



islington college

(इस्लिंग्टन कलेज)

Module Code & Module Title

CS6P05NI FINAL YEAR PROJECT

Assessment Weightage & Type

40% FYP Interim Report

Year and Semester

2023 Autumn/spring

Project Title: Real Time Threat Detection System With Threat Intelligence

Student Name: BISHWAS LIMBU

London Met ID: 20049443

College ID: NP01NT4S210077

Internal Supervisor: Mrs.Subekshya Pradhan

External Supervisor: Mr.Prabesh Hada

Assignment Due Date: 19th Dec

Assignment Submission Date: 19th Dec

Word Count: 8600

I confirm that I understand my coursework needs to be submitted online via MST under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Acknowledgement

First and foremost, I am grateful to my supervisors, Mr. Prabesh Hada and Mr. Mrs. Subekshya Pradhan, for their consistent enthusiasm and drive. I could not have completed my report without their assistance and coaching. I am also grateful to them.

Last but not least, I would like to express my appreciation towards my friends for assisting me with the project as well as providing inspirational motivation.

Abstract

This report summarizes the final year project on Realtime Threat Detection System with Threat Intelligence. This report provides a detailed overview of the developed system. The following chapters comprise this project: introduction, background/literature review, development, testing, and critical analysis, conclusion, social, legal, and ethical considerations, and future work. The introduction portion contains basic information about the issue, whereas the background/literature review section contains detailed information about the customer, technical terms, a system overview, and a review and analysis of relevant projects. The methodology chosen, the project's development design, and the execution procedure are all part of the development. The testing and critical analysis procedure includes testing the system and analysing the overall development process of the project. This project targets emerging trends in SIEM solutions with the goal of contributing to SIEM knowledge gaps by incorporating Threat intelligence. It delivers results that meet the project objectives outlined in the project proposal. To attain these goals, a thorough investigation was conducted on ELK STSCK (SIEM solution) and MISP (Threat intelligence platform) to get familiarity and insights into the project's global needs. A thorough examination was also undertaken into the detection of threats in real time, which helps to lessen the impact or allows for early response.

Contents

Chapter 1: Introduction.....	1
1.1. Problem Scenario	1
1.2. Project as a solution	12
1.3. Aim and Objectives	13
1.3.1. Aim	13
1.3.2. Objectives.....	13
1.4. Report Structure	14
1.4.1. Background	14
1.4.2. Development	14
1.4.3. Testing and Analysis	14
1.4.4. Conclusion.....	14
Chapter 2: Background	15
2.1 About the End-User	15
2.1.1. Client's Name and Description	15
2.1.2. Client's Requirements	15
2.3. Understanding the Solution	16
2.3.1. Overview of the System	16
2.3.2. Technical Terms and definition	16
2.5 Similar Project Review.....	21
2.5.1. Project 1: Detection of DoS attack and Zero Day Threat with SIEM.....	21
2.5.2. Project 2: A SIEM Architecture for Advanced Anomaly Detection	21
2.5.2. Project 3: Cyber Threat Intelligence from Honeypot Data using Elasticsearch	22
2.6. Comparison Table	23
2.7. Analysis and Conclusion of the Comparison.....	23
Chapter 3: Development	25
3.1. Selected Methodology	25
3.1.1. Evolutionary Prototyping	25
3.2 Work Breakdown Structure.....	28
3.3 Gantt Chart	29
3.5 Milestone.....	30
3.6 Survey Result	31
3.6.1 Pre-Survey Results	31
3.6.2 Post-Survey Results	32
3.7 Requirement and analysis	33
3.7.1 Feature Requirement	33
3.7.2. Software Requirements	33

3.7.3. Hardware Requirements	34
3.8 Design.....	35
3.8.1 Block Diagram of System workflow Layout	35
3.8.2 Flowchart.....	36
3.8.3 Network LAN Topology	37
3.9 Implementation	38
3.9.1 VMware Workstation 16 PRO	38
3.9.2 Windows 10	39
3.9.3 Debian 10 Linux	41
3.9.4 GNS3.....	42
3.9.5 ELKMemcached Server	46
3.9.6 Suricata NIDS (Network Intrusion Detection).....	54
3.9.7 MISP Server	55
3.9.8 Apache Web Server	56
3.9.9 ElastAlert	57
3.9.10 Telegram	59
3.9.11 Script for pulling MISP Event Data	61
Chapter 4: Testing and Analysis	66
4.1 Test Plan.....	66
4.1.1 Unit Testing, Test Plan	66
4.1.2 System Testing, Test Plan	67
4.2 Unit Testing.....	68
4.2.1 Test Case 1	68
4.2.2 Test Case 2	72
4.2.3 Test Case 3	77
4.2.4 Test case 4	83
4.2.5 Test Case 5	88
4.2.5 Test Case 6	91
4.2.7 Test Case 7	95
4.2.8 Test Case 8	103
4.2.9 Test Case 9	106
4.3 System Testing	109
4.3.1 Test Case 1	109
4.3.2 Test Case 2	123
4.3.3 Test Case 3	129
4.3.4 Test Case 4	135
4.3.5 Test Case 5	138

4.4 Critical Analysis	143
Chapter 5: Conclusion.....	145
5.1 Legal, Social and Ethical Issues	145
5.1.1 Legal Issues	145
5.1.2 Social Issues	146
5.1.3 Ethical Issues	146
5.2 Advantages	147
5.3 Limitations.....	147
6. References.....	148
7.Bibliography	153
CHAPTER 8: APPENDIX.....	155
8.1. Appendix A: PRE-SURVEY	155
8.1.1. Pre-Survey Form	155
8.1.2 Filled Pre-Survey Sample.....	161
8.1.3. Pre-Survey Result	167
8.2 APPENDIX B: POST SURVEY	174
8.2.1 Post-Survey Form	174
8.2.2 Filled Post-Survey Sample	178
8.2.3 Post-Survey Result.....	182
8.3. Appendix C: RESOURCE REQUIREMENT	188
8.3.1 Software	188
8.3.2 Hardware	191
8.4. APPENDIX D: CONSIDER METHODOLOGY FOR DEVELOPEMNT	192
8.4.1. Water Fall Methodology	192
8.4.2. Scrum	193
8.4.3. Kanban	194
8.5 APPENDIX E: Implementation Screenshot of the System	195
8.5.1 VMware Workstation 16 Pro.....	195
8.5.2 Window 10.....	204
8.5.3 Debian 10 Linux	220
8.5.4 GNS3.....	252
8.6.5 ELKMemcached Server	338
8.4.6 Suricata	359
8.4.7 MISP Server	377
8.4.8 Apache Server.....	382
8.4.9 ElastAlert	391
8.4.10. Telegram	396

8.4.11 Logstash config File	403
8.4.12 Configuration for Mikrotik router and Pfsense	413
8.6. APPENDIX F: USER FEEDBACK.....	425
8.6.1 USER FEEDBACK FROM	425
8.6.2 SAMPLE OF FILLED USER FEEDBACK FORMS	430
8.7 APPENDIX G: FUTURE WORK	435
8.7.1 FUTURE WORK.....	435
8.7.2. READING FOR FUTURE	435

Table of Table

Table 1: Unit Testing Plan.....	66
Table 2: System Testing Plan.....	67
Table 3: Test Case 1.....	68
Table 4: Test Case 2.....	72
Table 5: Test Case 3.....	77
Table 6: Test Case 4.....	83
Table 7: Test Case 5.....	88
Table 8: Test Case 6.....	91
Table 9: Test Case 7.....	95
Table 10: test case 8.....	103
Table 11: Test case 1.....	109
Table 12: Test Case 2.....	123
Table 13: Test Case 3.....	129
Table 14: Test Case 4.....	135
Table 15: Test Case 5.....	138
Table 17: Software requirements.....	190
Table 18: Hardware Requirement.....	191

Table of Figure

Figure 1: Phishing Status of Nepal Government.	2
Figure 2: Phishing Status of Commercial Companies in Nepal.	2
Figure 3: Status of Devices compromised after cyber-attack in Nepal.	3
Figure 4: Status of user compromised in government offices.	4
Figure 5: Status of user compromised on Banking Sector in Nepal	5
Figure 6: Status of user compromised in Sector link Data centre, Energy industries and hospital and medical industries.	6
Figure 7: Status of user data breach in banking sector in Nepal.	7
Figure 8: Status of user data breach in different sector in Nepal.....	8
Figure 9: Status of user data breach in different sector in Nepal.....	9
Figure 10: Types of attack techniques in 2022. (Passeri, 2022).....	10
Figure 11: Most targeted sector by cyber-attack. (Passeri, 2022)	10
Figure 12: Companies' financial losses in the United States as a result of cyber-attacks in 2022. (Statista, 2022)	11
Figure 13: Comparison Table.....	23
Figure 14: Evolutionary Prototyping. (Kris, 2018)	25
Figure 15: Work Breakdown Structure for this project.	28
Figure 16: Gantt chart for this project.	29
Figure 17: Milestone for this project.	30
Figure 18: Block Diagram of System workflow Layout.	35
Figure 19: Flowchart for Realtime threat detection with threat intelligence.	36
Figure 20: Network topology for this project.	37
Figure 21: VM ware.	38
Figure 22: Windows 10.	39
Figure 23; Windows 10 desktop.	40
Figure 24: Debian 10 Linux.	41
Figure 25: User interface of GNS3.....	43
Figure 26: Topology created in GNS3.	43
Figure 27: Imported VMs in GNS3.....	44
Figure 28: Import Mikrotik Router.	45
Figure 29: Elasticsearch.....	46
Figure 30: Accessing Elasticsearch.	47
Figure 31: Logstash installation.	48
Figure 32: Starting Logstash.	49
Figure 33: Kibana.....	50
Figure 34: Kibana UI.	51
Figure 35: Memcached.	52
Figure 36:Memcached.	53
Figure 37: Suricata service.	54
Figure 38: MISP	55
Figure 39: Apache web server.	56
Figure 40: Installing ElastAlert.	57
Figure 41: ElastAlert indices.	58
Figure 42:Example_frequency.yaml. rule.	59
Figure 43: Telegram receving log.	60
Figure 44: config.yml file.	61
Figure 45: Script for pulling misp data.	62
Figure 46: Script for pulling misp data.	63
Figure 47: script for set data to Memcached.....	64
Figure 48: ruby script filter used in Logstash.	65

Figure 49: Starting filebeat service and verifying its status in Suricata VM	69
Figure 50: Verifying status of Logstash service.	70
Figure 51: Verifying Logstash was listening in port 5044 allocated for Suricata log.....	71
Figure 52: Filebeat log was seen in index management.	71
Figure 53: Starting filebeat service and verifying its status in Apache web server.	73
Figure 54: Verifying status of Logstash service.	74
Figure 55: Verifying Logstash was listening in port 5042 allocated for Apache log.....	75
Figure 56: Apache log was seen in index management.	76
Figure 57: Starting Winlogbeat.	78
Figure 58: Winlogbeat started.	79
Figure 59: Verifying status of Logstash service.	80
Figure 60: Verifying Logstash was listening in port 5045 allocated for winlogbeat.	81
Figure 61: winlogbeat log was seen in index management.	82
Figure 62: Starting auditbeat.....	83
Figure 63: Auditbeat was started.	84
Figure 64: Verifying status of Logstash service.	85
Figure 65: Verifying Logstash was listening in port 5043 allocated for auditbeat.	86
Figure 66: Auditbeat log was seen in index management.	87
Figure 67: USB drive connecting to windows 10 VM.	88
Figure 68: Windows event log generated when USB device was plugged in.....	89
Figure 69: Win log event id 2003 present in Kibana discover dashboard which indicated USB device was plugged in.....	90
Figure 70: Win log event id 2102 present in Kibana discover dashboard which indicated USB device was unplugged.....	90
Figure 71: File was created.	91
Figure 72: bishwas.txt file creation alert in discover dashboard.	92
Figure 73: File was deleted.	93
Figure 74: bishwas.txt file deletion alert in discover dashboard.	94
Figure 75: Custom created event in MISP with source IPs.	96
Figure 76: Custom created event in MISP with SHA256 hashes.	97
Figure 77: Custom created event in MISP with destination IPs.	98
Figure 78: Custom created event in MISP with destination Domain address.	99
Figure 79: Executing python script.....	99
Figure 80: Fetching hash sha256 list from MISP server.	100
Figure 81: Fetching source Ips list from MISP server.	100
Figure 82: Fetching domain list from MISP server.....	101
Figure 83: Fetching destination IP list from MISP server.....	101
Figure 84: Performing telnet connection to Memcached server and verifying IOCs list.	102
Figure 85: Facebook domain site was visited.	104
Figure 86: An alert was display after match was found.	105
Figure 87: Editing "example_frequency.yaml" for sending alert message to telegram where bot token and room id for telegram was used.	107
Figure 88: Alert messaged passed by telegram to user.	108
Figure 89: filebeat log agent for Suricata VM in working and running state.	110
Figure 90: Filebeat log agent for Apache web server in running state.	111
Figure 91: Filebeat service logs	111
Figure 92: winlogbeat log agent for windows 10 VM in working and running state.	112
Figure 93: Winlogbeat service logs	113
Figure 94: Auditbeat log agent for Suricata VM in working and running state.....	114
Figure 95: Auditbeat service logs.....	115
Figure 96: Logstash service is working and running.	116
Figure 97: Logstash service is listening to all the ports assign to log agents.	117

Figure 98: All logs are centralized in ELK server.....	118
Figure 99: Winlogbeat logs are displayed.....	119
Figure 100: Filebeat logs for suricate are displayed.....	120
Figure 101: Auditbeat logs are displayed.....	121
Figure 102: filebeat logs for Apache web server are displayed.....	122
Figure 103: Hash value of executable file.....	123
Figure 104: Malware.exe file was executed. note (this was torrent file renamed to malware.exe)	124
Figure 105: Hash value got hit in the MISP event.....	125
Figure 106: Visiting to "misp.url" link for more info.....	126
Figure 107: Hash value of malware.exe found here.....	127
Figure 108: MISP Event Alert in Telegram.....	128
Figure 109: New Text Document.txt file created.....	129
Figure 110: Creation Event was recorded in ELK stack.....	130
Figure 111: Telegram alert for creation of file in system.....	131
Figure 112: File was Deleted.....	132
Figure 113: Deletion event was recorded in EKL stack.....	133
Figure 114: Telegram alert after deletion of file.....	134
Figure 115: USB device plugging in to win 10.....	135
Figure 116: An alert event with id 2003 was triggered in ELK stack dashboard.....	136
Figure 117: Telegram message was delivered when USB was plugged in to system.....	137
Figure 118: Elasticsearch was active and running.....	138
Figure 119: Testing Elasticsearch configuration.....	139
Figure 120: Kibana services was active and running.....	140
Figure 121: Kibana user interface.....	141
Figure 122: Logstash was active and running.....	142
Figure 123: Pre-Survey form 1.....	155
Figure 124: Pre-Survey form 2.....	156
Figure 125: Pre-Survey form 3.....	157
Figure 126: Pre-Survey form 4.....	158
Figure 127: Pre-Survey form 5.....	159
Figure 128: Pre-Survey form 6.....	160
Figure 129: Filled Pre-Survey Sample 1.....	161
Figure 130: Filled Pre-Survey Sample 2.....	162
Figure 131: Filled Pre-Survey Sample 3.....	163
Figure 132: Filled Pre-Survey Sample 4.....	164
Figure 133: Filled Pre-Survey Sample 5.....	165
Figure 134: Filled Pre-Survey Sample 6.....	166
Figure 135: Pre-survey Result 1.....	167
Figure 136: Pre-survey Result 2.....	167
Figure 137: Pre-survey Result 3.....	168
Figure 138: Pre-survey Result 4.....	168
Figure 139: Pre-survey Result 5.....	169
Figure 140: Pre-survey Result 6.....	169
Figure 141: Pre-survey Result 7.....	170
Figure 142: Pre-survey Result 8.....	170
Figure 143: Pre-survey Result 9.....	171
Figure 144: Pre-survey Result 10.....	171
Figure 145: Pre-survey Result 11.....	172
Figure 146: Pre-survey Result 12.....	172
Figure 147: Pre-survey Result 13.....	172
Figure 148: Pre-survey Result 14.....	173
Figure 149: Post survey question form 1.....	174

Figure 150: Post survey question form 2	175
Figure 151: Post survey question form 3	176
Figure 152: Post survey question form 5	177
Figure 153: Post survey question form 6	177
Figure 154: Post survey question form 7	178
Figure 155: Filled Form of Post survey 1	178
Figure 156: Filled Form of Post survey 2	179
Figure 157: Filled Form of Post survey 3	179
Figure 158: Filled Form of Post survey 4	180
Figure 159: Filled Form of Post survey 5	180
Figure 160: Filled Form of Post survey 6	181
Figure 161: Filled Form of Post survey 7	181
Figure 162: Post survey Result 1	182
Figure 163: Post survey Result 2	182
Figure 164: Post survey Result 3	183
Figure 165: Post survey Result 4	183
Figure 166: Post survey Result 5	184
Figure 167:Post survey Result 6	184
Figure 168: Post survey Result 7	185
Figure 169: Post survey Result 8	185
Figure 170: Post survey Result 9	186
Figure 171: Post survey Result 10	186
Figure 172: Post survey Result 11	187
Figure 173: Water Fall Methodology. (Adobe Communications Team, 2022).....	192
Figure 174: Scrum Methodology. (scrum.org, 2022)	193
Figure 175: Kanban Methodology. (kissflow.com, 2022).....	194
Figure 176: Installation of VM workstation 1	195
Figure 177: Installation of VM workstation 2	196
Figure 178: Installation of VM workstation 3	197
Figure 179: Installation of VM workstation 4	198
Figure 180: Installation of VM workstation 5	199
Figure 181: Installation of VM workstation 6	200
Figure 182: Installation of VM workstation 7	201
Figure 183: Installation of VM workstation 8	202
Figure 184:Installation of VM workstation 9	203
Figure 185: Installing windows 10 VM image 1	204
Figure 186: Installing windows 10 VM image 2	204
Figure 187: Installing windows 10 VM image 3	205
Figure 188: Installing windows 10 VM image 4	206
Figure 189: Installing windows 10 VM image 5	206
Figure 190: Installing winlogbeat ..	207
Figure 191: Configuring winlogbeat yml file 1	208
Figure 192: Configuring winlogbeat yml file 2	209
Figure 193: Configuring winlogbeat yml file 3	210
Figure 194: Starting winlogbeat service.....	211
Figure 195: Winlogbeat started.....	212
Figure 196: Installing auditbeat in win 10.....	213
Figure 197: Configuring auditbeat yml file 1	214
Figure 198: Configuring auditbeat yml file 2	215
Figure 199: Starting auditbeat service	216
Figure 200: Auditbeat service started 1	217
Figure 201: Auditbeat service started 2	218

Figure 202: Winlogbeat Dashboard.	219
Figure 203: Installing Debian 10 on VM 1.	220
Figure 204: Installing Debian 10 on VM 2.	221
Figure 205: Installing Debian 10 on VM 3.	222
Figure 206: Installing Debian 10 on VM 4.	223
Figure 207: Installing Debian 10 on VM 5.	224
Figure 208: Installing Debian 10 on VM 6.	225
Figure 209: Installing Debian 10 on VM 7.	226
Figure 210: Installing Debian 10 on VM 8.	227
Figure 211: Installing Debian 10 on VM 9.	228
Figure 212: Installing Debian 10 on VM 10.	229
Figure 213: Installing Debian 10 on VM 11.	230
Figure 214: Installing Debian 10 on VM 12.	231
Figure 215: Installing Debian 10 on VM 13.	232
Figure 216: Installing Debian 10 on VM 14.	233
Figure 217: Installing Debian 10 on VM 15.	234
Figure 218: Installing Debian 10 on VM 16.	235
Figure 219: Installing Debian 10 on VM 17.	236
Figure 220: Installing Debian 10 on VM 18.	237
Figure 221: Installing Debian 10 on VM 19.	238
Figure 222: Installing Debian 10 on VM 20.	239
Figure 223: Installing Debian 10 on VM 21.	240
Figure 224: Installing Debian 10 on VM 22.	241
Figure 225: Installing Debian 10 on VM 23.	242
Figure 226: Installing Debian 10 on VM 24.	243
Figure 227: Installing Debian 10 on VM 25.	244
Figure 228: Installing Debian 10 on VM 26.	245
Figure 229: Installing Debian 10 on VM 27.	246
Figure 230: Installing Debian 10 on VM 28.	247
Figure 231: Installing Debian 10 on VM 29.	248
Figure 232: Installing Debian 10 on VM 30.	249
Figure 233: Installing Debian 10 on VM 31.	250
Figure 234: Installing Debian 10 on VM 32.	251
Figure 235: Installing GNS3 1.	252
Figure 236: Installing GNS3 2.	253
Figure 237: Installing GNS3 3.	254
Figure 238: Installing GNS3 5.	255
Figure 239: Installing GNS3 6.	256
Figure 240: Installing GNS3 7.	257
Figure 241: Installing GNS3 8.	258
Figure 242: Installing GNS3 9.	259
Figure 243: Installing GNS3 11.	260
Figure 244: Installing GNS3 12.	261
Figure 245: Installing GNS3 13.	262
Figure 246: Installing GNS3 14.	263
Figure 247: Installing GNS3 15.	264
Figure 248: Installing GNS3 16.	265
Figure 249: Installing GNS3 17.	266
Figure 250: Installing GNS3 18.	267
Figure 251: Installing GNS3 19.	268
Figure 252: Installing GNS3 21.	269
Figure 253: Installing GNS3 22.	270

Figure 254: Installing GNS3 23	271
Figure 255: Installing GNS3 24	272
Figure 256: Installing GNS3 25	273
Figure 257: Installing GNS3 26	274
Figure 258: configuration for cloud interface 1	275
Figure 259: configuration for cloud interface 2	276
Figure 260: configuration for cloud interface 3	276
Figure 261: configuration for cloud interface 4	277
Figure 262: configuration for cloud interface 5	278
Figure 263: configuration for cloud interface 6	279
Figure 264: configuration for cloud interface 7	280
Figure 265: configuration for cloud interface 8	281
Figure 266: configuration for cloud interface 9	282
Figure 267: configuration for cloud interface 10	283
Figure 268: Importing Mikrotik router 1	284
Figure 269: Importing Mikrotik router 2	285
Figure 270: Importing Mikrotik router 3	285
Figure 271: Importing Mikrotik router 4	286
Figure 272: Importing Mikrotik router 5	287
Figure 273: Importing Mikrotik router 6	288
Figure 274: Importing Mikrotik router 7	288
Figure 275: Importing Mikrotik router 8	289
Figure 276: Importing Mikrotik router 9	289
Figure 277: Importing Mikrotik router 10	290
Figure 278: Importing Mikrotik router 11	290
Figure 279: Importing Mikrotik router 12	291
Figure 280: Importing Pfsense 1	292
Figure 281: Importing Pfsense 2	293
Figure 282: Importing Pfsense 3	294
Figure 283: Importing Pfsense 4	295
Figure 284: Importing Pfsense 5	296
Figure 285: Importing Pfsense 6	297
Figure 286: Importing Pfsense 7	298
Figure 287: Importing Pfsense 8	299
Figure 288: Importing Pfsense 9	300
Figure 289: Importing Pfsense 10	301
Figure 290: Importing Pfsense 13	302
Figure 291: Importing Pfsense 14	303
Figure 292: Importing Pfsense 15	304
Figure 293: Importing ELKMemcached server 1	305
Figure 294: Importing ELKMemcached server 2	306
Figure 295: Importing ELKMemcached server 3	307
Figure 296: Importing ELKMemcached server 4	308
Figure 297: Importing ELKMemcached server 5	309
Figure 298: Importing ELKMemcached server 6	310
Figure 299: Importing ELKMemcached server 7	311
Figure 300: Importing suricata VM server 1	312
Figure 301: Importing suricata VM server 2	313
Figure 302: Importing suricata VM server 3	314
Figure 303: Importing suricata VM server 4	315
Figure 304: Importing suricata VM server 5	316
Figure 305: Importing suricata VM server 6	317

Figure 306: Importing suricata VM server 7	318
Figure 307: Importing MISP VM server 1.....	319
Figure 308: Importing MISP VM server 2.....	320
Figure 309: Importing MISP VM server 3.....	321
Figure 310: Importing MISP VM server 4.....	322
Figure 311: Importing MISP VM server 5.....	323
Figure 312: Importing MISP VM server 6.....	324
Figure 313: Importing MISP VM server 7.....	325
Figure 314: Importing MISP VM server 8.....	326
Figure 315: Import Apache server 1	327
Figure 316: Import Apache server 2	328
Figure 317: Import Apache server 3	329
Figure 318: Import Apache server 4	330
Figure 319: Import Apache server 7	331
Figure 320: Import Apache server 8	332
Figure 321: Import Apache server 9	333
Figure 322: Importing windows 10 1	334
Figure 323: Importing windows 10 2	335
Figure 324: Importing windows 10 3	336
Figure 325: Importing windows 10 4	337
Figure 326: Installing Elasticsearch 1	338
Figure 327: Configuration of Elasticsearch yml 1	339
Figure 328: Configuration of Elasticsearch yml 2	340
Figure 329: Configuration of Elasticsearch yml 3	341
Figure 330: Configuration of Elasticsearch yml 4	342
Figure 331: Configuration of Elasticsearch yml 6	343
Figure 332: Configuration of jvm.options	343
Figure 333: Configuration of jvm.options 1	344
Figure 334: Installation complete	345
Figure 335: setting password 1	346
Figure 336: setting password 2	347
Figure 337: Accessing Elasticsearch	348
Figure 338: Installing Kibana	349
Figure 339: configuring yml files for kibana 1	350
Figure 340: configuring yml files for kibana 2	350
Figure 341: configuring yml files for kibana 3	351
Figure 342: installation complete	352
Figure 343: Dashboard of Kibana	353
Figure 344: Installing dependencies 1	354
Figure 345: Installing dependencies 2	355
Figure 346: Installing dependencies 3	356
Figure 347: Installing Memcached	357
Figure 348: Started Memcached	358
Figure 349: Installing suricata 1	359
Figure 350: Installing suricata 2	360
Figure 351: Installing Suricata 3	361
Figure 352: Installing Suricata 4	362
Figure 353: Installing suricata 5	363
Figure 354: Installing suricata 6	364
Figure 355: Configuration of suricata yml	365
Figure 356: Configuration of suricata.yml	366
Figure 357: Configuration of suricata.yml	367

Figure 358: Configuration of suricata.yml	368
Figure 359: Configuration of suricata.yml	369
Figure 360: Configuration of suricata .yml	370
Figure 361: Installing and configuring filebeat for Suricata.....	371
Figure 362: Installing and configuring filebeat for Suricata.....	372
Figure 363; Installing and configuring filebeat for Suricata.....	373
Figure 364: Installing and configuring filebeat for Suricata.....	374
Figure 365: Installing and configuring filebeat for Suricata.....	375
Figure 366: Installing and configuring filebeat for Suricata.....	376
Figure 367: Installing MISP	377
Figure 368: Installing MISP	378
Figure 369: Installing MISP	379
Figure 370: MISP web interface.....	380
Figure 371: Event in MISP	381
Figure 372: Installing Apache server.	382
Figure 373: Installing Apache server.	383
Figure 374: Installing Apache server.	384
Figure 375: Apache server web page.	385
Figure 376: Installing and configuring filebeat for Apache web server.	386
Figure 377: Installing and configuring filebeat for Apache web server.	387
Figure 378: Installing and configuring filebeat for Apache web server.	388
Figure 379: Installing and configuring filebeat for Apache web server.	389
Figure 380: Installing and configuring filebeat for Apache web server.	390
Figure 381: Installing dependencies.	391
Figure 382: Installing dependencies.	392
Figure 383: Installing dependencies.	393
Figure 384: creating index in Elasticsearch.	394
Figure 385: Index created in Elasticsearch.	395
Figure 386: creating bot..	396
Figure 387: Generated token and API link. For telegram.	397
Figure 388: Chat room created.	398
Figure 389: Configured yml file for alert generating in ElastAlert.	399
Figure 390: Testing that yml file to generate alert.....	400
Figure 391: Matched hit in test round.	401
Figure 392: Log was sent to telegram.....	402
Figure 393: Logstash pipeline.....	403
Figure 394: Logstash conf file for apache server 1.....	404
Figure 395: Logstash conf file for apache server 2.....	405
Figure 396: Logstash conf file for Suricata VM.	406
Figure 397: Logstash conf file for Suricata VM.	407
Figure 398: Logstash conf file for windows 10 VM(auditbeat).	408
Figure 399: Logstash conf file for windows 10 VM(auditbeat).	409
Figure 400: Logstash conf file for windows 10 VM(winlogbeat).....	410
Figure 401: Logstash conf file for windows 10 VM(winlogbeat).....	411
Figure 402: Logstash conf file for windows 10 VM(winlogbeat).....	412
Figure 403: Configuring Mikrotik router.....	413
Figure 404: Configuring Mikrotik router.....	413
Figure 405: Configuring Pfsense.	414
Figure 406: Configuring Pfsense.	415
Figure 407: Configuring Pfsense.	416
Figure 408: Configuring Pfsense.	417
Figure 409: Configuring Pfsense.	418

Figure 410:Configuring Pfsense	419
Figure 411: Configuring Pfsense	420
Figure 412: configuration of interfaces.....	421
Figure 413: Configuring firewall rules.	422
Figure 414: Configuring firewall rules.	423
Figure 415: Configuring firewall rules.	424
Figure 416: Configuring firewall rules.	424
Figure 417: User Feedback form 1.	425
Figure 418: User Feedback form 2.	426
Figure 419: User Feedback form 3.	427
Figure 420: User Feedback form 4.	427
Figure 421: User Feedback form 5.	428
Figure 422: User Feedback form 6.	429
Figure 423: User Feedback form 7.	429
Figure 424: User Filled Feedback form 1.....	430
Figure 425: User Filled Feedback form 2.....	431
Figure 426: User Filled Feedback form 3.....	432
Figure 427: User Filled Feedback form 4.....	432
Figure 428:User Filled Feedback form 5.....	433
Figure 429: User Filled Feedback form 6.....	434
Figure 430: User Filled Feedback form 7.....	434

Chapter 1: Introduction

Today, technology developments and falling prices have made the globe more linked via the internet than ever before, resulting in unparalleled opportunity, innovation, and growth. The COVID-19 pandemic has accelerated this tendency, but it may represent the start of a long-term structural change. Cyberspace is impacting economics, security, food distribution, healthcare, and transportation, as well as people's lives, work, and communication. Cyberspace is now critical to people's future wealth and security.

The size and speed of this upheaval, which often outpaces societal norms, legislation, and democratic processes, is creating unprecedented complexity, instability, and insecurity. In the last year, ransomware, commercial spyware, malware, DDoS assaults, and other cyberattacks have crippled hospitals, oil refineries, companies, and schools. If recognized early, such cyber threats may have been mitigated or prevented. (gov.uk, 2022)

Here, comes the role of SIEM solution where it helps to detect those types of threat in early stage. By using different logs collecting, filtering, enriching, parsing, analysing and visualizing techniques, this system helps to detect early threats in network or end devices that will help an organization to take action need as soon as they detect threat. This will automatically help an organization to prevent and minimize impact of such cyber threat which can cause great loss financially.

1.1. Problem Scenario

In Nepal, with the rapid growth of uses of technology in commercial and industrial area, governmental offices, etc also has led such organizations at high risk from cyber-attack like phishing, DDoS, malware, ransomware etc. Many governmental offices, banks, ISP company and common people are affected by the cyber-attacks in Nepal.

According the “Threat Report 2022” published by “Vairav Technology”, many government offices, agencies, banking institute, Ecommerce industries, Energy Industry, Telecom and ISP Sectors, Fintech industries, Hotels and Tourism, Pharmaceutical Industries etc and other many organisations were affected by data breach. Many devices and users of organization were compromised, and organization were also affected by phishing attack.

Phishing Status of Nepal Government

Total Phishing Pages: 20

Total Affected Websites: 11

Figure 1: Phishing Status of Nepal Government.

Phishing Status of Commercial Companies in Nepal

Total Phishing Pages: 100

Total Affected Websites: 35

Listings of the affected Commercial Company Websites by domain:

Phishing Themes carried out:

- BMO Bank of Montreal Online Banking
- Onedrive
- XFINITY
- Outlook
- We transfer
- DHL
- Chase Bank
- Bank of America
- SunTr

Figure 2: Phishing Status of Commercial Companies in Nepal.

Top 5 ISPs as per Compromised Devices

Internet Service Provider Name	No. of Compromised Devices
Nepal Telecommunications Corporation	502
WorldLink Communications Ltd	342
Subisu Cablenet (Pvt) Ltd	148
Websurfer Nepal Communication System Pvt. Ltd.	112
Vianet Communications Pvt. Ltd.	108

Top Government Organization as per Compromised Server

Government Organization	ASN	No. of Compromised Devices
National Information Technology Center: IT Agency of Government of Nepal	AS45353	14
Department of Information Technology, Government of Nepal	AS131341	3

Figure 3: Status of Devices compromised after cyber-attack in Nepal.

Top 5 adversely affected Department and Government Agencies...

Department Name	Total Number	Total Internal User (Employee)	Total External User (Consumer)
Inland Revenue Department	1594	7	1587
Department of National ID and Civil Registration	368	0	368
Department Of Revenue Investigation	244	0	244
Financial Comptroller General Office	216	0	216
Department of Foreign Employment	78	0	78

Figure 4: Status of user compromised in government offices.

Status of Banking and Financial Institutions

Description	Total Number
Compromised BFIs Users	2685
Affected BFIs	36
Internal Users (Staff)	35
External Users (Consumer)	2650
Affected Grade A Banks	24

Affected Grade B Banks	6
Affected Grade C Bank	1
Affected Grade D Banks	5

Figure 5: Status of user compromised on Banking Sector in Nepal

Status of Data Centers

Description	Total Number
Compromised Users	59
Internal Users (Staff)	7
External Users (Consumer)	52
Affected Organizations	4

Status of Energy Industries

Description	Total Number
Compromised Users	5
Internal Users (Staff)	1
External Users (Consumer)	3
Affected Organizations	4

Status of Hospitals and Medical Industries

Description	Total Number
Compromised Users	143
Internal Users (Staff)	2
External Users (Consumer)	141
Affected Organizations	18

Figure 6: Status of user compromised in Sector link Data centre, Energy industries and hospital and medical industries.

User Data Breached Status of Banking and Financial Institutions

Total Affected Employees: 2600

Total Affected Banking and Finance Institutions: 40

Total Affected Grade A Banks Employees: 2449

Total Affected Grade A Banks: 20

Total Affected Grade B Banks Employees: 98

Total Affected Grade B Banks: 9

Total Affected Grade C Banks Employees: 48

Total Affected Grade C Banks: 7

Total Affected Grade D Banks Employees: 5

Total Affected Grade D Banks: 4

Figure 7: Status of user data breach in banking sector in Nepal.

User Data Breached Status of Ecommerce Industries

Total Affected Ecommerce Employee: 98

Total Affected Ecommerce Organization: 24

User Data Breached Status of Telecom and ISP Sectors

Total Affected Telecom and ISP Users: 71,254

Total Affected Telecom and ISP Organizations: 20

Total Affected ISP Users: 66,538

Total Affected ISP Organizations: 17

Total Affected Telecom Users: 4,716

Total Affected Telecom Organizations: 3

The Status of User Data Breach in Fintech Industries

Total Affected Employees: 108

Total Affected Organizations: 9

Figure 8: Status of user data breach in different sector in Nepal.

The Status of User Data Breach in Energy Industry

Total Affected Employees: 187

Total Affected Organizations: 8

Company Name	No. of Users
Butwal power company	96
Himal power limited	75
Chilime hydropower company limited	5
Madi Power Pvt Ltd	5

The Status of User Data Breach in Data Centers

Total Affected Employees: 53

Total Affected Organizations: 6

The Status of User Data Breach in University and Colleges

Total Affected Employees: 2396

Total Affected Organizations: 65

The Status of User Data Breach in Hospitals and Medical Sectors

Total Affected Employees: 1131

Total Affected Organizations: 24

Figure 9: Status of user data breach in different sector in Nepal.

In World, With the usage of the internet and internet-based systems, there is also a significant increase in the danger to a company's assets such as data, network system, server, and application from cyber-attacks such as ransomware, phishing, malware assault, dos attack, and so on. Cybercriminals have employed a variety of advanced techniques and technologies to exploit flaws in computing systems. The percentage of successful attacks that compromise an organization has been also seen growing with the time.

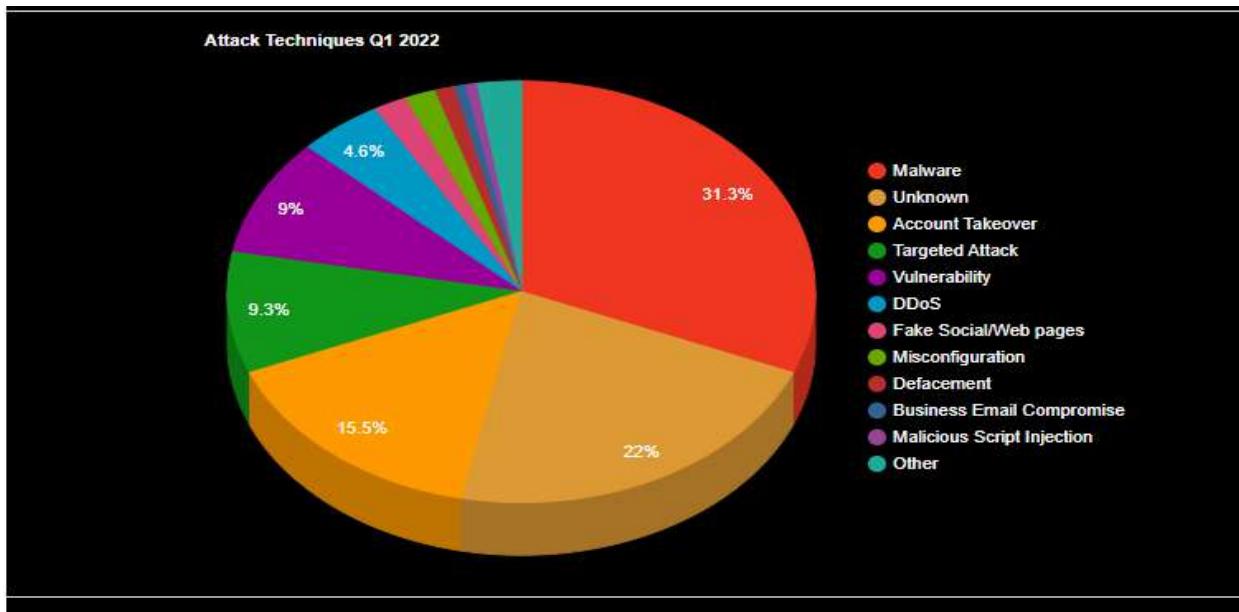


Figure 10: Types of attack techniques in 2022. (Passeri, 2022)

In above figure, shows that “malware” attack was the mostly used cyber-attack to harm any organization or individual for their gain.

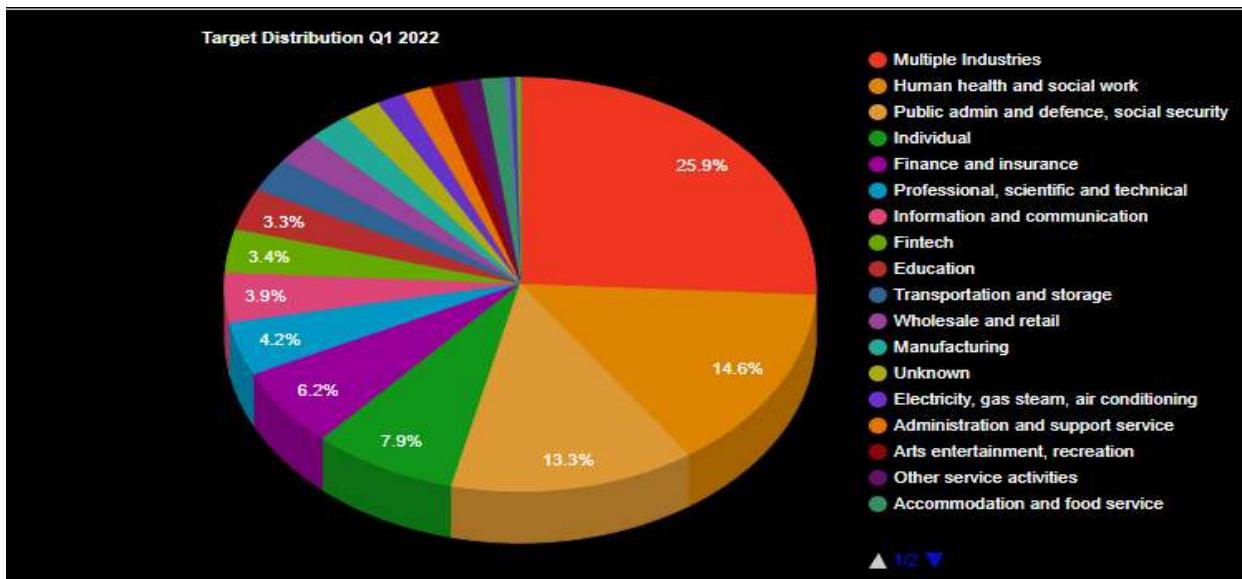


Figure 11: Most targeted sector by cyber-attack. (Passeri, 2022)

In above figure, shows that in 2022 organization like industrial and health sector, public admin and defence, social was the most targeted by cyber-attack. This can deal a great loss to those organization.

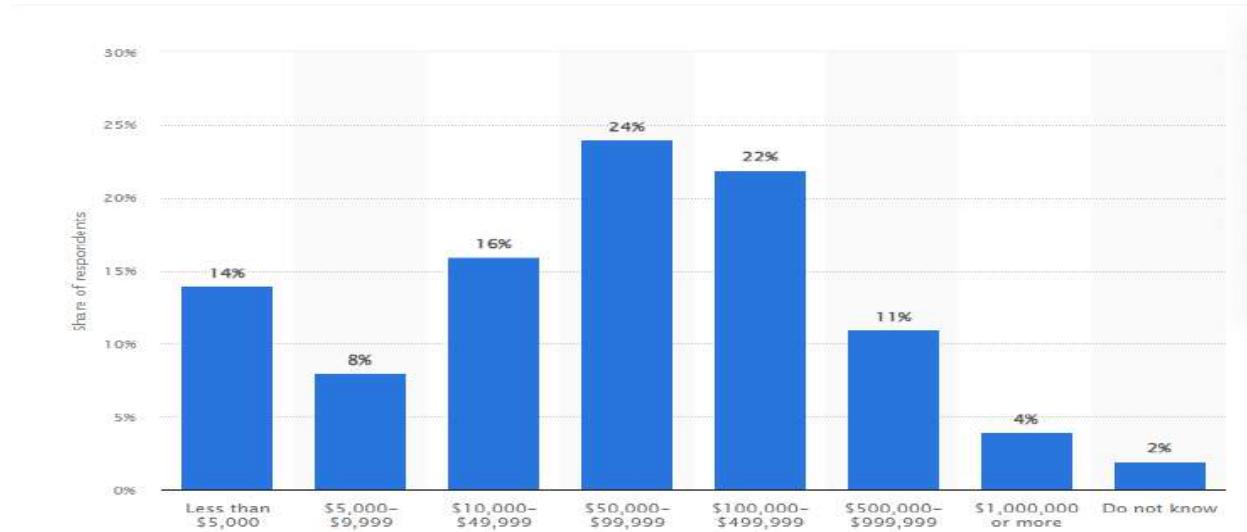


Figure 12: Companies' financial losses in the United States as a result of cyber-attacks in 2022. (Statista, 2022)

A study of IT decision-makers in the United States showed that about a quarter of organizations that have encountered a cyber-attack had lost between \$50,000 and \$100,000, according to a 2022 report. Another 22% of those surveyed reported losses ranging from \$100,000 to \$499,999 USD. A cyberattack has cost 4% of businesses more than a million dollars. (Statista, 2022)

1.2. Project as a solution

Real Time Threat Detection System with Threat Intelligence is security solution for any organization which monitors real time traffic of data flow, changes in systems, malicious activities, and behaviour of different types of systems and devices used by an organization. This purpose system will collect logs from different network devices, end devices, IDS system etc and store in centralized storage where those logs will be analysed and compared with help of a Threat Intel Platform for known threat and rule-based detection will be used for unknown threat to produce an alert when any malicious or intrusion activities are detected. This system helps to provide in depth detail about our system activities which help to keep track of any activities and detect any malicious behaviour and activities if found and prevent cyber threats in early stages. A notification in Telegram app will be also sent by this system if any alert is generated. This system can help to identify any cyber threat activities and safeguard any organization's assets from those threats that can cause negative affect to an organization. It can help to monitor in real time and track activities going on our system every day and provide security to our system.

1.3. Aim and Objectives

1.3.1. Aim

The main aim of this project is to develop real time threat detection system with integration of threat Intelligence that will detect and generate alert for various intrusion and malicious activities on host or any network so that any organization can efficiently respond to such threats to minimize effect of cyber-attack or block such threat.

1.3.2. Objectives

The following objectives were completed to achieve aim:

- Deep study on the various topic like SEIM, VM ware, GNS3, Networking, firewall, notification system, Threat Intelligence in various sources like journals, articles, reports, books blogs etc.
- Gained understanding and knowledge of programming language like python.
- Develop knowledge of proper understanding of APIs and their uses.
- Develop proper understanding of cache memory and its uses in larger system for faster data access and store.
- Gaining Knowledge and understandings of intrusion detection system for host and network.

1.4. Report Structure

1.4.1. Background

In this chapter, it contains information on about the End user, client description and an overview of the client's needs. It also gives information on the project's understanding and technical terms related to this project. The research for comparable projects is shown in this chapter, along with a comparison table highlighting the salient traits and features of the project "Real Time Threat Detection System With Threat Intelligence".

1.4.2. Development

The project's development phases are outlined in this chapter. It covers the methodologies that were under consideration, the methodology that was chosen, analysis of the methodology that was chosen, requirement analysis, design, and development implementation. Additionally, it compromises the analysis of pre- and post-survey results.

1.4.3. Testing and Analysis

Test strategies and test cases are included in the Testing and Analysis phase. There are two different kinds of testing scenarios: unit tests and machine tests. The topic of critical analysis is covered in depth last.

1.4.4. Conclusion

This chapter summarizes the report and discusses legal, ethical, and social concerns. This chapter also addresses system's benefits, limitations, and potential for the future. Overall, it provides a detailed account of the system and its outcomes.

Chapter 2: Background

2.1 About the End-User

This project is design for implementation in an organization as well as for personal purpose as security system where any end user can monitor, detect, and react to the threat which can include IT admin, security analysts and other IT professional with the responsibility of securing their data and network. . This can be implemented in financial institute, government server office, college, ecommerce industry etc to detect new emerging threat. This can be also used by anyone who are interested in network security domain and wants to secure their system and network.

2.1.1. Client's Name and Description

Name of the Client: Everest Bank LIMITED

Description of client: Everest Bank Limited is client for the project, which is in Lamahi, Dang. It is a commercial banking organization providing banking related services to many people. Everest Bank Limited has shown interest in this project and agreed to be client for this project as they found this project very essential to provide security to their IT system due to growing security risk from cyber-attack. They are willing to cooperate with the suggestions and by providing the necessary requirements.

2.1.2. Client's Requirements

Client's requirements are outline below: -

- Proposed system must provide Realtime monitoring features for all devices, network traffic in a LAN network.
- Proposed system must be able to detect malicious activities and intrusion in network as well as in host.
- Message alert function must be in the proposed system if any threats are detected.

2.3. Understanding the Solution

2.3.1. Overview of the System

In this project, Real Time Threat Detection System with threat Intelligence is implemented in GNS3 environment with network topology that have pfsense firewall where its wan interface is configured for port mirroring and forward that mirrored network traffic to Suricata which will act NIDS. It also have a DMZ zone where web server is setup. Then logs from all above-mentioned devices are forwarded to Logstash using log agents, where those logs are collected, filtered, enrich, lookup function with Memcached server are preformed and it is stored and indexed in Elasticsearch and with the help of Kibana logs are discovered and visualized in dashboard which provides detail information on logs. Similarly, MISP as CTI or threat intelligence platform is integrated for better correlation for new and existing threat. Where MISP server maintains the database for threat intel data. Likewise, Memcached is integrated as temporary storage area for holding intel IoCs data coming from MISP which is pulled in Memcached with python script using API and authentication token of MISP server and it helps to providing fast access and correlation capabilities to Logstash to detect threats. Threats are detected when Logstash lookup function finds matching data of own with Memcached threat intel data and different rules are written to properly generate alert upon detecting threat. Using ElastAlert which query is made with ELK stack for triggering alert and sending alert message to telegram.

