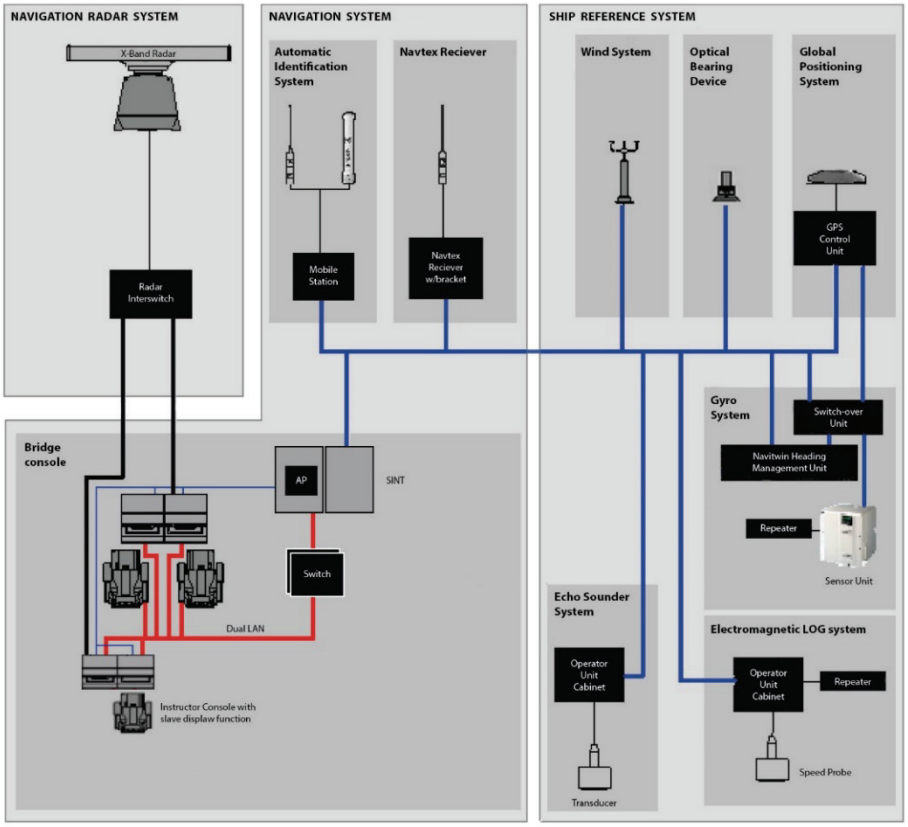

Fig. 1. Schematic of the INS from the vessel's documentation (courtesy of RNoN)

3 Attack

The intrusion kill chain [17] is a tool developed for analyzing cyber attacks. It provides a model that describes the work process of an attacker seeking to infiltrate computer system in seven phases: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives. Thus, while the model is developed as an aid to the defender it utilized the waypoint of the attacker. This makes it a good tool also for describing the development of an attack, because it will highlight

the assumptions and choices made, and the resources needed, in the development. In the following we explain our attack in detail according to the seven phases of the intrusion kill chain. At the end of the section we report from a demonstration of the attack.

3.1 Reconnaissance

In our reconnaissance efforts, we had four main sources of information:

1. A field trip was conducted on board a vessel identical to the vessel on which the demonstration was to take place. During this field trip we observed the workings of the system and had the opportunity to ask questions to representatives of the owner of the vessel and the provider of the INS.
2. During the field trip, we also had the opportunity to monitor and capture data traffic on the INS network while the vessel was sailing. This was achieved simply by plugging a regular laptop computer with Wireshark installed into a spare port in the switch using a network cable.
3. We obtained a laptop with the ECDIS software installed.
4. We were given access to technical documentation of the INS installation of the vessels.

In addition, some testing and troubleshooting was performed on a second field trip when the attack was demonstrated. The likelihood of an attacker being able to perform the same kind of reconnaissance is discussed in Section 5.1.

