Substantiating Anomalies In Wireless Networks Using Outlier Detection Techniques

by
Elankayer Sithirasenan
BScEng (Perad’ya), MSoftEng (Griff)

School of Information and Communication Technology
Science, Environment, Engineering and Technology Group
Griffith University, Gold Coast Campus, Australia

Submitted in fulfilment of the requirements of the degree of Doctor of Philosophy

November 2008
Abstract

With the increasing dependence on Wireless Local Area Networks (WLANs), businesses and educational institutions are in real need of a robust security mechanism. The latest WLAN security protocol, the IEEE 802.11i assures rigid security for wireless networks with the support of IEEE 802.1x protocol for authentication, authorization and key distribution. Nevertheless, users remain skeptical since they lack confidence on the practical trustworthiness of these security mechanisms. In this research we propose a novel Early Warning System (EWS), built on the foundations of IEEE 802.11i security architecture. Our proposed system can effectively detect anomalies, substantiate them, and also identify the basis for such malicious behavior. It has a number of levels of defense to scrutinize malicious behaviors of the wireless network, caused by a range of factors including security issues. Security alerts will be raised only when the legitimacy of abnormal conditions is validated using effective outlier based substantiation techniques.

Timing anomalies can occur due to various conditions including security vulnerabilities in the wireless environment. Hence, detecting and analyzing such anomalies may lead to significant advancement towards the detection of misbehaving wireless hosts. In this view, we have discussed the effectiveness of monitoring and analyzing round trip timing values between every request and response messages during the authentication process of wireless hosts. Further, to enhance the capabilities of our detection mechanism we have also considered the effect of behavioral anomalies of the wireless hosts. Every wireless host that tends to connect to the wireless network exhibits a particular behavior. This behavior may
vary depending on a number of issues including security vulnerabilities. Hence, in this study we have discussed the use of behavioral analysis for detecting abnormal conditions. We have used the standard theoretical/practical behavior profiles developed using a software model of the wireless hosts to compare the actual behavior during a specific authentication process.

Anomalies in the wireless environment can be triggered due to a number of factors including security vulnerabilities, mismanaged or misconfigured wireless hosts, atmospheric conditions, change in security settings etc. Reporting every timing and/or behavioral anomaly as an intrusion can lead to a large number of false positive reporting. Therefore, for effective intrusion detection we must carefully analyze all detected timing and/or behavioral anomalies before raising an alert. Hence, in our study, we have used outlier based data association techniques to substantiate all detected timing and/or behavioral anomalies.

Real time detection of outliers from large multi-level longitudinal data sets of network traces may lead to effective intrusion detection and prevention. Presently, due to lack of fast on-the-fly updating and processing capabilities, Intrusion Detection Systems (IDSs) do not detect all intrusions instantly. Also, achieving dynamic adaptation in real time has been a long standing desire for effective intrusion detection and prevention. Most IDSs cannot adapt their detection mechanism in real time to accommodate illegitimate dynamic changes. Furthermore, analyzing multivariate and latent variable data sets is complex and time consuming. Our proposed system validates anomalies by estimating an outlier score value for misbehaving hosts using outlier based substantiation techniques. Detecting outliers in real time is intricate and challenging. However, our method builds a data cube from the large multi-level longitudinal data sets of network traces to substantiate anomalies in real time. Our parallel data mining technique enables substantiation of anomalies viable and reliable facilitating precise fault reporting.

The two proposed substantiating mechanisms, which use Group Confidence Level (GCL) and Group Outlier Score (GOS) measures were validated using real
A range of experiments enabling various attack scenarios with different authentication methods were carried out. The results obtained with substantiation mechanisms are promising, showing a significant increase in the detection rate.

In the future study, the proposed outlier detection and substantiating mechanisms can be extended to other applications such as detection of credit card fraud, health monitoring systems, Internet security etc.
Statement of Originality

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Signed:
Elankayer Sithirasenan
October 2008
Acknowledgments

The successful completion of this research would not have been feasible if not for the continuous support and encouragement of my supervisor Dr. Vallipuram Muthukkumarasamy. I sincerely thank him for all his advice, enthusiasm and concern. Next, I wish to thank my associate supervisor Prof. Frank Dhene for his valuable & readily support and encouragement, without whom this research would not have been instigated. I also take this opportunity to thank Prof. Todd Eavis at Concordia University in Montreal, Canada for providing us with the initial software for generating and querying the data cubes. I will be disloyal if I don’t take this opportunity to thank Prof. Geoff Dromey for his tremendous support and inspiration from the beginning of my research career at Griffith University (GU).

I also wish to take this opportunity to acknowledge all my teachers at GU for their remarkable support and encouragement. Firstly, I wish to sincerely thank Mr. Don Abel, who was always ready to assist me with an optimistic approach guiding me throughout my association with GU. Next, I wish to earnestly show my gratitude to Dr. Jennifer Gaston, Mr. Bruce Hodgen, Mr. Ian Rose, Mr. Terry Rout and Ms. Angela Tuffly.

Next, I would like to thank my family. Firstly, I sincerely thank my wife Vasugi for her unprecedented support and shouldering family responsibilities throughout my study enabling me to continue my research without any hindrance. Next, I appreciate the love and affection shown by my children Balu, Theju and Lakshmi,
without which this research would be futile. I would also like to take this oppor-
tunity to thank my mother-in-law Mrs. Lalitha Nadarajah for her invaluable support.

Last but not least I would like to thank my sisters, brother and their families for their encouragement and support. I consider this research a tribute to my father Mr. S.C. Elankayer and my deceased mother Mrs. Rajamani Elankayer.
Acronyms

**ABDS**  Anomaly Based Detection System

**ABID**  Anomaly Based Intrusion Detection

**ADAM**  Audit Data Analysis and Mining

**AES**  Advanced Encryption Standard

**AP**  Access Point

**APL**  A Programming Language

**ARP**  Address Resolution Protocol

**BSS**  Basic Service Set

**BT**  Behavior Tree

**IBSS**  Independent Basic Service Set

**CBC-MAC**  Cipher Block Chaining Message Authentication Code

**CCA**  Clear Channel Assessment

**CCK**  Complimentary Code Keying

**CCM**  Counter with CBC-MAC

**CCMP**  Counter-mode/CBC-MAC Protocol
**CRC**  Cyclic Redundancy Checksum

**CSP**  Communicating Sequential Processes

**CTL**  Computation Tree Logic

**CTR**  Counter Mode

**CTS**  Clear-To-Send

**DBT**  Design Behavior Tree

**DCF**  Distributed Coordination Function

**DHCP**  Dynamic Host Configuration Protocol

**DIAMETER**  Authentication, Authorization and Accounting Protocol

**DoS**  Denial of Service

**DSSS**  Direct Sequence Spread Spectrum

**EAP**  Extensible Authentication Protocol

**EAPOL**  Extensible Authentication Protocol Over Local Area Network

**EAP-TLS**  Extensible Authentication Protocol - Transport Layer Security

**ESS**  Extended Service Set

**EWS**  Early Warning System

**FBI**  Federal Bureau of Investigations

**FHSS**  Frequency Hopping Spread Spectrum

**FIN**  Final - no more data

**FTP**  File Transfer Protocol
**GOS** Group Outlier Score

**GCL** Group Confidence Level

**GTK** Group Transient Key

**GSE** Genetic Software Engineering

**HSRP** Hot Standby Routing Protocol

**HTTP** Hypertext Transfer Protocol

**ICMP** Internet Control Message Protocol

**ICV** Integrity Check Value

**IDS** Intrusion Detection System

**IEEE** Institute of Electrical and Electronics Engineers

**IP** Internet Protocol

**IPsec** Internet Protocol Security

**IRPAS** Internetwork Routing Protocol Attack Suite

**IV** Initialization Vector

**LAN** Local Area Network

**LDAP** Lightweight Directory Access Protocol

**LEAP** Cisco LEAP protocol

**LOF** Local Outlier Factor

**LTL** Linear Temporal Logic

**MAC** Media Access Control
MD5  Message-Digest algorithm 5
MIC  Message Integrity Code
MitM  Man-in-the-Middle
MSDU  MAC Service Data Unit
NAV  Network Allocation Vector
NIC  Network Interface Card
OCB  Offset Codebook Mode
OFDM  Orthogonal Frequency Division Multiplexing
OLAP  On Line Analytical Processing
OSI  Open Systems Interconnection
OSPF  Open Shortest Path First
PAE  Port Access Entity
PDA  Personal Digital Assistant
PEAP  Protected Extensible Authentication Protocol
PHY  PHYsical Layer
PIN  Personal Identification Number
PMK  Pairwise Master Key
PMKID  Pairwise Master Key Identifier
PSK  Pre-Shared Key
PTK  Pairwise Transient Key
RADIUS  Remote Authentication Dial In User Service

RBT   Requirements Behavior Tree

RC4   Rivest Cipher 4 (Ron's Code)

RF    Radio Frequency

RFF   Radio Frequency Fingerprint

RIP   Routing Information Protocol

RSNA  Robust Security Network Association

RSN   IE Robust Security Network Information Element

RSS   Received Signal Strength

RTS   Request-To-Send

RTT   Round Trip Timing

SAL   Symbolic Analysis Laboratory

SET   Secure Electronic Transaction

SIFS  Short Inter-Frame Space

SSID  Service Set IDentifier

STA   Mobile Stations

SYN   Synchronize - sequence numbers

TCP   Transmission Control Protocol

TIM   Traffic Indication Map

TK    Temporal Key
**TKIP** Temporal Key Integrity Protocol

**TLS** Transport Layer Security

**TSN** Transient Security Network

**TTLS** Tunteled TLS Authentication Protocol

**UDP** User Datagram Protocol

**UMP** User Mobility Profiles

**VPN** Virtual Private Network

**WEP** Wired Equivalent Protocol

**WiFi** Wireless Fidelity

**WLAN** Wireless Local Area Network

**WPA** WiFi Protected Access

**WPA2** WPA with IEEE 802.11i Support

**WRAP** Wireless Robust Authenticated Protocol
Contents

Abstract i

Statement of Originality v

Acknowledgments vii

Acronyms ix

1 Introduction 1
  1.1 Background .................................................. 1
    1.1.1 Security Requirements ................................. 1
    1.1.2 Wireless Security Growth ............................ 2
  1.2 Malicious Behavior of Wireless Networks ...................... 6
  1.3 Substantiating Anomalies .................................. 7
  1.4 OLAP and the Data Cube ................................ 8
  1.5 Research Contribution ..................................... 11
  1.6 Dissertation Outline .................................... 12

2 Literature Review 15
  2.1 Introduction ................................................ 15
    2.1.1 Chapter Overview ...................................... 16
  2.2 Wireless LANs .............................................. 17
  2.3 Wireless Security Standards ................................. 19
    2.3.1 WEP .................................................. 19
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>IEEE 802.11i Software Model</td>
<td>68</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Component behavior projection</td>
<td>71</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Model Checking</td>
<td>73</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Formal Analysis</td>
<td>77</td>
</tr>
<tr>
<td>3.4</td>
<td>The Proposed Method</td>
<td>88</td>
</tr>
<tr>
<td>3.4.1</td>
<td>The Early Warning System</td>
<td>89</td>
</tr>
<tr>
<td>3.5</td>
<td>Conclusion</td>
<td>90</td>
</tr>
<tr>
<td>4</td>
<td>Early Warning System</td>
<td>93</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>93</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Chapter Overview</td>
<td>94</td>
</tr>
<tr>
<td>4.2</td>
<td>The EWS</td>
<td>94</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Anomaly Detection</td>
<td>95</td>
</tr>
<tr>
<td>4.3</td>
<td>Substantiation</td>
<td>98</td>
</tr>
<tr>
<td>4.3.1</td>
<td>OLAP and Data Views</td>
<td>99</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Data Views Vs Security Threats</td>
<td>101</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Outlier Detection</td>
<td>103</td>
</tr>
<tr>
<td>4.4</td>
<td>Experimental Platform</td>
<td>109</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Timing Anomaly Detection</td>
<td>112</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Behavioral Anomaly Detection</td>
<td>116</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Outlier Detection</td>
<td>118</td>
</tr>
<tr>
<td>4.5</td>
<td>Conclusion</td>
<td>122</td>
</tr>
<tr>
<td>5</td>
<td>Anomaly Detection</td>
<td>123</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>123</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Chapter Overview</td>
<td>124</td>
</tr>
<tr>
<td>5.2</td>
<td>EAP Authentication</td>
<td>125</td>
</tr>
<tr>
<td>5.2.1</td>
<td>EAP LEAP</td>
<td>125</td>
</tr>
<tr>
<td>5.2.2</td>
<td>EAP PEAP</td>
<td>127</td>
</tr>
<tr>
<td>5.2.3</td>
<td>EAP TLS</td>
<td>128</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>5.3</td>
<td>Timing Anomaly</td>
<td>129</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Normal Timings</td>
<td>130</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Abnormal Timings</td>
<td>132</td>
</tr>
<tr>
<td>5.4</td>
<td>Behavior Anomaly</td>
<td>138</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Normal Behavior</td>
<td>138</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Abnormal Behavior</td>
<td>141</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusion</td>
<td>143</td>
</tr>
<tr>
<td>6</td>
<td>Substantiation</td>
<td>145</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>145</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Chapter Overview</td>
<td>146</td>
</tr>
<tr>
<td>6.2</td>
<td>Group Confidence</td>
<td>146</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Normal Operation</td>
<td>146</td>
</tr>
<tr>
<td>6.2.2</td>
<td>During DoS Attack</td>
<td>150</td>
</tr>
<tr>
<td>6.2.3</td>
<td>During Replay Attack</td>
<td>154</td>
</tr>
<tr>
<td>6.3</td>
<td>Group Outlier Score</td>
<td>157</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Normal Conditions</td>
<td>158</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Replay Attack</td>
<td>161</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Perceptions of GOS values</td>
<td>165</td>
</tr>
<tr>
<td>6.3.4</td>
<td>Malicious Association</td>
<td>167</td>
</tr>
<tr>
<td>6.3.5</td>
<td>Malicious Authentication</td>
<td>170</td>
</tr>
<tr>
<td>6.3.6</td>
<td>Session Hijack</td>
<td>172</td>
</tr>
<tr>
<td>6.4</td>
<td>Effectiveness of GOS</td>
<td>174</td>
</tr>
<tr>
<td>6.5</td>
<td>Conclusion</td>
<td>176</td>
</tr>
<tr>
<td>7</td>
<td>Conclusions and Future Work</td>
<td>181</td>
</tr>
<tr>
<td>7.1</td>
<td>Anomaly Detection</td>
<td>182</td>
</tr>
<tr>
<td>7.2</td>
<td>Substantiation</td>
<td>183</td>
</tr>
<tr>
<td>7.3</td>
<td>Conclusions</td>
<td>186</td>
</tr>
<tr>
<td>7.4</td>
<td>Future Direction</td>
<td>188</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.6</td>
<td>EAP-TLS Timing Profile</td>
<td>131</td>
</tr>
<tr>
<td>5.7</td>
<td>Abnormal and Mean EAP-LEAP Timings</td>
<td>133</td>
</tr>
<tr>
<td>5.8</td>
<td>Abnormal and Mean EAP-PEAP Timings</td>
<td>134</td>
</tr>
<tr>
<td>5.9</td>
<td>Abnormal and Mean EAP-TLS Timings</td>
<td>135</td>
</tr>
<tr>
<td>5.10</td>
<td>EAP-LEAP Timings During DoS Attack</td>
<td>136</td>
</tr>
<tr>
<td>5.11</td>
<td>EAP-PEAP Timings During DoS Attack</td>
<td>137</td>
</tr>
<tr>
<td>5.12</td>
<td>EAP-TLS Timings During DoS Attack</td>
<td>138</td>
</tr>
<tr>
<td>5.13</td>
<td>EAP-TLS Events During Normal Behavior</td>
<td>139</td>
</tr>
<tr>
<td>5.14</td>
<td>EAP-PEAP Events During Normal Behavior</td>
<td>140</td>
</tr>
<tr>
<td>5.15</td>
<td>EAP-LEAP Events During Normal Behavior</td>
<td>141</td>
</tr>
<tr>
<td>5.16</td>
<td>EAP-TLS Raw Events During Replay Attack</td>
<td>141</td>
</tr>
<tr>
<td>5.17</td>
<td>EAP-PEAP Events During Replay Attack</td>
<td>142</td>
</tr>
<tr>
<td>5.18</td>
<td>EAP-LEAP Events During Replay Attack</td>
<td>142</td>
</tr>
<tr>
<td>6.1</td>
<td>Ideal Normalized Support</td>
<td>146</td>
</tr>
<tr>
<td>6.2</td>
<td>Confidence Levels During Normal Operation</td>
<td>147</td>
</tr>
<tr>
<td>6.3</td>
<td>EAP-TLS Event Count During Normal Operations</td>
<td>148</td>
</tr>
<tr>
<td>6.4</td>
<td>Confidence Levels Under DoS Attack</td>
<td>153</td>
</tr>
<tr>
<td>6.5</td>
<td>EAP-TLS Event Count During DoS Attack</td>
<td>153</td>
</tr>
<tr>
<td>6.6</td>
<td>Confidence Levels Under Replay Attack</td>
<td>154</td>
</tr>
<tr>
<td>6.7</td>
<td>GOS for EAP-LEAP Events Under Normal Operation</td>
<td>159</td>
</tr>
<tr>
<td>6.8</td>
<td>GOS for EAP-PEAP Events Under Normal Operation</td>
<td>159</td>
</tr>
<tr>
<td>6.9</td>
<td>GOS for EAP-TLS Events Under Normal Operation</td>
<td>160</td>
</tr>
<tr>
<td>6.10</td>
<td>GOS Threshold Values</td>
<td>161</td>
</tr>
<tr>
<td>6.11</td>
<td>RSNA Events on STA3 for D=11</td>
<td>161</td>
</tr>
<tr>
<td>6.12</td>
<td>Abnormal RSNA Events on STA3</td>
<td>162</td>
</tr>
<tr>
<td>6.13</td>
<td>GOS from different Views</td>
<td>162</td>
</tr>
<tr>
<td>6.14</td>
<td>RSNA Events on STA4</td>
<td>163</td>
</tr>
<tr>
<td>6.15</td>
<td>Malicious Association</td>
<td>168</td>
</tr>
<tr>
<td>6.16</td>
<td>GOS Values During Malicious Authentication</td>
<td>170</td>
</tr>
</tbody>
</table>
6.17 GOS Values During a Session Hijack .......................... 173
6.18 GOS Values from a Different View .............................. 173

B.1 Protocol Mapping ................................................. 193
B.2 Event Mapping .................................................. 194
# List of Figures

1.1 OLAP Architecture ............................................. 9  
1.2 Combining OLAP and Data Mining ................................. 10  

2.1 Extended Service Set (ESS) ...................................... 16  
2.2 Independent Basic Service Set .................................... 17  
2.3 RSN Association Process ......................................... 28  
2.4 802.1x Port Based Access Control ............................... 29  
2.5 “Bro” System Architecture ........................................ 39  
2.6 Roll-Up and Drill-Down on a Three Dimensional Cube .......... 52  
2.7 Slicing and Dicing on a Three Dimensional Cube ............... 52  
2.8 Pivot Operation .................................................. 53  
2.9 A Data Cube Lattice With all Possible Attribute Combinations . 57  
2.10 A Dimensional Data Cube ........................................ 59  

3.1 RBT for Requirement 5 .............................................. 71  
3.2 Component Behavior Projection .................................... 72  
3.3 Translation of BT into SAL specification .......................... 74  
3.4 Translation of BT with multiple paths ............................. 76  
3.5 Malicious Association ............................................. 82  
3.6 Malicious Association during Authentication ..................... 84  
3.7 A Session Hijack .................................................. 87  
3.8 The EWS Block Diagram .......................................... 90
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Anomaly Filter</td>
<td>95</td>
</tr>
<tr>
<td>4.2 Data Cube Views</td>
<td>100</td>
</tr>
<tr>
<td>4.3 Experimental Setup</td>
<td>109</td>
</tr>
<tr>
<td>4.4 Test Environment</td>
<td>110</td>
</tr>
<tr>
<td>4.5 Abnormal Behavior Projection</td>
<td>118</td>
</tr>
<tr>
<td>5.1 EAP-LEAP Authentication</td>
<td>126</td>
</tr>
<tr>
<td>5.2 EAP-PEAP Authentication</td>
<td>127</td>
</tr>
<tr>
<td>5.3 EAP-TLS Authentication</td>
<td>129</td>
</tr>
<tr>
<td>5.4 EAP-LEAP Timings</td>
<td>133</td>
</tr>
<tr>
<td>5.5 EAP-PEAP Timings</td>
<td>135</td>
</tr>
<tr>
<td>5.6 EAP-TLS Timings</td>
<td>136</td>
</tr>
<tr>
<td>6.1 Normal EAP-LEAP Authentication</td>
<td>147</td>
</tr>
<tr>
<td>6.2 Normal EAP-PEAP Authentication</td>
<td>148</td>
</tr>
<tr>
<td>6.3 Normal EAP-TLS Authentication</td>
<td>149</td>
</tr>
<tr>
<td>6.4 DoS Attack during EAP-LEAP Authentication</td>
<td>150</td>
</tr>
<tr>
<td>6.5 DoS Attack during EAP-PEAP Authentication</td>
<td>151</td>
</tr>
<tr>
<td>6.6 DoS Attack during EAP-TLS Authentication</td>
<td>152</td>
</tr>
<tr>
<td>6.7 Replay Attack during EAP-LEAP Authentication</td>
<td>154</td>
</tr>
<tr>
<td>6.8 Replay Attack during EAP-PEAP Authentication</td>
<td>155</td>
</tr>
<tr>
<td>6.9 Replay Attack during EAP-TLS Authentication</td>
<td>156</td>
</tr>
<tr>
<td>6.10 Normal and Abnormal RSNA Events</td>
<td>164</td>
</tr>
<tr>
<td>6.11 DoS During EAP-TLS Authentication</td>
<td>166</td>
</tr>
<tr>
<td>6.12 DoS Attack During EAP-LEAP Authentication</td>
<td>167</td>
</tr>
<tr>
<td>6.13 DoS Attack During EAP-PEAP Authentication</td>
<td>168</td>
</tr>
<tr>
<td>6.14 GOS for Malicious Association</td>
<td>169</td>
</tr>
<tr>
<td>6.15 Outlier During Malicious Authentication</td>
<td>172</td>
</tr>
<tr>
<td>6.16 EWS Accuracy for Replay Attack</td>
<td>174</td>
</tr>
<tr>
<td>6.17 EWS Accuracy for DoS Attack</td>
<td>175</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

During the last decade, significant research work has been carried out on improving the intrusion detection mechanisms, mainly for wired networks. For wireless networks, the issues are more challenging. Ever increasing security threats and the dynamic nature of wireless network usage demand more reliable and fast intrusion detection or prevention techniques. Unfortunately, many organizations find it difficult to use the available techniques effectively, mainly because of the complexity of deployment, lack of information about its appropriate use and the amount of false positives. In this view the main aim of this study is to facilitate network security administrators, integrators and end-users to confidently utilize their wireless networks to meet the expectations of their organization.

1.1 Background

1.1.1 Security Requirements

Network security has different perceptions depending on different applications, among which the essential requirements are data confidentiality, integrity, authentication, and availability.

A network MUST provide strong data confidentiality, integrity, and replay
CHAPTER 1. INTRODUCTION

protection for every transmitted message. Data confidentiality and integrity characteristics, in a way guarantee a secure channel for the user to communicate in an insecure environment, thereby enabling only the legitimate users to understand the received messages and to generate or modify the valid messages. Furthermore, replayed messages should be detected and discarded though they may pass the integrity check. These requirements could be satisfied by well-designed cryptographic techniques and appropriate replay protection mechanisms.

Mutual authentication enables the communicating peers authenticate each other’s identity. Further, the authentication process should combine with key generation, distribution and management of secret keys for the cryptographic function. Based on the authentication results, flexible authorization and access control policies could be deployed to restrict the privileges of users.

Availability is one of the most important quality of security requirements in wireless networks. The network should be able to prevent an adversary from shutting down the connectivity or reducing the performance of a legitimate host or the entire network. In other words, Denial of Service (DoS) attacks should be eliminated, or at least mitigated.

1.1.2 Wireless Security Growth

During the evolution of WLAN security, many efforts have been made to achieve data confidentiality, integrity and mutual authentication. In this process several shortfalls were identified and rectified. However, until the latest wireless security standard IEEE 802.11i [59] was released most issues remained unresolved. Although the new standard addresses most issues found in the earlier standards, availability has not been fully addressed resulting in numerous DoS attacks in the WLAN system.
Data Confidentiality, Integrity and Authentication

In order to provide data confidentiality equivalent to a wired network, the original wireless network standard IEEE 802.11 [55] defines Wired Equivalent Protocol (WEP). This mechanism adopts a common stream cipher known as RC4 (Rivest Cipher 4 (Ron.s Code)) [100], to encrypt messages with a shared key. This key is concatenated with a 24-bit Initialization Vector (IV) to construct a per-packet RC4 key. In order to provide data integrity, WEP calculates an Integrity Check Value (ICV) over the MAC Service Data Unit (MSDU), which is a common Cyclic Redundancy Checksum (CRC). The frame body, together with the corresponding ICV, is encrypted using the per-packet key. In addition, two authentication mechanisms are defined: the Open System Authentication, which is actually a null authentication, and the Shared Key Authentication, which is a Challenge-Response handshake based on the shared key.

However, numerous studies have shown that data confidentiality, integrity, and authentication are not guaranteed or not fully achieved through these mechanisms. First, the 40-bit shared key is too short for brute-force attacks [22, 90]. Though some vendors might support a longer key (128 bits, containing a 104-bit key and a 24-bit IV), it is still easy for an adversary to recover the plain-text traffic because of the small IV size and the static shared key that results in the high possibility of key stream reuse [22, 121], which trivially defeats any stream cipher. Furthermore, the concatenation of the IV and the shared key has inherent weakness for generating the per-packet RC4 key [41]; an adversary can discover the key by eavesdropping several million packets [117]. Moreover, because ICV is a linear and unkeyed function of the message [22], data integrity cannot be guaranteed; even without any knowledge of the key stream, an adversary is able to arbitrarily modify a packet without detection, or forge a packet with a valid ICV. This weak integrity also enables much easier plain-text recovery, as in IP redirection, reaction attacks [22], and inductive chosen plain-text attacks [8]. Finally, an adversary can trivially spoof the Shared Key Authentication [9, 22], through
observing an authentication process of a legitimate station. Additionally, WEP does not implement any mechanism to prevent replay attacks. FBI agents have publicly demonstrated that they can break a 128-bit WEP key in about three minutes [27].

Although WEP fails to satisfy any security requirements, it is not practical to expect users to completely discard their devices with WEP already implemented. Hence, the WiFi Alliance proposed an interim solution, called WiFi Protected Access (WPA), to improve security while reusing the legacy hardware. WPA adopts a Temporal Key Integrity Protocol (TKIP) for data confidentiality, which still uses RC4 for data encryption, but includes a key mixing function and an extended IV space to construct unrelated and fresh per-packet keys. WPA also introduces the Michael algorithm [59], a weak keyed Message Integrity Code (MIC), for improved data integrity under the limitation of the computation power available in the devices. Furthermore, in order to detect replayed packets, WPA implements a packet sequencing mechanism by binding a monotonically increasing sequence number to each packet.

In addition, WPA provides two improved authentication mechanisms. In one mechanism, the possession of a Pre-Shared Key (PSK) authenticates the peers; furthermore, a 128-bit encryption key and another distinct 64-bit MIC key can be derived from the PSK. Alternatively, IEEE 802.1x [60] and the Extensible Authentication Protocol (EAP) [21] can be adopted to provide a stronger authentication for each association, and generate a fresh common secret as part of the authentication process; all required keys can be derived from this shared secret afterward.

TKIP is proposed to address all known vulnerabilities in WEP; it does enhance the security in all aspects. However, there are weaknesses in WPA due to the limitation of reusing the legacy hardware. Although the TKIP key mixing function has stronger security than the WEP key scheduling algorithm, it is not as strong as expected. It is possible to find the MIC key given one per-packet
key. Furthermore, the whole security is broken for the duration of a Temporal Key (TK), given two per-packet keys with the same IV32 [85]. This vulnerability does not mean that TKIP is insecure; however, it discloses that parts of TKIP are weak on their own. Furthermore, the Michael algorithm is designed to provide only 20 bits, or slightly more, of security in order to minimize the impact on the performance, which means an adversary can construct one successful forgery every 219 packets. Therefore, countermeasures are necessary to limit the rate of the forgery attempts [59]. In addition, the 802.1x authentication may be vulnerable to Session Hijacking and Man-in-the-Middle attacks [83]. Though these attacks disappear when mutual authentications and strong encryption are used [28], there are deficiencies of using 802.1x, which was originally designed for a switched LAN.

As a long-term solution, IEEE 802.11i [59] is proposed to provide an enhanced MAC layer security. 802.11i provides mutual authentication, key management, and data confidentiality protocols that may execute concurrently over a network in which other protocols are also used. On the assumptions of upgrading the hardware, 802.11i defines a Counter-mode/CBC-MAC Protocol (CCMP) that provides strong confidentiality, integrity, and replay protection. In addition, an authentication process, combining the 802.1x authentication and key management procedures, is performed to mutually authenticate the devices and generate a fresh session key for data transmissions. Since 802.11i promises to be the right solution for wireless security, it should be able to prevent an adversary from advanced attacks even if the adversary has the most powerful equipments and techniques for breaking into the system. In other words, an implementation of 802.11i protocols in a WLAN should be able to provide sufficient data confidentiality, integrity, and mutual authentication.

Availability

The above discussion was focused on the data confidentiality, integrity, and mutual authentication for wireless security. Many DoS attacks have been disclosed
on the WLAN systems from the Physical Layer to the Application Layer. Some
might think that the DoS attacks are inevitable due to the physical characteristics
of wireless links. However, since many DoS attacks can be mounted by an adver-
sary with moderate equipments, and a successful DoS attack may facilitate other
advanced attacks, such as Session Hijacking and Man-in-the-Middle (MitM) [77],
they should be considered to be real threats to a WLAN implementation. There-
fore, it is necessary to deploy a security mechanism that can defend against DoS
attacks. Since 802.11i does not emphasize such objective, it is definitely valuable
to implement some mechanism to mitigate DoS attacks. The key point to miti-
gate these attacks is to impose relatively higher cost for an adversary, e.g., more
computation power, more message transmissions, or more memory consumption,
which could make the DoS attacks less appealing.

1.2 Malicious Behavior of Wireless Networks

Unlike wired networks wireless networks exhibit high incidences of malicious be-
havior not only due to their inherent qualities but due to the various security
threats targeted towards them. As discussed above the wireless network environ-
ment needs to be protected to uphold the vital security requirements; data confi-
dentiality, integrity, authentication, and availability. Although 802.11i guarantees
extensive security for the wireless environment it is still premature to exclude po-
tential future threats and/or anomalies. Further, there are many wireless instal-
lations, where appropriate security mechanisms are neither used nor implemented
effectively [19]. Hence in addition to improving the existing security mechanisms
we also need other protection techniques to guard the wireless environment from
new security threats and/or anomalies.

It is evident that the behavior of an access point varies from one vendor to
another and depends on many factors. Hence, maintaining a behavioral profile
for every participating wireless station (STA) - AP pair will be more appropriate
1.3. SUBSTANTIATING ANOMALIES

rather than having a generic profile for the environment. Throughput and resis-
tance to one single node (STA) flooding depends on the actual implementation and
operative system of the access point. A variation in timing during the association
process for a given station depends on factors such as type of transversing traffic
(number of packets and their sizes), UDP vs TCP traffic, number of traffic flows
etc. Therefore, maintaining dynamic profiles will be more effective than having
static profiles. Hence, employing an adaptive technique to update the timing and
behavioral profiles depending on the nature of the operative environment would
be effective.

Trying to model what is “normal” in an AP is not an easy task, hidden nodes
will trigger retransmissions and heavy real time UDP traffic will have an impact in
how packet queues are handled and hence timing of management frames. There-
fore, detecting a security threat in the wireless environment, merely based on
timing profiles can lead to large number of false positives. Hence, the use of out-
lier based data association techniques to substantiate the security threats would
be useful in reducing the number of false positives.

1.3 Substantiating Anomalies

Due to continuing advances in communication technology, the use of computer
networks has progressed exponentially. More and more hosts are connecting to
computer networks wired or wireless resulting in larger amounts of data being
collected for network security related analysis. To get the most out of this multi-
level longitudinal data sets, effective data analysis methods are required to extract
non-trivial, valid, and useful information. Considerable research work has been
carried out towards improving knowledge discovery in databases (KDD) in order
to meet these demands.

In several applications, such as network intrusion detection, sensor networks,
stock market analysis, health monitoring systems, etc., the problem of detecting rare events, malicious behaviors, and exceptions is very important. Methods for finding such outliers in large data sets are drawing increasing attention of researchers. The leading approaches to outlier detection can be classified as distance based \cite{69, 70}, depth-based \cite{66}, clustering \cite{64}, density-based \cite{23}, or discovery-driven \cite{98}.

In this study we discuss the possible use of associated data views for substantiating the legitimacy of security threats in wireless computer networks. Our proposed method, which tries to differentiate between legitimate and illegitimate events in the network environment is developed using outlier based data association techniques. The multivariate, latent variable and multi-level data collected from large wireless networks need to be processed in real-time for effective intrusion prevention and/or detection. Most statistical methods that are used to process multivariate data sets are ineffective for real-time applications. Hence, we look at the possibilities of using outlier based data association techniques in our research. For this purpose we have decided to use data cubes as our primary data storage structure on which we use the OnLine Analytical Processing for outlier detection.

1.4 OLAP and the Data Cube

Online analytical processing (OLAP) is a key feature supported by most data warehousing systems \cite{118, 31, 101}. OLAP is quite different from online transaction processing (OLTP) systems. OLTP focuses on automation of data collecting procedure. Keeping detailed, consistent, and up-to-date data is the most critical requirement for an OLTP application. Although as fundamental building blocks these transactional records are important to an organization, a decision maker is more interested in the summary data than investigating a particular record. Traditional relational database management system (DBMS) is not efficient enough to
satisfy the requirement of OLAP since to acquire summary information a number of aggregation SQL queries with group-by clauses are necessary.

The OLAP concept was introduced to satisfy the requirement of efficiency. Summary or aggregation data, such as sum, average, max, and min, is pre-calculated and stored in a data structure called the data cube. Compared with two-way relational tables normally used in OLTP, a data cube is multidimensional. Each dimension consists of one or more categorical attributes, and hierarchical structures generally exist in the dimensions. The architecture of a typical OLAP application is shown in Figure 1.1. The first (bottom) tier of this architecture is a warehouse database server which holds the transaction data that are extracted from operational databases and/or external sources using one or more application programs. The second (middle) tier is an OLAP server, that is, a special-purpose server that directly implements the OLAP data cube (multidimensional data) and operations. The third (top) tier is a client, which contains the presentation tools such as the query and reporting tools, analysis tools, and/or data mining tools.

Although OLAP is capable of providing summary information efficiently, how to make the final decision is still an art of applying the domain knowledge, sometimes common sense, of the decision-maker. A few quantitative data mining methods, like regression or classification, have been introduced into the OLAP area. On the other hand, traditional data mining algorithms are mostly designed
for two-dimensional datasets, and OLAP is not involved in developing the data mining algorithm. Since both OLAP and data mining are powerful tools for the decision making process, the ideal situation is to combine both of them to solve the real-world problem, as illustrated in Figure 1.2.

In this study, we investigate the possible ways of combining both the data mining and OLAP to solve the data association problem. Data association involves linking or grouping records in the database according to similarity or other mechanisms. Many applications can be treated as data association problems. For example, in multiple-sensor or multiple-target tracking [13, 53, 93], different tracks of the same target are associated; in document retrieval system [97], documents with a given search string are associated; in crime analysis [24], crime incidents by the same criminal are associated. Different approaches have been proposed to solve the data association problem. In our study, we propose and analyze a new data association method - an OLAP group outlier based data association method. This method integrates both the concept of outlier detection from data mining field and OLAP techniques seamlessly.
1.5 Research Contribution

Although 802.11i promises satisfactory security for the wireless environment it is still premature to exclude potential future threats. Further, there are many wireless installations where appropriate security mechanisms are not used or implemented properly. Hence in addition to improving the existing standards we also need effective mechanisms to protect the wireless environment from malicious activities. In this regard the primary aim of this study is to detect and effectively substantiate malicious behaviors of the wireless networks. Thus, enabling network security administrators, integrators and end-users to confidently utilize their wireless networks to meet the expectations of their organization.

Effective use of the wireless network will only be possible if security threats are detected early, preventing any potential catastrophe. In this respect, tracking wireless traces and properly analyzing them may reveal vital information about impending threats to the wireless environment. One such analysis considered in this study is based on the round trip timings associated with the management frames. Unusual timing values exhibited by wireless stations may be the beginning of an intended security breach. Furthermore, legitimate wireless hosts attempting to connect to an authorized Access Point (AP) usually demonstrate a set of defined behavior. Tracking and profiling such behavior of all stations in the wireless environment would normally present some useful information for detecting misbehaving stations. Therefore, tracking the management frames and appropriately analyzing them for timing and/or behavior anomalies is expected to enhance the ability of the detection mechanism to discover security threats in advance.

In addition, accumulating historical data and analyzing them to verify the legitimacy of detected anomalies could improve the effectiveness of the detection mechanism. Hence, in this study we have investigated the use of associated data views for substantiating the legitimacy of malicious behavior in wireless networks.
Our method differentiates between legitimate and illegitimate behavior using outlier based data association techniques. The main features of our approach are:

- We build a repository of network traces on a Beowulf cluster as a partial data cube compared to methods which consider building and manipulating the entire data cube.

- We update aggregates in real time as opposed to methods which calculate summary values during fixed time intervals.

- We substantiate the legitimacy of abnormal conditions in a network environment using an OLAP group outlier based data association methods enabling more accurate classification of anomalies.

- The data is first modeled into an OLAP data cube, and a group outlier score function is built over the OLAP cells.

Our method demonstrates online real time manipulation of multidimensional data sets represented as a partial data cube suitable for applications that requires real time response.

Using partial data cubes for fast querying and populating the aggregate values in real time makes our system viable and reliable. Our outlier detection algorithm makes at most three drill-down queries on the partial data cube to substantiate the legitimacy of an anomaly. Hence, our method demonstrates online real time manipulation of multidimensional data sets represented as a partial data cube suitable for applications that requires real time response. To the best of our knowledge, this is the first work to use group outliers for substantiating abnormal conditions in computer networks.

1.6 Dissertation Outline

The wireless environment is subject to a wide range of anomalies due to both the inherent qualities of wireless communications and various security issues. Hence
in addition to improving the existing standards we need effective mechanisms to
protect the wireless environment from malicious activities. In this regard this
dissertation outlines our approach for detecting and effectively substantiating all
malicious behaviors in wireless networks. Some of the results discussed in Chap-
ters 4, 5 and 6 can also be found in our published articles [105, 107, 111, 109].

