International Journal of Research in Engineering, Science and Management
Volume-2, Issue-11, November-2019
www.ijresm.com | ISSN (Online): 2581-5792

Preventing Man in the Middle Attack Using Machine Learning

T. Raghupathi[1], M. Sivabalan[2], S. S. Jeganath[3], K. Muthamil Sudar[4]
[1,2,3]*UG Scholar, Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Virudunagar, India*
[4]*Assistant Professor, Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Virudunagar, India*

***Abstract*: The nature of the wireless form leaves it unprotected to intentional interference attacks, it is referred as jamming. Typically, jamming is an external threat representation. However, adversaries with internal knowledge of protocol description and network secrets can launch low-effort jamming attacks that are difficult to recognize and counter. In this work, we address the issue of selective jamming attacks in wireless networks. In these attacks, the attacker is active only for a short period of time, selectively targeting messages of high importance. We embellish the advantages of selective jamming in terms of network staging mortification, and opponent effort by presenting two case studies; a selective pounce, on TCP and one on routing. We show that selective jamming attacks can be launched by execute real-time packet classification at the physical layer. To mitigate this pounce, we develop three schemes that intercept real-time packet classification by combining cryptography primitives with physical-layer attributes. We inspect the security of our procedure and evaluate their computational and communication overhead.**

We propose to use decision tree machine learning to improve the network status without considering congestion control mechanisms over the recent environments, even though DT presents a tree-based graph for prediction and classification. Therefore, DT should be developed for the prediction of the optimal congestion control mechanism among current congestion control transport protocol mechanisms. Despite these proposed methods, most studies only implemented machine learning capabilities in classifiers to improve network status without considering the current mechanisms required to adapt to new environments.

***Keywords*: Machine Learning**

1. Introduction

Wireless networks rely on the uninterrupted availability of the wireless medium to interconnect participating nodes. However, the open nature of this medium leaves it vulnerable to multiple security threats. Anyone with a transceiver can eavesdrop on wireless transmissions, inject spurious messages, or jam legitimate ones. While eavesdropping and message injection can be prevented using cryptographic methods, jamming attacks are much harder to counter. Typically, jamming attacks have been considered under an external threat model, in which the jammer is not part of the network.

The adversary exploits his internal knowledge for launching selective jamming attacks in which specific messages of "high importance" are targeted [3]. For example, a jammer can target route-request/route-reply messages at the routing layer to prevent route discovery, or target TCP acknowledgments in a TCP session to severely degrade the throughput of an end-to-end flow. To launch selective jamming attacks, the adversary must be capable of implementing a "classify-then-jam" strategy before the completion of a wireless transmission. Such strategy can be actualized either by classifying transmitted packets using protocol semantics, or by decoding packets on the fly [4].

2. Characteristics

It is a java based tool that can run on any platform and can generate TCL scripts for wired and Wireless scenarios for NS2.Main features of NSG are:

- Creating Wired and wireless nodes by drag and drop.
- Creating Simplex and Duplex links for wired network
- Creating Grid, Random and Chain topologies.
- Creating TCP and UDP agents. Also supports TCP
- Tahoe, TCP Reno, TCP New-Reno and TCP Vegas.
- Supports Ad Hoc routing protocols such as DSDV,
- AODV, DSR and TORA.
- Supports FTP and CBR applications.
- Supports node mobility.
- Setting the packet size, start time of simulation, end
- Time of simulation, transmission range and interference
- Range in case of wireless networks, etc.
- Setting other network parameters such as bandwidth, etc. for wireless scenarios.

3. Existing system

Jamming attacks are much harder to counter and more security problems. They have been shown to actualize severe Denial-of-Service (DoS) attacks against wireless networks [16]. In the simplest form of jamming, the adversary interferes with the reception of messages by transmitting a continuous

jamming signal, or several short jamming pulses jamming attacks have been considered under an external threat model, in which the jammer is not part of the network [18]. Under this model, jamming strategies include the continuous or random transmission of high power interference signals.