From the field trip we learned that operator stations are running Windows 7 and logged in with user profiles with administrator privileges. Inspection of the ECDIS software on the laptop we obtained showed that it uses Windows Sockets 2.0 (Winsock) for network communication. Inspection of its configuration files, held together with the documentation, gave useful information about the configuration of the network, e.g. that all nodes have static IP addresses. This, together with the network traffic capture, also told us the IP addresses of the network and port numbers used in the communication. Analysis of the network traffic capture revealed the format of the SINT signals, which turned out to be User Datagram Protocol (UDP) multicasts of plaintext messages formatted according to the NMEA 0183 standard, usually referred to as NMEA sentences. For learning the details of NMEA sentences we used more or less arbitrary Internet source [5,36].

3.2 Weaponization

The payload of the attack consists of two parts: a fake Windows socket dynamic-link library (Winsock DLL) and a script that crashes the computer by provoking a bluescreen. The fake Winsock DLL is a proxy for the proper Winsock DLL (`ws2_32.dll`) developed using available skeleton code [2]. It acts as a man-in-the-middle between the proper DLL and the ECDIS software, inspecting and manipulating data packages received from the network. The SINT transmits the NMEA sentences as ASCII plaintext; an NMEA sentence with GPS coordinates is of the following format:

```
$GPGGA,083548.53,6022.10378,N,00510.06015,E,1,11,0.8,
  54.74,M,,M,,*71
```

The letter code at the start (`GPGGA`) identifies the sentence as one carrying a GPS position. Identifying and changing the coordinate (60°22.10' N, 5°10.06' E) is straightforward; the ECDIS software will accept a modified sentence as long as the checksum at the end (`*71` in the example) is recalculated.

The other part of the payload is a script that causes a bluescreen. It utilizes the PsKill application of the Windows Sysinternals package [32] to kill an essential system process (the Client/Server Runtime Subsystem, `csrss.exe`). The script itself is written in VBScript, as are all other scripts used in the attack. The reason for this choice (as opposed to using for example PowerShell) is to make the attack compatible with basic installations of both Windows XP and Windows 7 as both Windows versions are commonly used as platforms for ECDIS applications [10,23,25].

Crucial for the development of the payload was the setup of a test environment. We used VMware software to virtualize the laptop with the ECDIS software installed and put it in a virtual network with a simple SINT simulator – a Debian VM running a small Python script generating UDP multicasts containing NMEA sentences with GPS coordinates.

3.3 Delivery

For delivery of the payload we made a device consisting of a Teensy microcontroller [29], a USB flash drive and a USB hub. Utilizing the possibility of programming the Teensy microcontroller to simulate a USB keyboard and a USB mouse at the same time, we developed a so-called USB HID attack (see e.g. [28]). The delivery is in three phase:

1. The ECDIS software has a built in key capturing feature that prevents keyboard shortcuts (such as Windows key + `R` to get the Windows Run dialog) from being effective. This is circumvented by simulating keyboard and mouse to enter a maintenance password which allows this feature to be disabled. (Some considerations on the passwords of the system are made in Section 5.1.)
2. With the key capturing feature disabled, the simulated keyboard uses keyboard shortcuts to open a command prompt. In the command prompt it uses command line tools to make necessary changes to the computer, as well as to type, save and run a small script.
3. The script identifies the USB flash drive and obtains the payload from it.

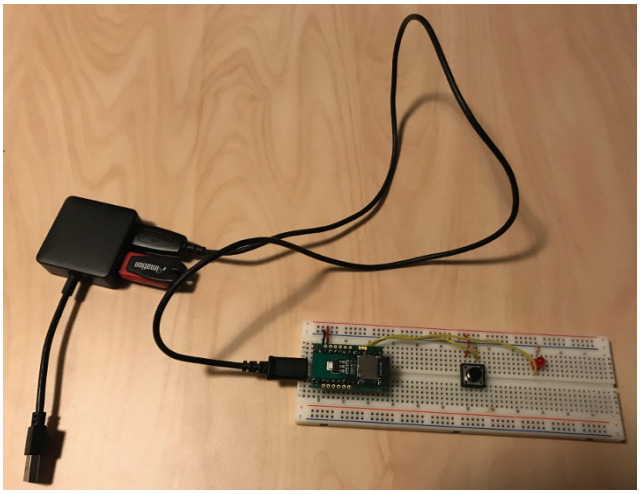

Fig. 2. Prototype USB delivery device consisting of a Teensy microcontroller, a USB flash drive and a USB hub