Chapter 2 gives a detailed overview of related work on intrusion detection
techniques, anomaly detection techniques, use of data mining in intrusion detec-
tion, use of OLAP and the data cube for data mining applications, outlier based
data association techniques and the use of data cubes for outlier detection. In
Chapter 3 we present the research methodology together with hypothesis and dis-
cuss the analysis of the IEEE 802.11i protocol. The proposed solution and the
related theories are explained and derived in Chapter 4. Details of anomaly detec-
tion techniques and the experimental results are presented in Chapter 5. Results
from the substantiation mechanisms are analyzed and discussed in Chapter 6.
Conclusions and future directions are discussed in Chapter 7.
Chapter 2

Literature Review

2.1 Introduction

Computer security has emerged as one of the most researched areas in Information Technology. Especially, the introduction of wireless devices within computer networks has increased the significance of computer network security. In this context, wireless network security has become one of the important aspect in securing computer networks. Hence, effective intrusion prevention and/or detection within wireless networks could drastically reduce the threat to computer network as a whole. In view of developing an effective intrusion prevention and/or detection mechanism firstly, we review the various wireless technologies, standards and their use. As for security, we review the different types of security threats to the wireless networking environment and examine the various countermeasures proposed in the literature.

Since the main aim of this research is to propose a suitable substantiation mechanism to validate the wireless security threats, we surveyed the different ways of analyzing multivariate, latent variable and longitudinal data sets. In this context, we opted to use outlier based data analysis as it was more suited for security related applications. Since, an outlier is an observation that is numerically distant from the rest of the data, statistics derived from data sets that include
outliers may be misleading. Therefore, using statistical means to detect outliers may be ineffective for our application. Outliers may be indicative of data points that belong to a different population than the rest of the sample set. Therefore, outlier based data association may be more appropriate for security related applications than statistical methods. In this view, we reviewed the various outlier based data association techniques used for this purpose.

2.1.1 Chapter Overview

This chapter is organized as follows: In Section 2.2 we provide a brief introduction of wireless LANs and in Section 2.3 we discuss the various wireless security standards. Security threats associated with the wireless environment together with their countermeasures are discussed in Section 2.4. In Section 2.5 we survey both anomaly and data mining based intrusion detection system employed to detect security attacks in the wireless environment. The advantages and disadvantages of these different intrusion detection systems are discussed in Section 2.6. Thereafter, in Section 2.7, we review the various methods used to detect outliers in longitudinal data sets.
2.2 Wireless LANs

A WLAN is a flexible and dynamic data communications system that transmits and receives information over the unguided medium. The Extended Service Set (ESS) is typically part of a larger network with interfaces to a wired LAN as shown in Figure 2.1. An access point bridges or interfaces the mobile hosts to the wired LAN. The wireless stations have NICs that interface the stations to the APs by radio frequency (RF) transmissions.

Another WLAN configuration consists of a stand-alone RF network that is made up of only STAs. It operates as an independent WLAN known as an ad-hoc or Independent Basic Service Set (IBSS) as shown in Figure 2.2 [55].

Before 1997, there were no standards for wireless LAN products. Each vendor defined and implemented their own proprietary network protocols and signaling waveforms for their line of wireless products. Different vendor products were not interoperable and required the wireless network to operate with only one vendor’s products. The IEEE formed a working group to standardize the protocols and signaling for wireless LANs. This working group had the first IEEE 802.11 standard accepted in 1997 which operated at 1 Mbps with an optional 2 Mbps
The IEEE 802.11g wireless standard is the latest to be ratified by the IEEE. It offers a number of improvements over the 802.11, 802.11a [56] and 802.11b [57] standards, including 54 Mbps signaling rate, legacy 802.11b backward compatibility, and increased security features. The 802.11g wireless clients can select from the widest possible range of both OFDM data rates of 54, 48, 36, 24, 18, 12, 9 Mbps and the CCK data rates of 11.5, 2 and 1 Mbps [58].

A WLAN uses radio waves to communicate among devices. An AP with an antenna is physically connected to a conventional wired Ethernet network and serves as a bridge to the wireless network. WLANs also support communications among the STAs, allowing the devices to communicate directly with one another in a peer-to-peer configuration.

Up to approximately 150 feet, a 802.11b WLAN typically can deliver broadband performance with a signaling speed of up to 11 Mbps. Beyond that distance, it can operate at fall back speeds of 5.5 Mbps, 2 Mbps and 1 Mbps. At these lower speeds the signal can travel as far as 1,500 feet. Directional antennas can be used to extend the range significantly. Actual performance depends upon the signal pattern and the number of walls, floors and other architectural obstacles in the area. 802.11g WLANs can achieve speeds of up to 54 Mbps within approximately 75 feet and the 802.11a WLANs can achieve this speeds within a somewhat reduced range.

In order to indicate its presence to STAs in its listening area, an AP announces itself by beaconing, or broadcasting, a SSID approximately 10 times per second. The SSID identifies the name of the network. Mobile Hosts that are within range can receive the SSID, associate with the WLAN and request an IP address that will allow them to connect to the WLAN, surf the Internet, and view network folders [124].
2.3 Wireless Security Standards

Wireless networking continues to gain momentum, especially in casual environments such as homes, coffee shops, and airports. However, enterprises and government entities have largely shunned wireless networking because of legitimate concerns over security vulnerabilities inherent in the IEEE 802.11 WEP standard [55]. The failure of WEP to provide reliable wireless network security has deterred the mass adoption of wireless networking in the enterprise environment - where data security is vital - despite the promise of increased productivity through greater mobility.

Although the IEEE 802.11 wireless standard includes provisions for security, these provisions are now known to be insufficient. As a result, good wireless security has relied on technology that does not adhere to the 802.11 standard. Instead, these stopgap measures either use proprietary extensions to 802.11 or rely on a higher level security protocol such as IPsec to create, for example, a VPN. The complexities of these provisions make wireless networking even less attractive to enterprises that have considered implementing it.

The IEEE 802.11i task group was formed to improve wireless security [59], but the standard is yet to be fully tested. To tap markets for wireless products in conservative business and government settings, wireless vendors - through the WiFi Alliance have introduced WiFi Protected Access (WPA) [123]. This intermediate, forward-compatible security specification incorporates most of the features in the final 802.11i standard. In the following sections we examine the flawed 802.11 security standard known as WEP, exploring how the 802.11i standard seeks to remedy the flaws and what features of 802.11i are incorporated in the WPA.

2.3.1 WEP

Wired Equivalent Protocol (WEP) was an encryption algorithm designed to provide wireless security for users implementing 802.11 wireless networks. WEP was
CHAPTER 2. LITERATURE REVIEW

devolved by a group of volunteer IEEE members. The intention was to offer se-
curity through an 802.11 wireless network while the wireless data was transmitted
from one end point to another over radio waves. WEP was used to protect wireless
communication from eavesdropping (confidentiality), prevent unauthorized access
to a wireless network (access control) and prevent tampering with transmitted
messages (data integrity).

WEP uses the RC4 stream cipher [100], combining a 40-bit WEP key with a
24-bit random number known as an IV to encrypt the data. The sender XORs the
stream cipher with the actual data to produce cipher text. The packet, combined
with the IV with the cipher text, is sent to the receiver. The receiver decrypts
the packet using the stored WEP key and the attached IV [55].

Unfortunately, the encryption protocol had not been subjected to a significant
amount of peer review before release. Serious security flaws were present in the
protocol. Although the application of WEP may stop casual sniffers, experienced
hackers can crack the WEP keys in a busy network within a short time. In general,
WEP was considered as a broken protocol [41]. However, large number of orga-
nizations and individuals still rely on security provided by the WEP mechanism
[19].

The vulnerability of WEP can be attributed to the following:

- WEP key recovery: WEP uses the same WEP key and a different IV to
  encrypt data. The IV has only a limited range (0 to 16777215) to choose
  from. Eventually, the same IVs may be used over and over again. By picking
  the repeating IVs out of the data stream, an attacker can ultimately have
  enough collection of data to crack the WEP key [22].

- Unauthorized decryption and the violation of data integrity: Once the WEP
  key is revealed, a hacker may transform the cipher text into its original form
  and understand the meaning of the data. Based on the understanding of
  the algorithm, a hacker may use the cracked WEP key to modify the cipher
text and forward the changed message to the receiver.

- **Poor key management:** A proper WEP key is typed into a wireless device associated in a wireless network to enable the WEP. Unfortunately, there are no mechanisms to renew the stored WEP key. Once the WEP key is compromised, for example, an employee leaves a company; the key has to be changed in order to retain the security. The change of keys may be applicable in a home or small business environment. However, in an enterprise environment with thousands of wireless mobile devices associated with the wireless network, the use of this method is almost impossible.

- **No mutual authentication:** WEP only provides a method for NICs to authenticate access points. There is no way for access points to authenticate the NICs. As a result, it is possible for a hacker to reroute the data to access points through an alternate unauthorized path.

### 2.3.2 IEEE 802.1x Authentication

The recent wireless security standard the IEEE 802.11i specification incorporates IEEE 802.1x, to improve access control on wireless networks through a more rigorous authentication mechanism. Five components are required to implement 802.1x authentication [60]:

- **Compatible client device:** Typical clients include notebook computers and Personal Digital Assistants (PDAs). A device that requests to join a wireless network is known as a supplicant.

- **Supplicant software:** This software provides the logic that a device needs to present its credentials and follow the proper protocol for joining the network as a client.

- **Authenticator:** The authenticator is a wireless access point that must verify the identity of a supplicant before granting the device network access.
CHAPTER 2. LITERATURE REVIEW

- **Authentication server**: The authentication server is a separate system typically running the Remote Authentication Dial In User Service (RADIUS) [94] or another service supporting the EAP [21] that handles authentication requests relayed by authenticators from supplicants. EAP authentication modes such as EAP-MD5, EAP-LEAP/PEAP, EAP-TLS and EAP-TTLS can be supported [95, 89, 1, 43].

- **User database**: The user database is a list of valid users and their credentials that the authentication server consults to validate authentication requests. This database may be a simple flat file or a service provided by a directory infrastructure, such as the Microsoft Active Directory service or the LDAP [127].

The authentication process begins when a client attempts to connect to the access point, which will open a restricted port. This port allows the client to pass only EAP packets to the authentication server on the wired side of the access point. All other traffic, such as HTTP or DHCP traffic, is blocked. The 802.1x authentication protocol involves seven basic steps:

- **Request access**: The supplicant presents the authenticator with an EAP response/identity request.

- **Limit access to authentication server**: The authenticator relays the request to the authentication server; at this point, the supplicant’s access is restricted to the authentication server.

- **Issue challenge**: The server issues a challenge and passes it back to the supplicant.

- **Answer challenge**: The supplicant answers the challenge by sending the necessary credentials back to the authentication server.

- **Validate response**: The server verifies the credentials against the user database; if valid, the server responds with a success message.
• Allow access to network: The authenticator increases the scope of the client’s access.

• Use other network devices: The authenticator notifies the client that it may now participate on the network.

Prior to the release of 802.11i, the 802.1x authentication was introduced by the WiFi Alliance in their WPA security solution for mutual authentication and key management.

2.3.3 WPA

In October 2002, the WiFi Alliance announced a security solution that counters the known weaknesses of WEP called WPA. It is a subset of the abilities of IEEE 802.11i, including better encryption with TKIP, easier setup using a pre-shared key, and the ability to use RADIUS based 802.1x authentication of users. WPA comes in two forms, one that’s easier for home users, and one for enterprises (the latter incorporates 802.1x). WPA is designed to work with the existing 802.11 based products and offers forward compatibility with 802.11i. All of the known shortcomings of WEP are addressed by WPA, which features packet key mixing, a message integrity check, an extended initialization vector, and a re-keying mechanism.

WPA includes the following key features to address WEP vulnerabilities [123]:

• Network security capability determination: This occurs at the 802.11 level and is communicated through WPA information elements in Beacon, Probe Response, and (Re) Association Requests. Information in these elements includes the authentication method (802.1x or Pre-shared key) and the preferred cipher suite (WEP, TKIP, or AES).

• 802.1x EAP based authentication: WPA adopts 802.1x to address the issue of user authentication in WEP. 802.1x was initially designed for wired
networks but is also applicable to wireless networks. The standard provides port-based access control and mutual authentication between clients and access points via an authentication server. There is a special case in 802.1x implementation for small installations. In a home or small business environment, an authentication server may not be available for authentication. As such, a pre-shared key mechanism is used. The shared key is placed to a supplicant and an authenticator manually. A similar WEP-like authentication is operated.

- Key management: WPA features a robust key generation/management system that ties together the authentication and data privacy functions. Keys are generated during a successful authentication and from these keys other keys are derived through a subsequent 4-way handshake between the station and AP. This handshake produces keys that are used to seed the data privacy algorithm.

- Data Privacy: TKIP is another element derived from 802.11i. It is aimed to address WEP’s known vulnerabilities in the area of data encryption. Specifically, TKIP tries to fix the security flaw of key reuse in WEP. TKIP packet is comprised of three parts:

  1. A 128-bit temporal key that is shared by both clients and access points.
  2. A MAC address of a client device.
  3. A 48-bit initialization vector describes a packet sequence number.

This combination enables various wireless clients to use different keys. In order to be compatible with existing hardware, TKIP uses the same encryption algorithm (RC4) as WEP. As such, only software or firmware upgrade is required to implement TKIP. Compared with WEP, TKIP changes the temporal keys every 10000 packets. This dynamic distribution leaves potential hackers little room to crack TKIP key.
• Data Integrity: Michael Message Integrity Check is used to enforce data integrity. A Message Integrity Code is a 64-bit message calculated using “Michael” algorithm [67]. It’s aim is to detect potential packet content alteration due to transmission error or deliberate manipulation. TKIP includes a MIC at the end of each plain-text message.

In general, the security advantages of WPA over WEP are [123]:

• Apply stronger network access control through mutual authentication.

• Support better security technologies like 802.1x, EAP, RADIUS and pre-shared keys.

• Adopt dynamic keys in TKIP to establish better key management.

• Enforce data integrity through Michael Message Integrity Check.

• Provide forward compatibility to ultimate wireless security solution, 802.11i.

However, WPA also presents some potential security issues:

• There are still potential encryption weaknesses in TKIP. Fortunately, the successful crack is expected to be heavy and expensive [85].

• Performance may be sacrificed potentially due to a more complex and computation intensive authentication and encryption protocols [44].

• The strength of WPA may still remain uncertain until further attack-proof methods are discovered.

### 2.3.4 IEEE 802.11i and WPA2

The long-waited IEEE security standard - the 802.11i for wireless LANs was ratified on in June 2004. The WiFi Alliance uses the abbreviation of “WPA2” when referring to 802.11i. WPA2 will not replace WPA, however, WPA will continue
to be available for homes and small businesses that don’t need the advanced encryption or RADIUS authentication. 802.11i/WPA2 products will be backwards compatible with WPA products, assuming they have the means to support AES.

The IEEE 802.11i specification includes several key features [34]:

- **Encryption algorithms**
  
  - **TKIP** - In order to support legacy devices, the 802.11i chooses TKIP as one of the encryption standard (same as WPA).
  
  - **CCMP** - A much stronger encryption standard. CCMP uses the CCM operation mode [122] of the AES algorithm [2] with a 128/196/256 bit key and a 128-bit block size. CCMP combines the Counter Mode for data confidentiality and the Cipher Block Chaining Message Authentication Code for data integrity, using an 8-octet MIC (Message Integrity Code) and a 2-octet Length field. However, AES-CCMP requires a hardware coprocessor to operate. Therefore, extra hardware is needed in the implementation of AES-CCMP.
  
  - **WRAP** - The last encryption that 802.11i includes is WRAP. Similar to CCMP, the encryption algorithm uses AES but Offset Codebook Mode for encryption and integrity.

- **Message Integrity.** A strong data integrity algorithm (Michael MIC) is applied (same as WPA).

- **Mutual Authentication.** 802.11i uses 802.1x/EAP for user authentication (same as WPA).

- **Other security features** - secure IBSS, secure fast hand-off, and secure Deauthentication and disassociation.

- **Roaming Support.**
The Wireless network that operates based on IEEE 802.11i standard is referred to as RSN. The standard defines two classes of security associations for wireless stations in RSN: RSNA and preRSNA. A station is called RSN-capable equipment if it is capable of creating RSNA, otherwise, it is pre-RSN equipment. The network that only allows RSNA with RSN-capable equipments is called a RSN security framework. The major difference between RSNA and preRSNA is the 4-way handshake. If the 4-way handshake is not included in the authentication/association procedures, stations are said to use preRSNA.

Figure 2.3 shows the various states of the entities during the RSNA. During the first two message (1-2) exchange the network and security capabilities are discovered by both supplicant and the authenticator. Messages 3 to 6 are included for backward compatibility so that 802.11 open system authentication and association could be done. In RSN the wireless hosts need to be 802.11 associated to commence 802.1x authentication. The 802.1x authentication commences with EAPOL Start message. However, this is an optional message and in some cases the 802.1x authentication starts with the authenticator issuing an EAP-Request Identity message to the supplicant. During the 802.1x authentication process the authenticator blocks it’s control port restricting its normal services. Only the EAP messages are passed to the authentication server via the uncontrolled port of the authenticator (Figure 2.4). The 802.1x authentication completes with the EAP-Success message from the authenticator to the supplicant. At this point the Pairwise Master Keys are established in both the supplicant and the authenticator. The next four messages installs the transient keys in both supplicant and the authenticator. With the pairwise transient keys established in both supplicant and the authenticator control port is unblocked for normal traffic. At this point the channel is fully secured with AES-CCMP.

802.11i has all the advantages provided by WPA and in addition, the 802.11i offers:

- Stronger Encryption through the implementation of AES.
Figure 2.3: RSN Association Process
2.4 Wireless Attacks and Countermeasures

Wireless threats come in all shapes and sizes; from someone connecting to your wireless access point without authorization, to grabbing packets out of the air and decoding them via a packet sniffer. Many wireless users have no idea what kinds of danger they face merely by connecting an AP to their wired network. This section discusses the most common threats faced by adding a wireless component to your network.

2.4.1 Wardriving

The wireless network environment is exposed to a range of intrusion attacks. The most common is “Wardriving” using a laptop and a wardriving software such as NetStumbler [86]. The attack generally exploits the famous weakness in WEP encryption used by IEEE 802.11b networks [22, 41]. This is usually a second stage following the detection of a secured AP by Wardriving. The most commonly used tool for WEP key extraction is the Linux program AirSnort [7]. AirSnort is a
passive monitor and does not throw any messages out.

2.4.2 Malicious or Accidental Association

A hacker can force an unsuspecting user station to connect to an undesired/spoofed 802.11 network, or alter the configuration of the station to operate in an ad-hoc networking mode. To begin, the hacker sets up a laptop as a soft access point using either freeware hacker’s tools, such as AirSnarf [6], or a commercially available tools (Companies such as PCTel provide commercial software that converts 802.11 devices into access points).

As the victim’s user station broadcasts a request to associate with an access point, the hacker’s soft access point responds to this request and establishes a connection between the two. Next, the soft access point provides an IP address to the victim’s user station. Once this is done, the hacker can scan the victim’s station with tools designed to find Windows’ vulnerabilities. The hacker can then steal information, install Trojan horses or other spyware, and if it is connected to the wired network, use the victim’s station as a launch pad to get access to other servers.

WLANs are also subject to diversion. Stations do not always know to which access point or network they are connecting. Stations can be tricked or forced to connect to a malicious access point, since access points are often not authenticated. This is an OSI Layer 2 (data link) vulnerability. Layer 3 (network) authentication offers no protection against it, nor does the use of VPNs. WLANs with 802.1x-based authentications (at Layer 2) do help protect against malicious associations, but are still vulnerable [9].

A malicious associations attack does not try to break the VPN or other security measures. Instead, it takes over the client at Layer 2. To prevent user stations from connecting to unauthorized access points and networks, enterprises must constantly monitor the airwaves of their WLANs to be aware of any potential hazards.
2.4. WIRELESS ATTACKS AND COUNTERMEASURES

2.4.3 Identity Theft (MAC Spoofing)

The theft of an authorized user’s identity is a serious threat to wireless networks. Even though SSIDs and MAC addresses act as Personal Identification Numbers for verifying the identity of authorized clients, existing encryption standards are not foolproof. Knowledgeable hackers can pick off authorized SSIDs and MAC addresses and steal bandwidth, corrupt or download files, and wreak havoc on the entire network [9].

Some enterprises secure their WLAN by using an authorized list of station MAC addresses for authentication. While this method provides some level of security for smaller deployments, MAC addresses were never intended for this purpose [9].

Even if you are using encryption or VPN, MAC addresses are always in the air. With software tools such as Kismet [68] or Ethereal [37], a hacker can easily capture the MAC address of a valid user. To perform identity theft, a hacker can change his MAC address to the victim’s MAC address using a spoofing utility such as SMAC (Spoof MAC), or, manually change the Windows registry entry. Once this has been done, the hacker can connect to the WLAN, bypassing any MAC address filtering [9].

There is a misconception that identity theft is only feasible if the MAC address is used for authentication, and that 802.1x-based authentication schemes such as LEAP are totally safe. Cracking LEAP to steal identity has become easy with tools like ASLEAP [10] and THC-LeapCracker [119]. Other authentication schemes, such as EAP-TLS and PEAP, may require more sophisticated attacks that exploit other known vulnerabilities in wired network authentication schemes, but are feasible.
2.4.4 Man-in-the-Middle Attacks

One of the most sophisticated attacks, the MitM attack, breaks even VPN connections between authorized stations and access points by inserting a malicious station between the victim’s station and the access point. The hacker becomes the “Man-in-the-Middle” [83].

These attacks are very similar to wired network MitM attacks, and tools to exploit these attacks on the wired networks can be easily used on the wireless network. Getting into the middle of a communication session is a problem on the wired side. This process is much easier with wireless networks. Using HostAP [54] software, a hacker can easily convert a wireless device into a soft access point, and position that access point in the middle of the communication session.

The more sophisticated MitM attack preys upon challenge and handshake protocols to perform a de-authentication attack. The de-authentication attack knocks a user from an access point, causing the user to search for a new access point with which to connect. With the hacker’s HostAP access point running, the user reconnects to the hacker’s laptop, PDA, or other device. Now the hacker, with a different wireless interface, connects to the real wireless LAN, passing all authentication traffic to the real wireless network. The victim is oblivious to this, and passes all data through the hacker. This scenario is possible because VPNs establish their connection at Layer 3 in the OSI model, while wireless NIC operates below the VPN, at Layer 1 and Layer 2. Additionally, freeware tools, such as Wireless AirJack [5], enable hackers to launch a MitM attack by automating the multiple steps required to perform it.

Only a highly capable IDS and 24-hour monitoring can detect these types of attacks on a WLAN [84]. An effective security solution keeps a constant watch on the network analyzing the network activity adaptively. Since this type of attack is not based on a single signature, a wireless IDS must be able to correlate and analyze data to show that this type of attack is occurring.

Lynn et. al. [77] discuss that if no authentication mechanisms are implemented
an adversary could establish two separate connections to the supplicant and the authenticator to construct a MitM attack.

MitM attack in tunneled authentication protocols is demonstrated by Asokan et. al. [11]. They have shown that when a client authentication protocol is tunneled within another protocol, it is necessary for each end point to demonstrate that it has participated in both protocols within the authentication exchange. If this is not demonstrated then the tunneled authentication protocol is vulnerable to MitM attack. Further, they have also shown that the required demonstration can be provided in an implicit or explicit way in a form of cryptographic binding between the tunnel protocol and the authentication protocol.

### Availability and DoS Attacks

Every network and security manager fears the downtime and loss of productivity that results from a crippling DoS attack. For a wireless network, the attack can come from any direction.

In the Physical Layer, a straightforward DoS attack is frequency jamming: an adversary can interfere the whole frequency band with a strong noise signal, blocking the legitimate data transmissions. There exists an easy approach to mounting a frequency jamming in a WLAN implementing Direct Sequence Spread Spectrum (DSSS) [12]. By exploiting the Clear Channel Assessment (CCA) procedure, an adversary can cause all WLAN nodes within range to consider the channel busy and defer transmissions of any data. Fortunately, the attack only affects a WLAN system implementing CCA, which is in DSSS (e.g., 802.11b/g), but not in OFDM (e.g., 802.11a/g).

As another possible attack, an adversary is able to transmit legitimate messages, without obeying the standard. Specifically, the adversary could use a smaller “backoff” time, in order to obtain an unfair allocation of the channel bandwidth. If the adversary adopts no “backoff”, he/she may ultimately cause a
DoS attack for legitimate users [71]. More DoS vulnerabilities arise from the unprotected management frames and control frames. An adversary is able to easily launch a DoS attack on a specific station or the entire BSS by forging the Deauthentication, Disassociation, Traffic Indication Map (TIM), or Poll messages [16]. Furthermore, DoS attacks could be mounted by exploiting the virtual carrier-sense scheme through forging any frame, especially RTS frame, with an extremely large value of NAV (Network Allocation Vector), which can fool the devices to consider the channel busy; thus, suppress the device from transmitting messages [16, 25]. Additionally, as in a wired LAN, an adversary can perform an ARP cache poisoning to mount a DoS attack if the adversary is able to access the network in someway [39].

Glass et. al. [47] claim that IEEE 802.11 networks are particularly vulnerable to DoS attacks and should not be used where availability is essential. Using equipment from a variety of manufacturers they have carried out experiments aimed at verifying the effectiveness of DoS attacks targeting the DCF. The experimental results confirmed that most 802.11 networks are vulnerable except for two cases - the PHY layer attack on a 802.11g network and the MAC layer RTS attack failing for certain networks cards where there is significant delays between frames.

Furthermore, if IEEE 802.1x authentication is implemented for stronger authentication, the adversary has more choices to mount a DoS attack through forging EAP-Start, EAP-Logoff, and EAP-Failure messages. The adversary can also exhaust the space of the EAP packet identifier, which is only 8 bits long, by sending more than 255 authentication requests simultaneously [9].

### 2.4.5 Network Injection Attacks

A newly-developed DoS, the network injection attack, exploits improperly configured wireless LANs or rogue access points to target the entire network. When an access point is attached to an unfiltered part of the enterprise network, it broadcasts network traffic, such as “Spanning Tree” (802.1D), OSPF, RIP, HSRP and
other broadcast or multicast traffic. By doing this, the packets invite attacks that take down wireless and wired network equipment and spur a meltdown of the entire internal network infrastructure, including hubs, routers, and switches [16].

The Spanning Tree algorithm normally ensures a loop-free Ethernet topology for networks that contain parallel bridges and multiple Ethernet segments. Loops occur when there are alternate routes between hosts. If a loop exists in an extended network, bridges may forward traffic to false or wrong Ethernet hosts indefinitely, increasing traffic and declining network performance to the point where the network stops responding. A hacker can inject traffic onto the wireless LAN segment and it will be propagated through the entire enterprise. This creates a DoS attack by intentionally inserting loops into the network.

Rogue sniffers initiate the DoS attack by echoing manipulated Spanning Tree sessions back to the wireless LAN access point. The access point echoes the packets to other internal hosts, causing a domino effect. Spanning Tree attacks usually render intelligent hubs, bridges, routers, and switches inoperative, requiring the devices to be rebooted or reconfigured to make them functional.

Routing attacks are another popular prey for enterprise DoS attacks. A hacker can use tools such as IRPAS or Routing Attack Tool to inject bogus routing updates into the network, changing the default gateways or destroying routing tables. Any rogue access point on the network that is not filtered by a gateway opens the network to this damaging attack.

2.4.6 Problems in IEEE 802.11i RSN

Analysis by He et al. [52] on IEEE 802.11i wireless networking highlights the weaknesses in the standard. They have discussed the possibilities of the following attacks on 802.11i networks.


2. Threat 2 - Message Injection/Active Eavesdropping.
3. Threat 3 - Message deletion and Interception.

4. Threat 4 - Masquerading and Malicious AP.

5. Threat 5 - Session Hijacking.

6. Threat 6 - Man-in-the-Middle.

7. Threat 7 - Denial of Service.

Based on the complete RSNA establishment procedure they analyze the security of 802.11i considering each threat separately. Since the management frames are not protected in a WLAN, threats 1, 2, 3, and 4 are possible during the first six message exchange of the RSNA establishment (please refer to Figure 2.3 for details of the RSNA establishment process). An adversary can send spoofed security capabilities and topological views of the network to a supplicant on behalf of an authenticator. Once this occurs, the supplicant will be forced to use inappropriate security parameters to communicate with the legitimate authenticator, or associate with a malicious AP. Alternatively, an adversary can also forge Association Requests to the authenticator with possibly weak security capabilities, which may cause problems if no further protections are adopted. However, they suggest these threats can be eliminated with EAP/802.1x authentication, if a strong mutual authentication is implemented. The authentication mechanism should prevent an adversary from forging, modifying, and replaying authentication packets, eliminating Threats 1, 2, and 3. Further, since credentials other than MAC addresses must be provided for successful mutual authentication, Threat 4 is not possible they claim.

Further, they state that threat 5 may exist even if a strong authentication mechanism is implemented. After a legitimate station has completed a successful authentication, the adversary could disconnect a station by forging Deauthentication or Disassociation messages, and resume the session with the AP on behalf
2.5 Detecting Wireless Attacks

In the foregoing section we discussed the various security issues haunting the wireless network environment. Over the last few years many researchers have been studying the weaknesses in wireless security systems and have recommended several techniques to detect attacks on wireless network. Nevertheless, the trustworthiness of these solutions is yet a major dilemma for end users. As discussed in the literature a continuous monitoring system for wireless networks is significant. Monitoring activities 24 hours 7 days a week and analyzing anomalies may reveal both existing and novel threats to the wireless networks. Most IDSs and Anomaly Based Detection Systems (ABDS) are designed to monitor wireless networks and alert the network administrator as and when an anomaly is detected. However, the rate of false acceptance or rejections in these systems is inevitably high compelling network administrators to ignore sometimes vital alarms.

The literature survey presented in the subsequent sections examines most common IDS and ABDS comparing their strengths and weaknesses.

2.5.1 Intrusion Detection Systems

Intrusion detection techniques can be categorized into misuse detection, which uses patterns of well known attacks or weak spots of the system to identify intrusions; and anomaly detection, which tries to determine whether deviation from
the established normal usage patterns can be flagged as intrusions. In view of studying the appropriateness of IDS in wireless environment we first take a look at the various misuse detection techniques employed in the wired networks. There are essentially two types considered in the wired network environment - the rule based intrusion detection and the event based intrusion detection.

Unlike rule based analysis tools that pattern match sequences of audit records to the expected audit trials of known penetrations, the state transition analysis proposed by Ilgun et al. [61] focuses on an audit record independent rule-base that is easier to read than current penetration rule bases. It also provides greater flexibility in identifying variations of known penetrations. State transition analysis also provides a modest, but intuitive procedure for rule generation, rather than ad hoc approaches that are in current use. Vigna and Kemmerer [120] extended the above work and developed a tool for Network-based Intrusion Detection - NetSTAT aimed at real-time network-based intrusion detection. It extends the state transition analysis technique to network based intrusion detection in order to represent attack scenarios in a networked environment. NetSTAT is oriented towards the detection of attacks in complex networks composed of several subnets. Although this system is effective in detecting attacks in wired networks it is not suitable for wireless network environments.

Paxson’s [91] stand-alone system “Bro” (Figure 2.5) observes network traffic directly and passively, using a packet filter. The system is conceptually divided into an “event engine” that reduces a stream of (filtered) packets to a stream of higher-level network events, and an interpreter for a specialized language is used to express a site’s security policy. The events are compared with the security policy for anomalies.

Bro uses a packet filter to capture the required packets only. It can be configured to capture the necessary packets, for example packets with destination port of 79 (Finger), 21 (FTP), or 23 (Telnet), and any TCP or UDP packets with a source or destination port of 111 (portmapper). The resulting filtered packet
stream is handed over to the next layer, the “event-engine”. This layer first performs several integrity checks to assure that the packet headers are well-formed. If the check fails the Bro generates an event indicating the problem and discards the packet. On the other hand the event engine looks up the connection state associated with the tuple of the two IP addresses and the two TCP or UDP port numbers, creating a new state if none already exists. It then dispatches the packet to a handler for the corresponding connection.

Different changes in the connection’s state generate different events. When the initial SYN packet requesting a connection is seen, the event engine schedules a timer for T seconds in the future (presently, five minutes); if the timer expires and the connection has not changed state, then the engine generates a connection_attempt event. If before that time, however, the other connection endpoint replies with a correct SYN acknowledgment packet, then the engine immediately generates a connection_established event, and cancels the connection attempt timer. On the other hand, if the endpoint replies with a RTS packet,
then the connection attempt has been rejected, and the engine generates connection_rejected. Similarly, if a connection terminates via a normal FIN exchange, then the engine generates connection_finished. It also generates several other events reflecting more unusual ways in which connections can terminate.

The “policy script interpreter” executes scripts written in specialized Bro language. These scripts specify event handlers, which are essentially identical to Bro functions except that they don’t return a value. For each event passed to the interpreter, it retrieves the (semi-)compiled code for the corresponding handler, binds the values of the events to the arguments of the handler, and interprets the code. This code in turn can execute arbitrary Bro scripting commands, including generating new events, logging real-time notifications (using UNIX syslog function), recording data to disk, or modifying internal state for access by subsequently invoked event handlers (or by the event engine itself).

In contrast to the above work, the Early Warning System (EWS) proposed by us focuses on tracking behavior of the wireless hosts. Unlike wired networks, in wireless networks the participating hosts have two distinct operating settings, pre-association and post-association. The operation of our proposed EWS centers on pre-association events of the participating hosts. We discuss these events and the behavior projection in chapter 3. Our EWS does not focus on any of the post-association states of the wireless hosts.

2.5.2 Anomaly Based Intrusion Detection Systems

Study on intrusion detection techniques for mobile wireless networks (Ad-Hoc) has been presented by Zhang et. al. [128]. They first discuss the significance of the vulnerabilities in wireless networks and highlight the need for intrusion detection systems for wireless networks. They also indicate the need for a distributed and cooperative approach for monitoring the wireless environment and propose architecture for a detection system. Furthermore, they also include an anomaly
2.5. DETECTING WIRELESS ATTACKS

detection model to detect abnormal behavior using a classifier trained using normal data to predict what is normally the next event given the previous $n$ events. The experimental results demonstrate that an anomaly detection approach can work well on different wireless ad-hoc networks. That is, the normal behavior of a protocol can be established and used to detect anomalies. Similar to this approach we also propose a distributed detection system for infrastructure wireless networks. However, in our approach the trace analysis and anomaly detection is made centrally as opposed to the cooperative approach proposed by them.

A case study using the MitM attack for wireless Intrusion detection and response by Schmoyer et. al. [99] further emphasizes the weaknesses of 802.11 based WLANs. They have enhanced the concepts proposed in [128] and have developed a framework for distributed, corporative intrusion response engine that utilizes adaptive response strategies based on alarm confidence, attack frequency, assessed risks and estimated response costs. However, this paper does not present any examination of new standards and implementations and they assume that not all existing infrastructure will be upgradeable or replaced, thus there will be still a need to find ways of increasing existing infrastructure security.

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Likelihood</th>
<th>Impact</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unauthorized Access</td>
<td>Likely</td>
<td>Medium</td>
<td>Critical</td>
</tr>
<tr>
<td>with device list-based auth.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>with manufacturer certificate based auth.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>with EAP-based auth.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rogue base station</td>
<td>Likely</td>
<td>High</td>
<td>Critical</td>
</tr>
<tr>
<td>without mutual auth.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>with EAP-based auth.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replay</td>
<td>Likely</td>
<td>High</td>
<td>Critical</td>
</tr>
<tr>
<td>without message auth.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>with HMAC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>with OMAC</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1: Risk of impersonation

Barbeau et. al. have analyzed impersonation attacks in future wireless and mobile networks [15]. They have performed a risk analysis on the possible types of impersonation attacks in wireless networks. Table 2.1 summarizes their findings. Summing up their findings they stress that the risk of impersonation in wireless
networks is critical since the threat can be materialized into several forms of
attacks. Hence they highlight the need for effective countermeasures to address
the threats.

The countermeasure proposed by [15] include the use of Radio Frequency
Fingerprint (RFF) and User Mobility Profiles (UMP) for Anomaly Based Intru-
sion Detection (ABID). RFF is a technology, which has been designed to capture
the unique characteristics of the radio frequency energy of a transceiver, in RF-
based wireless devices. The use of device and user profiles to detect anomalies has
been studied by Hall et al. [50]. They incorporate RFF into an ABID system to
associate a MAC-address of a device with the corresponding transceiver profile.
Hence, if an observed transceiver print from a claimed MAC-address, matches the
corresponding transceiver profile, then the MAC-address has not been spoofed.
They make use of this idea to verify the credibility of their UMP based ABID
system. They discuss enhancing their system by supplementing existing user and
device-based profiles, with those based on mobility. However, this system is more
suitable for addressing the problem of stolen cell phones, given that the mobility
behavior of the thief and the user are likely to be different. In the case of wireless
networks the attacker needs to be in the same domain as the user to carry out an
attack. Therefore the use of mobility profiles will not be suitable for infrastructure
wireless networks.

Although the authors claim significant improvement in the true detection rates
of anomalies in their RFF and UMP based ABID system, they have not tested the
effectiveness of their system in real-time environments. They declare the ABID
systems with real-time response is to be further researched.

The passive techniques for detecting Session Hijacking Attacks in IEEE 802.11
networks presented by Gill et. al. [46] uses Received Signal Strength (RSS)
and Round Trip Timing (RTT) for finding anomalies in wireless communications
between hosts. They have setup a simple network and have proved their system
with a number of examples. They claim their system of using RSS and RTT
works effectively against session hijacking attacks where the attacker and the legitimate STA are geographically separated and the differences in observed RSS and RTT between the attacker are very close. However, they confess, in the cases where the legitimate STA and the attacker are very close, it becomes necessary to not rely completely on the capabilities of just one technique, but rather enhance the confidence in the intrusion detection process by correlating information from numerous sources. They also admit that using a corporative and distributed system they could increase the degree of confidence in anomaly detection.

Further enhancing their research Gill et al. studied the anomaly detection response using correlation techniques in their later study [45]. Here, having detected an anomaly by either mechanism (RTT or RSS) they then wait for a confirmation from the other. For example, if the RSS detection technique registers an abrupt spike in RSS value for a particular MAC address, the correlation engine would register this and wait for the next RTS-CTS event for this MAC address and check the results of the RTT detection technique. If both techniques register an alert for that MAC address, then an alarm would be raised. From a number of experiments performed they claim nil false negative and a low number of false positives. Hence they claim anomaly detection systems could not rely only on one observation and that multiple observations is necessary together with effective correlation techniques to verify the legitimacy of each alarms raised.

Maxion et al. [82] apply benchmarking to prove that differences in data regularities influence anomaly detector performance, and such differences are found in natural environments. All anomaly-detection algorithms operate on data drawn from some kind of computing domain or environment. Embedded in each type of environment is a particular structuring of the data that is a function of the environment itself. The researchers provide quantitative results of running an anomaly detector on various data sets containing different structure. The results on different data regularities proved data regularity influences the anomaly detector performance. Hence they emphasize the need for anomaly detection systems
that can handle differences in data regularities effectively.

2.5.3 Data Mining Based Intrusion Detection Systems

The recent development in data mining has made available a wide variety of algorithms, drawn from the fields of statistics, pattern recognition, machine learning and database for the purpose of detecting irregularities in data sets. One of the major issues in intrusion detection systems is the ability to effectively verify the legitimacy of an illegitimate event in real time. The large amounts of network traces, their storage and analyzing them in real time has distinguished data mining as a promising mechanism to incorporate for intrusion detection. Lee et. al. [72], has presented several types of algorithms that are particularly relevant to intrusion detection:

- **Classification**: maps a data item into one of several predefined categories. These algorithms normally output “classifiers”, for example, in the form of decision trees or rules. An ideal application in intrusion detection will be to gather sufficient “normal” and “abnormal” audit data for a user or a program, then apply a classification algorithm to learn a classifier that will determine (future) audit data as belonging to the normal class or the abnormal class.