A. Disadvantage

1. They have been shown to actualize severe Denial-of-Service (DoS) attacks against wireless networks [7].
2. The simplest form of jamming, the adversary interferes with the reception of messages by transmitting a continuous jamming signal [4].
3. Jamming strategies include the continuous or random transmission of high power interference signals.

4. Proposed system

In this project, we address the problem of jamming under an internal threat representation. We consider a sophisticated adversary who is aware of network secrets and the implementation details of network protocols at any layer in the network stack [6]. The adversary exploits his internal knowledge for launching selective jamming attacks in which specific messages of "high importance" are targeted. We also use puzzle hiding scheme, to hide packet information before sending packet to the receiver. This scheme is used for randomize the packet for security purpose.

A. Advantage

- The adversary exploits his internal knowledge for launching selective jamming attacks in which specific messages of "high importance" are targeted.
- Jammer may decode the first few bits of a packet for recovering useful packet identifiers such as packet type, source and destination address using decision tree [16].
- Communication is easy and fast between people because of implementing new scheme.
- Security is more complicated in many application, but we overcome this problem using puzzle hiding scheme [23].

5. System architecture

A system architecture or systems architecture is the computational design that defines the structure and/or behavior of a system. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system.

NS2 provides users with an executable command "ns" which takes one input argument, the name of a Tcl simulation scripting file. In most cases, a simulation trace file is created and is used to plot graph and/or to create animation. NS2 consists of two key languages: CCC and Object-oriented Tool Command Language (OTcl). While the CCC defines the internal mechanism (i.e., a backend) of the simulation, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend). The CCC and the OTcl are linked together using TclCL.

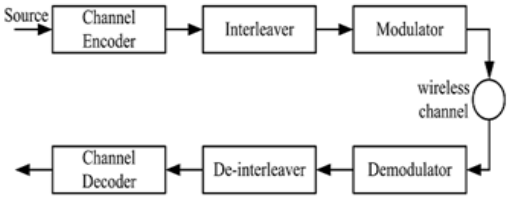

Fig. 1. System architecture

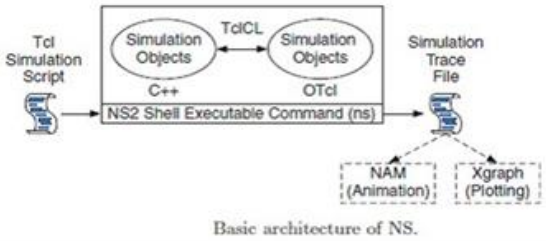

Fig. 2. NS2 architecture

6. Modules description

Real Time Packet Classification: At the PHY layer, a packet m is encoded, interleaved, and modulated before it is transmitted over the wireless channel [13]. At the receiver, the signal is demodulated, deinterleaved, and decoded, to recover the original packet m.

Moreover, even if the encryption key of a hiding scheme were to remain secret, the static portions of a transmitted packet could potentially lead to packet classification. This is because for computationally-efficient encryption methods such as block encryption, the encryption of a prefix plaintext with the same key yields a static ciphertext prefix [16]. Hence, an adversary who is aware of the underlying protocol specifics (structure of the frame) can use the static cipher text portions of a transmitted packet to classify it.

Selective Jamming Module: We illustrate the impact of selective jamming attacks on the network performance. Implement selective jamming attacks in two multi-hop wireless network scenarios [12]. In the first scenario, the attacker targeted a TCP connection established over a multi-hop wireless route. In the second scenario, the jammer targeted network-layer control messages transmitted during the route establishment process selective jamming would be the encryption of transmitted packets (including headers) with a static key. However, for broadcast communications, this static decryption key must be known to all intended receivers and hence, is susceptible to compromise. An adversary in possession of the decryption key can start decrypting as early as the reception of the first ciphertext block.