2.3.2. Technical Terms and definition

MISP

A threat intelligence platform that not only stores, shares, collaborates on indicators of compromise of targeted attacks, threat intelligence, financial fraud information, vulnerability information, or even counter-terrorism information, but also uses the indicators and information to detect and prevent attacks, frauds, or threats against ICT infrastructures, organizations, or people. (misp-project.org, 2022)

ELK Stack

Three open-source projects Elasticsearch, Logstash, and Kibana are referred to together as "ELK" stack. A search and analytics engine is Elasticsearch. A server-side data processing pipeline called Logstash receives data from numerous sources at once, alters it, and then delivers the result to a "stash" like Elasticsearch. Users of Elasticsearch can view data using Kibana's visualization and dashboard. (Elasticsearch B.V., 2023)

Logging Agent or Log Agent

A program which reads logs from one place and transfers them to another is known as a logging agent, sometimes known as a log shipper. They are frequently used to download individual events to a server for processing and centralization and read log files kept on a system. For software running on the host, such as applications, services, and operating system components, they essentially serve as log funnels. (Mezmo Inc, 2022)

Log and Event Log

An event that happened at a specific time and may have contextual metadata is recorded in a file called a log file. The events that take place within a system, such as transactions, mistakes, and intrusions, are recorded historically in log files. (Sharif, 2023)

Event log is specific file where incident is often documented where event or incident are caused by operating system, network, servers, firewall, ant virus software, hardware etc. (Sharif, 2023)

API

API stands for application programming interface, which is a software mediator that allows two apps to communicate with one another that provides an easy way to retrieve and distribute data within and across companies. (Frye, 2023)

Real-time threat detection

Real-time threat detection a method that improves architecture security by detecting malicious activity in real time which threatens IT infrastructure, website security, and data confidentiality. (Rock Content, 2021)

Threat Intel

Threat intelligence or threat intel is data collected about a potential danger or the process of acquiring, processing, and analysing that data to better understand threats. (Kaspersky, 2023)

Network Intrusion Detection System and Firewall

Network intrusion detection systems (NIDS) is system which are installed at strategic points throughout the network to examine traffic from all network devices. It monitors all traffic on the subnet and compares it to a database of known threats. (geeksforgeeks, 2023)

Firewalls are software or firmware that prohibit illegal network access. The firewall examines and blocks incoming and outgoing traffic using a set of rules. (simplilearn, 2023)

Malware

Malware, often known as malicious software, is any software or program that is designed to do harm to a computer, network, or server. (Lutkevich, 2023)

Cyber Threat Intelligence (CTI)

Cyber threat intelligences are data which are collected and used by an organization that help to better understand past, current, and future threats. The data which are acquired provides visibility into what's going within an organization's network, assisting in the identification of prospective risks and the prevention of future attacks. (SecurityScorecard, 2020)

Winlogbeat

Using Winlogbeat, windows event logs are sent to Elasticsearch or Logstash. It can also be set up as a Windows service. It uses Windows APIs to read from one or more event logs, filters the events depending on user-defined criteria, and then delivers the event data to Elasticsearch or Logstash as configured. (Elasticsearch B.V., 2023)

Auditbeat

Auditbeat is a lightweight audit data shipper which gathers Linux audit framework data or windows audit data and keep an monitoring on the integrity of files. These events are sent in real time by Auditbeat to the remainder of the Elastic Stack for further processing. (Elasticsearch B.V., 2023)

Filebeat

Filebeat is a log shipper that is lightweight which helps to keep the simple things simple by providing a lightweight solution to forward and centralize logs and files, whenever gathering logs from security devices, the cloud, containers, hosts. (Elasticsearch B.V., 2023)

ElastAlert

ElastAlert is a basic framework for alerting on anomalies, spikes, or other interesting patterns in Elasticsearch data which functions by merging Elasticsearch with two different types of components which are rule types and alerts. (ElastAlert, 2014)

Telegram

Telegram is an open-source communication program that can encrypt and self-destruct messages. (Telegram, 2023)

2.5 Similar Project Review

2.5.1. Project 1: Detection of DoS attack and Zero Day Threat with SIEM

Author: Sornalakshmi

SIEM (Security Information and Event Management) collects log data from different sources and correlates events to filter harmful activity or assaults. The suggested SIEM solution uses two submodules to monitor network and physical systems. We present a SIEM solution that uses log monitoring to identify Denial of Service (DoS), one of the most hazardous network attacks. There are few ways to identify and trace back a DoS attack, but we present a program that monitors web server logs and alerts as soon as feasible with a low false negative. Log analysis rules are established to create DoS attack alerts. Another SIEM module detects system zero-day threats. We suggest monitoring the alteration of a few common system parameters, which may be caused by a malicious program running in the system. Thus, we propose a module to monitor such metrics and issue warnings depending on the criteria. (Sornalakshmi.K, 2017)

2.5.2. Project 2: A SIEM Architecture for Advanced Anomaly Detection

Authors: Tim LaueA, Timo KleckerB, Carsten KleinerA, Kai-Oliver DetkenB

SIEM (Security Information and Event Management) collects log data from different sources and correlates events to filter harmful activity or assaults. The suggested SIEM solution uses two submodules to monitor network and physical systems. We present a SIEM solution that uses log monitoring to identify Denial of Service (DoS), one of the most hazardous network attacks. There are few ways to identify and trace back a DoS attack, but we present a program that monitors web server logs and alerts as soon as feasible with a low false negative. Log analysis rules are established to create DoS attack alerts. Another SIEM module detects system zero-day threats. We suggest monitoring the alteration of a few common system parameters, which may be caused by a malicious program running in the system. Thus, we propose a module to monitor such metrics and issue warnings depending on the criteria. (Tim LaueA, 2022)

2.5.2. Project 3: Cyber Threat Intelligence from Honeypot Data using Elasticsearch

Authors: Hamad AL-Mohannadi*, Irfan Awan, Jassim Al Hamar†, Andrea Cullen, Jules Pagan Disso‡ and Lorna Armitage

A new threat intelligence technique is proposed in this project by hosting two low interaction honeypots, Kippo and Dionea, on the AWS cloud. To attract attackers, those honeypots functioned as real operating systems. Honeypot events logging was enabled for activities such as root failed authentication password, remote error, channel direct-tcpip request, and so on. Event logs from honeypot are then forwarded to SIEM (ELK stack) where those logs are analysed to identify patterns in attacker behaviour. Data from honeypot logs was used to identify IoCs, which were then used to create custom CTI. Using a honeypot for CTI allowed us to collect IoC from various real-world attacks without disrupting the production environment. (Hamad AL-Mohannadi, 2018)

2.6. Comparison Table

S.N.	Features	Project 1	Project 2	Project 3	This Project
1.	Using SIEM to collect, correlate and visualize logs	✓	✓	✓	✓
2.	Alert notification system for threat detection.	✓	X	✓	✓
3.	Implementation of IDS System	X	✓	X	✓
4.	Integration with Cyber Threat Intelligence	X	X	✓	✓
5.	Integration of cached Server (Memcached)	X	X	X	✓

Figure 13: Comparison Table.

2.7. Analysis and Conclusion of the Comparison

In above comparison table, considering all of the features outline above, some of features are missing by those three similar projects. Only one out of the three projects mentioned above had custom CTI features for correlation and implementation of IDS system. None of the above have implemented cached server for quick access storage which help SIEM tool to access quickly to correlate IOCs from CTI such as generated alerts and the ability to query third-party CTI for correlation. All the three projects have SIEM functionality for collecting, correlating, and visualizing logs. Real Time Threat Detection System With Threat Intelligence includes all these features including those found in the other three similar projects. The combination of the above-mentioned core features enables This project to improve visibility, increase correlation capabilities, and decrease detection and

response times. This help to distinguishes this project from other projects of a similar nature.

Chapter 3: Development

3.1. Selected Methodology

3.1.1. Evolutionary Prototyping

The methodology that has been selected for this project is evolutionary prototyping methodology. Evolutionary prototyping is a technique that involves repeatedly altering the original prototype based on client feedback until it is finally approved. This strategy offers a more efficient approach, which saves time and work. (geeksforgeeks.org, 2022)

To implement external feedback, identify necessary demands, and confirm applicability as new requirements are added as per feedback from user, such methodology uses an interactive approach with client. (LAL, 2022)

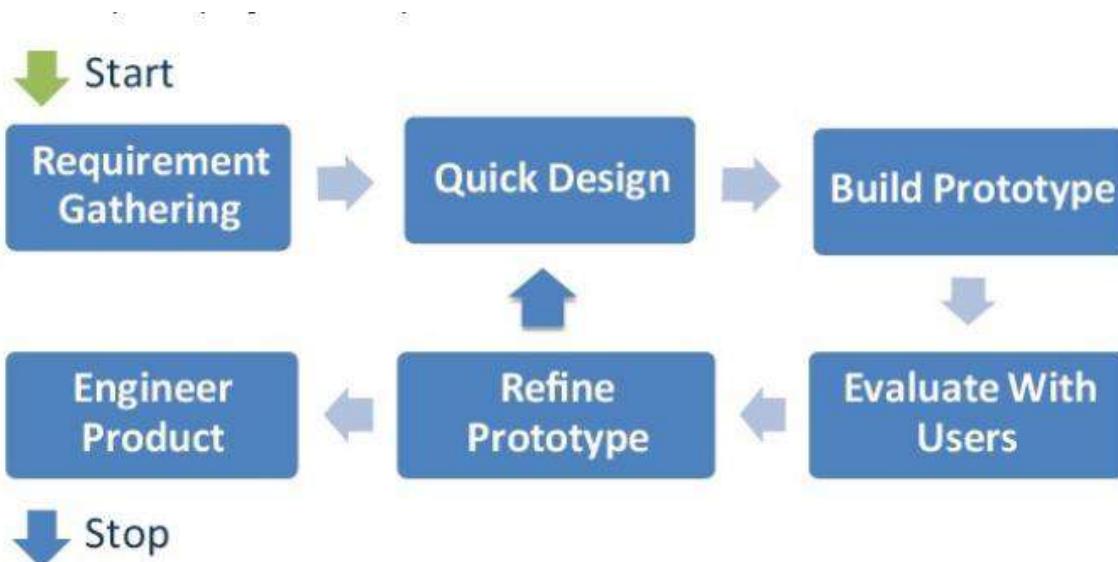


Figure 14: Evolutionary Prototyping. (Kris, 2018)

Reasons for selecting this methodology

- Gets user feedback as user engages with the system
- As per user feedback system are optimize which can insure to most likely to meet user requirements.
- In this methodology, system's delivery is accelerated.

- By investigating and analysing each phase's issues it helps to minimize cost. (intland software, 2021)
- This methodology helps to reduce the number of critical and significant problems during testing. (intland software, 2021)

3.1.2 Phases in Evolutionary Prototyping

- **Initial Phase (Requirement Gathering):** In this phase, in depth research on the project was done for clear vision of project, feasibility, resource requirement. Finalization of the client and required resources gathering task were carried out. Client was scheduled for meeting to know their requirement. Different research on similar project were done and pre purvey were conducted to know the importance and need of the project. In same phase, proposal for the project was finalized and goals and scope of the project was set.
- **Quick Design:** In this phase, flowchart, block diagram layout for this project and LAN network topology were design. Draw.io application was used to design flowchart and block diagram and GNS3 was used to design network topology.
- **Build Prototype:** After a draft design was developed it was followed by development process. In this phase development process for this project was kick started where a prototype of system will be created by analysing and implementing basic requirements gathered form the client. In this phase, LAN network was implemented with various systems were setup like ELK server, Memcached server, MISP server, Webserver, NIDS, windows client, and network devices like Mikrotik router, switch, pfsense firewall. Log shipping agent are installed as per required to devices for log processing and centralization. Script was developed for pulling misp data to Memcached and creating look up function with Memcached server to found match for threat. Alert system was also setup when threat was threat detected. In this way, working prototype was developed.

- **Evaluate with Client:** A client meeting was scheduled to have discussion on the initial prototype of system. In this phase, initial prototype was deployed and was available for end user for testing and reviewing of developed system for this project. Client gave feedback after reviewing the prototype and those feedback was registered.
- **Refine Prototype:** In this phase, there were activities for necessary system optimization, addition of features to developed system as per the client feedback and requirement. The developed system was continuously refined with the help of client's review, feedback, and requirement. This was continued until the developed system for this project satisfies the client. This process will be repeated until and unless the final product is launched.
- **Final Product:** In this phase, final product was produced by approving the developed system after number of enhancements, refinement, testing as per client feedback and review. Finally, the system was implemented in client's organization as a product.

Consider methodology in appendix D

3.2 Work Breakdown Structure

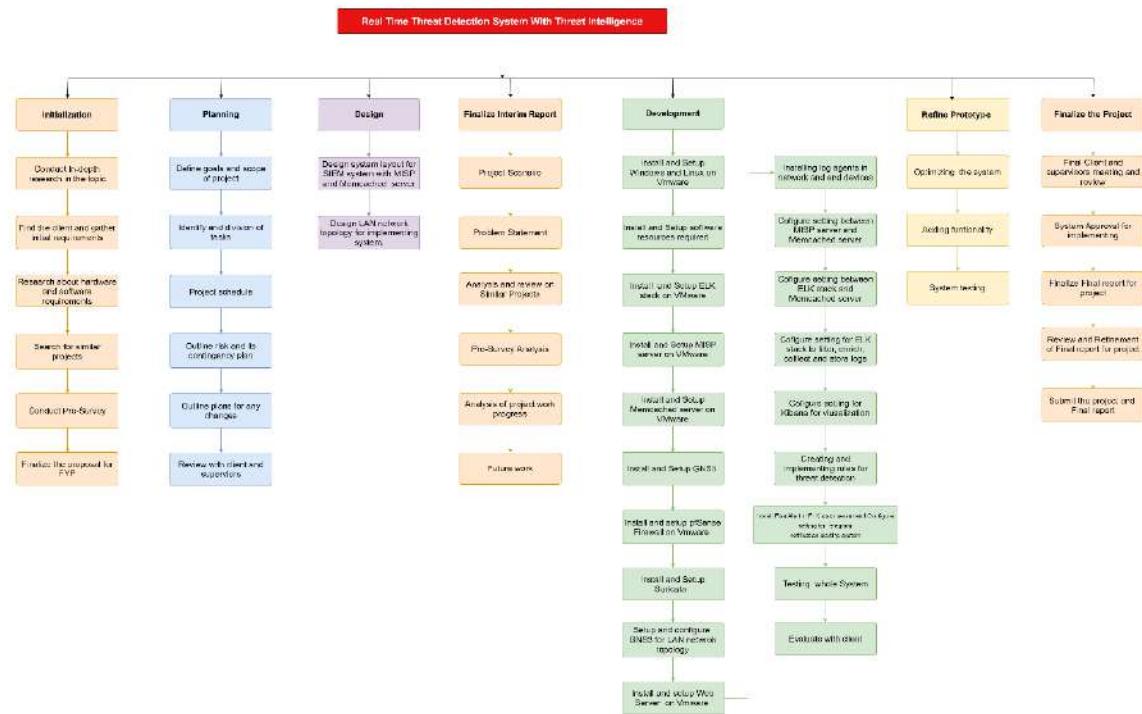


Figure 15: Work Breakdown Structure for this project.

3.3 Gantt Chart

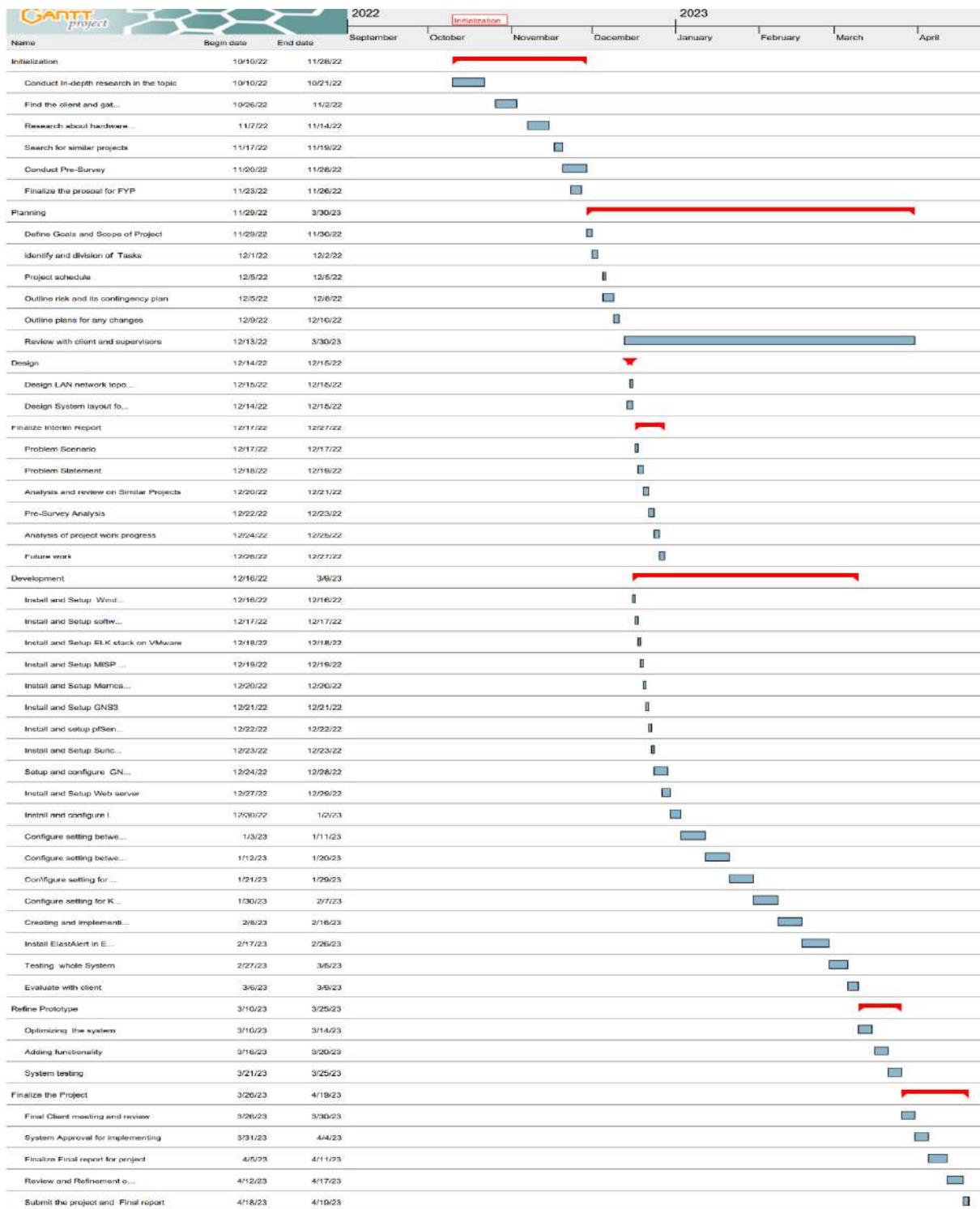


Figure 16: Gantt chart for this project.

3.5 Milestone

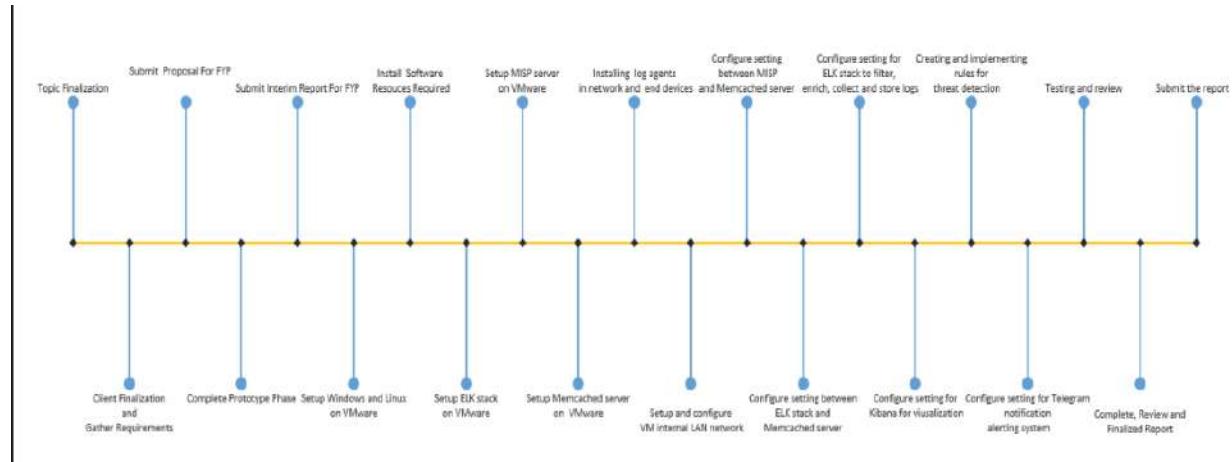


Figure 17: Milestone for this project.

3.6 Survey Result

3.6.1 Pre-Survey Results

Pre-Survey for this project was conducted among 15 people who provided their responses. By looking at pre-survey result 60% people used windows 11 and 40% used windows 10. 80% people updates their windows system and 80% are chrome browser users. Most people do not use antivirus prefer windows defender in their windows system and they update antivirus sometimes only. Malware and phishing attacks are most popular among these people. 80% people knew about malware attack where ransomware and viruses are most heard by them. About 50% People knew SIEM security solution and 50% does not. About 40% people thought real-time monitoring and threat detection is important for System Security, 40% very important. About 60% people have not heard of CTI platform. About 53% people thought that CTI can help in threat detection and 53% thought that CTI can help in SIEM solution for better monitoring and threat detection. Lastly, 66% people thought that cyber security is extremely important in their daily life. This survey shows that features like SIEM with integration of CTI are very important which can help to increase visibility, correlation and better threat detection.

[Detail information in Appendix A](#)

3.6.2 Post-Survey Results

Post-Survey for this project was done among 20 people who provided their response. By looking at post survey result 80% of people thought that system was effective and 10% thoughtless it is less effective. MISP lookup feature was most liked features among telegram alert, visualization. About 55% of participants, were satisfied with the quality of threat intel used in this project. 70% of participants thought the system was fast enough to identify threat. About 94% of participants, liked UI very much. According to 90.4% of participants, they thought that system provided sufficient context information about identifying threats. Lastly, about 70% of participant thought that system report and analytic was very help in the identifying threats.

[Detail information in Appendix B](#)

3.7 Requirement and analysis

3.7.1 Feature Requirement

Features for this system are completely based on the client request. Features are listed by client below: -

- Proposed system must provide Realtime monitoring features for all devices, network traffic in a LAN network.
This developed system will monitor the network and end devices 24hrs using ELK stack where logs from these devices was collected, analysed and visualized in dashboard. This will help to understand their network, end devices easily.
- Proposed system must be able to detect malicious activities and intrusion in network as well as in host.
For the detect of malicious activities MISP threat intel, file integrity check, usb detection was used which will help to detect threat in system and network of client.
- Message alert function must be in the proposed system if any threats are detected.
ElastAlert and Telegram app were used to satisfy client need of message alert when threats were detected.

3.7.2. Software Requirements

- VMware workstation pro
- Linux Debian 10 (64 bit)
- Elasticsearch
- Logstash
- Kibana
- Windows 7 (64 bit)
- Python
- MISP
- Memcached
- Telegram
- GNS3

- Mikrotik Router
- Cisco Switch
- PFsense
- Suricata
- ElastAlert

3.7.3. Hardware Requirements

- Laptop

[Detail information in Appendix C](#)

3.8 Design

3.8.1 Block Diagram of System workflow Layout

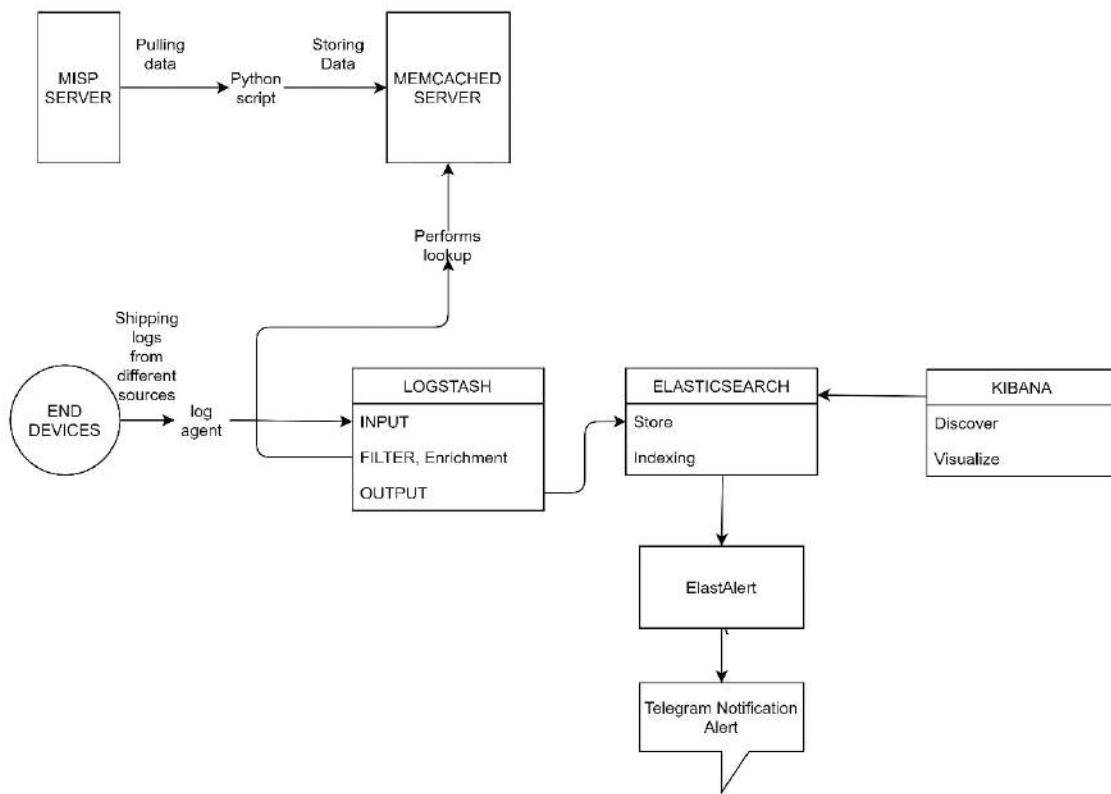


Figure 18: Block Diagram of System workflow Layout.

3.8.2 Flowchart

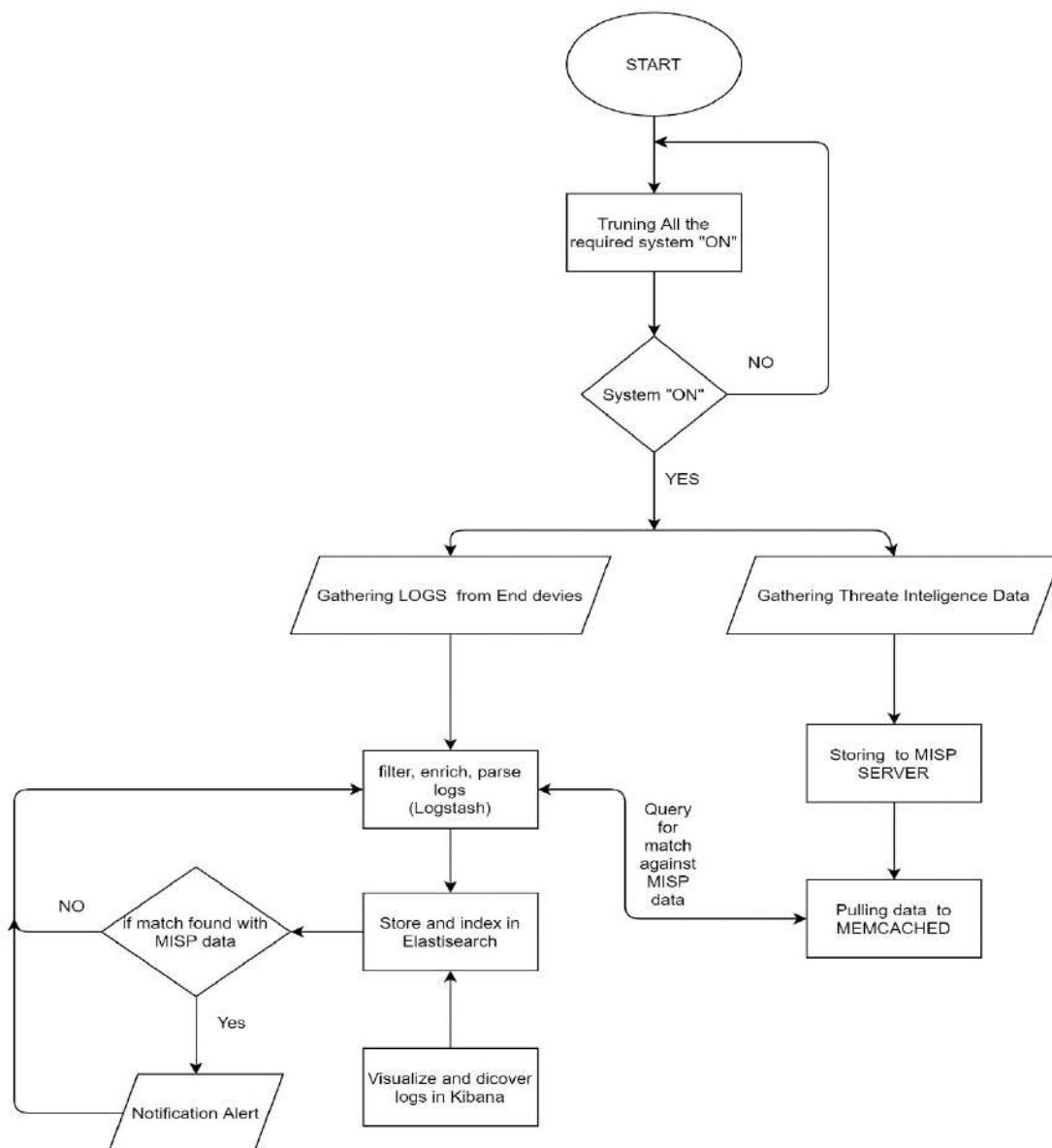


Figure 19: Flowchart for Realtime threat detection with threat intelligence.

3.8.3 Network LAN Topology

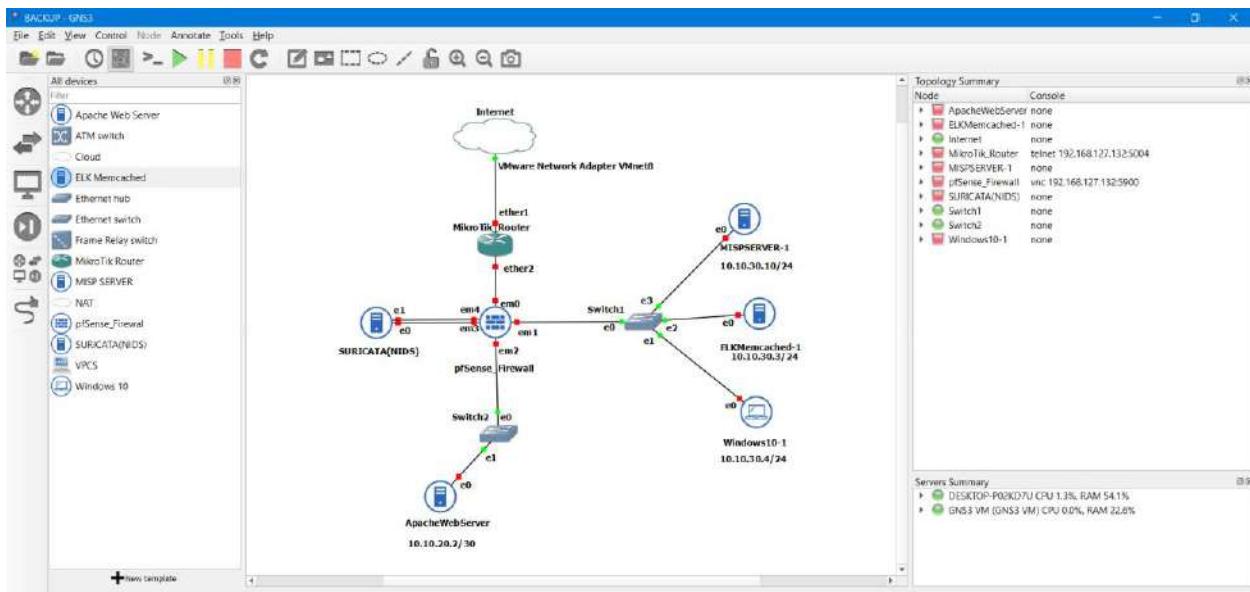


Figure 20: Network topology for this project.

3.9 Implementation

For the development of this project, evolution prototyping methodology has been strictly followed. Varieties of software tools and procedure were used to implement the system in this project.

3.9.1 VMware Workstation 16 PRO

VMware Workstation 16 Pro having 64-bit architecture was used as hypervisor that hosted all the required software tool and operating system implemented in this project. Windows 10, ELK stack, Memcached server, MISP server, Debian 10 Linux, Suricata NIDS, Apache web server, GNS VM etc were hosted in VMware.

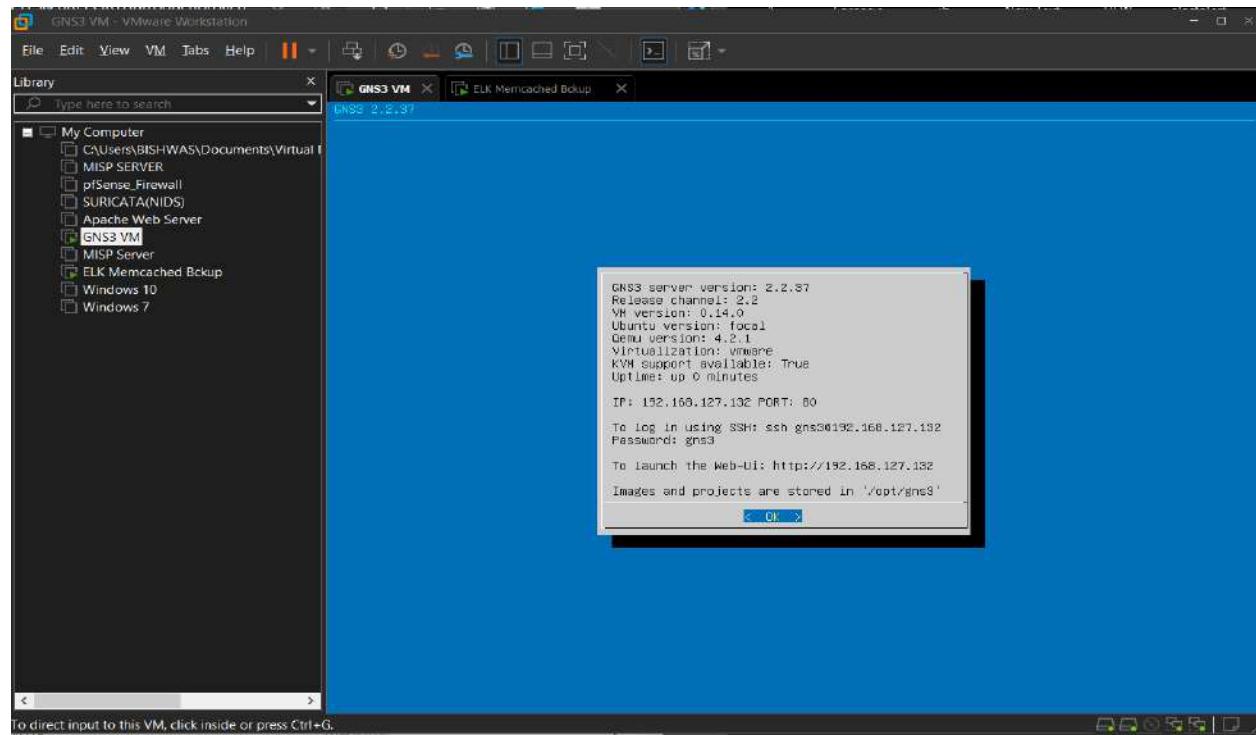


Figure 21: VM ware.

[Full installing Screenshot in Appendix D: VMware Workstation 16 Pro](#)

3.9.2 Windows 10

Windows 10 operating system was used as the End device from where host logs were collected using auditbeat and winlogbeat and send the log to ELK stack for log analysis. Screenshots of software tools and configuration are used to show the implementation of those tools in this project are explain below:

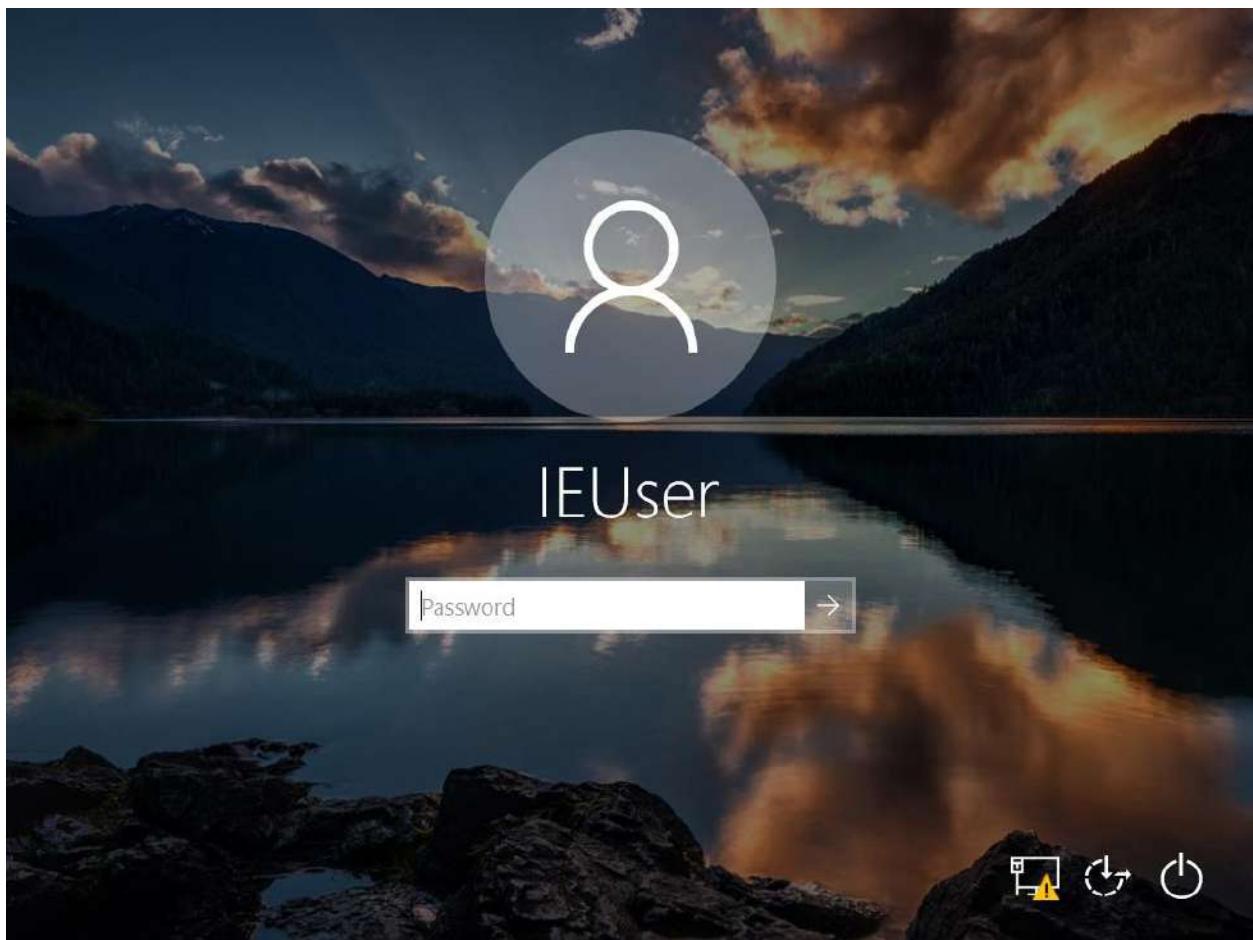


Figure 22: Windows 10.

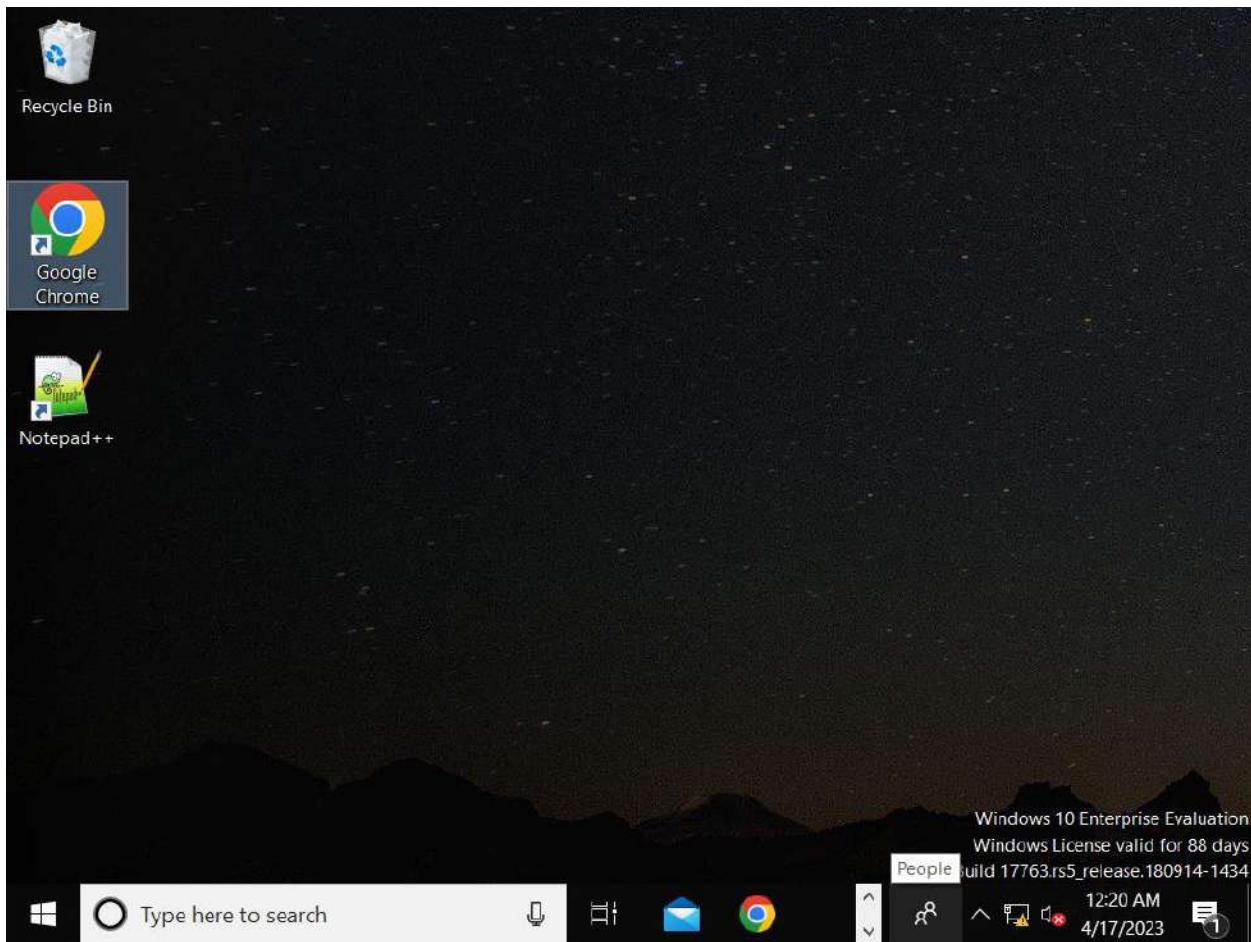


Figure 23; Windows 10 desktop.

[Full installing Screenshot in Appendix D: Windows 10](#)

3.9.3 Debian 10 Linux

Debian 10 Linux was used for base Operating system for running ELK server, MISP server, Apache Web server, Memcached Server, Suricata NIDS. This OS was installed within VMware and same operating system was used for running all servers and NIDS used in this project. Same procedure was followed for installing this operating system for all servers and NIDS.

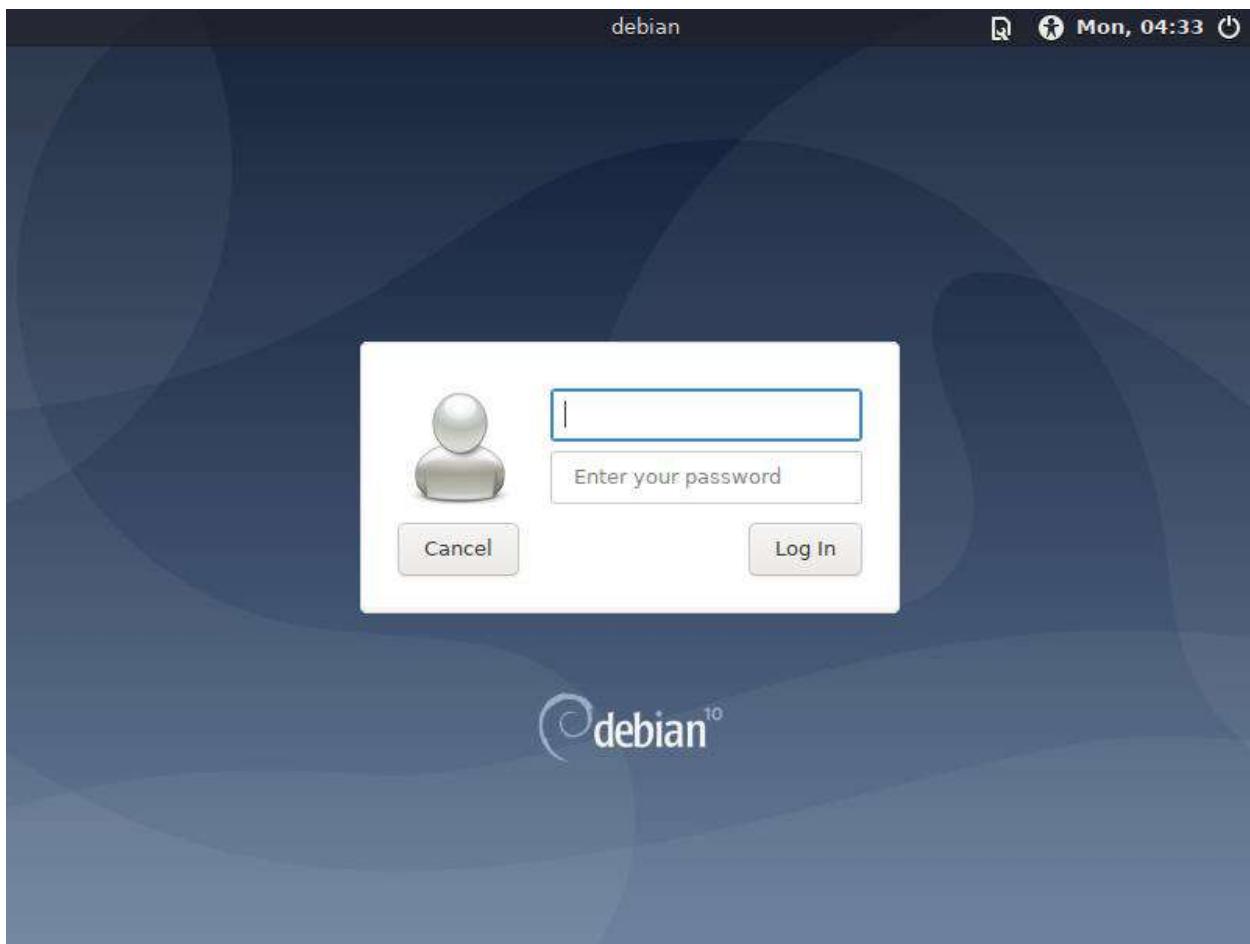


Figure 24: Debian 10 Linux.

[Full installing Screenshot in Appendix D: Debian 10 Linux](#)

3.9.4 GNS3

GNS3 was used to design and implement local LAN network to create real world scenario. All the VMs, network devices were connected to each other to create a LAN network used in this project.

