International Journal of Computer Science Engineering & Technology
IJCSET
APPROVED BY
NATIONAL SCIENCE LIBRARY (NSL)
NATIONAL INSTITUTE OF SCIENCE COMMUNICATION AND INFORMATION RESOURCES (NISCAIR)
COUNCIL OF SCIENTIFIC AND INDUSTRIAL RESEARCH (CSIR)- NEW DELHI INDIA .
ISSN :2455-9091

STEALTHY EXPLOITS ATTACKS ON WIRELESS SENSOR NETWORKS

[1] P.E. Elango, [2] S. Subbaiah
[1] Ph.D Research Scholar, [2] Assistant Professor,
[1] PG & Research Dept of Computer Science, [2] Department of Computer Applications,
[1] Periyar University, [2] Vivekananda College of Arts & Science (Autonomous),
[1, 2] Salem, T.N, India.

Abstract - As we stated earlier, spying is an invasion of privacy that can lead to serious repercussions if the data collected lands into unscrupulous hands. We have demonstrated the disastrous effects of such malware to the host network by building Spy-Sense, the first instance of a spyware tool capable of compromising a sensor network's confidentiality and functionality. Spy-Sense is undetectable, hard to recognize and get rid of, and once activated; it runs in a discrete background operation without interfering or disrupting normal network operation. It provides the ability of executing stealthy exploit sequences that can be used in a variety of attacks ranging from retrieving or manipulating sensitive network data to shutting down a node entirely. In this paper, wireless sensor network security is an important research direction and tools like the current ones may be used in coming up with even more attractive solutions for defending these types of networks.

Keywords: [Spy-Sense, Wireless sensor network, Stealthy, spyware, malicious.]

1. INTRODUCTION

On investigating the depth a new set of memory related vulnerabilities that can be exploited by an adversary for penetrating the security profile of a wireless sensor network. We showed how she can manipulate the existence of a software-based hole(i.e. buffer overflow) for smashing the call stack and intruding a remote node over the radio channel. Then, she can inject malicious programs in order to take full control of a node, change and/or disclose its security parameters upon will. Continuing our work on studying this new threat model (from the attacker's point of view), we move one step further and show how an adversary can perform a code injection attack for permanently injecting spying exploits in the remote nodes. Spying is an invasion of privacy that can lead to serious repercussions if the data collected lands into unscrupulous hands. Therefore, it constitutes a severe threat that is usually overlooked in the design of secure sensor network applications. Motivated by this unexplored security aspect, in this chapter we demonstrate Spy-Sense, a spyware tool that can be useful not only in highlighting the importance of defending sensor network applications against permanent code injection attacks but also in studying the severity of their effects on the sensor network itself. This in turn can lead to the development of more secure applications and better detection/prevention mechanisms. Spy-Sense permits remote injection, through extraordinarily made messages, of different code exploits in the core of every hub in a sensor network. Once infused, it is imperceptible, difficult to perceive and dispose of (as it stays inactive in an unused memory area), and when actuated, it runs in discrete foundation mode without meddling or upsetting typical network exercises. It enables an aggressor to undermine network security through the execution of infused stealthy exploits. Exploits are arrangements of machine code directions that cause unintended conduct to happen on the host sensor.

The intuition behind this work is to introduce the notion of spyware programs in sensor networks and highlight their disastrous effects on their security profile in terms of functionality, content and transactional confidentiality. Content confidentiality is to ensure that no external entity cans infer the meaning of the messages being sent whereas transactional confidentiality involves preventing adversaries from learning information based on message creation and flow within the network. Our tool is capable of threatening all of the above since even in its most benign form, it can simply consume CPU cycles and network bandwidth. When utilized fully, it can lead to stolen cryptographic material and other critical application data, breaches in privacy, and the creation of “backdoor “entries that adversaries can use to target the network with more direct attacks such as Sinkhole attacks, Denial of Service attacks, Wormholes, etc. As the name recommends, Spy-Sense is malicious software that "spies" on sensor hub exercises and transfers gathered data back to the enemy. It can introduce remotely, covertly, and without assent, various stealthy exploits for undermining the network's security profile. As we referenced before, instances of exploits incorporate information control, splitting and network damage. As the all out size of these exploits (312 bytes) is little, Spy-Sense can be effectively and quickly infused into the hubs of a sensor network.