The choice for this means of delivery was based on the assumption that the INS is air-gapped. Thus, it was necessary with a method based on some form of physical access. Using simulated keyboard and mouse have the benefit that we do not need to assume an exploitable vulnerability in the operating system or other software. Furthermore, we could not rely on downloading the payload through the Internet. Having the simulated keyboard type the whole payload would take an unacceptable amount of time. (In a test, the simulated keyboard needed approximately seven and a half minutes to type a Base64 encoding of the 161 kB payload.) To avoid this, the choice fell on combining the Teensy with a USB flash drive though a USB hub. The (prototype) device is shown in Fig. 2.

The scenario for application of this means of delivery would be for the attacker to discretely insert the device into a USB socket on one of the operator stations during a visit on the bridge of a vessel. Other means of delivery that we believe may work are discussed in Section 4.1.

3.4 Exploitation

The attack does not exploit technical vulnerabilities in the operating system or other software. The vulnerabilities exploited in the delivery is the fact that the target computer by default is logged in with a user profile with administrator privileges, combined with physical access and knowledge of the maintenance password of the ECDIS software. The attack itself exploits the possibility of tricking the ECDIS software into loading the fake Winsock DLL in what is sometimes called a DLL search order hijacking attack (see e.g. [12]).

3.5 Installation

A common way of installing malware is to first deliver a small program or script which downloads the main payload via an Internet connection, a so-called dropper. However, we worked under the assumption that the target system is air-gapped. In the attack we therefore have the simulated keyboard deliver the dropper – by typing, saving and executing a script – and then have the dropper copy the payload – a zip file – from the USB flash drive to the hard drive of the target computer and unzipping it. A combination of keystrokes from the simulated keyboard and scripts contained in the payload then accomplishes the installation: The fake Winsock DLL is copied to the installation folder of the ECDIS software, and the registry of the computer is updated to exclude Winsock from KnownDLLs in order to trick the software into loading the fake DLL. (KnownDLLs is a mechanism for ensuring that certain standard DLLs are loaded from the Windows distribution, see e.g. [12].) In addition, a scheduled task is created that runs a script for provoking bluescreens. (The bluescreen script is described in more detail in Section 3.7). Finally, the computer is restarted to make the changes come into effect; the default user profile is automatically logged in and the ECDIS application launched on startup. In the demonstration of the attack (see Section 3.8) the time used by the full delivery and installation was 5 minutes, 17 seconds, including restart of the computer and the ECDIS software which took 2 minutes, 26 seconds. This means the time used by the USB device was 2 minutes, 51 seconds, but there is a potential for optimizing the process. We believe the time used by this first part (i.e. the time the USB device has to remain in the USB socket) can be reduced by approximately one and a half minutes. The installation is visible, but one option for preventing this could be to have the device dim the screen as its first action.

3.6 Command and Control

As the target system is air-gapped, the attack does not rely on any command and control mechanisms. However, in Section 4.2 we discuss a possibility of devising a simple form of command and control without violating the assumption that the system is air-gapped.

3.7 Actions on Objectives

When installed, the attack can perform two kinds of actions: It can manipulate GPS coordinates received from the SINT via the network and it can provoke the operator station to crash with a bluescreen. The manipulation of GPS coordinates is performed by adding a small accumulating (positive or negative) deviation to the latitude, longitude or both, each time an NMEA sentence carrying GPS coordinates is received. A bluescreen, as mentioned above, is provoked by the killing of an essential system process.