This LAN network topology was design and setup in GNS3 environment where develop system for this project will be implemented. For these 4 different networks 192.168.239.0, 10.10.10.0, 10.10.20.0, 10.10.30.0 were used. Network Address 192.168.239.0 was used for internet access by configuring Mikrotik router in dhcp client mode interface and using VM adapter interface for Internet cloud. pfsense as firewall was install in VM and connected to Mikrotik router with network address of 10.10.10.0. Suricata was install in VM and configured in same network where interface of Suricata and pfsense will be configured in bridge mode for port mirroring to mirror incoming and outgoing network traffic. Web server was install in VM and its was connect to firewall through switch working in 10.10.20.0 network. ELK stack as SIEM and Memcached server as cached server was setup and hosted on same VM. MISP server as CTI was hosted on another VM. Windows 10 VM was added to network for as end PCs. These server and end PCs was configured in 10.10.30.0 network and connect to firewall through switch.

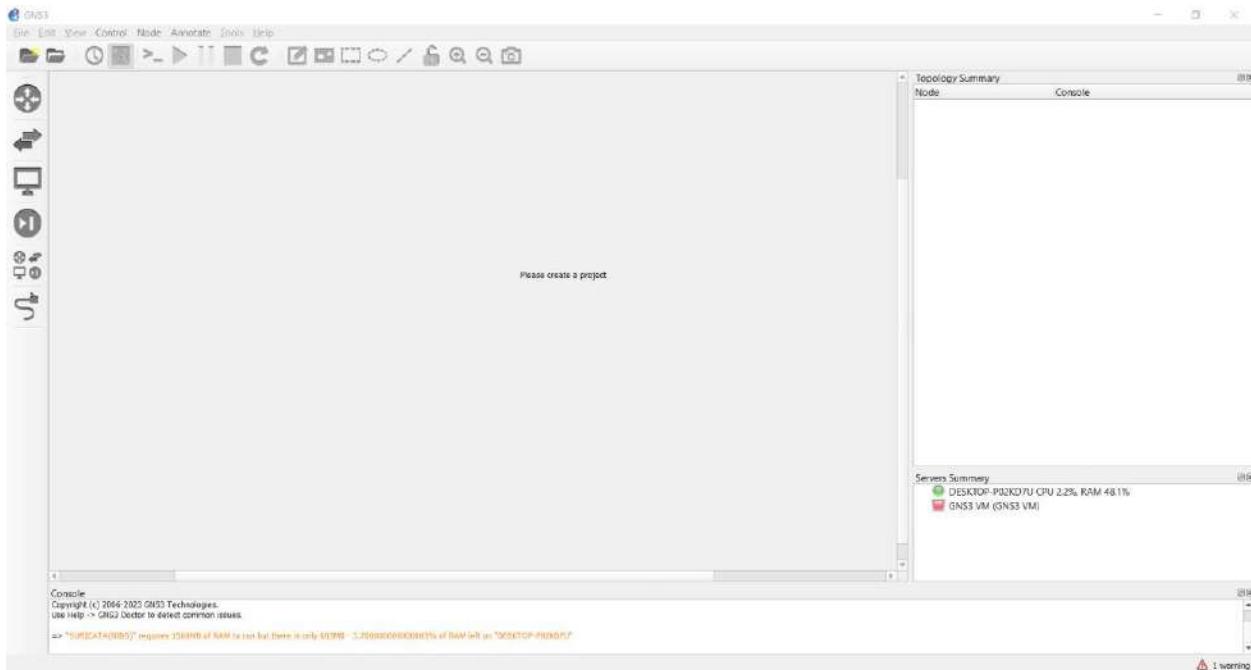


Figure 25: User interface of GNS3.

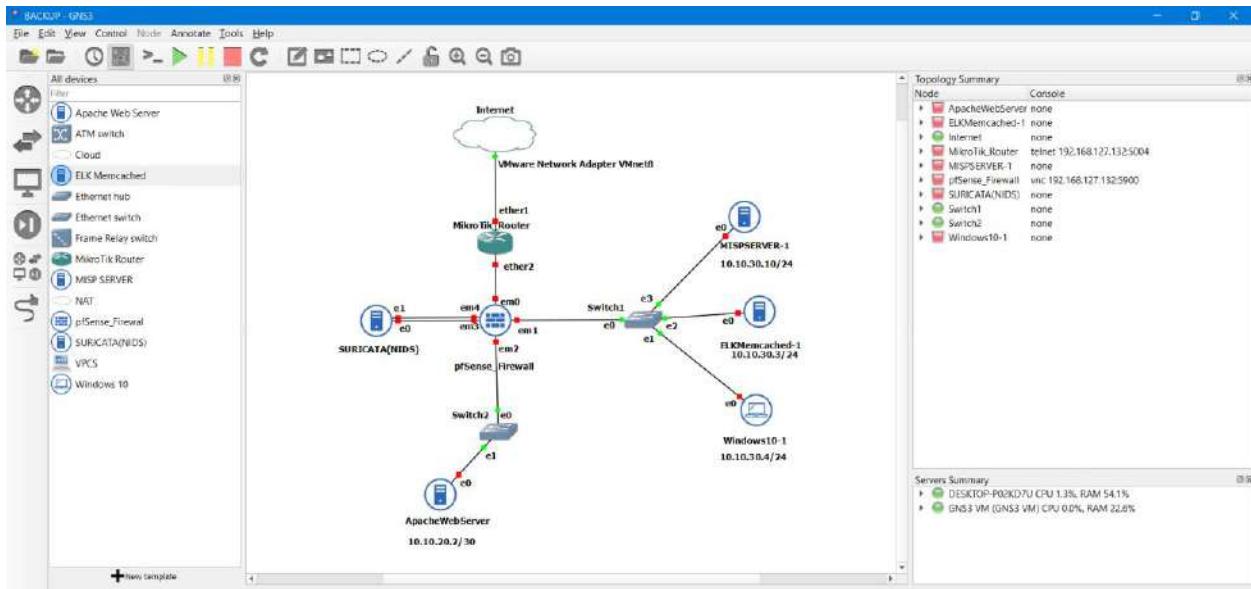


Figure 26: Topology created in GNS3.

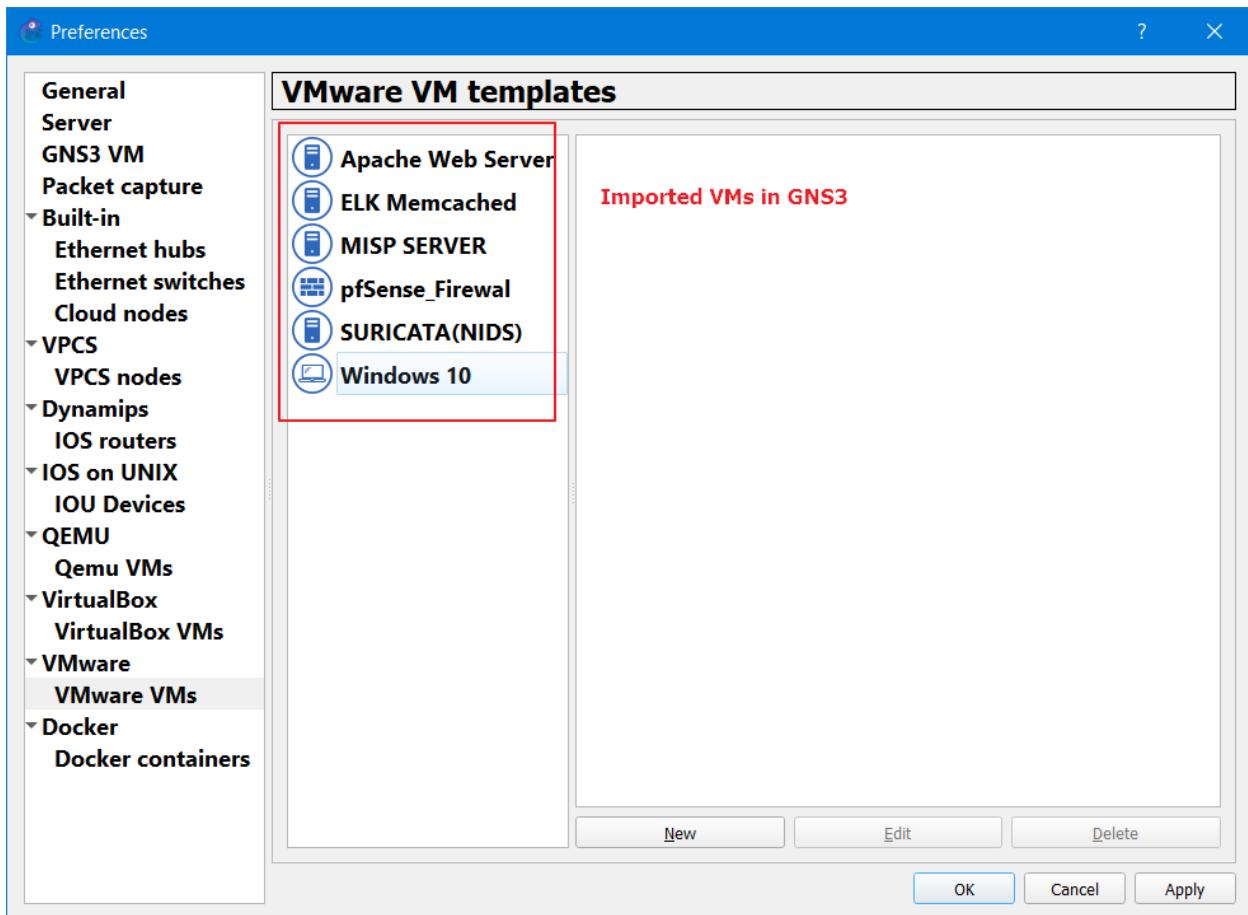


Figure 27: Imported VMs in GNS3.

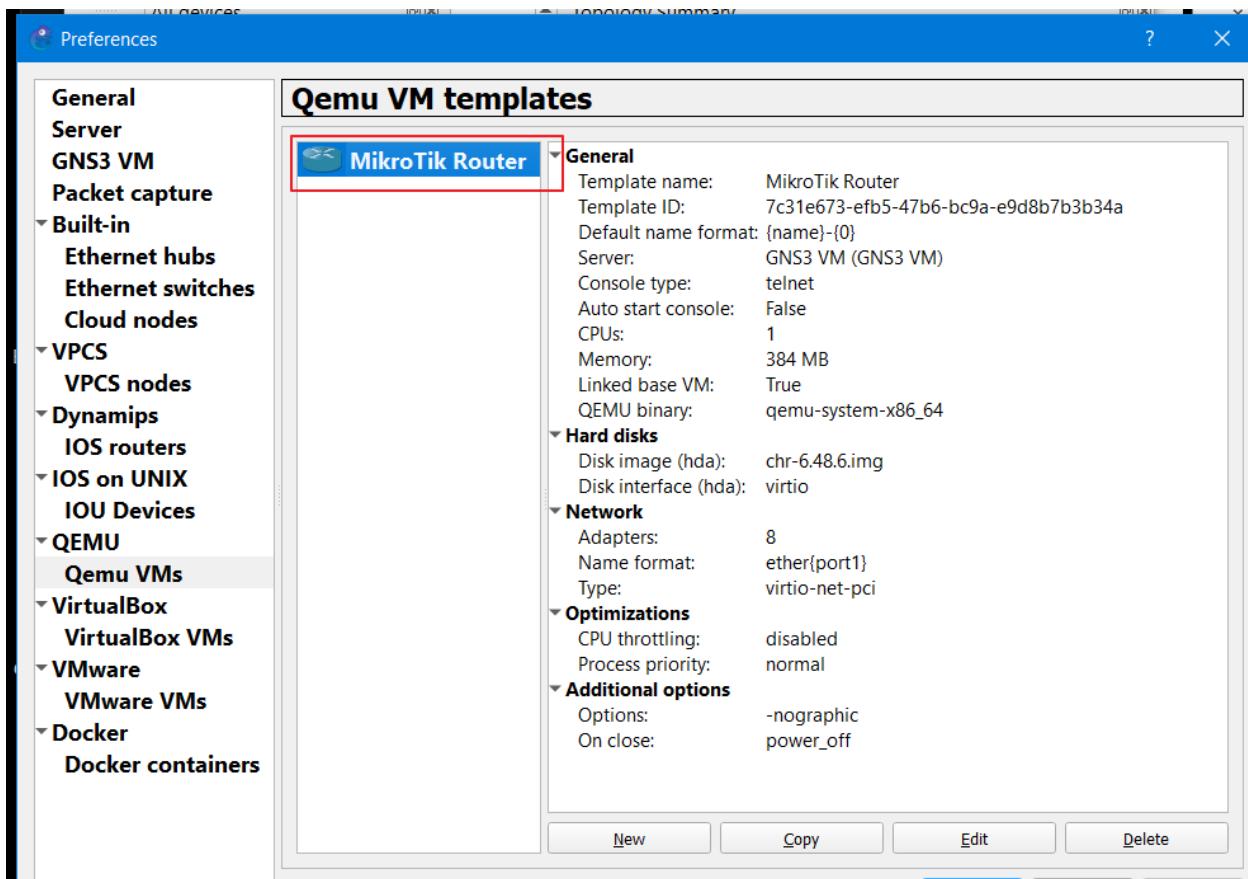


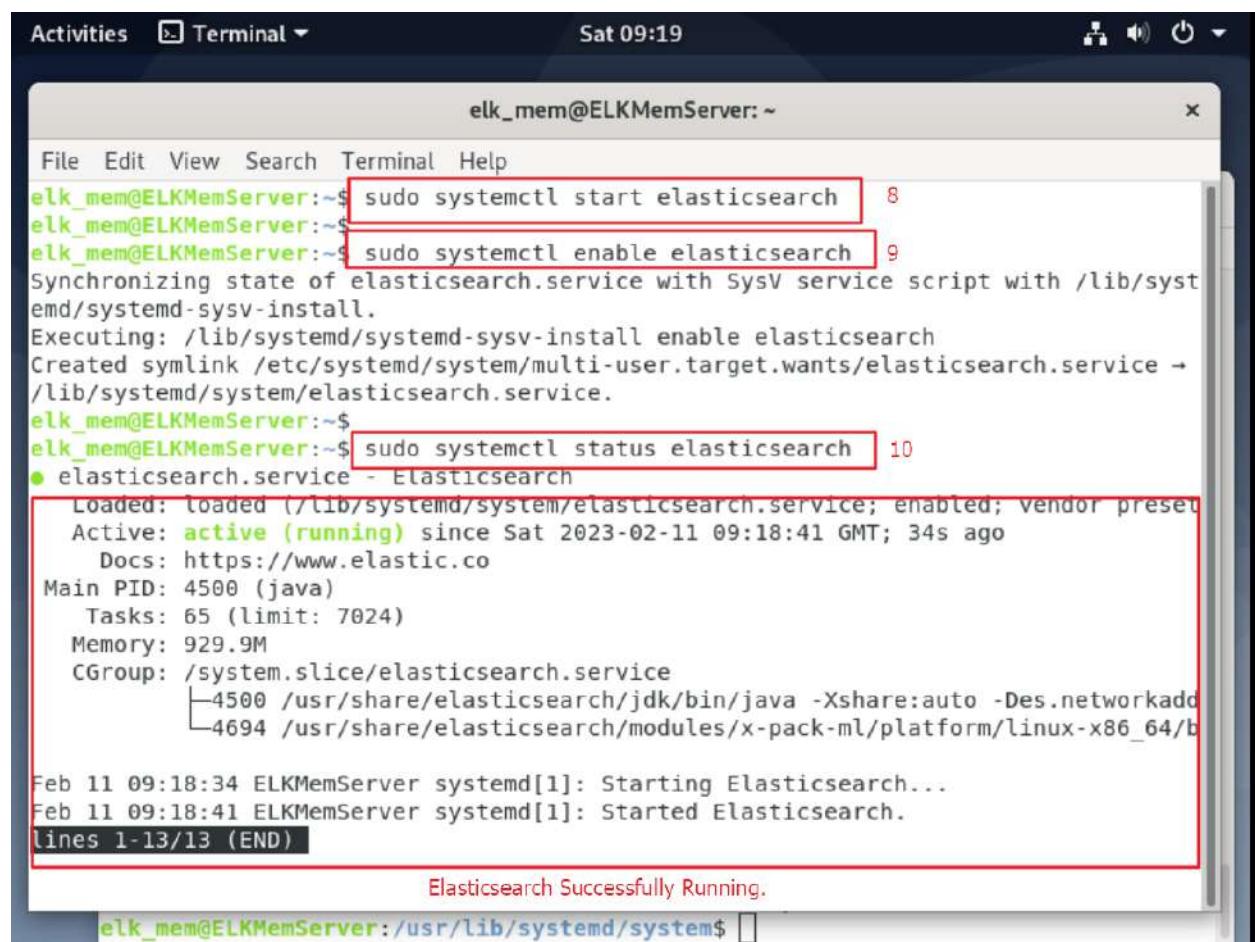
Figure 28: Import Mikrotik Router.

[Full installing Screenshot in Appendix D: GNS3](#)

3.9.5 ELKMemcached Server

This server was made from the combination of Elasticsearch, Kibana, Logstash and Memcached where Logstash collects, filters, enriches the logs using log agents like filebeat, auditbeat, winlogbeat and send to Elasticsearch for storing and indexing and Kibana discovers the logs and visualizes the logs. Memcached was used to store data coming from MISP server temporarily and accessed by Logstash for lookup.

ElasticSearch



The screenshot shows a terminal window titled "elk_mem@ELKMemServer:~". The terminal displays the following command sequence:

```
elk_mem@ELKMemServer:~$ sudo systemctl start elasticsearch          8
elk_mem@ELKMemServer:~$ sudo systemctl enable elasticsearch         9
Synchronizing state of elasticsearch.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service →
/lib/systemd/system/elasticsearch.service.
elk_mem@ELKMemServer:~$ sudo systemctl status elasticsearch        10
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: active)
   Active: active (running) since Sat 2023-02-11 09:18:41 GMT; 34s ago
     Docs: https://www.elastic.co
 Main PID: 4500 (java)
    Tasks: 65 (limit: 7024)
   Memory: 929.9M
      CGroup: /system.slice/elasticsearch.service
              └─4500 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networkaddress.cache.size=64 -Des.nodes=1 -Des.node.name=elasticsearch -Des.http.c...  
Feb 11 09:18:34 ELKMemServer systemd[1]: Starting Elasticsearch...
Feb 11 09:18:41 ELKMemServer systemd[1]: Started Elasticsearch.
```

A red box highlights the status command output, which includes the process ID (4500), memory usage (929.9M), and the command line arguments for the Java process. Below the terminal window, a message "Elasticsearch Successfully Running." is displayed in red text.

Figure 29: Elasticsearch

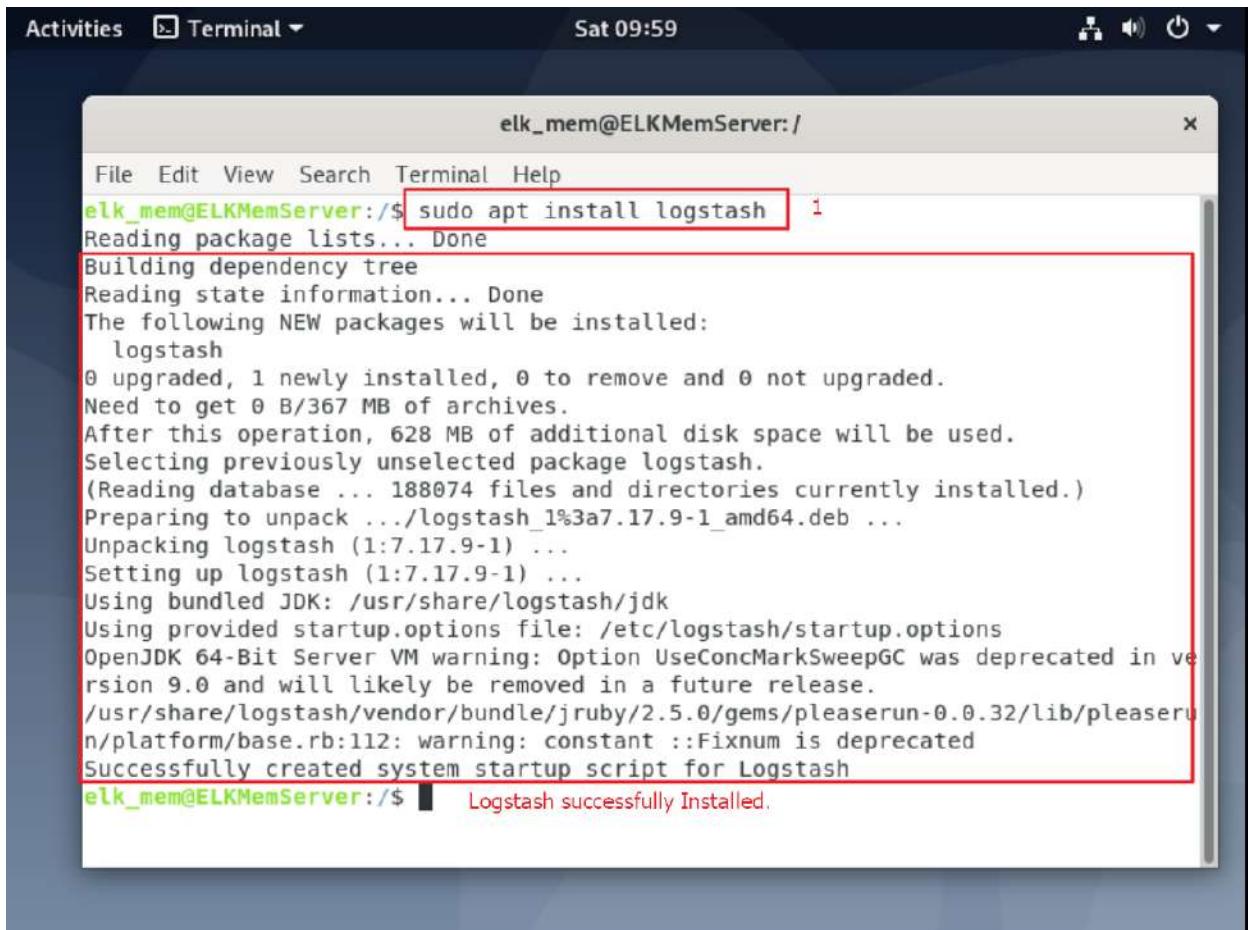
The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The terminal displays several log entries about password changes for users like logstash_system, beats_system, remote_monitoring_user, and elastic. Below these, a command is run to curl the Elasticsearch API at localhost:9200. The output of this command is highlighted with a red box and labeled "OUTPUT". The output shows the Elasticsearch configuration, including its name, cluster name, UUID, and various version details. The configuration object is defined as follows:

```
{  
  "name" : "ELKMemServer",  
  "cluster_name" : "elasticsearch",  
  "cluster_uuid" : "z17KbuWBT-mhpfe16x_4QQ",  
  "version" : {  
    "number" : "7.17.9",  
    "build_flavor" : "default",  
    "build_type" : "deb",  
    "build_hash" : "ef4822227ee6b9e70e502f0f0daa52435ee634d",  
    "build_date" : "2023-01-31T05:34:43.305517834Z",  
    "build_snapshot" : false,  
    "lucene_version" : "8.11.1",  
    "minimum_wire_compatibility_version" : "6.8.0",  
    "minimum_index_compatibility_version" : "6.0.0-beta1"  
  },  
  "tagline" : "You Know, for Search"  
}
```

At the bottom of the terminal, there are two blank command lines: "root@ELKMemServer:~#" and "root@ELKMemServer:~#".

Figure 30: Accessing Elasticsearch.

Logstash



The screenshot shows a terminal window titled "Activities Terminal" with the command "elk_mem@ELKMemServer:/\$ sudo apt install logstash" highlighted in red. The output of the command is displayed below, showing the progress of the package installation. A large red rectangle highlights the entire output area of the terminal.

```
File Edit View Search Terminal Help
elk_mem@ELKMemServer:/$ sudo apt install logstash 1
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  logstash
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/367 MB of archives.
After this operation, 628 MB of additional disk space will be used.
Selecting previously unselected package logstash.
(Reading database ... 188074 files and directories currently installed.)
Preparing to unpack .../logstash_1%3a7.17.9-1_amd64.deb ...
Unpacking logstash (1:7.17.9-1) ...
Setting up logstash (1:7.17.9-1) ...
Using bundled JDK: /usr/share/logstash/jdk
Using provided startup.options file: /etc/logstash/startup.options
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
/usr/share/logstash/vendor/bundle/jruby/2.5.0/gems/pleaserun-0.0.32/lib/pleaserun/platform/base.rb:112: warning: constant ::Fixnum is deprecated
Successfully created system startup script for Logstash
elk_mem@ELKMemServer:/$ Logstash successfully Installed.
```

Figure 31: Logstash installation.

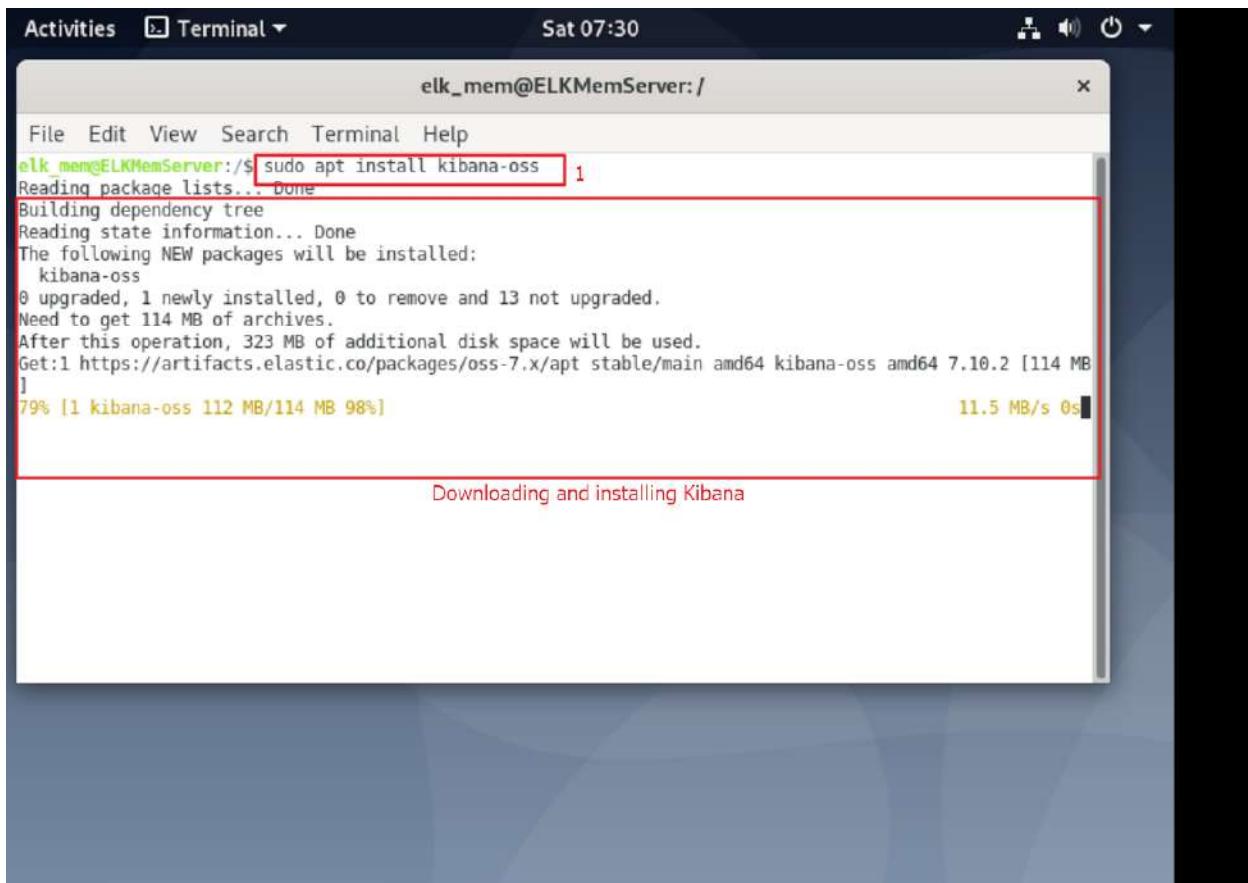
The screenshot shows a terminal window titled "Terminal" with the command prompt "elk_mem@ELKMemServer: /etc/logstash". The terminal displays the following sequence of commands and their outputs:

```
elk_mem@ELKMemServer: /etc/logstash$ sudo systemctl start logstash 2
elk_mem@ELKMemServer: /etc/logstash$ sudo systemctl enable logstash 3
Created symlink /etc/systemd/system/multi-user.target.wants/logstash.service → /etc/sys
sh.service.
elk_mem@ELKMemServer: /etc/logstash$ sudo systemctl status logstash 4
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset:
     Active: active (running) since Sat 2023-02-11 10:19:58 GMT; 21s ago
   Main PID: 8712 (java)
     Tasks: 41 (limit: 7024)
    Memory: 496.4M
      CGroup: /system.slice/logstash.service
              └─8712 /usr/share/logstash/jdk/bin/java -Xms256m -Xmx256m -XX:+UseCon

Feb 11 10:20:06 ELKMemServer logstash[8712]: [2023-02-11T10:20:06,718][WARN ][lo
Feb 11 10:20:06 ELKMemServer logstash[8712]: [2023-02-11T10:20:06,746][INFO ][lo
Feb 11 10:20:06 ELKMemServer logstash[8712]: [2023-02-11T10:20:06,771][INFO ][lo
Feb 11 10:20:06 ELKMemServer logstash[8712]: [2023-02-11T10:20:06,782][INFO ][lo
Feb 11 10:20:07 ELKMemServer logstash[8712]: [2023-02-11T10:20:07,183][INFO ][lo
Feb 11 10:20:07 ELKMemServer logstash[8712]: [2023-02-11T10:20:07,235][INFO ][lo
Feb 11 10:20:07 ELKMemServer logstash[8712]: [2023-02-11T10:20:07,246][INFO ][lo
Feb 11 10:20:07 ELKMemServer logstash[8712]: [2023-02-11T10:20:07,267][INFO ][fi
Feb 11 10:20:07 ELKMemServer logstash[8712]: [2023-02-11T10:20:07,268][INFO ][lo
Feb 11 10:20:07 ELKMemServer logstash[8712]: [2023-02-11T10:20:07,355][WARN ][fi
lines 1-19/19 (END)          Logstash Successfully Running.
```

Figure 32: Starting Logstash.

Kibana



A screenshot of a Linux desktop environment showing a terminal window titled "Activities Terminal". The terminal window has a red border around its content area. Inside, a command is being run: `sudo apt install kibana-oss`. The output shows the package lists being read, dependencies being built, and state information being updated. It then lists packages to be installed, including "kibana-oss". It shows 0 upgraded, 1 newly installed, and 0 removed. A download progress bar indicates "79% [1 kibana-oss 112 MB/114 MB 98%]" at "11.5 MB/s 0s". A status message at the bottom says "Downloading and installing Kibana". The desktop background is a blue abstract pattern.

Figure 33: Kibana

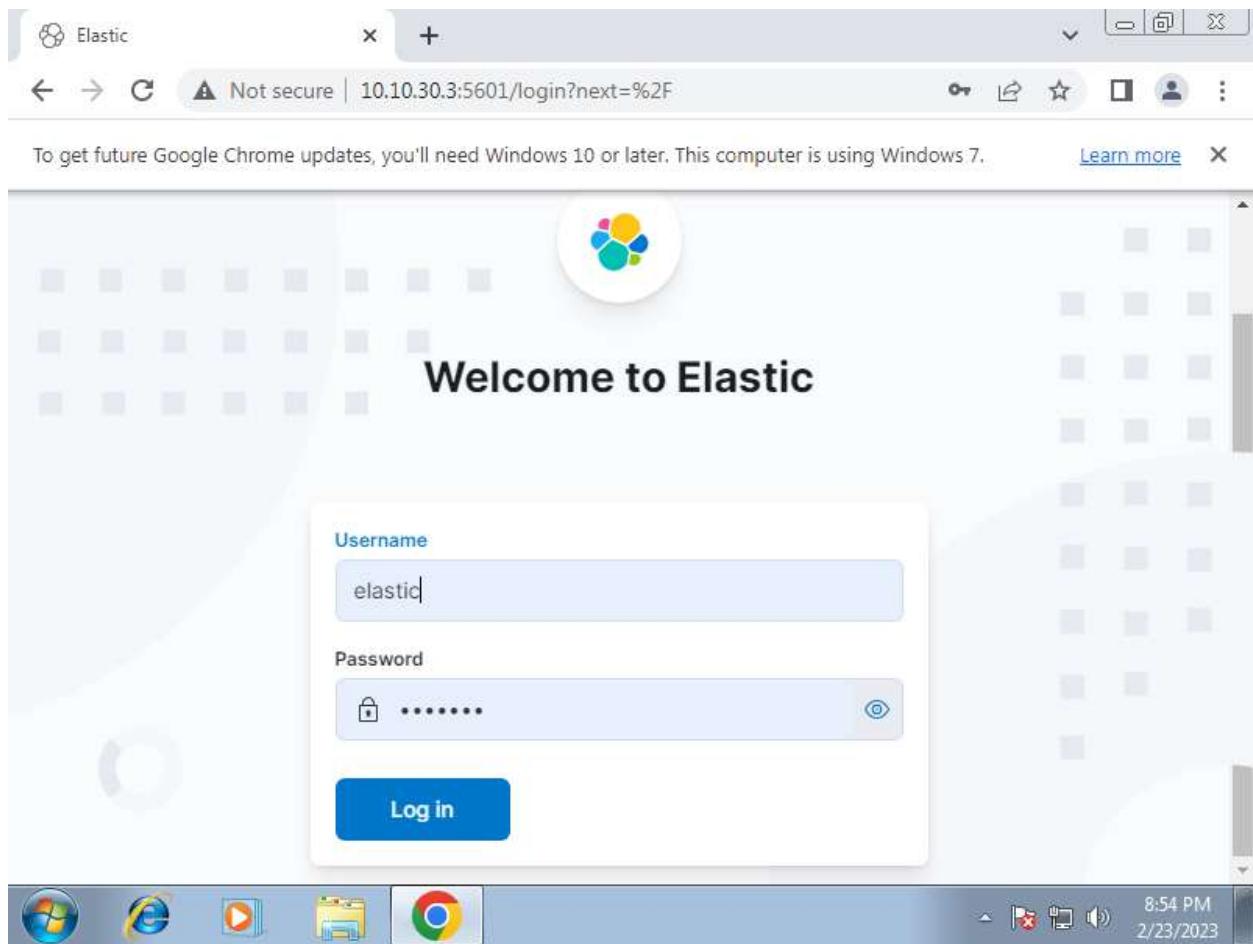
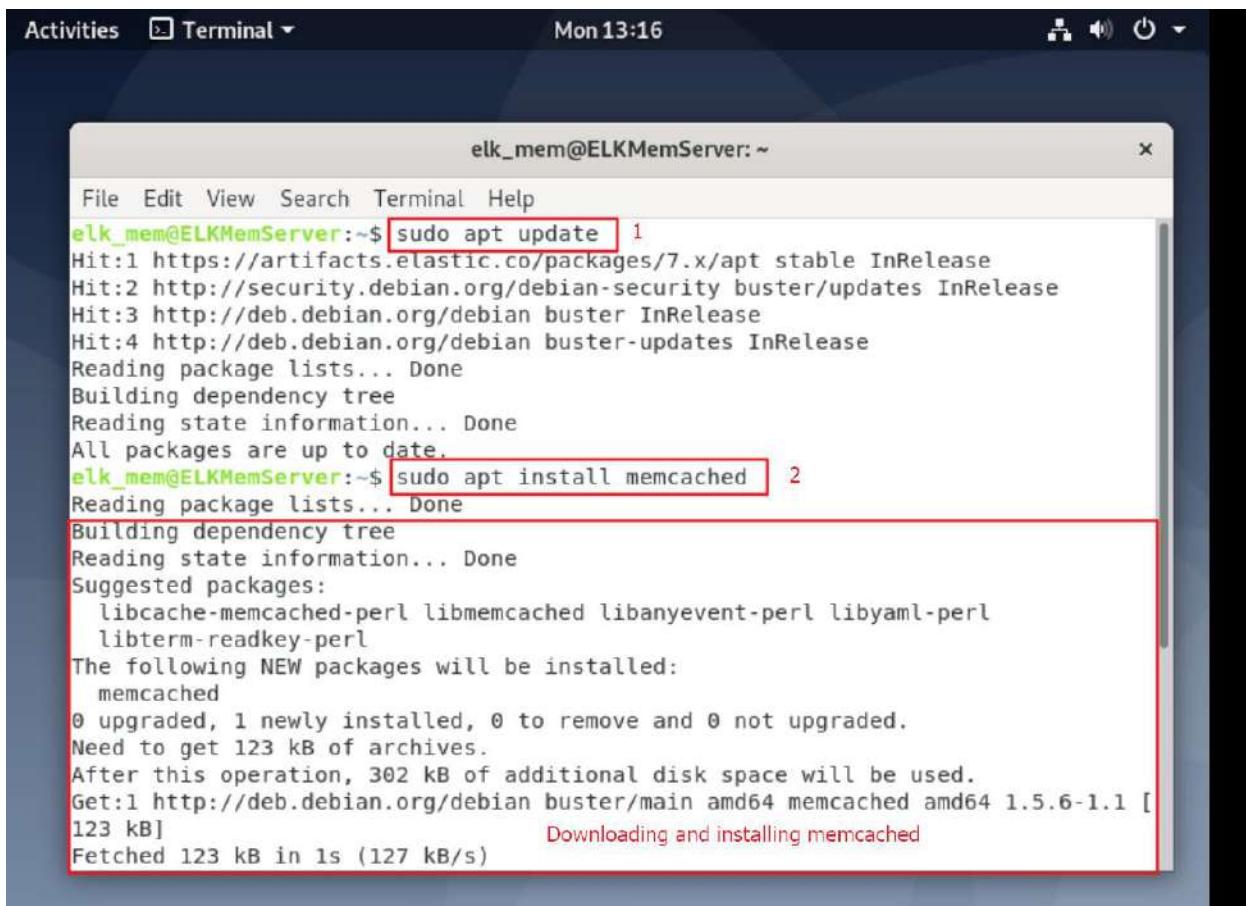


Figure 34: Kibana UI.

Memcached



The screenshot shows a terminal window titled "Activities Terminal" with the status bar indicating "Mon 13:16". The terminal session is running on a server named "elk_mem@ELKMemServer". The user has run the command "sudo apt update" (line 1), which has fetched packages from various repositories. The user then ran "sudo apt install memcached" (line 2), which is shown as a progress bar with the message "Downloading and installing memcached". The terminal also lists suggested packages like libcache-memcached-perl and libmemcached.

```
File Edit View Search Terminal Help
elk_mem@ELKMemServer:~$ sudo apt update 1
Hit:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Hit:2 http://security.debian.org/debian-security buster/updates InRelease
Hit:3 http://deb.debian.org/debian buster InRelease
Hit:4 http://deb.debian.org/debian buster-updates InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
elk_mem@ELKMemServer:~$ sudo apt install memcached 2
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  libcache-memcached-perl libmemcached libanyevent-perl libyaml-perl
  libterm-readkey-perl
The following NEW packages will be installed:
  memcached
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 123 kB of archives.
After this operation, 302 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian buster/main amd64 memcached amd64 1.5.6-1.1 [123 kB]          Downloading and installing memcached
Fetched 123 kB in 1s (127 kB/s)
```

Figure 35: Memcached.

The screenshot shows a terminal window titled "Activities Terminal" with the status "Mon 13:28". The terminal session is as follows:

```
elk_mem@ELKMemServer:~$ sudo systemctl start memcached      5
elk_mem@ELKMemServer:~$ sudo systemctl enable memcached      6
Synchronizing state of memcached.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable memcached
elk_mem@ELKMemServer:~$ sudo systemctl status memcached      7
● memcached.service - memcached daemon
   Loaded: loaded (/lib/systemd/system/memcached.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-02-13 13:15:54 GMT; 12min ago
     Docs: man:memcached(1)
 Main PID: 9833 (memcached)
    Tasks: 10 (limit: 7024)
   Memory: 3.9M
          CGroup: /system.slice/memcached.service
                  └─9833 /usr/bin/memcached -m 64 -p 11211 -u memcache -l 127.0.0.1 -P

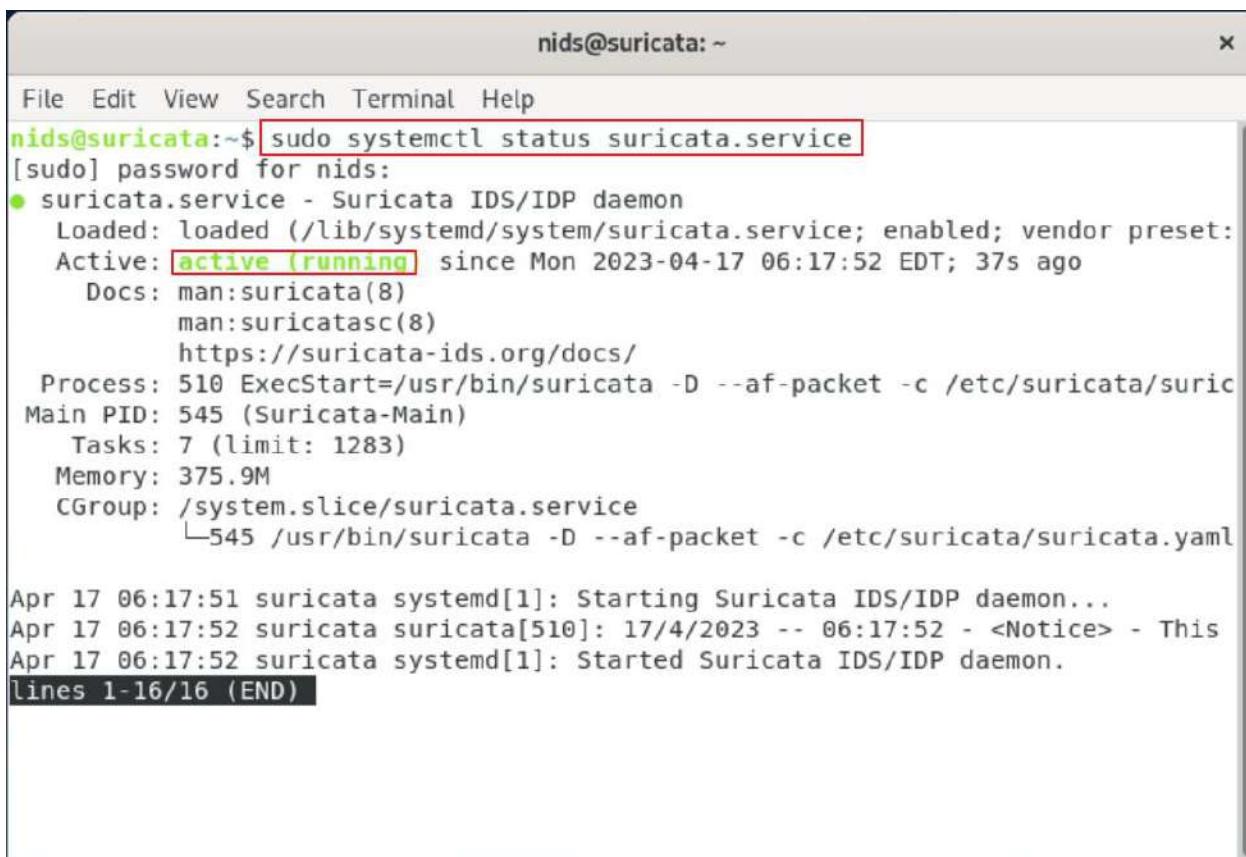
Feb 13 13:15:54 ELKMemServer systemd[1]: Started memcached daemon.
Feb 13 13:15:54 ELKMemServer systemd[1]: /lib/systemd/system/memcached.service:1
Feb 13 13:28:36 ELKMemServer systemd[1]: /lib/systemd/system/memcached.service:1
Feb 13 13:28:36 ELKMemServer systemd[1]: /lib/systemd/system/memcached.service:1
Feb 13 13:28:37 ELKMemServer systemd[1]: /lib/systemd/system/memcached.service:1
lines 1-15/15 (END)
```

Figure 36:Memcached.

[Full installing Screenshot in Appendix D: ELKMemcached Server](#)

3.9.6 Suricata NIDS (Network Intrusion Detection)

Suricata was used as network intrusion Detection in LAN network that monitors network traffic flowing from pfsense interface to outside network. Suricata was given with two interface one for monitoring and one for sending log to Logstash and internet purpose. Suricata's "e1" interface which connected in "em4" interface of PfSense was configured with span port which mirror traffic flowing from that interface and other Suricata "e0" interface connected to "em3" interface of PfSense was configured as normal interface. Filebeat agent with Suricata module enabled was used to shipped log for Suricata to Logstash. Installation and configuration screenshot are given below.



The screenshot shows a terminal window titled "nids@suricata: ~". The terminal displays the output of the command "sudo systemctl status suricata.service". The output shows that the Suricata service is active and running. It provides details about the service's state, load, memory usage, and process information. The terminal also shows the logs for the service's start and successful startup.

```
nids@suricata:~$ sudo systemctl status suricata.service
[sudo] password for nids:
● suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/lib/systemd/system/suricata.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-04-17 06:17:52 EDT; 37s ago
     Docs: man:suricata(8)
           man:suricatasc(8)
           https://suricata-ids.org/docs/
   Process: 510 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml
   Main PID: 545 (Suricata-Main)
     Tasks: 7 (limit: 1283)
    Memory: 375.9M
      CGroup: /system.slice/suricata.service
              └─545 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml

Apr 17 06:17:51 suricata systemd[1]: Starting Suricata IDS/IDP daemon...
Apr 17 06:17:52 suricata suricata[510]: 17/4/2023 -- 06:17:52 - <Notice> - This
Apr 17 06:17:52 suricata systemd[1]: Started Suricata IDS/IDP daemon.

lines 1-16/16 (END)
```

Figure 37: Suricata service.

[Full installing Screenshot in Appendix D: Suricata](#)

3.9.7 MISP Server

MISP Server is opensource threat intelligence platform was hosted locally in VMware running Debian 10 Linux. This server was used as database for threat intel. Here, in this project, MISP server send IoCs like IPs, hashes, domain to Memcached server for lookup to detect threat.

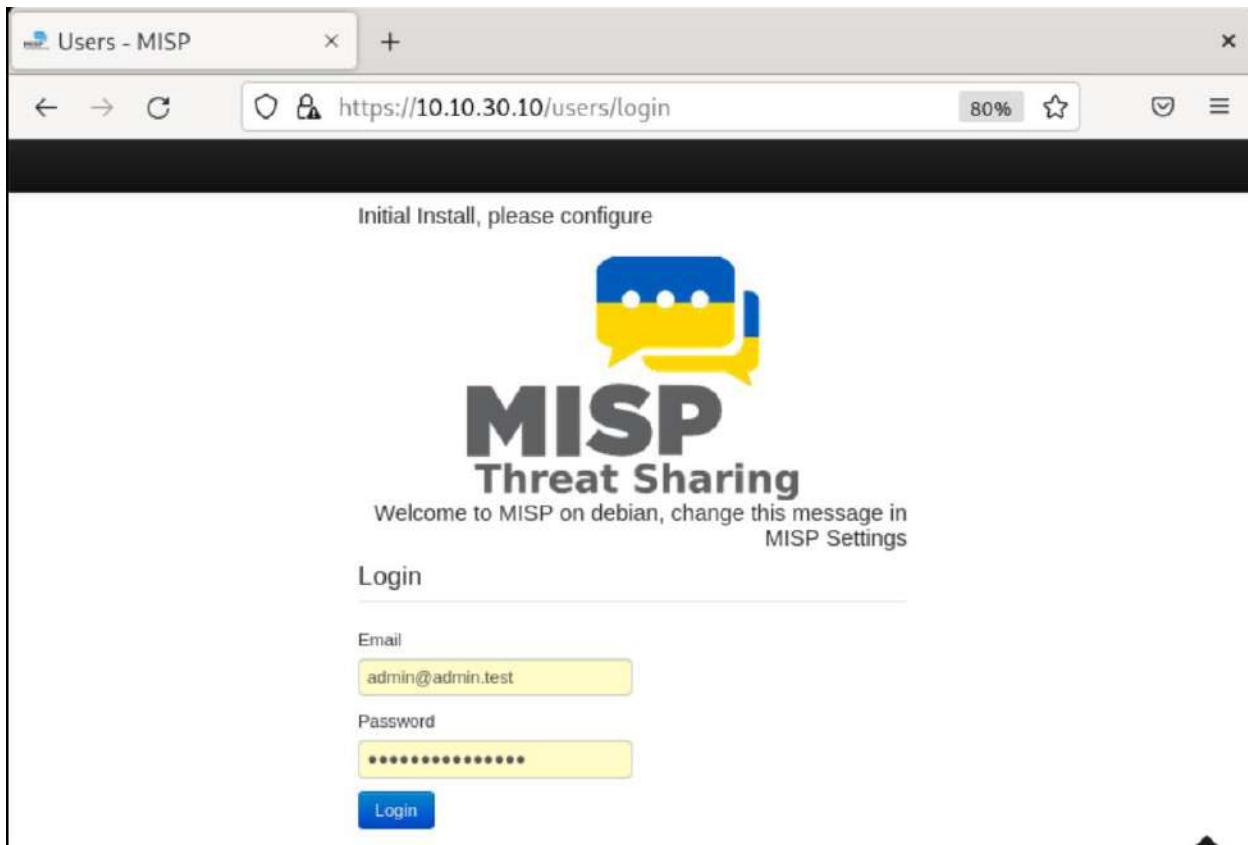


Figure 38: MISP

[Full installing Screenshot in Appendix D: MISP Server](#)

3.9.8 Apache Web Server

Apache Web server was hosted locally on VMware running Debian 10 Linux. This was created as dummy web server placed in DMZ zone which cannot access internal network. Logs created by this server was shipped using filebeat with apache2 module enables. All the installation and configuration screenshot are given below.