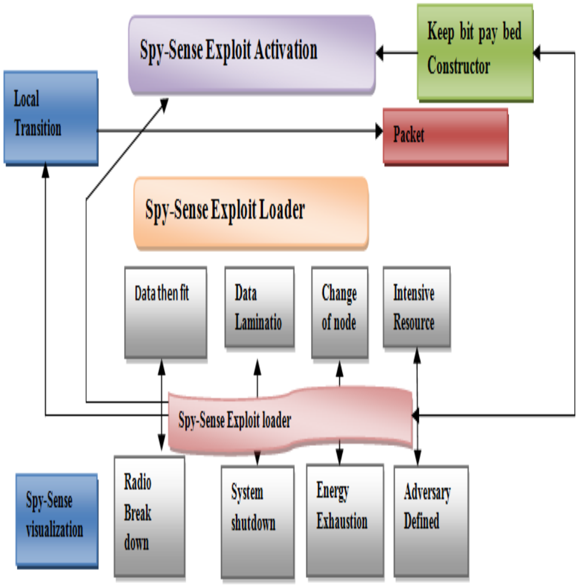

Figure 1: Spy-Sense spyware Architecture Layout

Commonly, a sensor hub is undermined through a software helplessness (e.g., support flood, group string indicated, number flood, and so on.) that permits arrangements of code guidelines to be infused and put away anyplace in the bit's memory. As we portrayed in, since all sensor hubs execute a similar program picture and hold a similar memory addresses for specific activities (as the aftereffect of just static memory assignment support), finding such a helplessness can leave the whole network presented to abuse injection and not only a little segment of it. Spy-Sense exploits will dwell in a consistent memory district in the host sensor stage. They can work in stealth mode as they are customized to change and reestablish the progression of the framework's control in such a manner along these lines, that they don't release the fundamental miniaturized scale controller into a flimsy state. These exploits utilize the presence of a vacant, unused and unchecked memory locale saved to be utilized as the pile for dynamic memory distribution. This fills in as an umbrella of althea exploits disguising their reality and dependably sidestepping recognition. Besides, it brings about a lasting endeavor injection; the miniaturized scale controller's fundamental rationale doesn't play out any activities on the pile area, and accordingly, the main method for deleting load substance is by genuinely catching a hub and compelling it to \hard" reset itself. Spy-Sense consequently deals with the development and transmission of the vital message stream for sending every single stacked endeavor. Once infused, exploits will stay inert until initiation. Enactment requires from an enemy to send one last uniquely created packet that diverts the control stream to the start of the infused shell code, so it very well may be executed in stealth mode. Execution can happen the same number of times varying so as to accomplish the gatecrasher's objectives.

Impact to Sensor Networks

The threat that is imposed by Spy-Sense to the host network is that of any spyware program: injected shell codes are hidden, they are difficult to detect and can collect small pieces of information without the knowledge of the network's owners. Spy-Sense can be used for cracking the network and creating “bonnets" of compromised nodes that are commonly controlled by the adversary. This leads not only to possible loss of important data (e.g., cryptographic material, environmental dataset.) but also to intensive resource usage. One of Spy-Sense's most severe effects is data manipulation, the ability to steal and/or modify important or confidential information. Examples include cryptographic keys, transactional data or even private sensitive information in the case of smart environments or assistive healthcare scenarios. An extension to this \spying" behavior is the ability to track and record all network activities. Any data or legless reported back to the adversary are transmitted in stealth mode, through the used communication channel, but in periods of light network traffic in order to look less conspicuous and avoid detection.

2. SPY-SENSE ARCHITECTURE LAYOUT

Spy-Sense depends on an astute segment based framework. The hosted segments are equipped for stacking predefined misuse profiles, infusing them to the focused on network through a straightforward transmission of a progression of uncommonly created messages, accepting and logging of all hub answers that report back mentioned framework data. Its core functionality is based on four main conceptual modules, as depicted in Figure 1. One of the key design goals of Spy-Sense is its wide applicability; it supports exploit injection attacks and compromise of a wide variety of sensor hardware and network protocols. It can exploit all vulnerabilities and weaknesses arising from a specific platform despite the followed memory architecture (Von Neumann and Harvard) since subsequent code injection can be performed in either of them. Furthermore, while capturing and logging of all node replies is performed in real-time, content analysis can be done either online or o²ine. We believe that o²ine analysis provides better way of extracting information regarding network activities and information patterns. In what follows we give a more detailed description of the four basic system components.