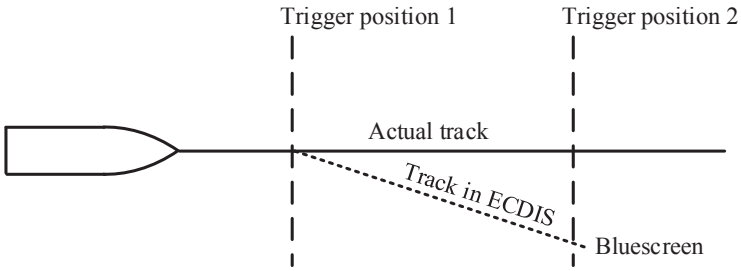

Fig. 3. Illustration of test run during the demonstration

Both actions are triggered by the position of the vessel. The intercepted GPS coordinates are compared to a set of specified triggering conditions, essentially rectangles defined by northern, southern, eastern and western limits. In the case of the GPS manipulation, the deviation will grow with a specified value every time a new GPS coordinate is received (approximately twice each second) as long as the received coordinates are within the specified limits. Outside of the limits the deviation will still be added to the coordinates, but as a constant. In the case of the bluescreen, a file is written to the hard drive of the computer if the triggering conditions are met. A script running every minute (activated by a scheduled task) checks the presence of the file, and if the file exists performs the process kill that provokes the bluescreen.

3.8 Demonstration

Both parts of the attack, as well as the delivery, were successfully demonstrated in four test runs during a passage in the littoral waters outside of Bergen, Norway in late August 2017. The vessel is equipped with several operator stations (see Fig. 1); by infecting one of them with the malware we could compare an infected and an uninfected operator station during the test runs. A typical test run is illustrated in Fig. 3. We specified two trigger positions. At the first trigger position the malware started accumulating the deviation, and thus the position shown in the ECDIS software started drifting from the actual track. At the second triggering position the malware provoked a bluescreen which made the operation station crash.

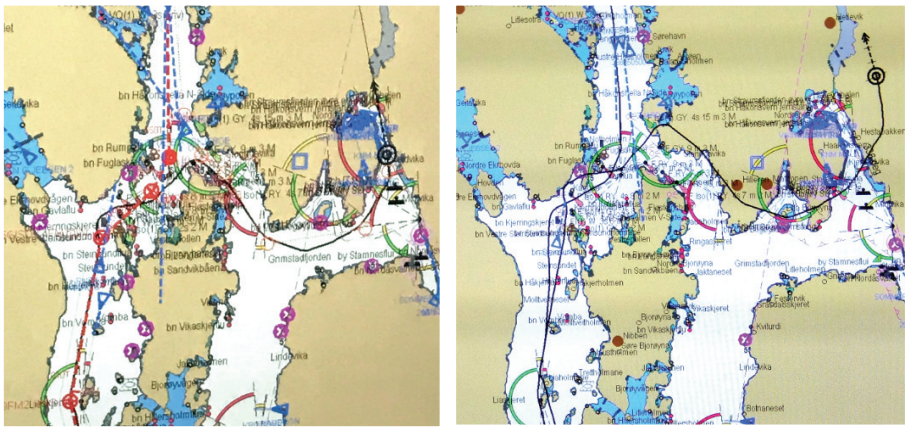

Fig. 4. Route plotted by ECDIS software using GPS data. Uninfected operator station to the left, infected operator station to the right

Fig. 4 shows the route plotted by the GPS in one of the test runs (in which the bluescreen was not tested) – uninfected operator station to the left and infected operator station to the right. In this test run the deviation grew by approximately 0.8 m toward north and 0.4 m toward east per second.

4 Further Development

The attack as implemented utilizes a specific vector of delivery and has no command and control structure. Both are in a sense features of the attack that are separate from its core, which is the manipulation of GPS data. In this section we examine how the attack may be developed further by using other means of delivery or by implementing a command and control structure.

4.1 Delivery

Our means of delivery was a special USB device that in any realistic scenario must be brought wittingly to the bridge of the vessel under attack by an agent. However, it should be worth considering other methods of delivery that do not have this requirement. The most obvious would be to make a malware that can spread via USB flash drives. For example, a case in which an ECDIS computer on board a large tanker was infected by malware when charts were updated using an infected USB flash drive has been reported [4]. Another obvious option would be to utilize an Internet connection. We developed the attack for a vessel on which the network is air-gapped, and traditionally such systems have been air-gapped due to lack of good Internet connections at sea. This is however changing, and navigation systems are now increasingly being equipped with Internet connections over satellite and/or 4G broadband (for use when sailing close

to shore) [4,7,10,25]. Furthermore, it has been demonstrated how this can be exploited to launch attacks [10,34].