- **Link analysis**: determines relation between fields in the database. Finding out the correlations in audit data will provide insight for selecting the right set of system features for intrusion detection.

- **Sequence Analysis**: models sequential patterns. These algorithms can help us understand what (time based) sequence of audit events are frequently encountered together. These frequent event patterns are important elements of the behavior profile of a user or a program.
Bloedorn et al. [20] discuss how these data mining techniques could contribute to an intrusion detection project. They first identify the major challenges to intrusion detection projects as:

- Removal of normal activity from alarm data to allow analysts to focus on real attacks.
- Identify false alarm generators and “bad” sensor signatures.
- Find anomalous activity that uncovers a real attack.
- Identify long, ongoing patterns (different IP address, same activity).

Thereafter they highlight how data mining could be effectively employed to solve these issues:

- Data summarization with statistics, including finding outliers.
- Visualization: presenting a graphical summary of data.
- Clustering of data into natural categories [79].
- Association rule discovery: defining normal activity and enabling the discovery of anomalies [29, 14].
- Classification: predicting the category to which a particular record belongs [73].

When using data mining for intrusion detection one could use data at the level of TCP dump [73] or at the alarm level [79]. In both cases of data we will find fields for source IP address, destination IP address, source port number, destination port number, date/time, transfer protocol (TCP, UDP, ICMP, etc.) and traffic duration (or equivalently, both start and end times). These base attributes give a good description of the individual connection or alarm, but they often are insufficient to identify anomalous or malicious activity because they do not take
into account the larger context. For this reason we add additional fields containing values derived from the base fields. For example, we could distinguish traffic originating from outside our network from traffic originating inside our network.

Another type of derived data, called an aggregation, is a summary count of traffic matching some particular pattern. For example, we might want to know, for a particular source IP address X, and address Y, how many unique destinations IP addresses were contacted in a specific time window Z. A high value of this measure could give an indication of IP mapping, which is a pre attack reconnaissance of the network. Aggregations are generally more expensive to compute.

A third type of derived data is a flag indicating whether a particular alarm satisfies a heuristic rule. Because data mining methods handle many attributes well, and because we don’t know for sure which one will be useful, one approach is to compute a large number of attributes (over one hundred) and store them in the database with the base alarm fields.

Bloedorn et al. [20] list the following guidelines for a network intrusion detection project integrating data mining:

- Choose your requirements carefully and be realistic.
- Assemble a team with broad, relevant capabilities.
- Invest in adequate infrastructure to support data collection and data mining.
- Design, compute and store appropriate attributes with your data.
- Reduce data volumes with filtering rules.
- Refine the overall architecture for your system, taking into account both automated processing and human analysis.
- Use data mining techniques such as classification, clustering and anomaly detection, to suggest new filter rules.
- Make sure that the automated data processing can be done efficiently.
Classification is used to assign examples to pre-defined categories. Machine learning software performs this task by extracting or learning discrimination rules from examples of correctly classified data. Since classification assumes that incoming data will match that seen in the past, classification may be an inappropriate approach to finding new attacks [20].

Much of the work in outlier detection has been approached from a statistical point of view and is primarily concerned with one or very few attributes. However, because the network data has many dimensions, the use of clustering for anomaly detection has been investigated. Clustering is an unsupervised machine learning technique for finding patterns in unlabeled data with many dimensions. $K$-means clustering is used to find natural groupings of similar alarm records. Records that are far from any of these clusters indicate unusual activity that may be part of a new attack [20].

A technique called pseudo-Bayes estimators to enhance an anomaly detection system’s ability to detect new attacks while reducing the false alarm rate has been proposed by Barbara et al. [14]. This work is based on an anomaly detection system called Audit Data Analysis and Mining (ADAM).

ADAM applies mining association rules techniques to look for the abnormal events in network traffic data, and then it uses a classification algorithm to classify the abnormal events into normal instances and abnormal instances. The abnormal events can further be categorized into attack names if ADAM has gained knowledge about the attacks. With the help of the classifier, the number of false alarms is greatly reduced because the abnormal associations that belong to normal instances will be filtered out. However, the normal instances and attacks that the classifier is able to recognize are limited to those that appear in the training data.

To overcome this limitation, the pseudo-Bayes estimator method is used to estimate the prior and posterior probabilities of new attacks. Thereafter, a Naive Bayes classifier is constructed to classify the instances into normal instances,
known attacks and new attacks. Experimental results show that this method helps in detecting new attacks whose properties are different and distinguishable from the normal instances of training data. For new attacks that do not differ much from the normal instances, the method does not work well and will misclassify them as normal instances (misclassification is not unique in pseudo-Bayes, and it exists in every supervised classifier when some classes are distinguishable).

An Improved Algorithm of Fuzzy Data Mining for Intrusion Detection is proposed by Florez et al. [40]. Fuzzy logic is appropriate for the intrusion detection problem because quantitative features such as the number of different connections or messages is often used for anomaly detection and because security itself involves fuzziness [76]. In order to detect anomalous behavior, we mine sets of fuzzy association from new audit data and compute the similarity with sets mined from ”normal” data. If the similarity values are below a threshold value, an alarm is issued.

2.6 Limitations And Problems in Current Intrusion Detection Systems

In the previous sections we have reviewed about the various security mechanisms, threats, attacks and the available countermeasures. It was highlighted that there is a valid need for innovative techniques to reduce if not completely eliminate the security threats in wireless networks.

The anomaly based intrusion detection studies carried out so far has proved that a single mode of anomaly detection is insufficient. Researchers have introduced a second or third level of input to further strengthen their results. The studies also indicate the need for comparing results of the different input using correlation or similar techniques to reduce the number of false positives. For example the use of RFF and UMP is presented in [15, 50] and the use of RTT and RSS is presented in [46, 45]. These studies confirm the difficulties faced by
anomaly based intrusion detection systems due to the large number of false acceptance or rejection. Furthermore, merely having more than one technique to detect anomalies does not improve the performance of the detection system. The effectiveness of the detection system invariably depends on how the legitimacy of the detected anomalies are validated. Specially in the case of wireless networks this is a major problem since the occurrence of illegitimate conditions is considerably high due to its inherent qualities. Therefore, the above studies reveal the necessity for an effective mechanism to authenticate the legitimacy of illegitimate events in wireless anomaly based intrusion detection systems.

Most data mining based intrusion detection systems detect unusual events or anomalous behavior in networks effectively. However, the major challenge is the validation of such illegitimate events. Almost all of the work found in the literature present techniques to detect abnormal or rare events in networks. However, having found an anomalous or rare event how do we classify it as legitimate or illegitimate is the major challenge yet to be addressed. In this view, discovery of outliers to extract a few data objects with abnormal behavior patterns, which are more interesting than common patterns in some cases, can be of practical significance in intrusion detection systems, credit fraud detection, etc. Hence, we intend to exploit the use of an OLAP group outlier based data association method to address the problem of validating the illegitimate events. The data is first modeled into an OLAP data cube, and a novel parameter called the group outlier score is built over the OLAP cells. The next section provides a brief overview of OLAP and Data Cube.

2.7 OLAP and Data Cube

The multi-dimensional analysis of data can in some sense be traced back to the introduction of the programming language APL in the early 1960s [63]. While
particularly obscure syntactically, APL was interesting from a data analysis perspective in that it explicitly supported the use of multi-dimensional variables. Nevertheless, its unfriendly nature prevented it from being more widely used.

Nearly a decade later, the first true multi-dimensional product Express [88] arrived on the market. It stored data in an array-based format and permitted some degree of dimensional analysis. Over the next ten to fifteen years, numerous other decision support products, from companies such as Comshare [32], Metaphor, and Pilot [92], were developed. Over time, these systems moved away from time-shared mainframe implementations towards client/server network-centric applications. Still, none were widely supported since they either required excessive hardware resources, were limited in functionality, or were non-intuitive to use. In fact, during this time period, it was the spreadsheet that was most often associated with multi-dimensional analysis.

By the 1990s, processing power had grown to such a degree that serious data-intensive applications were now completely viable on cost-effective PC networks. In addition, the emergence of the Internet/Web lead to a marked increase in the amount of digital information available for analysis. The combination of these two factors eventually encouraged the introduction of a whole new generation of advanced multi-dimensional tools, a trend that has continued into the present time frame.

Despite the long history of applications for multi-dimensional analysis, the term OLAP was not coined until 1992. In that year, E. F. Codd, the originator of the relational database model, produced a report entitled “Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate” [31] that formally identified the new field. The following list, taken from that report, identifies the twelve features that would/should make up any OLAP application:

1. Multidimensional conceptual view - This supports “slice-and-dice” operations and is usually required in financial modeling.
2. Transparency - OLAP systems should be part of an open system that supports heterogeneous data sources. Furthermore, the end user should not have to be concerned about the details of data access or conversions.

3. Accessibility - OLAP should present the user with a single logical schema of the data.

4. Consistent reporting performance - Performance should not degrade unduly as the number of dimensions in the model increases.

5. Client/server architecture - Requirement for open, modular systems.

6. Generic dimensionality - Not limited to 3-D and not biased toward any particular dimension. A function applied to one dimension should also be able to be applied to another.

7. Dynamic sparse-matrix handling - Related both to the idea of nulls in relational databases and to the notion of compressing large files, a sparse matrix is one in which not every cell contains data. OLAP systems should accommodate varying storage and data-handling options.

8. Multiuser support - OLAP systems need to support multiple concurrent users, including their individual views or slices of a common database.

9. Unrestricted cross-dimensional operations - Similar to Rule 6; all dimensions are created equal, and operations across data dimensions do not restrict relationships between cells.

10. Intuitive data manipulation - Ideally, users shouldn’t have to use menus or perform complex multiple-step operations when an intuitive drag-and-drop action will do.

11. Flexible reporting - Users should be able to print just what they need, and any changes to the underlying financial model should be automatically reflected in reports.
12. Unlimited dimensional and aggregation levels. A serious tool should support at least 15, and preferably 20, dimensions.

The list is largely self-explanatory and clearly emphasizes the multi-dimensionality of the data and the ease with which users should be able to access it. However, while significant in that it was the first meaningful attempt to describe the OLAP environment in a structured manner, it is worth noting that the report did not become the definitive industry standard as had Codd’s earlier work on relational databases [30]. Perhaps some of the public skepticism stems from the fact that the report was commissioned by Arbor Software, a leader in the OLAP application field. Nevertheless, it remains one of the few formal OLAP specifications.

### 2.7.1 OLAP: A Functional Definition

![Figure 2.6: Roll-Up and Drill-Down on a Three Dimensional Cube](image1)

![Figure 2.7: Slicing and Dicing on a Three Dimensional Cube](image2)
Though the Codd report indirectly emphasized the functionality that a commercial OLAP application should possess, it did not explicitly define the core OLAP operations. In fact, there are five such fundamental features that have come to be synonymous with the OLAP label. The list below presents these functions, while graphical examples are provided in Figures 2.6, 2.7 and 2.8.

- **Roll-up** - The roll-up operation collapses the hierarchy along a particular dimension(s) so as to present the remaining dimensions at a coarser level of aggregation. Figure 2.2 illustrates how the “location” dimension, originally listed in a city-by-city fashion, is aggregated in order to provide provincial totals.

- **Drill-down** - In contrast, the drill-down function allows users to obtain a more detailed view of a given dimension. Again, in Figure 2.2, we see how the “product” dimension is broken down from its initial, broad categories into product-specific listings.

- **Slice** - Here, the objective is to extract a slice of the original cube corresponding to a single value of a given dimension. No aggregation is required with this operation. Instead, we are allowing the user to focus in on values of interest. Figure 2.3 illustrates the process for a single value of the “product
CHAPTER 2. LITERATURE REVIEW

dimension”.

- Dice - A related operation is the dice. In this case, we are defining a sub cube of the original space. In other words, by specifying value ranges on one or more dimensions, the user can highlight meaningful blocks of aggregated data. In Figure 2.3, a subset of dimension values on product, location, and customer have produced the $3 \times 2 \times 2$ sub cube.

- Pivot - The pivot is a simple operation that allows OLAP users to visualize cube values in more natural or intuitive ways. While Figure 2.4 provides a simple example with a symmetrical $4 \times 4 \times 4$ cube, the pivot operation can be equally effective with dimensions of varying cardinality.

2.7.2 MOLAP, ROLAP and Multi-dimensional Data

One of the most important elements of OLAP environments is their reliance upon multi-dimensional data values. Since the data cube suggests a multi-dimensional interpretation of the data space, a number of OLAP vendors have chosen to physically model the cube as a multi-dimensional array. These multi-dimensional OLAP (MOLAP) products offer rapid response time on OLAP queries since, in theory at least, it is possible to index directly into the data cube structure to retrieve subsets of aggregated data. Unfortunately, MOLAP solutions have not proven to scale effectively to large, high-dimensionality data sets [18] (though MOLAP capacity has grown significantly in recent years). The problem is that as the number of dimensions grows, the data in the data cube becomes increasingly sparse. In other words, many of the attribute combinations represented by the data cube structure do not contain any aggregated data. As such, a fully materialized MOLAP array can contain an enormous number of empty cells, resulting in unacceptable storage requirements [96]. Though compression techniques are often used to alleviate this problem, doing so destroys the natural indexing that makes MOLAP so appealing. As a result, awkward hybrid indexing schemes that
combine both sparse and dense sub-arrays are required.

In contrast, relational OLAP (ROLAP) seeks to exploit the maturity and power of the relational paradigm. Instead of a multi-dimensional array, the ROLAP data cube is implemented as a collection of relational tables, each representing a particular view of the data. Because the views are now conventional database tables, they can be processed and queried with traditional RDBMS techniques (e.g., indexes and joins). More importantly, however, they can be more efficient on very large data warehouses since only those data cube cells that actually contain information are housed within the tables. The “empty” portions of the space do not have to be represented in any way. On the downside, there is no “built-in” indexing with a ROLAP cube as there would be with a MOLAP implementation. Instead, all attribute values within the record must be included with the aggregated or summary values so that the record’s position within the cube can be determined. One might liken this to a fully-qualified path name in operating system terminology. This additional overhead tends to offset some of the space savings, particularly in dense environments. Furthermore, the absence of an implicit index implies that an explicit one must be provided. In practice, this can be a challenge of considerable importance since efficient multi-dimensional indexing techniques are complex.

### 2.7.3 The Data Cube

In the above section, we informally introduced a multi-dimensional structure called the data cube, describing it as a data abstraction that allows one to view aggregated data from a number of perspectives. In this section, we will formalize the data cube concept.

First, we introduce some of the basic terminology necessary for a thorough understanding of the problem. We begin by noting that a standard OLAP analysis environment consists of a group of dimensions, each of which has been identified
by the data warehouse designers as being of interest to the user community. Dimensions are also known as attributes; we will use these terms interchangeably throughout the remainder of the thesis. Attributes can be of two types. Feature attributes refer to those dimensions that represent entities or concepts central to the structure of the organization. Examples would be things such as customer and product. Measure attributes, on the other hand, refer to the items of interest, the values that will be aggregated in terms of the feature attributes. We note that while there may be many feature attributes, in most cases there will be a very small number of measure attributes (often just one).

Measure attributes may be calculated using functions from one of three distinct categories.

- **Distributive** - Functions in this category have the unique feature that when computed across independent data partitions, the partial results can be combined into a single aggregate. Examples are sum, min, and max.

- **Algebraic** - Simply put, an algebraic function is one that can be produced by combining distributive functions. Examples would include average and standard deviation.

- **Holistic**. Functions in this category cannot be decomposed into algebraic functions. Median and rank are common examples. In general, holistic functions are difficult to compute efficiently and practical implementations often resort to approximations.

For convenience, we will utilize a single distributive measure attribute throughout the thesis, namely *summation*. Virtually all data cube related research papers employ this same simplification.

We also note that with a \( m \)-dimension space, each of the \( m \) attributes \( A_1, A_2, \ldots, A_m \) has a cardinality that identifies the number of unique values for that attribute. For example, if one of the data cube dimensions is “Product”, and there are 275
individual products in our database, then the cardinality of Product, denoted |Product|, is 275.

In total, a \( m \)-dimensional data warehouse is associated with \((2^m - 1)\) views (sometimes referred to as the view Power Set). In OLAP terminology, views are also known as cuboids or group-bys. Each view or cuboid represents a distinct combination of feature attributes, and can be seen as depicting an aggregation of the measure attribute at a given level of granularity. For example, given the attribute set \( ABC \), we say that the aggregated view \( A \) is of coarser granularity than the aggregated view \( AB \). Note that we are interested in attribute combinations, not permutations, since the order of the attributes does not matter. In practice, we usually substitute letter labels for the attribute names. For example, Customer may be attribute ‘A’, Product may be attribute ‘B’, etc. The “Customer/Product” cuboid is therefore simply referred to as \( AB \).

\[
\begin{array}{cccc}
| & | & | & | \\
\text{ABCD} & 4-\text{D Cuboid} & | & | \\
\text{ABC} & \text{ABD} & \text{ACD} & \text{BCD} & 3-\text{D Cuboid} \\
\text{AB} & \text{AC} & \text{AD} & \text{BC} & \text{BD} & \text{CD} & 2-\text{D Cuboid} \\
\text{A} & \text{B} & \text{C} & \text{D} & 1-\text{D Cuboid} \\
\text{None} & | & | & | & 0-\text{D Cuboid} \\
\end{array}
\]

Figure 2.9: A Data Cube Lattice With all Possible Attribute Combinations

The relationship between the \((2^m - 1)\) views in terms of common attributes is typically represented by a lattice [51]. See Figure 2.9 for a graphical illustration. Starting with the base cuboid the finest granularity view containing the full complement of \( m \) dimensions the lattice branches out by connecting every parent node with the set of child nodes/views that can be derived from its dimension list. In
other words, the attributes of a parent view must be a superset of the attributes
of a child view. A parent containing \( k \) dimensions can be connected to \( k \) views at
the next level in the lattice, each of which contains \( k - 1 \) attributes. Conversely,
a child view can be associated with \( d - k \) parents (if this is not obvious, note that
because the lattice is perfectly symmetrical, the number of parents for a given
view at level \( k \) is equivalent to the number of children for a given view at level
\( d - k \)). Finally, it should be understood that parent/child relationships are not
exclusive - parents can share common children just as children may have common
parents.

Conceptually, then, the data cube consists of the base cuboid, surrounded by
a collection of \( 2^m - 1 \) sub-cubes/cuboids that represent the aggregation of the
base cuboid along one or more dimensions. Since the base cuboid contains all
feature attributes, it can be used to compute all of the other coarser cuboids by
aggregating across one or more of its component dimensions. In other words, it
may be possible to initially compute only a subset of all possible views, leaving
the materialization of the remaining views to some later time (if necessary). As
such, a data cube can be described as full if it contains all \( (2^m - 1) \) possible views,
or partial if only a subset of views has actually been constructed.

Figure 2.10 depicts a small, practical data cube example from the automotive
industry. Note that this is actually a partial cube since the single dimension views
are not depicted. This particular data cube has three feature attributes - make,
color, and year - and a single measure attribute - sales. Sales is computed with the
distributive sum function. By selecting cells (a “point” query), planes (a “slice”
query), or sub-cubes (a “dice” query) from the base cuboid, we can analyze sales
figures at varying granularities.

The Data Cube Operator Strictly speaking, no special operators or SQL ex-
tensions are required to take a raw data set, composed of detailed transaction-
level records, and turn it into a data structure, or group of structures, capable
of supporting subject-oriented analysis. Rather, the SQL group — by and union
operators can be used in conjunction with $2^m$ sorts of the raw data set to produce all cuboids. However, such an approach would be both tedious to program and immensely inefficient, given the obvious inter-relationships between the various views. Consequently, in 1995, the data cube operator - an SQL syntactical extension was proposed by Gray et al. [48] as a means of simplifying the process of data cube construction.

### 2.8 Outlier Detection

As discussed in the preceding sections, the serious limitation in most IDS and ABDS is the ability to validate the legitimacy of illegitimate events or conditions. Although there has been several techniques proposed to overcome this problem, the trustworthiness of such techniques will greatly depend on the rate of false positives/negatives raised and the ability for real time response. In this context we first present a brief survey on the various outlier detection techniques found in the literature.
2.8.1 Outlier Detection Techniques

Several methods for detecting outliers in multivariate data without apriori assumption of the distribution have been proposed. Knorr and Ng [69] proposed the distance-based approach where an object in a data set $P$ is a distance-based outlier if at least a fraction $b$ of the objects in $P$ is further than $r$ from it. This outlier definition is based on a single, global criterion determined by the parameters $r$ and $b$. However, this can lead to problems when the data set has both dense and sparse regions.

The depth-based approach computes different layers of $k$-d convex hulls [66]. Objects in the outer layer are detected as outliers. However, it is a well-known fact that the algorithms employed suffer from the dimensionality curse and cannot cope with large $k$. In the case of clustering, many algorithms detect outliers as by-products [64]. Their main objective is clustering; hence they are not optimized for outlier detection. Furthermore, in most cases, the outlier definition or detection criteria are implicit and cannot easily be inferred from the clustering procedures.

The density-based approach proposed by Breunig, et al. [23] relies on the Local Outlier Factor (LOF) of each object, which depends on the local density of its neighborhood. The neighborhood is defined by the distance to the MinPts-th nearest neighbor. In typical use, objects with a high LOF are flagged as outliers. Jin, et al. [65] proposed an algorithm to efficiently discover top-n outliers using clusters, for a particular value of MinPts. LOF does not suffer from the local density problem. However, selecting MinPts is non-trivial. In order to detect outlying clusters, MinPts has to be as large as the size of these clusters, and computation cost is directly related to MinPts. Furthermore, the method exhibits some unexpected sensitivity on the choice of MinPts. Aggarwal et al. [3] claim that both distance-based and local outliers do not work well for high dimensional dataset since the data are sparse, and outliers should be defined in sub-space projections. They proposed an evolutionary algorithm to find the outliers.

All of the above outlier detection methods are based on individual outliers, and...
the association of outliers has not been considered. In contrast, the “discovery-driven” method proposed by Sarawagi, et al. [98] aims at finding exceptions in data cube cells. In this method of data exploration an analyst’s search for anomalies is guided by pre-computed indicators of exceptions at various levels of detail in a data cube. They define a cell as an exception if the aggregate of the cell differs significantly from its anticipated value. The anticipated value is calculated by some formula and they suggest an additive or multiplicative form. They also give a formula to estimate the standard deviation. When the difference between the cell value and its anticipated value is greater than 2.5 standard deviation, the cell is considered an exception.

Similar to this work, Lin and Brown [74, 75] also focus on OLAP cube cells in their analysis. They define a function on OLAP cube cells to measure the extremeness of the OLAP cell, which is called the outlier score. Instead of defining outlier for individual record, they consider to build the outlier measure for a group of data points. These data points are “similar” on some attributes and are “different” on other attributes. If these common characteristics are quite “unusual”, or in other words, they are “outliers”, these data points are well separated from other points. Hence, they claim that this “weird” characteristics strongly suggest that these data points are generated by a particular mechanism, and could be associated. This method combines both outlier detection in data mining and concepts of OLAP. They also describe a real world example.

Although both the above methods focus on detecting exceptions they have not considered real time operations which requires very fast response as in security or health related applications. The use of OLAP cubes for online real time applications is yet to be investigated. Furthermore, maintaining large data sets and updating them in real time may be intricate and impractical.

In this view we propose to develop an outlier detection technique, which includes several novel features. We build our OLAP cubes as partial data cubes, because in many cases we don’t have to materialize all of the views. We intend
to update the data cube on-the-fly enabling a continuous learning strategy. And finally, when it becomes necessary to rebuild the partial data cube we re-construct only those views that needs to be updated. Furthermore, materializing selected views and using surrogate views for querying the data cube offer the query performance necessary for applications, such as network intrusion prevention and health monitoring systems which require real time response [103].

The proposed Early Warning System, the outlier detection technique and the research methodology are discussed in the next chapter.
Chapter 3

Methodology

3.1 Introduction

We present the research question and the associated hypothesis in this chapter. We then describe our proposed methodology together with the theoretical and experimental details.

As the first step in our research we derived a Behavior Tree (BT) model for the IEEE 802.11i protocol and formally analyzed it. During this exercise we were able to study the exact behavior of the protocol and see how the protocol flaws could be exploited to carry out various attacks on the wireless networks. We also formally prove that Malicious Association and Session Hijack attacks could be carried out on these networks if the recommended precautions are not implemented. With this experience we commenced our research on building an Early Warning System for intrusion prevention and/or detection on wireless networks.

Quantitative research is the systematic scientific investigation of quantitative properties and phenomena and their relationships. The objective of quantitative research is to develop and employ mathematical models, theories and/or hypotheses pertaining to natural phenomena. The process of measurement is central to quantitative research because it provides the fundamental connection between empirical observation and mathematical expression of quantitative relationships.
Quantitative research is widely used in both the natural sciences and social sciences, from physics and biology to sociology and journalism. It is also used as a way to research different aspects of education. Quantitative research is generally approached using scientific methods, which include:

- The generation of models, theories and hypotheses.
- The development of instruments and methods for measurement.
- Experimental control and manipulation of variables.
- Collection of empirical data.
- Modeling and analysis of data.
- Evaluation of results.

Qualitative research, on the other hand, is a field of inquiry that crosscuts disciplines and subject matters [35]. Qualitative researchers aim to gather an in-depth understanding of human behavior and the reasons that govern human behavior. Qualitative research relies on reasons behind various aspects of behavior. Simply, it investigates the why and how of decision making, not just what, where, and when. Hence, the need is for smaller but focused samples rather than large random samples, which qualitative research categorizes data into patterns as the primary basis for organizing and reporting results. Qualitative researchers typically rely on four methods for gathering information: (1) participation in the setting, (2) direct observation, (3) in depth interviews, and (4) analysis of documents and materials [80].

Having this in mind and considering the underlying nature of the problems in intrusion detection, we use a quantitative approach in our research deriving the appropriate model and the necessary mathematical theory for the proposed Early Warning System. The proposed model for intrusion prevention and/or detection includes a packet capturing module, an event engine, a timing anomaly detection
module, a behavioral anomaly detection module, an intrusion prevention module and the data mining engine. The data mining engine will be the main component of our system with the ability of processing outlier based data association requests effectively, preventing and/or detecting intrusions in real time. In this research we have developed two new outlier based data association parameters called the Group Outlier Score (GOS) and the Group Confidence Level (GCL. In a longitudinal data set the GOS defines the influence of an outlying event against the associated group of events. Whereas GCL computes the cohesiveness of a group of associated events.

### 3.1.1 Chapter Overview

This chapter is organized as follows: Section 3.2 defines the research questions, the hypothesis, the selected types of security issues considered in this research and the need for a better security. In Section 3.3 we describe details of the BT model together with the formal analysis. The proposed model is explained in Section 3.4.

### 3.2 Research Question

In the previous chapter a number of techniques used to counter the problems in the wireless networking environment were discussed. Further, the limitations in such available techniques were identified. Taking into consideration of the limitations, the following research questions are addressed in this research.

- How can we effectively detect the common wireless security threats during the IEEE 802.11i RSN association process?

- How can we substantiate the legitimacy of the detected threats in real time?
3.2.1 Hypotheses

1. Appropriately combining timing and behavioral analysis we can detect most of the common wireless security threats during the IEEE 802.11i RSN association process.

2. The legitimacy of the detected threats can be effectively substantiated by analyzing the behavior of the wireless hosts from different views of the data cube.

3. Using behavioral anomaly detection techniques false negatives are to be eliminated altogether. The number of false positives are to be limited to less than five in one hundred alarms.

4. For those threats, which can not be substantiated effectively, a threat level of LOW, MEDIUM or HIGH can be reported.

Our concept is based on the assumption that the wireless hosts are effectively secured by modern crypto systems once they become RSN associated. Hence our research does not focus on post association states of the wireless hosts. On the other hand we assume that the only possible phase where the wireless hosts are vulnerable to possible attacks is the association phase itself.

3.2.2 Types of Threats Considered

The wireless environment is subject to a variety of security threats. However, in our study we are focusing only on the more common threats. Thus we are specially interested in the following security threats:

1. Message Injection and Active Eavesdropping

2. Message Deletion

3. Masquerading and Malicious AP
3.2. RESEARCH QUESTION

4. Session Hijack

5. Man-In-The-Middle

6. Denial of Service

Threats 1 and 2 targets all three types of frames in the Link Layer, possibly breaking data confidentiality and integrity of the wireless LAN. Threats 3, 4 and 5 overpower mutual authentication and generally they arise from compositions of Threats 1 and 2 on management frames. Threat 6 hinder availability, and results from Threats 1 and 2 on any type of frames.

3.2.3 Significance of this Research

In view of addressing the above research questions we carried out several investigations. First, we modeled the IEEE 802.11i security architecture and derived a suitable model to study the behavior of wireless hosts during the first time association process. The model was developed using Behavior Trees [36] and was formally verified using the SAL model checker [49] for correctness and completeness [113, 115]. The behavior of IEEE 802.11i RSN association process during two different security threats have been analyzed and verified in [115].

Although IEEE 802.11i security architecture offers sufficient protection to the wireless environment it is up to the implementer to guarantee that all issues are addressed and the appropriate security measures are implemented. A single misconfigured station could lead the way for a cowardly attack and expose your organizational network. For example, if no authentication mechanisms are implemented an adversary could establish two separate connections to the supplicant and the authenticator to construct a MitM attack as discussed by Lynn et. al. [77]. Furthermore, if mutual authentication mechanism is not appropriately implemented an adversary will be able to launch a MitM attack and learn the PMK as illustrated by Asokan et. al. [11]. However, although these vulnerabilities are
not directly connected with 802.11i, any implementer of 802.11i should consider this problem warily, and keep monitoring the wireless hosts to guarantee proper implementation of the security mechanisms.

In this context the following sections describe our research approach to counter the various issues faced by the wireless networking environment. Firstly, we discuss the modeling aspects of the IEEE 802.11i security protocol and then demonstrate that the protocol is susceptible to various security attacks if not implemented properly. Next, we introduce our proposed Early Warning System and describe its components. A detailed record of the EWS together with the mathematical theories derived to substantiate the anomalies detected in the wireless environment are discussed in chapter 4.

3.3 IEEE 802.11i Software Model

In this section a software model for the 802.11i security architecture is developed using Genetic Software Engineering (GSE) [36]. This model is later used in our proposed intrusion detection system for behavioral anomaly detection.

By developing and analyzing the software model for IEEE 802.11i security protocol, we have shown that the security protocol is vulnerable to various security threats if not implemented properly. Although the GSE methodology adopted to develop the various RSN models itself is novel, the main aim of presenting this analysis is to illustrate the wireless protocol and the associated security issues graphically. The graphical representations facilitate easy understanding even to non-technical readers. In this view, we discuss the possibility of a number of security attacks and verify their prospects in relation to misconfigured wireless networks.

Requirements translation is the first formal step in the GSE design process. Its purpose is to translate each natural language functional requirement, one at a time, into one or more behavior trees. This translation identifies the components
(including actors and users), the states they realize (including attribute assignments), the events and decision/constraints that they are associated with, the data exchange, and the casual, logical and temporal dependencies associated with component interactions.

Following the translation and integration of the requirements, we evolve the requirements representations into design/architecture representations. The first phase of this architectural process is the “Component Architecture Transformation” and the output of that phase is a “Component Interaction Network”. The primary thing that happens here is that components, which may be represented at many points in the requirements representation, are isolated out to appear only once in the solution representation. This amounts to algorithmically transforming the integrated tree of requirements into a network of components that interact (the traditional conceptual view of a system). The final stage of this process is the “Component Behavior Projection”. Here, component behaviors are concentrated by separately projecting each component’s behavior from the integrated requirements tree. The result of this process is a skeleton behavior tree for each component that will deliver the behavior it needs to exhibit to function as an encapsulated component in the component interaction network.

The IEEE 802.11i standard defines two classes of security framework for IEEE 802.11 WLANs: RSN and pre-RSN security frameworks. This study is mainly focused on the RSN security framework since it is expected to drive the future of distributed wireless networks. STAs in the RSN environment can make contact with the ESS in one of two ways: initial contact or Roaming. In case of roaming we are not concerned of whether the STAs are navigating inter-subnet or intra-subnet since the security policy in both cases will be the same.

Clauses 8.4.1 to 8.4.10 in the IEEE 802.11i [59] standard describe the steps involved in the RSN security association life cycle. We have made use of these steps to develop the Requirements Behavior Tree (RBTs) for the RSN. Each individual functional requirement is translated into one or more corresponding
RBTs. Altogether, we assembled twelve functional requirements and an RBT was developed for each. As an example we have listed the fifth requirement below:

Requirement 5, Policy selection in ESS: The STA initiating an association shall insert an RSN IE into its (Re) Association Request whenever the targeted AP indicates RSN support. The initiating STA’s RSN IE shall include one authentication and pairwise cipher suite from among those advertised by the targeted AP in its Beacons and Probe Responses. It shall also specify a group key cipher suite specified by the targeted AP. If the RSN capable AP receives a (Re) Association request including an RSN IE, and if it chooses to accept the association, the AP shall, to secure this association use the authentication and pairwise cipher suites specified in the RSN IE sent by the STA. The AP shall then include the selected suites Association response to the STA. Once both AP and STA agree on a common security policy they are said to be at the CONNECTING state.

Figure 3.1 shows the RBT for this requirement. The shaded boxes in the tree denote assumed or missing requirements. In a similar fashion RBTs are created for all of the twelve requirements extracted from clause 8 of the standard. During the development of the RBTs we encountered several incompleteness and uncertainties in requirements. We used appropriate domain expertise to resolve these ambiguities.

Having developed all the RBTs, we systematically and incrementally integrated the twelve RBTs to construct the Design Behavior Tree (DBT) that satisfies all its requirements. During the integration process several integration issues were identified. These integration issues, which are mostly due to inconsistencies in pre and post conditions, were again resolved using appropriate domain expertise. A complete record of the modeling technique and the various models derived from the requirements can be found in [102].
3.3. IEEE 802.11I SOFTWARE MODEL

3.3.1 Component behavior projection

Component projection models for the supplicant and the authenticator are derived by systematic inspection of the DBT. We did this by simply ignoring the component-states of all components other than the one we are currently projecting. The resulting connected behavior tree for a particular component defines the behavior of the component that we will need to implement and encapsulate in the final component-based implementation. The projected component behavior for the supplicant and the authenticator are shown in Figure 3.2.

In the component projection models, both the STA and the AP are initially at the DISCONNECTED state, which means that the control port is at unauthorized state. From this state the STA begins the dot11 association by sending a...
ProbeReq signal. This dot11 association state of the STA is indicated as dot11 ASSOCIATION in the STA projection model. In case if the STA is unable to establish common security parameters with the AP, the STA will decline connection reverting to the DISCONNECTED state. Once dot11 association is completed both the STA and the AP transfer to the CONNECTING state enabling the port filters.

Next the dot1x authentication begins. During this process both the AP and the STA first transit to AUTHENTICATING state followed by AUTHENTICATED as discussed in the earlier section. However, at any instance if the supplicant is unable to establish its identity the authenticator declines connection with the STA, thereby pushing the STA to the DISCONNECTED state.

When dot1x authentication process successfully completes, both STA and the AP share a common PMK and reach AUTHENTICATED state. This state initiates the 4-way handshake. During the 4-way handshake both the PTK and the GTK are installed on both the STA and the AP. At this point both the AP and the STA become RSN-ASSOCIATED and the controlled port reach the authorized state allowing normal data traffic. During the 4-way handshake if the STA
or the AP does not disclose the correct RSN capabilities as advertised in the dot11 association process, the STA will DISCONNECT from the AP and vice-versa.

The projection models of GSE are comparable to the supplicant and authenticator PAE state machines presented in the 802.11 standards. Unlike the state machines which are created by deduction, the projection models are derived in a systematic manner from the requirements. Although the projection models do not show state transition details as in the state machines, such information can be obtained from the DBT with the requirement tracking numbers present in the BT models [113].

The projection models derived show the normal behavior of the STA and the AP. Accordingly, both STA and the AP have definite state changes with reversions permitted to DISCONNECTED states only. This normal behavior of our model does not permit state transitions from one intermediate state to another. Tracking this behavior in the software implementations of the STA and the AP can be a useful exercise for anomaly detection and intrusion prevention.

### 3.3.2 Model Checking

First, we used a special tool set [116] developed by the ARC Center for Complex Systems to automatically translate the BT model into formal notations like CSP [125] and SAL [17]. The static analysis of the translated specification is possible using different analysis tools available for CSP and SAL. In this study we have translated the integrated model for security policy selection, authentication and key distribution into SAL specification.

SAL is an integrated environment of static analysis tools that includes tools for model checking and theorem proving. In the SAL environment the systems are specified using a description language for state transition. The system properties of interest are calculated in SAL based on the system expressed as a transition system in this description language. In the SAL environment a number of tools provide support for abstraction, program analysis, theorem proving, and model
CHAPTER 3. METHODOLOGY

checking. A detailed description of the translation of BT to SAL specification is beyond the scope of this paper but only a brief overview has been provided here. In BT models concurrent systems are expressed as state transitions, hence, the translation rules for a subset of BT notation is relatively straightforward. A wider coverage of translation of BT notation can be found in [125].

The BT is represented in a single SAL transition system module. The components and their states are declared as state types in the module. The BT events that are marked with INPUT tags are translated as input variables. A set of special variables called pc1..pcN (program counters) is used to keep track of concurrent state transitions in the tree. An atomic action can be either manually specified or automatically generated from a set of BT state transitions between two external (observable) events.

![Figure 3.3: Translation of BT into SAL specification]

In the BT segment shown in Figure 3.3, the state transition from the root node (AP [Disconnected]) until the node right before the event (AP ??ProbeRequest[Receive]...
is considered as one action which is guarded by the initial value of the program counter. The SAL translation generated by the translator for the above BT segment is shown below:

```
INITIALIZATION
  pc1=1
TRANSITION
[]
A1: pc1=1
   aP'=apDisconnected;
   sTA'=staDisconnected;
   pTK'=notDone;
   port_Data'=pdValid;
   sTA_ProbeRequest'=pbSend;
   pc1'=2;
[]
A2: pc1=2 AND aP_ProbeRequestpbRecieve
   aP_ProbeRequest'=pbProcess;
   aPSSID'=idSend;
   pc1'=3;
[].
```

In the above SAL specification the program counter (pc1) has been initialized to 1. This program counter serves as a guard for the first action A1, which starts at the root of the tree and includes states until the node just before the first event. The new states of the state variables in the SAL code directly correspond to the first segment of the BT shown in Figure 3.3. The action A1 also increments the program counter pc1 to 2 to indicate the process control can now move to the next possible action. The second action, A2, is guarded by the incremented value of program counter (pc = 2) and the input variable aP_ProbeRequestpbRecieve. This input variable corresponds to the first event in the BT example illustrated in Figure 3.3. If the conditions for the second action (A2) are satisfied, then the set of state transitions occur which correspond to the BT state transitions shown after the first event in Figure 3.3.