Spy-Sense Exploit Loader Component

The adventure loader is answerable for introducing the software by bringing in all predefined abuse profiles that dwell in the Spy-Sense root envelope. Such profiles contain (i) the machine code directions that will be infused into the host sensor hub, and (ii) their emblematic portrayal written in assembly language. Exploit loading and registration can occur anytime during Spy-Sense operation; either upon system boot up or during normal operation by updating the contents of the corresponding storage folder. All exploit code instructions are contained in files and are loaded one at a time. This is the most convenient and platform-independent way for a user to define his/her own exploit profiles that need to be imported in Spy-Sense. Again, new additions can either be performed at boot up timer during system operation.

Spy-Sense Setup Engine

This incredible segment is capable of conveying imported exploits to a chose segment of network hubs. It comes into play once the Spy-Sense Exploit Loader has successfully finished loading and registration of any predefined malicious shell codes. The constructed series of malicious packets are transmitted to the target node in order to inject the selected instruction sequence into its memory. Fundamental to this operation is the definition of an address pointer, namely $ADDR_{copyTo}$, which points to an appropriate memory address (inside the heap region) where the code will be stored. After the successful completion of the injection process, k bytes (k is multiple of 2) of code will have been copied into the target region. Overall, the exploit payload constructor creates packets consisting of three parts. The first part provides the data for buffer overflow, as well as the memory address (where the buffer of received messages is stored), at which the program flow will be directed. The second part provides the necessary MOV instructions for copying blocks of the exploit code to the heap target region. Finally, the third part provides the BR (anch) command for restoring the original flow. In the first case, the setup engine starts a sequential, transparent transmission of the specially crafted messages created by the payload constructor module. Upon completion, an appropriate message is displayed for informing the user on the result of the injection attempt. In the second case, a preview of all message payloads (that are ready for transmission) is printed to the corresponding exploit information panel. Prior to the selection of any of these actions, it is mandatory for the user to update all the exploit injection process settings: (i) the ID of the targeted sensor node, (ii) the value of ADDRcopyToaddress, and (iii) the memory addresses reserved for holding any \exploit function arguments". Such arguments describe the number of bytes and the target memory address from where/to data will be retrieved/injected, the identifier of the spawned exploit task or the time period that the host node will enter into an intensive resource usage state. Once these settings are configured, the user can successfully start deploying any of the loaded Spy-Sense exploits. Status and additional information regarding the currently running injection process are displayed in real time by the system visualization component.

Spy-Sense Exploit Activation Component

Once the transmission process is completed, the Spy-Sense setup engine has succeeded to remotely inject exploit shell codes into the targeted sensor network. Then, the only step remaining is to activate the malware in order to execute its functions. This is where the exploit activation component comes into play. It handles the last messages that should be sent for actuating a chose endeavor to at least one of the host sensor hubs. The enactment procedure requires the transmission of a progression of uniquely created packets for diverting the program stream to the start of the endeavor shell code, in the heap target region (ADDRstartT r), so that it can be executed. Again, the exploit payload constructor module is responsible for creating such a message stream containing: (i) the values of the selected “exploit function arguments", and (ii) a BR instruction that is executed for setting the instruction pointer to the starting address of the target region, ADDRstartTr. Activation may result to one-time or recursive exploit execution by firing an internal periodic task. In the first case, the targeted exploit returns to an idle state, after execution, and waits for the next activation message. In the second case, a periodic “activation task" is spawned and every time it fires, it signals the exploit payload constructor module to repeat the transmission of the corresponding exploit message stream.

Exploit Analysis & Machine Code Break Down

Spy-Sense (in its current version) provides a list of predefined exploits capable of performing data manipulation, cracking and network damage. Fundamental to a successful exploit injection and activation is the definition of a memory symbol table describing where in the host's memory the injected shell code, along with its "function arguments". The symbol table is a list of all the absolute memory addresses that are used by Spy-Sense Setup engine and are configured by the user before injection. All provided values depend on the binary representation of the program image that is loaded in the sensor node. Once the memory symbol table is finalized, all shell code assembler instructions are ready for injection and execution. The targeted microcontroller register file consists of 16 registers of 16 bits each, numbered from R0 to R15. The first four are reserved by the OS whereas the rest are for general use and will be used by the injected shell code, e.g., holding instruction operands or function arguments. In what follows we will cover the details of all instruction sequences, contained in each one of the malwares, and how they are executed by the host scheduler.