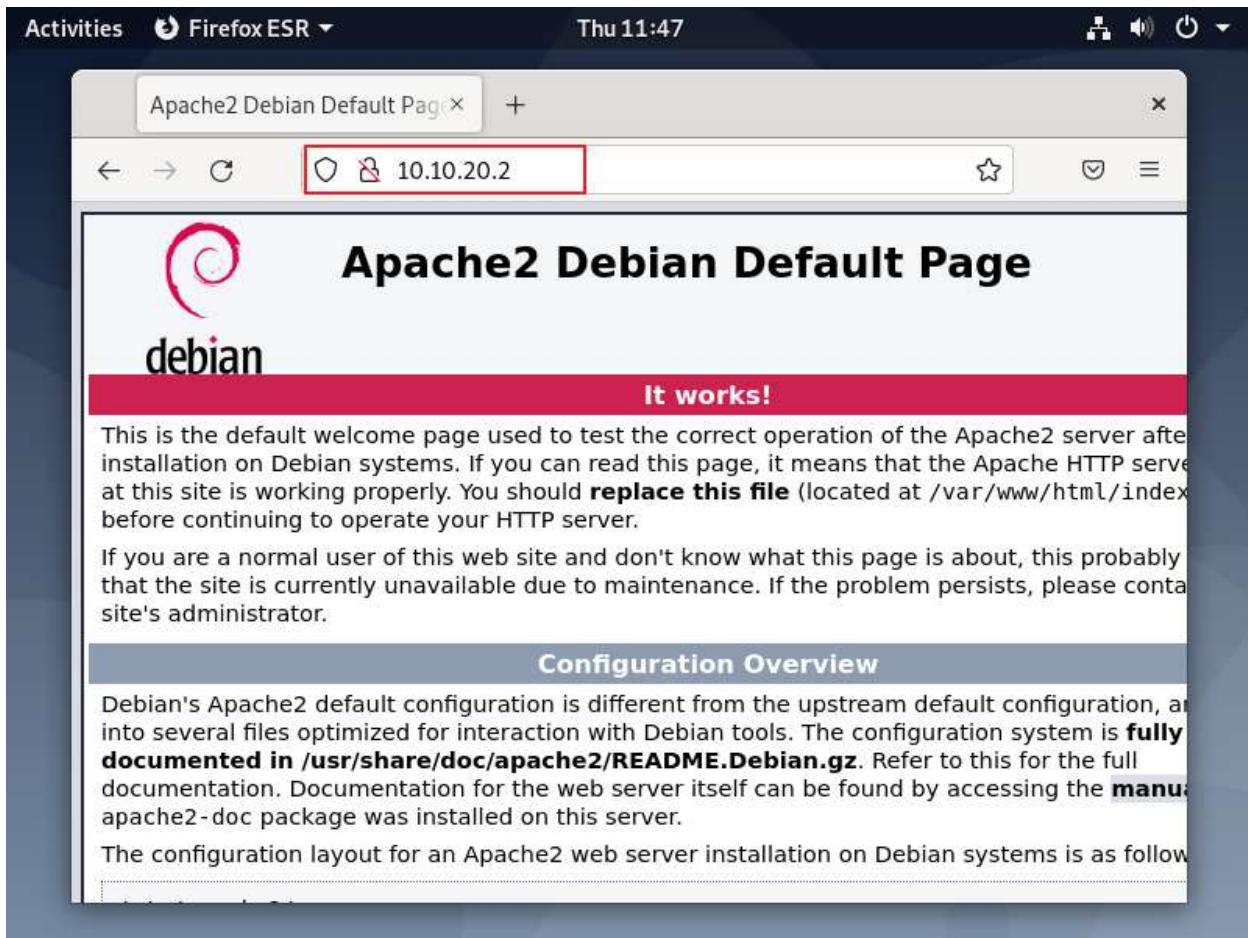
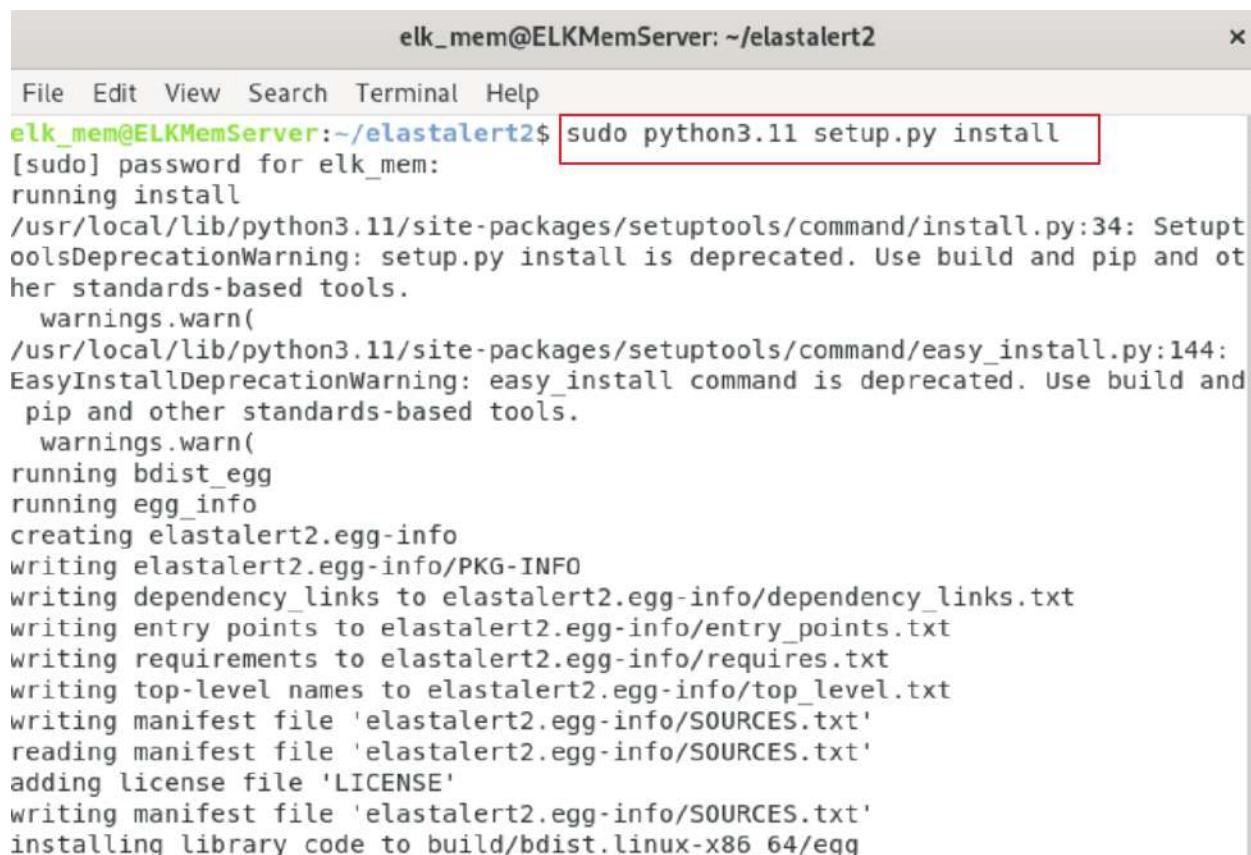


Figure 39: Apache web server.

[Full installing Screenshot in Appendix D: Apache Web Server](#)

3.9.9 ElastAlert

ElastAlert was used to query with Elasticsearch and throw alert when certain condition mention in ElastAlert config file was made. It sends message to telegram after condition was matched. Here, ElastAlert creates its own indices in with in Elasticsearch and look for the matched index in Elasticsearch and their condition to trigger alter to send message to telegram. ElastAlert was hosted on same host as ELKMemcached server. All the installation and configuration for ElastAlert are given below.



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~/elastalert2". The window contains a command-line session where the user is installing the ElastAlert Python package. The command "sudo python3.11 setup.py install" is being run, and the output shows the progress of the installation, including various warning messages about deprecated tools like setup.py and easy_install.

```
elk_mem@ELKMemServer:~/elastalert2$ sudo python3.11 setup.py install
[sudo] password for elk_mem:
running install
/usr/local/lib/python3.11/site-packages/setuptools/command/install.py:34: SetupToolsDeprecationWarning: setup.py install is deprecated. Use build and pip and other standards-based tools.
  warnings.warn(
/usr/local/lib/python3.11/site-packages/setuptools/command/easy_install.py:144: EasyInstallDeprecationWarning: easy_install command is deprecated. Use build and pip and other standards-based tools.
  warnings.warn(
running bdist_egg
running egg_info
creating elastalert2.egg-info
writing elastalert2.egg-info/PKG-INFO
writing dependency_links to elastalert2.egg-info/dependency_links.txt
writing entry points to elastalert2.egg-info/entry_points.txt
writing requirements to elastalert2.egg-info/requirements.txt
writing top-level names to elastalert2.egg-info/top_level.txt
writing manifest file 'elastalert2.egg-info/SOURCES.txt'
reading manifest file 'elastalert2.egg-info/SOURCES.txt'
adding license file 'LICENSE'
writing manifest file 'elastalert2.egg-info/SOURCES.txt'
installing library code to build/bdist.linux-x86_64/egg
```

Figure 40: Installing ElastAlert.

Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
elastalert_status	yellow	open	1	1	0	226b	
elastalert_status_silence	yellow	open	1	1	0	226b	
filebeat-7.17.9-2023.04.15	yellow	open	1	1	11946	8.6mb	
filebeat-7.17.9-2023.04.14	yellow	open	1	1	842	2.7mb	
filebeat-7.17.9-2023.04.16	yellow	open	1	1	33341	29.3mb	
elastalert_status_error	yellow	open	1	1	0	226b	
elastalert_status_status	yellow	open	1	1	0	226b	
elastalert_status_past	yellow	open	1	1	0	226b	
auditbeat-7.17.9-2023.04.16	yellow	open	1	1	68	271.4kb	
rsyslog-2023.04.16	yellow	open	1	1	307	82.9kb	

Figure 41: ElastAlert indices.

3.9.10 Telegram

Telegram messenger application was used to provide end user with alter message send by ElastAlert after it query with Elasticsearch and match pre-configured condition. Telegram bot token and chat room ID was used to connect telegram with ElastAlert. For this purpose, telegram application was installed on the device and a bot named “MISP_ALERT” was created. Telegram bot token was provided after bot was created.



```

Open + example_frequency.yaml ~elastalert2/examples/rules Save ×
#es_username: someusername
#es_password: somepassword

# (Required)
# Rule name, must be unique
name: Example frequency rule

# (Required)
# Type of alert.
# the frequency rule type alerts when num_events events occur with timeframe time
type: frequency

# (Required)
# Index to search, wildcard supported
index: filebeat-7.17.9-* # (Required, frequency specific)
# Alert when this many documents matching the query occur within a timeframe
num_events: 2 # (Required, frequency specific)
# num_events must occur within this amount of time to trigger an alert
timeframe:
  hours: 4

# (Required)
# A list of Elasticsearch filters used for find events
# These filters are joined with AND and nested in a filtered query
# For more info: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html
filter:
  - term:
      agent.hostname: "suricata" # condition to be matched for alert
# (Required)
# The alert is use when a match is found
alert:
  - "telegram"
    telegram_bot_token: "5622465916:AAGBpIaBwpjLNseac46BkyOH0QGyKdF_jNk"
    telegram_room_id: "6267921323"

```

YAML ▾ Tab Width: 8 ▾ Ln 1, Col 1 ▾ INS

Figure 42:Example_frequency.yaml. rule.

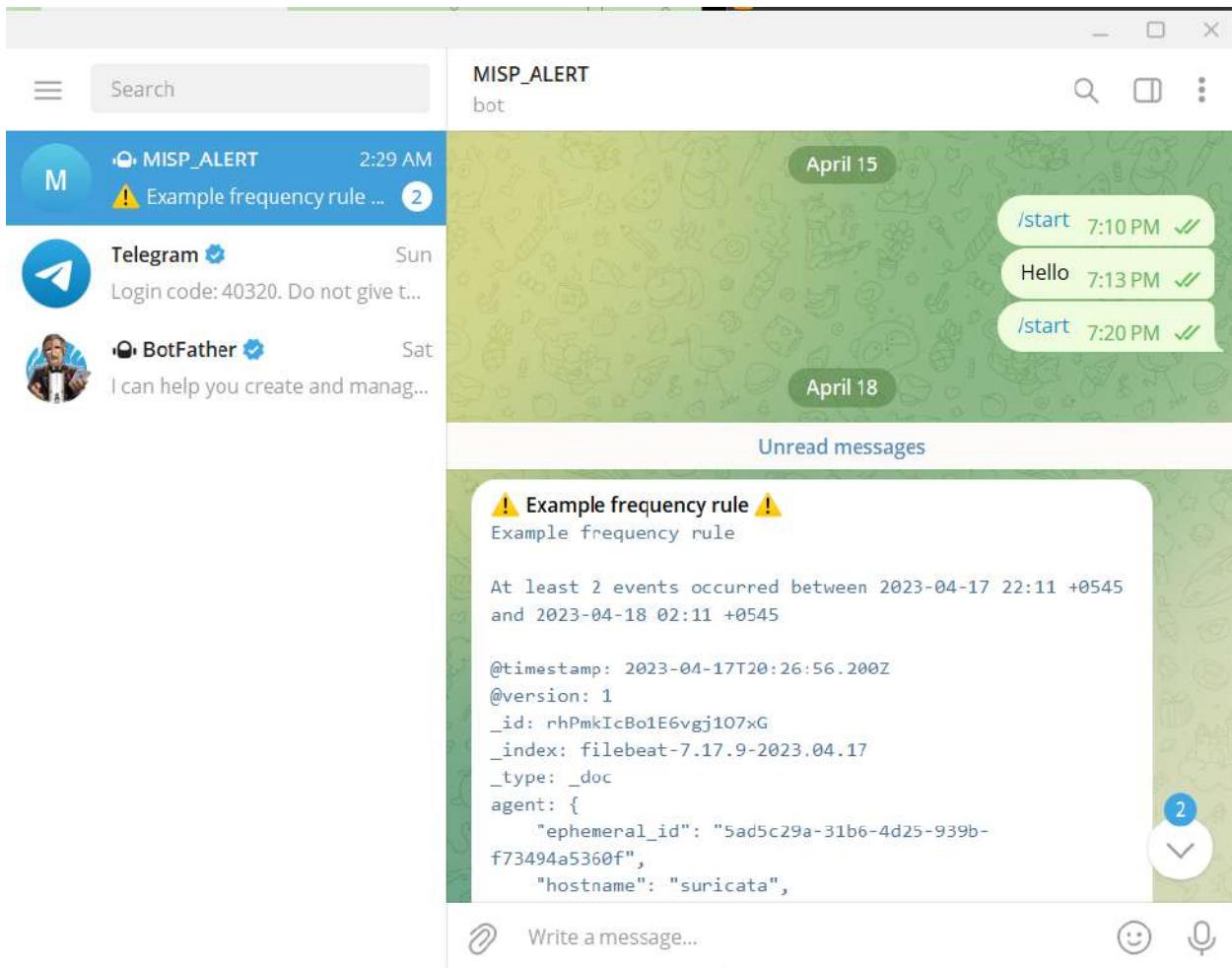


Figure 43: Telegram receiving log.

3.9.11 Script for pulling MISP Event Data

These are the YAML and python script file that were used to get MISP event IoCs like Ips, Hashes and domain to Memcached. Such IoCs are helpful in detecting threat in real time.

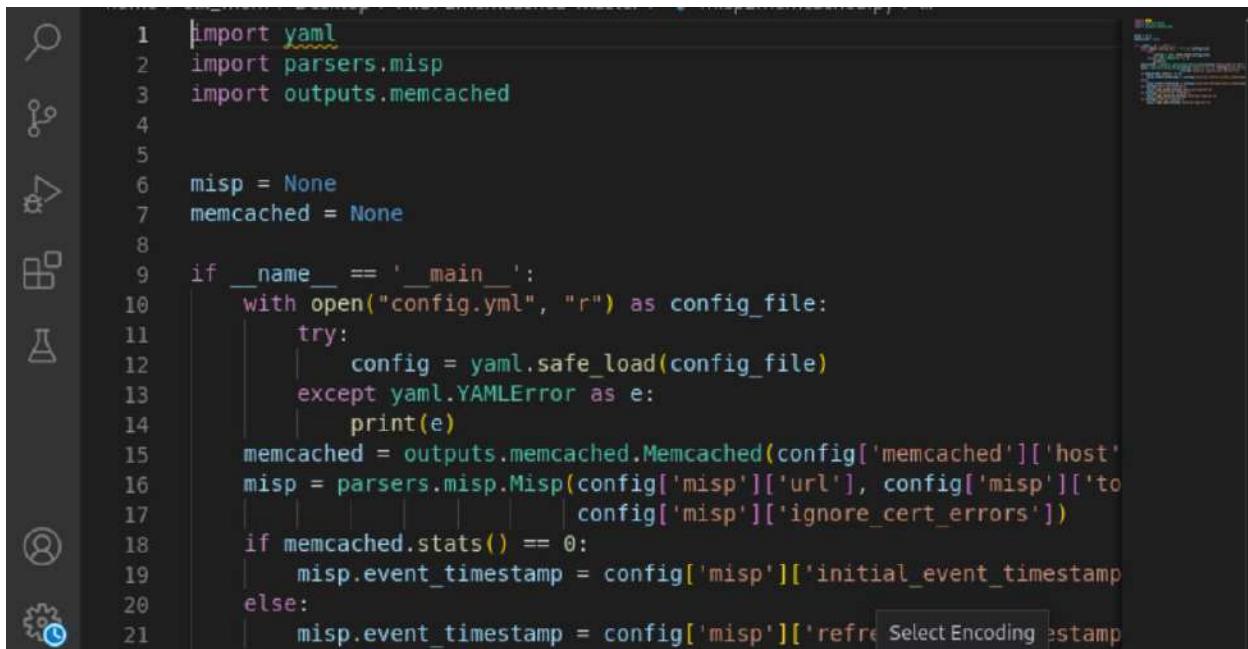


The screenshot shows a code editor window with the file name "config.yml" at the top right. Below the title bar, it says "~/Desktop/MISP2memcached-master". On the left side of the editor, there are two buttons: "Open" with a dropdown arrow and a plus sign button. The main area of the editor contains the following YAML configuration:

```
memcached:
  host: '10.10.30.3'
  port: '11211'
misp:
  url: 'https://10.10.30.10/attributes/restSearch'
  token: "rz2jVQYK0uaPC0uZ6tf3W4KzqIU2UkW4vC1v1WAN"
  ignore_cert_errors: true
  initial_event_timestamp: 365d
  refresh_event_timestamp: 1h
hash:
  enabled: true
  expires: 0
network:
  enabled: true
  expires: 0
web:
  enabled: true
  expires: 0
```

Figure 44: config.yml file.

In above figure, Memcached server's ip address and its listening port were mentioned as well as API and token to connect MISP server were given in YAML file.



```
1 import yaml
2 import parsers.misp
3 import outputs.memcached
4
5
6 misp = None
7 memcached = None
8
9 if __name__ == '__main__':
10     with open("config.yml", "r") as config_file:
11         try:
12             config = yaml.safe_load(config_file)
13         except yaml.YAMLError as e:
14             print(e)
15         memcached = outputs.memcached.Memcached(config['memcached']['host'])
16         misp = parsers.misp.Misp(config['misp']['url'], config['misp']['to'],
17                                  config['misp']['ignore_cert_errors'])
18         if memcached.stats() == 0:
19             misp.event_timestamp = config['misp']['initial_event_timestamp']
20         else:
21             misp.event_timestamp = config['misp']['refresh_timestamp']
```

Figure 45: Script for pulling misp data.

This Python script contains It imports modules from the "parsers" and "outputs" packages, as well as the PyYAML library, which allows it to interact with YAML configuration files. It loads a configuration file ("config.yml") that contains settings for connecting to a MISP instance and a Memcached server. It then makes Misp and Memcached instances with these settings, sending the latter as a parameter to the former. It also analyses the configuration file for the presence of hashing, network, and web settings, and if present, invokes Misp instance methods to load data from these sources into Memcached.

```

import requests
import json
from requests.packages.urllib3.exceptions import InsecureRequestWarning

class MisP:
    #url = "https://10.10.30.10"
    url = None
    token = None
    ignore_cert_errors = None
    memcached = None
    event_timestamp = None
    # MISP_API_KEY = 'rz2jVQYK0uaPC0uZ6tf3W4KzqIU2UkW4vC1vlWAN'

    def __init__(self, url, token, memcached, ignore_cert_errors=False):
        self.url = url
        self.token = token
        self.ignore_cert_errors = ignore_cert_errors
        self.memcached = memcached
        print("url: " + self.url)
        print("token: " + self.token)
        print("boolean: " + str(self.ignore_cert_errors))
        print("memcached" + str(self.memcached))

    def fetch_data(self, misp_types):
        headers = {
            'Authorization': self.token,
            'Accept': 'application/json',
            'Content-type': 'application/json',
        }
        data = ('{"returnFormat":"json",\n" type": {"OR":"' + json.dumps(misp_types) + '"},\n"to_ids":"yes",\n"event timestamp":"' + self.event_timestamp + '"}')
        if self.ignore_cert_errors:
            requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
        response = requests.post(self.url, headers=headers, data=data, verify=(not self.ignore_cert_errors))

```

Figure 46: Script for pulling misp data.

This python script helps to load IOCs (indicator of compromise) from the MISP server in to Memcached server. This script helps to fetch hash type, network type, web type IOCs from MISP and loads to Memcached. It uses MISP url and token to connect with MISP server ignoring SSL certificates errors.

```
import pymemcache

class Memcached:
    host = None
    port = None
    client = None

    def __init__(self, host, port):
        self.host = host
        self.port = port
        self.client = pymemcache.client.base.Client((self.host, self.port))
        memcache_value = self.client.get("misp-stats")
        if memcache_value is None:
            self.client.set("misp-stats", 0)

    def stats(self):
        memcache_value = self.client.get("misp-stats")
        return int(memcache_value)

    def insert(self, namespace, lookup_value, tag, expire):
        lookup_value = lookup_value.strip()
        print("lookupvalue: "+lookup_value)
        tag = tag.replace(',', '')
        print("tag: "+tag) # Remove "," since it's being used as separator in memcached
        key = namespace + ":" + lookup_value
        print("key: "+key)
        memcache_value = self.client.get(key)
        print("memcache value: " + str(memcache_value))
        if memcache_value:
            existing_tags = memcache_value.decode("utf-8").split(',')
            print("Existing tags for {}: {}".format(key, existing_tags))
            if tag not in existing_tags:
                self.client.append(key, "," + tag, expire=expire)
                return True
            else:
                return False
        else:
            self.client.set(key, tag, expire=expire)
            return True

```

Figure 47: script for set data to Memcached.

This script used for interaction with Memcached server using host and port value with the propose of storing key-value pair that has an expiry time.

```
1 def filter(event)
2   event_tmp = event.get("[enrich][tmp]")
3   if event_tmp.nil?
4     return [event]
5   else
6     ids = event_tmp.split(',')
7   end
8   ids.each do |item|
9     key_value = item.split("#")
10    if !event.get('[misp][event_id]')
11      event.set('[misp][event_id]', [key_value[0]])
12      event.set('[misp][type]', [key_value[1]])
13    else
14      event.set('[misp][event_id]', event.get('[misp][type]'))
15      event.set('[misp][type]', event.get('[misp][type]'))
16    end
17  end
18  return [event]
19end
```

Figure 48: ruby script filter used in Logstash.

This piece of code was written in ruby and used as filter in Logstash where it checks “event” parameter that has “[enrich][tmp]” key or it is nil. If it is nil it will return original event and if not then this will add “misp.type” and “misp.event_id” field which will be seen in Kibana dashboard

Chapter 4: Testing and Analysis

4.1 Test Plan

4.1.1 Unit Testing, Test Plan

Test Case	Objective
1	To test whether the ELK stack receiving logs from filebeat in Suricata VM.
2	To test whether the ELK stack receiving logs from filebeat in Apache webserver.
3	To test whether the ELK stack receiving logs from winlogbeat in windows 10.
4	To test whether the ELK stack receiving logs from auditbeat in windows 10.
5	To test USB detection Alerts on EKL server's discover dashboard.
6	To test file integrity status detection alerts on EKL server's discover dashboard
7	To test whether script was fetching IOCs from MISP and loading to Memcached
8	To test if IOCs loaded in Memcached can be lookup and detected in ELK's discover tab.
9	To whether ElastAlert can send message to Telegram

Table 1: Unit Testing Plan.

4.1.2 System Testing, Test Plan

Test Case	Objective
1	To check if log agents like winlogbeat, auditbeat, and filebeat are collecting logs and sending to Logstash and visualized in Kibana dashboard.
2	To test the system if it can detect malware based on hash and provide alert message in telegram.
3	To test the system if it can alert the user through telegram message when file is been modified in host system.
4	To test the system if it can alert the user through telegram message when USB devices plugged in to system.
5	To test for verifying if all the ELK services are running properly without errors.

Table 2: System Testing Plan.

4.2 Unit Testing

4.2.1 Test Case 1

Test Case 1	
Objective	To test whether the ELK stack receiving logs from filebeat in Suricata VM.
Action	Started filebeat service in Suricata VM and Logstash service in ELKMemcached server. Verified if Logstash was listening in port 5044 and filebeat was started.
Expected Test Result	"filebeat-7.17.9-" filename log will be seen in index management tab of Kibana.
Actual Test Result	"filebeat-7.17.9-" filename log was seen in index management tab of Kibana.
Conclusion	Test was Successful.

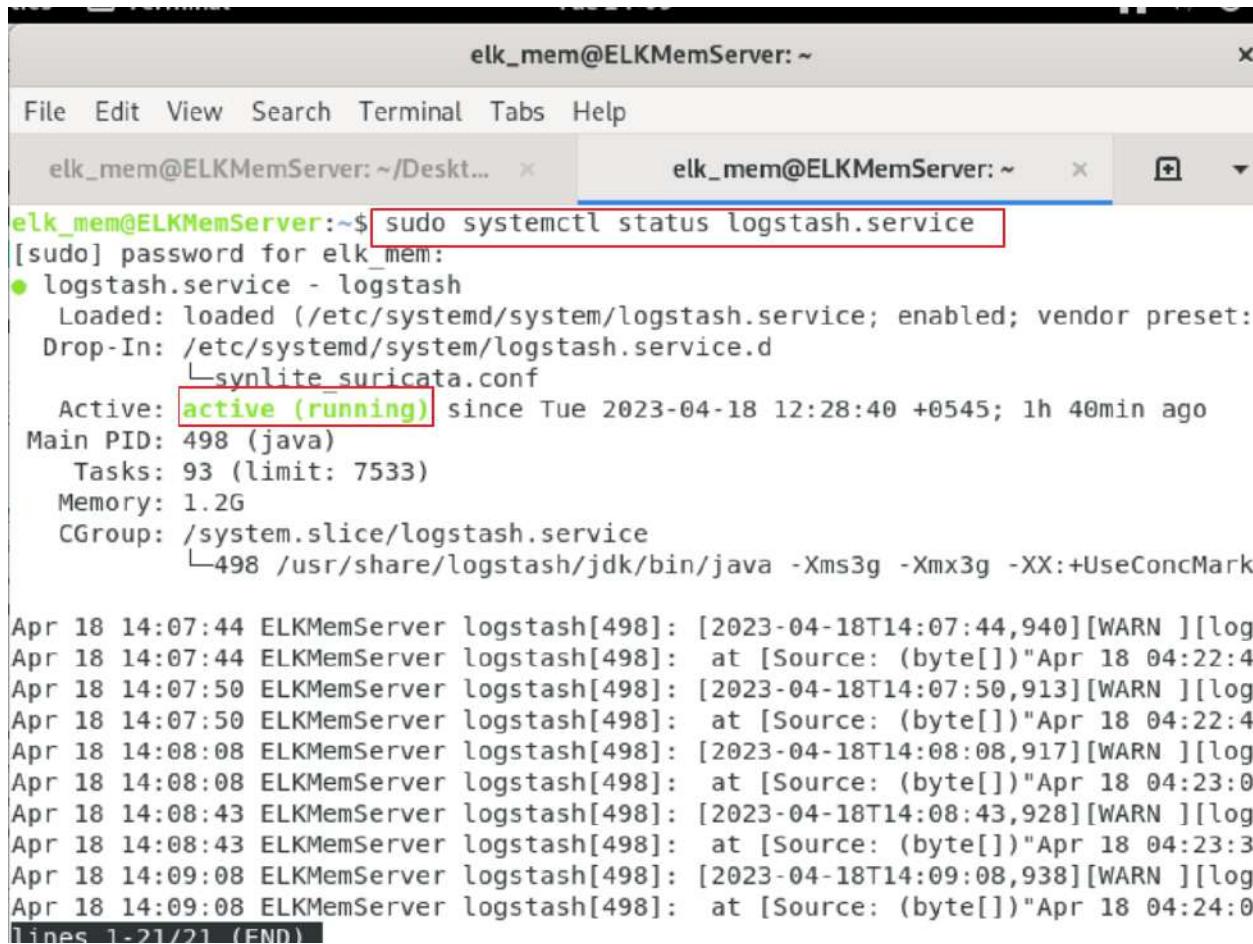
Table 3: Test Case 1

The screenshot shows a terminal window titled "nids@suricata: ~". The user has run the command "sudo systemctl start filebeat.service" and then checked its status with "sudo systemctl status filebeat.service". The output indicates the service is active and running. Below this, several log entries from filebeat are displayed, showing various log levels (WAR, ERR, INF) and timestamps. The terminal also shows the command "Lines 1-20/20 (END)" at the bottom.

```
nids@suricata:~$ sudo systemctl start filebeat.service
nids@suricata:~$ sudo systemctl status filebeat.service
● filebeat.service - Filebeat sends log files to Logstash or directly to Elastic
  Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset:
  Active: active (running) since Tue 2023-04-18 04:20:56 EDT; 1min 44s ago
    Docs: https://www.elastic.co/beats/filebeat
   Main PID: 2157 (filebeat)
     Tasks: 7 (limit: 1283)
    Memory: 120.9M
      CGroup: /system.slice/filebeat.service
              └─2157 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc

Apr 18 04:22:18 suricata filebeat[2157]: 2023-04-18T04:22:18.998-0400      WAR
Apr 18 04:22:18 suricata filebeat[2157]: 2023-04-18T04:22:18.998-0400      ERR
Apr 18 04:22:18 suricata filebeat[2157]: 2023-04-18T04:22:18.998-0400      INF
Apr 18 04:22:18 suricata filebeat[2157]: 2023-04-18T04:22:18.999-0400      INF
Apr 18 04:22:18 suricata filebeat[2157]: 2023-04-18T04:22:18.999-0400      INF
Apr 18 04:22:20 suricata filebeat[2157]: 2023-04-18T04:22:20.006-0400      INF
Apr 18 04:22:20 suricata filebeat[2157]: 2023-04-18T04:22:20.007-0400      INF
Apr 18 04:22:20 suricata filebeat[2157]: 2023-04-18T04:22:20.007-0400      INF
Apr 18 04:22:20 suricata filebeat[2157]: 2023-04-18T04:22:20.013-0400      INF
Apr 18 04:22:30 suricata filebeat[2157]: 2023-04-18T04:22:30.204-0400      INF
Lines 1-20/20 (END)
```

Figure 49: Starting filebeat service and verifying its status in Suricata VM.



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". It contains two tabs, both labeled "elk_mem@ELKMemServer: ~/Desktop...". The second tab is active. The command "sudo systemctl status logstash.service" is being run in the terminal. The output shows the logstash service is active and running. The log section at the bottom displays several warning messages from April 18, 2023, at 14:07:44, indicating issues with logstash[498].

```
elk_mem@ELKMemServer:~$ sudo systemctl status logstash.service
[sudo] password for elk_mem:
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: Drop-In: /etc/systemd/system/logstash.service.d
         └─synlrite_suricata.conf
   Active: active (running) since Tue 2023-04-18 12:28:40 +0545; 1h 40min ago
     Main PID: 498 (java)
        Tasks: 93 (limit: 7533)
       Memory: 1.2G
      CGroup: /system.slice/logstash.service
              └─498 /usr/share/logstash/jdk/bin/java -Xms3g -Xmx3g -XX:+UseConcMark

Apr 18 14:07:44 ELKMemServer logstash[498]: [2023-04-18T14:07:44,940][WARN ][log]
Apr 18 14:07:44 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:22:4
Apr 18 14:07:50 ELKMemServer logstash[498]: [2023-04-18T14:07:50,913][WARN ][log]
Apr 18 14:07:50 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:22:4
Apr 18 14:08:08 ELKMemServer logstash[498]: [2023-04-18T14:08:08,917][WARN ][log]
Apr 18 14:08:08 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:23:0
Apr 18 14:08:43 ELKMemServer logstash[498]: [2023-04-18T14:08:43,928][WARN ][log]
Apr 18 14:08:43 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:23:3
Apr 18 14:09:08 ELKMemServer logstash[498]: [2023-04-18T14:09:08,938][WARN ][log]
Apr 18 14:09:08 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:24:0
Lines 1-21/21 (END)
```

Figure 50: Verifying status of Logstash service.

```

[2023-04-18T14:11:54,626][INFO ][logstash.javapipeline] [[audit]] Starting pipeline {:pipeline_id=>"audit", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max_inflight"=>500, "pipeline.sources"=>["/etc/logstash/winAudit/auditConfig.conf"], :thread=>"#<Thread:0x3365beld@/usr/share/logstash/logstash-core/lib/logstash/java_pipeline:138 run>"}
[2023-04-18T14:11:54,628][INFO ][logstash.javapipeline] [[apache]] Starting pipeline {:pipeline_id=>"apache", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max_inflight"=>500, "pipeline.sources"=>["/etc/logstash/conf.d/apache.conf"], :thread=>"#<Thread:0x60cf26df run>"}
[2023-04-18T14:11:55,357][INFO ][logstash.javapipeline] [[audit]] Pipeline Java execution initialization time {"seconds"=>0.72}
[2023-04-18T14:11:55,400][INFO ][logstash.javapipeline] [[suricata]] Pipeline Java execution initialization time {"seconds"=>0.77}
[2023-04-18T14:11:55,420][INFO ][logstash.javapipeline] [[win]] Pipeline Java execution initialization time {"seconds"=>0.8}
[[suri]] Starting input listener {:address=>"0.0.0.0:5044"}
[[win]] Starting input listener {:address=>"0.0.0.0:5045"}
[[audit]] Starting input listener {:address=>"0.0.0.0:5043"}
[[apache]] Pipeline Java execution initialization time {"seconds"=>0.86}
[2023-04-18T14:11:55,471][INFO ][logstash.inputs.beats]
[2023-04-18T14:11:55,471][INFO ][logstash.inputs.beats]
[2023-04-18T14:11:55,470][INFO ][logstash.inputs.beats]
[2023-04-18T14:11:55,486][INFO ][logstash.javapipeline]
[2023-04-18T14:11:55,497][INFO ][logstash.javapipeline]
[2023-04-18T14:11:55,499][INFO ][logstash.inputs.beats]
[2023-04-18T14:11:55,500][INFO ][logstash.javapipeline]
[2023-04-18T14:11:55,503][INFO ][logstash.javapipeline]
[[audit]] Pipeline started {"pipeline.id"=>"audit"}
[[apache]] Starting input listener {:address=>"0.0.0.0:5042"}
[[apache]] Pipeline started {"pipeline.id"=>"apache"}
[[suricata]] Pipeline started {"pipeline.id"=>"suricata"}

```

Figure 51: Verifying Logstash was listening in port 5044 allocated for Suricata log.

Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
elastalert_status	yellow	open	1	1	54	158.6kb	
elastalert_status_silence	yellow	open	1	1	54	43kb	
filebeat-7.17.0-2023.04.18	yellow	open	1	1	395	1.7mb	
elastalert_status_error	yellow	open	1	1	0	226b	
elastalert_status_status	yellow	open	1	1	167	51kb	
elastalert_status_past	yellow	open	1	1	0	226b	

Figure 52: Filebeat log was seen in index management.

4.2.2 Test Case 2

Test Case 2	
Objective	To test whether the ELK stack receiving logs from filebeat in Apache webserver.
Action	Started winlogbeat service in Apache web server VM and Logstash service in ELKMemcached server. Verified if Logstash was listening in port 5042 and filebeat was started.
Expected Test Result	"apachelog-*" filename log will be seen in index management tab of Kibana.
Actual Test Result	"apachelog-*" filename log was seen in index management tab of Kibana.
Conclusion	Test was Successful.

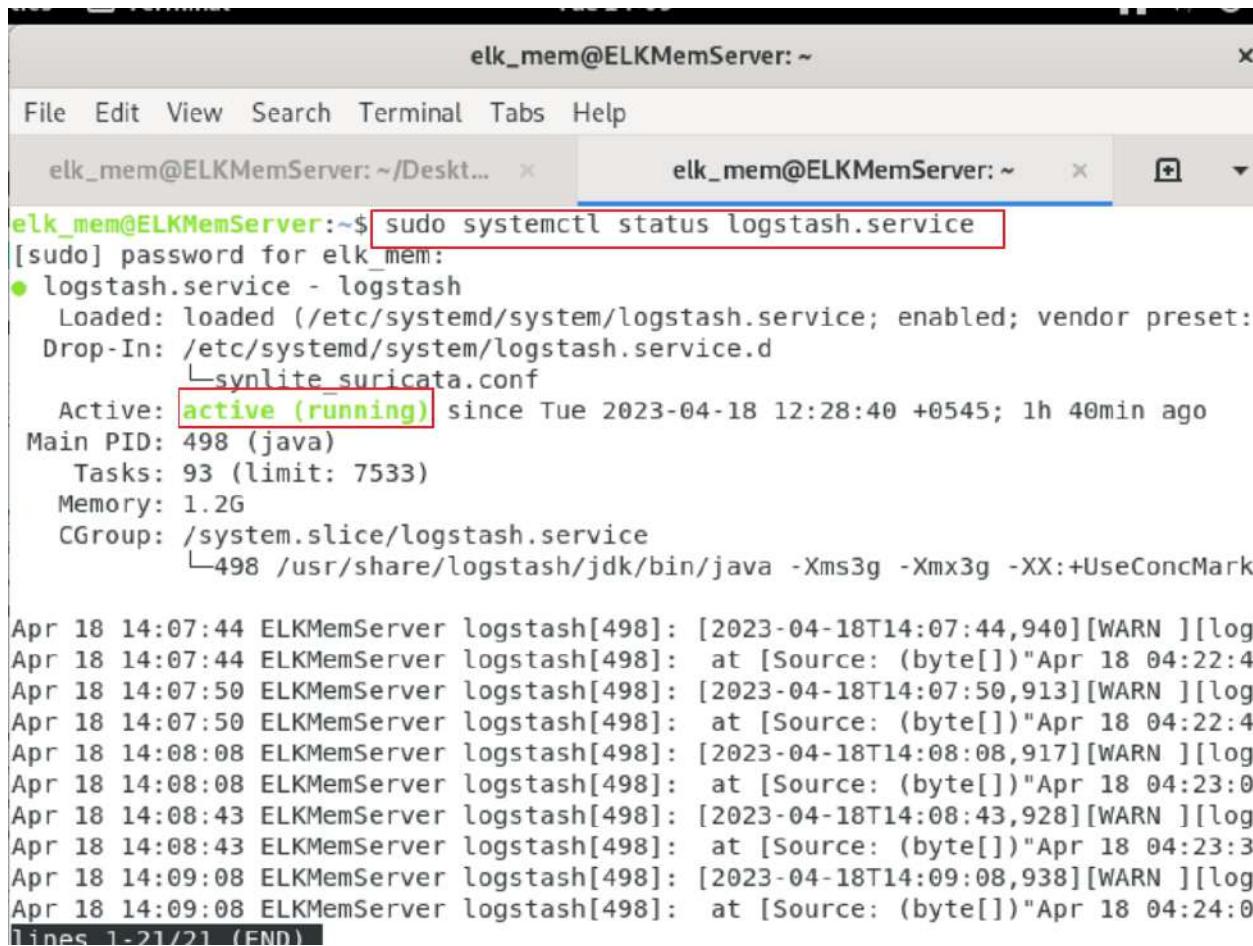
Table 4: Test Case 2

The screenshot shows a terminal window titled "apchserver@apchserver: ~". The window contains the following text:

```
File Edit View Search Terminal Help
apchserver@apchserver:~$ sudo systemctl start filebeat.service
apchserver@apchserver:~$ sudo systemctl status filebeat.service
● filebeat.service - Filebeat sends log files to Logstash or directly to Elastic
  Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset:
  Active: active (running) since Tue 2023-04-18 09:54:29 BST; 8s ago
    Docs: https://www.elastic.co/beats/filebeat
      Main PID: 2240 (filebeat)
        Tasks: 8 (limit: 1194)
       Memory: 100.1M
         CGroup: /system.slice/filebeat.service
                   └─2240 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc

Apr 18 09:54:32 apchserver filebeat[2240]: 2023-04-18T09:54:32.916+0100 I
Apr 18 09:54:32 apchserver filebeat[2240]: 2023-04-18T09:54:32.918+0100 I
Apr 18 09:54:32 apchserver filebeat[2240]: 2023-04-18T09:54:32.919+0100 I
Apr 18 09:54:32 apchserver filebeat[2240]: 2023-04-18T09:54:32.919+0100 I
Apr 18 09:54:35 apchserver filebeat[2240]: 2023-04-18T09:54:35.857+0100 I
lines 1-20/20 (END)
```

Figure 53: Starting filebeat service and verifying its status in Apache web server.

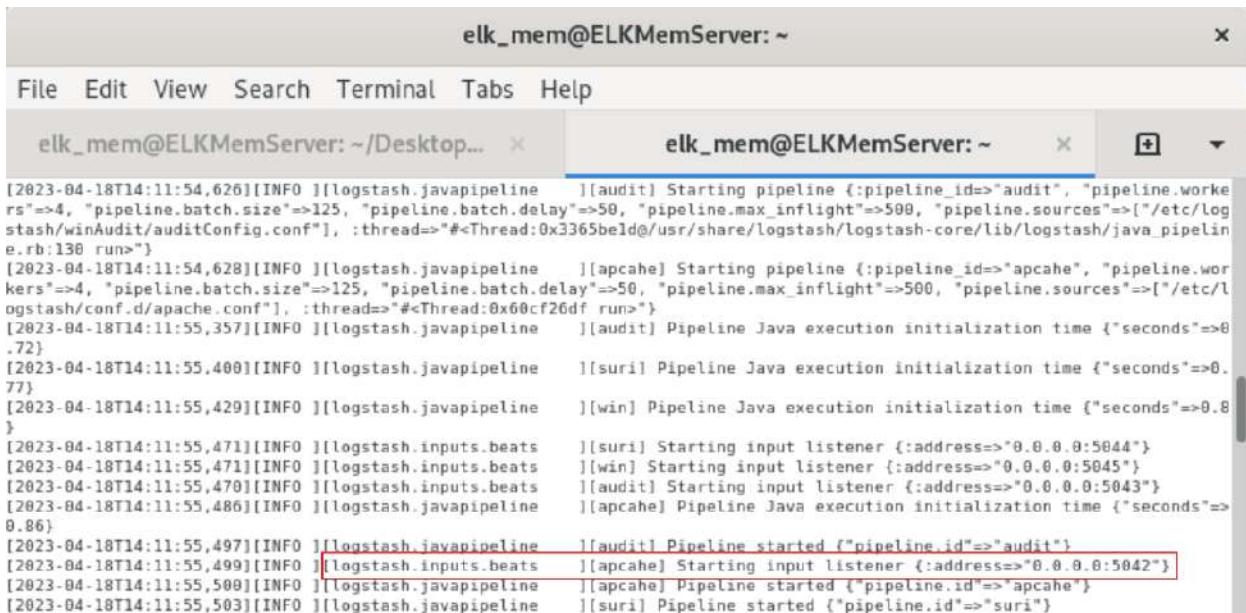


The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window has two tabs: "elk_mem@ELKMemServer: ~/Deskt..." and "elk_mem@ELKMemServer: ~". The second tab is active. The command "sudo systemctl status logstash.service" is being run in the terminal. The output shows the service is loaded, enabled, and active (running) since April 18, 2023. It also lists the main PID (498), tasks (93), memory usage (1.2G), and the CGroup path. Below the service status, there is a log output from April 18, 2023, showing several WARN messages. The log ends with "Lines 1-21/21 (END)".

```
elk_mem@ELKMemServer:~$ sudo systemctl status logstash.service
[sudo] password for elk_mem:
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: Drop-In: /etc/systemd/system/logstash.service.d
           └─synlrite_suricata.conf
   Active: active (running) since Tue 2023-04-18 12:28:40 +0545; 1h 40min ago
     Main PID: 498 (java)
        Tasks: 93 (limit: 7533)
       Memory: 1.2G
      CGroup: /system.slice/logstash.service
              └─498 /usr/share/logstash/jdk/bin/java -Xms3g -Xmx3g -XX:+UseConcMark

Apr 18 14:07:44 ELKMemServer logstash[498]: [2023-04-18T14:07:44,940][WARN ][log]
Apr 18 14:07:44 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:22:4
Apr 18 14:07:50 ELKMemServer logstash[498]: [2023-04-18T14:07:50,913][WARN ][log]
Apr 18 14:07:50 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:22:4
Apr 18 14:08:08 ELKMemServer logstash[498]: [2023-04-18T14:08:08,917][WARN ][log]
Apr 18 14:08:08 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:23:0
Apr 18 14:08:43 ELKMemServer logstash[498]: [2023-04-18T14:08:43,928][WARN ][log]
Apr 18 14:08:43 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:23:3
Apr 18 14:09:08 ELKMemServer logstash[498]: [2023-04-18T14:09:08,938][WARN ][log]
Apr 18 14:09:08 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:24:0
Lines 1-21/21 (END)
```

Figure 54: Verifying status of Logstash service.



```
[2023-04-18T14:11:54,626][INFO ][logstash.javapipeline ]{[audit]} Starting pipeline {:pipeline_id=>"audit", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max_inflight"=>500, "pipeline.sources"=>["/etc/logstash/winAudit/auditConfig.conf"], :thread=>"#<Thread:0x3365beld@/usr/share/logstash/logstash-core/lib/logstash/java_pipeline.rb:138 run>"}  
[2023-04-18T14:11:54,628][INFO ][logstash.javapipeline ]{[apcahe]} Starting pipeline {:pipeline_id=>"apcahe", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max_inflight"=>500, "pipeline.sources"=>["/etc/logstash/conf.d/apache.conf"], :thread=>"#<Thread:0x60cf26df run>"}  
[2023-04-18T14:11:55,357][INFO ][logstash.javapipeline ]{[audit]} Pipeline Java execution initialization time {"seconds"=>0.72}  
[2023-04-18T14:11:55,400][INFO ][logstash.javapipeline ]{[suril]} Pipeline Java execution initialization time {"seconds"=>0.77}  
[2023-04-18T14:11:55,420][INFO ][logstash.javapipeline ]{[win]} Pipeline Java execution initialization time {"seconds"=>0.8}  
[2023-04-18T14:11:55,471][INFO ][logstash.inputs.beats ]{[suri]} Starting input listener {:address=>"0.0.0.0:5044"}  
[2023-04-18T14:11:55,471][INFO ][logstash.inputs.beats ]{[win]} Starting input listener {:address=>"0.0.0.0:5045"}  
[2023-04-18T14:11:55,470][INFO ][logstash.inputs.beats ]{[audit]} Starting input listener {:address=>"0.0.0.0:5043"}  
[2023-04-18T14:11:55,486][INFO ][logstash.javapipeline ]{[apcahe]} Pipeline Java execution initialization time {"seconds"=>0.86}  
[2023-04-18T14:11:55,497][INFO ][logstash.javapipeline ]{[audit]} Pipeline started {"pipeline_id"=>"audit"}  
[2023-04-18T14:11:55,499][INFO ][logstash.inputs.beats ]{[apcahe]} Starting input listener {:address=>"0.0.0.0:5042"}  
[2023-04-18T14:11:55,500][INFO ][logstash.javapipeline ]{[apcahe]} Pipeline started {"pipeline_id"=>"apcahe"}  
[2023-04-18T14:11:55,503][INFO ][logstash.javapipeline ]{[suril]} Pipeline started {"pipeline_id"=>"suri"}
```

Figure 55: Verifying Logstash was listening in port 5042 allocated for Apache log.