In the next example we show how BT with multiple paths as shown in Figure 3.4 is translated into SAL code. In this example, the set of transitions in action
A3 is guarded by input variable STA_SSIDidReceive and pc1 = 3 along with the condition STA_SSID=idInvalid. If these conditions are true then the set of state transitions indicated in action A3 will be performed. The other two actions (A4 and A5) are also possible if and when the event happens. Both of these actions are guarded by a set of conditions i.e. pc1=3, STA_SSIDidReceive, STA_SSID= idValid, STA_Fields=Overlap and pc1=3, STA_SSIDidReceive, STA_SSID=idValid, STA_Fields= notOverlap, respectively. It may be noted that the program counter has been set to 3 before the process flow can continue down any of the paths (A3, A4 or A5) in the BT. This indicates that, at this point in the BT the process flow control can go to any branch that fulfills the conditions imposed on it. The SAL code for the example is shown below:

```sal
A3: pc1=3 AND STA_SSIDidReceive AND STA_SSID=idInvalid
   STA_SSID'=idReceive;
   STA_Decline'=dSend;
   STA'=staDisconnected;
   pc1'=6;
[/]
A4: pc1=3 AND STA_SSIDidReceive AND STA_SSID=idValid AND STA_Fields=overlap
   STA_SSID'=idReceive;
   STA_dot11Request'=d11Send;
   pc1'=7;
```
Once the system has been specified in the SAL environment language, a number of analyzes can be performed on the system specification. The SAL environment contains a symbolic model checker called sal-smc. It allows users to specify properties in Linear Temporal Logic (LTL) and CTL. However, the current version of SAL does not provide counterexamples for CTL properties. When users provide an invalid property in LTL, a counter example is produced. LTL formulas state properties about each linear path induced by a module (transition system). Typical LTL operators are:

- $G(p)$ (read “always $p$”), stating $p$ is always true
- $F(p)$ (read “eventually $p$”), stating that $p$ will be eventually true
- $U(p, q)$ (read “$p$ until $q$”), stating that $p$ holds until a state is reached where $q$ holds
- $X(p)$ (read “next $p$”), stating that $p$ is true in the next state

For example, the formula $G(p \Rightarrow F(q))$ states that whenever $p$ holds, $q$ will eventually hold. The formula $G(F(p))$ states that $p$ holds infinitely often. In the following section we describe how these properties are used to model check the IEEE 802.11i security protocol, using the sal-smc.

### 3.3.3 Formal Analysis

Having generated the SAL code for the IEEE 802.11i security protocol we developed a number of theorems to formally verify the model. Firstly, we verified all
the assumptions made by us during the modeling process to resolve ambiguities and/or inconsistencies. The GSE modeling technique compels all ambiguities and inconsistencies to be resolved during the requirements translation and integration. The component behavior projection models shown in Figure 3.2 demonstrate the normal behavior of our security model. Accordingly, both STA and AP are expected to transit states simultaneously as defined by the following LTL formulas.

- \( G((sTA = staConnecting) => (aP = apConnecting)) \)
- \( G((sTA = staAuthenticating) => (aP = apAuthenticating)) \)
- \( G((sTA = staAuthenticated) => (aP = apAuthenticated)) \)
- \( G((aP = rsnAssociated) => (sTA = rsnAssociated)) \)

All of the above four LTL theorems were proved confirming the normal behavior as per our component behavior projection. The normal behavior of the STA and AP was further verified with the following LTLs.

- \( U((sTA = staConnecting), ((sTA = staDisconnected) OR (sTA = staAuthenticating))) \)
- \( U((sTA = staAuthenticating), ((sTA = staDisconnected) OR (sTA = staAuthenticated))) \)
- \( U((sTA = staAuthenticated), ((sTA = staDisconnected) OR (sTA = rsnAssociated))) \)

The above LTL theorems were also proved confirming our aim that the participating components are disconnected from every intermediate state if they do not transit to the next state. We have made these state transitions mandatory in our models to protect our system from possible security threats which can arise from exploiting the unsynchronized behavior of the STA and AP.
Once the model was verified for normal behavior, we then verified the consistency of the security association process. Clauses 8.4.2 and 8.4.3 of the standard describe the security policy selection process in the WLAN. To begin with, an AP returns the SSID in response to a probe request from the STA. An STA has to decline associating with the AP if it receives an invalid SSID. The following LTL theorem was established to test this condition. The STA.SSID is a sub state in our BT model and is reached once the SSID is received from the AP.

\[ G(\text{STA.SSID} = \text{idInvalid}) \Rightarrow F(\text{STA} = \text{staDisconnected}) \]

The above LTL formula was proved to confirm that an STA eventually disconnects itself, if it receives an invalid SSID. However, if the STA receives a valid SSID it then validates the RSN capabilities advertised by the AP. If the AP is not RSN capable or does not match the capabilities of the STA, the STA disconnects itself maintaining strict RSN policy. In the following theorem we check for the validity of authentication and cipher suites.

\[ G((\text{AP.RSN} = \text{inCapable}) \lor (\text{AP.Cipher} = \text{inValid}) \lor (\text{AP.Auth} = \text{inValid})) \Rightarrow F(\text{STA} = \text{staDisconnected}) \]

The above LTL formula was proved confirming the behavior of the STA when it receives invalid RSN suites. On the other hand, when the STA advertises its capabilities the AP also performs similar validation. Once both STA and AP agree on the common security policy they both transit to CONNECTING state and continue with the authentication process.

In our BT model we introduced a component named “control.Port” to indicate the controlled port between the AP and the STA. This component was initialized to pUnauthorized state to indicate the unsecured state of the port.
During IEEE 802.1x authentication and key distribution process this port continues to remains unauthorized. Finally, only when the PTK is installed on both the AP and the STA it transits to the authorized state - pAuthorized enabling normal data traffic. Having this in mind, we used the following LTL formula to examine that the control port remains unauthorized during the authentication and the 4-way key handshake process.

\[
G((control\_Port = pUnauthorized) => ((sTA = staConnecting)OR(sTA = staAuthenticating)OR(sTA = staAuthenticated)OR(sTA = 4wayHandshake)))
\]

This formula was proved confirming the notion that the control port remains unauthorized throughout the authentication and key distribution process. Finally, we tested the authorized state of the control port with the following LTL theorem.

\[
G((control\_Port = pAuthorized) => (aP = apptkInstalled)AND(sTA = staptkInstalled)))
\]

The above LTL theorem was proved to confirm the authorized state of the control port implying PTK is installed on both AP and STA. All of the above analysis was done to verify the normal behavior of our model during the security association process. However, one must keep in mind that during this process the participating hosts will demonstrate their normal behavior if both communicating hosts operate in a synchronized manner. If we cannot guarantee the synchronized operation of the hosts the security of the participating hosts become dubious as discussed below.

**Security Issues**

As highlighted in Clause 8.4.1 permitting APs to advertise their SSIDs can lead to malicious associations. Furthermore, as shown in Figure 3.2, during the dot11
3.3. IEEE 802.11I SOFTWARE MODEL

association both supplicant and the authenticator operate independently without any form of software synchronization. Therefore, in a situation where the supplicant or the authenticator is allowed to make presumptions about the characteristics of other participating components can lead to malicious associations or Identity-Theft [9]. In case of a re-association request by a roaming STA, we first transit the STA into DISCONNECTED state before it is made to associate with the new AP. This makes the RSN more reliable so that session-hijack attacks [83] can be avoided. In our model the following LTL formula was proved to be false endorsing our decision:

\[ U((sTA = rsn\text{Associated}), ((sTA = sta\text{Disconnected}) OR (sTA = sta\text{Authenticating})) \]

Although this is a possible state transition in an RSN, we have deliberately DISCONNECTED the STA to make the RSN more reliable, i.e. an attacker cannot pretend as an Authenticating STA. Further, in Clause 8.4.2, we force the STA to DISCONNECT from the AP if it is not RSN capable. Similarly an AP DISCONNECTs itself if it sees an STA that is RSN incapable.

\[ G((aP_{Rsn = in\text{Capable}}) \rightarrow F(sTA = sta\text{Disconnected})) \]
\[ G((sTA_{Rsn = in\text{Capable}}) \rightarrow F(aP = ap\text{Disconnected})) \]

In Clause 8.4.3 both STA and AP disassociate with each other if they don’t mutually agree on a common security policy. In Figure 3.5, we have illustrated a malicious association scenario making use of this condition. The BT model shows an intruder, InAP (shaded boxes), reading messages from a legitimate AP and appropriately issuing an Association Request to the legitimate AP with a wrong RSN IE (as if it is sent by the associating STA). The legitimate AP disassociates itself assuming the STA is not RSN capable. In the mean time the intruder pretends as the legitimate AP and associates with the mistaken STA.
CHAPTER 3. METHODOLOGY

Figure 3.5: Malicious Association
3.3. IEEE 802.11I SOFTWARE MODEL

\[ G(\text{inAP\_dot11Request} = d11Send) \Rightarrow (sTA\_Rsn = \text{inCapable}) \land (aP = \text{apDisconnected}) \]

If this type of malicious association takes place the intruder can simply walk through the legitimate STA via the entire process of authentication and key distribution acquiring all the necessary security information and characteristics of the legitimate STA. This type of an association cannot be stopped by software means and requires some form of synchronization at lower layers of communication.

Let us now focus on the authentication process. As shown in Figure 3.6 this process is initiated by the STA issuing the EapStart message. The STA eventually communicates with the Authentication Server (AS) via a secure channel to validate its credentials. As said in Clause 8.4.6, if the AS does not recognize the challenge sent by the STA, the STA has to eventually reach the disconnected state as proved by the following LTL formula.

\[ G((aS\_AccChallenge = \text{stachReject}) \Rightarrow F(sTA = \text{staDisconnected})) \]

This situation also implies that STA and AP will never reach a state where the controlled port becomes authorized. The following LTL formula proves that the control port never reaches the authorized state if STA does not prove its identity.

\[ G((aS\_AccChallenge = \text{stachReject}) \Rightarrow (\text{control\_Port} = \text{pUnauthorized})) \]

We use this condition to demonstrate another malicious association scenario in Figure 3.6. The intruder having gained vital information about a legitimate STA during the dot11 association, now wants to act as a legitimate STA, shown as InSTA (shaded boxes) in the BT model. The intruder keeps monitoring the messages exchanged during the authentication process. The intruder tracks all messages and just before the AP is ready to authenticate the STA (by sending
Figure 3.6: Malicious Association during Authentication
EapSuccess message), it disassociates the STA by sending an EapFailure message as if sent from the legitimate AP. The intruder then associates with the AP receiving the EapSuccess message.

\[ G((\text{InSTA}_Eap = \text{instaFailure}) \rightarrow F(sTA = \text{staDisconnected})) \]
\[ G((\text{ap}_Eap = \text{apSuccess}) \rightarrow F(\text{InsTA} = \text{instaAuthenticated})) \]

As such, if the intruder successfully associates he will now be in possession of the PMK, thereby enabling him to continue the association process and becoming RSN associated. Thus, the intruder has penetrated the organizational network. Although, this sought of an association is difficult with the use of digital certificates or smart card like authentication options, the chances of poorly configured users associating with a wireless AP are high. Unless, every user is educated on the consequences of security breaches and are adequately trained to protect their own environments, the attackers will find these loopholes to gain access to the organizational networks. According to Clause 8.4.6, the control port remains unauthorized during and after the Authentication process. However, the STA reaches the Authenticated state once PMK is received by both AP and the STA or if the shared key is installed.

\[ G((sTA = \text{staAuthenticated}) \rightarrow (((sTA \_PMK = \text{stapmkInstalled}) \text{AND} (AP \_PMK = \text{appmkInstalled})) \text{OR}(PMK = PSK))) \]

When both STA and the AP reach the Authenticated state the AP begins the 4-way key handshake by sending the first message of the handshake process. The purposes of the 4-way handshake are:

Confirm the existence of the PMK. The second message transfer occurs only if the PMKIDs of STA and AP match each other.
Ensure that security association keys are fresh. If both STA and AP derive the transient keys they will eventually transit to RSN Associated state.

\[
G((sTA_{Key2} = K2Send) \Rightarrow (STA_{PMKID} = AP_{PMKID}))
\]

Synchronize the installation of temporal keys. For both STA and AP to reach RSN Associated state transient keys must be installed.

\[
G((sTA_{PTK} = staptkDerive) \Rightarrow F(sTA = staRsnAssociated))
\]
\[
G(((aP_{PTK} = apptkDerive)) \Rightarrow F(aP = apRsnAssociated))
\]

Other than the session hijack issue discussed earlier, the IEEE 802.11i security mechanism has another weak point in the key distribution process. In the earlier case we described how the intruder could gain access to the PMK. Now, having gained access to the PMK, we show in Figure 3.7 how the intruder can perform a session hijack. During the second and third message exchanges both the STA and AP validate the RSN IEs. They both compare the RSN IEs advertised during the dot11 association with that of the RSN IEs transferred during the second and third messages. If the RSN IEs do not match they disassociate. Therefore, an intruder monitoring the key exchanges can deliberately disassociate an STA by issuing a third message with an invalid RSN IE as if it is sent by the AP. Having disconnected the legitimate STA, the intruder will now receive the correct third message from the AP, installs the PTK and become RSN associated. As such, the intruder is now in full control of the wireless network.

However, one could argue that the above security issues are insignificant and with the strong authentication and key distribution mechanisms proposed in IEEE 802.11i, they are annulled. Yet, the point is that intruders one way or another
Figure 3.7: A Session Hijack
make use of such loopholes to gain some knowledge of the participating hosts and eventually succeed in penetrating the network. Therefore, it is our responsibility to take utmost precaution in every perspective to protect the WLAN.

The above discussion shows the importance of a complete and a consistent set of requirements together with a proven analysis technique. Issues in requirements can lead to defects in the final system. In this case such defects can lead to significant wireless attacks that could endanger an entire organization. However, merely conducting a rigorous analysis is not ultimate. It is also necessary to ensure that issues resolved during the analysis are effective and pertinent. The GSE technique together with formal verification is effective not only for requirements analysis and validation; but it also provides a systematic approach to integrate the system ensuring that all parts of the system corporate and coordinate correctly with good traceability. The formal nature of the Behavior Tree notation used within GSE requires all ambiguities, incompleteness and inconsistencies are resolved at the time of integration.

3.4 The Proposed Method

The wireless hosts attempting to connect to a wireless network has to negotiate a number of security policy information before they can associate with each other. Hence, it is evident that the time taken for this negotiation is comparable with a historical timing profile established over a period of time. The timing studies will be used to justify the first two hypotheses listed in Section 3.2.1.

Further, wireless hosts exhibit a certain behavior during the association process as per the protocol. Hence it could also be verified with a behavioral profile established over a period of time. This behavioral profile however, in principle has to match the projection model established through the modeling process discussed in [113]. Hence, analyzing the behavioral aspects of the wireless hosts can be used to justify the third and fourth hypotheses listed in Section 3.2.1.
Having discovered anomalies in timings and/or behavior, it is important to validate it before concluding that the anomaly is hazardous. The trustworthiness of the intrusion detection system will immensely depend on the number of false alarms raised. In this aspect we make use of historical data to prove the legitimacy of detected anomalies. We use an outlier based data association technique to verify the legitimacy of all illegitimate conditions during the first time association process of the wireless hosts. Hence, we need to collect sufficient data from a number of wireless hosts during their first time association process to justify the hypotheses listed in Section 3.2.1.

The legitimacy of an abnormal event will be verified using an outlier score. We intend to use outlier based data association techniques to establish this score. Our outlier detection process will include several novel features. First, we intend building partial data cubes, because in many cases we don’t have to materialize all of the views to establish the required data associations. Further, we will update the data cube on-the-fly enabling online monitoring and response. Finally, when it becomes necessary to rebuild the partial data cube we intend to re-construct only those views that needs to be updated. Materializing selected views and using surrogate views for querying the data cube offer the query performance necessary for applications, such as network intrusion prevention, health monitoring systems etc. which require real time response [103].

3.4.1 The Early Warning System

Figure 3.8 illustrates the block diagram of the proposed EWS. It includes a packet capturing module, an event engine, a timing anomaly detection module, a behavioral anomaly detection module, an intrusion prevention module and the data mining engine [104]. The data mining engine will be the main component of our system with the ability of processing outlier based data association requests effectively, preventing intrusions in real time. The EWS will have several levels of defense offering improved reliability for anomaly detection. The first level of
defense is the discovery of timing anomalies followed by the discovery of behavioral anomalies. If an event is discovered with either or both anomalies a third level of defense is triggered to validate the legitimacy of the anomalies based on historical data. Since the EWS needs to search enormous amounts of historical data in real time we intend to use parallel processing techniques to search the database. In the following chapter we take a closer look at the various modules of the EWS.

3.5 Conclusion

In this chapter we have presented our research question together with the relevant hypothesis. Further, we have discussed our research methodology justifying the hypothesis.

The GSE methodology adopted to develop the various RSN models illustrates the wireless protocol and the associated security issues graphically. The graphical representations facilitate easy understanding of the protocol and the associated security issues. In this view, we have discussed the possibility of a number of security attacks and verified their prospects in relation to misconfigured wireless networks.

Further, we have shown that analysing the state transitions of the wireless hosts during the RSN association is not sufficient to detect abnormal conditions. The formal analysis of the various attack scenarios demonstrate that an anomaly...
3.5. CONCLUSION

detection mechanism has to monitor every single message passed between the wireless hosts if meaningful decisions are to be made about an abnormal condition. Therefore, in addition to state transition analysis, tracking and analysing every message transfer offers better anomaly detection performance.

In this chapter, we also have described our proposed method in brief and adjourn the detailed explanation to the next chapter.
Chapter 4

Early Warning System

4.1 Introduction

The proposed EWS is our foundation for intrusion detection and prevention. It integrates the novel idea of using timing and behavioral analysis for anomaly detection together with effective substantiation.

With the use of a timing anomaly detector we are able to detect timing related anomalies in wireless networks. However, timing anomalies could occur not only due to attacks directed towards the wireless networks, but also due to the random nature of the wireless communication itself. Therefore, we cannot rule out every timing anomaly as a catastrophic threat. Similarly, behavioral anomalies too could not be categorized as hazardous merely because we detect one or two events that are abnormal. Abnormal events can get triggered due a number of reasons, including the malfunction of wireless hosts or configuration issues in wireless hosts.

In this context, validating a detected anomaly is inevitable. Hence, as discussed in the earlier chapter, we decided to use outlier based data association techniques to substantiate the anomalies detected in the wireless environment [106]. In this chapter we discuss in detail the detection mechanisms used for identifying timing and behavioral anomalies and the theoretical analysis of OLAP group outlier parameters used for substantiating thus detected anomalies.
4.1.1 Chapter Overview

In Section 4.2 of this chapter we explain the EWS and in Section 4.3, the concept of OLAP, Data Cubes and the mathematical theory established to substantiate anomalies are described. Finally, in Section 4.4, the experimental platform used for various testing purposes is described together with some examples of timing, behavioral and outlier detection analysis.

4.2 The EWS

The proposed EWS (Figure 3.8) includes a packet capturing module, an event engine, a timing anomaly detection module, a behavioral anomaly detection module, an intrusion prevention module and the data mining engine. The data mining engine will be the main component of our system with the ability of processing outlier based data association requests effectively, preventing intrusions in real time. The EWS will have two levels of defense offering improved reliability for anomaly detection. The first level of defense is the discovery of timing and behavioral anomalies. If an event is discovered with either or both anomalies a second level of defense is triggered to validate the legitimacy of the anomalies based on historical data. Since the EWS needs to search enormous amounts of historical data in real time we intend to use parallel processing techniques to search the database. In the following chapter we take a closer look at the various modules of the EWS.

The packet capture module will capture wireless data in promiscuous mode and delivers the captured IEEE 802.11 management frames to the event engine. The event engine will perform several reliability checks to assure the correctness and completeness of the management frames. If the check fails the EWS needs to generate a log indicating the problem and discard the packet. On the other hand the event engine looks up the connection state associated with the management frame and has to generate the corresponding event to be added to the master
4.2. THE EWS

The event engine also has to keep track of missing and/or dubious message flows. For successful association between two wireless hosts every message fired from the sender needs to be acknowledged by the receiver. However, due to intended or unintended reasons there may be situations where round trip messages may not reach the respective hosts within the expected time frames. This type of situations could be the beginning of an intended attack on the wireless network and has to be traced effectively. The event engine has to track such dubious message flows for analysis by the intrusion prevention module.

4.2.1 Anomaly Detection

![Anomaly Filter](image)

Figure 4.1: Anomaly Filter

As shown in Figure 4.1 anomaly filters detect anomalies that are outside the theoretical or practical behavior region of a protocol. Anomalies in these regions result in a large number of false positives in wireless networks due to the inherent qualities of the wireless environment. Anomaly detection systems that operate within this region do not effectively detect anomalies that follow theoretical or practical behavior of the protocol. Anomalies outside the theoretical or practical behavior regions occur often and hence are usually easy to detect and substantiate. On the other hand, anomalies within the theoretical or practical behavior regions are rare and lead to false negative reporting. Detecting these anomalies is
challenging and requires analyzing the protocol behavior from different perspectives. Our proposed system detects such anomalies using timing inconsistencies and then substantiates the anomalies by analyzing its behavior from different view points. A wireless host which behaves normally from one view point can behave differently from another view point.

The first stage in our anomaly detection process is the examination of timing anomalies. This is achieved by maintaining a timing profile for every participating host in the wireless network. For example, we maintain the mean and the standard deviation for each round trip message transfer i.e. the time taken to send a message and to receive the acknowledgment. A timing anomaly is triggered when a host exhibits an abnormal timing value during a message exchange. Typically, a variation of one standard deviation is regarded as abnormal [114], since our desire is to detect all possible anomalies in the wireless environment. The timing profiles are dynamically updated in the data cube with new values for mean and standard deviation depending on the current operational nature of the wireless network. However, the intrusion prevention module takes precedence over all decisions made by this module.

The second stage is the behavioral anomaly detection. Here, the current behavior of the wireless host during the association phase is compared with that of a normally behaving host. A normally behaving host has to traverse all of the legitimate states of the RSNA process as shown in the RSN projection model (Figure 3.2). If anomalies occur there can be situations where hosts fall into illegitimate states and fail to match the projection model. However, our system does not instantly classify such anomalies as illegitimate, instead, they trace such events and forward them to the intrusion prevention module for further processing.

The third stage of defense is the most important in our system and the module that executes this is called the intrusion prevention module. This module has to arrive at important decisions to verify the legitimacy of the anomalies discovered by the previous modules. Hence this module plays a vital role in maintaining the
reliability and dependability of our system. In order to facilitate real-time intrusion prevention, we adopt an efficient querying technique to search our database for outlier-based data associations.

```
{AP1, Authenticating, STA1, Disconnected, 98.4}
{AP1, Associated, STA3, Disconnected, 45.1}
{AP1, Authenticated, STA5, Connecting, 67.7}
{AP1, Disconnected, STA8, Authenticating, 45.6}
{AP1, Authenticated, STA12, Connecting, 67.7}
{AP1, Authenticating, STA17, Disconnected, 65.7}
{AP1, Associated, STA23, Disconnected, 67.2}
{AP1, Authenticated, STA25, Connecting, 75.5}
{AP1, Disconnected, STA36, Authenticating, 78.7}
{AP1, Authenticated, STA53, Connecting, 74.0}
{AP2, Disconnected, STA5, Authenticating, 79.7}
{AP2, Connecting, STA11, Disconnected, 92.4}
{AP2, Connecting, STA15, Authenticated, 94.0}
{AP2, Authenticating, STA17, Associated, 4.3}
{AP2, Associated, STA14, Connecting, 78.3}
{AP2, Connecting, STA15, Authenticated, 78.6}
{AP2, Authenticated, STA57, Associated, 7.8}
{AP2, Associated, STA64, Connecting, 68.8}
```

Table 4.1: Behavioral Associations

The intrusion prevention module can execute a number of data association analyses to decide whether anomalies detected by the anomaly modules are significant or not. One such method is to explore the connection states between the access points and stations as shown in Table 4.1. In this association, all states associated with stations and access points are investigated. The numerical values in each row indicate the support of the association obtained from the ratio of counts between a particular event and its parent. For example, if a station is found to exhibit both timing and behavioral irregularity, we will search the database for state space associations relating to the connection states of the station and the corresponding access point. Thus, if we find an association with significant remoteness, we will consider this as an anomaly and take appropriate action. On the other hand, if an anomaly does not meet the required threshold, we can ignore it and update the profiles to reflect such states. However, if a situation arises where the database does not contain any state corresponding to that was searched, we can leave it for the discretion of the network administrator to add such profiles for future analysis or not.
Table 4.2: Access Associations

Table 4.2 shows another form of data association that can be considered by our intrusion prevention module. This association is between the access points and the stations and gives an indication of which stations are mostly associated with a particular access point. A station exhibiting behavioral anomaly is most likely to roam between several access points and hence data associations between stations and access points can be used to track stations roaming abnormally. In Table 4.2 the association with support 1.1 is considered abnormal because it exhibits very low support. Hence, such data associations can also be used to verify anomalies and update the profiles suitably. Thus, the major challenge of the intrusion prevention module is the ability to verify the legitimacy of abnormal events from different views. Hence, the more data associations analyzed will present more precise information on the legitimacy of an anomaly.

4.3 Substantiation

The intrusion prevention module executes a number of data association analysis to decide whether anomalies detected by the anomaly modules are legitimate or not. This is achieved by analyzing the detected anomaly from different view points. Wireless attacks such as "Session Hijack“ or ”Man-In-The-Middle” do exactly follow the protocol but can exhibit differences in timings. Hence, such attacks cannot be substantiated merely by looking at just one characteristic of the wireless environment. Instead, analyzing anomalies caused by these attacks from different view points can reveal better results. An anomaly can exhibit
normal behavior from one characteristic, but behave abnormally on a different characteristic. Therefore, the more characteristics analyzed will deliver better detection results than merely analyzing single characteristic. In the following sections we discuss the details of our substantiation techniques.

4.3.1 OLAP and Data Views

A popular model for On-line Analytical Processing (OLAP) applications is the multidimensional database also known as the data cube [48]. A data cube consists of two kinds of attributes: dimensions and measures. The set of dimensions may consist of elements like IP addresses, port addresses, event identities etc. that together form a key. Measures are typically numeric elements like packet size, duration etc. Data cube queries represent an important class of OLAP queries in decision support systems. The pre-computation of the different views of a data cube (i.e., the forming of aggregates for every combination of GROUP BY attributes) is critical to improving the response time of the queries [34]. However, in many cases not all views are needed for decision making, therefore it is advantageous to use only selected views. Such cubes with only selected views are referred to as partial data cubes [33].

Figure 4.2 shows a data cube with thirty one views made up of five attributes. In an actual application although the number of attributes can be many, the required number of views may be very few depending on the requirements. For example, if we assume that attribute “A” and “B” represent the identities of APs and STAs respectively, then for data association analysis between the APs and STAs we will consider only those views that consists of attributes “A” and “B”, i.e. views “ABCDE”, “ABCD”, “ABCE”, “ABDE”, “ABC”, “ABD”, “ABE” and “AB”. Furthermore, in some cases we may also need to establish associations with child views to further strengthen our validation. In this manner, for intrusion detection purposes we selected the appropriate set of attributes, bearing in mind of the various security threats on the wireless networking environment.
Table 4.3 presents the list of attributes that are stored in the data cube for our analysis. Since we are concerned with wireless traces during RSNA, we store only those attributes that are necessary for this purpose. In this respect the identities (Source and Destination IDs) of the two communicating hosts, the current message (event) passed, the time during the message is passed, the Channel, Protocol and Cipher used are stored in the data cube.

The attribute “Event ID” can take values 0 to 56 (please refer Appendix B), representing the 57 different messages exchanged during an RSNA [21, 62, 89, 1]. The attribute Time of Day represents one of twenty four time periods during the day. It starts from value 0 (for time period midnight - 1.00 am) and continues up to value 23 (for time period 11.00 pm - midnight). Channel ID is from 1 to 11 for IEEE 802.11b/g networks and 36 to 161 (36, 40, 44, 48, 52, 56, 60, 64,
4.3. **SUBSTANTIATION**

149, 153, 157, 161) for IEEE 802.11a networks. Protocol ID ranges from 0 to 5, representing the six protocols; IEEE802.11, EAP, TLS, PEAP, LEAP and EAPOL considered in our study. The Cipher ID is 0 for AES/CCMP, 1 for TKIP and 2 for WEP. In addition to these attributes we can also store information related to the network identities and traffic related information if desired. The examples considered in our study are mainly focusing on wireless networks that adopt IEEE 802.11i security mechanism.

### 4.3.2 Data Views Vs Security Threats

<table>
<thead>
<tr>
<th>ABCDFG</th>
<th>BCDFG</th>
<th>ABCFG</th>
<th>BDF</th>
<th>BDFG</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABCDFG</td>
<td>AF</td>
<td>BF</td>
<td>BEF</td>
<td>ABDE</td>
</tr>
<tr>
<td>ADE</td>
<td>BCFG</td>
<td>AC</td>
<td>ABEF</td>
<td>ACE</td>
</tr>
<tr>
<td>AE</td>
<td>ACDEFG</td>
<td>ACD</td>
<td>ABEG</td>
<td>BDG</td>
</tr>
<tr>
<td>BCEFG</td>
<td>AB</td>
<td>BDEF</td>
<td>ACEFG</td>
<td>ABF</td>
</tr>
<tr>
<td>ABDEF</td>
<td>ABCE</td>
<td>ADG</td>
<td>ACDE</td>
<td>BDEG</td>
</tr>
<tr>
<td>ABC</td>
<td>BE</td>
<td>ABCDEFG</td>
<td>AEF</td>
<td>ABDEG</td>
</tr>
<tr>
<td>ABCEFG</td>
<td>BEG</td>
<td>BD</td>
<td>ADF</td>
<td>AD</td>
</tr>
<tr>
<td>AG</td>
<td>ABD</td>
<td>BCE</td>
<td>AEG</td>
<td>BC</td>
</tr>
<tr>
<td>ADEF</td>
<td>ABDG</td>
<td>ABDF</td>
<td>ABG</td>
<td>ACFG</td>
</tr>
<tr>
<td>BDE</td>
<td>BG</td>
<td>BCDEFG</td>
<td>ABE</td>
<td>ABCDE</td>
</tr>
<tr>
<td>ADEG</td>
<td>A</td>
<td>BCDE</td>
<td>BCD</td>
<td>B</td>
</tr>
<tr>
<td>ACDFG</td>
<td>ABCD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4: Selected Views

Having decided on the attributes, next, we established the views that are necessary to substantiate the security threats. Table 4.4 shows all of the views considered in our analysis. As mentioned earlier, almost all of the views selected include either or both attributes A and B. Using these views we can query the data cube on any of the associations related to access points and/or hosts. However, since we are interested in substantiating security threats it is important to establish a relationship between these views and security threats.

Table 4.5 shows the relationship between data cube views and some common security threats. This table was established considering each threat and identifying the attributes that are associated with the threat. Having identified the attributes we then categorized the views that are vital for substantiating the security threat. For example in the case of Threat 1 - Replay Attack, we need to
Table 4.5: Views Associated With Security Threats

<table>
<thead>
<tr>
<th>No.</th>
<th>Threat</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Replay attack</td>
<td>ABCDE, ABCE, ABCD, ABC,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACD, ACE, BCD, BCE, AC, BC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCDEFG, BCFG, BCDF, BCDG,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCEF, BCEG, BCD, BCE, BCF,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BCG, BC, BD, BE, BF, BG</td>
</tr>
<tr>
<td>2</td>
<td>Masquerading and Malicious AP</td>
<td>ABCDE, ABCF, ABCE, ABC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABF, ABF, AB</td>
</tr>
<tr>
<td>3</td>
<td>Session Hijack</td>
<td>ABCDE, ABCF, ABCE, ABC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABF, ABF, AB</td>
</tr>
<tr>
<td>4</td>
<td>Man-In-The-Middle (MitM)</td>
<td>ABCDE, ABCF, ABCE, ABC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABF, ABF, AB</td>
</tr>
<tr>
<td>5</td>
<td>Denial of Service (DoS)</td>
<td>ABC, AB</td>
</tr>
</tbody>
</table>

track the source and destination of every message in addition to the event identity. A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack by IP packet substitution (such as stream cipher attack). Hence, attributes A, B and C are important to analyze this type of attack. However, attributes D (Time of Day) and E (Channel ID) can be used to further refine the analysis. Therefore, to substantiate this threat we consider all views that are associated with these attributes.

In the case of Threat 2 - Masquerading and Malicious AP, an adversary uses a legitimate hosts’ identity to masquerade as a legitimate host. Here, we consider attributes B (Destination ID) and C (Event ID), since we need to track messages directed towards the Malicious host. For this type of threats attributes F (Protocol ID) and G (Cipher ID) are also significant since the masquerader may want to force a legitimate host to downgrade its capabilities in order to secure access. Hence, views BCFG, BCF and BCG, BF and BG will be important.

Threat 3 - Session Hijack is a more advanced attack, where the association between a legitimate station and the access point is hijacked by an illegitimate user. In this case the illegitimate user will force a channel change with the access point/station and masquerade as a legitimate access point/station. Hence, in this kind of a threat we need to track the source, destination, event, channel and protocol of the messages exchanged. By tracking the protocol we establish whether
the illegitimate session establishes a different kind of association. Therefore, for this type of a threat, views associated with attributes A, B, C, E and F are are considered.

Threat 4 - Man-In-The-Middle attack is not very much significant in the context of effective confidentiality measures. However, if a MitM attack turns into a session hijack attack then our detection mechanism can be of use. Hence, we can make use of the same attributes as for session hijack attack. Furthermore, if a MitM is actively participating in the communication between the two legitimate hosts, then a timing anomaly detector will be able to detect some form of timing anomaly. But, it will again be the intrusion prevention module that will have to substantiate the anomaly.

For Threat 5 - Denial-of-Service attack we consider the events associated with the source and the destination. Hence, attributes A, B and C will be considered to substantiate this attack.

The rationale for using data cubes is (i) we can readily utilize the advantages of OLAP, such as systematic storage of historical data and fast querying (ii) by linking the threat to the data cube views we can provide a classification about the type of threat and the possible solutions and (iii) the scaling of the entire system to meet the ever growing needs of a computer network.

### 4.3.3 Outlier Detection

In statistics, an outlier is a single observation “far away” from the rest of the data. Often statistics derived from data sets that include outliers are misleading. For example, if we are calculating the average temperature of some objects in a room, and most are between 20 – 25°C, but an oven is at 350°C, the median of the data may be 23 and the average temperature may be 55. In this case, the median is more likely to reflect the temperature of a randomly sampled object than the mean. Therefore, outliers may be indicative of data points that belong to a different population than the rest of the sample set. In this view, outliers
can be of significant use in intrusion detection applications for detecting abnormal conditions. Hence, before we explore the use of outliers for intrusion detection, we formally define the concepts and notations used in the rest of the chapter.

**Axioms**

$A_1, A_2, \ldots, A_m$ are the $m$ attributes we consider in our study, and $D_1, D_2, \ldots, D_m$ are their domains respectively. Let $z^{(i)}$ be the $i$th incident, and $z^{(i)}.A_j$ be the value of the $j$th attribute of incident $i$. $z^{(i)}$ can be represented as $z^{(i)} = (z_1^{(i)}, z_2^{(i)}, \ldots, z_m^{(i)})$, where $z_k^{(i)} = z^{(i)}.A_k \in D_k, k \in \{1, \ldots, m\}$. $Z$ is the set of all incidents.

**Definition 1. The Cell**

The concept of the cell and other concepts used in this paper are similar to the concepts used in the field of OLAP [48, 87].

Cell $c$ is a vector of the values of attributes with dimension $t$, where $t \leq m$. Therefore, a cell is a subset of the Cartesian product of $D_1 \times D_2 \times \ldots \times D_m$. A cell can be represented as $c = (c_1, c_2, \ldots, c_t)$, where $i_1, \ldots, i_t \in \{1, \ldots, m\}$, and $c_{i_t} \in D_{i_t}$. In order to standardize the definition of a cell, for each $D_i$, we add a “wildcard” element “*”. Now we allow $D'_i = D_i \cup \{\ast\}$. For cell $c = (c_1, c_2, \ldots, c_t)$, we can represent it as $c = (c_1, c_2, \ldots, c_m)$, where $c_j \in D'_j$, and $c_j = \ast$ if and only if $j \notin \{i_1, i_2, \ldots, i_t\}$. $c_j = \ast$ means that we do not care about the value on the $j$th attribute. $C$ denotes the set of all cells. Since each incident can also be treated as a cell, we define a function $Cell : Z \to C$. if $z = (z_1, z_2, \ldots, z_m)$, $Cell(z) = (z_1, z_2, \ldots, z_m)$.

**Definition 2. Contains**

Cell $c = (c_1, c_2, \ldots, c_t)$ contains incident $z$ if and only if $z.A_j = c_j, j \in \{i_1, \ldots, i_t\}$. With the “wildcard” element “*”, we can also say that cell $c = (c_1, c_2, \ldots, c_m)$ contains incident $z$ if and only if $z.A_j = c_j$ or $c_j = \ast, j =$
4.3. SUBSTANTIATION

1, 2, ..., m. We also say cell $c' = (c'_1, c'_2, \ldots, c'_m)$ contains cell $c = (c_1, c_2, \ldots, c_m)$ if and only if $c'_j = c_j$ or $c'_j = \ast$, $j = 1, 2, \ldots, m$.

Definition 3. Cell contents

We define function content as $\text{content}(c) : C \rightarrow 2^Z$, which returns all the incidents that cell $c$ contains. $\text{content}(c) = \{z | \text{cell } c \text{ contains } z\}$.

Definition 4. Count

Function count is defined in a natural way over the non-negative integers. $\text{count}(c)$ is the number of incidents that cell $c$ contains. $\text{count}(c) = |\text{content}(c)|$.

Definition 5. Parent Cell

Cell $c' = (c'_1, c'_2, \ldots, c'_m)$ is the parent cell of cell $c$ on the $k$th attribute when $c'_k = \ast$ and $c'_j = c_j$, for $j \neq k$. Function parent$(c, k)$ returns parent cell of cell $c$ on the $k$th attribute.

Definition 6. Neighborhood

$P$ is called the neighborhood of cell $c$ on the $k$th attribute when $P$ is a set of cells that takes the same values as cell $c$ in all attributes but $k$, and does not take the wildcard value $\ast$ on the $k$th attribute, i.e., $P = \{c^{(1)}, c^{(2)}, \ldots, c^{(|P|)}\}$ where $c^{(i)}_l = c^{(j)}_l$ for all $l \neq k$, and $c^{(i)}_k \neq \ast$ for all $i = 1, 2, \ldots, |P|$. Here, $|P|$ is the number of attributes in cell $c$ minus one. Function neighbor$(c, k)$ returns the neighborhood of cell $c$ on attribute $k$. Neighborhood can also be defined in another way: the neighborhood of cell $c$ on attribute $k$ is a set of all cells whose parent on the $k$th attribute are same as cell $c$.

The above six definitions are obtained from the OLAP area with some changes to the words used as discussed in [74]. Having introduced a common set of notations
for the data cube terminology, we now derive the necessary mathematical expres-
sions to establish the concept of *Group Outliers*.

**Definition 7. Relative Frequency**

\[
freq(c, k) = \frac{\text{count}(c)}{\sum_{c' \in \text{neighbor}(c, k)} \text{count}(c')}
\]  

(4.1)

We call \(freq(c, k)\) given by equation 4.1 as *relative frequency* of cell \(c\) with
respect to attribute \(k\). The relative frequency can also be defined as:

\[
freq(c, k) = \frac{\text{count}(c)}{\text{count}(\text{parent}(c, k))}
\]  

(4.2)

Here \(\text{count}(c)\) is the number of incidents that a data cube cell \(c\) contains. We
then define the *normalized support* for cell \(c\) as follows:

\[
N(c, k) = -freq(c, k) \log(freq(c, k))
\]  

(4.3)

Higher values for *normalized support* indicate lower *relative frequencies*. There-
fore, an event with very high \(\text{count}\) value will have lessor *normalized support* value
compared to those events with less \(\text{count}\) values.