Data Manipulation Exploits

Data manipulation exploits include shell codes for data theft and data modification. Data theft code occupies 114 bytes and, thus, 30 packets will be needed by the setup engine for injecting it into the heap target region. Two functions are involved in the data theft: (i) retrieval of the selected data memory region, and (ii) construction and transmission (back to Spy-Sense) of the appropriate reply message that will hold the extracted information. The code for data modification occupies 56 bytes and, thus, 14 packets will be needed for injecting it. As the name suggests, it gives an adversary the ability to secretly modify the value of an existing memory data structure. This may involve the alteration of either incoming or outgoing information, and can be as small as manipulating a single byte or an entire data stream. Since this kind of data interference may not be that obvious to the system host, such exploits can induce great damage to the targeted network.

Cracking Exploits

Cracking exploits include shell codes for energy exhaustion and manipulation of the host node ID. Energy exhaustion code occupies 102 bytes and, thus, 26 packets will be needed by the setup engine for injecting it into the heap target region. The main logic involves the initiation of unnecessary communications until the host node drains all its energy out.

Algorithm - Data Alteration Exploit - Assembly Code

Data: Memory Symbol Table

Begin

- CMP ≠0, &ADD$_{\text{RexplArg1}}$;
- JZ $ 34;
- CLR R11;
- MOV &ADD$_{\text{RexplArg1}}$,R12;
- MOV ≠270E, r13;
- MOV &ADD$_{\text{RexplArg1}}$,R14;
- MOV R11, R9;
- MOV R9, R8;
- ADD R12, R9
- ADD R13,R8;

End

Network Damage Exploits

Network damage exploits include shell codes for intensive resource usage and radio communication break downs. Resource usage code occupies 22 bytes and, thus, 6 packets will be needed for injecting it into the heap target region. The main logic requires two loop-throughs for consuming CPU cycles. The outer loop is always set to the highest possible 2-byte integer value, ffffh, whereas the inner loop is configurable and defines the actual time spent in this intensive cycle usage state. Algorithm contains the complete assembly code. The requested argument, ADDRexplArg4, holds the time that the host node will be \stuck" at the exploit sustain level (SL) and depends on the value of the inner loop (IL). After experiments, we have found that the average time (in seconds) wasted is given by the expression SL = 0:0062 ¤ IL.

User Defined Exploits

All the above described exploit shell codes are provided by the current version of Spy-Sense. They reside in the corresponding root folder and they are imported by the system exploit loader component. However, it is possible for an adversary to define her own new exploit profiles. This requires the creation of a file, containing all the exploit code instructions, inside the Spy-Sense exploit folder. Further loading and registration will be taken care by the tool either upon system boot up or during normal operation. The path to this folder is configurable and can be altered by the user through the Spy-Sense central page, as depicted.

3. EXPERIMENTAL RESULTS

Spy-Sense exploits will reside in a continuous memory region in the host sensor platform. They can operate in stealth mode as they are programmed to change and restore the flow of the system's control in such a way so that they don't let the underlying micro-controller go into an unstable state. These exploits utilize the presence of a vacant, unused and unchecked memory locale saved to be utilized as the stack for dynamic memory portion. This functions as an umbrella of althea exploits disguising their reality and dependably avoiding recognition. Besides, it brings about a changeless endeavor injection; the small scale controller's fundamental rationale doesn't play out any activities on the stack district, and in this way, the main method for deleting pile substance is by truly catching a hub and driving it to hard reset itself.

No Of Nodes	Existing 1	Existing 2	Propose d
40	29	14	34
80	55	32	69
120	83	45	90
160	125	101	133
200	160	129	182

Table 1: No of nodes identified

Table 1 represented into no of nodes identified in external attack values. SPY- Sense is proposed into this phase. Proposed SPY-Sense is detected the more than external attacks in this phase. So it is better proposed concept of this phase.