A perhaps less obvious attack vector can be illustrated by tests we did during our reconnaissance when we plugged a laptop computer into the switch of the INS. This possibility is a common feature since stand-alone laptops are often used to plan routes, which are later transferred to the operator stations using the network. We were able to communicate (using e.g. Ping and nmap) with the other computers in the network after assigning our laptop a static IP address in the range used by the network. Clearly, there is a potential for using an external computer plugged to the network as a vector for delivery. Similarly, there would be a potential for the malware to spread between operator stations connected to the network.

4.2 Command and Control

In the attack we used the reception of specific GPS coordinates as a trigger for the malware to perform certain actions. However, we could easily have used any other information transmitted on the network by the SINT as the triggering condition. In our virtual test environment, we have demonstrated how the Automatic Identification System (AIS) can be used to develop a simple kind of command and control. AIS is an automated tracking system used extensively in the maritime world for exchange of information for anti-collision purposes. Ships and land-based stations equipped with AIS transceivers broadcast AIS messages containing static information (identity, vessel type, etc.) as well as voyage related (destination) and dynamic information (position, heading, speed) using the VHF spectrum [27]. As illustrated in Fig. 1, the INS integrates AIS. The ECDIS software of the operator stations receives AIS messages from the SINT and renders information from nearby ships in the navigation charts. An AIS message is basically a number of values packed in a bit string. This bit string is encoded as ASCII characters in a fashion similar to Base64 encoding and transmitted as part of an NMEA sentence [31]. We had the fake Winsock DLL inspect AIS messages received from the simulated SINT and use the reception of specific vessel names as the triggering condition. We also encoded encrypted commands into AIS messages to make the malware on an operator station trigger bluescreens, change its triggering conditions, write to the hard drive and execute commands. We believe that constructing an AIS transceiver to transmit such coded messages would be doable given moderate resources, see [3]. Furthermore, the AIS on the vessel gets certain kinds of information, e.g. the specified destination, from the ECDIS software via the network and the SINT. Even though it was not tested, we believe this can be exploited to have the malware make the AIS transmit simple messages, e.g. to acknowledge received commands. The downside of this means of command and control is the relatively low bitrate of AIS which makes transfer of large files or data unpractical. (AIS uses two channels each with a bit rate of 9.6 kbits/s, but due to overhead, data encoding and conflict resolution the practical transfer rate is at least an order of magnitude lower [21]).

As mentioned in the previous section, navigation systems with Internet connections are becoming more common. In addition to providing another vector for delivery, this obviously opens up for other methods of command and control.

5 Feasibility and Counter-Measures

The attack was made under certain assumptions; the feasibility of the attack obviously depends on their validity. In the following we discuss the feasibility of the attack in the view of these assumptions. First, however, it should be recognized that the attack was successfully tested on an INS installation on a vessel without any prior modification of the system to make the attack work. Furthermore, the attack did not exploit any known or unpatched vulnerabilities in the installation. This in itself should demonstrate that the attack is possible. The question of the feasibility of the attack is therefore a question of whether we had unreasonable good intelligence and access, and a question of the effectiveness of barriers and counter-measures and to what degree they are implemented. We will discuss these two questions in turn.

5.1 Reconnaissance

It cannot be denied that we were given an excellent opportunity with access to software, documentation, network traffic, a full installation and experts on the system. On the other hand, these systems are commercially available and any actor could get the same access given sufficient resources. It would certainly be achievable for a state actor or a large criminal organization. The only piece of information used in the attack that we would not be able to obtain by buying an installation of the system is the maintenance password used to bypass the key capturing functionality. How easy it would be to obtain this password illegitimately will of course be speculation. The use of shared or role based passwords, as well as relying on access control implemented at the application level rather than the operating system, are considered security challenges in the context of industrial control systems (SCADA systems) [22]; these worries should carry over to navigation systems. However, research on password use suggests that password mechanisms should reflect the nature of the resources protected in order to avoid undermining the mechanism. Thus, using shared passwords to protect shared resources such as shared information or shared tasks can be an appropriate means of protection [1]. On the other hand, it has been found that shared passwords sometimes are weak and long lived since the challenges of managing shared passwords discourages good password practices [18], and that users may be more willing to disclose shared passwords [35].