The screenshot shows the Elasticsearch Index Management interface. On the left, there's a sidebar with sections for Management, Ingest, Data, Alerts and Insights, and Security. The main area has tabs for Indices, Data Streams, Index Templates, and Component Templates, with Indices selected. A search bar and filters for Lifecycle status and phase are at the top. Below is a table of indices:

Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
elastalert_status	yellow	open	1	1	54	158.6kb	
elastalert_status_silence	yellow	open	1	1	54	43kb	
filebeat-7.17.9-2023.04.18	yellow	open	1	1	650	2.7mb	
elastalert_status_error	yellow	open	1	1	0	226b	
elastalert_status_status	yellow	open	1	1	167	51kb	
elastalert_status_past	yellow	open	1	1	0	226b	
apache.log-2023.04.18	yellow	open	1	1	8	57.7kb	

A red box highlights the row for 'apache.log-2023.04.18'. At the bottom, there are pagination controls and a note about rows per page.

Figure 56: Apache log was seen in index management.

4.2.3 Test Case 3

Test Case 3	
Objective	To test whether the ELK stack receiving logs from winlogbeat in windows 10.
Action	Started winlogbeat service in windows 10 VM and Logstash service in ELKMemcached server. Verified if Logstash was listening in port 5045 and winlogbeat was started.
Expected Test Result	“winlogbeat-7.17.9-” filename log will be seen in index management tab of Kibana.
Actual Test Result	“winlogbeat-7.17.9-” filename log was seen in index management tab of Kibana.
Conclusion	Test was Successful.

Table 5: Test Case 3

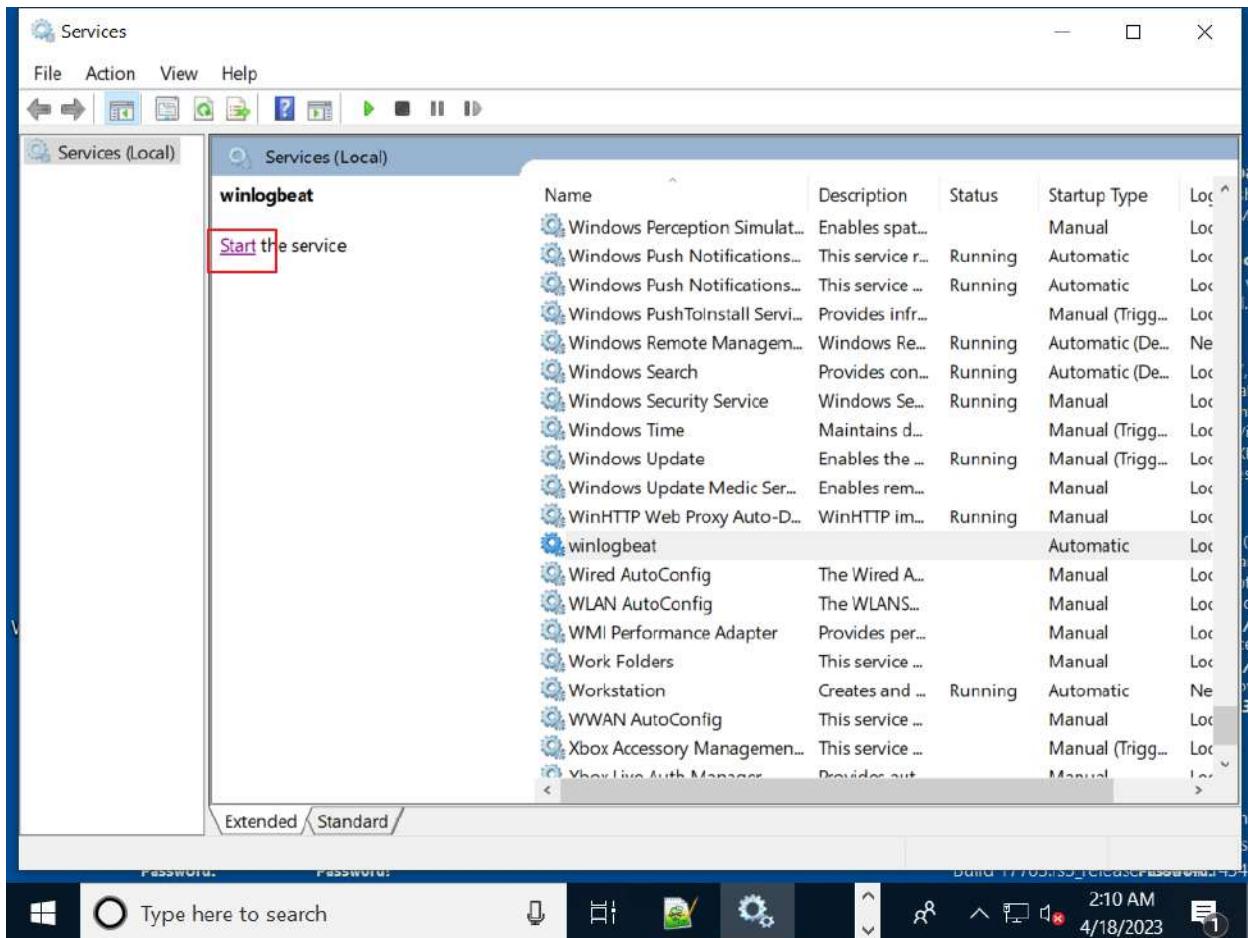


Figure 57: Starting Winlogbeat.

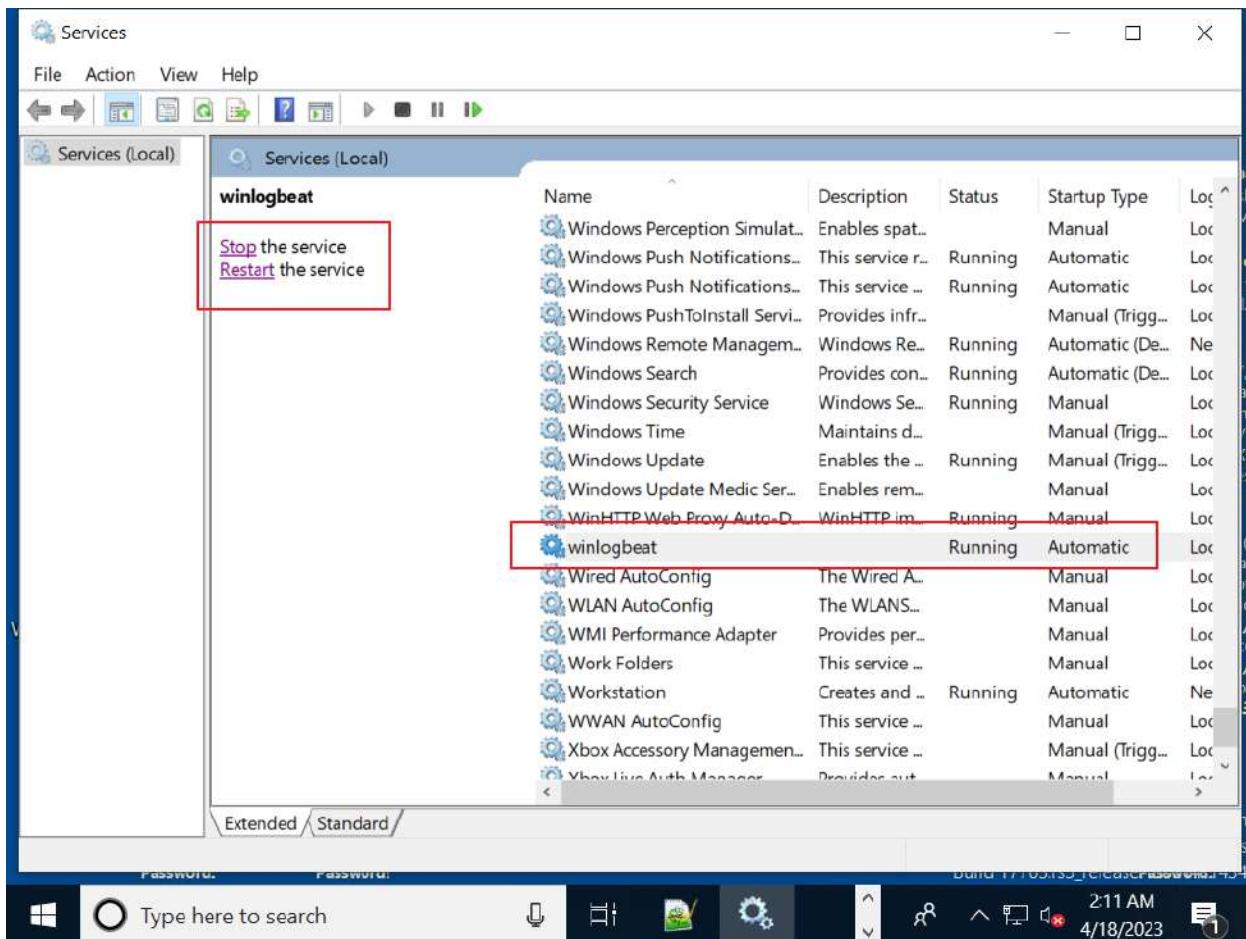
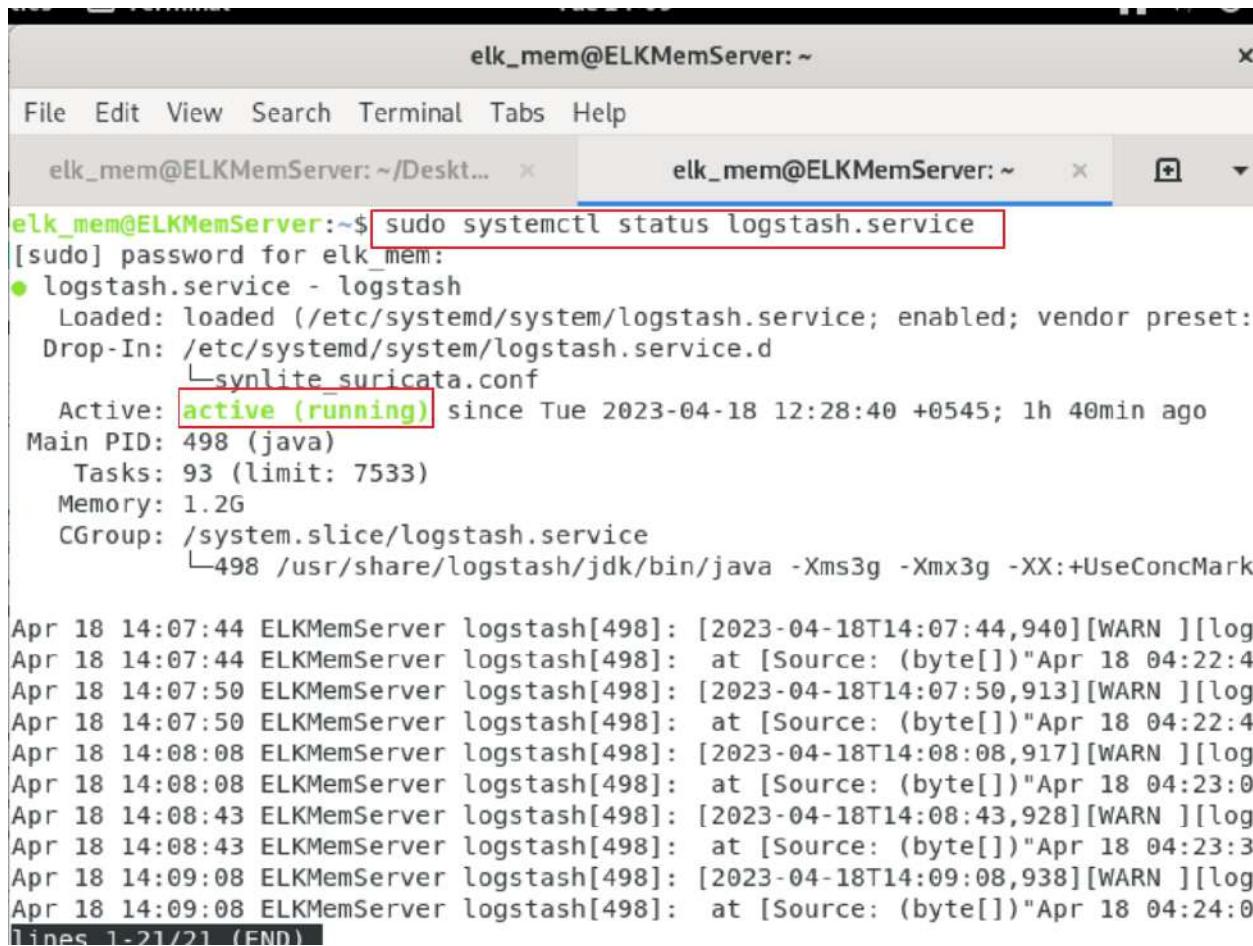


Figure 58: Winlogbeat started.

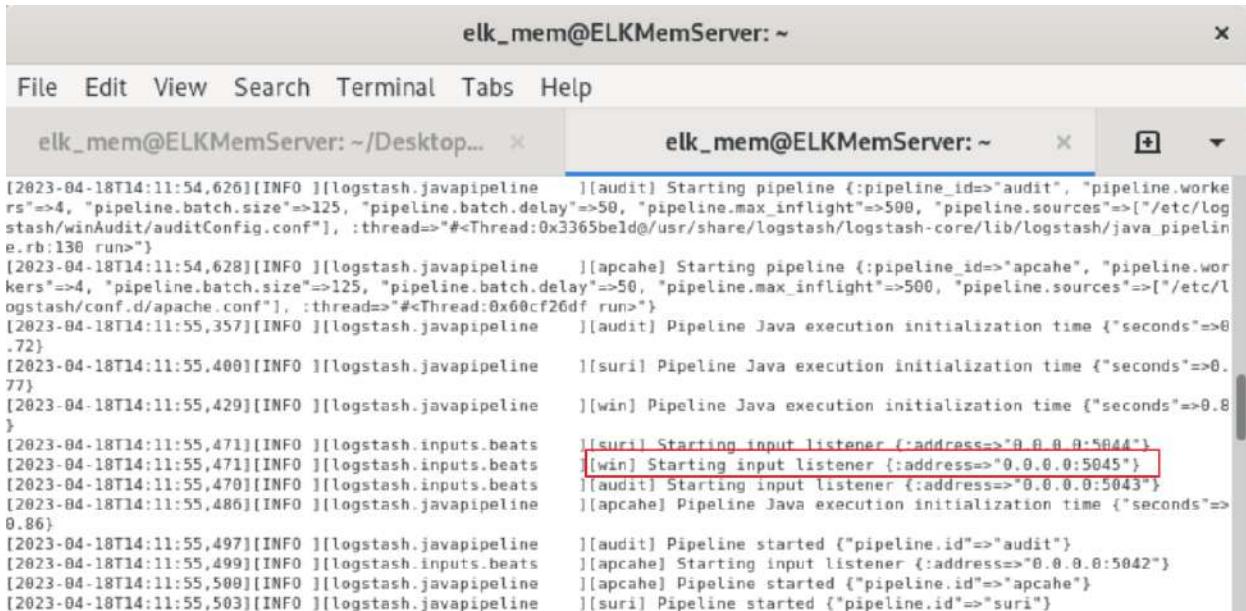


The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window has two tabs: "elk_mem@ELKMemServer: ~/Deskt..." and "elk_mem@ELKMemServer: ~". The second tab is active. The command "sudo systemctl status logstash.service" is being run in the terminal. The output shows the logstash service is loaded, enabled, and active (running) since April 18, 2023. It also lists memory usage and a Java process ID (498). Below the service status, there is a log of errors from the logstash application, spanning from April 18, 14:07:44 to 14:09:08. The log entries are mostly [WARN] messages related to byte[] arrays and timestamps.

```
elk_mem@ELKMemServer:~$ sudo systemctl status logstash.service
[sudo] password for elk_mem:
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: 
   Drop-In: /etc/systemd/system/logstash.service.d
     └─synlrite_suricata.conf
   Active: active (running) since Tue 2023-04-18 12:28:40 +0545; 1h 40min ago
     Main PID: 498 (java)
       Tasks: 93 (limit: 7533)
      Memory: 1.2G
        CGroup: /system.slice/logstash.service
                  └─498 /usr/share/logstash/jdk/bin/java -Xms3g -Xmx3g -XX:+UseConcMark

Apr 18 14:07:44 ELKMemServer logstash[498]: [2023-04-18T14:07:44,940][WARN ][log]
Apr 18 14:07:44 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:22:4
Apr 18 14:07:50 ELKMemServer logstash[498]: [2023-04-18T14:07:50,913][WARN ][log]
Apr 18 14:07:50 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:22:4
Apr 18 14:08:08 ELKMemServer logstash[498]: [2023-04-18T14:08:08,917][WARN ][log]
Apr 18 14:08:08 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:23:0
Apr 18 14:08:43 ELKMemServer logstash[498]: [2023-04-18T14:08:43,928][WARN ][log]
Apr 18 14:08:43 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:23:3
Apr 18 14:09:08 ELKMemServer logstash[498]: [2023-04-18T14:09:08,938][WARN ][log]
Apr 18 14:09:08 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:24:0
Lines 1-21/21 (END)
```

Figure 59: Verifying status of Logstash service.



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". It has two tabs open: "elk_mem@ELKMemServer: ~/Desktop..." and "elk_mem@ELKMemServer: ~". The second tab is active and displays Logstash startup logs. The logs show the configuration files being loaded, pipelines starting, and input listeners being initialized. Several log entries are highlighted with red boxes:

- [2023-04-18T14:11:55,420][INFO][logstash.javapipeline] [[apcahe] Starting input listener {:address=>"0.0.0.0:5045"}]
- [2023-04-18T14:11:55,420][INFO][logstash.javapipeline] [[win] Starting input listener {:address=>"0.0.0.0:5043"}]
- [2023-04-18T14:11:55,471][INFO][logstash.inputs.beats] [[suril] Starting input listener {:address=>"0.0.0.0:5044"}]

```
[2023-04-18T14:11:54,626][INFO ][logstash.javapipeline] [[audit] Starting pipeline {:pipeline_id=>"audit", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max_inflight"=>500, "pipeline.sources"=>["/etc/logstash/winAudit/auditConfig.conf"], :thread=>"#<Thread:0x3365beld@/usr/share/logstash/logstash-core/lib/logstash/java_pipeline.rb:138 run"}]
[2023-04-18T14:11:54,628][INFO ][logstash.javapipeline] [[apcahe] Starting pipeline {:pipeline_id=>"apcahe", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max_inflight"=>500, "pipeline.sources"=>["/etc/logstash/conf.d/apache.conf"]}, :thread=>"#<Thread:0x60cf26df run"}]
[2023-04-18T14:11:55,357][INFO ][logstash.javapipeline] [[audit] Pipeline Java execution initialization time {"seconds"=>0.72}
[2023-04-18T14:11:55,400][INFO ][logstash.javapipeline] [[suril] Pipeline Java execution initialization time {"seconds"=>0.77}
[2023-04-18T14:11:55,420][INFO ][logstash.javapipeline] [[win] Pipeline Java execution initialization time {"seconds"=>0.8}
[2023-04-18T14:11:55,471][INFO ][logstash.inputs.beats] [[suril] Starting input listener {:address=>"0.0.0.0:5044"}]
[2023-04-18T14:11:55,471][INFO ][logstash.inputs.beats] [[win] Starting input listener {:address=>"0.0.0.0:5045"}]
[2023-04-18T14:11:55,470][INFO ][logstash.inputs.beats] [[audit] Starting input listener {:address=>"0.0.0.0:5043"}]
[2023-04-18T14:11:55,486][INFO ][logstash.javapipeline] [[apcahe] Pipeline Java execution initialization time {"seconds"=>0.86}
[2023-04-18T14:11:55,497][INFO ][logstash.javapipeline] [[audit] Pipeline started {"pipeline.id"=>"audit"}]
[2023-04-18T14:11:55,499][INFO ][logstash.inputs.beats] [[apcahe] Starting input listener {:address=>"0.0.0.0:5042"}]
[2023-04-18T14:11:55,500][INFO ][logstash.javapipeline] [[apcahe] Pipeline started {"pipeline.id"=>"apcahe"}]
[2023-04-18T14:11:55,503][INFO ][logstash.javapipeline] [[suril] Pipeline started {"pipeline.id"=>"suril"}]
```

Figure 60: Verifying Logstash was listening in port 5045 allocated for winlogbeat.

The screenshot shows the Elasticsearch management interface in Firefox ESR. The title bar indicates it's Tuesday at 15:00. A tooltip message says "Automatic suspend Computer will suspend very soon because of inactivity." The URL in the address bar is 10.10.50.5:5601/app/management/data/index_management.

The left sidebar has a navigation menu with sections: Management, Ingest, Data, Index Management (which is selected and highlighted in blue), Alerts and Insights, and Security. The main content area is titled "Management" and "Index Management". It displays a table of indices with the following columns: Name, Health, Status, Primaries, Replicas, Docs count, Storage size, and Data stream. The table lists several indices, including elastalert_status, elastalert_status_silence, filebeat-7.17.9-2023.04.18, elastalert_status_error, elastalert_status_status, elastalert_status_past, auditbeat-7.17.9-2023.04.18, apache2023.04.18, and winlogbeat-7.17.9-2023.04.18. The winlogbeat index is highlighted with a red border around its row.

Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
elastalert_status	yellow	open	1	1	54	158.6kb	
elastalert_status_silence	yellow	open	1	1	54	43kb	
filebeat-7.17.9-2023.04.18	yellow	open	1	1	650	2.7mb	
elastalert_status_error	yellow	open	1	1	0	226b	
elastalert_status_status	yellow	open	1	1	167	51kb	
elastalert_status_past	yellow	open	1	1	0	226b	
auditbeat-7.17.9-2023.04.18	yellow	open	1	1	133	281.3kb	
apache2023.04.18	yellow	open	1	1	8	57.7kb	
winlogbeat-7.17.9-2023.04.18	yellow	open	1	1	1197	1.9mb	

Figure 61: winlogbeat log was seen in index management.

4.2.4 Test case 4

Test Case 4	
Objective	To test whether the ELK stack receiving logs from auditbeat in windows 10.
Action	Started auditbeat service in windows 10 VM and Logstash service in ELKMemcached server. Verified if Logstash was listening in port 5043 and audiotbeat was started.
Expected Test Result	"auditbeat-7.17.9-" filename log will be seen in index management tab of Kibana.
Actual Test Result	"auditbeat-7.17.9-" filename log was seen in index management tab of Kibana.
Conclusion	Test was Successful.

Table 6: Test Case 4

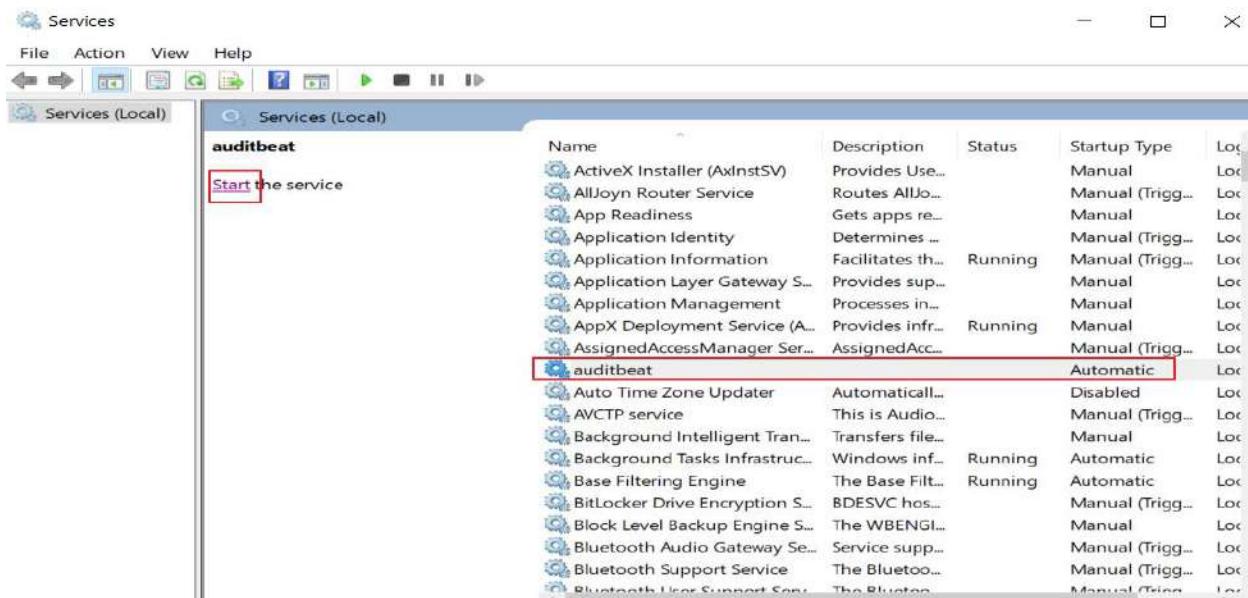


Figure 62: Starting auditbeat.

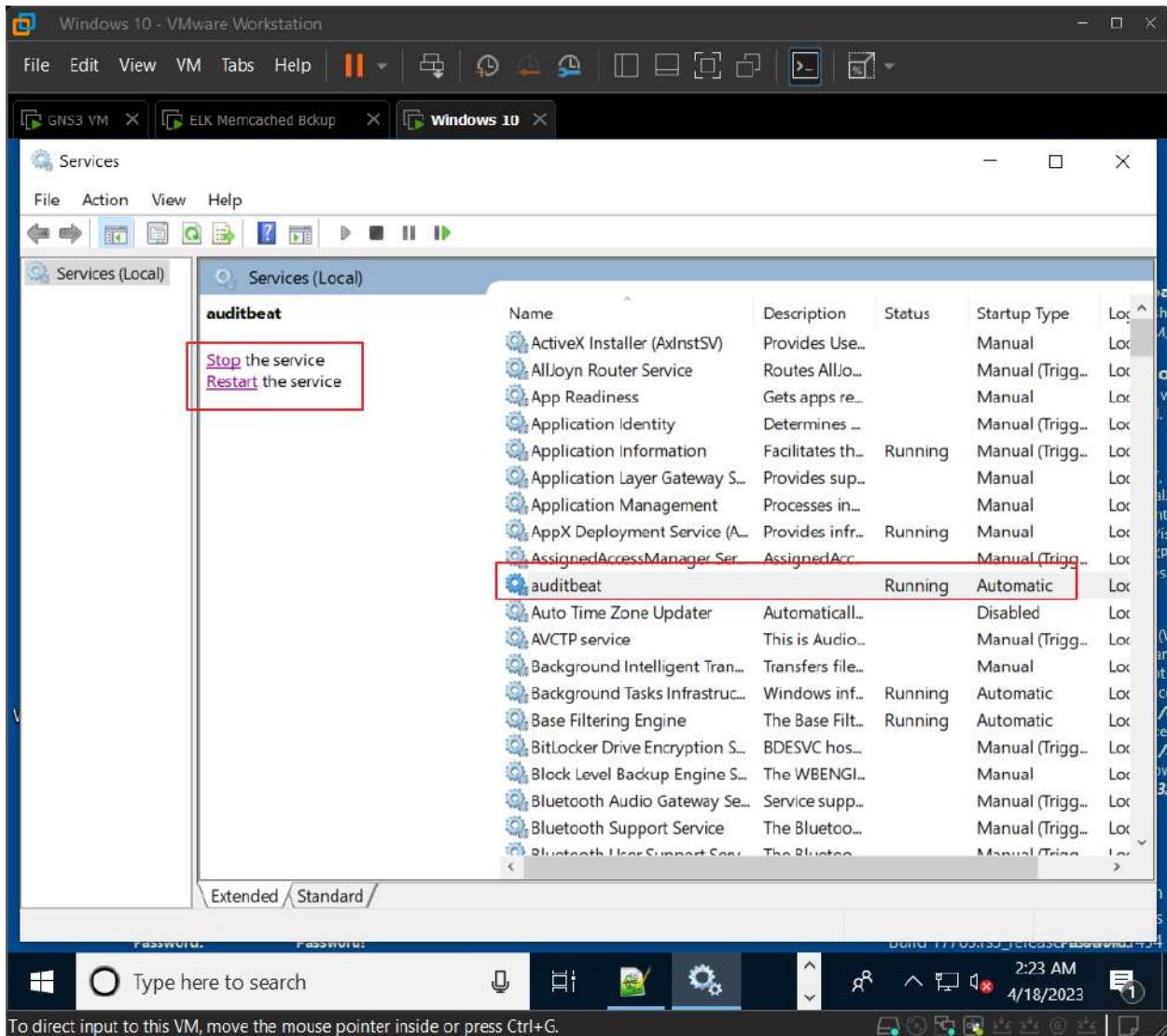
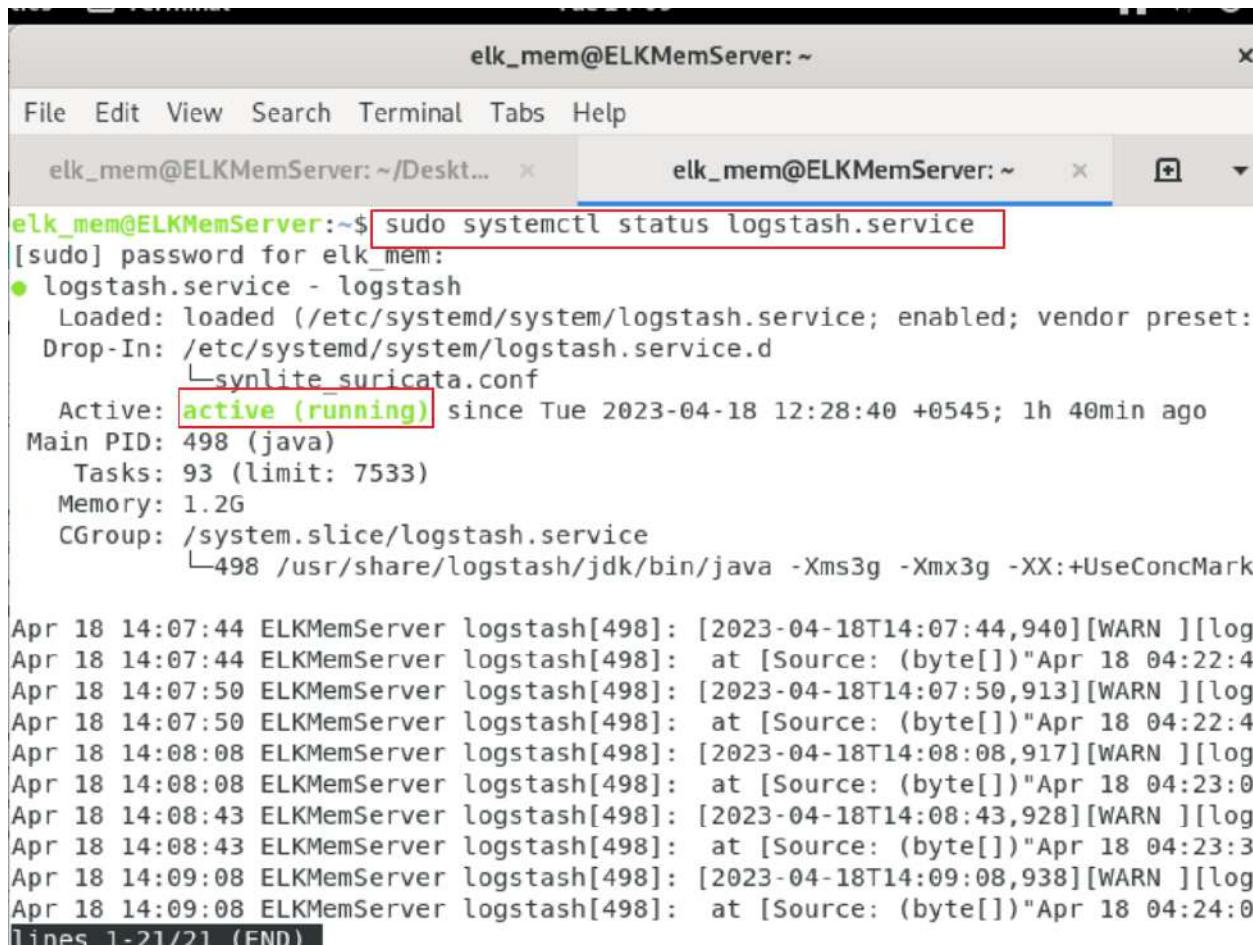


Figure 63: Auditbeat was started.

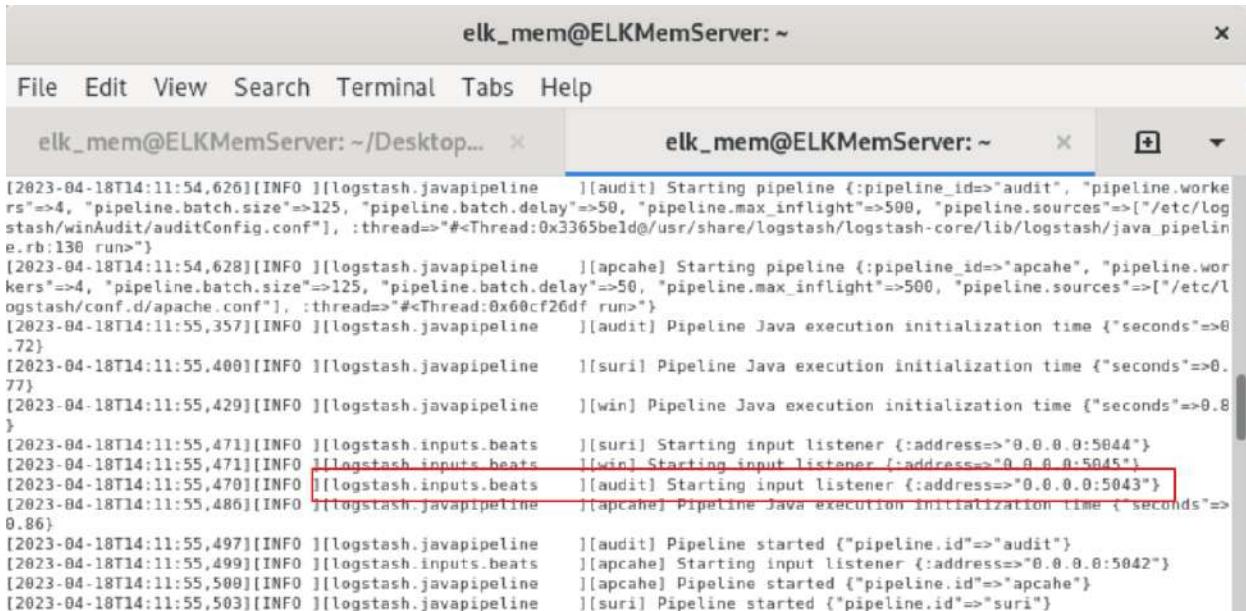


The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window has two tabs: "elk_mem@ELKMemServer: ~/Deskt..." and "elk_mem@ELKMemServer: ~". The second tab is active. The user is running the command "sudo systemctl status logstash.service". The output shows the service is loaded, enabled, and active (running) since April 18, 2023. It lists the main PID (498), tasks (93), memory usage (1.2G), and the CGroup path. Below the service status, there is a log output from logstash showing several WARN messages from April 18, 2023, at various times between 14:07:44 and 14:09:08. The log entries are as follows:

```
elk_mem@ELKMemServer:~$ sudo systemctl status logstash.service
[sudo] password for elk_mem:
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: 
   Drop-In: /etc/systemd/system/logstash.service.d
     └─synlrite_suricata.conf
   Active: active (running) since Tue 2023-04-18 12:28:40 +0545; 1h 40min ago
     Main PID: 498 (java)
       Tasks: 93 (limit: 7533)
      Memory: 1.2G
        CGroup: /system.slice/logstash.service
                  └─498 /usr/share/logstash/jdk/bin/java -Xms3g -Xmx3g -XX:+UseConcMark

Apr 18 14:07:44 ELKMemServer logstash[498]: [2023-04-18T14:07:44,940][WARN ][log]
Apr 18 14:07:44 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:22:4
Apr 18 14:07:50 ELKMemServer logstash[498]: [2023-04-18T14:07:50,913][WARN ][log]
Apr 18 14:07:50 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:22:4
Apr 18 14:08:08 ELKMemServer logstash[498]: [2023-04-18T14:08:08,917][WARN ][log]
Apr 18 14:08:08 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:23:0
Apr 18 14:08:43 ELKMemServer logstash[498]: [2023-04-18T14:08:43,928][WARN ][log]
Apr 18 14:08:43 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:23:3
Apr 18 14:09:08 ELKMemServer logstash[498]: [2023-04-18T14:09:08,938][WARN ][log]
Apr 18 14:09:08 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:24:0
Lines 1-21/21 (END)
```

Figure 64: Verifying status of Logstash service.



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". It has two tabs open: "elk_mem@ELKMemServer: ~/Desktop..." and "elk_mem@ELKMemServer: ~". The second tab is active and displays Logstash logs. The logs show the startup of various pipelines and inputs. Several log entries are highlighted with red boxes:

- [2023-04-18T14:11:55,471][INFO][logstash.inputs.beats] [[suri]] Starting input listener {:address=>"0.0.0.0:5043"}
- [2023-04-18T14:11:55,471][INFO][logstash.inputs.beats] [[win]] Starting input listener {:address=>"0.0.0.0:5045"}
- [2023-04-18T14:11:55,470][INFO][logstash.inputs.beats] [[audit]] Starting input listener {:address=>"0.0.0.0:5043"}

The logs also mention the startup of audit, apache, and suri pipelines.

```
[2023-04-18T14:11:54,626][INFO ][logstash.javapipeline] [[audit]] Starting pipeline {:pipeline_id=>"audit", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max_inflight"=>500, "pipeline.sources"=>["/etc/logstash/winAudit/auditConfig.conf"], :thread=>"#<Thread:0x3365beld@/usr/share/logstash/logstash-core/lib/logstash/java_pipeline.rb:138 run>"}
[2023-04-18T14:11:54,628][INFO ][logstash.javapipeline] [[apache]] Starting pipeline {:pipeline_id=>"apache", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max_inflight"=>500, "pipeline.sources"=>["/etc/logstash/conf.d/apache.conf"], :thread=>"#<Thread:0x60cf26df run>"}
[2023-04-18T14:11:55,357][INFO ][logstash.javapipeline] [[audit]] Pipeline Java execution initialization time {"seconds"=>0.72}
[2023-04-18T14:11:55,400][INFO ][logstash.javapipeline] [[suril]] Pipeline Java execution initialization time {"seconds"=>0.77}
[2023-04-18T14:11:55,420][INFO ][logstash.javapipeline] [[win]] Pipeline Java execution initialization time {"seconds"=>0.8}
[2023-04-18T14:11:55,471][INFO ][logstash.inputs.beats] [[suri]] Starting input listener {:address=>"0.0.0.0:5044"}
[2023-04-18T14:11:55,471][INFO ][logstash.inputs.beats] [[win]] Starting input listener {:address=>"0.0.0.0:5045"}
[2023-04-18T14:11:55,470][INFO ][logstash.inputs.beats] [[audit]] Starting input listener {:address=>"0.0.0.0:5043"}
[2023-04-18T14:11:55,486][INFO ][logstash.javapipeline] [[apache]] Pipeline Java execution initialization time {"seconds"=>0.86}
[2023-04-18T14:11:55,497][INFO ][logstash.javapipeline] [[audit]] Pipeline started {"pipeline.id"=>"audit"}
[2023-04-18T14:11:55,499][INFO ][logstash.inputs.beats] [[apache]] Starting input listener {:address=>"0.0.0.0:5042"}
[2023-04-18T14:11:55,500][INFO ][logstash.javapipeline] [[apache]] Pipeline started {"pipeline.id"=>"apache"}
[2023-04-18T14:11:55,503][INFO ][logstash.javapipeline] [[suril]] Pipeline started {"pipeline.id"=>"suril"}
```

Figure 65: Verifying Logstash was listening in port 5043 allocated for auditbeat.

The screenshot shows the Elasticsearch Index Management interface in Firefox ESR. The left sidebar has a navigation menu with sections like Management, Ingest, Data, Index Management, Alerts and Insights, and Security. The 'Index Management' section is currently selected. The main area displays a table of indices with columns: Name, Health, Status, Primaries, Replicas, Docs count, Storage size, and Data stream. The 'auditbeat-7.17.9-2023.04.18' index is highlighted with a red border.

Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
elastalert_status	yellow	open	1	1	54	158.6kb	
elastalert_status_silence	yellow	open	1	1	54	43kb	
filebeat-7.17.9-2023.04.18	yellow	open	1	1	650	2.7mb	
elastalert_status_error	yellow	open	1	1	0	226b	
elastalert_status_status	yellow	open	1	1	167	51kb	
elastalert_status_past	yellow	open	1	1	0	226b	
auditbeat-7.17.9-2023.04.18	yellow	open	1	1	133	334.3kb	
apache_log-2023.04.18	yellow	open	1	1	8	57.7kb	
winlogbeat-7.17.9-2023.04.18	yellow	open	1	1	1654	2.4mb	

Figure 66: Auditbeat log was seen in index management.

4.2.5 Test Case 5

Test Case 5	
Objective	To test USB detection Alerts on EKL server's discover dashboard.
Action	USB device was plugged in to windows 10 VM.
Expected Test Result	After USB device was plugged in to windows VM, that event log will register into ELK stack.
Actual Test Result	After USB device was plugged in to windows VM, the event log was registered into ELK stack.
Conclusion	Test was successful.

Table 7: Test Case 5

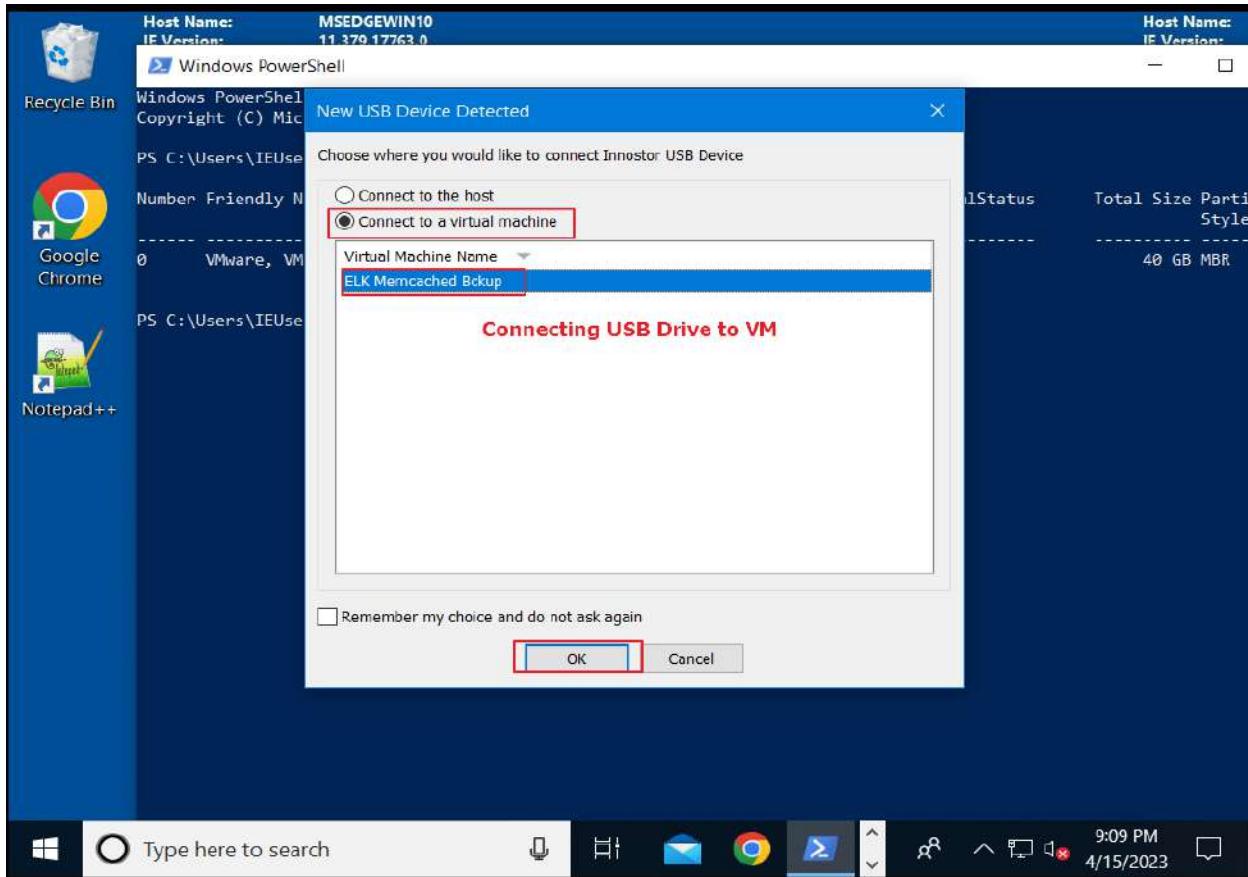


Figure 67: USB drive connecting to windows 10 VM.