**Definition 8. Uncertainty Function**

Next, we define function \(U\) to measure the uncertainty of a neighborhood. This
uncertainty measure is defined in terms of the relative frequencies of the neigh-
borhood. If we use \(P = \{c^{(1)}, c^{(2)}, \ldots, c^{(|P|)}\}\) to denote the neighborhood of cell \(c\)
on attribute \(k\), then \(U : R^{|P|} \rightarrow R^+\), where

\[
U(c, k) = U(freq(c^{(1)}, k), freq(c^{(2)}, k), \ldots, freq(c^{(|P|)}, k))
\]  

(4.4)

\(U\) is symmetric for all cells \(c^{(1)}, c^{(2)}, \ldots, c^{(|P|)}\). \(U\) takes a smaller value if the
“uncertainty” in the neighborhood is low. One candidate uncertainty function that satisfies the above properties is the entropy: $H(X) = -\sum p_i \log(p_i)$ [74]. Now, we have $U(c, k) = H(c, k)$, where

$$H(c, k) = -\sum_{c' \in \text{neighbor}(c, k)} \text{freq}(c', k) \log(\text{freq}(c', k)) \quad (4.5)$$

This is also the expression for entropy, conditional on the neighborhood. When $\text{freq} = 0$, we define $0 \log(0) = 0$, as in information theory. Further, in this case the log scale is used to quantify the information conveyed by any one message of a group of messages with equal likelihood.

The uncertainty function provides an estimate as to how much abnormal an incident is with respect to its neighbors. Hence, using this function we can establish a score value to determine the extent to which an anomaly has occurred. We name this score value as the group outlier score and we define it in the next section.

**Group Outlier Score**

Function $G$ is defined to measure the abnormality of an incident within the group it represents and we call it the Group Outlier Score (GOS). The more abnormal an incident, the more higher value for GOS it gets. Function $G$ is defined as:

$$G(c, k) = -\left(\frac{\log(\text{freq}(c, k))}{H(c, k)}\right) \quad (4.6)$$

When $H(c, k) = 0$, we say $\frac{\log(\text{freq}(c, k))}{H(c, k)} = 0$. We also verify that this function satisfies the following properties:

1. If $c^{(1)}$ and $c^{(2)}$ are two one dimension cells, and both of them take non-\(^*\) values on the same attribute, then $G(c^{(1)}, k) \geq G(c^{(2)}, k)$, iff $\text{count}(c^{(1)}) \leq \text{count}(c^{(2)})$. 


2. Assume that \( c^{(1)} \) and \( c^{(2)} \) are two one-dimension cells, and they take non-* values on two different attributes, say \( i \) and \( j \) respectively. If \( \text{freq}(c^{(1)}) = \text{freq}(c^{(2)}) \), then
\[
G(c^{(1)}, i) \geq G(c^{(2)}, j)
\]
holds iff \( H(c^{(1)}, i) \leq H(c^{(2)}, j) \).

3. \( G(c^{(1)}, k) \geq G(c^{(2)}, k) \) always holds
if \( \exists k, c^{(2)} = \text{parent}(c^{(1)}, k) \).

From the above declarations we can derive the GOS values for each of the three EAP group events under ideal conditions. Under ideal conditions, with no abnormalities the information ensued by all events must be the same. Hence, in practical situations we can categorize events with GOS values away from a given threshold value as outliers. In our study we establish a common threshold value to suit all three authentication mechanisms, EAP-LEAP, PEAP and TLS, since they all have almost equal number of associated events. Details of the EAP events and the derivation of the threshold value are discussed in later chapters. Having introduced the GOS value, we next define another measure for estimating the level of confidence of a group of associated events.

**Group Confidence Level**

Using the mathematical expression for *Uncertainty Function* we establish the notion of *Group Confidence Level* GCL in relation to a group of associated events. This uses the relative frequencies of \( n \) disjoint events associated with attribute \( k \) and is defined as the ratio between the abnormal behavior and the normal behavior during a similar security association process. Hence, the *Group Confidence Level* \( C \) is defined by the following equation as a percentage.

\[
C(c, k) = \left[ -\sum_{c' \in \text{groupevents}} \frac{\text{freq}(c', k) \log(\text{freq}(c', k))}{\log(n)} \right] \times 100\% \quad (4.7)
\]

In information theory, if a message recipient may expect any one of \( n \) possible messages with equal likelihood, then the amount of information conveyed
by any one such message is quantified as $\log n$. The Group Confidence Level $C$, thus obtained could be used as a scale to measure the cohesiveness of a group of associated events. Hence, a GCL of 100% assures that the group events are highly cohesive. As the GCL declines the cohesiveness of the group events also decline indicating some abnormality. Therefore, the use of GCL becomes important to study the union of group events as a whole rather than looking at the individual events as in the case of GOS. Thus, in our study we use the GCL to measure the cohesiveness of EAP-LEAP, PEAP and TLS authentication mechanisms to substantiate any anomalies. In the next section we describe our experimental setup and give examples for timing, behavior and outlier detection studies.

4.4 Experimental Platform

![Experimental Setup Diagram]

Our experimental data capturing and analyzing setup is shown as an overall diagram in Figure 4.3. It consists of a small Beowulf cluster and several 802.11i enabled APs. The APs are configured to communicate with 802.11i enabled STAs. The Beowulf cluster has three nodes including the head node. The head node is
a Pentium 4 machine with 1GB internal memory and 120GB secondary storage. The back nodes are all Pentium III machines with 256MB internal memory and 40GB secondary storage. The head node also includes a Dlink AG530 wireless adapter configured to capture wireless network traffic in promiscuous mode. The captured packets are then passed to the event engine for further processing. The head node (Node01) also runs the RADIUS server software and acts as an authentication server for the access points. In this case we have used only four nodes in our cluster. The number of nodes in the Beowulf cluster could be increased without hindering the activities of the data capturing and analyzing setup, depending on the amount of wireless traces to be processed. Our present setup has one monitoring device and is exposed to the usual campus wide wireless traffic. However, in a distributed monitoring environment we can have more than one monitoring device. Using this experimental system we have conducted several tests to verify our hypothesis.

The experimental results discussed below were obtained from wireless traces captured from our test environment shown in Figure 4.4. The test bed consists four hosts and a CISCO Aironet 1130AG Access Point (AP1). The AP’s SSID was set as “kandy” with broadcast turned ON. The ciphers was set to AES CCMP +
4.4. EXPERIMENTAL PLATFORM

TKIP with mandatory WPA key management. On the CISCO Aironet 1130AG, WPA key management stands for 802.11i/WPA2. The authentication mode was set as Open + EAP.

Three of the four hosts were installed with Fedora Core 4 Linux (kernel 2.6.11) [38] and one with Windows XP. “Monitor” is a Linux box with a Dlink DWL AG530 wireless card, STA1 is a Linux box with Dlink DWL G520 wireless card, STA2 is a Windows box with a Ubiquiti SWX-SRC wireless card and the “Adversary” is again a Linux box with Dlink DWL AG530 wireless card. All network cards used in the test bed were capable of performing 802.1x authentication. Further, the Dlink DWL AG530 and Ubiquiti SWX-SRC wireless cards are 802.11 a/b/g compatible. Hence, our experimental setup can be configured to work on any wireless network configuration.

```plaintext
ssid="kandy"
proto=RSN
key_mgmt=WPA-EAP
pairwise=CCMP TKIP
group=TKIP
eap=TLS
identity="Sith@wireless.net"
ca_cert="/etc/1x/1/root.pem"
client_cert="/etc/1x/cert-clt.pem"
private_key="/etc/1x/cert-clt.pem"
private_key_passwd="whatever"
priority=1
```

Table 4.6: wpa_supplicant.conf

For 802.1x operation we installed wpa supplicant (version 0.4.9) [126] on the Linux boxes and on the Windows platform we used the client software supplied by the wireless card manufacturer. Further, on the Linux platform we had to configure the wpa supplicant to integrate “madwifi” driver (r1417-20060128) [78] since the Dlink card uses Atheros chipset. Part of the wpa configuration options used to establish the RSN association are shown in Table 4.6.

Moreover the access point was wired to a Linux box (Node01) as shown in Figure 4.3. For 802.1x authentication purposes Freeradius (version 1.1.0) [42] was installed on the Linux box. The Radius server was configured to operate
on all EAP authentication modes. For EAP-TLS authentication we created self-
signed certificates using the utilities provided by Openssl. These certificates were
installed on the server and the clients.

To perform wireless network attacks we used “aireplay-ng” software [4] on the
third Linux box (Adversary). Using the “Adversary” we performed DoS and Re-
play attacks on the experimental wireless network and obtained several readings.
We have used the outcome of the attacks to discuss the anomaly detection process
in the following sections.

4.4.1 Timing Anomaly Detection

Table 4.7: Messages Exchanged During EAP-TLS Authentication

Table 4.7 lists the raw wireless network traces obtained from our experimental
setup. In these traces STA1 represents the MAC address 00:11:95:c3:15:5e of the
Dlink card on the Linux box and AP1 represents the MAC address 00:14:f1:ad:da:80
of the CISCO access point. Since our aim is to investigate anomalies during the
RSN association process we concentrate only on the 802.11 management and data
frames which includes the open system association and IEEE 802.1x/EAP related
4.4. EXPERIMENTAL PLATFORM

The above traces relate to the RSN association process of the Linux box and the CISCO access point based on IEEE 802.11i security architecture.

<table>
<thead>
<tr>
<th>Message</th>
<th>Trial 1</th>
<th>Trial 2</th>
<th>Trial 3</th>
<th>Trial 4</th>
<th>Trial 5</th>
<th>Trial 6</th>
<th>Trial 7</th>
<th>Trial 8</th>
<th>Trial 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Authentication</td>
<td>0.97</td>
<td>0.43</td>
<td>0.41</td>
<td>0.41</td>
<td>0.61</td>
<td>0.45</td>
<td>0.37</td>
<td>0.35</td>
<td>0.36</td>
</tr>
<tr>
<td>Open Association Request</td>
<td>6.24</td>
<td>4.34</td>
<td>0.91</td>
<td>4.34</td>
<td>2.64</td>
<td>2.29</td>
<td>0.54</td>
<td>0.58</td>
<td>0.54</td>
</tr>
<tr>
<td>Open Association Response</td>
<td>0.06</td>
<td>0.06</td>
<td>0.08</td>
<td>0.11</td>
<td>0.04</td>
<td>2.63</td>
<td>1.11</td>
<td>1.18</td>
<td>1.08</td>
</tr>
<tr>
<td>EAP Request, PEAP</td>
<td>0.08</td>
<td>1.03</td>
<td>1.03</td>
<td>4.31</td>
<td>1.03</td>
<td>0.10</td>
<td>0.79</td>
<td>0.78</td>
<td>0.79</td>
</tr>
<tr>
<td>EAP Request, EAP-TLS</td>
<td>6.64</td>
<td>4.88</td>
<td>0.09</td>
<td>4.44</td>
<td>10.36</td>
<td>9.20</td>
<td>5.89</td>
<td>4.66</td>
<td>4.62</td>
</tr>
<tr>
<td>TLS Client Hello</td>
<td>3.84</td>
<td>9.39</td>
<td>1.71</td>
<td>3.75</td>
<td>1.10</td>
<td>0.08</td>
<td>0.85</td>
<td>0.89</td>
<td>0.84</td>
</tr>
<tr>
<td>TLS Server Hello</td>
<td>0.08</td>
<td>8.65</td>
<td>0.08</td>
<td>3.64</td>
<td>10.71</td>
<td>7.25</td>
<td>3.09</td>
<td>2.91</td>
<td>2.92</td>
</tr>
<tr>
<td>TLS Change Cipher</td>
<td>1.49</td>
<td>1.30</td>
<td>4.16</td>
<td>1.49</td>
<td>0.08</td>
<td>1.48</td>
<td>1.22</td>
<td>1.32</td>
<td>1.49</td>
</tr>
<tr>
<td>EAP Request, EAP-TLS</td>
<td>7.70</td>
<td>15.63</td>
<td>10.06</td>
<td>9.34</td>
<td>8.41</td>
<td>26.36</td>
<td>8.94</td>
<td>9.42</td>
<td>9.10</td>
</tr>
<tr>
<td>Overall</td>
<td>216.15</td>
<td>189.56</td>
<td>232.05</td>
<td>219.42</td>
<td>194.76</td>
<td>244.02</td>
<td>240.01</td>
<td>214.1</td>
<td>225.40</td>
</tr>
</tbody>
</table>

Table 4.8: EAP-TLS Authentication RTTs (ms)

First step in our analysis is to check for timing anomalies, hence we need to obtain the round trip timings of every event during the RSN association process. Table 4.8 gives the round trip timings obtained during the various request/response events between the station and the access point. We have presented individual timings and the overall timings for nine different RSN associations under normal conditions. The timings listed between “EAP Success” and “EAP Key 4” messages includes the four 4-way key handshake messages. From the timing values listed, although the individual timings seems to vary drastically during some events the overall timings are similar in most cases. Hence, maintaining timing profiles for every participating wireless host and access point duo can help finding abnormal timings during the RSN association.

The software model for storing and calculating the timing values was built as a separate module. Basically the model takes the wireless traces as an input, processes it and stores them in a database table. Once the traces are stored in
the database in a predefined format, the model queries the database and takes each of the EAP type traces separately and processes them and stores the RTT values together with various statistical information in another database table. One separate result table is used for one EAP Type method, for example if we want EAP TLS timing profile table we can generate it dynamically after the traces were stored in the database. The following pseudo code describes the derivation of the various statistical values from the database tables.

\[
\text{\$result1 = query EAP specific Association Request Events from STA to AP}
\text{\$result2 = query EAP specific Association Response Events from AP to STA}
\text{\$rttsum = 0}
\text{\$rttcount = 0}
\]

\[
\text{while more rows in \$result1}
\text{fetch the next row in \$result1 into \$row1}
\text{fetch the next row in \$result2 into \$row2}
\text{while \$row1 timing value > \$row2 timing value}
\text{fetch the next row in \$result2 into \$row2}
\text{end while}
\text{calculate current RTT value}
\text{store current RTT value if it is the Maximum}
\text{store current RTT value if it is the Minimum}
\text{add current RTT value to \$rttsum}
\text{add one to \$rttcount}
\text{end while}
\text{calculate Average RTT}
\text{calculate Standard Deviation for RTT}
\]

The output RTT values table is defined based on each of the events associated with the respective EAP type. The timing profiles in these tables are the timing values to complete a particular round trip event during each of the EAP type specific authentication process. In this context a round trip event is considered to be the completion of two messages. The Request sent by the AP or STA and corresponding response received by the AP or STA. In order to present the values in a more functional manner the Mean, Maximum, Minimum, Standard deviation
4.4. EXPERIMENTAL PLATFORM

and the number of request and responses are calculated dynamically and stored. The Mean, Maximum and Minimum timing values together with the standard deviation gives the range of timings allowed for the round trip events during the normal operations.

<table>
<thead>
<tr>
<th>Event</th>
<th>Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Auth</td>
<td>0.97</td>
<td>0.35</td>
</tr>
<tr>
<td>Open Assn</td>
<td>6.24</td>
<td>0.54</td>
</tr>
<tr>
<td>EAP Ident</td>
<td>2.63</td>
<td>0.04</td>
</tr>
<tr>
<td>EAP PEAP</td>
<td>4.31</td>
<td>0.08</td>
</tr>
<tr>
<td>TLS Client Hello</td>
<td>10.36</td>
<td>0.09</td>
</tr>
<tr>
<td>EAP TLS</td>
<td>9.39</td>
<td>0.08</td>
</tr>
<tr>
<td>TLS Server Hello</td>
<td>28.87</td>
<td>27.02</td>
</tr>
<tr>
<td>TLS Certificate</td>
<td>10.71</td>
<td>0.08</td>
</tr>
<tr>
<td>TLS Change Cipher</td>
<td>4.16</td>
<td>0.08</td>
</tr>
<tr>
<td>EAP Key</td>
<td>26.36</td>
<td>7.70</td>
</tr>
<tr>
<td>Overall</td>
<td>244.62</td>
<td>189.56</td>
</tr>
</tbody>
</table>

Table 4.9: EAP-TLS Timing Profile

Table 4.9 show the timing profile between the experimental access point and the station. The timing profile gives the maximum and minimum timings required to complete a particular event during EAP-TLS authentication process. In this context an event is considered to be the completion of two messages; a request and the corresponding response. In this profile the maximum timing values give us an indication of how long a particular event should take during normal operation. Similarly the minimum values also show the least possible timing necessary to complete that event. Maintaining such profiles for every participating hosts could reveal any timing anomalies that could arise due to a network threat.

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Auth</td>
<td>0.36</td>
</tr>
<tr>
<td>Open Assn</td>
<td>0.55</td>
</tr>
<tr>
<td>EAP Ident</td>
<td>1.12</td>
</tr>
<tr>
<td>EAP PEAP</td>
<td>0.80</td>
</tr>
<tr>
<td>TLS Client Hello</td>
<td>101.14</td>
</tr>
<tr>
<td>EAP TLS</td>
<td>283.09</td>
</tr>
<tr>
<td>TLS Server Hello</td>
<td>665.78</td>
</tr>
<tr>
<td>TLS Certificate</td>
<td>259.81</td>
</tr>
<tr>
<td>TLS Change Cipher</td>
<td>1.66</td>
</tr>
<tr>
<td>EAP Key</td>
<td>8.71</td>
</tr>
<tr>
<td>Overall</td>
<td>1424.35</td>
</tr>
</tbody>
</table>

Table 4.10: EAP-TLS Abnormal Timing
The timings shown in Table 4.10 above were obtained during a denial of service attack. Here neither the access point nor the station was targeted. However we injected some Deauthentication frames with a bogus MAC address jamming the media. During this period we initiated an authentication process and the timings were noted. From the timings it is obvious that the values shown for the challenge/exchange process is extra-ordinarily high resulting in the high overall timing. Thus we could detect that the hosts are experiencing an abnormal condition. Although the DoS attack did not affect the authentication process itself the timing anomaly detection module will show only a time delay in the authentication process. To further investigate the legitimacy of this anomaly we pass this information to the third phase - the intrusion prevention module.

4.4.2 Behavioral Anomaly Detection

The next module in our proposed EWS is the behavioral analysis of the participating hosts. As discussed in Section 3.6 events shown in Figure 3.2 represents the normal behavior of the station and the access point during a normal association process. However, when there are network threats directed towards the wireless network the behavior may change as shown in Figures 3.5, 3.6, and 3.7. In view of studying the behavioral anomalies we performed a replay attack on a wireless host. Table 4.11 shows the captured packets during the replay attack.

It is evident from the traces shown in Table 4.11 this wireless host does not match the normal behavior as in Figure 3.2. During normal behavior a wireless host cannot traverse from the AUTHENTICATING state to the open AUTH/ASSN state. The only possible state from AUTHENTICATING state is the DISCONNECTED state. Figure 4.5 shows the projection model for this abnormal behavior. Hence this example shows how we could detect behavioral anomalies using behavioral profiles. Thus maintaining behavioral profiles for every participating host will enable us to closely monitor each host and detect anomalies due to various threats.
Table 4.11: A Replay Attack

Table 4.12 shows packets captured from a replay attack performed by the same station, i.e. the connecting station and the attacking station are the same. Hence the 4-way handshake takes place straightaway since the station is already in possession of the PMK. Although this type of attack is not of interest (from the same station), it is a possible behavior if an adversary manages to get possession of the PMK.

In the above examples, although the wireless host is showing abnormal behavior it is actually not at risk. In the first example the host disconnects itself and re-associates with the access point and the second case is similar to a roaming situation. This was possible due to the strong authentication mechanisms used in the EAP-TLS association process. However, in our EWS we still pass this information to the third phase to verify the legitimacy of the scenario.
4.4.3 Outlier Detection

Having detected timing or behavioral anomalies, it is vital to verify the legitimacy of such anomalies. The major role played in this phase is to validate the anomalies detected by the previous modules. In this context we present our initial results obtained by processing a data cube built from the wireless traces captured during the scenarios discussed earlier. These wireless traces were obtained over a period of one week in the Griffith University WLAN environment using Ethereal [37]. The total number of packets collected exceeds one million with a disk size of 1.7GB approximately. These traces were then converted with suitable mappings
4.4. EXPERIMENTAL PLATFORM

to match the requirements of the data cube algorithms [26].

<table>
<thead>
<tr>
<th>Reference</th>
<th>Attribute</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Time</td>
<td>Dimension</td>
</tr>
<tr>
<td>B</td>
<td>Source ID</td>
<td>Dimension</td>
</tr>
<tr>
<td>C</td>
<td>Destination ID</td>
<td>Dimension</td>
</tr>
<tr>
<td>D</td>
<td>Event ID</td>
<td>Dimension</td>
</tr>
<tr>
<td>E</td>
<td>Protocol ID</td>
<td>Dimension</td>
</tr>
<tr>
<td></td>
<td>Count</td>
<td>Measure</td>
</tr>
<tr>
<td></td>
<td>GOS</td>
<td>Measure</td>
</tr>
</tbody>
</table>

Table 4.13: Test Data Cube Attributes

Once the captured data was converted using the parsing program we built the data cube with the five attributes shown in Table 4.13. The “Count” attribute listed in the table is considered as a measure value. Details of the parsing program are given in Appendix A. The time taken to build the data cube, as explained in Section 4.3.1 was 28.32 seconds. Indexing the data cube took a further 242.5 seconds. These timing values are presented here to give an idea about the time required to build and index the data cube. With a dimension of five we get 32 different views of the data as shown in Figure 4.2. However, we need not process all 32 views to answer most of our queries. For example, consider the following data associations needed for analyzing network anomalies between stations and access points.

1. Source ID, Destination ID, Event ID
2. Source ID, Event ID, Protocol ID
3. Destination ID, Event ID, Protocol ID
4. Source ID, Destination ID
5. Source ID, Event ID
6. Destination ID, Event ID

In order to process these queries we require only BCD, BDE, CDE, BC, BD and CD views. However, the other views will be useful to get more information
Table 4.14: Query Results to Validate Timing Anomalies Under DoS Attack

about an anomaly from different perspectives. Query results shown in Table 4.14 were obtained using view BD (Source ID, Event ID). The time taken to execute this query was 2.85ms in our experimental system. The query result shows, for two different stations, the associated events and the number of occurrences of such events as “Count” value. Here Source ID 12 refers to station STA2 and Source ID 14 refers to station STA1. Event IDs ranging from 0 to 361 refers to the various events, such as 0 representing ProbeRequest, 27 representing Authentication, 42 representing Deauthentication and so on (a mapping of Events vs EventID could be found in Appendix A). Hence from this table it is evident that station STA2 (12) has an unacceptable number of Deauthentication events compared to that of station STA1 (14). To validate this results we then queried the data cube on view BC (Source ID, Dest. ID) for the total number of associations for both stations. The query results are: STA2 (12) had 128928 associations and STA1 (14) had 12019 associations. Although these query results exhibit some abnormal behavior this information alone is not sufficient to classify the behavior as illegitimate. Hence, having obtained the query results we then use our substantiation techniques to establish whether the anomalies are legitimate or not. We adjourn the application of our substantiation techniques until Chapter 6.

Next, we executed queries to verify the behavioral anomalies observed during the replay attack. For this purpose we queried the data cube using views BCD and BC. Here we queried for events necessary to identify the effect of the replay attack.
Hence, we used only those events related to the authentication process. In a IEEE 802.11i/WPA2 environment a station requests to associate with an access point giving its credentials. The access point will respond favorably if it agrees with the security credentials advertised by the station. Therefore, we obtained query results for AssociationRequest, AssociationResponse and EAPSuccess events. Table 4.15 shows the results obtained from these queries. The first block of two rows show the number of Association Request events for STA2 (12) and STA1 (14) respectively. The second block shows the number of Association Response events for STA2 (12) and STA1 (14) respectively. The third block give the number of EAP Success events for the two stations. These results indicate some abnormality with STA2 (12). Station STA1 (14) has almost equal number of EAP Success events corresponding to the number of Association Response messages sent by the Access Point (0). Whereas, with STA2 (12), for 974 Association Response messages only 64 EAP Success messages are recorded. Therefore, with these count values we can use our substantiation techniques to establish whether STA2 (12) is facing any behavioral anomaly.

The above examples show the potential of our proposal for using outlier based data association techniques to substantiate the legitimacy of anomalies. However, the effectiveness of this method will strongly depend on the ability to tune the timing and behavior profiles accurately. In these examples we have shown the count values from one particular view. Later in Chapter 6, we have discussed the use of different data views to establish a more reliable end result to scale the vulnerability of a misbehaving host. Although the data cube results discussed thus far are obtained in an off-line mode, in the following chapters we provide real-time

<table>
<thead>
<tr>
<th>Source ID</th>
<th>Destination ID</th>
<th>Event ID</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>0</td>
<td>76</td>
<td>11891</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>76</td>
<td>664</td>
</tr>
<tr>
<td>0</td>
<td>12</td>
<td>95</td>
<td>974</td>
</tr>
<tr>
<td>0</td>
<td>14</td>
<td>95</td>
<td>276</td>
</tr>
<tr>
<td>0</td>
<td>12</td>
<td>323</td>
<td>64</td>
</tr>
<tr>
<td>0</td>
<td>14</td>
<td>323</td>
<td>282</td>
</tr>
</tbody>
</table>

Table 4.15: Query Results to Validate Replay Attack
results that could be extended for any type and size of networks in online mode. We have tested the capabilities of the data cube algorithms rigorously [103].

4.5 Conclusion

In this chapter we have discussed our proposed EWS in detail. The EWS consists of a Packet Capturing module, Event Engine, Timing Anomaly module, Behavior Anomaly module, intrusion prevention module and the Data mining engine. Each of these modules do their respective tasks in order to achieve the final goal of intrusion detection/prevention.

As discussed in subsection 4.4.3, out of all modules the intrusion prevention module does the important task of substantiating all detected anomalies. As a part of substantiation, we have first discussed the notion of data cube views and how the various views are related to the security threats in the wireless environment. Then, we have derived the mathematical expressions for GCL and GOS, which are used for substantiation. The GCL is used to estimate the group cohesiveness of a group of associated events and the GOS is used to find the level of association of individual events within the group of associated events.

The data cube described in this chapter, holds all wireless traces captured and parsed by the event engine. On these traces we execute one or more queries and use Equation (4.7) to determine the Group Confidence Level or Equation (4.6) to find the Group Outlier Score. The actual use of these values in substantiating anomalies is presented in Chapter 6.

Further, as a precursor, we have discussed the notion of timing and behavior anomalies with some examples. We have also described the idea of outlier detection with some simple query results obtained from the data cube.
Chapter 5

Anomaly Detection

5.1 Introduction

Since, we were interested in studying the behavior of wireless stations configured for EAP-LEAP, EAP-PEAP and EAP-TLS authentication mechanisms, firstly, we had to configure the wireless stations to associate with the experimental access points on these different authentication schemes. We used both Linux and Windows XP boxes for our experiments. As shown in our experimental setup in Figure 4.4, Station STA1 was configured for EAP-LEAP authentication, STA2 was configured for EAP-PEAP authentication and STA3 for EAP-TLS authentication. Details of how we configured the Linux boxes for these experiments are discussed first in this chapter. Further, the wireless node “Monitor” in the diagram is also part of the data processing unit that is used to capture wireless network traffic in promiscuous mode. The node “Adversary” is capable of introducing various anomalies into the wireless network.

In order to validate our proposed EWS we collected and analyzed large amounts of wireless network traces in several steps. Firstly, we conducted the timing studies. Timing analysis were conducted on all three types of authentication mechanisms, EAP-LEAP, EAP-PEAP and EAP-TLS. We conducted over fifteen experiments collecting both normal and abnormal data for timing analysis. Details of
the analysis are discussed in this chapter.

The next phase of defense in our EWS is the detection of behavior anomalies. In order to study the behavior of the wireless stations during the RSNA we first modeled the RSNA process using Behavior Trees. Next, we established the formal model and verified its correctness during both normal and abnormal behavior [108]. Once we were satisfied with the formal analysis we used this experience in practice to analyze the behavior of wireless hosts in the real wireless environment. We captured and analyzed large number of wireless traces for behavioral anomalies. When the experiments were carried out the test setup was exposed to the normal wireless environment of the Griffith University, attracting traffic from various unspecified sources. Details of these experiments are discussed in the succeeding sections.

Once the timing and behavior experiments were complete, we commenced experimenting with the intrusion prevention and/or detection module. In order to do this, the wireless traces captured by the packet capture module (details of the packet capturing program is presented in Appendix A) was parsed (details of the parsing program is given in Appendix B) and prepared for storage into the data cube. Thereafter, the data mining engine executes a number of queries to substantiate the anomalies raised by the timing and/or behavior module. In this chapter we mainly discuss about the timing and behavior detection process and leave the details of the substantiation mechanism together with the analysis for the next chapter.

5.1.1 Chapter Overview

Details of our experimental work is described in this chapter as follows. In Section 5.2, we first describe the EAP authentication methods LEAP, PEAP and TLS. In Section 5.3 we describe the details of timing experiment in the wireless environment. Here we discuss the effect of timings in the three EAP methods. Thereafter, in Section 5.4 we discuss about the behavioral analysis. Here again
we compare and contrast anomalies in the three EAP methods.

5.2 EAP Authentication

As mentioned in Section 2.3.2, EAP is a universal authentication framework frequently used in wireless networks and Point-to-Point connections. Although the EAP protocol is not limited to wireless LANs and can be used for wired LAN authentication, it is most often used in wireless LANs. The WPA and WPA2 standards have officially adopted five EAP types as its official authentication mechanisms.

EAP is an authentication framework, not a specific authentication mechanism. The EAP provides some common functions and a negotiation of the desired authentication mechanism. Such mechanisms are called EAP methods and there are currently about 40 different methods. Commonly used modern methods capable of operating in wireless networks include EAP-TLS, EAP-SIM, EAP-AKA, PEAP, LEAP and EAP-TTLS. However, in our experiments we use EAP-TLS, PEAP and LEAP to verify the credibility of our proposed detection mechanisms.

5.2.1 EAP LEAP

LEAP uses 802.1x EAPOL messages [62], performs server authentication, achieves username/password (over MS-CHAP) as the user authentication mechanism, uses a RADIUS server as the authentication server, and provides mechanisms for deriving and distributing encryption keys. The entities that participate in a LEAP exchange are the RADIUS server, the AP, and the Station.

Initially, the client and the RADIUS server should have the shared secret, usually a username/password database of all users in the RADIUS server, and each client should have its own username and password. After a client establishes connectivity, it initiates the authentication process by an EAPOL-start message


Figure 5.1: EAP-LEAP Authentication

as shown in Figure 5.1 by message 3, to which the AP responds by an EAP-request-identity message over EAPOL (message 4 in figure). The client response with identity is sent to the RADIUS server in a RADIUS message. From this point on, the AP acts as a relay between the client and the RADIUS server.

Next, the client authentication begins with challenge-response mechanism. The server sends a challenge, to which the client responds with a hash calculated using the password and the LEAP algorithm (message exchange 6). The server also calculates the hash, and if they are equal, the authentication is success. As you can see, the client authentication happens based on existing infrastructure and still not transmitting the credential (here the password).

Thereafter, the server authentication happens through a similar mechanism, and at the end, the server sends the encryption keys to the AP (message exchange 7). The AP distributes the required key material by broadcast. The client derives the encryption key from the key materials, and from then on, the AP and the client can use the encryption keys to have a secure conversation.

In our experiments we used a Linux box configured to authenticate in the above manner and a number of tests were carried out to explore the anomalies
created by various abnormal conditions. Table 5.1 shows the configuration used on the Linux box to connect to the RADIUS server.

### 5.2.2 EAP PEAP

PEAP uses TLS to create an encrypted channel between an authenticating PEAP client, and a PEAP authenticator, such as the RADIUS server. PEAP does not specify an authentication method, but provides additional security for
other EAP authentication protocols, such as EAP-MSCHAPv2, that can operate through the TLS encrypted channel provided by PEAP. There are two stages in the PEAP authentication process between PEAP client and authenticator. As shown in Figure 5.2, the first stage sets up a secure channel between the PEAP client and the authenticating server. The second stage provides EAP authentication between the EAP client and authenticator.

Here again a Linux box was configured to perform EAP-PEAP authentication and we conducted several experiments to study the anomalies created by various abnormal conditions. Table 5.2 shows the configuration used on the Linux box to connect to the RADIUS server.

\begin{table}[h]
\centering
\begin{tabular}{|l|}
\hline
ssid="kandy"  
proto=RSN  
key_mgmt=WPA-EAP  
pairwise=CCMP TKIP  
group=TKIP  
eap=PEAP  
identity="Sith@wireless.net"  
ca_cert="/etc/1x/r/root.pem"  
priority=1  
\hline
\end{tabular}
\caption{wpeap.conf}
\end{table}

\section{5.2.3 EAP TLS}

As previously mentioned, EAP-TLS authentication is based on 802.1x/EAP architecture \cite{1}. Components involved in the 802.1x/EAP authentication process are: supplicant (the end entity, or end user’s machine), the authenticator (the access point), and the authentication server (back-end RADIUS server). The supplicant and the RADIUS server must support EAP-TLS authentication. The access point has to support the 802.1x/EAP authentication process. (The access point is not aware of the EAP authentication protocol type.)

As illustrated in Figure 5.3, the RADIUS server provides its certificate to the client and requests the client’s certificate. The client validates the server certificate and responds with an EAP response message containing its certificate and also starts the negotiation for cryptographic specifications (cipher and compression
5.3 Timing Anomaly

In order to study the behavior of wireless hosts during different EAP type specific authentication process, we collected 802.11 management traces from different algorithms). After the client’s certificate is validated, the server responds with cryptographic specifications for the session.

```plaintext
ssid="kandy"
proto=RSN
key_mgmt=WPA-EAP
pairwise=CCMP TKIP
group=TKIP
eap=TLS
identity="Sith@wireless.net"
ca_cert="/etc/1x/r/root.pem"
client_cert="/etc/1x/r/cert-clt.pem"
private_key="/etc/1x/r/cert-clt.pem"
private_key_password="whatever"
priority=1
```

Table 5.3: wtls.conf

Table 5.3 shows the configuration used to connect to the RADIUS server.

**5.3 Timing Anomaly**

In order to study the behavior of wireless hosts during different EAP type specific authentication process, we collected 802.11 management traces from different
wireless hosts configured to authenticate using LEAP, PEAP and TLS authentication mechanisms. In this case the same access point was used with the three different stations ST1, STA2 and STA3. Firstly, we present the measured RTT values of EAP LEAP, PEAP and TLS authenticated hosts during the RSN association process, under normal operations. Thereafter, we present the RTT values of these hosts under abnormal conditions.

### 5.3.1 Normal Timings

<table>
<thead>
<tr>
<th>Event</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
<th>Mean (ms)</th>
<th>StDev</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPEN AUTHENTICATION</td>
<td>0.08</td>
<td>3.40</td>
<td>0.61</td>
<td>0.53</td>
<td>65</td>
</tr>
<tr>
<td>OPEN ASSOCIATION</td>
<td>0.51</td>
<td>2.65</td>
<td>0.65</td>
<td>0.29</td>
<td>63</td>
</tr>
<tr>
<td>EAP 1</td>
<td>0.05</td>
<td>2.57</td>
<td>0.79</td>
<td>0.43</td>
<td>65</td>
</tr>
<tr>
<td>EAP 2 (PEAP)</td>
<td>0.03</td>
<td>1.55</td>
<td>0.35</td>
<td>0.21</td>
<td>63</td>
</tr>
<tr>
<td>EAP 3 (LEAP)</td>
<td>0.42</td>
<td>1.30</td>
<td>0.48</td>
<td>0.13</td>
<td>65</td>
</tr>
<tr>
<td>EAP 4 (LEAP)</td>
<td>3.05</td>
<td>9.77</td>
<td>3.32</td>
<td>0.89</td>
<td>63</td>
</tr>
<tr>
<td>EAP Key Exchange</td>
<td>0.07</td>
<td>11.36</td>
<td>2.54</td>
<td>2.16</td>
<td>126</td>
</tr>
<tr>
<td>Overall</td>
<td>12.81</td>
<td>321.97</td>
<td>18.96</td>
<td>37.06</td>
<td>69</td>
</tr>
</tbody>
</table>

Table 5.4: EAP-LEAP Timing Profile

Table 5.4 shows the EAP-LEAP RTT statistics between AP1 and station STA1 obtained from a trace file with more than sixty EAP-LEAP RSN association trials, under normal conditions. The trace file was created by our packet capturing module over a period of one day. The RTT values were obtained using the “Packet Time” values listed in the sample traces shown in Appendix C.1. The script used to “Deauthenticate” the wireless stations in order to make them re-associate is listed in Appendix D.1. The timing profile shows the typical timing range required to complete a particular round trip event (time difference between a request and the corresponding response) during normal EAP-LEAP authentication process. In this case, there are a total of seven EAP-LEAP request/response messages. For each trial, the timing measurements for the seven messages and the overall timing were recorded (similar to the data in Table 4.8). Then the minimum, maximum, mean and standard deviation values were calculated. This gives us an indication of the possible range of RTT values during the normal operations.
5.3. TIMING ANOMALY

<table>
<thead>
<tr>
<th>Event</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
<th>Mean (ms)</th>
<th>StDev</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPEN AUTHENTICATION</td>
<td>0.37</td>
<td>0.61</td>
<td>0.44</td>
<td>0.05</td>
<td>62</td>
</tr>
<tr>
<td>OPEN ASSOCIATION</td>
<td>0.55</td>
<td>1.41</td>
<td>0.61</td>
<td>0.10</td>
<td>64</td>
</tr>
<tr>
<td>EAP 1</td>
<td>0.10</td>
<td>2.03</td>
<td>0.57</td>
<td>0.46</td>
<td>69</td>
</tr>
<tr>
<td>EAP 2 (PEAP)</td>
<td>0.23</td>
<td>4.24</td>
<td>2.05</td>
<td>0.80</td>
<td>69</td>
</tr>
<tr>
<td>EAP 3 (PEAP)</td>
<td>0.08</td>
<td>1.76</td>
<td>0.40</td>
<td>0.21</td>
<td>68</td>
</tr>
<tr>
<td>EAP 4 (PEAP)</td>
<td>0.02</td>
<td>1.49</td>
<td>0.83</td>
<td>0.41</td>
<td>63</td>
</tr>
<tr>
<td>EAP 5 (PEAP)</td>
<td>0.58</td>
<td>1.08</td>
<td>0.62</td>
<td>0.06</td>
<td>65</td>
</tr>
<tr>
<td>EAP 6 (PEAP)</td>
<td>0.36</td>
<td>0.59</td>
<td>0.45</td>
<td>0.03</td>
<td>65</td>
</tr>
<tr>
<td>EAP 7 (PEAP)</td>
<td>0.72</td>
<td>4.29</td>
<td>2.70</td>
<td>0.51</td>
<td>62</td>
</tr>
<tr>
<td>EAP 8 (PEAP)</td>
<td>0.08</td>
<td>2.50</td>
<td>0.60</td>
<td>0.44</td>
<td>70</td>
</tr>
<tr>
<td>EAP 9 (PEAP)</td>
<td>0.18</td>
<td>1.65</td>
<td>0.50</td>
<td>0.28</td>
<td>67</td>
</tr>
<tr>
<td>EAP Key Exchange</td>
<td>0.05</td>
<td>7.08</td>
<td>1.62</td>
<td>1.22</td>
<td>139</td>
</tr>
<tr>
<td>Overall</td>
<td>48.65</td>
<td>68.09</td>
<td>52.25</td>
<td>3.78</td>
<td>63</td>
</tr>
</tbody>
</table>

Table 5.5: EAP-PEAP Timing Profile

Table 5.5 shows the EAP-PEAP RTT statistics between AP1 and station STA2, obtained from a trace file consisting of more than sixty EAP-PEAP RSN association trials, under normal conditions. The timing profile shows the typical timing range required to complete a particular round trip event during normal EAP-PEAP authentication process. Unlike for EAP-LEAP authentication, EAP-PEAP involves a total of twelve EAP-PEAP request/response messages.