Cryptographic Puzzle Hiding Scheme (CPHS): We present a packet hiding scheme based on cryptographic puzzles. The main idea behind such puzzles is to force the recipient of a puzzle execute a pre-defined set of computations before he is able to extract a secret of interest. The time required for obtaining the solution of a puzzle depends on its hardness and the computational ability of the solver. The advantage of the puzzle based scheme is that its security does not rely on the PHY layer parameters. However, it has higher computation and communication overhead [25].

7. System implementation

Network Simulator (Version 2), widely known as NS2, is simply an event-driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors. Due to its flexibility and modular nature, NS2 has gained constant popularity in the networking research community since its birth in 1989. Ever since, several revolutions and revisions have marked the growing maturity of the tool, thanks to substantial contributions from the players in the field. Among these are the University of California and Cornell University who developed the REAL network simulator,1 the foundation on which NS is invented. Since 1995 the Defense Advanced Research Projects Agency (DARPA) supported the development of NS through the Virtual Inter Network Testbed (VINT) project [10]. 2 Currently the National Science Foundation (NSF) has joined the ride in development. Last but not the least, the group of researchers and developers in the community are constantly working to keep NS2 strong and versatile. Again, the main objective of this book is to provide the readers with insights into the NS2 architecture. This chapter gives a brief introduction to NS2. NS2 Beginners are recommended to go through the detailed introductory online resources.

Network simulation (NS) is one of the types of simulation, which is used to simulate the networks such as in MANETs, VANETs etc. It provides simulation for routing and multicast protocols for both wired and wireless networks. NS is licensed for use under version 2 of the GNU (General Public License) and is popularly known as NS2. It is an object-oriented, discrete event-driven simulator written in C++ and Otcl/tcl.

NS-2 can be used to implement network protocols such as TCP and UPD, traffic source behavior such as FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ, routing algorithms and many more. In ns2, C++ is used for detailed protocol implementation and Otcl is used for the setup. The compiled C++ objects are made available to the Otcl interpreter and in this way, the ready-made C++ objects can be controlled from the OTcl level.

A. Design and considerations

- It is a discrete event simulator for networking research.
- It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, https and DSR.
- It simulates wired and wireless network.
- It is primarily Unix based.
- Uses TCL as its scripting language.
- Otcl: Object oriented support.
- Tclcl: C++ and otcl linkage.
- Discrete event scheduler.

B. Design

NS2 stands for Network Simulator Version 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks [19].

NS2 uses OTcl to create and configure a network, and uses C++ to run simulation. All C++ codes need to be compiled and linked to create an executable file.

Use OTcl

- For configuration, setup, or one-time simulation, or
- To run simulation with existing NS2 modules.
- This option is preferable for most beginners, since it does not involve complicated internal mechanism of NS2.
- Unfortunately, existing NS2 modules are fairly limited. This option is perhaps not sufficient for most researchers.
- Use C++
- When you are dealing with a packet, or - when you need to modify existing NS2 modules.

This option perhaps discourages most of the beginners from using NS2. This book particularly aims at helping the readers understand the structure of NS2 and feel more comfortable in modifying NS2 modules [26].

C. Considerations

The following constraints must be kept in mind while developing the design:

- Power supply should be cut off when module not in use.
- Module has to be installed in a place where there is availability of strong network for notifications.
- The food items have to be placed in their respective slots.
- The design must be applicable any existing refrigerator.
- “Embedded C“ was the effective option among others and is chosen as programming language.
- The output is shown on a circuit board which consists of the smart refrigeration module. LED“s are placed at various places to indicate the flow of execution.

The possible source of error includes replacing of food items without using them leading false notification. To recover from

such an error, the module could be reset or dedicated application could be developed.