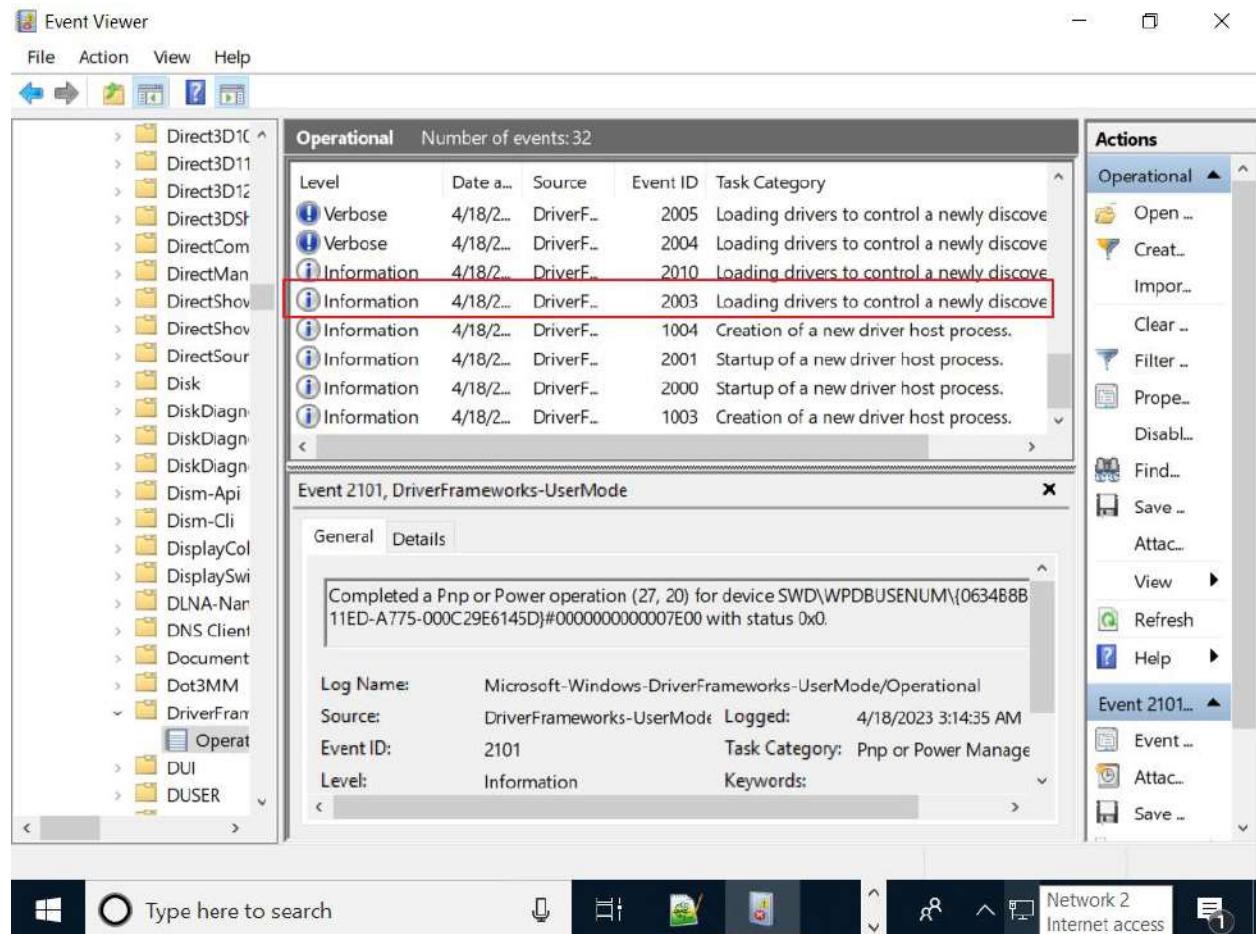


Figure 68: Windows event log generated when USB device was plugged in.

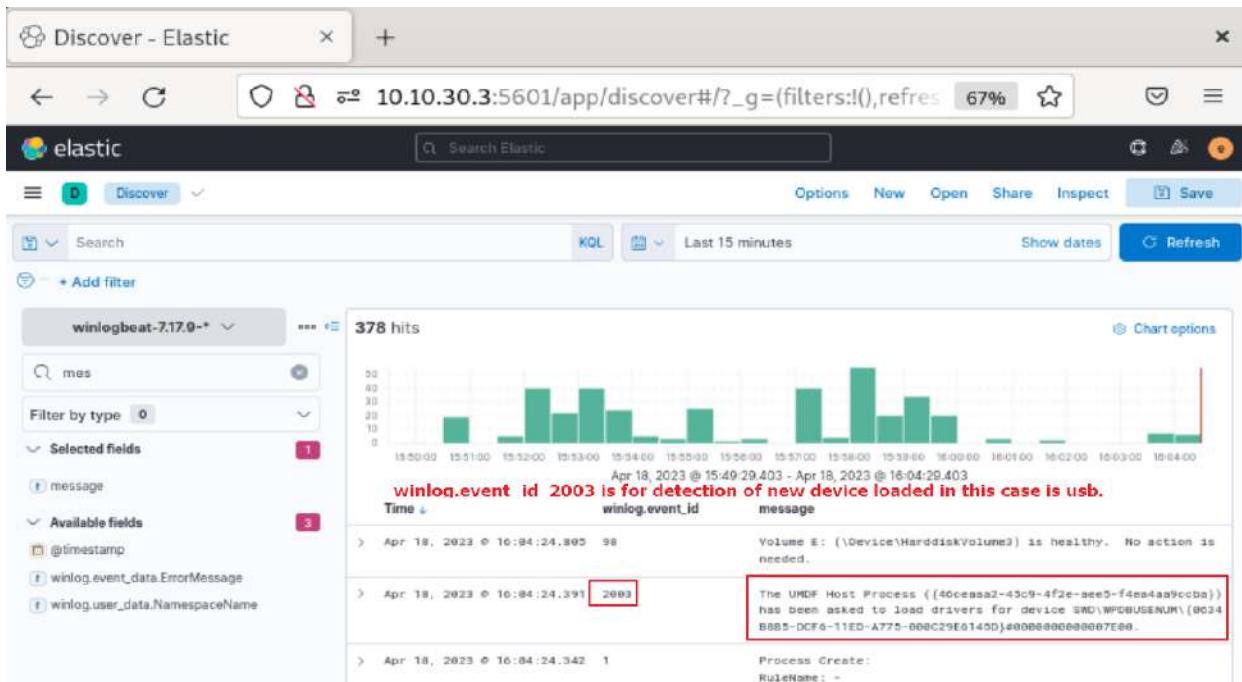


Figure 69: Win log event id 2003 present in Kibana discover dashboard which indicated USB device was plugged in.

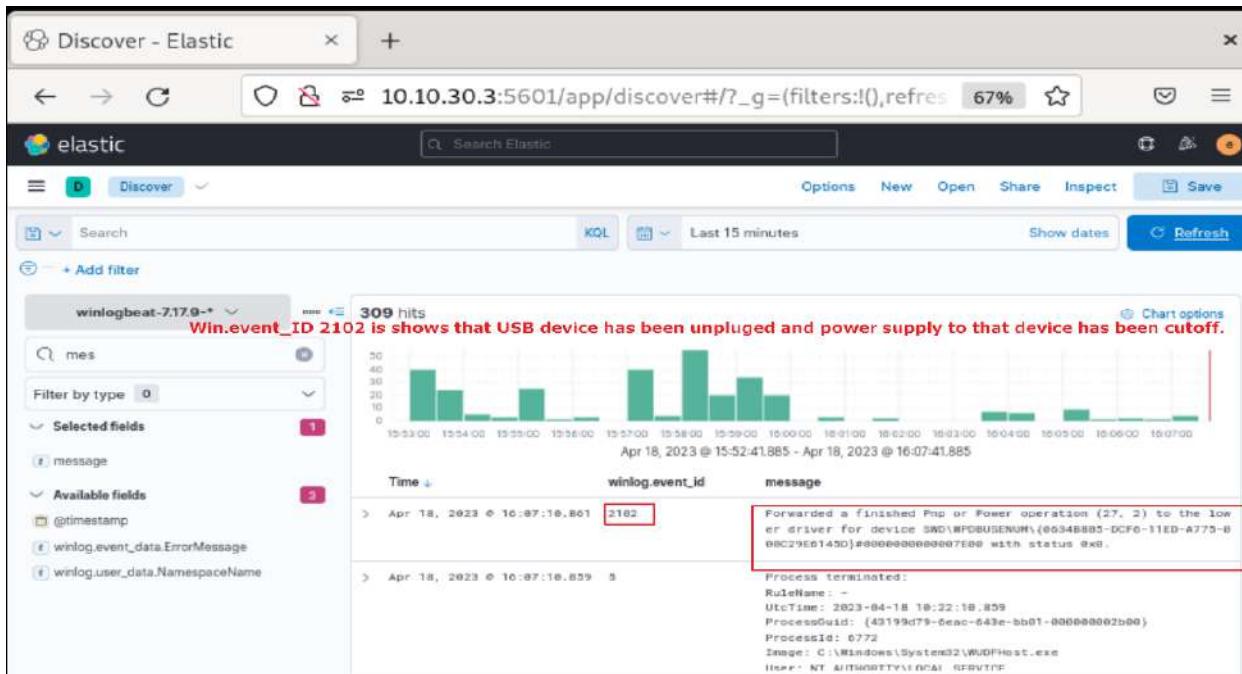


Figure 70: Win log event id 2102 present in Kibana discover dashboard which indicated USB device was unplugged.

4.2.5 Test Case 6

Test Case 6	
Objective	To test file integrity status detection alerts on EKL server's discover dashboard.
Action	A File was created, and it was deleted in windows 10 VM.
Expected Test Result	In ELK's discover dashboard, alert will be displayed showing file event like Creation and deletion.
Actual Test Result	In ELK's discover dashboard, alert was displayed showing file event like Creation and deletion.
Conclusion	Test was successful.

Table 8: Test Case 6



Figure 71: File was created.

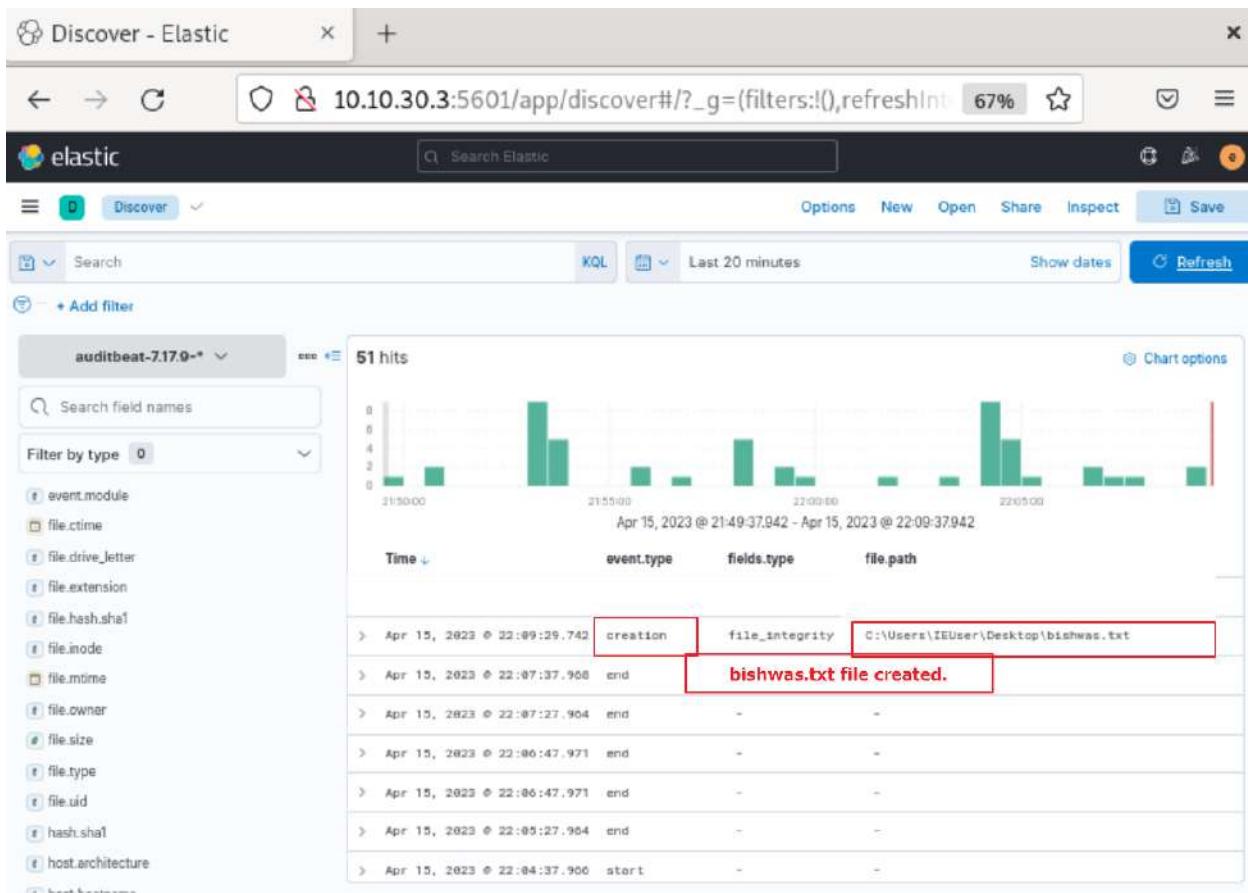


Figure 72: bishwas.txt file creation alert in discover dashboard.

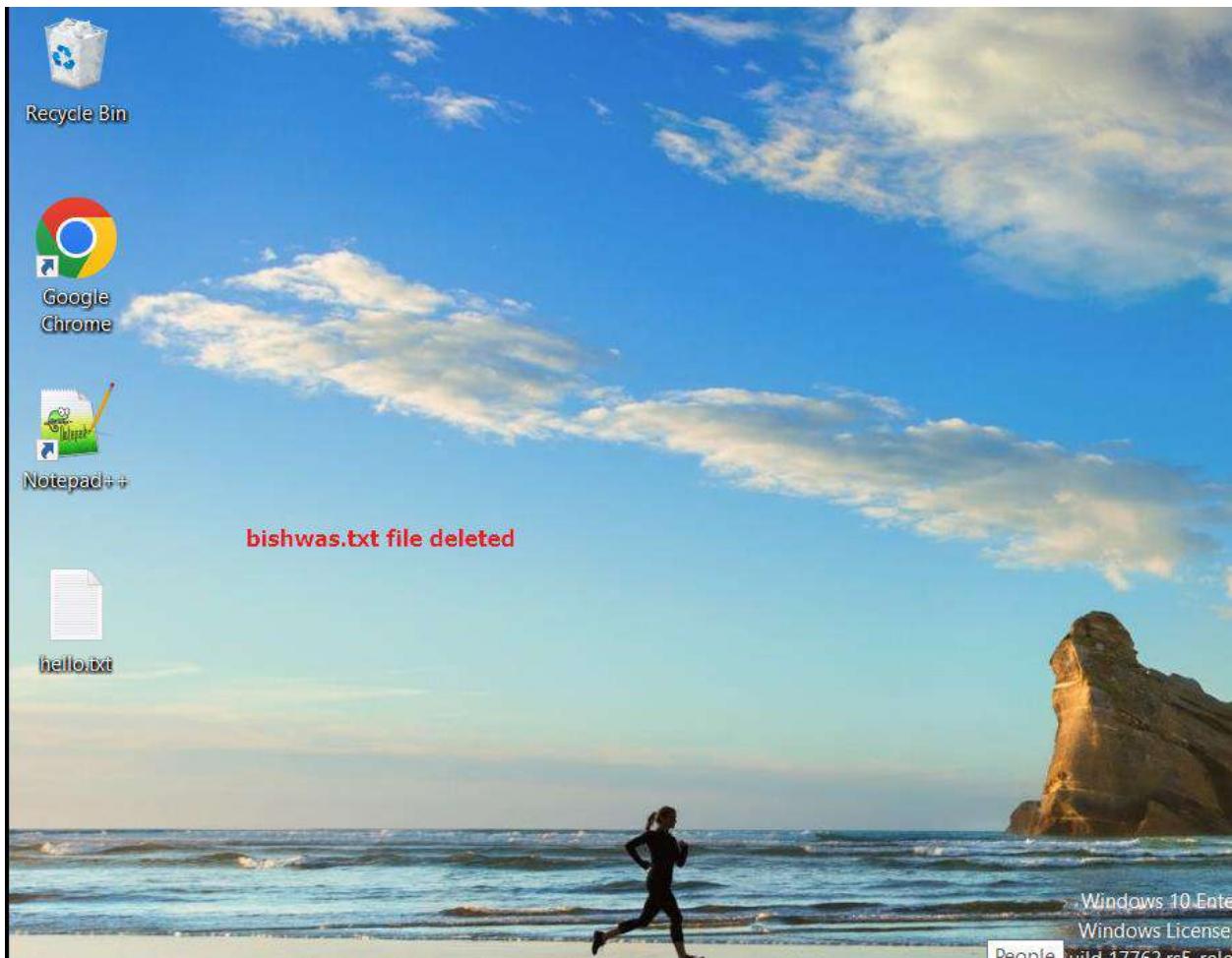


Figure 73: File was deleted.

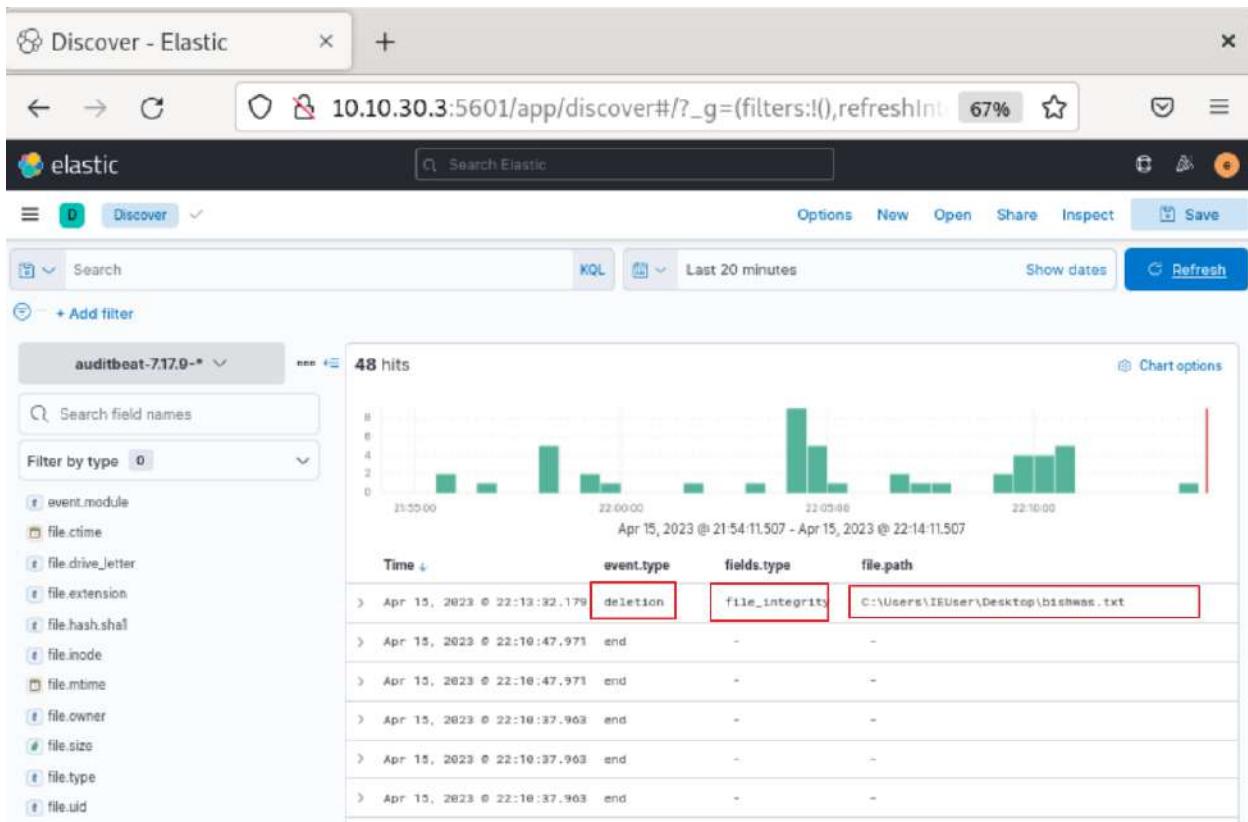


Figure 74: bishwas.txt file deletion alert in discover dashboard.

4.2.7 Test Case 7

Test Case 7	
Objective	To test whether script was fetching IOCs from MISP and loading to Memcached.
Action	<p>Customs events containing IOCs are created. After that python script was executed where it contains parameters for fetching and storing IOCs like hash sha256, IPs and domain.</p> <p>Parameters are given below:</p> <p>Memcached IP => 10.10.30.3</p> <p>Memcached Port => 11211</p> <p>MISP Url => https://10.10.30.10./attributes/restSearch</p> <p>MISP token =></p> <p>rz2jVQYK0uaPC0uZ6tf3W4KzqIU2UkW4vClvlWAN</p> <p>Here, telnet connection was made to Memcached sever to verify IOCs present in the server.</p>
Expected Test Result	After running python script, it will fetch all custom created IOCs from MISP server and load to Memcached server
Actual Test Result	After running python script, it fetched all custom created IOCs from MISP server and loaded to Memcached server
Conclusion	Test was Successful.

Table 9: Test Case 7

The screenshot shows the MISP web interface for Event #1610. The top navigation bar includes links for Home, Event Actions, Dashboard, Galaxies, Input Filters, Global Actions, Sync Actions, Administration, Logs, and API. The main content area displays a table of event details. The table has columns for Date, Category, Type, Value, Tags, Galaxies, Comment, and Correlate. Four rows of data are shown, each representing a network activity event with an IP source value. The 'Value' column for these four rows is highlighted with a red box. Below the table, a message indicates 'Page 1 of 1, showing 1 records out of 4 total, starting on record 1, ending on 4'. A 'Discussion' section follows, containing a text input field with buttons for Quote, Event, Thread, Link, and Code.

Date	Category	Type	Value	Tags	Galaxies	Comment	Correlate
2023-04-15	Network activity	ip-src	10.10.30.4	3+ 1+	3+ 1+		✓
2023-04-14	Network activity	ip-src	10.10.10.2	3+ 1+	3+ 1+		✓
2023-04-14	Network activity	ip-src	10.10.10.1	3+ 1+	3+ 1+		✓
2023-04-14	Network activity	ip-src	10.10.30.5	3+ 1+	3+ 1+		✓

Figure 75: Custom created event in MISP with source IPs.

<input type="checkbox"/> 2023-04-15 Payload delivery sha256 8e0d841929ecd6e1223205e0163e1f5b4938973f7bd743a6094a84c3d4db9f56 
<input type="checkbox"/> 2023-04-15 Payload delivery sha256 e272d0e33299bca60479683eae0d9b5d880293c01b337d5759fee812ea631a39 
<input type="checkbox"/> 2023-04-15 Payload delivery sha256 01b407af0200b66a34d9b1fa6d9eaab758efa36a36bb99b554384f59f8690b1a 
<input type="checkbox"/> 2023-04-15 Payload delivery sha256 5dfc43333a2360ad916f67bf783d8260a32d811a738b3d2e58427b1b384ff9a3 
<input type="checkbox"/> 2023-04-15 Payload delivery sha256 dc5797098068465da646550e109ac7652eaf66a727dfe1b4cf6994b1a7f6cb1 
<input type="checkbox"/> 2023-04-15 Payload delivery sha256 3f82d416fbbe431b0ae798078f9c354711577f48b38901788c784a5ec0dd13b3 
<input type="checkbox"/> 2023-04-15 Payload delivery sha256 7fd065bac18c5278777ae44908101cdfed72d26fa741367f0ad4d02020787ab6 
<input type="checkbox"/> 2023-04-15 Payload delivery sha256 b8e176fe76a1454a00c4af0f8bf8870650d9c33d3e333239a59445c5b35c9a37 
<input type="checkbox"/> 2023-04-15 Payload delivery sha256 78097c7cd0e57902536c60b7fa17528c313db20869e5f944223a0ba4c801d39b 

Figure 76: Custom created event in MISP with SHA256 hashes.

The screenshot shows the MISP web interface for Event #1614. The URL in the browser is <https://10.10.30.10/events/view/1614>. The main content area displays a table of event details. One row in the table is highlighted with a red border, showing a timestamp (2023-04-18), category (Network activity), type (ip-dst), and value (10.10.30.10). Below the table, a message indicates "Page 1 of 1, showing 1 records out of 1 total, starting on record 1, ending on 1". A "Discussion" section follows, featuring a text input field and a "Send comment" button.

Figure 77: Custom created event in MISP with destination IPs.

The screenshot shows the MISP web interface for Event #1612. The URL is https://10.10.30.10/events/view/1612. The main content area displays a table of event attributes. A red box highlights the 'Value' column for the 'Type' 'Domain'. The table has columns: Date, Category, Type, Value, Tags, Galaxies, Comment, and Correlate. The 'Value' column contains domain names like www.reddit.com, www.wikipedia.org, etc. The 'Tags' column shows various threat intelligence tags. The 'Galaxies' column shows network connections. The 'Comment' and 'Correlate' columns contain checkmarks.

Date	Category	Type	Value	Tags	Galaxies	Comment	Correlate
2023-04-15	Network activity	domain	www.reddit.com	[tags]	[galaxies]		✓
2023-04-15	Network activity	domain	www.wikipedia.org	[tags]	[galaxies]		✓
2023-04-15	Network activity	domain	trencansky.sk	[tags]	[galaxies]		✓
2023-04-15	Network activity	domain	www.google.com	[tags]	[galaxies]		✓
2023-04-15	Network activity	domain	www.gmail.com	[tags]	[galaxies]		✓
2023-04-15	Network activity	domain	play.google.com	[tags]	[galaxies]		✓
2023-04-15	Network activity	domain	static.xx.fbcdn.net	[tags]	[galaxies]		✓
2023-04-15	Network activity	domain	www.facebook.com	[tags]	[galaxies]		✓

Page 1 of 1, showing 1 records out of 8 total, starting on record 1, ending on 8

Figure 78: Custom created event in MISP with destination Domain address.

The screenshot shows a terminal window titled 'elk_mem@ELKMemServer: ~/Desktop/Misp2MemScript Folder'. The command ./misp2memcached.py is run. The output shows the URL https://10.10.30.10/attributes/restSearch, token rz2jVQYK0uaPC0uZ6tf3W4KzqIU2UkW4vClvlWAN, boolean True, and a Memcached object at 0x7f9b50285860. The script then exits with a status of 0.

```
elk_mem@ELKMemServer:~/Desktop/Misp2MemScript Folder$ ./misp2memcached.py
url: https://10.10.30.10/attributes/restSearch
token: rz2jVQYK0uaPC0uZ6tf3W4KzqIU2UkW4vClvlWAN
boolean: True
memcached<outputs.memcached.Memcached object at 0x7f9b50285860>
<Response [200]>
finish
[{"response": [{"Attribute": {"id": "280047", "event_id": "1612", "object_id": "0", "ob": "0", "value": "0"}]}]
```

Figure 79: Executing python script.

```

elk_mem@ELKMemServer: ~/Desktop/Misp2MemScript Folder
File Edit View Search Terminal Help
SH1', 'orgc_id': '1', 'uuid': 'd917dc4-20e2-4f2c-8cc7-c6a40cf54556'}]}]
lookupvalue: 78097c7cd0e57902536c60b7fa17528c313db20869e5f944223a0ba4c801d39b
tag: 1613#sha256
key: misp-sha256:78097c7cd0e57902536c60b7fa17528c313db20869e5f944223a0ba4c801d39b
memcache_value: None
key1: misp-sha256:78097c7cd0e57902536c60b7fa17528c313db20869e5f944223a0ba4c801d39b tag: 1613#sha256
h namespace: misp-sha256
h lookup_value: 78097c7cd0e57902536c60b7fa17528c313db20869e5f944223a0ba4c801d39b
h event_id: 1613
h tag: 1613#sha256
lookupvalue: 3f82d416fbeec431b0ae798078f9c354711577f48b38901788c784a5ec0dd13b3
tag: 1613#sha256
key: misp-sha256:3f82d416fbeec431b0ae798078f9c354711577f48b38901788c784a5ec0dd13b3
memcache_value: None
key1: misp-sha256:3f82d416fbeec431b0ae798078f9c354711577f48b38901788c784a5ec0dd13b3 tag: 1613#sha256
h namespace: misp-sha256
h lookup_value: 3f82d416fbeec431b0ae798078f9c354711577f48b38901788c784a5ec0dd13b3
h event_id: 1613
h tag: 1613#sha256
lookupvalue: 7fd065bac18c5278777ae44908101cdfed72d26fa741367f0ad4d02020787ab6
tag: 1613#sha256
key: misp-sha256:7fd065bac18c5278777ae44908101cdfed72d26fa741367f0ad4d02020787ab6
memcache_value: None
key1: misp-sha256:7fd065bac18c5278777ae44908101cdfed72d26fa741367f0ad4d02020787ab6 tag: 1613#sha256

```

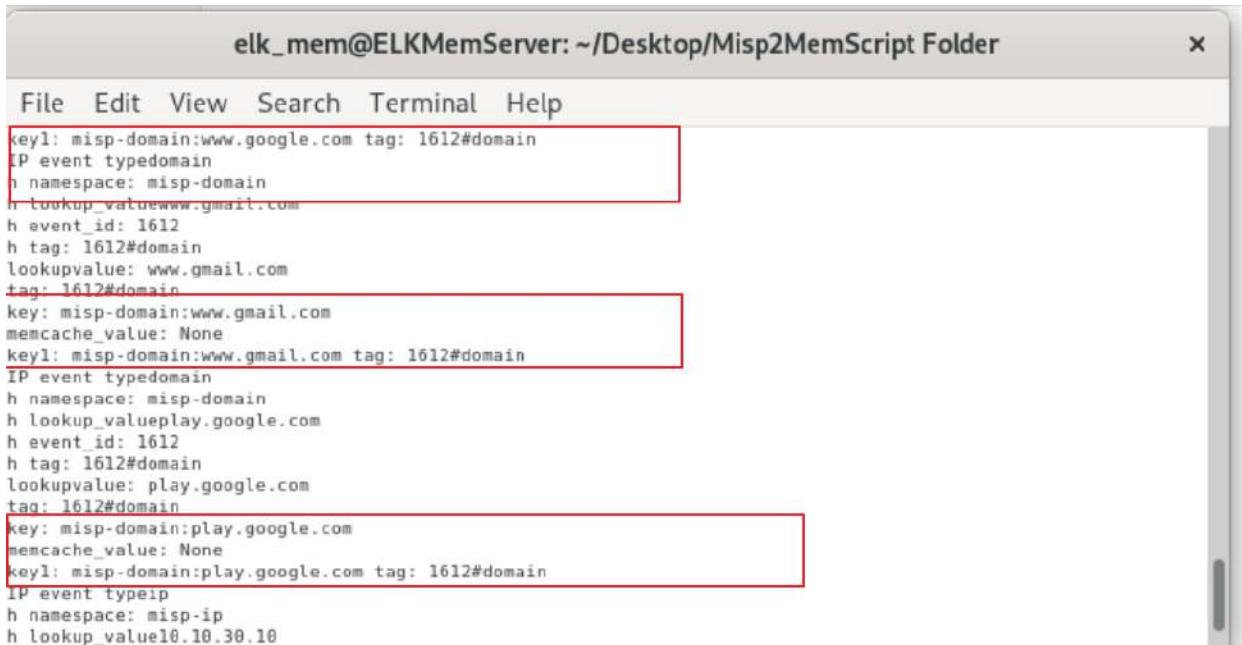
Figure 80: Fetching hash sha256 list from MISP server.

```

elk_mem@ELKMemServer: ~/Desktop/Misp2MemScript Folder
File Edit View Search Terminal Help
-0877643f873c'}]}
IP event typeip
h namespace: misp-ip
h lookup_value: 10.10.30.5
h event_id: 1610
h tag: 1610#ip-src
lookupvalue: 10.10.30.5
tag: 1610#ip-src
key: misp-ip:10.10.30.5
memcache_value: None
key1: misp-ip:10.10.30.5 tag: 1610#ip-src
IP event typeip
h namespace: misp-ip
h lookup_value: 10.10.10.1
h event_id: 1610
h tag: 1610#ip-src
lookupvalue: 10.10.10.1
tag: 1610#ip-src
key: misp-ip:10.10.10.1
memcache_value: None
key1: misp-ip:10.10.10.1 tag: 1610#ip-src
IP event typeip
h namespace: misp-ip
h lookup_value: 10.10.10.2

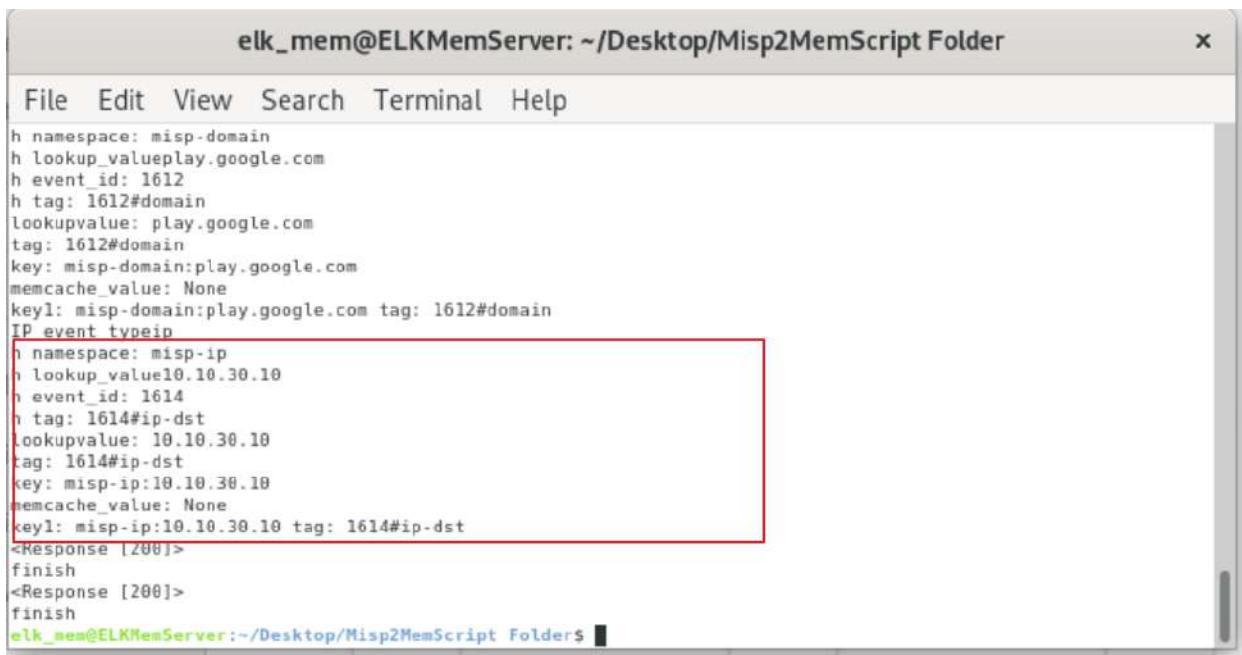
```

Figure 81: Fetching source Ips list from MISP server.



```
elk_mem@ELKMemServer: ~/Desktop/Misp2MemScript Folder
File Edit View Search Terminal Help
key: misp-domain:www.google.com tag: 1612#domain
IP event typedomain
h namespace: misp-domain
h lookup_valuewww.gmail.com
h event_id: 1612
h tag: 1612#domain
lookupvalue: www.gmail.com
tag: 1612#domain
key: misp-domain:www.gmail.com
memcache_value: None
key1: misp-domain:www.gmail.com tag: 1612#domain
IP event typedomain
h namespace: misp-domain
h lookup_valueplay.google.com
h event_id: 1612
h tag: 1612#domain
lookupvalue: play.google.com
tag: 1612#domain
key: misp-domain:play.google.com
memcache_value: None
key1: misp-domain:play.google.com tag: 1612#domain
IP event typeip
h namespace: misp-ip
h lookup_value10.10.30.10
```

Figure 82: Fetching domain list from MISP server.



```
elk_mem@ELKMemServer: ~/Desktop/Misp2MemScript Folder
File Edit View Search Terminal Help
h namespace: misp-domain
h lookup_valueplay.google.com
h event_id: 1612
h tag: 1612#domain
lookupvalue: play.google.com
tag: 1612#domain
key: misp-domain:play.google.com
memcache_value: None
key1: misp-domain:play.google.com tag: 1612#domain
IP event typeip
h namespace: misp-ip
h lookup_value10.10.30.10
h event_id: 1614
h tag: 1614#ip-dst
lookupvalue: 10.10.30.10
tag: 1614#ip-dst
key: misp-ip:10.10.30.10
memcache_value: None
key1: misp-ip:10.10.30.10 tag: 1614#ip-dst
<Response [200]>
finish
<Response [200]>
finish
elk_mem@ELKMemServer:~/Desktop/Misp2MemScript Folder$
```

Figure 83: Fetching destination IP list from MISP server.

```
elk_mem@ELKMemServer: ~/Desktop/Misp2MemScript Folder
File Edit View Search Terminal Tabs Help
elk_mem@ELKMemServer: ~/Deskt... elk_mem@ELKMemServer: ~/Deskt...
elk_mem@ELKMemServer:~/Desktop/Misp2MemScript Folder$ telnet 10.10.30.3 11211
Trying 10.10.30.3...
Connected to 10.10.30.3.
Escape character is '^]'.
get misp-ip:10.10.30.5
VALUE misp-ip:10.10.30.5 0 11
1610#ip-src
END

ERROR
get misp-ip:10.10.30.10
VALUE misp-ip:10.10.30.10 0 11
1614#ip-dst
END
get misp-domain:www.google.com
VALUE misp-domain:www.google.com 0 11
1612#domain
END
```

Figure 84: Performing telnet connection to Memcached server and verifying IOCs list.

4.2.8 Test Case 8

Test Case 8	
Objective	To test if IOCs loaded in Memcached can be lookup and detected in ELK's discover tab.
Action	" www.facebook.com " domain was visited in Suricata VM.
Expected Test Result	An Alert will be display with MISP event type and ID showing match found in IOCs present in Memcached server.
Actual Test Result	An Alert was display with MISP event type and ID showing match found in IOCs present in Memcached server.
Conclusion	Test was Successful.

Table 10: test case 8

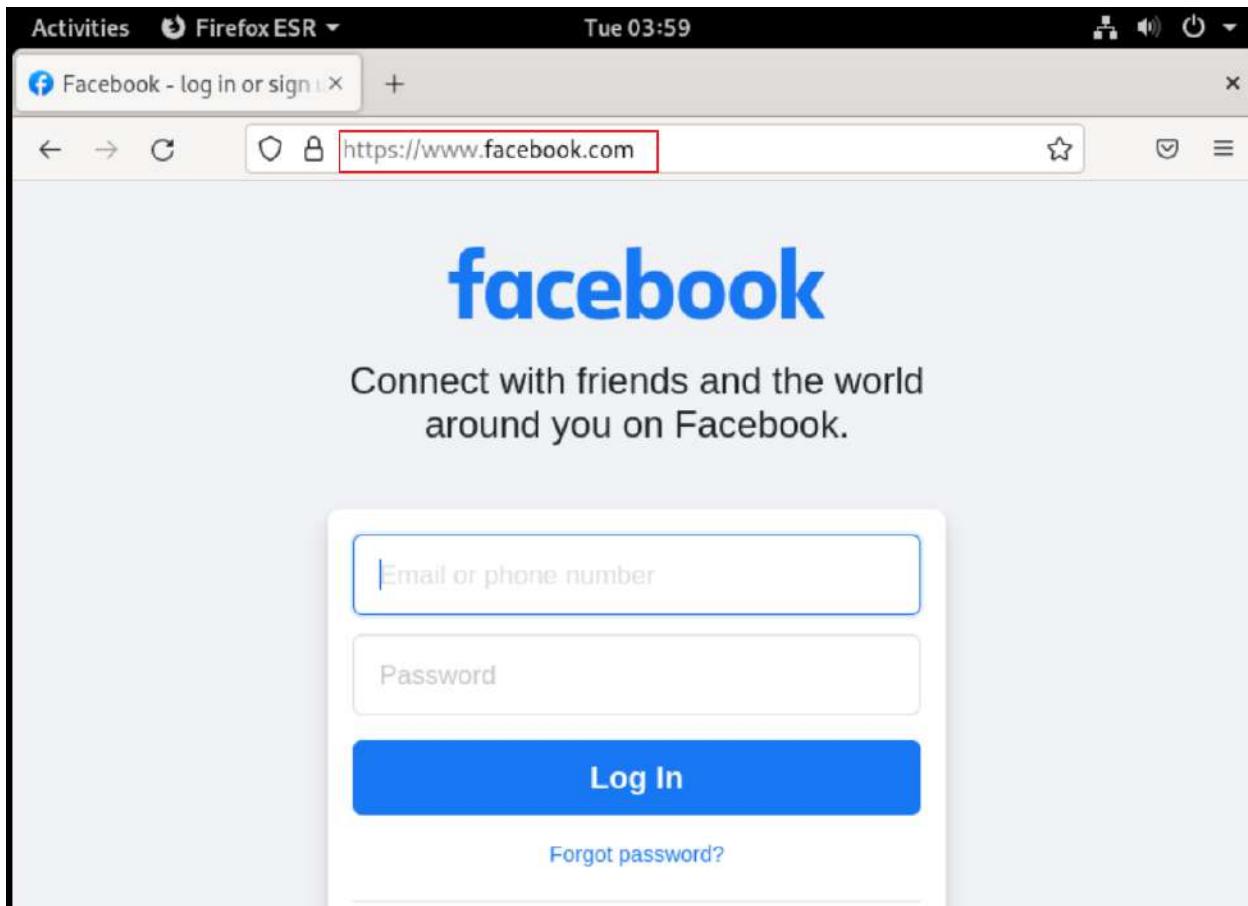


Figure 85: Facebook domain site was visited.

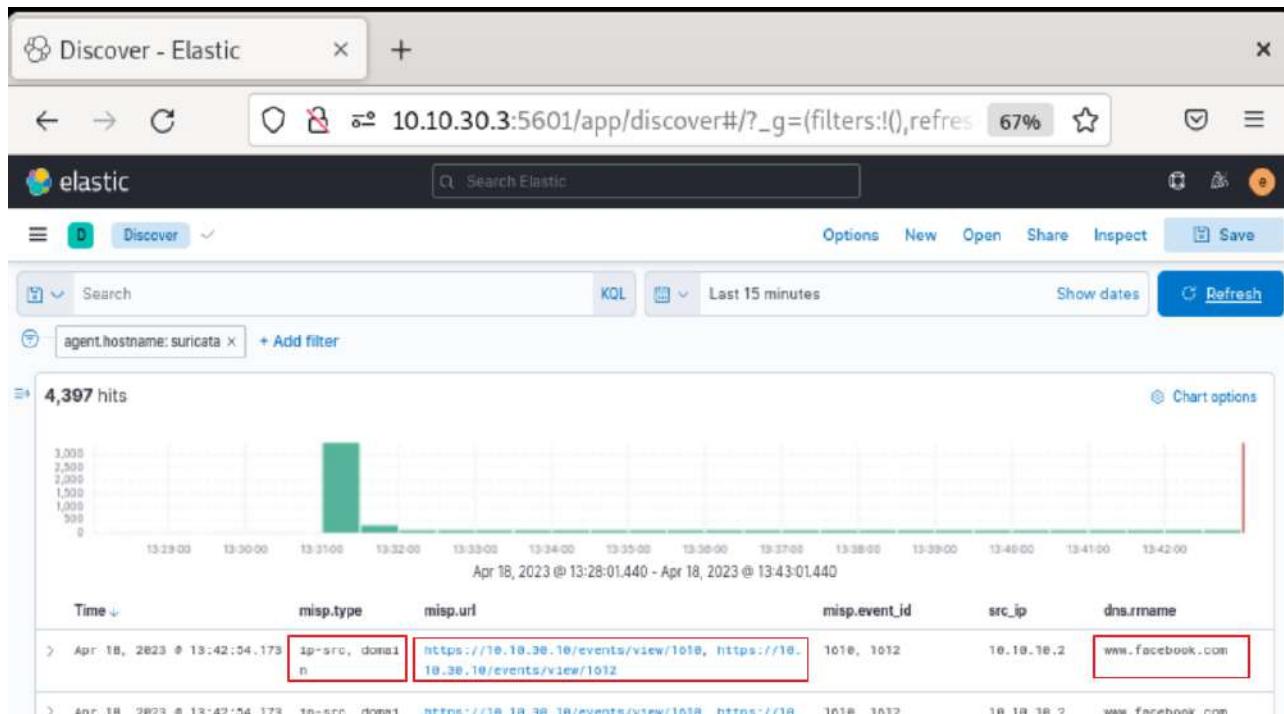


Figure 86: An alert was display after match was found.

4.2.9 Test Case 9

Test Case 9	
Objective	To whether ElastAlert can send message to Telegram.
Action	“example_frequency.yaml” file of ElastAlert was edited to send message about matched misp event type in filebeat index in Elasticsearch.
Expected Test Result	A message will pass by telegram application to user showing: hostname: suricata misp event ID: ['1610','1612',] domain: www.facebook.com MISP EVENT ALERT
Actual Test Result	A message was pass by telegram application to user showing same detail as mention above.
Conclusion	Test was successfully.

The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~/elastalert2/examples/rules". The file being edited is "example_frequency.yaml". The terminal interface includes a menu bar with File, Edit, View, Search, Terminal, and Help, and a status bar indicating "GNU nano 3.2" and the file name "example_frequency.yaml". The code in the editor is as follows:

```
#- misp.event_id
#- src_ip
#alert_text_type: alert_text_only

#telegram_config:
#telegram_bot_token: "5622465916:AAGBpIa8wpjlNseac46Bky0H0QGyKdF_jNk"
#telegram_room_id: "6267921323"
#message_template: "Alert: hostname {{agent.hostname}} misp type {{misp.type}}"
# (required, email specific)
# a list of email addresses to send alerts to
#email:

filter:
- term:
  - misp.type: "domain"
  # (Required)
  # The alert is use when a match is found
alert:
- "telegram"

alert_subject: "hostname: {0} \nmisp event ID: {1} \n domain: {2}"
alert_subject_args:
- agent.hostname
- misp.event_id
- dns.rrname
alert_text_type: alert_text_only

#telegram_config:
telegram_bot_token: "5622465916:AAGBpIa8wpjlNseac46Bky0H0QGyKdF_jNk"
telegram_room_id: "6267921323"
```

Two sections of the configuration are highlighted with red boxes: the "filter" section and the "telegram" section under "alert".

Figure 87: Editing "example_frequency.yaml" for sending alert message to telegram where bot token and room id for telegram was used.

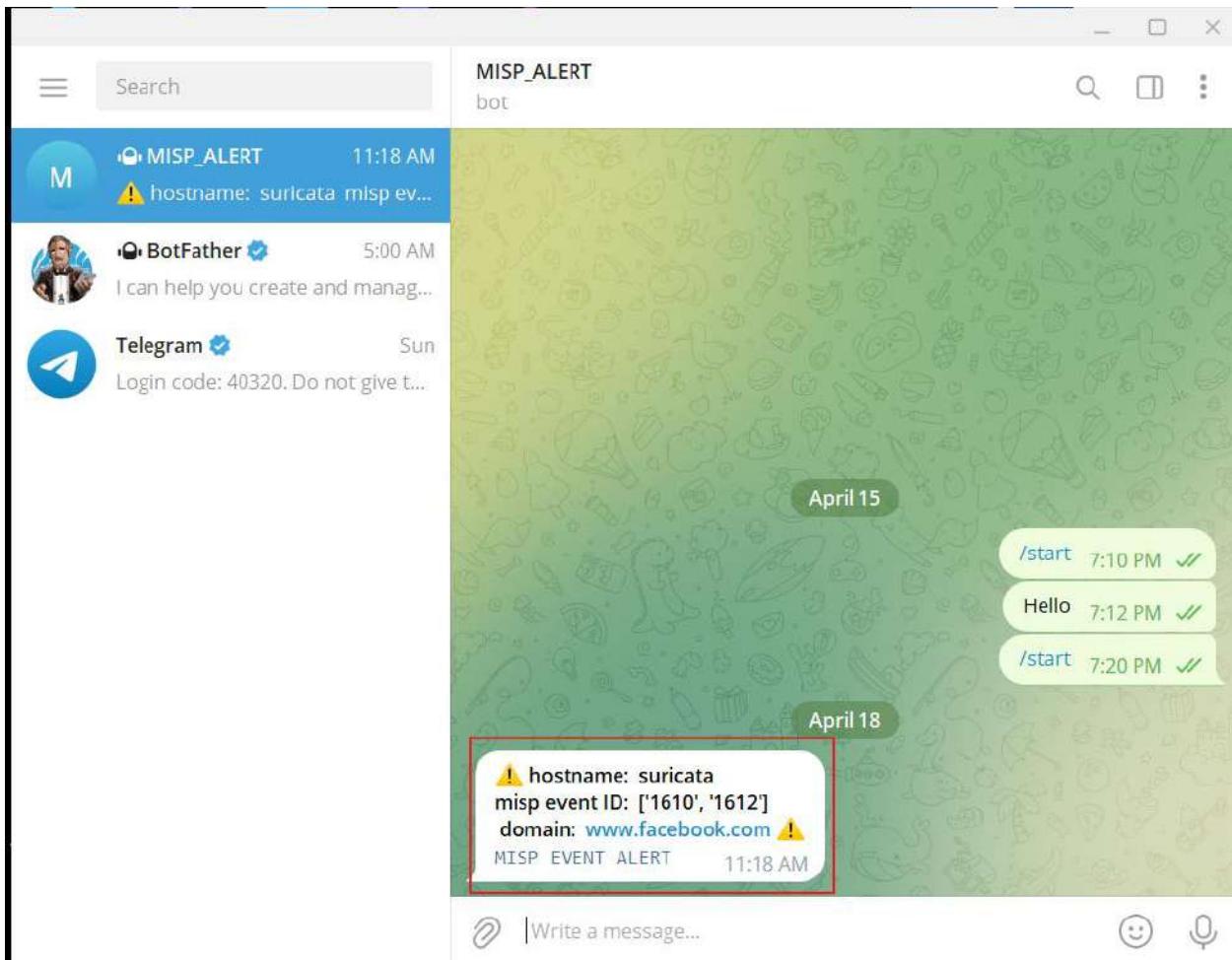


Figure 88: Alert message passed by telegram to user.