5.2 Barriers and Counter-Measures

As described in Section 3.4 the attack does not exploit technical vulnerabilities. As it turns out, the security of the target INS installation relies heavily on air-gapping and physical protection while the INS itself is quite open once access is established. It is however commonly accepted that this strategy in itself does not provide sufficient security and that the troubles of keeping such systems up do date will contribute to undermine their security [6,22]. In this section we explore various ways to counter the attack, apart from the air-gapping and the physical security.

Security Mechanisms. The operator station under attack was logged in with a user profile with administrator privileges. This was exploited to copy files to the installation folder of the ECDIS software, to make changes to the registry and to create scheduled tasks. The obvious counter-measure is to create a user profile with a more restricted set of privileges for use in the daily operation of the system. This would force the attacker to device a more sophisticated attack with privilege escalation, unless he/she had access to the administrator password (which will have many of the same issues as the ECDIS maintenance password; see Section 5.1). A similar counter-measure would be to disable the Windows Script Host, which is used to execute VBScript.

The operator station did not have anti-virus software installed. According to statements from experts on maritime cyber security this is quite common for INS installations [4]. On the other hand, it is not certain that an anti-virus program would be sufficient to detect and prevent the attack. We submitted our payload to VirusTotal (www.virustotal.com), and only two out of the 60 antivirus programs the service uses to analyze submissions flagged it as unsafe while the remaining 58 flagged it as clean.

More advanced counter-measures would include mechanisms to prevent the fake DLL from loading, and some mechanism to preserve the integrity of the network traffic such as cryptographic signing by the SINT. (See [25] for further discussion on the latter option.)

Redundancy. The INS implements several redundancy features. While these can be considered safety mechanisms, it is still interesting to see if they have any impact on our attack. There are three source of redundancy: A duplication of the LAN, different sensors providing overlapping information, and sensors with serial connections to the operator stations. The dual LAN does not affect the attack since the Winsock DLL reads the network traffic of both LANs. The INS integrates several sensors in addition to the GPS that are also used for positioning such as heading and speed sensors. If the deviation between the GPS position and dead reckoning based on other sensors exceeds a limit, the ECDIS software will sound a Position Deviation Alarm and this may give an indication that there is something wrong with the integrity of the position data. On the other hand, data from the other sensors are transmitted over the network in the same way as the GPS data. It would probably not be very difficult to have the malware manipulate also these data to remove or reduce the deviation from the manipulated GPS data.

The last of the redundancy mechanisms, however, poses a challenge for the attacker. The INS installation on which we tested the attack also has sensors connected to the operator stations using serial connections (seen as the thin blue lines connecting the operator stations to the SINT in Fig. 1). The ECDIS software compares GPS data received over the LAN with GPS data received over the serial connection and sounds the Position Deviation Alarm if the deviation between the two sources of GPS data exceeds a limit. This alarm was in fact sounded during the test runs of our demonstration. To avoid this deviation, the malware would have to manipulate also serial input to the ECDIS software. Since serial input seems to be handled by DLLs, a strategy similar to the manipulation of network traffic should be feasible. However, it would require a more sophisticated malware as manipulation of data across DLLs would need

to be synchronized in some way. An even more sophisticated (but also less feasible) alternative would be a malware installed on the SINT and manipulating the serial input there. On the other hand, the Position Deviation Alarm will only notify the operator that there are discrepancies in the positioning data and will not reveal this as the result of malware.

6 Conclusions

Maritime cyber security is an emerging field, and still the state of cyber security at sea is shrouded behind speculation and anecdotes. There is a need for studies of the concrete systems and threats which populate the maritime domain. Our contribution reported in this paper is the development and demonstration of a cyber attack against an Integrated Navigation System (INS).