D. Block diagram

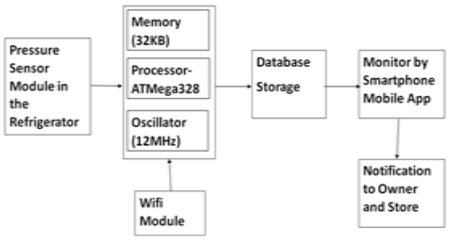

Fig. 3. Block diagram

8. Results

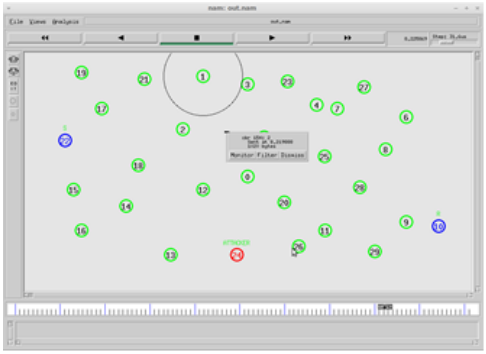

Fig. 4. Output of MITM

Fig. 4. shows that there are sender, receiver and an attacker. The sender sends data to the receiver without any interruption of the attacker.

9. Conclusion

We addressed the problem of selective jamming attacks in wireless networks. We considered an internal adversary model in which the jammer is part of the network under attack, thus being aware of the protocol specifications and shared network secrets. We showed that the jammer can classify transmitted packets in real time by decoding the first few symbols of an ongoing transmission. We evaluated the impact of selective jamming attacks on network protocols such as TCP and routing. Our findings show that a selective jammer can significantly impact performance with very low effort. We developed three schemes that transform a selective jammer to a random one by preventing real-time packet classification. Our schemes combine cryptographic primitives such as commitment schemes, cryptographic puzzles, and all-or-nothing transformations (AONTs) with physical layer characteristics. We analyzed the security of our schemes and quantified their computational and communication overhead.

10. Future work

In future, an external selective jammer who targets various control packets at the MAC layer. To perform packet classification, the adversary exploits inter-packet timing information to infer eminent packet transmissions. We proposed the estimation of the probability distribution of inter-packet transmission times for different packet types based on network traffic analysis. Future transmissions at various layers were predicted using estimated timing information. Using their model, the authors proposed selective jamming strategies for well-known sensor network MAC protocols. We may consider several packet identifiers for encrypted packets such as packet size, precise timing information of different protocols, and physical signal sensing. To prevent selectivity, the unification of packet characteristics such as the minimum length and inter-packet timing was proposed. Similar packet classification techniques were investigated.

References

[1] V. Anantharam and S. Verdu, “Bits through queues,” IEEE Trans. Inf. Theory, vol. 42, no. 1, pp. 4–18, Jan. 1996.
[2] G. Morabito, “Exploiting the timing channel to increase energy efficiency in wireless networks,” IEEE J. Sel. Areas Commun., vol. 29, no. 8, pp. 1711–1720, Sep. 2011.
[3] L. Galluccio, G. Morabito, and S. Palazzo, “TC-Aloha: A novel access scheme for wireless networks with transmit-only nodes,” IEEE Trans.Wireless Commun., vol. 12, no. 8, pp. 3696–3709, Aug. 2013.
[4] W. Xu, W. Trappe, and Y. Zhang, “Anti-jamming timing channels for wireless networks,” inProc. 1st ACM Conf. Wireless Netw. Security, 2008, pp. 203–213.
[5] S. D’Oro, L. Galluccio, G. Morabito, and S. Palazzo, “Efficiency analysis of jamming-based countermeasures against malicious timing channel in tactical communications,” in Proc. IEEE ICC, 2013, pp. 4020–4024.
[6] W. Xu, K. Ma, W. Trappe, and Y. Zhang, “Jamming sensor networks: Attack and defense strategies,”I EEE Netw., vol. 20, no. 3, pp. 41–47, May/Jun. 2006.
[7] R. Saranyadevi, M. Shobana, and D. Prabakar, “A survey on preventing jamming attacks in wireless communication,” Int. J. Comput. Appl., vol. 57, no. 23, pp. 1–3, Nov. 2012.
[8] R. Poisel, Modern Communications Jamming Principles and Techniques. Norwood, MA, USA: Artech House, 2004, ser. Artech House information warfare library.
[9] R.-T. Chinta, T. F. Wong, and J. M. Shea, “Energy-efficient jamming attack in IEEE 802.11 MAC,” inProc. IEEE MILCOM, 2009, pp. 1–7.
[10] Y. W. Law, L. Van Hoesel, J. Doumen, P. Hartel, and P. Havinga, “Energy efficient link-layer jamming attacks against wireless sensor network MAC protocols,” inProc. 3rd ACM Workshop Security Ad Hoc Sensor Netw., 2005, pp. 76–88.
[11] Transmission control protocol. RFC 793, RFC Editor, September 1981. https://tools.ietf.org/html/rfc793.Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. ACM Trans. Inf. Syst. Secur., 3(4):262–294, Nov. 2000.
[12] Cisco 2014 annual security report. Technical report, Cisco, 2014. http://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2014_ASR.pdf.
[13] Ponemon 2014 ssh security vulnerability report. Technical report, Venafi, 2014. https://www.venafi.com/assets/pdf/Ponemon_2014_SSH_Security_Vulnerability_Report.pdf.
[14] Mcafee labs, threats report. Technical report, Intel Security, March 2016. https://www.mcafee.com/us/resources/reports/rp-quarterly-threats-mar-2016.pdf.
[15] Worldwide infrastructure security report. Technical report, ARBOR Networks, The security division of netscout, 2016. https://www.arbornetworks.com/images/documents/WISR2016_EN_Web.pdf.
[16] H. Abdi and L. J. Williams. Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4):433–459, 2010.