4.3 System Testing

4.3.1 Test Case 1

Test Case 1	
Objective	To check if log agents like winlogbeat, auditbeat, and filebeat are collecting logs and sending to Logstash and visualized in Kibana dashboard.
Action	Verifying all the log agents are working and sending logs to Logstash and Logstash is listening to their respective ports.
Expected Test Result	All logs' agents generated logs will be displayed in Kibana dashboard.
Actual Test Result	All logs' agents generated logs was displayed in Kibana dashboard.
Conclusion	

Table 11: Test case 1.

The screenshot shows a terminal window titled "nids@suricata: ~". The user has run two commands: "sudo systemctl start filebeat.service" and "sudo systemctl status filebeat.service". The status output shows the service is active and running. Below the status, there is a log of filebeat's activity from April 18, 2023, at 04:22:18, with entries for WAR, ERR, and INF levels.

```
nids@suricata:~$ sudo systemctl start filebeat.service
nids@suricata:~$ sudo systemctl status filebeat.service
● filebeat.service - Filebeat sends log files to Logstash or directly to Elastic
  Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset:
  Active: active (running) since Tue 2023-04-18 04:20:56 EDT; 1min 44s ago
    Docs: https://www.elastic.co/beats/filebeat
   Main PID: 2157 (filebeat)
     Tasks: 7 (limit: 1283)
    Memory: 120.9M
      CGroup: /system.slice/filebeat.service
              └─2157 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc

Apr 18 04:22:18 suricata filebeat[2157]: 2023-04-18T04:22:18.998-0400      WAR
Apr 18 04:22:18 suricata filebeat[2157]: 2023-04-18T04:22:18.998-0400      ERR
Apr 18 04:22:18 suricata filebeat[2157]: 2023-04-18T04:22:18.998-0400      INF
Apr 18 04:22:18 suricata filebeat[2157]: 2023-04-18T04:22:18.999-0400      INF
Apr 18 04:22:18 suricata filebeat[2157]: 2023-04-18T04:22:18.999-0400      INF
Apr 18 04:22:20 suricata filebeat[2157]: 2023-04-18T04:22:20.006-0400      INF
Apr 18 04:22:20 suricata filebeat[2157]: 2023-04-18T04:22:20.007-0400      INF
Apr 18 04:22:20 suricata filebeat[2157]: 2023-04-18T04:22:20.007-0400      INF
Apr 18 04:22:20 suricata filebeat[2157]: 2023-04-18T04:22:20.013-0400      INF
Apr 18 04:22:30 suricata filebeat[2157]: 2023-04-18T04:22:30.204-0400      INF
Lines 1-20/20 (END)
```

Figure 89: filebeat log agent for Suricata VM in working and running state.

```

apchserver@apchserver:~$ sudo systemctl start filebeat.service
apchserver@apchserver:~$ sudo systemctl status filebeat.service
● filebeat.service - Filebeat sends log files to Logstash or directly to Elastic
  Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset:
  Active: active (running) since Tue 2023-04-18 09:54:29 BST; 8s ago
    Docs: https://www.elastic.co/beats/filebeat
   Main PID: 2240 (filebeat)
     Tasks: 8 (limit: 1194)
    Memory: 100.1M
      CGroup: /system.slice/filebeat.service
              └─2240 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc

Apr 18 09:54:32 apchserver filebeat[2240]: 2023-04-18T09:54:32.916+0100 I
Apr 18 09:54:32 apchserver filebeat[2240]: 2023-04-18T09:54:32.918+0100 I
Apr 18 09:54:32 apchserver filebeat[2240]: 2023-04-18T09:54:32.919+0100 I
Apr 18 09:54:32 apchserver filebeat[2240]: 2023-04-18T09:54:32.919+0100 I
Apr 18 09:54:32 apchserver filebeat[2240]: 2023-04-18T09:54:32.919+0100 I
Apr 18 09:54:35 apchserver filebeat[2240]: 2023-04-18T09:54:35.857+0100 I
lines 1-20/20 (END)

```

Figure 90: Filebeat log agent for Apache web server in running state.

```

t/filebeat.yml --path.home /usr/share/filebeat --path.config /etc/filebeat --path.data /var/lib/filebeat --path.logs /var/log/filebeat
[beat]      instance/beat.go:1067      Host info      {"system.info": {"host": {"architecture": "x86_64", "boot_time": "2023-04-18T08:45:08-04:00", "con...
[beat]      instance/beat.go:1096      Process info    {"system.info": {"process": {"capabilities": {"inheritable": null, "permitted": ["chown", "dac...
instance/beat.go:291      Setup Beat: filebeat; Version: 7.17.9
[publisher]    pipeline/module.go:113      Beat name: suricata
[add_cloud_metadata] add_cloud_metadata/add_cloud_metadata.go:101      add_cloud_metadata: hosting provider type not detected.
beater/filebeat.go:202      Filebeat is unable to load the ingest pipelines for the configured modules because the Elasticsearch output is not config...
[monitoring]    log/log.go:142      Starting metrics logging every 30s
kibana/client.go:180      Kibana url: http://10.10.30.3:5601
kibana/client.go:188      Kibana url: http://10.10.30.3:5601
[monitoring]    log/log.go:184      Non-zero metrics in the last 30s      {"monitoring": {"metrics": {"beat": {"cgroup": {"memory": {"id": "filebeat.s...

```

Figure 91: Filebeat service logs

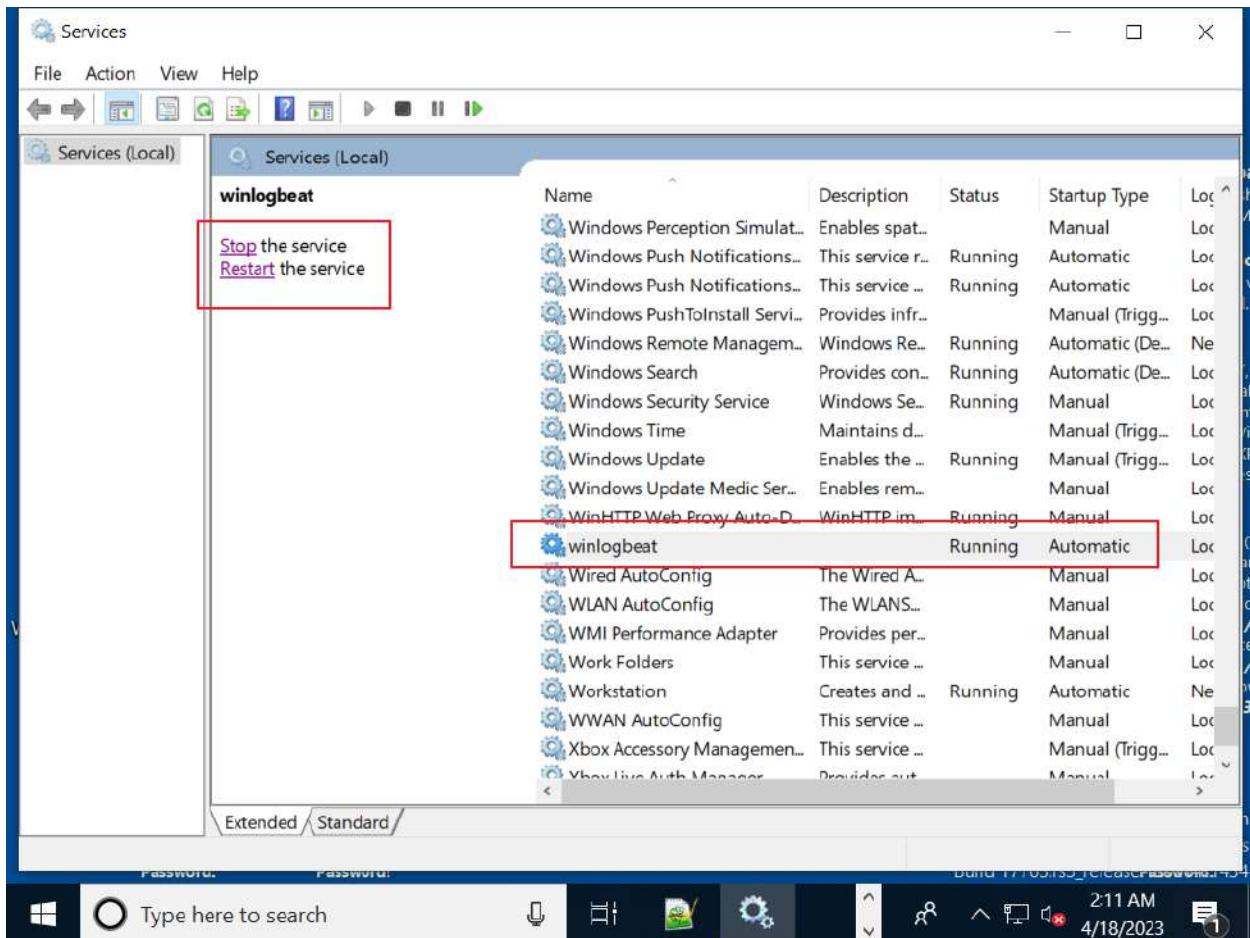


Figure 92: winlogbeat log agent for windows 10 VM in working and running state.

```
C:\ProgramData\winlogbeat\logs\winlogbeat - Notepad++  
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?  
+ ×  
winlogbeat.yml winlogbeat .yml auditbeat.yml auditbeat  
172 [INFO] [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": {"tick  
173 [INFO] [publisher] pipeline/retry.go:219 retrier: send unwait signal to consumer  
174 [ERROR] [logstash] logstash/async.go:280 Failed to publish events caused by: write tcp 10.10.30.4:50000->10.10.30.3:5045: wsasend:  
175 [INFO] [publisher] pipeline/retry.go:223 done  
176 [ERROR] [publisher_pipeline_output] pipeline/output.go:180 failed to publish events: write tcp 10.10.30.4:50000->10.10.30.3:5045: ws  
177 [INFO] [publisher_pipeline_output] pipeline/output.go:143 Connecting to backoff(async(tcp://10.10.30.3:5045))  
178 [INFO] [publisher] pipeline/retry.go:219 retrier: send unwait signal to consumer  
179 [INFO] [publisher] pipeline/retry.go:223 done  
180 [INFO] [publisher_pipeline_output] pipeline/output.go:151 Connection to backoff(async(tcp://10.10.30.3:5045)) established  
181 [INFO] [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": {"tick  
182 [INFO] [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": {"tick  
183 [INFO] [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": {"tick  
184 [INFO] [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": {"tick  
185 [INFO] [publisher] pipeline/retry.go:219 retrier: send unwait signal to consumer  
186 [INFO] [publisher] pipeline/retry.go:223 done  
187 [ERROR] [logstash] logstash/async.go:280 Failed to publish events caused by: write tcp 10.10.30.4:50002->10.10.30.3:5045: wsasend:  
188 [ERROR] [publisher_pipeline_output] pipeline/output.go:180 failed to publish events: write tcp 10.10.30.4:50002->10.10.30.3:5045: ws  
189 [INFO] [publisher_pipeline_output] pipeline/output.go:143 Connecting to backoff(async(tcp://10.10.30.3:5045))  
190 [INFO] [publisher] pipeline/retry.go:219 retrier: send unwait signal to consumer  
191 [INFO] [publisher] pipeline/retry.go:223 done  
192 [INFO] [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": {"tick  
193 [INFO] [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": {"tick  
194 [ERROR] [publisher_pipeline_output] pipeline/output.go:154 Failed to connect to backoff(async(tcp://10.10.30.3:5045)): dial tcp 10.10  
195 [INFO] [publisher] pipeline/retry.go:219 retrier: send unwait signal to consumer  
196 [INFO] [publisher] pipeline/retry.go:223 done  
197 [INFO] [publisher_pipeline_output] pipeline/output.go:145 Attempting to reconnect to backoff(async(tcp://10.10.30.3:5045)) with 1 re  
198 [INFO] [publisher_pipeline_output] pipeline/output.go:151 Connection to backoff(async(tcp://10.10.30.3:5045)) established  
199
```

Figure 93: Winlogbeat service logs

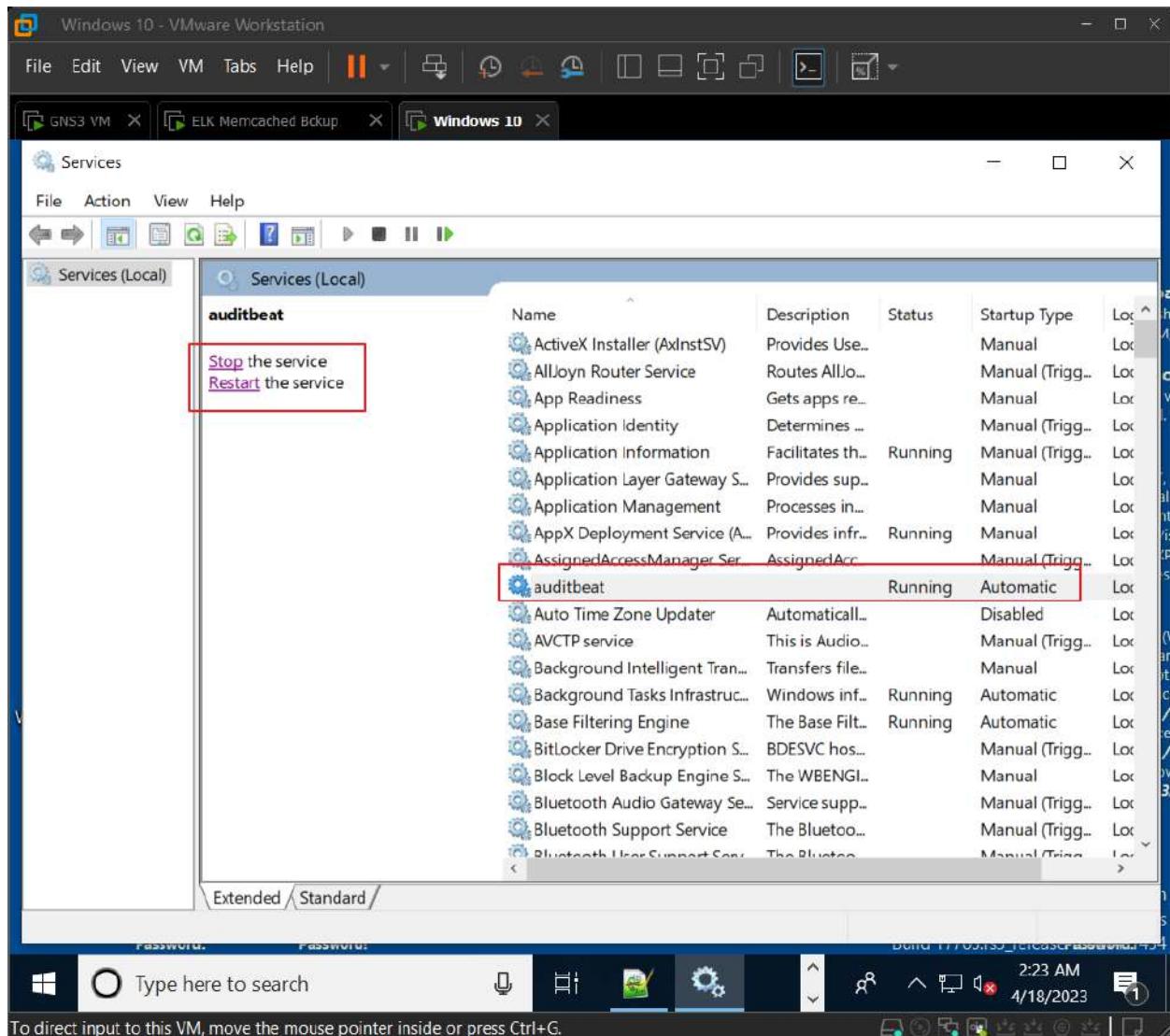
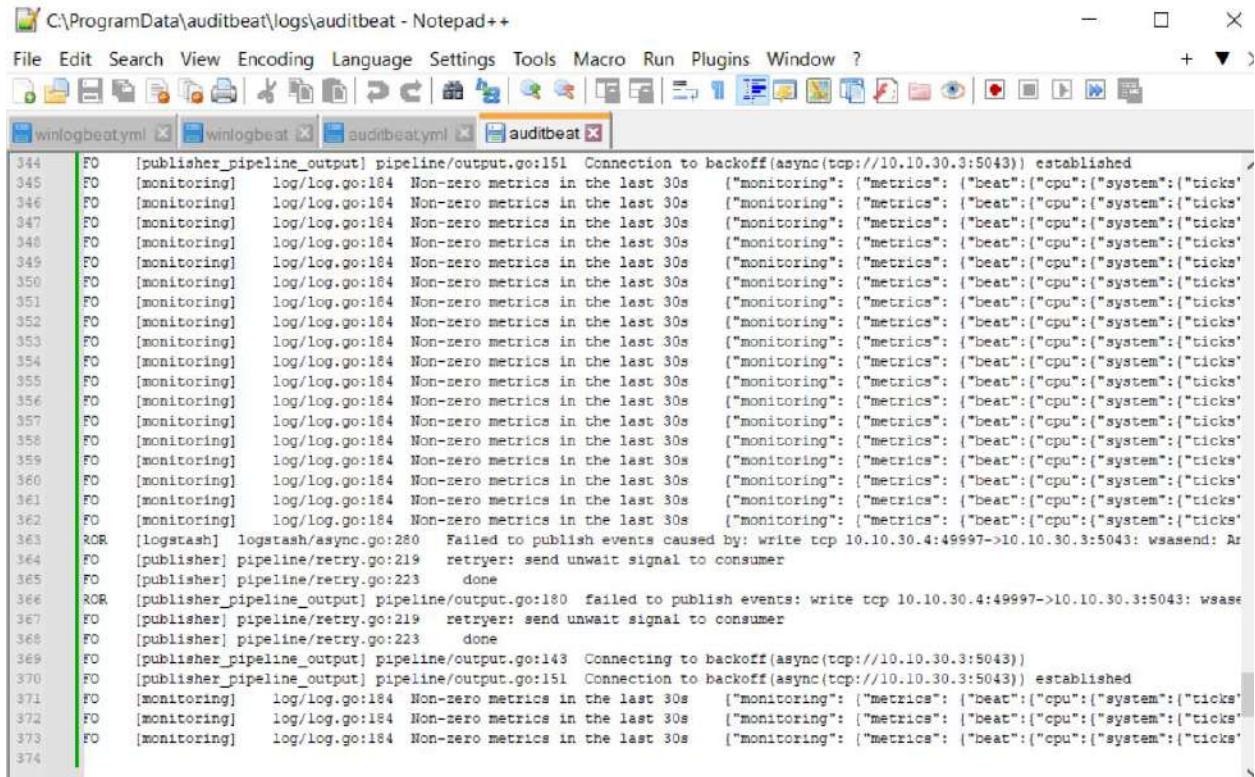


Figure 94: Auditbeat log agent for Suricata VM in working and running state.



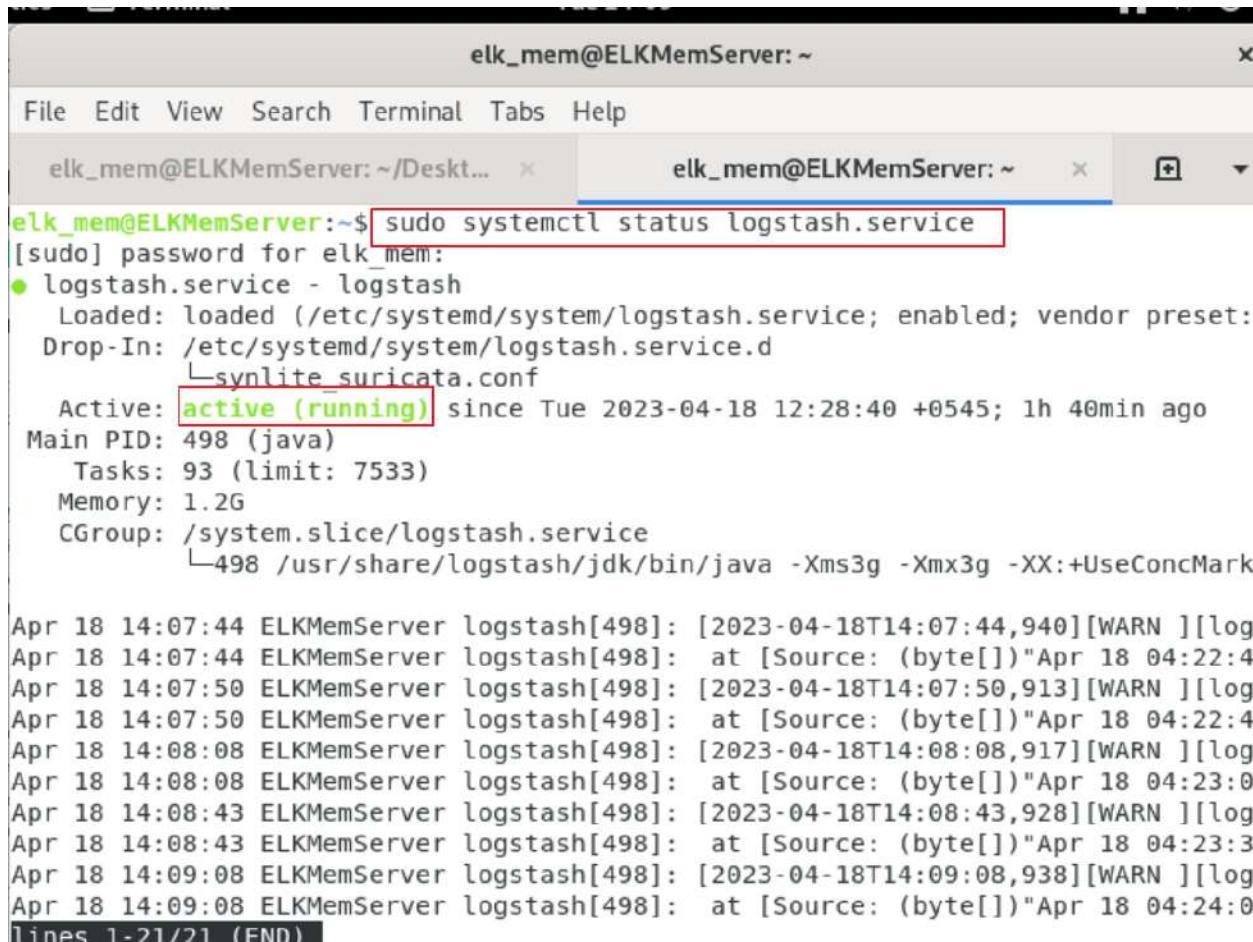
The screenshot shows a Notepad++ window displaying log files. The tabs at the top are labeled 'winlogbeatm1', 'winlogbeat', 'auditbeatm1', and 'auditbeat'. The 'auditbeat' tab is active, showing a large amount of log entries. The log entries are timestamped and include details about monitoring metrics and connection attempts. The log file path is indicated as 'C:\ProgramData\auditbeat\logs\auditbeat - Notepad++'.

```

344 FO [publisher_pipeline_output] pipeline/output.go:151 Connection to backoff(async(tcp://10.10.30.3:5043)) established
345 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
346 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
347 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
348 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
349 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
350 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
351 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
352 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
353 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
354 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
355 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
356 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
357 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
358 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
359 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
360 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
361 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
362 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
363 ROR [logstash] logstash/async.go:280 Failed to publish events caused by: write tcp 10.10.30.4:49997->10.10.30.3:5043: wsasend: Ar
364 FO [publisher] pipeline/retry.go:219 retrier: send unwait signal to consumer
365 FO [publisher] pipeline/retry.go:223 done
366 ROR [publisher_pipeline_output] pipeline/output.go:180 failed to publish events: write tcp 10.10.30.4:49997->10.10.30.3:5043: wsase
367 FO [publisher] pipeline/retry.go:219 retrier: send unwait signal to consumer
368 FO [publisher] pipeline/retry.go:223 done
369 FO [publisher_pipeline_output] pipeline/output.go:143 Connecting to backoff(async(tcp://10.10.30.3:5043))
370 FO [publisher_pipeline_output] pipeline/output.go:151 Connection to backoff(async(tcp://10.10.30.3:5043)) established
371 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
372 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
373 FO [monitoring] log/log.go:184 Non-zero metrics in the last 30s {"monitoring": {"metrics": {"beat": {"cpu": {"system": "ticks" }}}}}
374

```

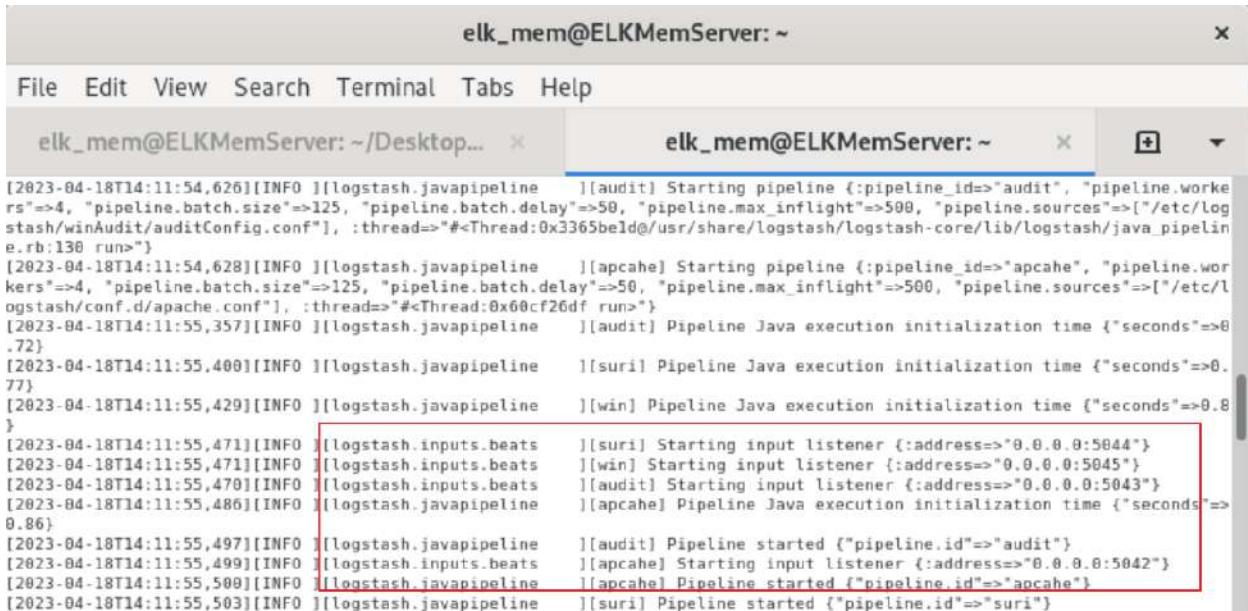
Figure 95: Auditbeat service logs



```
elk_mem@ELKMemServer:~$ sudo systemctl status logstash.service
[sudo] password for elk_mem:
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: Drop-In: /etc/systemd/system/logstash.service.d
             └─synlrite_suricata.conf
   Active: active (running) since Tue 2023-04-18 12:28:40 +0545; 1h 40min ago
     Main PID: 498 (java)
        Tasks: 93 (limit: 7533)
      Memory: 1.2G
        CGroup: /system.slice/logstash.service
                  └─498 /usr/share/logstash/jdk/bin/java -Xms3g -Xmx3g -XX:+UseConcMark

Apr 18 14:07:44 ELKMemServer logstash[498]: [2023-04-18T14:07:44,940][WARN ][log]
Apr 18 14:07:44 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:22:4
Apr 18 14:07:50 ELKMemServer logstash[498]: [2023-04-18T14:07:50,913][WARN ][log]
Apr 18 14:07:50 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:22:4
Apr 18 14:08:08 ELKMemServer logstash[498]: [2023-04-18T14:08:08,917][WARN ][log]
Apr 18 14:08:08 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:23:0
Apr 18 14:08:43 ELKMemServer logstash[498]: [2023-04-18T14:08:43,928][WARN ][log]
Apr 18 14:08:43 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:23:3
Apr 18 14:09:08 ELKMemServer logstash[498]: [2023-04-18T14:09:08,938][WARN ][log]
Apr 18 14:09:08 ELKMemServer logstash[498]: at [Source: (byte[])"Apr 18 04:24:0
Lines 1-21/21 (END)
```

Figure 96: Logstash service is working and running.



```
[2023-04-18T14:11:54,626][INFO ][logstash.javapipeline    ]{audit} Starting pipeline {:pipeline_id=>"audit", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max_inflight"=>500, "pipeline.sources"=>["/etc/logstash/winAudit/auditConfig.conf"], :thread=>"#<Thread:0x3365beld@/usr/share/logstash/logstash-core/lib/logstash/java_pipeline.rb:138 run>"}  
[2023-04-18T14:11:54,628][INFO ][logstash.javapipeline    ]{apcahe} Starting pipeline {:pipeline_id=>"apcahe", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max_inflight"=>500, "pipeline.sources"=>["/etc/logstash/conf.d/apache.conf"], :thread=>"#<Thread:0x60cf26df run>"}  
[2023-04-18T14:11:55,357][INFO ][logstash.javapipeline    ]{audit} Pipeline Java execution initialization time {"seconds"=>0.72}  
[2023-04-18T14:11:55,400][INFO ][logstash.javapipeline    ]{suril} Pipeline Java execution initialization time {"seconds"=>0.77}  
[2023-04-18T14:11:55,420][INFO ][logstash.javapipeline    ]{win} Pipeline Java execution initialization time {"seconds"=>0.8}  
[2023-04-18T14:11:55,471][INFO ][logstash.inputs.beats] {suri} Starting input listener {:address=>"0.0.0.0:5044"}  
[2023-04-18T14:11:55,471][INFO ][logstash.inputs.beats] {win} Starting input listener {:address=>"0.0.0.0:5045"}  
[2023-04-18T14:11:55,470][INFO ][logstash.inputs.beats] {audit} Starting input listener {:address=>"0.0.0.0:5043"}  
[2023-04-18T14:11:55,486][INFO ][logstash.javapipeline    ]{apcahe} Pipeline Java execution initialization time {"seconds"=>0.86}  
[2023-04-18T14:11:55,497][INFO ][logstash.javapipeline    ]{audit} Pipeline started {"pipeline.id"=>"audit"}  
[2023-04-18T14:11:55,499][INFO ][logstash.inputs.beats] {apcahe} Starting input listener {:address=>"0.0.0.0:5042"}  
[2023-04-18T14:11:55,500][INFO ][logstash.javapipeline    ]{apcahe} Pipeline started {"pipeline.id"=>"apcahe"}  
[2023-04-18T14:11:55,503][INFO ][logstash.javapipeline    ]{suri} Pipeline started {"pipeline.id"=>"suri"}
```

Figure 97: Logstash service is listening to all the ports assign to log agents.

The screenshot shows the Elasticsearch Index Management interface. On the left, there's a sidebar with navigation links for Ingest, Data (with Index Management selected), Alerts and Insights, and Security. The main area displays a table of indices with columns for Name, Health, Status, Primaries, Replicas, Docs count, Storage size, and Data stream. Several indices are highlighted with red boxes: filebeat-7.17.9-2023.04.18, auditbeat-7.17.9-2023.04.18, apache.log-2023.04.18, and winlogbeat-7.17.9-2023.04.18. A message at the bottom states "All log agents's generated logs are centralized in ELK server."

Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
elastalert_status	yellow	open	1	1	54	158.6kb	
elastalert_status_silence	yellow	open	1	1	54	43kb	
filebeat-7.17.9-2023.04.18	yellow	open	1	1	650	2.7mb	
elastalert_status_error	yellow	open	1	1	0	226b	
elastalert_status_status	yellow	open	1	1	187	51kb	
elastalert_status_past	yellow	open	1	1	0	226b	
auditbeat-7.17.9-2023.04.18	yellow	open	1	1	643	744.9kb	
apache.log-2023.04.18	yellow	open	1	1	8	57.7kb	
winlogbeat-7.17.9-2023.04.18	yellow	open	1	1	2450	3.8mb	

Figure 98: All logs are centralized in ELK server.

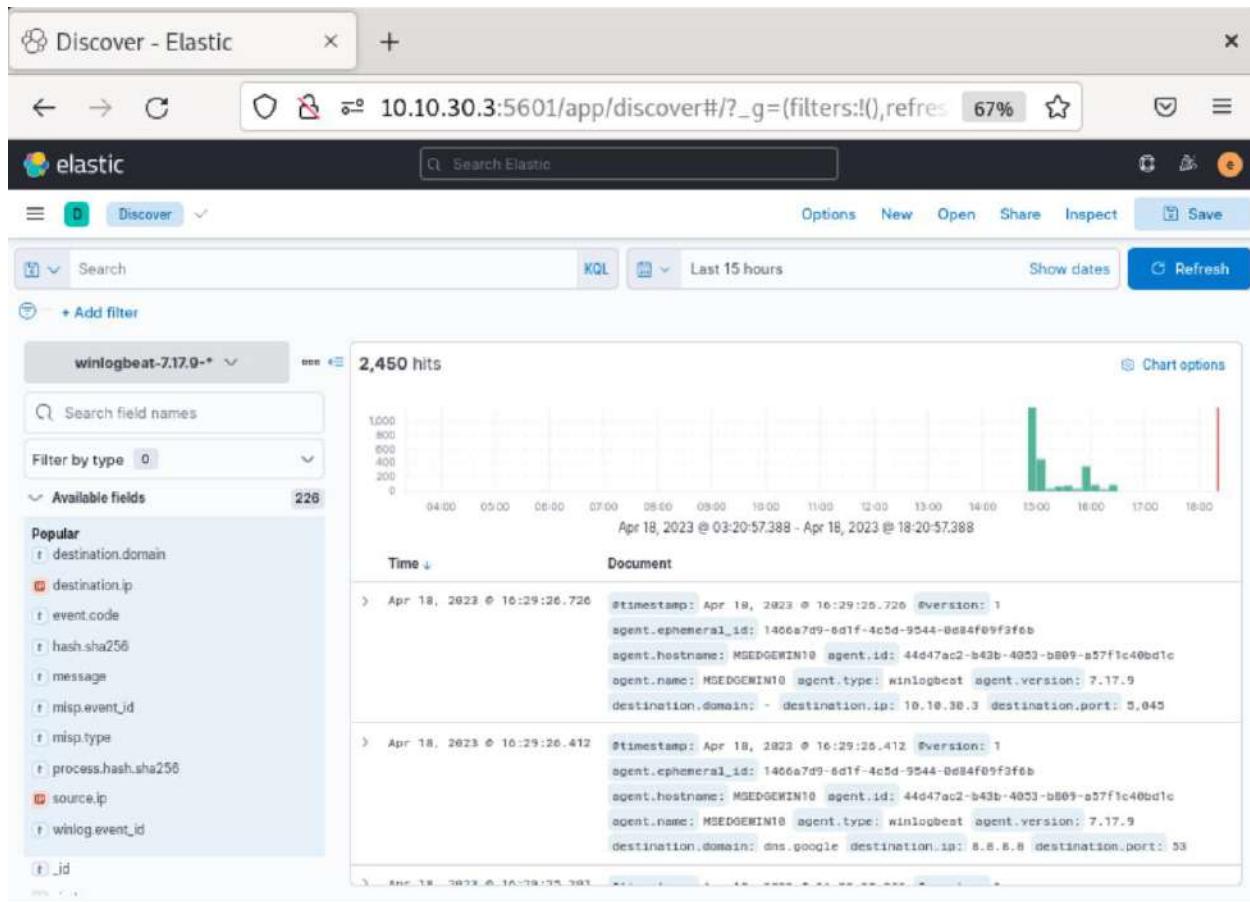


Figure 99: Winlogbeat logs are displayed.

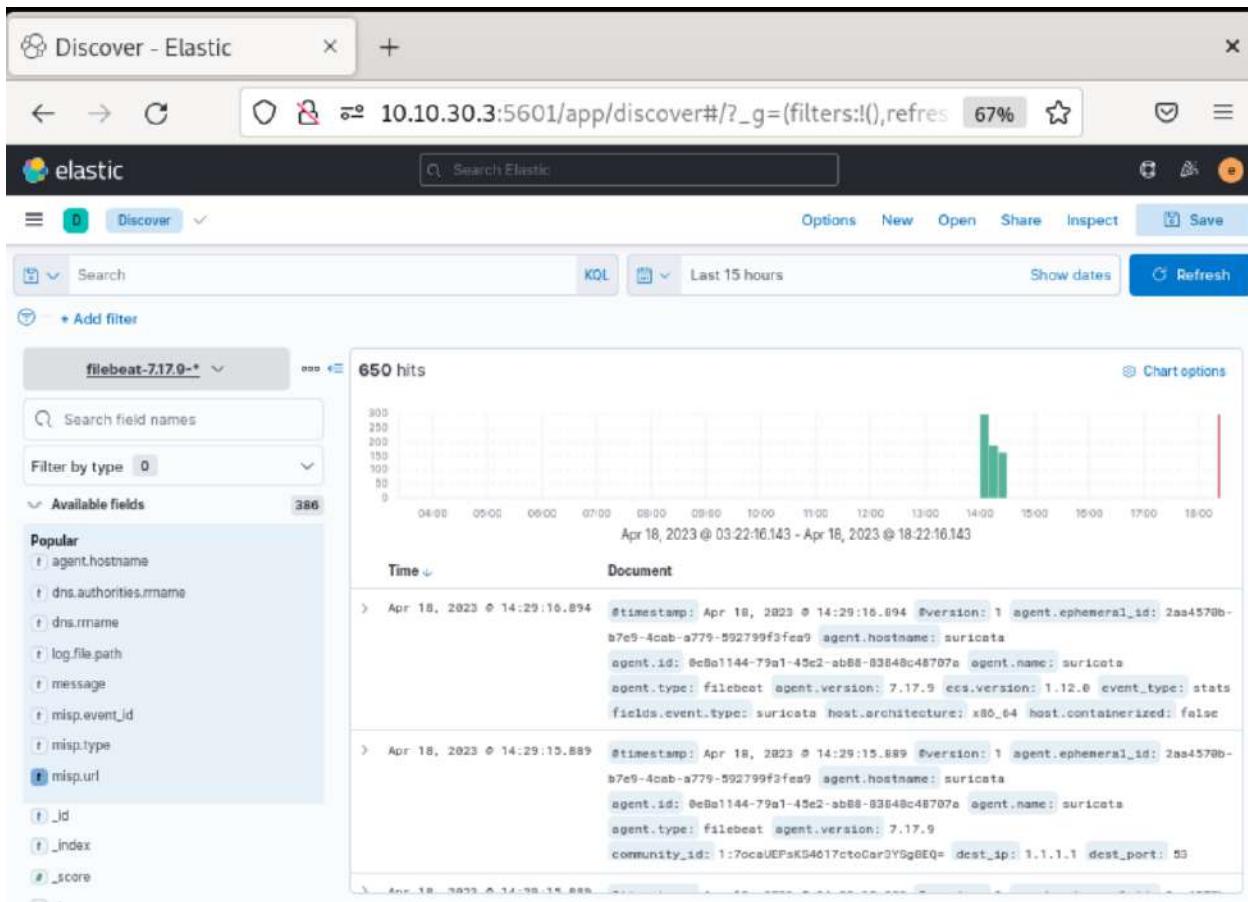


Figure 100: Filebeat logs for suricata are displayed.

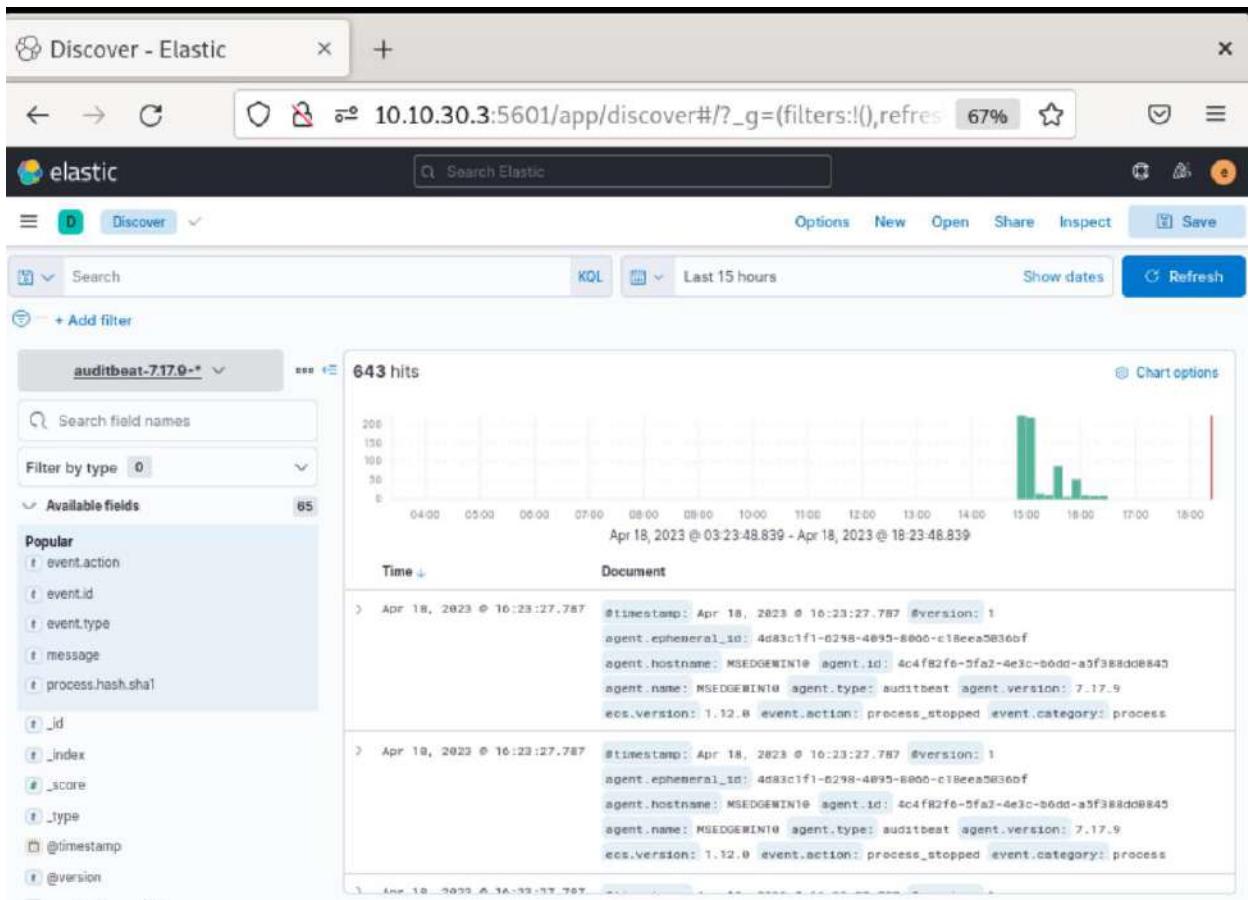


Figure 101: Auditbeat logs are displayed.

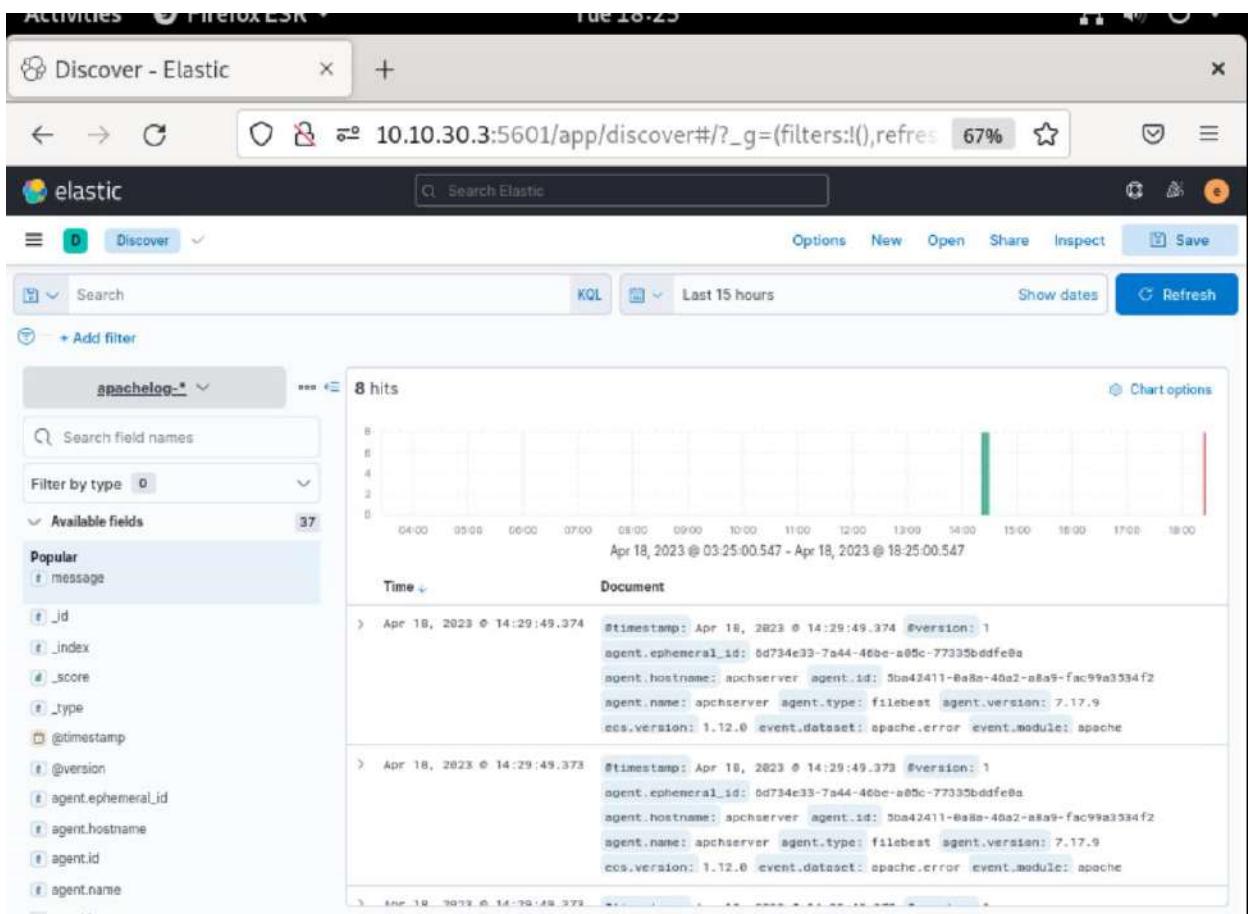
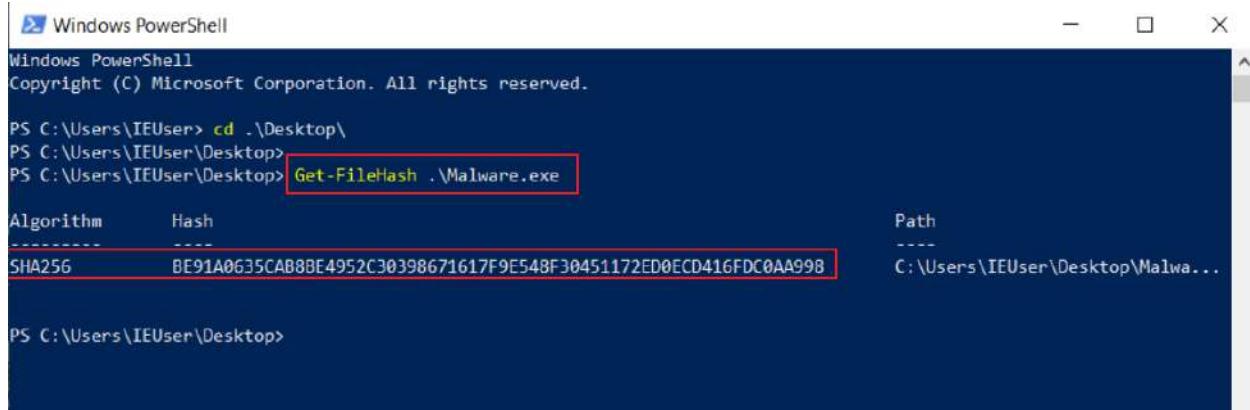


Figure 102: filebeat logs for Apache web server are displayed.