<table>
<thead>
<tr>
<th>Event</th>
<th>Min (ms)</th>
<th>Max (ms)</th>
<th>Mean (ms)</th>
<th>StDev</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPEN AUTHENTICATION</td>
<td>0.11</td>
<td>9.07</td>
<td>0.85</td>
<td>0.85</td>
<td>59</td>
</tr>
<tr>
<td>OPEN ASSOCIATION</td>
<td>36.25</td>
<td>106.13</td>
<td>53.91</td>
<td>8.24</td>
<td>64</td>
</tr>
<tr>
<td>EAP 1</td>
<td>10.89</td>
<td>89.74</td>
<td>25.69</td>
<td>11.11</td>
<td>69</td>
</tr>
<tr>
<td>EAP 2 (PEAP)</td>
<td>0.71</td>
<td>17.72</td>
<td>4.01</td>
<td>4.41</td>
<td>66</td>
</tr>
<tr>
<td>EAP 3 (TLS)</td>
<td>6.56</td>
<td>56.55</td>
<td>43.89</td>
<td>10.98</td>
<td>71</td>
</tr>
<tr>
<td>EAP 4 (TLS)</td>
<td>0.08</td>
<td>43.39</td>
<td>4.63</td>
<td>9.04</td>
<td>72</td>
</tr>
<tr>
<td>EAP 5 (TLS)</td>
<td>13.66</td>
<td>71.68</td>
<td>17.83</td>
<td>8.16</td>
<td>71</td>
</tr>
<tr>
<td>EAP 6 (TLS)</td>
<td>0.738</td>
<td>39.41</td>
<td>4.36</td>
<td>5.01</td>
<td>65</td>
</tr>
<tr>
<td>EAP Key Exchange</td>
<td>1.74</td>
<td>76.21</td>
<td>22.67</td>
<td>20.27</td>
<td>124</td>
</tr>
<tr>
<td>Overall</td>
<td>134.45</td>
<td>244.30</td>
<td>162.74</td>
<td>17.41</td>
<td>61</td>
</tr>
</tbody>
</table>

Table 5.6: EAP-TLS Timing Profile

Table 5.6 shows the EAP-TLS RTT statistics between AP1 and station STA3, obtained from a trace file consisting of more than sixty EAP-TLS RSN association trials, under normal conditions. The timing profile shows the typical timing range required to complete a particular round trip event during normal EAP-TLS authentication process. In this context a round trip event is considered to be the completion of two messages; a request and the corresponding response.
By approximating the RTT values to a standard normal distribution, we can specify an upper and a lower limit for RTT values for a given confidence level. Hence, in order to give a comprehensive coverage of timing anomalies by our EWS, we consider, as a typical measure, one standard deviation (68%) above and below the mean as the norm for each EAP event [114]. For example, in the case of EAP-TLS authentication, using the mean and standard deviation values listed in Table 5.6, the lower limit of the overall RTT value would be 145.33ms (162.74 - 17.41) and the upper limit would be 180.15ms (162.74 + 17.41). We can use these values as typical for timing anomaly detection.

5.3.2 Abnormal Timings

Above discussed RTT profiles represent for normal behavior of EAP authenticated hosts. However when anomalies occur this behavior can change. There can be situations where the numbers of events are extraordinarily high or low. There can also be situations where events can totally disappear from the respective range of the monitoring devices. Furthermore, abnormalities can be due to various wireless security attacks, such as, DoS, Man-in-the-Middle attack, Replay Attacks etc. Other reasons may be environmental disturbances. Since we are using IEEE 802.11 WLAN which operates on 2.4 GHz frequency band, it has higher probability of attracting noise and interferences. WLAN characteristics can vary widely, depending on device characteristics (e.g. antenna design and orientation) and environmental factors (e.g. floor plan). Following are some observations obtained during the abnormal conditions.

Table 5.7 is an example of abnormal RTT values. These values are from a single experiment carried out by injecting abnormal management frames with EAP-LEAP authenticated hosts. Sample abnormal traces are listed in Appendix C.2. The script used to introduce the abnormal condition is listed in Appendix D.2. According to the EAP-LEAP RTT values shown in the above table, almost
5.3. TIMING ANOMALY

<table>
<thead>
<tr>
<th>Event</th>
<th>Abnormal Time (ms)</th>
<th>Normal Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPEN AUTHENTICATION</td>
<td>6.22</td>
<td>0.61</td>
</tr>
<tr>
<td>OPEN ASSOCIATION</td>
<td>4.28</td>
<td>0.65</td>
</tr>
<tr>
<td>EAP 1</td>
<td>6.11</td>
<td>0.79</td>
</tr>
<tr>
<td>EAP 2 (PEAP)</td>
<td>4.15</td>
<td>0.35</td>
</tr>
<tr>
<td>EAP 3 (LEAP)</td>
<td>3.27</td>
<td>0.48</td>
</tr>
<tr>
<td>EAP 4 (LEAP)</td>
<td>12.92</td>
<td>3.32</td>
</tr>
<tr>
<td>EAP Key Exchange</td>
<td>16.19</td>
<td>2.54</td>
</tr>
<tr>
<td>Overall</td>
<td>996.44</td>
<td>18.96</td>
</tr>
</tbody>
</table>

Table 5.7: Abnormal and Mean EAP-LEAP Timings

![Figure 5.4: EAP-LEAP Timings](image)

Every message exchange show significantly high RTT values resulting in high overall timing of 996.44ms, compared with the normal values in Table 5.4. Also it can be seen that the timing values of every message exchange are highly unpredictable. Furthermore, these timing values are abnormal since they do not fall within one standard deviation from mean. Figure 5.4 shows the mean and the upper limit with one standard deviation from mean values of EAP-LEAP RTT timings together with abnormal RTT values (overall time is not shown). In this discussion we have not considered repeated trials, since our aim was to demonstrate the abnormality and show the timing values that could be expected under such abnormal conditions.
Table 5.8: Abnormal and Mean EAP-PEAP Timings

<table>
<thead>
<tr>
<th>Event</th>
<th>Abnormal Time (ms)</th>
<th>Normal Mean (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPEN AUTHENTICATION</td>
<td>8.26</td>
<td>0.44</td>
</tr>
<tr>
<td>OPEN ASSOCIATION</td>
<td>7.10</td>
<td>0.61</td>
</tr>
<tr>
<td>EAP 1</td>
<td>8.85</td>
<td>0.57</td>
</tr>
<tr>
<td>EAP 2 (PEAP)</td>
<td>6.07</td>
<td>2.05</td>
</tr>
<tr>
<td>EAP 3 (PEAP)</td>
<td>7.11</td>
<td>0.40</td>
</tr>
<tr>
<td>EAP 4 (PEAP)</td>
<td>12.47</td>
<td>0.83</td>
</tr>
<tr>
<td>EAP 5 (PEAP)</td>
<td>8.45</td>
<td>0.62</td>
</tr>
<tr>
<td>EAP 6 (PEAP)</td>
<td>11.99</td>
<td>0.45</td>
</tr>
<tr>
<td>EAP 7 (PEAP)</td>
<td>5.87</td>
<td>2.70</td>
</tr>
<tr>
<td>EAP 8 (PEAP)</td>
<td>9.64</td>
<td>0.60</td>
</tr>
<tr>
<td>EAP 9 (PEAP)</td>
<td>7.91</td>
<td>0.50</td>
</tr>
<tr>
<td>EAP Key Exchange</td>
<td>6.68</td>
<td>1.62</td>
</tr>
<tr>
<td>Overall</td>
<td>628.40</td>
<td>52.25</td>
</tr>
</tbody>
</table>

Table 5.8 shows a set of RTT values for an abnormal EAP-PEAP authenticated host. These values are also from a single experiment carried out during an EAP-PEAP authentication process by injecting abnormal management frames. Comparing these values with that of the normal timing values listed in Table 5.5, it is evident that the abnormal timing values are significantly high. Further, as discussed earlier, most of the RTT values do not fall within one standard deviation from mean. Also, the first three RTT values are considerably high from the mean compared to the others. This is due to the unpredictable nature of the Open System association process, where hosts can be made to repeatedly deauthenticate and re-associate. The $z$-score (the number of standard deviations away from the mean) for the overall timing value is 150 in this case. Figure 5.5 shows the mean and one standard deviation from mean values of EAP-PEAP RTT timings together with abnormal RTT values. Again the overall timing is not shown in the figure.

The RTT values shown in Table 5.9 are for an abnormal EAP-TLS association. The values can be considered as abnormal since the timing values for the EAP-TLS Request/Response messages are significantly high resulting in a very high overall timing value. These RTT values also do not fall within one standard deviation from mean. Figure 5.6 shows the mean and one standard deviation from mean values of EAP-TLS RTT timings together with abnormal RTT values. As we can
5.3. TIMING ANOMALY

![Figure 5.5: EAP-PEAP Timings](image)

<table>
<thead>
<tr>
<th>Event</th>
<th>Abnormal Time (ms)</th>
<th>Normal Mean (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPEN AUTHENTICATION</td>
<td>3.07</td>
<td>0.56</td>
</tr>
<tr>
<td>OPEN ASSOCIATION</td>
<td>78.47</td>
<td>0.85</td>
</tr>
<tr>
<td>EAP 1</td>
<td>79.17</td>
<td>53.91</td>
</tr>
<tr>
<td>EAP 2</td>
<td>120.36</td>
<td>25.60</td>
</tr>
<tr>
<td>EAP 3 (PEAP)</td>
<td>16.11</td>
<td>4.01</td>
</tr>
<tr>
<td>EAP 4 (TLS)</td>
<td>89.24</td>
<td>43.89</td>
</tr>
<tr>
<td>EAP 5 (TLS)</td>
<td>25.26</td>
<td>4.63</td>
</tr>
<tr>
<td>EAP 6 (TLS)</td>
<td>33.21</td>
<td>17.83</td>
</tr>
<tr>
<td>EAP 7 (TLS)</td>
<td>44.75</td>
<td>4.36</td>
</tr>
<tr>
<td>EAP Key Exchange</td>
<td>35.69</td>
<td>22.67</td>
</tr>
<tr>
<td>Overall</td>
<td>1149.15</td>
<td>162.74</td>
</tr>
</tbody>
</table>

Table 5.9: Abnormal and Mean EAP-TLS Timings

see from this figure that except for the key exchange message (Event 10), no other message falls within the acceptable deviation range. In this case the z-score for the overall timing value is 56.

As seen in figures 5.4, 5.5 and 5.6 the mean RTT values for EAP-LEAP, PEAP, and TLS authenticated hosts are very much less than the abnormal RTT values. The three graphs can be used to visualize the significant differences between normal RTTs and abnormal RTTs for each event. Throughout our experiment we assumed that the calculated RTTs (using sample wireless traces) behave according to the normal distribution. Also monitoring RTT of a message exchange is a
passive detection mechanism to detect unusual behaviors of the wireless as well as wired networks. Hence this technique may be practically used to detect intruders who can adversely affect the wireless network environment.

Next, we injected some abnormal management frames forcing the wireless hosts to deauthenticate and reassociate frequently creating a Replay attack situation. This was done by injecting “Deauthentication” frames using the “airreplay” [4] tool as shown by the script in Appendix D.2.

<table>
<thead>
<tr>
<th>Event</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
</tr>
<tr>
<td>OPEN AUTHENTICATION</td>
<td>0.41</td>
</tr>
<tr>
<td>OPEN ASSOCIATION</td>
<td>0.16</td>
</tr>
<tr>
<td>EAP 1</td>
<td>5.65</td>
</tr>
<tr>
<td>EAP 2 (PEAP)</td>
<td>3.04</td>
</tr>
<tr>
<td>EAP 3 (LEAP)</td>
<td>2.29</td>
</tr>
<tr>
<td>EAP 4 (LEAP)</td>
<td>36.55</td>
</tr>
<tr>
<td>EAP Key Exchange</td>
<td>9.11</td>
</tr>
<tr>
<td>Overall</td>
<td>157.00</td>
</tr>
</tbody>
</table>

Table 5.10: EAP-LEAP Timings During DoS Attack

The EAP-LEAP timing values shown in Table 5.10 is from three different RSNA trials during a DoS attack. Here neither the access point nor the station was targeted. However we continuously injected “Deauthentication” frames with
fake MAC addresses jamming the medium. During this period we initiated a EAP-LEAP authentication process and the timings were noted. To make sure the timing anomaly module detects every possible anomaly, let us consider the RSNA with minimum timings. From the timing measurements in Table 5.10 it can be seen that almost every message exhibits significantly high timing values resulting in high overall timing. Although, some of these timing values fall within the $Max$ timings in the normal profile (Table 5.4), it will be considered abnormal since they do not fall within one standard deviation from the mean.

<table>
<thead>
<tr>
<th>Event</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
</tr>
<tr>
<td>OPEN AUTHENTICATION</td>
<td>0.63</td>
</tr>
<tr>
<td>OPEN ASSOCIATION</td>
<td>5.36</td>
</tr>
<tr>
<td>EAP 1</td>
<td>0.27</td>
</tr>
<tr>
<td>EAP 2 (PEAP)</td>
<td>7.44</td>
</tr>
<tr>
<td>EAP 3 (PEAP)</td>
<td>2.52</td>
</tr>
<tr>
<td>EAP 4 (PEAP)</td>
<td>5.30</td>
</tr>
<tr>
<td>EAP 5 (PEAP)</td>
<td>0.45</td>
</tr>
<tr>
<td>EAP 6 (PEAP)</td>
<td>0.20</td>
</tr>
<tr>
<td>EAP 7 (PEAP)</td>
<td>37.63</td>
</tr>
<tr>
<td>EAP 8 (PEAP)</td>
<td>0.47</td>
</tr>
<tr>
<td>EAP 9 (PEAP)</td>
<td>4.56</td>
</tr>
<tr>
<td>EAP Key Exchange</td>
<td>79.07</td>
</tr>
<tr>
<td>Overall</td>
<td>426.70</td>
</tr>
</tbody>
</table>

Table 5.11: EAP-PEAP Timings During DoS Attack

The timing values shown in Table 5.11 is from three separate EAP-PEAP RSNA trials during a DoS attack. As in the above case of EAP-LEAP, while we continuously injected “Deauthentication” frames with fake MAC addresses we initiated EAP-PEAP authentication and recorded the round trip times. Here, “EAP 7” and the “Key Exchange” messages show very high timing values (compared to the values for normal operation given in Table 5.5) resulting in very high overall timing.

The timing values shown in Table 5.12 is from three EAP-TLS RSNA trials during a DoS attack. Here again, like in the other cases we continuously injected “Deauthentication” frames with fake MAC addresses jamming the medium. During this period we initiated an EAP-TLS authentication process and the timings were noted. From the timing measurements in Table 5.12 it can be seen that the
values shown for some “EAP TLS Request/Response” messages are significantly higher (compared to the values for normal operation given in Table 5.6) resulting in a high overall timing. Thus our Timing Anomaly module detects the abnormal conditions experienced by the wireless hosts.

Although the DoS attack did not affect the authentication process itself the timing anomaly detection module detects a time delay in the authentication process. Such delays could be due to other factors of the wireless environment as well, and hence, needs to be further investigated. Therefore, without merely reporting this as an anomaly we need to investigate the legitimacy of the anomaly. Hence, our EWS passes this information to the intrusion prevention module, where the anomaly is substantiated using our proposed outlier based data association techniques.

## 5.4 Behavior Anomaly

### 5.4.1 Normal Behavior

The next stage in our anomaly detection process is the behavioral analysis of the participating hosts. The events shown in Table 5.13 represent the normal behavior of a station during an EAP-TLS association process. These events are extracted from the captured traces (as shown in Appendix C.1) by the event module. The
first four events on this table represent 802.11 open system authentication. The next event initiate the 802.1x mutual authentication with the “EAP REQUEST IDENTITY 1” message. The 802.1x authentication begins with the access point requesting the wireless station to identify itself. The response from the wireless station is redirected to the authentication server which in turn initiates the EAP type specific authentication process. Depending on the security configuration appropriate EAP type specific authentication will commence. Table 5.13 lists the ten EAP-TLS authentication events (40-45, 47-50). Event 40 - “EAP REQUEST 3 (PEAP)” is initiated by the authentication server since we have set the default EAP type to PEAP in our RADIUS authentication server. The Event IDs in the above table are the Event numbers introduced by the event module suitable for storage in the data cube at a later time. A complete map of the Event IDs and Events is given in Appendix B.

Similarly, Table 5.14 lists the EAP-PEAP authentication events. Here also the association begins with open system authentication followed by the 802.1x authentication. Unlike EAP-TLS authentication EAP-PEAP authentication consists of sixteen EAP type specific events (20, 22, 23, 25-27, 29-38). It can be

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OPEN AUTHENTICATION</td>
</tr>
<tr>
<td>3</td>
<td>OPEN ASSOCIATION REQUEST</td>
</tr>
<tr>
<td>6</td>
<td>EAP REQUEST 1 (IDENTITY)</td>
</tr>
<tr>
<td>7</td>
<td>EAP RESPONSE 1 (IDENTITY)</td>
</tr>
<tr>
<td>20</td>
<td>EAP REQUEST 2 (IDENTITY)</td>
</tr>
<tr>
<td>21</td>
<td>EAP RESPONSE 2 (IDENTITY)</td>
</tr>
<tr>
<td>40</td>
<td>EAP REQUEST 3 (PEAP)</td>
</tr>
<tr>
<td>41</td>
<td>EAP RESPONSE 3 (NAK)</td>
</tr>
<tr>
<td>42</td>
<td>EAP REQUEST 4 (TLS)</td>
</tr>
<tr>
<td>43</td>
<td>EAP RESPONSE 4 (TLS)</td>
</tr>
<tr>
<td>44</td>
<td>EAP REQUEST 5 (TLS)</td>
</tr>
<tr>
<td>45</td>
<td>EAP RESPONSE 5 (TLS)</td>
</tr>
<tr>
<td>47</td>
<td>EAP REQUEST 6 (TLS)</td>
</tr>
<tr>
<td>48</td>
<td>EAP RESPONSE 6 (TLS)</td>
</tr>
<tr>
<td>49</td>
<td>EAP REQUEST 7 (TLS)</td>
</tr>
<tr>
<td>50</td>
<td>EAP RESPONSE 7 (TLS)</td>
</tr>
<tr>
<td>51</td>
<td>EAP SUCCESS</td>
</tr>
<tr>
<td>52</td>
<td>EAPOL KEY</td>
</tr>
<tr>
<td>52</td>
<td>EAPOL KEY</td>
</tr>
<tr>
<td>52</td>
<td>EAPOL KEY</td>
</tr>
</tbody>
</table>

Table 5.13: EAP-TLS Events During Normal Behavior
Table 5.14: EAP-PEAP Events During Normal Behavior

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OPEN AUTHENTICATION</td>
</tr>
<tr>
<td>0</td>
<td>OPEN AUTHENTICATION</td>
</tr>
<tr>
<td>3</td>
<td>OPEN ASSOCIATION REQUEST</td>
</tr>
<tr>
<td>4</td>
<td>OPEN ASSOCIATION RESPONSE</td>
</tr>
<tr>
<td>6</td>
<td>EAP REQUEST 1 (IDENTITY)</td>
</tr>
<tr>
<td>7</td>
<td>EAP RESPONSE 1 (IDENTITY)</td>
</tr>
<tr>
<td>20</td>
<td>EAP REQUEST 2 (PEAP)</td>
</tr>
<tr>
<td>22</td>
<td>EAP RESPONSE 2 (PEAP)</td>
</tr>
<tr>
<td>23</td>
<td>EAP REQUEST 3 (PEAP)</td>
</tr>
<tr>
<td>25</td>
<td>EAP RESPONSE 3 (PEAP)</td>
</tr>
<tr>
<td>26</td>
<td>EAP REQUEST 4 (PEAP)</td>
</tr>
<tr>
<td>27</td>
<td>EAP RESPONSE 4 (PEAP)</td>
</tr>
<tr>
<td>29</td>
<td>EAP REQUEST 5 (PEAP)</td>
</tr>
<tr>
<td>30</td>
<td>EAP RESPONSE 5 (PEAP)</td>
</tr>
<tr>
<td>31</td>
<td>EAP REQUEST 6 (PEAP)</td>
</tr>
<tr>
<td>32</td>
<td>EAP RESPONSE 6 (PEAP)</td>
</tr>
<tr>
<td>33</td>
<td>EAP REQUEST 7 (PEAP)</td>
</tr>
<tr>
<td>34</td>
<td>EAP RESPONSE 7 (PEAP)</td>
</tr>
<tr>
<td>35</td>
<td>EAP REQUEST 8 (PEAP)</td>
</tr>
<tr>
<td>36</td>
<td>EAP RESPONSE 8 (PEAP)</td>
</tr>
<tr>
<td>37</td>
<td>EAP REQUEST 9 (PEAP)</td>
</tr>
<tr>
<td>38</td>
<td>EAP RESPONSE 9 (PEAP)</td>
</tr>
<tr>
<td>39</td>
<td>EAP SUCCESS</td>
</tr>
<tr>
<td>52</td>
<td>EAPOL KEY</td>
</tr>
<tr>
<td>52</td>
<td>EAPOL KEY</td>
</tr>
<tr>
<td>52</td>
<td>EAPOL KEY</td>
</tr>
<tr>
<td>52</td>
<td>EAPOL KEY</td>
</tr>
</tbody>
</table>

noticed that this list does not have event “EAP RESPONSE (NAK)”, since the wireless station is configured for EAP-PEAP authentication itself.

Table 5.15 lists the EAP-LEAP authentication events. Here again we have events 20 and 21 since the wireless station is configured for EAP-LEAP authentication. On all of the three types of authentication processes the last four messages (52) represent the key distribution phase. EAP-LEAP authentication process consists of five events (11-14, 18) only.

Tables 5.13, 5.14 and 5.15 all list the EAP type specific events during normal behavior. However, when anomalies arise this behavior can change. There can be situations where the number of events are extraordinarily high or low. There can also be situations where events can totally disappear from the receptive range of the monitoring devices. Therefore tracking these events and analyzing them appropriately may reveal vital information regarding any abnormality in the wireless environment. As discussed before here again we hold behavior profiles for every
5.4. BEHAVIOR ANOMALY

Table 5.15: EAP-LEAP Events During Normal Behavior

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OPEN AUTHENTICATION</td>
</tr>
<tr>
<td>0</td>
<td>OPEN AUTHENTICATION</td>
</tr>
<tr>
<td>3</td>
<td>OPEN ASSOCIATION REQUEST</td>
</tr>
<tr>
<td>4</td>
<td>OPEN ASSOCIATION RESPONSE</td>
</tr>
<tr>
<td>6</td>
<td>EAP REQUEST 1 (IDENTITY)</td>
</tr>
<tr>
<td>7</td>
<td>EAP RESPONSE 1 (IDENTITY)</td>
</tr>
<tr>
<td>20</td>
<td>EAP REQUEST 2 (PEAP)</td>
</tr>
<tr>
<td>21</td>
<td>EAP RESPONSE 2 (NAK)</td>
</tr>
<tr>
<td>11</td>
<td>EAP REQUEST 3 (LEAP)</td>
</tr>
<tr>
<td>12</td>
<td>EAP RESPONSE 3 (LEAP)</td>
</tr>
<tr>
<td>13</td>
<td>EAP SUCCESS</td>
</tr>
<tr>
<td>14</td>
<td>EAP REQUEST 4 (LEAP)</td>
</tr>
<tr>
<td>18</td>
<td>EAP RESPONSE 4 (LEAP)</td>
</tr>
<tr>
<td>52</td>
<td>EAPOL KEY</td>
</tr>
<tr>
<td>52</td>
<td>EAPOL KEY</td>
</tr>
<tr>
<td>52</td>
<td>EAPOL KEY</td>
</tr>
<tr>
<td>52</td>
<td>EAPOL KEY</td>
</tr>
</tbody>
</table>

participating station - access point pair. Therefore, when stations roam and reassociate via another access point we will then be analyzing the behavior with a profile specific to that access point.

5.4.2 Abnormal Behavior

In order to study the behavior of the wireless stations during abnormal conditions, we injected some abnormal management frames using the script listed in Appendix D.2, forcing the wireless hosts to deauthenticate and reassociate frequently. This was done by injecting “Deauthentication” and “Association Request” frames simulating a Replay attack.

Table 5.16: EAP-TLS Raw Events During Replay Attack

<table>
<thead>
<tr>
<th>Raw Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>STA3 - AP1 444.803634 360 AUTHENTICATION</td>
</tr>
<tr>
<td>STA3 - AP1 444.805244 362 ASSOCIATION REQUEST</td>
</tr>
<tr>
<td>AP1 - STA3 444.873872 813 DEAUTHENTICATION</td>
</tr>
<tr>
<td>STA3 - AP1 444.888390 374 ASSOCIATION REQUEST</td>
</tr>
<tr>
<td>AP1 - STA3 444.888758 814 DEAUTHENTICATION</td>
</tr>
<tr>
<td>STA3 - AP1 444.902434 376 ASSOCIATION REQUEST</td>
</tr>
<tr>
<td>AP1 - STA3 444.902874 815 DEAUTHENTICATION</td>
</tr>
<tr>
<td>STA3 - AP1 447.425796 053 AUTHENTICATION</td>
</tr>
<tr>
<td>AP1 - STA3 447.426138 883 AUTHENTICATION</td>
</tr>
<tr>
<td>STA3 - AP1 447.427177 054 ASSOCIATION REQUEST</td>
</tr>
<tr>
<td>AP1 - STA3 447.427769 884 ASSOCIATION RESPONSE</td>
</tr>
<tr>
<td>STA3 - AP1 447.428256 490 EAP REQUEST IDENTITY 1</td>
</tr>
<tr>
<td>STA3 - AP1 447.435340 056 EAP RESPONSE IDENTITY 1</td>
</tr>
<tr>
<td>.... adopts normal behavior from here</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
</tbody>
</table>
Table 5.16 lists the traces obtained during EAP-TLS authentication between station STA3 and the access point. Each line of the table corresponds to a single message passed between two wireless hosts, indicating the identities, direction of message transfer, the time of transfer, the sequence number and the event. A more comprehensive list of abnormal traces can be found in Appendix C.2. Here, the station adopts normal behavior once it successfully responds to the EAP Request Identity message, enabling the EAP authentication process.

<table>
<thead>
<tr>
<th>Raw Event</th>
<th>Event ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1 - STA2 85.025380 0509 AUTHENTICATION</td>
<td>0</td>
</tr>
<tr>
<td>STA2 - AP1 85.025936 0117 ASSOCIATION REQUEST</td>
<td>3</td>
</tr>
<tr>
<td>AP1 - STA2 85.026522 0510 ASSOCIATION RESPONSE</td>
<td>4</td>
</tr>
<tr>
<td>STA2 - AP1 85.027379 3694 ASSOCIATION REQUEST</td>
<td>3</td>
</tr>
<tr>
<td>AP1 - STA2 85.027884 1371 EAP REQUEST IDENTITY</td>
<td>6</td>
</tr>
<tr>
<td>STA2 - AP1 85.042119 3694 ASSOCIATION REQUEST</td>
<td>3</td>
</tr>
<tr>
<td>STA2 - AP1 85.042220 3694 ASSOCIATION REQUEST</td>
<td>3</td>
</tr>
<tr>
<td>AP1 - STA2 86.016005 0846 DEAUTHENTICATION</td>
<td>1</td>
</tr>
<tr>
<td>STA2 - AP1 90.743014 0216 AUTHENTICATION</td>
<td>0</td>
</tr>
<tr>
<td>AP1 - STA2 90.743440 0912 AUTHENTICATION</td>
<td>0</td>
</tr>
<tr>
<td>STA2 - AP1 90.744476 0217 ASSOCIATION REQUEST</td>
<td>3</td>
</tr>
<tr>
<td>AP1 - STA2 90.745070 0913 ASSOCIATION RESPONSE</td>
<td>4</td>
</tr>
<tr>
<td>AP1 - STA2 90.745531 1386 EAP REQUEST IDENTITY</td>
<td>6</td>
</tr>
<tr>
<td>STA2 - AP1 90.763463 0218 EAP RESPONSE IDENTITY</td>
<td>7</td>
</tr>
<tr>
<td>...... adopts normal behavior from here</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.17: EAP-PEAP Events During Replay Attack

<table>
<thead>
<tr>
<th>Raw Event</th>
<th>Event ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP1 - STA1 205.665782 1789 AUTHENTICATION</td>
<td>0</td>
</tr>
<tr>
<td>STA1 - AP1 205.666533 2725 ASSOCIATION REQUEST</td>
<td>3</td>
</tr>
<tr>
<td>AP1 - STA1 205.667928 1790 ASSOCIATION RESPONSE</td>
<td>4</td>
</tr>
<tr>
<td>AP1 - STA1 205.668035 1409 EAP REQUEST IDENTITY</td>
<td>6</td>
</tr>
<tr>
<td>AP1 - STA1 235.657322 2145 DEAUTHENTICATION</td>
<td>1</td>
</tr>
<tr>
<td>STA1 - AP1 241.369603 2824 AUTHENTICATION</td>
<td>0</td>
</tr>
<tr>
<td>AP1 - STA1 241.370002 2213 AUTHENTICATION</td>
<td>0</td>
</tr>
<tr>
<td>STA1 - AP1 241.370995 2825 ASSOCIATION REQUEST</td>
<td>3</td>
</tr>
<tr>
<td>AP1 - STA1 241.371589 2214 ASSOCIATION RESPONSE</td>
<td>4</td>
</tr>
<tr>
<td>AP1 - STA1 241.372046 1424 EAP REQUEST IDENTITY</td>
<td>6</td>
</tr>
<tr>
<td>STA1 - AP1 241.373288 2826 EAP RESPONSE IDENTITY</td>
<td>7</td>
</tr>
<tr>
<td>...... adopts normal behavior from here</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.18: EAP-LEAP Events During Replay Attack

Tables 5.16, 5.17 and 5.18 list the wireless traces obtained from a number of RSNA during EAP-TLS, EAP-PEAP and EAP-LEAP authentication process between the access point and stations STA3, STA2 and STA1 respectively. In all three scenarios we see a behavioral anomaly where the access point de-authenticate the station without responding to the replayed messages sent by the adversary.
This is due to the adversary sending an Association Request message without the correct credentials to meet the requirements of the access point.

However, although all three authentication mechanisms do not demonstrate similar behavior, EAP-PEAP and EAP-LEAP authentication mechanisms show similar behavior (Tables 5.17 and 5.18) compared to the EAP-TLS authentication process. This is because STA3, which is used for EAP-TLS, is a Windows XP client, whereas the other two are Linux clients. Since the adversary is also a Linux client the replayed Association Request message is acknowledged and the access point initiates EAP authentication with the EAP REQUEST IDENTITY message, which the adversary does not recognize and sends another Association Request message. At this point the access point realizes foul play and disconnects the station which results in the legitimate station also getting disconnected.

Here we have discussed results from a single experiment since the aim is to demonstrate what type of traces are captured during a behavioral anomaly. The behavior anomaly detector will detect all three scenarios as anomaly, but it will be up to the intrusion prevention module to substantiate the legitimacy of the anomalies. Hence, how the intrusion prevention module will substantiate these anomalies is discussed in the next chapter.

5.5 Conclusion

In this chapter, firstly, we have reported the use of timing analysis to detect timing anomalies in the wireless environment. Timing anomalies can occur in the wireless environment due to various factors including security vulnerabilities. Security attacks such as MitM can give rise to difference in round trip timing values. Hence, it may be vital to keep monitoring round trip timing values between round trip messages so that any abnormal timings could be further analyzed. In this study we have mainly developed the timing profiles using the mean and standard deviation values of round trip timing values. However, depending on the
wireless environment it may be useful to consider other statistical methods for calculating the timing profiles. Therefore, timing study of wireless messages itself can lead to further study.

Next, we have discussed the use of behavioral analysis for detecting abnormal conditions in the wireless environment. Here, we have used the standard theoretical/practical behavior profiles of EAP-LEAP, PEAP, and TLS authenticated wireless hosts to compare the actual behavior during a specific authentication process. If any abnormal behavior is detected we analyzed it further to verify its legitimacy. Behavioral anomalies can be triggered due to various factors including security vulnerabilities. Mismanaged or misconfigured wireless hosts, atmospheric conditions, change in security settings etc. could also lead to behavioral anomalies. Hence, it may not be possible to declare every detected behavioral anomaly as legitimate resulting a large number of false positive reporting. Therefore, it is important to effectively substantiate every detected behavioral anomaly.

The anomalous conditions considered in this chapter are mainly due to DoS and Replay attacks. Although we have not considered other forms of vulnerabilities, our main aim was to demonstrate the effect of timing and/or behavioral anomalies and not the analysis of the security issues. In this respect, we have revealed the influence of timing and/or behavioral anomalies in the wireless environment and discussed how it could be used as the first level of defense in intrusion detection process.

In the next chapter we examine the validity of the proposed substantiating measures, the Group Confidence Level (GCL) and the Group Outlier Score (GOS). We analyze the various results obtained from a number of tests carried out and discuss the advantages and the disadvantages of our approach. We also discuss the effectiveness of the proposed substantiating mechanism and present the best possible applications of our approach.
Chapter 6

Substantiation

6.1 Introduction

In Sections 5.3 and 5.4, we presented a variety of experimental results demonstrating the use of timings and behavioral analysis in detecting anomalies in wireless networks. As discussed in section 4.2.1 detecting timing and/or behavioral anomalies alone does not lead to any fruitful results unless the anomalies are substantiated effectively. Thus, in this chapter we validate our two proposed approaches to substantiating anomalies.

First, we discuss the use of Group Confidence Level (GCL) in measuring the cohesiveness of a group of events. Any associated group of events under normal conditions should demonstrate high cohesion in relation to their association. Hence, we established the notion of Group Confidence Level to measure the consistency of a set of associated group events. Any inconsistencies in the associated group events may be due to some abnormality. We have used GCL to measure the cohesiveness of EAP-LEAP, PEAP, and TLS authentication processes and discussed the effect of this measure in detecting security threats. We have also analyzed the accuracy of our detection mechanism with and without using GCL.

Next, although the association process in wireless networks always consist of associated events, it is also vital to study the effect of individual events within
the group events. In this context, here we analyze the use of Group Outlier Score (GOS) for identifying individual outliers. We have used GOS to identify outliers during EAP-LEAP, PEAP, and TLS authentication process and examined the effect of this measure in detecting security threats. We have also analyzed the accuracy of this method with different threshold values for GOS.

### 6.1.1 Chapter Overview

Details of analysis is described in this chapter as follows: In Section 6.2 we present and discuss the results obtained from using Group Confidence Level for detecting anomalies in an associated group of events. In Section 6.3, we discuss the effect of GOS in substantiating security threats and in Section 6.4 we examine the effectiveness of our approach.

### 6.2 Group Confidence

We examine the application of GCL under three different conditions: (i) Normal operation, (ii) DoS Attack, and (iii) Replay Attack. Under each of the three conditions we examine the effect of the three authentication schemes, EAP-LEAP, PEAP and TLS. Within each authentication schemes we examine both IEEE 802.11 group events and EAP group events.

#### 6.2.1 Normal Operation

<table>
<thead>
<tr>
<th>EAP-LEAP</th>
<th>EAP-PEAP</th>
<th>EAP-TLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.13</td>
<td>0.11</td>
<td>0.14</td>
</tr>
<tr>
<td>0.07</td>
<td>0.15</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 6.1: Ideal Normalized Support

Figures 6.1, 6.2 and 6.3 show the normalized support (Equation 4.3) of EAP type specific events of wireless stations configured for EAP-LEAP, EAP-PEAP and EAP-TLS authentication respectively. These readings were obtained from
6.2. GROUP CONFIDENCE

Figure 6.1: Normal EAP-LEAP Authentication

our experimental setup during normal operations. Here, we use the term normal operation meaning that there was no abnormal events introduced explicitly. However, the wireless environment was exposed to the regular campus wide wireless traffic. The three figures also indicate the ideal normalized support (Table 6.1) for both IEEE 802.11 and EAP events. The event numbers used in these diagrams are serial values that correspond to the event IDs in Tables 5.13, 5.14 and 5.15 in the same order. The ideal normalized support values were calculated considering EAP type specific events without any abnormal conditions. For example, consider a EAP-TLS authentication process with fifteen EAP events (Table 5.13). Assuming only EAP-TLS configured wireless stations, according to Equation 4.3, the ideal support for each event would be $-1/15 \log(1/15) = 0.08$.

<table>
<thead>
<tr>
<th>Trial</th>
<th>EAP-LEAP</th>
<th>EAP-PEAP</th>
<th>EAP-TLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>96.78</td>
<td>97.40</td>
<td>97.14</td>
</tr>
<tr>
<td>2</td>
<td>95.21</td>
<td>97.65</td>
<td>96.72</td>
</tr>
<tr>
<td>3</td>
<td>96.21</td>
<td>96.89</td>
<td>96.45</td>
</tr>
</tbody>
</table>

Table 6.2: Confidence Levels During Normal Operation

In the above three figures that show the behavior of wireless hosts under normal
operations, the normalized support values of most events have values close to the ideal normalized support. Although some EAP-LEAP and EAP-PEAP events show larger deviation from the ideal support values, the overall confidence is guaranteed as shown in Table 6.2.

<table>
<thead>
<tr>
<th>No.</th>
<th>Event ID</th>
<th>Protocol ID</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>74</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>66</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>88</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>60</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>1</td>
<td>71</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>1</td>
<td>68</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>1</td>
<td>61</td>
</tr>
<tr>
<td>9</td>
<td>23</td>
<td>1</td>
<td>65</td>
</tr>
<tr>
<td>10</td>
<td>24</td>
<td>1</td>
<td>63</td>
</tr>
<tr>
<td>11</td>
<td>42</td>
<td>2</td>
<td>68</td>
</tr>
<tr>
<td>12</td>
<td>43</td>
<td>2</td>
<td>64</td>
</tr>
<tr>
<td>13</td>
<td>44</td>
<td>2</td>
<td>65</td>
</tr>
<tr>
<td>14</td>
<td>45</td>
<td>2</td>
<td>61</td>
</tr>
<tr>
<td>15</td>
<td>47</td>
<td>2</td>
<td>60</td>
</tr>
<tr>
<td>16</td>
<td>48</td>
<td>2</td>
<td>62</td>
</tr>
<tr>
<td>17</td>
<td>49</td>
<td>2</td>
<td>64</td>
</tr>
<tr>
<td>18</td>
<td>50</td>
<td>2</td>
<td>62</td>
</tr>
<tr>
<td>19</td>
<td>51</td>
<td>1</td>
<td>66</td>
</tr>
</tbody>
</table>

Table 6.3: EAP-TLS Event Count During Normal Operations

This table (Table 6.2) lists the the confidence levels (Equation 4.7) of IEEE
802.11 and EAP group events under EAP-LEAP, EAP-PEAP and EAP-TLS authentication. To calculate these values, the outlier detection process made at most two drill-down queries on the data cube and the time taken to execute a single result was approximately 0.0324 ms in our EWS. We made use of the count measure stored on each cell to calculate the individual frequencies of the events and their aggregate frequencies. Table 6.3 lists the query results of the first trial, showing the individual counts of the IEEE 802.11 and EAP-TLS events. To calculate the aggregate values we executed a second query, from this query we obtained 348 as the aggregate for IEEE 802.11 events and 964 as aggregate for EAP events. From these results the confidence level for the EAP protocol events is calculated using Equation 4.7 as follows:

\[
\sum_{c' \in \text{EAP-TLS events}} freq(c', k) \log(freq(c', k)) = 1.1756
\]

\[
C(c, k) = \left[ -\sum_{c' \in \text{group events}} freq(c', k) \log(freq(c', k)) \right] \times 100\% = 99.63\%
\]

In a similar manner we calculated the confidence levels for EAP-LEAP and
PEAP authentication methods. In many data warehousing applications the measure values stored in cells are fixed. In our case since we need real time response the measure values need to be updated on-the-fly when new legitimate events are identified.