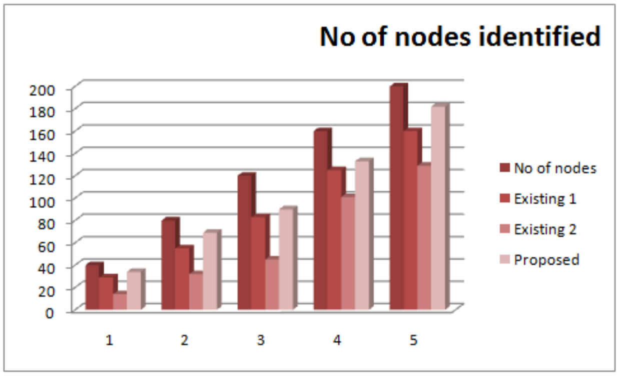

Figure 2: No of nodes identified

Figure 2 is represented into no of nodes identified values in graphs. External attacks find the existing values are high but their SPY-Sense values are detect the lower than among the nodes in the external attacks.

No of nodes	Existing 1	Existing 2	Proposed
50	26	15	35
100	51	33	77
150	122	104	130
200	159	129	167
250	218	200	230

Table 2: Reliability

Table 2 represented into reliability in external attack values. SPY- Sense is proposed into this phase. Proposed SPY- Sense is detected the more than external attacks in this phase. So it is better proposed concept of this phase.

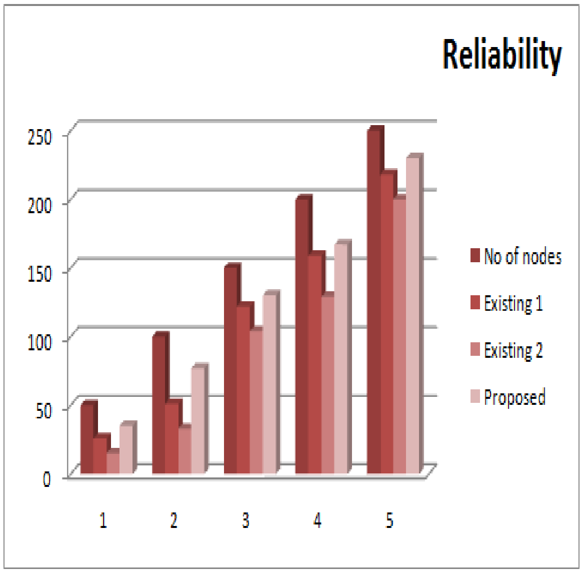

Figure 3: Reliability

Figure 3 is represented into reliability values in graphs. External attacks find the existing values are high but their SPY-Sense values are detect the lower than among the nodes in the external attacks.

No of nodes	Existing 1	Existing 2	Proposed
100	69	47	85
200	156	142	170
300	267	233	288
400	354	329	377
500	451	430	469

Table 3: Consistency

Table 3 represented into consistency in external attack values. SPY- Sense is proposed into this phase. Proposed SPY- Sense is detected the more than external attacks in this phase. So it is better proposed concept of this phase.

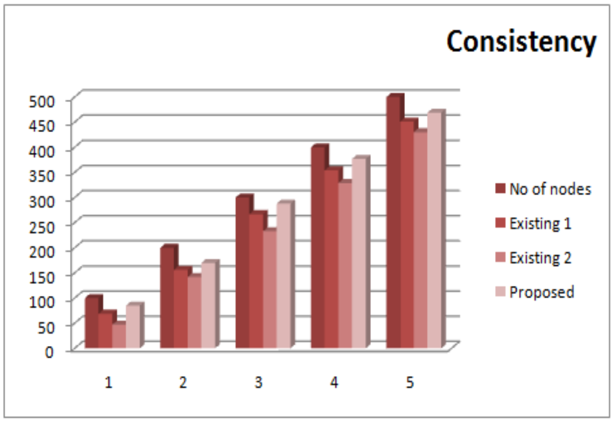

Figure 4: Consistency

Figure 4 is represented into consistency values in graphs. External attacks find the existing values are high but their SPY-Sense values are detect the lower than among the nodes in the external attacks.

No of nodes	Existing 1	Existing 2	Proposed
25	17	22	9
50	36	41	25
75	60	69	47
100	79	95	62
125	99	111	89

Table 4: Error Reporting

Table 4 represented into error reporting in external attack values. SPY- Sense is proposed into this phase. Proposed SPY- Sense is detected the more than external attacks in this phase. So it is better proposed concept of this phase.