We were able to successfully manipulate the GPS position displayed in the ECDIS application of a vessel during a passage. The attack was tailored to the INS and ECDIS delivered by a specific vendor (i.e. the Original Equipment Manufacturer (OEM) of the INS and ECDIS of the vessel). However, a survey of navigation systems shows that many characteristics are shared across vendors and that the INS and ECDIS studied in this paper are fairly typical [25]. We therefore argue that the principles of the attack will apply also to other INS and ECDIS and that similar attacks can be implemented independently of the specific INS and ECDIS products. The attack demonstrated in this paper can in this sense be seen as a representative of a type of attacks against INS and ECDIS.

To the best of our knowledge this kind of attack is novel. Though, cyber attacks manipulating positions displayed in electronic charts have been suggested earlier [11,23], and a demonstration similar to ours was reported in December 2017 [34]. Being a proof-of-concept, the attack has a certain lack of sophistication. However, the investment was less than two person-months of work including the reconnaissance phase; with more resources invested we believe this kind of attack could pose a real threat. On the positive side, we have seen that a combination of technical security measures, physical protection and security policies in many cases can prevent such attacks.

Developing the attack, rather than merely speculating, serves to make explicit its feasibility, consequences and counter-measures. In this paper, these are discussed from a technical perspective. Feasibility, consequences and countermeasures of the attack as seen from the perspectives of navigation and maritime operations are discussed elsewhere [15]. These discussions highlight how maritime cyber security in some respects is similar to cyber security in general and resembles security of SCADA systems, and how it in some respects is a domain that requires domain specific knowledge of both attacker and defender.

Acknowledgement. The work on which this paper reports was partially funded by the Norwegian Armed Forces CD&E grant EP1710 Concepts for CND in joint operations and partially by the Royal Norwegian Naval Academy R&D grant. We want to thank

the provider of the INS and its representatives for supporting the project, and the owner of the vessels and their crews for facilitating our reconnaissance and testing.

References

1. Adams, A., Sasse, M.A.: Users are not the enemy. Commun. ACM **42**(12), 41–46 (1999)
2. Auriemma, L.: Proxocket (2012), http://aluigi.altervista.org/mytoolz.htm#proxocket
3. Balduzzi, M., Pasta, A., Wilhoit, K.: A security evaluation of AIS Automated Identification System. In: 30th Annual Computer Security Applications Conference (ACSAC 2014). pp. 436–445. ACM (2014)
4. Baraniuk, C.: How hackers are targeting the shipping industry. BBC News (Aug 18, 2017), http://www.bbc.com/news/technology-40685821
5. Betke, K.: The NMEA 0183 Protocol (2001)
6. Byres, E.: The air gap: SCADA's enduring security myth. Commun. ACM **58**(8), 29–31 (2013)
7. CyberKeel: Maritime cyber-risks: Virtual pirates at large on the cyber seas (2014)
8. Demchak, C., Patton, K., Tangredi, S.J.: Why are our ships crashing? Competence, overload, and cyber considerations. Center for International Maritime Security (Aug 25, 2017), http://cimsec.org/ships-crashing-competence-overload-cyber-considerations
9. Direnzo, III, J., Drumiller, N.K., Roberts, F.S. (eds.): Issues in Maritime Cyber Security. Westphalia Press (2017)
10. Dyryavyy, Y.: Preparing for Cyber Battleships – Electronic Chart Display and Information Systems Security. NCC Group (2014)
11. Dyryavyy, Y.: Can you hack an ECDIS? United Kingdom Maritime Pilots' Association (Aug 26, 2016), http://ukmpa.org/can-you-hack-an-ecdis-yevgen-dyryavyy/
12. FireEye: Malware Persistence without the Windows Registry (Jul 15, 2010), https://www.fireeye.com/blog/threat-research/2010/07/malware-persistence-windows-registry.html
13. Fitton, O., Price, D., Germond, B., Lacy, M.: The Future of Maritime Cyber Security. Lancaster University (2015)
14. Goward, D.: Mass GPS spoofing attack in Black Sea? The Maritime Executive (Jul 11, 2017), http://maritime-executive.com/editorials/mass-gps-spoofing-attack-in-black-sea
15. Hareide, O.S., Jøsok, Ø., Lund, M.S., Helkala, K., Ostnes, R.: Enhancing navigator competence by demonstrating maritime cyber security. Journal of Navigation (2018), to appear
16. Hareide, O.S., Ostnes, R.: Scan pattern for the maritime navigator. International Journal on Marine Navigation and Safety of Sea Transportation (TransNav) **11**(1), 39–47 (2017)
17. Hutchins, E.M., Clopperty, M.J., Amin, R.M.: Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. In: 6th International Conference on Information Warfare and Security (ICIW 2011)
18. Inglesant, P., Sasse, M.A.: The true cost of unusable password policies: Password use in the wild. In: SIGCHI Conference on Human Factors in Computing Systems (CHI 2010). pp. 383–392. ACM (2010)
19. International Maritime Organization (IMO): Resolution MSC.232(82): Adoption of the revised performance standards for Electronic Chart Display and Information Systems (ECDIS) (2006)
20. International Maritime Organization (IMO): Resolution MSC.252(83): Adoption of the Revised Performance Standard for Integrated Navigation Systems (INS) (2007)