From these results it is evident that all three authentication schemes demonstrate high confidence (between 96.89% and 99.18% as shown in Table 6.2) during normal operations irrespective of some events deviating from normalized ideal support values. This is due to the fact that all IEEE 802.11 and EAP events, as a group, behave in a predictable manner under normal conditions. However, under abnormal conditions their group behavior may not be guaranteed. To further emphasize this claim we consider two abnormal situations in the wireless environment and discuss their results in the next two subsections.

6.2.2 During DoS Attack

![figure](image)

Figure 6.4: DoS Attack during EAP-LEAP Authentication

Figures 6.4, 6.5 and 6.6 show the normalized support for EAP type specific events on wireless stations configured for EAP-LEAP, EAP-PEAP and EAP-TLS.
6.2. **GROUP CONFIDENCE**

authentication respectively, during a DoS attack. In all three cases the average support for IEEE 802.11 events deviate from the normalized ideal support by more than 34%. Deviation during EAP-TLS authentication was the highest at 42.53% followed by EAP-LEAP authentication at 34.98% and during EAP-PEAP authentication at 34.62%. These deviation percentages are calculated by finding the ratio between normalized ideal support and the difference between the average deviation of normalized support and the normalized ideal support. These results demonstrate the unreliable nature of 802.11 open system association caused by the injected “Deauthentication” frames during the attack.

During the DoS attack, events “AUTHENTICATION” (1,2), “ASSOCIATION REQUEST” (5) and “ASSOCIATION RESPONSE” (6) have similar support in both EAP-LEAP and EAP-PEAP authentication modes. This is due to the legitimate reassociation after a deauthentication. However, “DEAUTHENTICATION” (3,4) events have high frequency because of the injected “Deauthentication” frames. Furthermore, the EAP events on all three stations show less deviation from the normalized ideal support (20.49% for EAP-LEAP, 19.36 for EAP-PEAP and 4.14% for EAP-TLS) compared to the 802.11 events. This is
because of the large number of 802.11 “DEAUTHENTICATION” events injected and the corresponding response. However, reassociation does not take place at the same frequency, since the channel is busy with “DEAUTHENTICATION” events. In some cases when 802.11 open system authentication is in progress there has been interruptions due to the injected “DEAUTHENTICATION” events resulting in fresh reassociation. Also, out of all three authentication process the EAP-TLS events show very low deviation (4.14%) from the normalized ideal support. This confirms the reliability of EAP-TLS authentication process.

Further, if we look at the EAP events, except for the first “EAP REQUEST IDENTITY 1” event all other EAP events have support values close to ideal. This is because “EAP REQUEST IDENTITY 1” event is triggered by the access point and at this point the actual EAP authentication has not commenced. Therefore, a bogus “DEAUTHENTICATION” event can still interrupt the association process resulting in fresh reassociation.

As discussed above, Figures 6.4, 6.5 and 6.6 all demonstrate the behavior of RSNA events during the DoS attack. To further investigate this behavior we
calculated the confidence level for each case. Table 6.4 shows the confidence levels of IEEE 802.11 and EAP events obtained from three different trials. Table 6.5, shows the query results of EAP-TLS events for the first trial. As described above the confidence levels were calculated using the aggregate values obtained from a second query. Table 6.4 lists the confidence levels. Accordingly, the confidence level for IEEE 802.11 events are comparatively low (between 58% and 79% as shown in Table 6.4) for all three authentication types in all three trials. Whereas, the confidence level for EAP events are high. However, station STA3 configured for EAP-TLS authentication demonstrates very high confidence (in the range of 96.81% to 98.12%) for EAP events in all three trials. This confirms the robustness of EAP-TLS authentication process. This can be further justified from the normalized support diagrams where EAP-TLS events demonstrate an average deviation of 4.14% from the normalized ideal support compared to 20.48% for EAP-LEAP events and 19.36% for EAP-PEAP events.
6.2.3 During Replay Attack

Figures 6.7, 6.8 and 6.9 show the normalized support for EAP type specific events on wireless stations configured for EAP-LEAP, EAP-PEAP and EAP-TLS authentication respectively, during a Replay attack. Here again, we calculated the deviation of 802.11 events from the normalized ideal support. Accordingly, we had 15.38%, 28.58% and 40.67% deviations during EAP-LEAP, PEAP and TLS authentication processes respectively. This is because of the replayed “Authentication” and “Association Request” messages. Consequently, as in the case of DoS attack this demonstrates the unreliable nature of 802.11 open system association. In this case the events “AUTHENTICATION” (1) and ”ASSOCIATION REQUEST” (4), both replayed messages, have very high frequency count.

<table>
<thead>
<tr>
<th>Trial</th>
<th>Authentication Type</th>
<th>EAP-LEAP</th>
<th>EAP-PEAP</th>
<th>EAP-TLS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>802.11</td>
<td>EAP</td>
<td>802.11</td>
<td>EAP</td>
</tr>
<tr>
<td>1</td>
<td>71.52</td>
<td>94.77</td>
<td>80.52</td>
<td>94.03</td>
</tr>
<tr>
<td>2</td>
<td>68.52</td>
<td>94.03</td>
<td>71.52</td>
<td>94.77</td>
</tr>
<tr>
<td>3</td>
<td>63.74</td>
<td>91.23</td>
<td>64.46</td>
<td>93.21</td>
</tr>
</tbody>
</table>

Table 6.6: Confidence Levels Under Replay Attack

To further investigate, as in the case of DoS attack, we calculated the GCL for
6.2. GROUP CONFIDENCE

Figure 6.8: Replay Attack during EAP-PEAP Authentication

each case. Table 6.6 shows the GCL values obtained from three trials for EAP-LEAP, PEAP and TLS authentication. The confidence levels for 802.11 events are low for all three authentication schemes in all three trial (less than 80.52%). However, the confidence level for EAP events are comparatively high (greater than 91.23%). From these results it is evident that the IEEE 802.11 events are more vulnerable to the replay attack than the EAP events. It can be further noticed that the EAP-TLS events show more confidence than the EAP-LEAP and PEAP events. This can be justified from the normalized support diagrams where EAP-TLS events demonstrate a variations of 4.67% from the ideal support compared to 9.37% for EAP-LEAP events and 8.88% for EAP-PEAP events.

Analyzing, both DoS and Replay attack scenarios it can be noticed that the variations in the average normalized support for EAP-LEAP and PEAP events (20.48% and 19.36% respectively) during the DoS attack are considerably high compared to that during Replay attack (9.37% for EAP-LEAP and 8.88% for EAP-PEAP). This is confirmed with GCL values which are low during DoS attack (58.82% to 78.45%) compared to that during Replay attack (63.74% to 80.52%).
This is because during the DoS attack due to the injected “DEAUTHENTICATION” messages the authentication process gets interrupted after the first EAP message since at this stage the RADIUS authentication server does not come into action. It is only after the RADIUS authentication server issues the “EAP REQUEST PEAP 2” (since the default EAP type is set to PEAP on the RADIUS authentication server) message the EAP events become stable. Hence, as soon as the EAP type specific authentication begins the association process becomes reliable. Although, the DoS attack can jam the authentication process delaying certain message exchange (as seen in Table 5.10), the process itself is robust due to the effectiveness of the EAP.

The above examples show promising results for our proposed concept of using timing and/or behavioral analysis with outlier based confidence measure for detecting and substantiating abnormal conditions. Although in our experiments we have not tested the vulnerabilities of the EAP authentication process, the same concept can be used to detect future exploits on the EAP process. Further, in most attack scenarios the attacker needs to inject many messages before he could actually compromise the credentials of a legitimate station. Therefore, as soon
as we detect an abnormal condition we can continue to monitor the suspicious stations and once sufficient traces are collected we can substantiate the abnormality using outlier based data association techniques. In this manner we can either raise an alarm that a security breach is on the verge or give an indication of the level of threat for the exposed station.

In the foregoing paragraphs we have discussed the vulnerabilities of the various protocols and how they affect the overall confidence of the RSNA. However, in the case of RSN, it is important to note, that although IEEE 802.11 and EAP protocols may demonstrate low confidence levels, it is up to the security policy maker to substantiate it appropriately depending on user privileges. Because in the two attack scenarios discussed, even though the behavior indicates that the RSN is unstable during the 802.11 and EAP protocol phases, it only affects the “availability” of the stations and does not compromise the “authenticity” of the participating stations. Hence, when formulating the security threat levels one has to consider a range of factors, mainly the user priorities, level of sensitivity of the network etc. and make appropriate decisions.

### 6.3 Group Outlier Score

In the proposed EWS, when timing and/or behavioral anomalies are detected, it is vital to verify the legitimacy of such anomalies. The intrusion prevention module plays a major role by validating and substantiating the anomalies detected by the timing and/or behavioral anomaly detection modules. In this context in the previous section we presented results from analyzing associated group of events using GCL. In contrast, in this section we present the results obtained from analyzing the association of individual events within the group of events. The type and nature of the wireless traces collected for analysis can depend on the type of attack/attacks targeted on the wireless environment. In this context, firstly, we collected about ten million wireless traces (over a period of one week)
from a controlled environment to build the initial data cube. A special capturing tool was developed for this purpose [81]. The traces were converted with suitable mappings to suit the requirements of the data cube algorithms [26].

The captured traces were converted using a parsing program and the data cube was built with the seven attributes shown in Table 4.3. The time taken to build the data cube was 1.32 seconds. Indexing the data cube took a further 1.51 seconds. With a dimension of seven we get total of 128 different views of the captured traces. However, we built the data cube with only those selected views (Table 4.4) necessary to process our queries. For example, consider the following data associations needed for analyzing network anomalies between stations and access points.

- Source ID, Destination ID, Event ID
- Source ID, Event ID, Protocol ID
- Destination ID, Event ID, Protocol ID
- Source ID, Destination ID
- Source ID, Event ID
- Destination ID, Event ID

As explained in subsections 4.3.1 and 4.3.2, in order to process these queries we require only ABC, ACF, BCF, AB, AC and BC views. However, depending on the anticipated security threats we may need to analyze more number of views as discussed in section 4.3.2. Let us now consider the substantiation mechanism, first under normal conditions and thereafter with various attack scenarios: Replay attack, Malicious Authentication and Session Hijack.

### 6.3.1 Normal Conditions

Before exploring the application of GOS in substantiating abnormal events within associated group of events, we analyze the GOS values under normal conditions to establish threshold values. For this purpose we first analyze the GOS values
6.3. **GROUP OUTLIER SCORE**

for EAP-LEAP, PEAP and TLS events under normal conditions.

<table>
<thead>
<tr>
<th>No.</th>
<th>Src. ID</th>
<th>Dest. ID</th>
<th>Event ID</th>
<th>Count</th>
<th>GOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>128</td>
<td>247</td>
<td>0</td>
<td>70</td>
<td>1.09</td>
</tr>
<tr>
<td>2</td>
<td>247</td>
<td>128</td>
<td>0</td>
<td>69</td>
<td>1.09</td>
</tr>
<tr>
<td>3</td>
<td>128</td>
<td>247</td>
<td>1</td>
<td>86</td>
<td>1.01</td>
</tr>
<tr>
<td>4</td>
<td>247</td>
<td>128</td>
<td>2</td>
<td>75</td>
<td>1.06</td>
</tr>
<tr>
<td>5</td>
<td>247</td>
<td>128</td>
<td>3</td>
<td>64</td>
<td>1.12</td>
</tr>
<tr>
<td>6</td>
<td>128</td>
<td>247</td>
<td>4</td>
<td>72</td>
<td>1.08</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>247</td>
<td>6</td>
<td>83</td>
<td>1.03</td>
</tr>
<tr>
<td>8</td>
<td>247</td>
<td>128</td>
<td>7</td>
<td>76</td>
<td>1.06</td>
</tr>
<tr>
<td>9</td>
<td>128</td>
<td>247</td>
<td>11</td>
<td>72</td>
<td>1.08</td>
</tr>
<tr>
<td>10</td>
<td>247</td>
<td>128</td>
<td>12</td>
<td>78</td>
<td>1.05</td>
</tr>
<tr>
<td>11</td>
<td>128</td>
<td>247</td>
<td>13</td>
<td>63</td>
<td>1.12</td>
</tr>
<tr>
<td>12</td>
<td>247</td>
<td>128</td>
<td>14</td>
<td>64</td>
<td>1.12</td>
</tr>
<tr>
<td>13</td>
<td>128</td>
<td>247</td>
<td>18</td>
<td>74</td>
<td>1.07</td>
</tr>
<tr>
<td>14</td>
<td>128</td>
<td>247</td>
<td>20</td>
<td>68</td>
<td>1.10</td>
</tr>
<tr>
<td>15</td>
<td>128</td>
<td>247</td>
<td>21</td>
<td>75</td>
<td>1.06</td>
</tr>
</tbody>
</table>

Table 6.7: GOS for EAP-LEAP Events Under Normal Operation

<table>
<thead>
<tr>
<th>No.</th>
<th>Src. ID</th>
<th>Dest. ID</th>
<th>Event ID</th>
<th>Count</th>
<th>GOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>249</td>
<td>128</td>
<td>0</td>
<td>66</td>
<td>1.03</td>
</tr>
<tr>
<td>2</td>
<td>128</td>
<td>249</td>
<td>0</td>
<td>74</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>128</td>
<td>249</td>
<td>1</td>
<td>67</td>
<td>1.02</td>
</tr>
<tr>
<td>4</td>
<td>249</td>
<td>128</td>
<td>2</td>
<td>66</td>
<td>1.03</td>
</tr>
<tr>
<td>5</td>
<td>249</td>
<td>128</td>
<td>3</td>
<td>62</td>
<td>1.05</td>
</tr>
<tr>
<td>6</td>
<td>128</td>
<td>249</td>
<td>4</td>
<td>71</td>
<td>1.01</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>249</td>
<td>6</td>
<td>50</td>
<td>1.11</td>
</tr>
<tr>
<td>8</td>
<td>249</td>
<td>128</td>
<td>7</td>
<td>71</td>
<td>1.01</td>
</tr>
<tr>
<td>9</td>
<td>128</td>
<td>249</td>
<td>20</td>
<td>52</td>
<td>1.10</td>
</tr>
<tr>
<td>10</td>
<td>249</td>
<td>128</td>
<td>22</td>
<td>57</td>
<td>1.07</td>
</tr>
<tr>
<td>11</td>
<td>128</td>
<td>249</td>
<td>23</td>
<td>68</td>
<td>1.02</td>
</tr>
<tr>
<td>12</td>
<td>249</td>
<td>128</td>
<td>25</td>
<td>61</td>
<td>1.05</td>
</tr>
<tr>
<td>13</td>
<td>128</td>
<td>249</td>
<td>26</td>
<td>63</td>
<td>1.04</td>
</tr>
<tr>
<td>14</td>
<td>249</td>
<td>128</td>
<td>27</td>
<td>61</td>
<td>1.05</td>
</tr>
<tr>
<td>15</td>
<td>128</td>
<td>249</td>
<td>29</td>
<td>54</td>
<td>1.09</td>
</tr>
<tr>
<td>16</td>
<td>249</td>
<td>128</td>
<td>30</td>
<td>61</td>
<td>1.05</td>
</tr>
<tr>
<td>17</td>
<td>128</td>
<td>249</td>
<td>31</td>
<td>69</td>
<td>1.02</td>
</tr>
<tr>
<td>18</td>
<td>249</td>
<td>128</td>
<td>32</td>
<td>65</td>
<td>1.03</td>
</tr>
<tr>
<td>19</td>
<td>128</td>
<td>249</td>
<td>33</td>
<td>62</td>
<td>1.05</td>
</tr>
<tr>
<td>20</td>
<td>249</td>
<td>128</td>
<td>34</td>
<td>62</td>
<td>1.05</td>
</tr>
<tr>
<td>21</td>
<td>128</td>
<td>249</td>
<td>35</td>
<td>61</td>
<td>1.05</td>
</tr>
<tr>
<td>22</td>
<td>249</td>
<td>128</td>
<td>36</td>
<td>62</td>
<td>1.05</td>
</tr>
<tr>
<td>23</td>
<td>128</td>
<td>249</td>
<td>37</td>
<td>67</td>
<td>1.02</td>
</tr>
<tr>
<td>24</td>
<td>249</td>
<td>128</td>
<td>38</td>
<td>66</td>
<td>1.03</td>
</tr>
<tr>
<td>25</td>
<td>128</td>
<td>249</td>
<td>39</td>
<td>56</td>
<td>1.08</td>
</tr>
</tbody>
</table>

Table 6.8: GOS for EAP-PEAP Events Under Normal Operation

In order to establish the GOS values under normal conditions we used the initial data collected from the controlled environment. Tables 6.7, 6.8 and 6.9 all show the GOS values obtained for all EAP-LEAP, PEAP and TLS events under normal conditions. Here, the count values show the number of events captured during the test period. The GOS values in the tables are calculated using Equation 4.6. In order to calculate the GOS we need to know the relative
frequency (Equation 4.2) of the event and its uncertainty value (Equation 4.5). For example if we consider event 43 in Table 6.9 it has a count of 140. To calculate the relative frequency we need to make a second query to find the parent count, which in this case was 2615. Then the GOS value is calculated as follows:

Using Equation 4.2

\[
freq(c, k) = \frac{140}{2615}
\]

\[
log(freq(c, k)) = -1.271
\]

Using Equation 4.5

\[
H(c, k) = -\sum_{c' \in \text{neighbor}(c, k)} freq(c', k) \log(freq(c', k))
\]

\[
= 1.232
\]

Using Equation 4.6

\[
G(c, k) = -\frac{\log(freq(c, k))}{H(c, k)}
\]

\[
\text{Hence}, \ G(c, k) = -\left(\frac{-1.271}{1.232}\right)
\]

\[
= 1.03
\]
6.3. GROUP OUTLIER SCORE

Having calculated the GOS values, it is essential to establish a viable threshold for effective substantiation of anomalies. In order to do this we used the GOS values where the number event count is the lowest for the three authentication schemes. It could be noticed from the tables that the GOS values increase as the event count decrease. When the event count decreases with reference to other associated events it means the event is deviating from its normal behavior.

<table>
<thead>
<tr>
<th>Authentication</th>
<th>Max GOS</th>
<th>Threshold GOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAP-LEAP</td>
<td>1.12</td>
<td>1.17</td>
</tr>
<tr>
<td>EAP-PEAP</td>
<td>1.11</td>
<td>1.18</td>
</tr>
<tr>
<td>EAP-TLS</td>
<td>1.14</td>
<td>1.20</td>
</tr>
</tbody>
</table>

Table 6.10: GOS Threshold Values

From tables 6.7, 6.8 and 6.9 the maximum GOS values for EAP-LEAP, PEAP and TLS authentication schemes are 1.12, 1.11 and 1.14 respectively. However, since we must also accommodate legitimate reduction in count which may arise due to the inherent nature of the wireless environment, we use the maximum GOS value with a tolerance of 5% to determine the threshold values. Table 6.10 shows the threshold values thus derived for the three authentication schemes.

6.3.2 Replay Attack

<table>
<thead>
<tr>
<th>No.</th>
<th>Src. ID</th>
<th>Dest. ID</th>
<th>Event ID</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>247</td>
<td>128</td>
<td>3</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>128</td>
<td>247</td>
<td>4</td>
<td>36</td>
</tr>
<tr>
<td>3</td>
<td>128</td>
<td>247</td>
<td>6</td>
<td>32</td>
</tr>
<tr>
<td>4</td>
<td>247</td>
<td>128</td>
<td>7</td>
<td>41</td>
</tr>
<tr>
<td>5</td>
<td>128</td>
<td>247</td>
<td>20</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>247</td>
<td>128</td>
<td>21</td>
<td>41</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>247</td>
<td>40</td>
<td>36</td>
</tr>
<tr>
<td>8</td>
<td>247</td>
<td>128</td>
<td>41</td>
<td>36</td>
</tr>
<tr>
<td>9</td>
<td>128</td>
<td>247</td>
<td>42</td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>247</td>
<td>128</td>
<td>43</td>
<td>32</td>
</tr>
<tr>
<td>11</td>
<td>128</td>
<td>247</td>
<td>44</td>
<td>33</td>
</tr>
<tr>
<td>12</td>
<td>247</td>
<td>128</td>
<td>45</td>
<td>38</td>
</tr>
<tr>
<td>13</td>
<td>128</td>
<td>247</td>
<td>47</td>
<td>35</td>
</tr>
<tr>
<td>14</td>
<td>247</td>
<td>128</td>
<td>48</td>
<td>28</td>
</tr>
<tr>
<td>15</td>
<td>128</td>
<td>247</td>
<td>49</td>
<td>32</td>
</tr>
<tr>
<td>16</td>
<td>247</td>
<td>128</td>
<td>50</td>
<td>39</td>
</tr>
<tr>
<td>17</td>
<td>128</td>
<td>247</td>
<td>51</td>
<td>41</td>
</tr>
</tbody>
</table>

Table 6.11: RSNA Events on STA3 for D=11
CHAPTER 6. SUBSTANTIATION

The query results shown in Table 6.11 were obtained using view ABCD. Here attributes A, B, C and D represent "Src. ID", "Dest. ID", "Event ID" and "Time of Day" respectively. In this query attribute D has the value of 11 (11am to 12noon). The query result shows the number of events associated with a particular station during a particular time period. Here Source/Dest ID 247 refers to station STA3 and Source/Dest ID 128 refers to access point AP1. Event IDs ranging from 3 to 51 refers to events 2 to 20 listed in Table 4.7, such as 3 representing “Association Request”, 4 representing “Association Response”, 6 representing “EAP Request Identity 1” etc. Table 6.12 shows two abnormal events captured during a different time period (D=16). From this table it is evident that station STA3 is issuing some superfluous messages without any prerequisites. In such a scenario if we consider only view ABCE, provided there are no other messages passed on other channels, we will have an accumulated count of 43 for event with ID 7, and 33 for event with ID 43 as shown in Table 6.13.

The GOS values in Table 6.13 are calculated using Equation 4.6. In order to calculate the GOS we need to know the relative frequency (Equation 4.2) of the event and its uncertainty value (Equation 4.5). For example if we consider event 43 in Table 6.11 it has a count of 32. To calculate the relative frequency we need to make a second query to find the parent count, which in this case is 647. Then the GOS value is calculated as follows:

Using Equation 4.2

\[
freq(c, k) = \frac{32}{647}
\]
6.3. **GROUP OUTLIER SCORE**

\[
\log(\text{freq}(c, k)) = -1.306
\]

Using Equation 4.5

\[
H(c, k) = -\sum_{c' \in \text{neighbor}(c, k)} \text{freq}(c', k) \log(\text{freq}(c', k))
\]

\[
= 1.189
\]

Using Equation 4.6

\[
G(c, k) = -\left(\frac{\log(\text{freq}(c, k))}{H(c, k)}\right)
\]

\[
\text{Hence, } G(c, k) = -\left(\frac{-1.306}{1.189}\right)
\]

\[
= 1.10
\]

<table>
<thead>
<tr>
<th>No.</th>
<th>Src. ID</th>
<th>Dest. ID</th>
<th>Event ID</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>249</td>
<td>128</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>128</td>
<td>247</td>
<td>4</td>
<td>62</td>
</tr>
<tr>
<td>3</td>
<td>128</td>
<td>249</td>
<td>6</td>
<td>33</td>
</tr>
<tr>
<td>4</td>
<td>249</td>
<td>128</td>
<td>7</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>128</td>
<td>249</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>249</td>
<td>128</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>249</td>
<td>20</td>
<td>38</td>
</tr>
<tr>
<td>8</td>
<td>249</td>
<td>128</td>
<td>21</td>
<td>39</td>
</tr>
<tr>
<td>9</td>
<td>128</td>
<td>249</td>
<td>40</td>
<td>35</td>
</tr>
<tr>
<td>10</td>
<td>249</td>
<td>128</td>
<td>41</td>
<td>37</td>
</tr>
<tr>
<td>11</td>
<td>128</td>
<td>249</td>
<td>42</td>
<td>45</td>
</tr>
<tr>
<td>12</td>
<td>249</td>
<td>128</td>
<td>43</td>
<td>62</td>
</tr>
<tr>
<td>13</td>
<td>128</td>
<td>249</td>
<td>44</td>
<td>46</td>
</tr>
<tr>
<td>14</td>
<td>249</td>
<td>128</td>
<td>45</td>
<td>39</td>
</tr>
<tr>
<td>15</td>
<td>128</td>
<td>249</td>
<td>47</td>
<td>39</td>
</tr>
<tr>
<td>16</td>
<td>249</td>
<td>128</td>
<td>48</td>
<td>38</td>
</tr>
<tr>
<td>17</td>
<td>128</td>
<td>249</td>
<td>49</td>
<td>35</td>
</tr>
<tr>
<td>18</td>
<td>249</td>
<td>128</td>
<td>50</td>
<td>44</td>
</tr>
<tr>
<td>19</td>
<td>128</td>
<td>249</td>
<td>51</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 6.14: RSNA Events on STA4

With reference to the GOS values in Table 6.13, if we set a threshold value of 1.1 (as discussed in Section 6.3.1) to detect rare events, our system will report both incidents as normal on view ABCE and as abnormal on view ABCD. Since events with ID 7 and 43 do not have any other associations during this time period we substantiate this as an anomaly and categorize it as a Replay Attack because of view ABCD (as discussed in section 4.3.2).
In another similar experiment we considered a different incident involving view ABCE, triggered due to a behavioral anomaly. Table 6.14 shows the query results obtained using view ABCE on channel 11 for station STA4. In this scenario we can see two events (with ID’s 9 & 10) having a very low count. The GOS value for those two events worked out to be 2.53, revealing them as rare events according to the threshold values established in the previous section. Thus we substantiate those two incidents as anomalies and categorize them as a Replay Attack because of views ABCE and ABCD.

![Figure 6.10: Normal and Abnormal RSNA Events](image)

To further illustrate the effectiveness of our substantiation mechanism we present the GOS values of the various incidents associated with the different stations in a single representation. Figure 6.10 shows the GOS values of RSNA events during an EAP-TLS authentication for stations STA2, STA3 and STA4. The GOS values of all events for station STA2 are in the range of 1.02 to 1.08, and therefore we consider it as normal since the values are less than 1.2. Whereas, station ST3 has two events with high GOS values (1.46 and 1.33). These are the two events with ID 7 and 43 (in Figure 6.10 events 5 and 11), which are usually present during a regular EAP-TLS authentication process. However, in this case these
two events have been captured in isolation without the other relevant messages. Hence when we calculate their GOS values on view ABCD, they appear to be high and therefore we categorized it as due to a Replay Attack. Similarly, station STA4 also has two events with high GOS values (1.85 and 1.81). Events with ID’s 9 and 10 (in Figure 6.10 events 6 and 7) are considered abnormal because these two events do not match the normal behavior of station STA4 during an EAP-TLS authentication process. The GOS values of these two events on view ABCE are also high. Hence, with a threshold value of 1.2 we categorize those events as abnormal.

Above, we have illustrated some experiments showing the use of GOS values for detecting rare events and grouping events based on their remoteness. Detecting rare events in the wireless environment could be useful in identifying any unusual messages passed between the wireless hosts that may eventually lead to an impending security breach. In the case of an impending security threat it would be extremely useful to categorize the type of threat so that relevant warning alarms could be raised.

6.3.3 Perceptions of GOS values

Figure 6.11 shows the GOS values for the EAP-TLS authentication events during a DoS attack. In the case of DoS attack the wireless environment was flooded with “Deauthentication” messages to disrupt stations from connecting to the access point. Hence, in such cases huge amounts of messages of a particular type are injected from a malicious station. In this diagram the normal traces have GOS values in the range 1 to 1.2. When a station is targeted with a DoS attack the total number of events associated with the station increases enormously, resulting in the increase of GOS values. If we consider the DoS plot we can group the first five events in one category where the GOS values are less than 2.5. These five events are in effect due to the DoS attack, where IEEE 802.11 association is repeated several times. The next fifteen events are the actual EAP-TLS events which
do not get affected by the DoS attack, although the attack can disconnect the station after it completes the EAP-TLS authentication process. These events can be grouped into another category having GOS values above 4. Hence, from this example we can see that the GOS values of normal traces (the EAP type specific events) can take high values than the abnormal (IEEE 802.11) events. Therefore, one must be cautious when applying the GOS in cases where the abnormal event count is higher than the normal event count. However, we can use this notion in a different perspective to relate group events depending on the GOS value. Therefore, the choice and use of the threshold value can depend on the type of threat that we are interested and how we want to interpret the results. To further exploit the GOS values we consider EAP-LEAP and EAP-PEAP authentication.

Figures 6.12 and 6.13 show the GOS values for EAP-LEAP and EAP-PEAP authentication events during both normal and DoS attack. As in the case of EAP-TLS, these two authentication schemes also demonstrate a similar behavior during the DoS attack. In all three authentication schemes the EAP events have almost constant GOS values demonstrating the reliability of EAP authentication. Hence, we conclude that the RSNA events can be grouped based on the GOS values and
Figure 6.12: DoS Attack During EAP-LEAP Authentication

depending on the type of abnormality it could present different types of information. Next, let us consider the different types of attacks described in Section 3.3.3, namely Malicious Association, Malicious Association during authentication and the Session Hijack.

6.3.4 Malicious Association

As analyzed in Section 3.3.3, attackers can perform malicious associations with a legitimate wireless host during the discovery phase. This can be done by masquerading as a legitimate host and providing incorrect credentials to the access point. This is done by providing invalid RSN IE information in the “Association Request” message. Having received the incorrect credentials, the access point will ignore the legitimate host assuming it as RSN incapable. Thereafter, the intruder host can masquerade as the legitimate access point and continue the association with the legitimate host. This experiment was performed in our test-bed and relevant data was collected.

The query results shown in Table 6.15 were obtained using view ABCF. Here
attributes A, B, C and F represent “Src. ID”, ”Dest. ID”, ”Event ID” and “Protocol ID” respectively. The query result shows the number of events associated with a particular station during EAP-LEAP authentication process. Here Source/Dest ID 247 refers to station STA3 and Source/Dest ID 128 refers to access point AP1. Event IDs ranging from 0 to 21 refers to the various EAP-LEAP events, such as 3 representing “Association Request”, 4 representing “Association Response”, 6 representing “EAP Request Identity 1” and so on.

The events and their corresponding counts listed in Table 6.15 were obtained
6.3. **GROUP OUTLIER SCORE**

during a malicious association attack. According to the table, event number 3 has almost double the number of count compared with the other events. This is due to the “Association Request” message sent by the intruder and the legitimate host. Event 3 - “Association Request” has a count of thirty one compared to the thirteen “EAP Success” messages (Event 13).

![GOS for Malicious Association](image)

**Figure 6.14: GOS for Malicious Association**

Figure 6.14 shows the effect of GOS values in this type of attack. As in the figure, except for event with Event ID 3 rest of the events have GOS values in the range of 1.07 to 1.14. However, in this case since the abnormal event has higher count than the normal events the GOS value of the abnormal event turn out to be 0.83. This is another perspective of using the GOS value. Therefore, it is apparent that the use of GOS is versatile and in this situation it is used to identify a more frequent individual event. However, in our anomaly detection process we use the GOS measure mainly for substantiating abnormal conditions with less frequent events. The EWS does not allow individual frequent events to accumulate since we have fixed the lower limit for GOS threshold to 1. In the next example we discuss an attack with less number of abnormal events than the normal events.
CHAPTER 6. SUBSTANTIATION

6.3.5 Malicious Authentication

In Section 3.3.3, we discussed how a malicious authentication could take place during the EAP authentication phase of the RSN association. This can be done during a EAP-PEAP authentication if MD5 password is used. An intruder could send a “EAP Failure” message to a legitimate host and associate with the legitimate access point providing the correct “Challenge” response. This is done by tracking the EAP authentication messages and providing the required authentication information after discovering the MD5 password. We conducted this experiment in our test-bed and data was collected.

<table>
<thead>
<tr>
<th>No.</th>
<th>Src. ID</th>
<th>Dest. ID</th>
<th>Event ID</th>
<th>MalAuth1</th>
<th>MalAuth2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Count 1</td>
<td>GOS</td>
</tr>
<tr>
<td>1</td>
<td>249</td>
<td>128</td>
<td>0</td>
<td>68</td>
<td>1.05</td>
</tr>
<tr>
<td>2</td>
<td>128</td>
<td>249</td>
<td>0</td>
<td>74</td>
<td>1.02</td>
</tr>
<tr>
<td>3</td>
<td>128</td>
<td>249</td>
<td>1</td>
<td>69</td>
<td>1.04</td>
</tr>
<tr>
<td>4</td>
<td>249</td>
<td>128</td>
<td>2</td>
<td>72</td>
<td>1.03</td>
</tr>
<tr>
<td>5</td>
<td>249</td>
<td>128</td>
<td>3</td>
<td>69</td>
<td>1.07</td>
</tr>
<tr>
<td>6</td>
<td>128</td>
<td>249</td>
<td>4</td>
<td>71</td>
<td>1.03</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>249</td>
<td>6</td>
<td>69</td>
<td>1.04</td>
</tr>
<tr>
<td>8</td>
<td>249</td>
<td>128</td>
<td>7</td>
<td>70</td>
<td>1.03</td>
</tr>
<tr>
<td>9</td>
<td>128</td>
<td>249</td>
<td>20</td>
<td>72</td>
<td>1.03</td>
</tr>
<tr>
<td>10</td>
<td>249</td>
<td>128</td>
<td>22</td>
<td>67</td>
<td>1.05</td>
</tr>
<tr>
<td>11</td>
<td>128</td>
<td>249</td>
<td>23</td>
<td>68</td>
<td>1.05</td>
</tr>
<tr>
<td>12</td>
<td>128</td>
<td>249</td>
<td>28</td>
<td>12</td>
<td>1.56</td>
</tr>
<tr>
<td>13</td>
<td>249</td>
<td>128</td>
<td>25</td>
<td>71</td>
<td>1.03</td>
</tr>
<tr>
<td>14</td>
<td>128</td>
<td>249</td>
<td>26</td>
<td>73</td>
<td>1.03</td>
</tr>
<tr>
<td>15</td>
<td>249</td>
<td>128</td>
<td>27</td>
<td>69</td>
<td>1.04</td>
</tr>
<tr>
<td>16</td>
<td>128</td>
<td>249</td>
<td>29</td>
<td>74</td>
<td>1.02</td>
</tr>
<tr>
<td>17</td>
<td>249</td>
<td>128</td>
<td>30</td>
<td>70</td>
<td>1.04</td>
</tr>
<tr>
<td>18</td>
<td>128</td>
<td>249</td>
<td>31</td>
<td>69</td>
<td>1.04</td>
</tr>
<tr>
<td>19</td>
<td>249</td>
<td>128</td>
<td>32</td>
<td>72</td>
<td>1.03</td>
</tr>
<tr>
<td>20</td>
<td>128</td>
<td>249</td>
<td>33</td>
<td>73</td>
<td>1.03</td>
</tr>
<tr>
<td>21</td>
<td>249</td>
<td>128</td>
<td>34</td>
<td>68</td>
<td>1.05</td>
</tr>
<tr>
<td>22</td>
<td>128</td>
<td>249</td>
<td>35</td>
<td>71</td>
<td>1.03</td>
</tr>
<tr>
<td>23</td>
<td>249</td>
<td>128</td>
<td>36</td>
<td>70</td>
<td>1.04</td>
</tr>
<tr>
<td>24</td>
<td>128</td>
<td>249</td>
<td>37</td>
<td>69</td>
<td>1.04</td>
</tr>
<tr>
<td>25</td>
<td>249</td>
<td>128</td>
<td>38</td>
<td>72</td>
<td>1.03</td>
</tr>
<tr>
<td>26</td>
<td>128</td>
<td>249</td>
<td>39</td>
<td>73</td>
<td>1.03</td>
</tr>
</tbody>
</table>

Table 6.16: GOS Values During Malicious Authentication

In Table 6.16 we provide details of the EAP-PEAP authentication events and discuss how this type of attacks could be substantiated. These query results were obtained using view ABCF. Here attributes A, B, C and F represent "Src. ID", "Dest. ID", "Event ID" and “Protocol ID” (not listed in the table) respectively.
The query result shows the number of events associated with a particular station during EAP-PEAP authentication process. Here Source/Dest ID 249 refers to station STA4 and Source/Dest ID 128 refers to access point AP1. Event IDs ranging from 0 to 39 refers to the various EAP-PEAP events, such as 3 representing “Association Request”, 4 representing “Association Response”, 6 representing “EAP Request Identity 1” and so on.

Table 6.16 shows the GOS values during a malicious authentication. In this table we have shown two situations where “Count 1” and the adjacent “GOS” values are from seventy four authentication trails (MalAuth1) and “Count 2” and the adjacent “GOS” values are from three authentication trials (MalAuth2). We have provided these two different data sets as a comparison to demonstrate the capabilities of GOS. In the table event 28 stands for “EAP Failure” that triggers a behavioral anomaly since it cannot be present in a normal RSN association. If there has been twelve failures, the number of successful associations too should be less by twelve. However, from the table it is evident that even with the failure messages the number of “EAP Success” messages (Event 39) is seventy three. Similar observation could be made with the less number of trials too. Hence, our behavior anomaly detector triggers the twelve associations with “EAP Failure” messages as anomalies. The figure below illustrates how this attack could be substantiated by our intrusion prevention module.

As evident in Figure 6.15, the abnormal event - “EAP failure”, has a higher GOS value compared to the other events. Both “MalAuth1” and “MalAuth2” demonstrate similar behavior although one has large number of trials than the other. Therefore, using GOS values it is possible to differentiate whether an abnormal event is legitimate or not, provided we select the correct threshold value for the GOS.
6.3.6 Session Hijack

A Session Hijack is a more advanced attack, where the association between a legitimate station and the access point is hijacked by an illegitimate user. In this case the illegitimate user can force a channel change with the access point/station and masquerade as a legitimate access point/station. Hence, in this kind of a threat we need to track the source, destination, event, channel and protocol of the messages exchanged. By tracking the protocol we establish whether the illegitimate session establishes a different kind of association. Therefore, for this type of a threat, views associated with attributes A, B, C, E and F are considered. In section 3.3.3 we have formally verified the prospect of a session hijack during the association process. We now consider this vulnerability and discuss how it could be detected using outlier detection techniques.