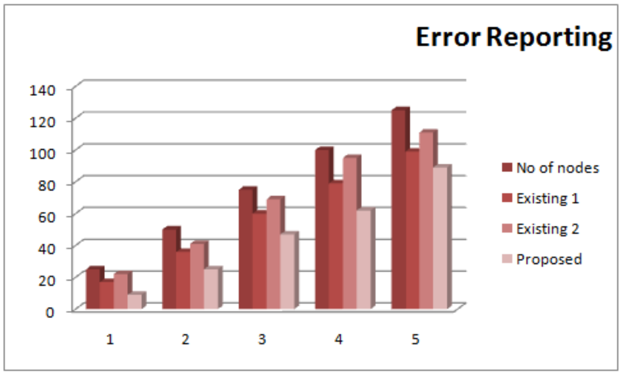

Figure 5: Error Reporting

Figure 5 is represented into error reporting values in graphs. External attacks find the existing values are high but their SPY-Sense values are detect the lower than among the nodes in the external attacks.

No of nodes	Existing 1	Existing 2	Proposed
30	17	26	9
60	38	49	22
90	66	81	58
120	96	113	83
150	135	140	111

Table 5: Traffic Latency

Table 5 represented into traffic latency in external attack values. SPY- Sense is proposed into this phase. Proposed SPY- Sense is detected the more than external attacks in this phase. So it is better proposed concept of this phase.

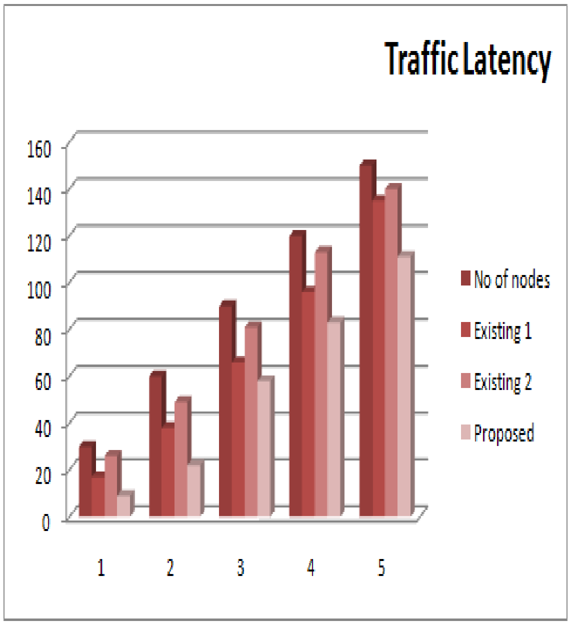

Figure 6: Traffic Latency

Figure 6 is represented into traffic latency values in graphs. External attacks find the existing values are high but their SPY-Sense values are detect the lower than among the nodes in the external attacks.

No of nodes	Existing 1	Existing 2	Proposed
100	61	39	85
250	146	101	197
300	201	177	269
450	333	316	399
500	409	388	454

Table 6 : IDS throughput

Table 6 represented into IDS throughput in external attack values. SPY- Sense is proposed into this phase. Proposed SPY- Sense is detected the more than external attacks in this phase. So it is better proposed concept of this phase.

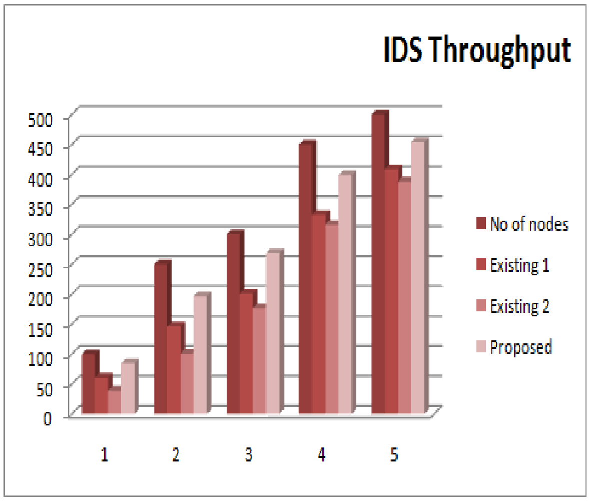

Figure 7: IDS throughput

Figure 7 is represented into IDS throughput values in graphs. External attacks find the existing values are high but their SPY-Sense values are detect the lower than among the nodes in the external attacks.