21. International Telecommunication Union, Radiocommunication Sector (ITU-R): Recommendation ITU-R M.1371-5 (02/2014): Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band (2014)
22. Johnson, III, R.E.: Survey of SCADA security challenges and potential attack vectors. In: 2010 International Conference for Internet Technology and Secured Transactions (ICITST 2010). IEEE (2010)
23. Jones, K.D., Tam, K., Papadaki, M.: Threats and impacts in maritime cyber security. Engineering & Technology Reference (Apr 22, 2016)
24. Kugler, L.: Why GPS spoofing is a threat to companies, countries. Commun. ACM **60**(9), 18–19 (2017)
25. Lund, M.S., Gulland, J.E., Hareide, O.S., Jøsok, Ø., Weum, K.O.C.: Integrity of Integrated Navigation Systems. In: International Workshop on Cyber-Physical Systems Security (CPS-SEC 2018), to appear
26. Munro, K.: OSINT from ship satcoms (Oct 13, 2017), https://www.pentestpartners.com/security-blog/osint-from-ship-satcoms
27. Norris, A.: Integrated Bridge Systems Vol 1: Radar and AIS. The Nautical Institute(2008)
28. Pavković, N., Perkov, L.: Social Engineering Toolkit – A systematic approach to social engineering. In: 34th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2011). pp. 1485–1489. IEEE (2011)
29. PJRC: Teensy USB Development Board, https://www.pjrc.com/teensy
30. Psiaki, M.L., Humphreys, T.E.: GPS lies. IEEE Spectr. **58**(8), 26–32, 52–53 (2016)
31. Raymond, E.S.: AIVDM/AIVDO protocol decoding, version 1.52 (Aug 2016), http://catb.org/gpsd/AIVDM.html
32. Russinovich, M.: PsKill v1.16 (Jun 29, 2016), https://docs.microsoft.com/en-us/sysinternals/downloads/pskill
33. U. S. Coast Guard: Special issue on cybersecurity. Proceedings of the Marine Safety& Security Council, the Coast Guard Journal of Safety & Security at Sea **71**(4) (Winter 2014–2015)
34. Wee, V.: Naval Dome exposes vessel vulnerabilities to cyber attack. Seatrade Maritime News (Dec 22, 2017), http://www.seatrade-maritime.com/news/europe/naval-dome-exposes-vessel-operational-vulnerabilities-to-cyber-attack.html
35. Weirich, D., Sasse, M.A.: Pretty good persuasion: A first step towards effective password security in the real world. In: New Security Paradigms Workshop (NSWP 2001). pp. 137–143. ACM (2001)
36. Wikipedia: NMEA 0183 (Mar 11, 2017), https://en.wikipedia.org/w/index.php?title=NMEA 0183&oldid= 769737723