Table 6.17 shows the GOS values for EAP-TLS events obtained from view ABCF. Here except for event 46, “EAP Failure”, all other events have GOS values less than 1.2. In this scenario although the GOS value of the “EAP Failure”
6.3. GROUP OUTLIER SCORE

Table 6.17: GOS Values During a Session Hijack

<table>
<thead>
<tr>
<th>No.</th>
<th>Src. ID</th>
<th>Dest. ID</th>
<th>Event ID</th>
<th>Count</th>
<th>GOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>249</td>
<td>128</td>
<td>3</td>
<td>45</td>
<td>1.05</td>
</tr>
<tr>
<td>2</td>
<td>128</td>
<td>247</td>
<td>4</td>
<td>41</td>
<td>1.04</td>
</tr>
<tr>
<td>3</td>
<td>128</td>
<td>249</td>
<td>6</td>
<td>38</td>
<td>1.05</td>
</tr>
<tr>
<td>4</td>
<td>249</td>
<td>128</td>
<td>7</td>
<td>45</td>
<td>1.06</td>
</tr>
<tr>
<td>5</td>
<td>128</td>
<td>249</td>
<td>9</td>
<td>39</td>
<td>1.07</td>
</tr>
<tr>
<td>6</td>
<td>249</td>
<td>128</td>
<td>10</td>
<td>42</td>
<td>1.06</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>249</td>
<td>20</td>
<td>41</td>
<td>1.03</td>
</tr>
<tr>
<td>8</td>
<td>249</td>
<td>128</td>
<td>21</td>
<td>45</td>
<td>1.03</td>
</tr>
<tr>
<td>9</td>
<td>128</td>
<td>249</td>
<td>40</td>
<td>39</td>
<td>1.05</td>
</tr>
<tr>
<td>10</td>
<td>249</td>
<td>128</td>
<td>41</td>
<td>36</td>
<td>1.05</td>
</tr>
<tr>
<td>11</td>
<td>128</td>
<td>249</td>
<td>42</td>
<td>42</td>
<td>1.03</td>
</tr>
<tr>
<td>12</td>
<td>249</td>
<td>128</td>
<td>43</td>
<td>39</td>
<td>1.04</td>
</tr>
<tr>
<td>13</td>
<td>128</td>
<td>249</td>
<td>44</td>
<td>45</td>
<td>1.05</td>
</tr>
<tr>
<td>14</td>
<td>249</td>
<td>128</td>
<td>45</td>
<td>44</td>
<td>1.08</td>
</tr>
<tr>
<td>15</td>
<td>128</td>
<td>249</td>
<td>46</td>
<td>7</td>
<td>1.62</td>
</tr>
<tr>
<td>16</td>
<td>128</td>
<td>249</td>
<td>47</td>
<td>51</td>
<td>1.06</td>
</tr>
<tr>
<td>17</td>
<td>249</td>
<td>128</td>
<td>48</td>
<td>45</td>
<td>1.03</td>
</tr>
<tr>
<td>18</td>
<td>249</td>
<td>128</td>
<td>49</td>
<td>48</td>
<td>1.06</td>
</tr>
<tr>
<td>19</td>
<td>128</td>
<td>249</td>
<td>50</td>
<td>49</td>
<td>1.04</td>
</tr>
</tbody>
</table>

event is above the threshold value, the message itself cannot be considered abnormal because it is a possible behavior during a EAP-TLS authentication process. Further, although events 47 to 51 show slightly higher count compared to the other events, their GOS values are still within the allowable range (less than 1.2). Therefore, to further verify this abnormality we consider this scenario from a different view point. Hence, we now consider view ABCE which includes the channel ID of the communication.

Table 6.18: GOS Values from a Different View

<table>
<thead>
<tr>
<th>No.</th>
<th>Src. ID</th>
<th>Dest. ID</th>
<th>Event ID</th>
<th>Count</th>
<th>GOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>249</td>
<td>128</td>
<td>3</td>
<td>45</td>
<td>1.08</td>
</tr>
<tr>
<td>2</td>
<td>128</td>
<td>247</td>
<td>4</td>
<td>41</td>
<td>1.08</td>
</tr>
<tr>
<td>3</td>
<td>128</td>
<td>249</td>
<td>6</td>
<td>38</td>
<td>1.09</td>
</tr>
<tr>
<td>4</td>
<td>249</td>
<td>128</td>
<td>7</td>
<td>45</td>
<td>1.10</td>
</tr>
<tr>
<td>5</td>
<td>128</td>
<td>249</td>
<td>9</td>
<td>39</td>
<td>1.08</td>
</tr>
<tr>
<td>6</td>
<td>249</td>
<td>128</td>
<td>10</td>
<td>42</td>
<td>1.09</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>249</td>
<td>20</td>
<td>41</td>
<td>1.08</td>
</tr>
<tr>
<td>8</td>
<td>249</td>
<td>128</td>
<td>21</td>
<td>45</td>
<td>1.09</td>
</tr>
<tr>
<td>9</td>
<td>128</td>
<td>249</td>
<td>40</td>
<td>39</td>
<td>1.08</td>
</tr>
<tr>
<td>10</td>
<td>249</td>
<td>128</td>
<td>41</td>
<td>36</td>
<td>1.09</td>
</tr>
<tr>
<td>11</td>
<td>128</td>
<td>249</td>
<td>42</td>
<td>42</td>
<td>1.08</td>
</tr>
<tr>
<td>12</td>
<td>249</td>
<td>128</td>
<td>43</td>
<td>39</td>
<td>1.10</td>
</tr>
<tr>
<td>13</td>
<td>128</td>
<td>249</td>
<td>44</td>
<td>45</td>
<td>1.09</td>
</tr>
<tr>
<td>14</td>
<td>249</td>
<td>128</td>
<td>45</td>
<td>44</td>
<td>1.10</td>
</tr>
<tr>
<td>15</td>
<td>128</td>
<td>249</td>
<td>46</td>
<td>7</td>
<td>1.61</td>
</tr>
<tr>
<td>16</td>
<td>128</td>
<td>249</td>
<td>47</td>
<td>6</td>
<td>1.66</td>
</tr>
<tr>
<td>17</td>
<td>128</td>
<td>249</td>
<td>48</td>
<td>7</td>
<td>1.61</td>
</tr>
<tr>
<td>18</td>
<td>249</td>
<td>128</td>
<td>49</td>
<td>5</td>
<td>1.72</td>
</tr>
<tr>
<td>19</td>
<td>128</td>
<td>249</td>
<td>50</td>
<td>7</td>
<td>1.61</td>
</tr>
<tr>
<td>19</td>
<td>128</td>
<td>249</td>
<td>51</td>
<td>6</td>
<td>1.66</td>
</tr>
</tbody>
</table>
As could be seen in Table 6.18 the GOS values for events with Event IDs 3 to 45 are less than 1.2 and for events with Event IDs 46 to 51 are more than 1.2. This confirms that all events from 15 to 19 are rare and abnormal. Hence, it is evident that there are other rare events that are associated with the rare event detected from analyzing the earlier view ABCF (Table 6.17). This concludes that this abnormal condition is due to a session hijack attack (Table 4.5). The intruder does a channel change with the legitimate host with event 46 and then continues to masquerade as a legitimate access point duping the legitimate host. The legitimate host unknowingly associates with the intruder. The group confidence level for this group of events is 93.14 (on view ABCE) as opposed to 98.94 in the case of group of events on view ABCF.

### 6.4 Effectiveness of GOS

Here we investigate the effectiveness of the proposed substantiating mechanisms. Figure 6.16 shows the accuracy of EWS with and without substantiation of EAP-TLS, PEAP and LEAP authenticated hosts during a Replay attack.
saying without substantiation, we mean all possible anomalies detected by the anomaly detection module. Here, the abnormalities were forced with some having an effect on the wireless hosts and the others being replications, by a group of seven students over a period of one week. The EWS without substantiation raised a large number of false alarms and the EWS with substantiation demonstrated almost 99% accuracy for a GOS threshold of 1.2. The large number of alarms raised when substantiation was not active was due to the fact that the EWS was merely reporting all timing and/or behavior anomalies. According to figure 6.16 all three authentication mechanisms demonstrate high accuracy with substantiation. Whereas, without substantiation the detection accuracy of EAP-LEAP authenticated hosts is the lowest compared to that of EAP-TLS and PEAP.

![Figure 6.17: EWS Accuracy for DoS Attack](image)

Similarly, Figure 6.17 shows the accuracy of EAP-TLS, PEAP and LEAP authenticated hosts during a DoS attack with and without substantiation. As for the replay attack, the abnormalities were forced by the same group of students over a period of one week. Here again all three authentication mechanisms demonstrate high accuracy with substantiation. However, the accuracy of EAP-LEAP authenticated hosts is the least without substantiation as in the case of replay.
In view of analyzing the effectiveness of the substantiation mechanism for different threshold values, we studied the performance of the EWS introducing various types of abnormalities. Figure 6.18 shows the accuracy of the substantiation mechanism for different GOS threshold values. Accordingly, in this scenario the optimal value for the threshold value appears to be 1.2. However, in practice we cannot fix the threshold level to any particular value and it will be left to the discretion of the network security administrator to choose the best possible value depending on the nature of operation at any given time.

6.5 Conclusion

In this chapter, first we have discussed the use of group confidence level for substantiating anomalies. For this purpose we have considered both replay and DoS attacks and demonstrated the use of confidence level as measure of group cohesiveness. Any associated group of events under normal conditions should demonstrate very high confidence in relation to their cohesiveness. Hence, we established the
notion of confidence level to measure the consistency of a set of associated group of events. Any inconsistencies in the associated group of events may be due to some abnormality. In this context, we have shown that the experimental results obtained with the 802.11i based network are promising and confirming the concept of the proposed method [111].

The GCLs obtained for EAP-LEAP, PEAP and TLS show that the EAP events as a group demonstrate high confidence than the IEEE 802.11 events. This confirms the effectiveness of IEEE 802.1x based authorization and authentication adapted by IEEE 802.11i security mechanism. The GCL approach to substantiating anomalies in this context is effective for studying the cohesiveness of associated group events [109]. Further, in the case of wireless environment there can be abnormal situations where the event count of individual events is very small compared to that of other related group of events. In such situations we need to study the effect of individual events in relation to the group. For this purpose we have used and analyzed the effectiveness of GOS in this chapter.

In order to start using the GOS we had to initially determine the GOS threshold for EAP-LEAP, PEAP and TLS events. This was done using the wireless traces obtained during normal operations. Using these threshold values we analyzed a number of abnormal conditions in the wireless environment. Firstly, we demonstrated how a Replay attack could be detected using the GOS values obtained from different views of the data cube [112]. Next, we have discussed the use of GOS values in a scenario where the number of abnormal events are comparatively high. The inspiration with the GOS value is that it could be used to detect individual rare/frequent events and to group associated events. Hence, it was discussed how various events could be grouped based on the GOS values. In contrast to the group confidence level, the GOS gives a suggestion as to which events can be grouped based on their association. This phenomenon is useful not only in the study of security issues, but in areas where we need to classify groups of events based on their relationship, such as stock market analysis, crime type
However, the use of GOS values will not be viable in situations where the normal events count are very low compared to the abnormal events. In such scenarios, higher threshold values need to be set for normal operation and lower values for abnormal events which are no longer rare. Anyhow, the problem of detecting frequent events is not challenging as for detecting rare events. Frequent individual events can be detected and substantiated merely by counting because they have to be in par with all other associated events. Whereas this rule cannot be applied for rare individual events. In this view, the use of GOS values is very much effective in applications where individual rare events need to be detected and substantiated.

Next, we have considered Replay Attacks and discussed the use of GOS values in detecting such security vulnerabilities. It has been proved that our methodology is efficient in detecting rare events within a group of associated events. We have then studied the accuracy of GOS values in substantiating anomalies during EAP-TLS, PEAP and LEAP authentication process. In all three situations the EWS demonstrated 96 to 98% accuracy in detecting legitimate anomalies for a GOS value of 1.2. We further studied the effectiveness of having different threshold values for GOS.

We have also discussed the use of GOS values in a situation where the number of abnormal events are very low. For this purpose we have considered different types of attacks in the wireless environment. First, we have discussed the effectiveness of GOS values during a malicious association process and then discussed a session hijack attack [110]. In both scenarios the number of abnormal events are low compared to the regular events. As discussed earlier it is the behavior anomaly detection modules that first detects such anomalies and passes it to the intrusion prevention module for substantiation. In both the attack scenarios the substantiation mechanism proved to detect the attack. However, in the case of the session hijack attack the use of multiple views was essential in classifying the
type of attack.

The EWS uses the GOS measures to detect individual rare/frequent events and uses different views to substantiate it. In cases where there is a (sub)group of rare/frequent events within the group of associated events, the EWS uses the GCL to substantiate it. In such case the GOS measures are used only to (sub)group the rare/frequent events. It must be noted that the GCL values are calculated external to the data cube, whereas, GOS values are stored in the data cube itself as a measure.
Chapter 7

Conclusions and Future Work

Over the last few years the use of wireless networks have seen a considerable rise. More and more institutions, organizations and individuals are counting on wireless networks for their day to day activities due to its cost effectiveness and flexibility. With this increasing dependence on WLANs, the need for a robust security mechanism has become inevitable. The latest WLAN security protocol, the IEEE 802.11i assures rigid security for wireless networks with the support of IEEE 802.1x protocol for authentication, authorization and key distribution. Nevertheless, due to the ever increasing threats and/or anomalies experienced in the wireless environment users remain skeptical on the practical trustworthiness of these security mechanisms. In this context, we proposed a novel Early Warning System that can effectively detect anomalies, substantiate them and also identify the basis for such malicious behavior. Our proposed system has a number of levels of defense to scrutinize malicious behaviors of the wireless network environment, caused by a range of factors including security issues. Security alerts are raised only when the legitimacy of abnormal conditions is validated using effective outlier based substantiation techniques.
7.1 Anomaly Detection

Anomalies in the wireless environment can be triggered due to a number of factors including security vulnerabilities, mismanaged or misconfigured wireless hosts, atmospheric conditions, change in security settings etc. Reporting every timing and/or behavioral anomaly as an intrusion can lead to a large number of false positive reporting. Therefore, for effective intrusion detection we must carefully analyze all detected timing and/or behavioral anomalies before raising an alarm.

Timing anomalies can occur due to various conditions including security vulnerabilities in the wireless environment. Hence, detecting and analyzing such anomalies may lead to significant advancement towards the detection of misbehaving wireless hosts. In this view, we have reported the use of timing analysis to detect timing anomalies in the wireless environment. Security attacks such as MitM can give rise to difference in round trip timing values. Hence, it may be vital to keep monitoring round trip timing values between round trip messages so that any abnormal timings could be further analyzed. In this study we have developed a model and analyzed the use of timing profiles to detect timing anomalies. Normal timing profiles were first built using the mean and standard deviation of round trip timing values and was used to compare with the actual round trip timings of every wireless host connecting to an access point. Although, in this study we have used only the standard deviation approach, depending on the wireless environment it may be useful to further investigate other means of calculating the timing profiles as well.

Further, to enhance the capabilities of our detection mechanism we have also considered the effect of behavioral anomalies of the wireless hosts. Every wireless host that connects to the wireless network exhibits a particular behavior. This behavior may vary depending on a number of issues including security vulnerabilities. Mismanaged or misconfigured wireless hosts, atmospheric conditions, change in security settings etc., could also lead to behavioral anomalies. In this
context, we have shown that detecting and analyzing the behavior of wireless hosts can be of significant importance for detecting abnormal conditions in the wireless environment. Hence, we have firstly developed the normal behavior profiles of EAP-LEAP, PEAP, and TLS configured wireless hosts considering the formal models developed and their practical behavior during normal operations. Behavior anomalies were detected by comparing the profiles of normal behavior and the actual behavior of the wireless hosts during a specific authentication process. If any abnormal behavior is detected we analyze it further to verify its legitimacy. Behavioral anomalies can be triggered due to various conditions including security vulnerabilities. Declaring every detected behavioral anomaly as legitimate results in a large number of false positive reporting. Therefore, it is important to effectively substantiate every detected behavioral anomaly.

The anomalous conditions considered throughout our study are mainly due to DoS and Replay attacks. Although we have not considered other forms of vulnerabilities, our main aim was to demonstrate the effect of timing and/or behavioral anomalies and not the analysis of the security issues. In this respect, we have revealed the effectiveness of timing and/or behavioral anomalies in the wireless environment and discussed how it could be used as the first level of defense in intrusion detection process. It is shown with a number of examples, that the proposed timing and behavioral anomaly detection modules do effectively detect anomalies.

7.2 Substantiation

Real time detection of outliers from large multi-level longitudinal data sets of network traces can lead to effective intrusion detection and prevention. Presently, due to lack of fast on-the-fly updating and processing capabilities, Intrusion Detection Systems do not detect intruders instantly. Also, achieving dynamic adaptation in real time has been a long standing desire for effective intrusion detection and
prevention. Most IDSs cannot adapt their detection mechanism in real time to accommodate illegitimate dynamic changes. Furthermore, analyzing multivariate and latent variable data sets is complex and time consuming. Under these circumstance, we have discussed a novel substantiation mechanism to validate anomalies using outlier based data association techniques.

Detecting outliers in real time is intricate and challenging. However, we proposed two methods to substantiate anomalies in real time. First we have discussed the use of group confidence level for substantiating anomalies in associated group events. For this purpose we have considered both replay and DoS attacks and demonstrated the use of confidence level as measure of group cohesiveness. Any associated group of events under normal conditions should demonstrate very high cohesiveness. Hence, we established the notion of Group Confidence Level to measure the cohesiveness of a set of associated group events. Any inconsistencies in the associated group events may be due to some abnormality. In this context, the experimental results obtained with the 802.11i based network are promising and confirming the concept of the proposed method.

The GCLs obtained for EAP-LEAP, PEAP and TLS show that the EAP events as a group demonstrate high confidence than the IEEE 802.11 events. This confirms the effectiveness of IEEE 802.1x based authorization, authentication and key distribution adapted by the IEEE 802.11i security mechanism. The GCL approach to substantiating anomalies is effective in analyzing the cohesiveness of associated group events. Further, in the case of wireless environment there can be situations where the number of individual events can be very small compared to that of other related group events. In such situations we need to study the coupling of individual events in relation to the group. For this purpose we have developed and analyzed another factor called the Group Outlier Score.

In order to use the GOS for substantiation we had to initially fix a GOS threshold for EAP-LEAP, PEAP and TLS events. This was done using the wireless traces obtained during normal operations. Once we determine the threshold
values, we then analyzed a number of abnormal conditions in the wireless environment. Firstly, we demonstrated how a Replay attack could be effectively detected using the GOS values obtained from different views of the data cube. Next, we have discussed the use of GOS values in a scenario where the number of abnormal events are comparatively high. Further, the GOS values can also be used to group associated events. In this context, we have discussed how various events could be grouped based on the GOS values. In contrast to the group confidence level, the GOS can be used to group events based on their coupling within the group. This phenomenon is useful not only in the study of security issues, but in areas where we need to classify group of events based on their coupling, such as stock market analysis, crime type analysis, credit card fraud etc.

The use of GOS is also feasible in situations where the frequency of normal events are very low compared to that of abnormal events. In such scenarios, higher threshold values need to be set for normal operation and lower values for abnormal events which are no longer rare. However, in such situations the use of GCL will be more effective if the number of abnormal events is more than one. Also, for detecting and substantiating individual frequent events we can use simple counting, since they have to be in par with all other associated group events. Hence, the use of GOS values is very much effective in applications where rare events need to be detected. In this view, we studied the effectiveness of GOS values in substantiating anomalies during EAP-TLS, PEAP and LEAP authentication process. In all three situations the EWS demonstrated 96 to 98% accuracy in detecting legitimate anomalies with a GOS threshold value of 1.2. Further, in order to study the effect of our detection mechanism under different threshold values we conducted several experiments and found that a threshold value of 1.2 was the optimum for our experimental setup.

We have also discussed the use of GOS values in a situation where the number of abnormal events are very much low. For this purpose we have considered different types of attacks in the wireless environment. First, we have discussed
the effectiveness of GOS values during a malicious association process and then discussed a session hijack attack. In both scenarios the number of abnormal events are low compared to the regular events. As discussed earlier it is the behavior anomaly detection modules that first detects such anomalies and passes it to the intrusion prevention module for substantiation. In both the attack scenarios the experimental results demonstrate that the substantiation mechanism proved to detect the attack. However, in the case of the session hijack attack the use of multiple views was essential in classifying the attack.

7.3 Conclusions

In this study, we have reported the use of timing and/or behavioral analysis for detecting abnormal conditions in the wireless environment and using outlier based data association techniques to substantiate the abnormal conditions. The experimental results obtained with the 802.11i based network are promising and confirming to the concept of the proposed detection mechanism. Although, the proposed system is not tested to detect all of the security threats, the main aim was to validate our methodology. In this view it is shown that analyzing the wireless environment from different view points, such as, protocols used, channels used, time used etc., can effectively substantiate the legitimacy of abnormal conditions created by unusual events.

Further, we have proposed two novel mechanisms to detect abnormal events GCL and GOS. The proposed methods are developed based on combining data mining methods with OLAP techniques. We have shown how a security breach could be evaluated considering the cohesiveness of group events and the coupling of related events. We have also demonstrated how the GOS values could be interpreted to raise alarm. Furthermore, we have discussed how different views of the data cube could be used to analyze the abnormality and find the appropriate cause of a security breach, that is the type of attack.
Furthermore, we have discussed the use of GOS values to detect rare events and associate them to a security threat in the wireless environment. We have also shown the effectiveness of our mechanism with and without substantiation mechanisms. However, as there are no other similar systems available we have not been able to do a direct comparison with any other publicly available techniques. Further, we have discussed the use of GOS values to group events based on their coupling, however, we have not explored its application other than for detecting DoS attacks. DoS attacks generate large amounts of abnormal events and hence are easily detectable.

The analysis of the results demonstrate the effectiveness of our proposed system in detecting various security threats and the ability to classify the type of threats. However, we have not tested our system with abnormalities that can interfere with the EAP authentication process itself. As of to date EAP vulnerability on the wireless environment has not been reported. However, the results presented here shows that since our detection mechanism is more generic it may be capable of detecting future exploits of EAP vulnerability. Although this technique is expected to provide very promising results in cooperate networks, their applicability on smaller networks may be limited due to resource requirements.

The main contribution of this study is the two effective substantiation mechanisms used to validate anomalies. Although anomalies can be of several forms, detecting rare events and grouping them based on their remoteness is challenging. Our concept could be applied to several fields including credit card transactions, health monitoring systems, Internet security, maritime border security, air traffic control and the like. The wireless environment considered in this study is one such example where the number of anomalies and their nature vary drastically. Hence, we have demonstrated that our validation mechanism is capable of managing such situations successfully.
7.4 Future Direction

The introduction of data cubes and the related data association techniques enables our proposed EWS to be scalable to meet the needs of large distributed networking environments. Further, with the current advancement of multi-processor systems it is also possible to implement the EWS on a single computer with multiple sensors that is capable of monitoring small and medium scale organizations. Hence, as a future expansion, we intend to develop our proposed EWS capable of detecting and substantiating anomalies in an integrated network environment. As a first step the usefulness of our anomaly detection and substantiation mechanism in a sensor network environment can be studied. It is also possible to investigate the feasibility of integrating the operations of a mesh network to study the configuration issues in mesh networks.

Further, detecting and analyzing abnormal conditions are not specific to computer networks. It can be useful in many applications that demonstrates a behavior according to a set protocol. Hence, the theory developed in this study could be applied to many applications such as credit card transactions, health monitoring systems, Internet security, maritime border security, air traffic control etc. Therefore, the theories developed in this study can be extended to those applications.
Appendix A

Packet Capturing

The first phase of the EWS is the capture of wireless traces. The raw wireless traces need to be captured and only the management frames need to be selected and passed to the other modules for our further processing. The packet capturing module was developed in c++ and executed on the Monitoring Linux Box. The important sections of code is listed here for future reference.

```c
FILE *infile;
infile=fopen("output","w");

struct timeval start_time, end_time;
double elapsed;

/* Define the device */

if (argc!=3)
{
    fprintf(stderr, "Usage: ex2<dev_name><number_of_packets_to_capture>
");
    return(2);
}

dev = argv[1];
int packetcount = atoi(argv[2]);

if (dev == NULL)
{
    fprintf(stderr, "Couldn't find default device: %s\n", errbuf);
    return(2);
}
```
APPENDIX A. PACKET CAPTURING

/* Find the properties for the device */
if (pcap_lookupnet(dev, &net, &mask, errbuf) == -1)
{
    fprintf(stderr, "Couldn’t get netmask for device %s: %s
", dev, errbuf);
    net = 0;
    mask = 0;
}

/* Open the session in promiscuous mode */
handle = pcap_open_live(dev, BUFSIZ, 1, 1000, errbuf);
if (handle == NULL)
{
    fprintf(stderr, "Couldn’t open device %s: %s
", dev, errbuf);
    return(2);
}

/* Grab packets */
int j=0; // for counting packets
int i; // for counting packet contentmeofday ( &start_time, 0 );
gettimeofday(&start_time, 0);

while (j++ < packetcount)
{
    arq=arp=rrq=rrp=prp=res=bea=atim=diss=auth=deauth=mpq=no;
    packet = pcap_next(handle, &header);

    printf(" Packet %d with header length %d: %n", j, header.len);
    /* Print its length */
    gettimeofday(&end_time, 0);

    elapsed = ((double)(end_time.tv_sec - start_time.tv_sec)
            + ((double)(end_time.tv_usec - start_time.tv_usec)/1000000.0));
    curtime = time(NULL);
    localtime=localtime (&curtime);
    strcpy(t, asctime(localtime));

    fc1 = packet[start];
    fc2 = packet[sizefc1];
    d1=packet[totfc];
    d2=packet[duration1];
seq1=packet[sequ1] << bit;
seq2=packet[sequ2] >> bit;
llc1=packet[llc01];
llc2=packet[llc02];
llc3=packet[llc03];
typ1=packet[ty1];
typ2=packet[ty2];
eap1=packet[eappack1];
eap2=packet[eappack2];
eaptype=packet[eaptype];
eapid=packet[ideap];

/*
 in rest of the code we rebuild the traces suitable
 for parsing. The parsing details are presented in
 the next section. */

} /* end while */

The remaining sections of the above program is where we rebuild the packet
suitable for our timing and/or behavioral analysis. Once the packet is rebuilt
suitable for our analysis it is then passed to the next module for further processing.
Appendix B

The Parsing Program

In this section we give details of the ‘c’ program developed for parsing Ethereal traces suitable to be input to the data cube programs. The data cube requires all table values to be unsigned integers. Hence all attributes parsed from the Ethereal traces need to be converted into unsigned integers. For this purpose we used different mappings.

Table B.1 shows the mapping used to convert protocols into appropriate numerical values. Since we are interested in wireless network traces and in particular the EAP-TLS, only those protocols necessary for this purpose are considered.

Table B.2 shows the mappings used to convert all the events during the association process. Here again we have considered only those events relevant to EAP-TLS authentication process.

<table>
<thead>
<tr>
<th>ID</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IEEE802.11</td>
</tr>
<tr>
<td>1</td>
<td>EAP</td>
</tr>
<tr>
<td>2</td>
<td>TLS</td>
</tr>
<tr>
<td>3</td>
<td>PEAP</td>
</tr>
<tr>
<td>4</td>
<td>LEAP</td>
</tr>
<tr>
<td>5</td>
<td>EAPOL</td>
</tr>
</tbody>
</table>

Table B.1: Protocol Mapping
APPENDIX B. THE PARSING PROGRAM

<table>
<thead>
<tr>
<th>ID</th>
<th>Event</th>
<th>ID</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Authentication</td>
<td>29</td>
<td>EAP Request PEAP 5</td>
</tr>
<tr>
<td>1</td>
<td>Deauthentication</td>
<td>30</td>
<td>EAP Response PEAP 5</td>
</tr>
<tr>
<td>2</td>
<td>Disassociation</td>
<td>31</td>
<td>EAP Request PEAP 6</td>
</tr>
<tr>
<td>3</td>
<td>Association Request</td>
<td>32</td>
<td>EAP Response PEAP 6</td>
</tr>
<tr>
<td>4</td>
<td>Association Response</td>
<td>33</td>
<td>EAP Request PEAP 7</td>
</tr>
<tr>
<td>5</td>
<td>EAPOL Start</td>
<td>34</td>
<td>EAP Response PEAP 7</td>
</tr>
<tr>
<td>6</td>
<td>EAP Request Identity 1</td>
<td>35</td>
<td>EAP Request PEAP 8</td>
</tr>
<tr>
<td>7</td>
<td>EAP Response Identity 1</td>
<td>36</td>
<td>EAP Response PEAP 8</td>
</tr>
<tr>
<td>8</td>
<td>EAP Failure 1</td>
<td>37</td>
<td>EAP Request PEAP 9</td>
</tr>
<tr>
<td>9</td>
<td>EAP Request Identity 2</td>
<td>38</td>
<td>EAP Response PEAP 9</td>
</tr>
<tr>
<td>10</td>
<td>EAP Response Identity 2</td>
<td>39</td>
<td>EAP Success 9</td>
</tr>
<tr>
<td>11</td>
<td>EAP Request LEAP 3</td>
<td>40</td>
<td>EAP Request TLS 3</td>
</tr>
<tr>
<td>12</td>
<td>EAP Response LEAP 3</td>
<td>41</td>
<td>EAP Response TLS 3</td>
</tr>
<tr>
<td>13</td>
<td>EAP Success 4</td>
<td>42</td>
<td>EAP Request TLS 4</td>
</tr>
<tr>
<td>14</td>
<td>EAP Request LEAP 4</td>
<td>43</td>
<td>EAP Response TLS 4</td>
</tr>
<tr>
<td>15</td>
<td>EAP Response LEAP 4</td>
<td>44</td>
<td>EAP Request TLS 5</td>
</tr>
<tr>
<td>16</td>
<td>EAP Success 5</td>
<td>45</td>
<td>EAP Response TLS 5</td>
</tr>
<tr>
<td>17</td>
<td>EAP Request LEAP 5</td>
<td>46</td>
<td>EAP Failure 5</td>
</tr>
<tr>
<td>18</td>
<td>EAP Response LEAP 5</td>
<td>47</td>
<td>EAP Request TLS 6</td>
</tr>
<tr>
<td>19</td>
<td>EAP Response LEAP 6</td>
<td>48</td>
<td>EAP Response TLS 6</td>
</tr>
<tr>
<td>20</td>
<td>EAP Request PEAP 2</td>
<td>49</td>
<td>EAP Request TLS 7</td>
</tr>
<tr>
<td>21</td>
<td>EAP Response NAK 2</td>
<td>50</td>
<td>EAP Response TLS 7</td>
</tr>
<tr>
<td>22</td>
<td>EAP Response PEAP 2</td>
<td>51</td>
<td>EAP Success</td>
</tr>
<tr>
<td>23</td>
<td>EAP Request PEAP 3</td>
<td>52</td>
<td>EAPOL Key</td>
</tr>
<tr>
<td>24</td>
<td>EAP Response NAK 3</td>
<td>53</td>
<td>Data</td>
</tr>
<tr>
<td>25</td>
<td>EAP Response PEAP 3</td>
<td>54</td>
<td>QoS Data</td>
</tr>
<tr>
<td>26</td>
<td>EAP Request PEAP 4</td>
<td>55</td>
<td>ReAssociation Request</td>
</tr>
<tr>
<td>27</td>
<td>EAP Response PEAP 4</td>
<td>56</td>
<td>Null Function (No Data)</td>
</tr>
<tr>
<td>28</td>
<td>EAP Failure 4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table B.2: Event Mapping

All MAC addresses are converted by considering the last two hex values of the MAC address. As of now since we are dealing with a small number of wireless hosts this conversion is sufficient. However, in our final system we hope to introduce a hashing mechanism to convert station IDs and access point IDs into numerical values.

The Ethereal traces were first exported as a text file with only the packet summary lines. This text file was passed to the parsing program for conversion. The parsing program consists of a ‘main’ section and several functions:

- The ‘protoId’ function accepts the protocol value as a string and returns an integer value as per the mappings in Table B.1.

- the ‘eventId’ function accepts the event value as a string and returns an integer value as per the mappings in Table B.2.
• The ‘staId’ function accepts a MAC address and returns an integer value considering the last two digits of the station ID.

• The ‘manId’ function accepts a MAC address and returns an integer value considering the last two digits of the manufacturer ID.

• The ‘splitLine’ function accepts one summery line from the text file, splits it and passes the portions to the appropriate functions for conversion. Once converted it then packs the five attributes necessary for the data cube into a integer buffer and returns it to the calling function to be stored into a binary file.

Although this program is used to convert the Ethereal traces, it could eventually be used in our final program for similar purposes.
Appendix C

Sample Data

C.1 Normal Traces

Following is a list of normal traces obtained during EAP-LEAP, EAP-PEAP and EAP-TLS authentication processes. The list includes the Source MAC address, Destination MAC address, Packet Time, Sequence ID, Channel ID, Day of Month, Time of Day, Protocol and Event.

<table>
<thead>
<tr>
<th>Source MAC</th>
<th>Destination MAC</th>
<th>Time (in milliseconds)</th>
<th>Sequence ID</th>
<th>Channel ID</th>
<th>Day</th>
<th>Time of Day</th>
<th>Protocol</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>D0:F7</td>
<td>D0:F9</td>
<td>2781.163640</td>
<td>144</td>
<td>11</td>
<td>Mon</td>
<td>19-Mar-2007</td>
<td>13:53:19</td>
<td>IEEE 802.11 QoS DATA</td>
</tr>
<tr>
<td>DA:80</td>
<td>D0:F9</td>
<td>2782.204239</td>
<td>2163</td>
<td>11</td>
<td>Mon</td>
<td>19-Mar-2007</td>
<td>13:53:20</td>
<td>EAP EAP REQUEST IDENTITY 1</td>
</tr>
<tr>
<td>DA:80</td>
<td>D0:F9</td>
<td>2782.204326</td>
<td>70</td>
<td>11</td>
<td>Mon</td>
<td>19-Mar-2007</td>
<td>13:53:20</td>
<td>EAP EAP RESPONSE IDENTITY 1</td>
</tr>
<tr>
<td>D0:F9</td>
<td>DA:80</td>
<td>2782.259599</td>
<td>74</td>
<td>11</td>
<td>Mon</td>
<td>19-Mar-2007</td>
<td>13:53:20</td>
<td>EAPOL EAPOL KEY</td>
</tr>
</tbody>
</table>
RTT value of a particular event is calculated by finding the difference in “Packet Time” between a request and a response. For example if we consider “EAP REQUEST TLS 6” and the corresponding “EAP RESPONSE TLS 6” the RTT would be $(1086.517652 - 1086.553459) \approx 0.035807$ s.

C.2 Abnormal Traces

Following is a list of abnormal traces obtained during an EAP-LEAP authentication process. Here, a number of “Deauthentication” frames were injected during the authentication process. It could be seen that there are a number of ‘Deauthenticated’ frames having the same sequence number. Further, it could be seen that there are two “EAP SUCCESS” messages with sequence ID 890 and 897, which results in extended RTT for the key exchange process. The RTT for key exchange in this case happens to be $(4140.247508 - 4140.246410) = 0.109798$ s.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4140.371430</td>
<td>EAPOL KEY</td>
<td>899</td>
<td>Sun 3–Jun–2007 02:21:42</td>
<td>EAPOL KEY</td>
</tr>
<tr>
<td>4140.372019</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
<td>341</td>
<td>Sun 3–Jun–2007 02:21:42</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
</tr>
<tr>
<td>4140.454603</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
<td>341</td>
<td>Sun 3–Jun–2007 02:21:42</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
</tr>
<tr>
<td>4140.454707</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
<td>341</td>
<td>Sun 3–Jun–2007 02:21:42</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
</tr>
<tr>
<td>4140.454764</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
<td>341</td>
<td>Sun 3–Jun–2007 02:21:42</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
</tr>
<tr>
<td>4140.454827</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
<td>1397</td>
<td>Sun 3–Jun–2007 02:21:42</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
</tr>
<tr>
<td>4140.454938</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
<td>341</td>
<td>Sun 3–Jun–2007 02:21:42</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
</tr>
<tr>
<td>4140.454994</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
<td>341</td>
<td>Sun 3–Jun–2007 02:21:42</td>
<td>IEEE 802.11 DEAUTHENTICATION</td>
</tr>
</tbody>
</table>
Appendix D

Scripts

We used a number of scripts in our experiments to conduct various attacks on the wireless environment. In the following listings we provide details of these scripts together with their usage.

D.1 Deauthenticating Stations

```
#!/bin/sh
#
# Script to "Deauthenticate" all stations.
# Can specify "number" of times to execute.
# Waits for two minutes between runs.
#
if [ $# -eq 0 ]; then
    echo "Error: missing argument"
    echo "Syntax:$0 count"
    echo "Script_to_deauthenticate_all_stations"
    exit 1
fi

sta1="00:1b:11:ca:82:a2"
sta2="00:13:46:fe:d0:f7"
sta3="00:13:46:fe:d0:f9"
sta4="00:13:ce:ba:66:e9"
sta5="00:13:46:fe:f9:96"
sta6="00:13:46:fe:f8:fb"
```
n=$1
ap="00:14:f1:ad:da:80"

for (( i = 1 ; i <= $n ; i++ )); do
    /usr/local/sbin/aireplay-ng --deauth 1 -a $ap -c $sta1 ath0
    /usr/local/sbin/aireplay-ng --deauth 1 -a $ap -c $sta2 ath0
    /usr/local/sbin/aireplay-ng --deauth 1 -a $ap -c $sta3 ath0
    /usr/local/sbin/aireplay-ng --deauth 1 -a $ap -c $sta4 ath0
    /usr/local/sbin/aireplay-ng --deauth 1 -a $ap -c $sta5 ath0
    /usr/local/sbin/aireplay-ng --deauth 1 -a $ap -c $sta6 ath0
    sleep 120
done

The above script was used to "Deauthenticate" the wireless hosts repeatedly. During this time we collected the management frames associated with EAP-LEAP, PEAP and TLS authentication for timing analysis. Traces from more than fifty associations were collected and a normal profile was built.

D.2 DoS Attack

#!/bin/sh
#
# Script to inject "Deauthentication" frames.
# Waits for three seconds after the first frame
# and then injects the number of frames specified
# in the command line argument.
#
if [ $# -eq 0 ]; then
echo "Error: missing argument"
echo "Syntax: $0 staID count"
echo "Script to inject_deauthentication_frames"
exit 1
fi

if [ $1 -eq 1 ]; then
    sta="00:1b:11:ca:82:a2"
elif [ $1 -eq 2 ]; then
    sta="00:13:46:fe:d0:f7"
elif [ $1 -eq 3 ]; then
    sta="00:13:46:fe:d0:f9"
else
    echo "$Wrong_STA_value\_must\_be\_1\_to\_3"
fi

n=$2
ap="00:14:f1:ad:da:80"

/usr/local/sbin/aireplay-ng --deauth 1 -a $ap -c $sta ath0
sleep 5

for (( i = 1 ; i <= $n ; i++ )); do
    /usr/local/sbin/aireplay-ng --fakeauth 1 -e kandy -a $ap -h $sta ath0
done

The above script was used to perform DoS attacks on wireless stations. Here we first deauthenticate a wireless station and then issue fake authentication requests repeatedly.
Bibliography


[81] S. Mathews. RSN Association Traces - Capturing Tool. Technical report, School of Information and Communication Technology, Griffith University, Gold Coast, Australia, October 2006.