[17] D. Al Abri. Detection of mitm attack in lan environment using payload matching. In Industrial Technology (ICIT), 2015 IEEE International Conference on, pages 1857–1862. IEEE, 2015.
[18] M. Al-Hemairy, S. Amin, and Z. Trabelsi. Towards more sophisticated arp spoofing detection/prevention systems in lan networks. In Current Trends in Information Technology (CTIT), 2009 International Conference on the, pages 1–6. IEEE, 2009.
[19] E. Alata, V. Nicomette, M. Kaˆaniche, M. Dacier, and M. Herrb. Lessons learnedfrom the deployment of a high-interaction honeypot. In Dependable Computing Conference, 2006. EDCC'06. Sixth European, pages 39–46. IEEE, 2006.
[20] J. A. ´Alvarez-Jareˇno, E. Badal-Valero, J. M. Pav´ıa, et. al. Using machine learning for financial fraud detection in the accounts of companies investigated for money laundering. Technical report, 2017.
[21] E. B. Claise. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. Request for Comments: 5101, RFC Editor, January 2008.
[22] J. M. Beaver, C. T. Symons, and R. E. Gillen. A learning system for discriminating variants of malicious network traffic. In Proceedings of the Eighth Annual
[23] Cyber Security and Information Intelligence Research Workshop, CSIIRW '13, pages 23:1–23:4, New York, NY, USA, 2013. ACM.
[24] J. Bennett, S. Lanning, et al. The netflix prize. In Proceedings of KDD cup and workshop, volume 2007, page 35. New York, NY, USA, 2007.
[25] M. L. Berenson, M. Goldstein, and D. Levine. Intermediate Statistical Methods and Applications: A Computer Package Approach 2nd Edition. Prentice Hall,1983.
[26] C. M. Bishop. Pattern recognition. Machine Learning, 128:1–58, 2006.
[27] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference.
[28] C. D. Brown and H. T. Davis. Receiver operating characteristics curves and related decision measures: A tutorial. Chemometrics and Intelligent Laboratory Systems, 80(1):24–38, 2006.
[29] S. Byers, A. D. Rubin, and D. Kormann. Defending against an internet based attack on the physical world. ACM Transactions on Internet Technology (TOIT), 4(3):239–254, 2004.
[30] C. Calvert, T. M. Khoshgoftaar, C. Kemp, and M. M. Najafabadi. Capturing man in the middle attack traffic on a live network environment. In 22nd ISSAT International Conference on Reliability and Quality in Design, pages 203–209, 2016.