CONCLUSION

We moved one step further and identified some of the sensor networks vulnerabilities that can be exploited by an attacker for launching permanent code injection attacks and, eventually, spyware programs. As we stated earlier, spying is an invasion of privacy that can lead to serious repercussions if the data collected lands into unscrupulous hands. We have demonstrated the disastrous effects of such malware to the host network by building Spy-Sense, the first instance of a spyware tool capable of compromising a sensor network's confidentiality and functionality. Spy-Sense is undetectable, hard to recognize and get rid of, and once activated; it runs in a discrete background operation without interfering or disrupting normal network operation. It provides the ability of executing stealthy exploit sequences that can be used in a variety of attacks ranging from retrieving or manipulating sensitive network data to shutting down a node entirely. By studying the after-effects of various exploits on the network itself, we wish to motivate a better design of security protocols that can make them even more resilient to tools like Spy-Sense and the one presented in the next chapter. As we highlighted in this paper, wireless sensor network security is an important research direction and tools like the current ones may be used in coming up with even more attractive solutions for defending these types of networks.

REFERENCES

[1] Sunil Gupta,Authentication Framework against “Malicious Attack in Mobile Wireless Sensor Networks”, Vol II, IMECS 2017, March 15 - 17, 2017

[2] Chaudhari H.C. and Kadam L.U,”Wireless Sensor Networks: Security, Attacks and Challenges International Journal of Networking” ,Volume 1, Issue 1, 2011, pp-04-16

[3] Hu, Perrig, and Johnson, “Malicious Node Detection in Wireless Sensor Networks” Waldir Ribeiro Pires J´unior Thiago H. de Paula Figueiredo Hao Chi Wong Antonio A.F. Loureiro

[4] Deepmala Verma, Gajendra Singh, Kailash Patidar, Detection of Vampire Attack in Wireless Sensor Networks , Vol. 6 (4) , 2015, 3313-3317

[5] L. Lamport.” Constructing digital signatures from one-way function”.in technical report SRI-CSL-98, SRI International, Oct. 1979.

[6] Dr. Adwan Yasin ,Kefaya Sabaneh ,”Enhancing Wireless Sensor Network Security using Artificial Neural Network based Trust Model” , Vol. 7, No. 9, 2016

[7] H. Gorine, M. Ramadan Elmezughi, “Security Threats on Wireless Sensor Network Protocols,” 18-19 August 2016

[8] Soram Rakesh Singh , Narendra Babu C R,Improving the “Performance of Energy Attack Detection in Wireless Sensor Networks by Secure forward mechanism”, Volume 4, Issue 7, July 2014

[9] D. I. Curiac, O. Banias, F. Dragan, C. Volosencu and O. Dranga, “Malicious Node Detection in Wireless Sensor Networks Using an Autoregression Technique,” 3rd International Conference on Networking and Services, Athens, 19-25 June 2007, p. 83

[10] DelPHI: “worm hole detection mechanism for ad hoc wireless network proposed” by Hon Sun Chiu and King-Shan Lui in international Symposium on wireless Pervasive Computing ,Phuket, Thailand, 16-18 january 2006.

[11] H.Chen, H.Wu, X.Zhou,”Reputation-based Trust in Wireless Sensor Network”, in IEEE International Conference on Multimedia and Ubiquitous Engineering, 26th -27th April, (MUE’07), 2007, Shanghai.,pp.603-607.

[12] Andriy Stetsko, Lukas Folkman, Vashek Matyas, Neighbor-based” Intrusion Detection for Wireless Sensor Networks”, 6th International Conference on Wireless and Mobile Communications (ICWMC), 2010, pp. 420-425

[13] Abdul Wahid Pavan Kumar,”A Survey On Attacks, Challenges and Security Mechanisms In Wireless Sensor Network”.

[14] International Telecommunications Union (ITU-T), Recommendation X.200 (07/94): Open Systems Interconnection - Basic Reference Model, July 1994.

[15] I. Demirkol, C. Ersoy, and F. Alagoz, MAC protocols for wireless sensor networks: a survey IEEE Communications Magazine, vol. 44, pp. 115{121, Apr. 2006.