4.3.2 Test Case 2

Test Case 2	
Objective	To test the system if it can detect malware based on hash and provide alert message in telegram.
Action	Malware.exe file was executed
Expected Test Result	A MISP event alert message will be delivered through telegram app.
Actual Test Result	A MISP event alert message will be delivered through telegram app.
Conclusion	Test was successful.

Table 12: Test Case 2



The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The command entered is "Get-FileHash .\Malware.exe". The output table includes columns for Algorithm, Hash, and Path. The SHA256 hash value is highlighted with a red box.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\IEUser> cd ..\Desktop\
PS C:\Users\IEUser\Desktop> Get-FileHash .\Malware.exe
Algorithm      Hash
-----      -----
SHA256        BE91A0635CAB8BE4952C30398671617F9E548F30451172ED0ECD416FDC0AA998
Path
-----
C:\Users\IEUser\Desktop\Malwa...

```

Figure 103: Hash value of executable file.

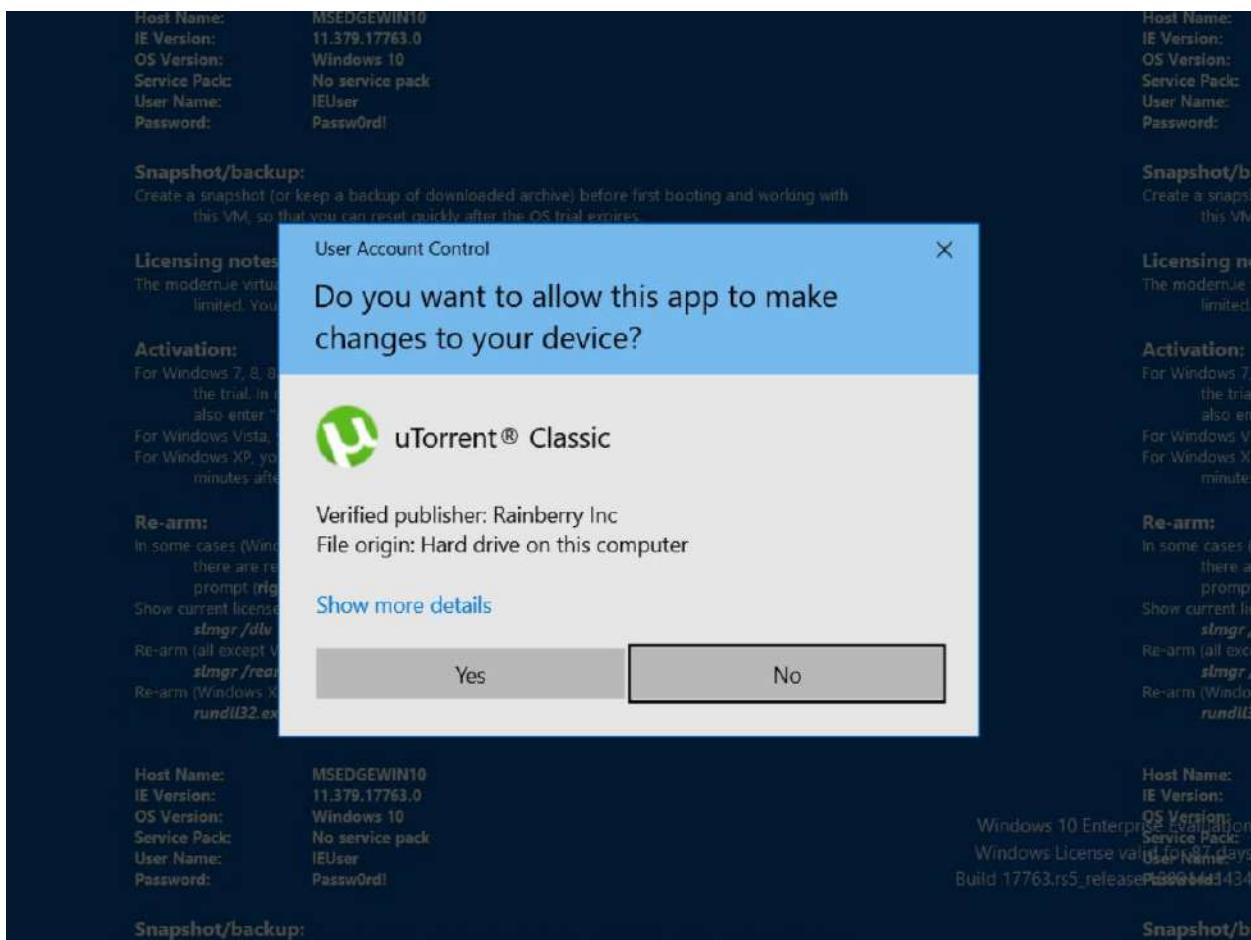


Figure 104: Malware.exe file was executed. note (this was torrent file renamed to malware.exe)

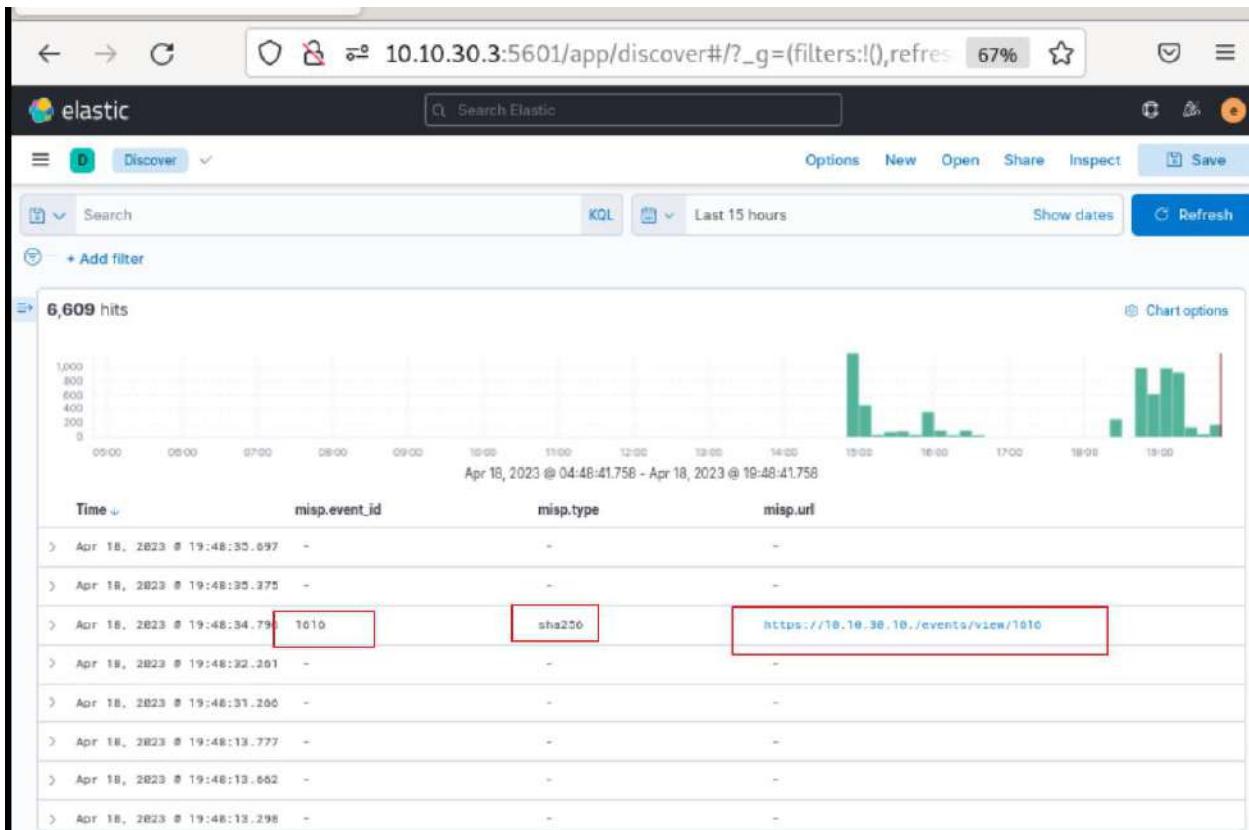


Figure 105: Hash value got hit in the MISP event.

The screenshot shows a Firefox browser window with the title bar "Activities Firefox ESR" and the date "Tue 19:18". The main content is the "Event #1616 - MISP" page, which displays the following information:

- Event ID:** 1616
- UUID:** 44bb90c0-328e-4403-9811-a1d3b470150c
- Creator org:** ORGNAME
- Owner org:** ORGNAME
- Creator user:** admin@admin.test
- Protected Event (experimental):** Event is in unprotected mode. (Switch to protected mode)
- Tags:** (empty)
- Date:** 2023-04-18
- Threat Level:** High
- Analysis:** Initial
- Distribution:** This community only

A sidebar on the left lists various actions: View Event, View Correlation Graph, View Event History, Edit Event, Delete Event, Add Attribute, Add Object, Add Attachment, Add Event Report, Populate from..., Enrich Event, Merge attributes from..., Publish Event, Publish (no email), Contact Reporter, Download as..., List Events, and Add Event.

The URL in the address bar is <https://10.10.30.10/events/view/1616>.

Figure 106: Visiting to "misp.url" link for more info.

The screenshot shows the MISP web interface for Event #1616. The top navigation bar includes tabs for Home, Event Actions, Dashboard, Galaxies, Input Filters, Global Actions, Sync Actions, Administration, and Logs. Below the navigation is a search bar with the URL https://10.10.30.10/events/view/1616. A sidebar on the left titled 'Galaxies' contains icons for creating new galaxies and adding users. The main content area displays a table of event details. The table has columns for Date, Category, Type, and Value. One row is highlighted with a red border, showing the date 2023-04-18, category Payload delivery, type sha256, and value be91a0635cab8be4952c30398671617f9e548f30451172ed0ecd416fdc0aa9. The value column also shows a decay score of 98. Navigation links at the bottom include '« previous', 'next »', and 'view all'. A footer bar at the bottom of the page includes links for 'Download: PGP public key' and 'This is an initial install Powered by MISP 2.4.169 Please configure and harden according'.

Figure 107: Hash value of malware.exe found here.

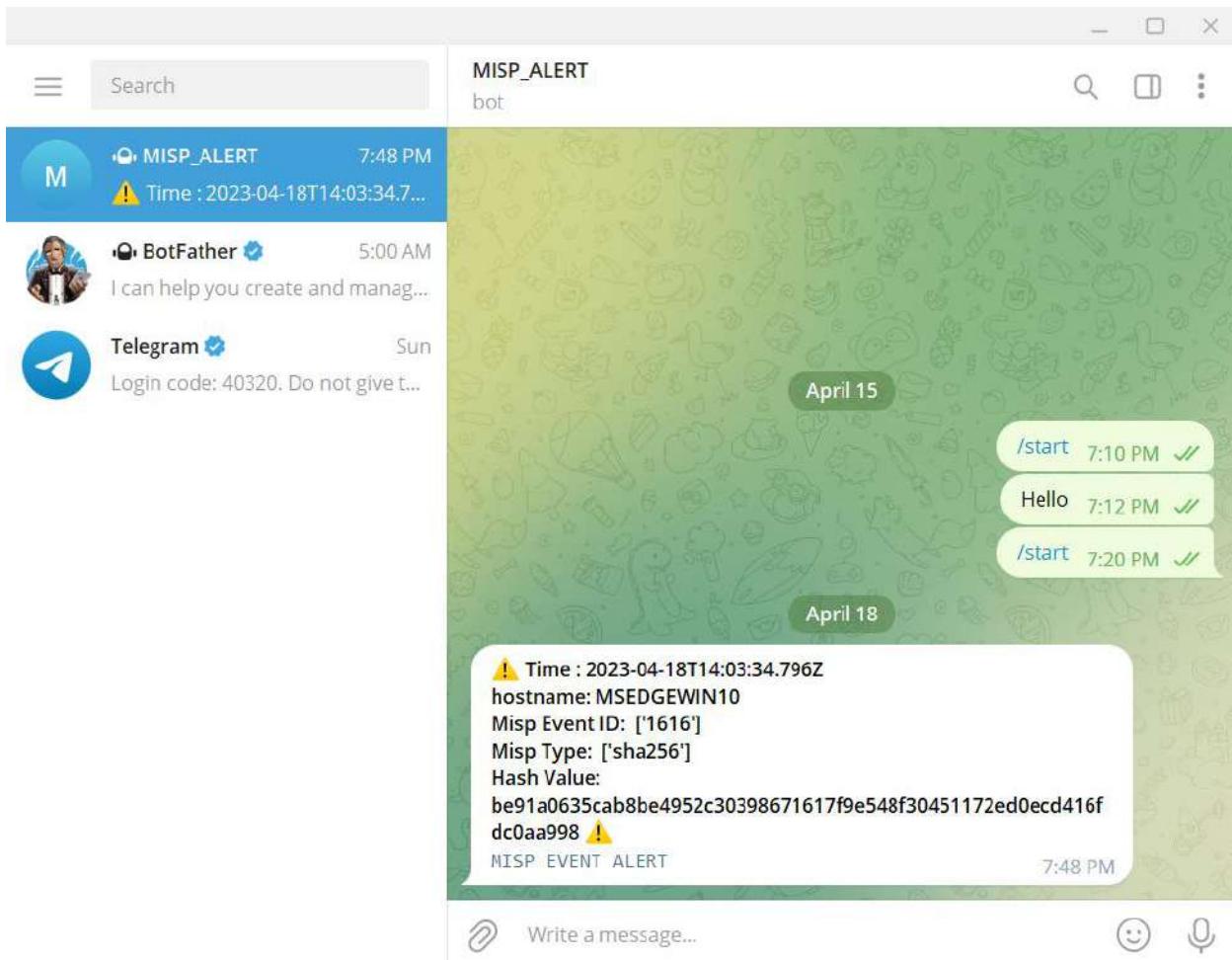


Figure 108: MISP Event Alert in Telegram.

4.3.3 Test Case 3

Test Case 3	
Objective	To test the system if it can alert the user through telegram message when file is been modified in host system.
Action	Malware.txt file was created and deleted.
Expected Test Result	An alert message will be delivered through telegram app when file will be created or deleted.
Actual Test Result	An alert message will be delivered through telegram app when file will be created or deleted.
Conclusion	Test was successful.

Table 13: Test Case 3

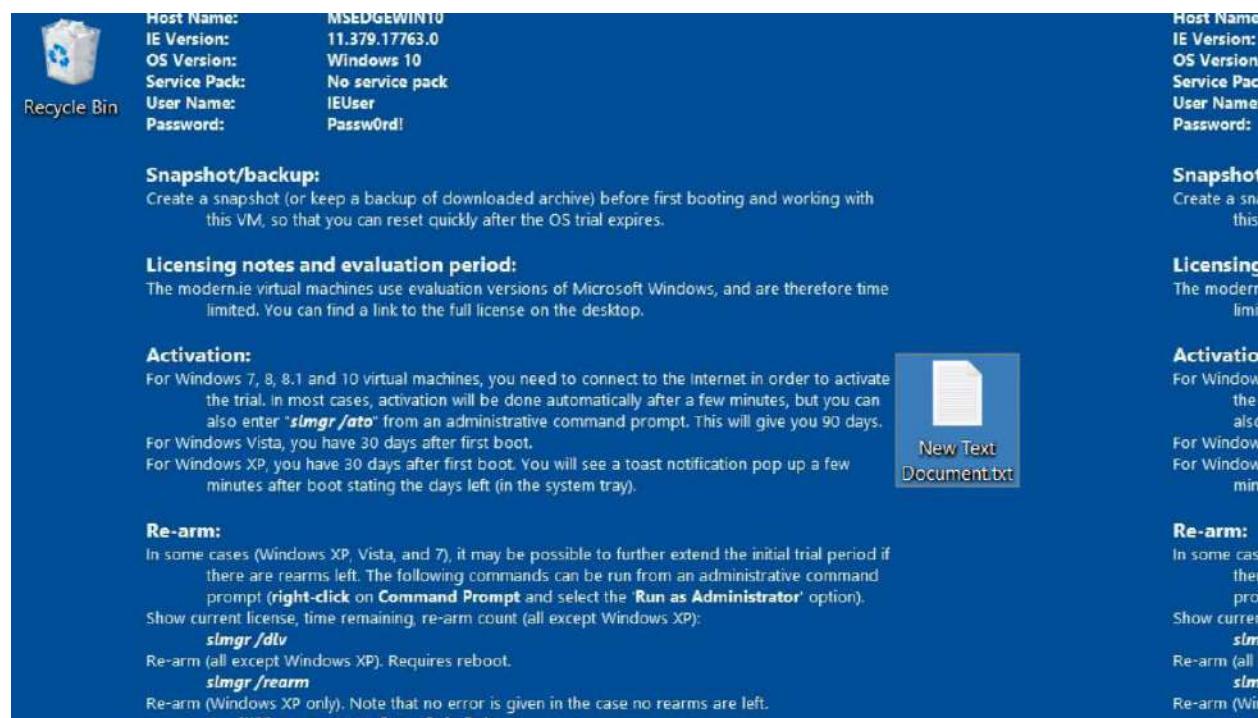


Figure 109: New Text Document.txt file created.

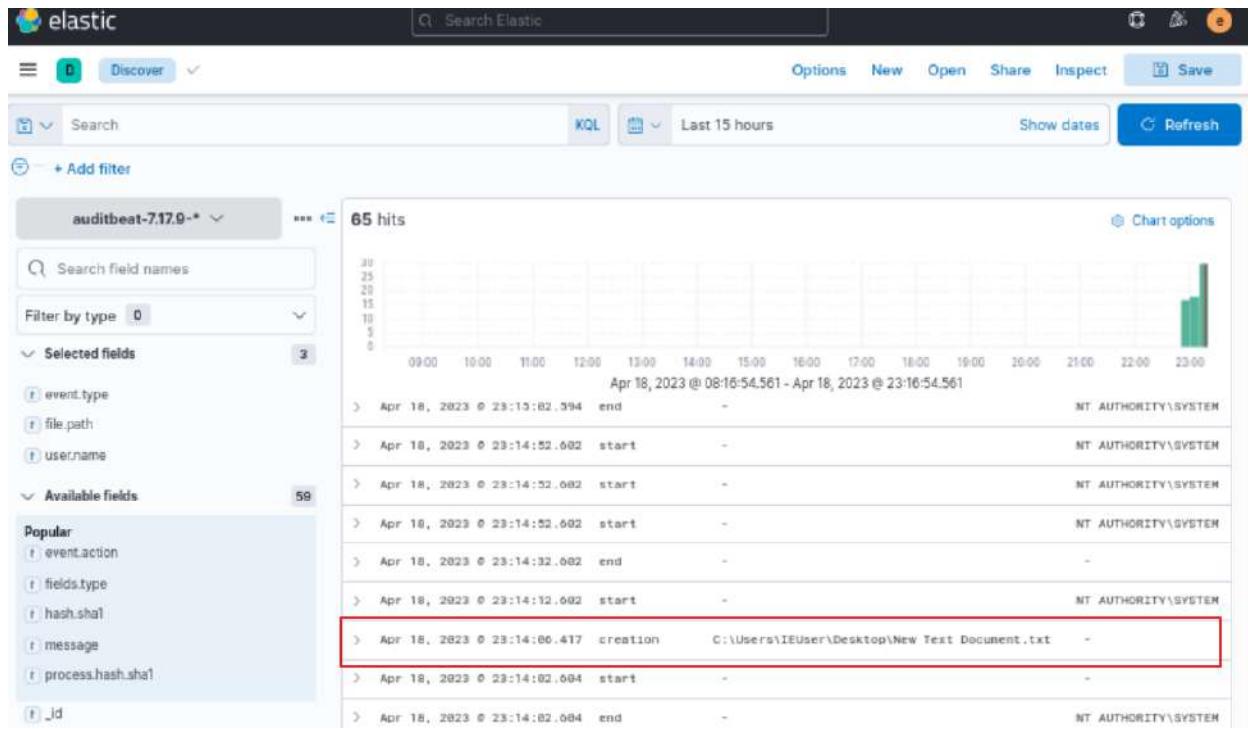


Figure 110: Creation Event was recorded in ELK stack.

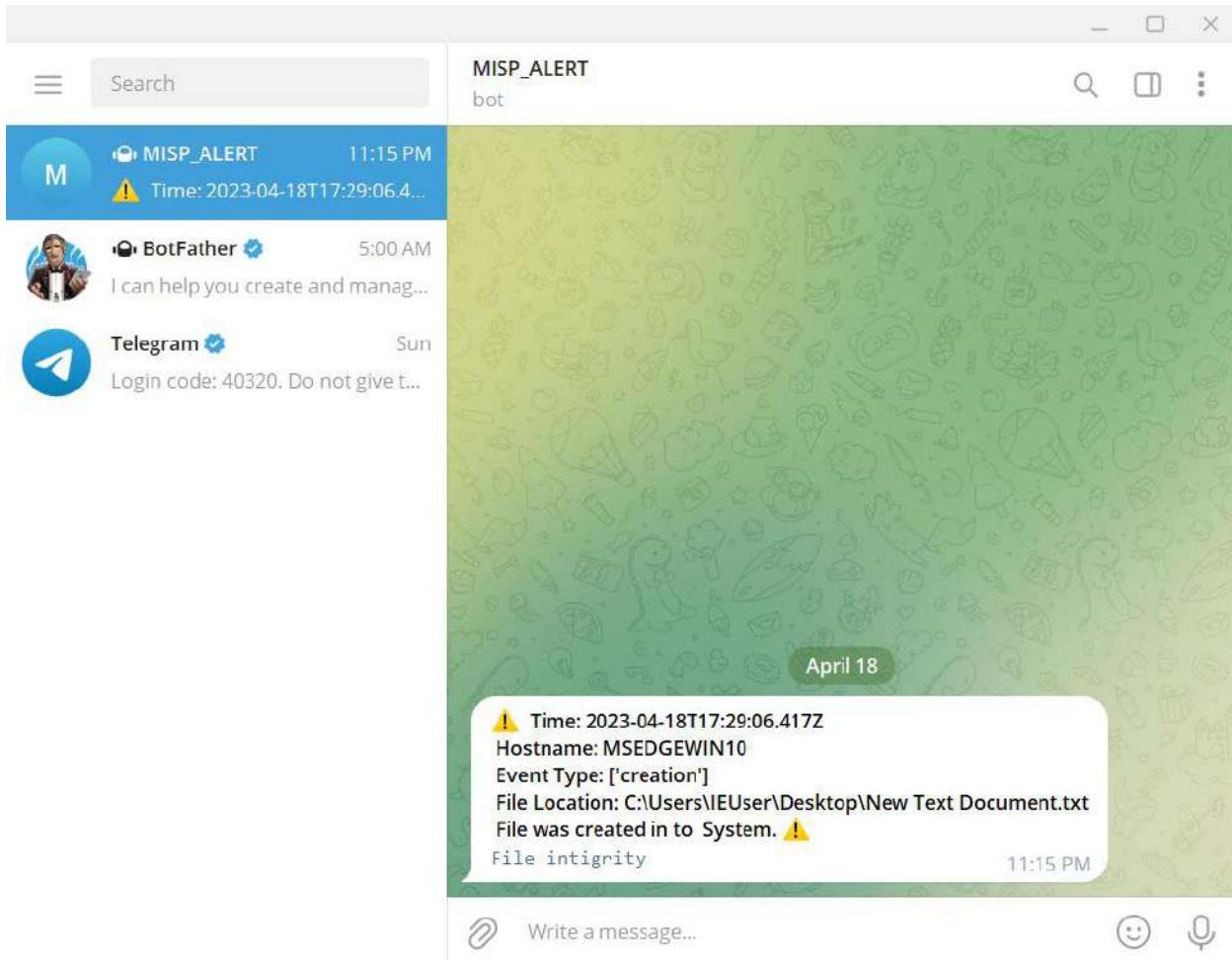


Figure 111: Telegram alert for creation of file in system.

 <p>Host Name: MSEdgeWIN10 IE Version: 11.379.17763.0 OS Version: Windows 10 Service Pack: No service pack User Name: IEUser Password: PasswOrd!</p> <p>Snapshot/backup: Create a snapshot (or keep a backup of downloaded archive) before first booting and working with this VM, so that you can reset quickly after the OS trial expires.</p> <p>Licensing notes and evaluation period: The modern.ie virtual machines use evaluation versions of Microsoft Windows, and are therefore time limited. You can find a link to the full license on the desktop.</p> <p>Activation: For Windows 7, 8, 8.1 and 10 virtual machines, you need to connect to the Internet in order to activate the trial. In most cases, activation will be done automatically after a few minutes, but you can also enter <code>slmgr /ato</code> from an administrative command prompt. This will give you 90 days. For Windows Vista, you have 30 days after first boot. For Windows XP, you have 30 days after first boot. You will see a toast notification pop up a few minutes after boot stating the days left (in the system tray).</p> <p>Re-arm: In some cases (Windows XP, Vista, and 7), it may be possible to further extend the initial trial period if there are rearms left. The following commands can be run from an administrative command prompt (right-click on Command Prompt and select the 'Run as Administrator' option). Show current license, time remaining, re-arm count (all except Windows XP): <code>slmgr /dlv</code> Re-arm (all except Windows XP). Requires reboot. <code>slmgr /rearm</code> Re-arm (Windows XP only). Note that no error is given in the case no rearms are left. <code>rundll32.exe syssetup,SetupObbeBnk</code></p>	<p>Host Name: IE Version: OS Version: Service Pack: User Name: Password:</p> <p>Snapshot/backup: Create a snapshot (or keep a backup of downloaded archive) before first booting and working with this VM.</p> <p>Licensing notes and evaluation period: The modern.ie virtual machines use evaluation versions of Microsoft Windows, and are therefore time limited.</p> <p>Activation: For Windows 7, the trial also ent For Windows Vi For Windows XP minutes</p> <p>Re-arm: In some cases (V there ar prompt Show current lic <code>slmgr /</code> Re-arm (all exce <code>slmgr /</code> Re-arm (Window <code>rundll3</code></p>
---	---

Figure 112: File was Deleted.

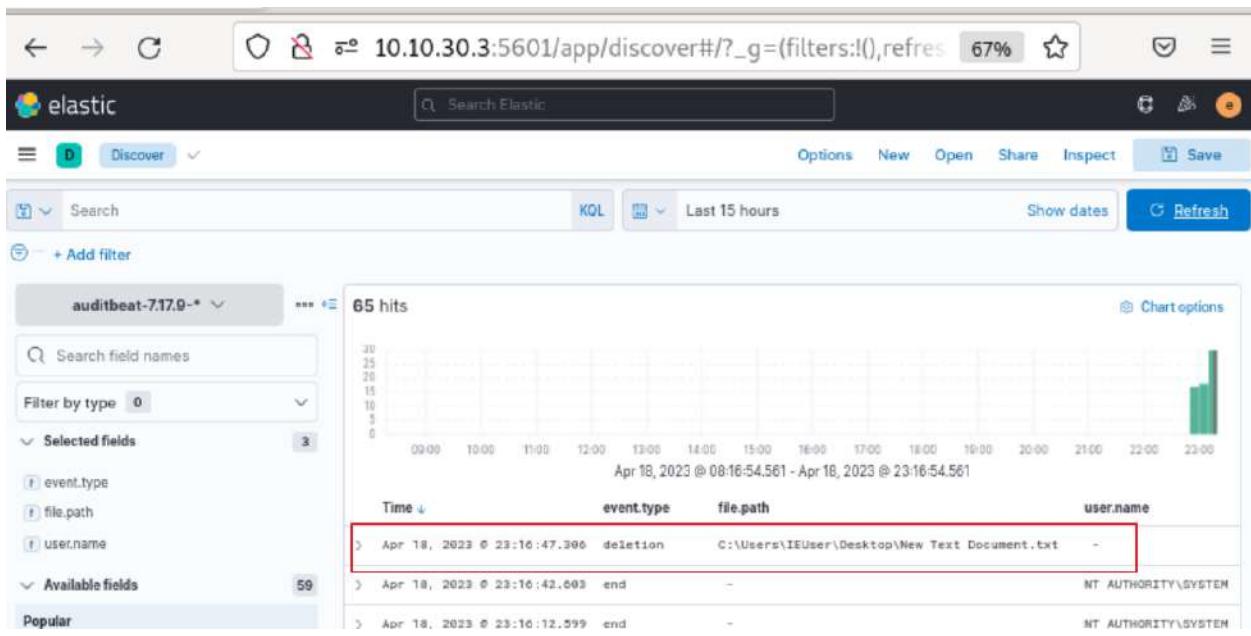


Figure 113: Deletion event was recorded in EKL stack.

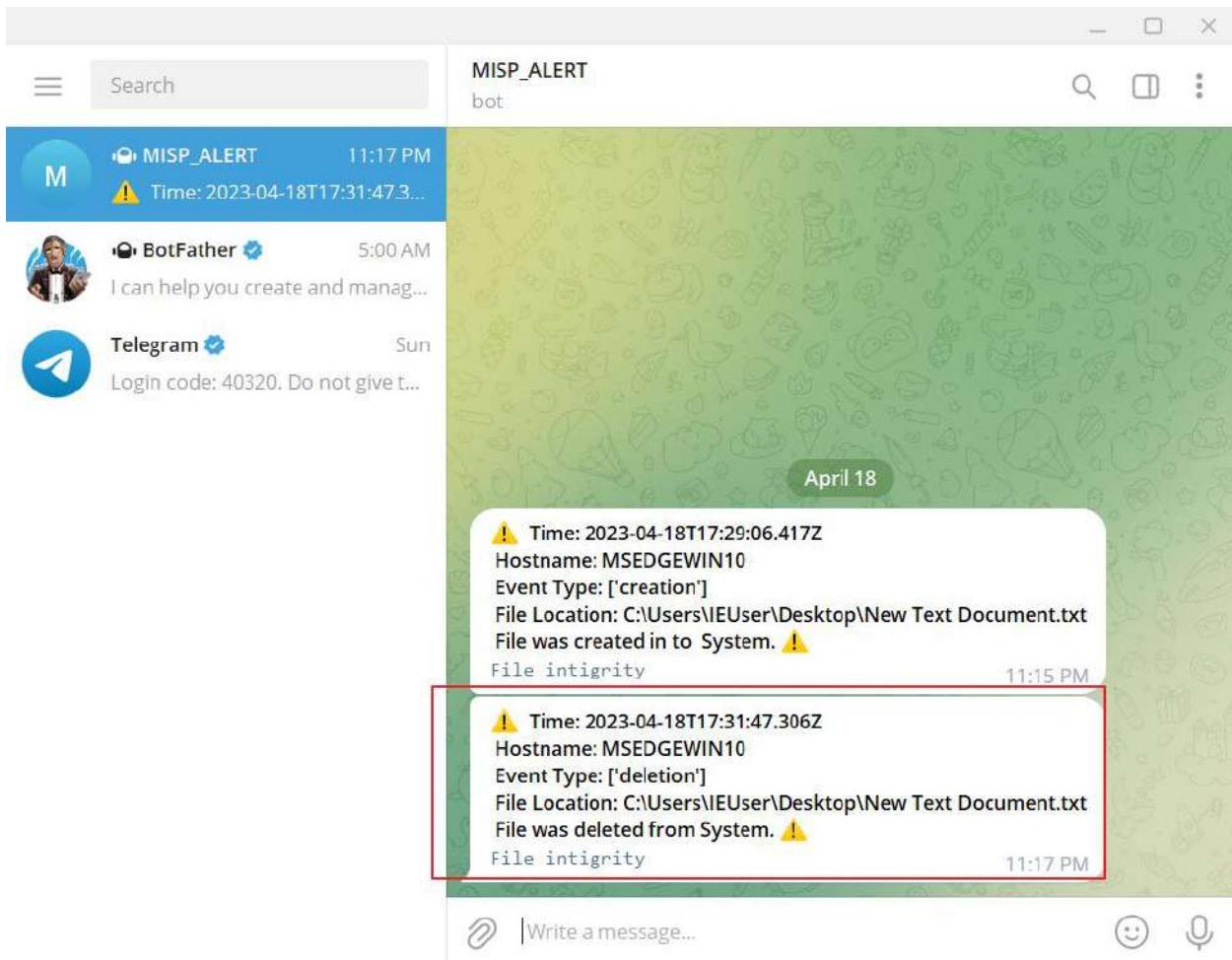


Figure 114: Telegram alert after deletion of file.

4.3.4 Test Case 4

Test Case 4	
Objective	To test the system if it can alert the user through telegram message when USB devices plugged in to system.
Action	USB device was plugged in to system.
Expected Test Result	An alert message will be delivered through telegram app when usb device is plugged in to system.
Actual Test Result	An alert message was delivered through telegram app when usb device is plugged in to system.
Conclusion	Test was successful.

Table 14: Test Case 4

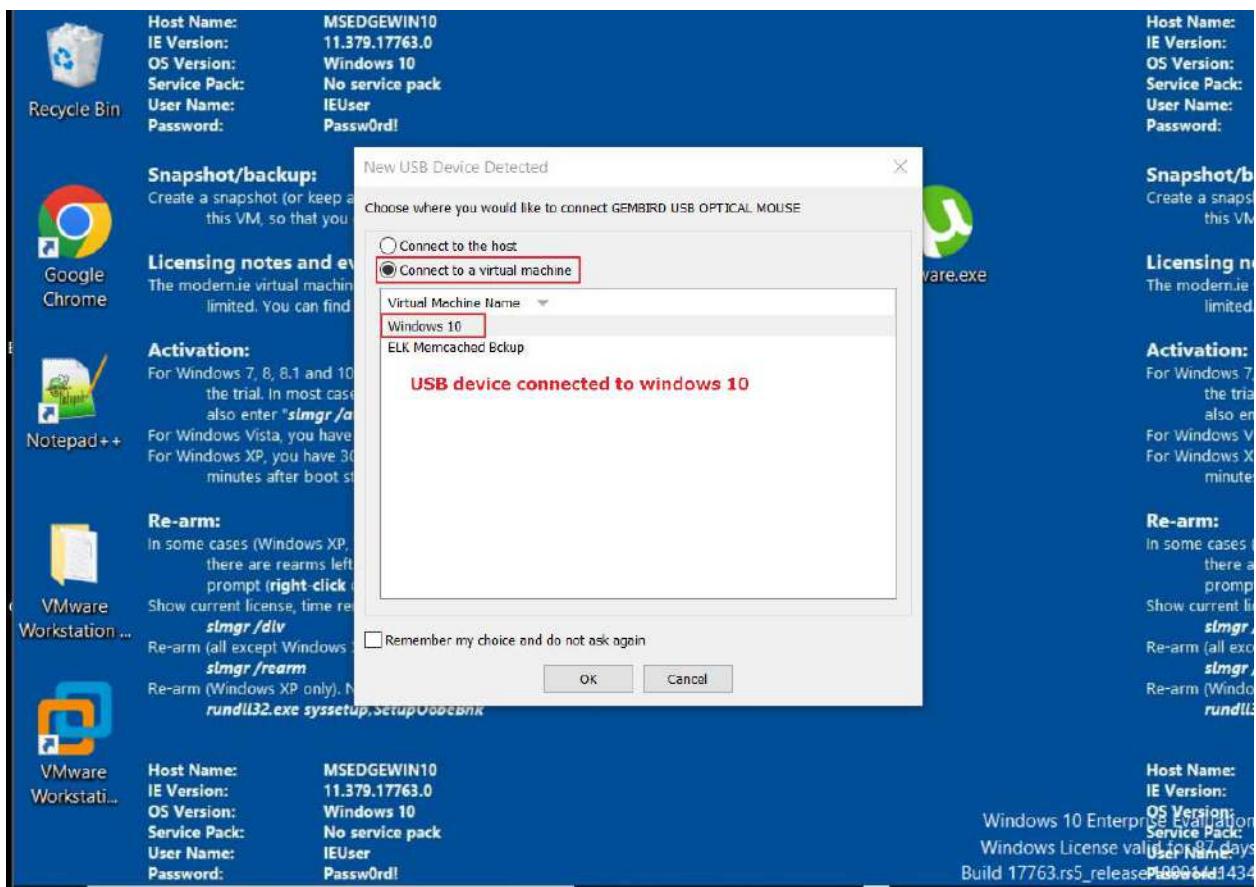


Figure 115: USB device plugging in to win 10.

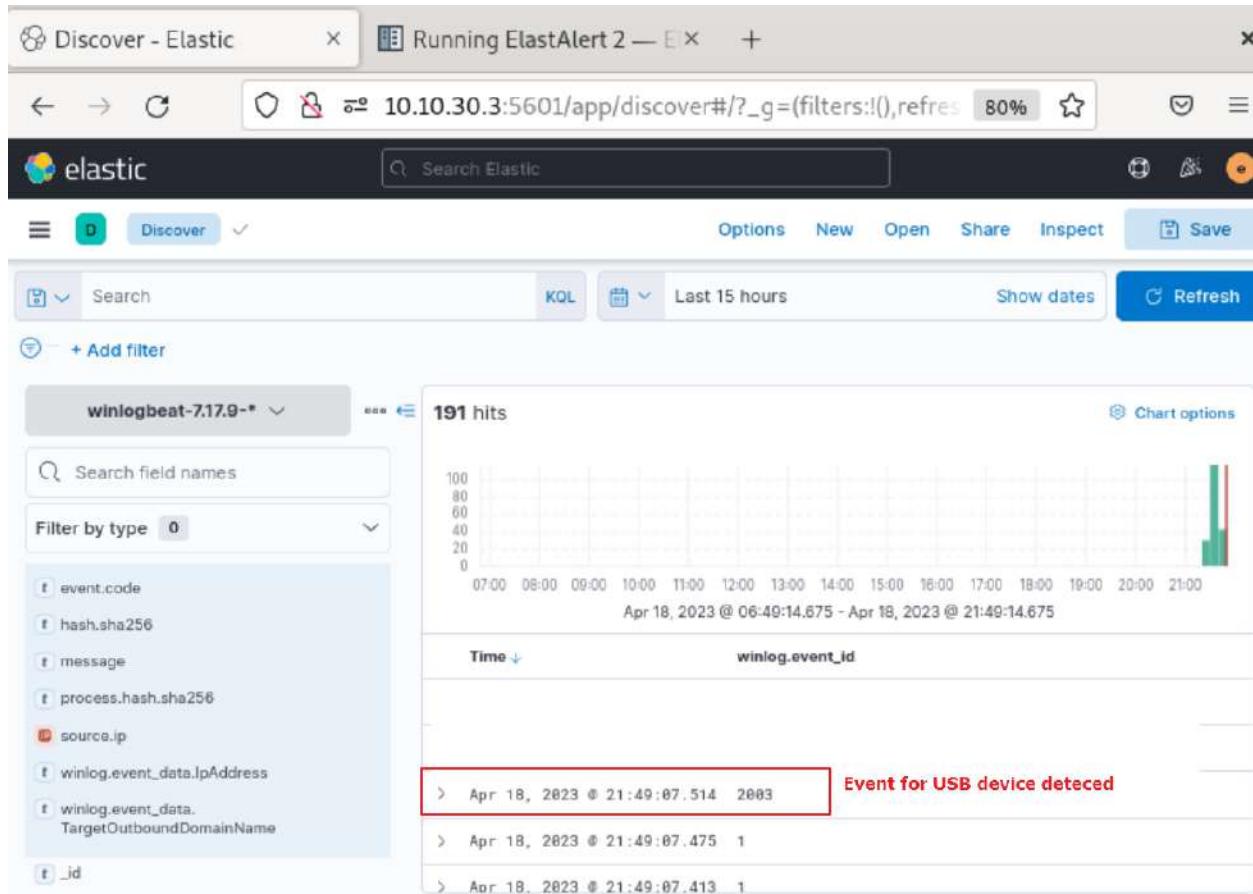


Figure 116: An alert event with id 2003 was triggered in ELK stack dashboard.

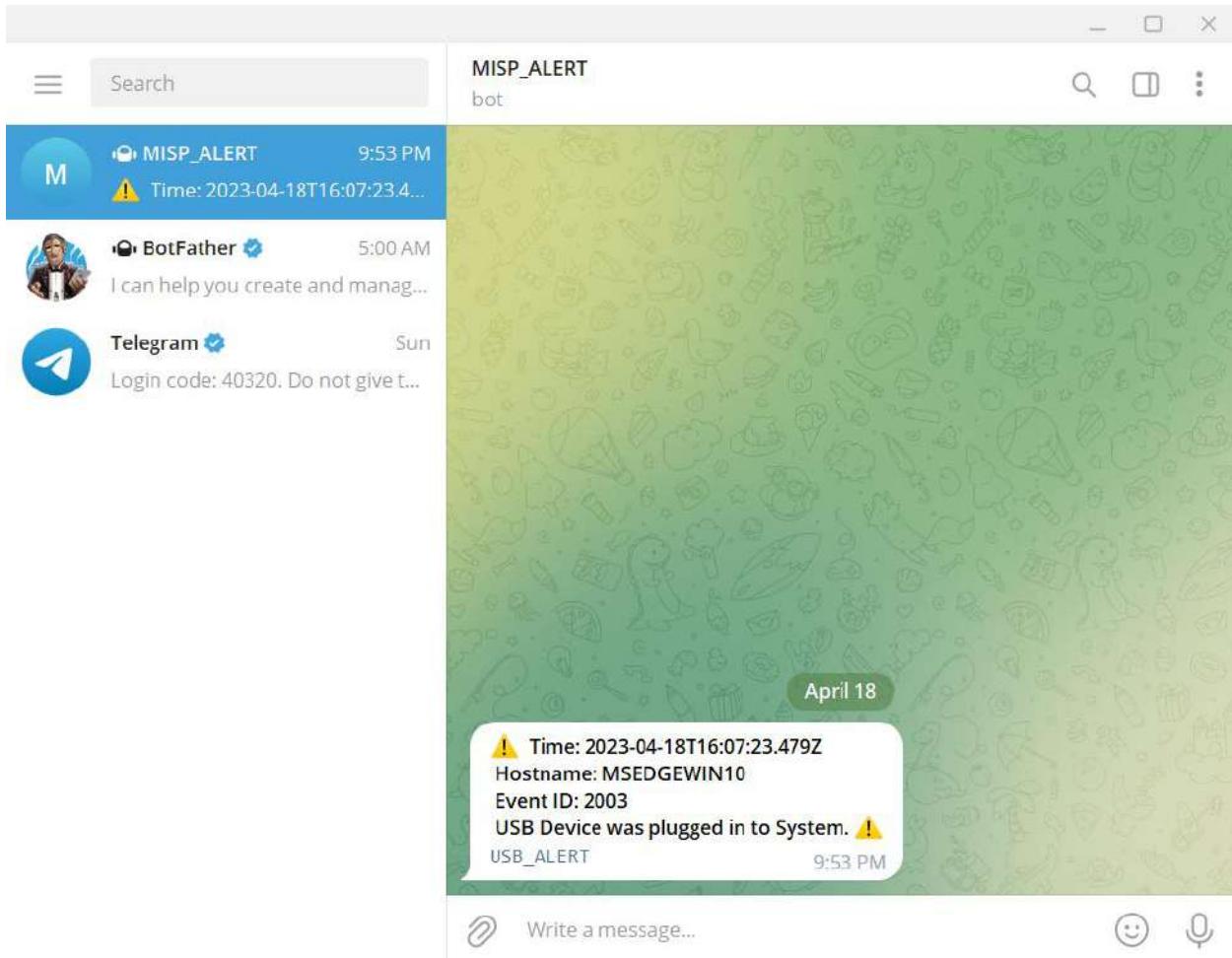


Figure 117: Telegram message was delivered when USB was plugged in to system.

4.3.5 Test Case 5

Test Case 5	
Objective	To test for verifying if all the ELK services are running properly without errors.
Action	Checked the status of Elasticsearch, Logstash and Kibana services are running properly.
Expected Test Result	All three services should be in active and running state.
Actual Test Results	All three services was in active and running state.
Conclusion	Test was successful

Table 15: Test Case 5

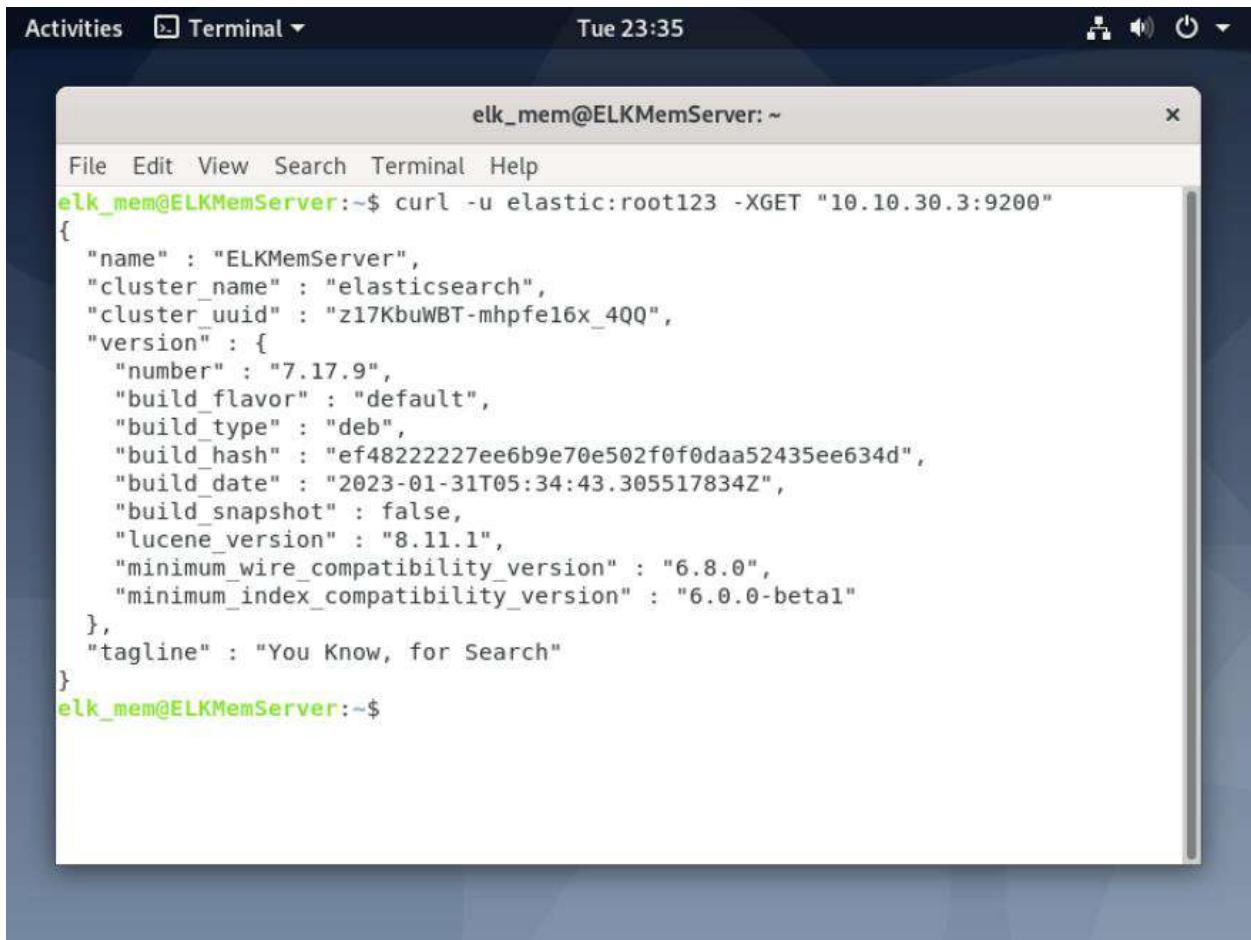
```

elk_mem@ELKMemServer:~$ sudo systemctl status elasticsearch.service
[sudo] password for elk_mem:
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor pr
   Active: active (running) since Tue 2023-04-18 19:12:02 +0545; 4h 21min ago
     Docs: https://www.elastic.co
 Main PID: 698 (java)
    Tasks: 92 (limit: 7533)
   Memory: 1.1G
      CPU: 1.1G
      CGroup: /system.slice/elasticsearch.service
              └─ 698 /usr/share/elasticsearch/jdk/bin/java -Xshare:auto -Des.networ
                  └─ 1151 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64

Apr 18 19:11:38 ELKMemServer systemd[1]: Starting Elasticsearch...
Apr 18 19:12:02 ELKMemServer systemd[1]: Started Elasticsearch.
lines 1-13/13 (END)

```

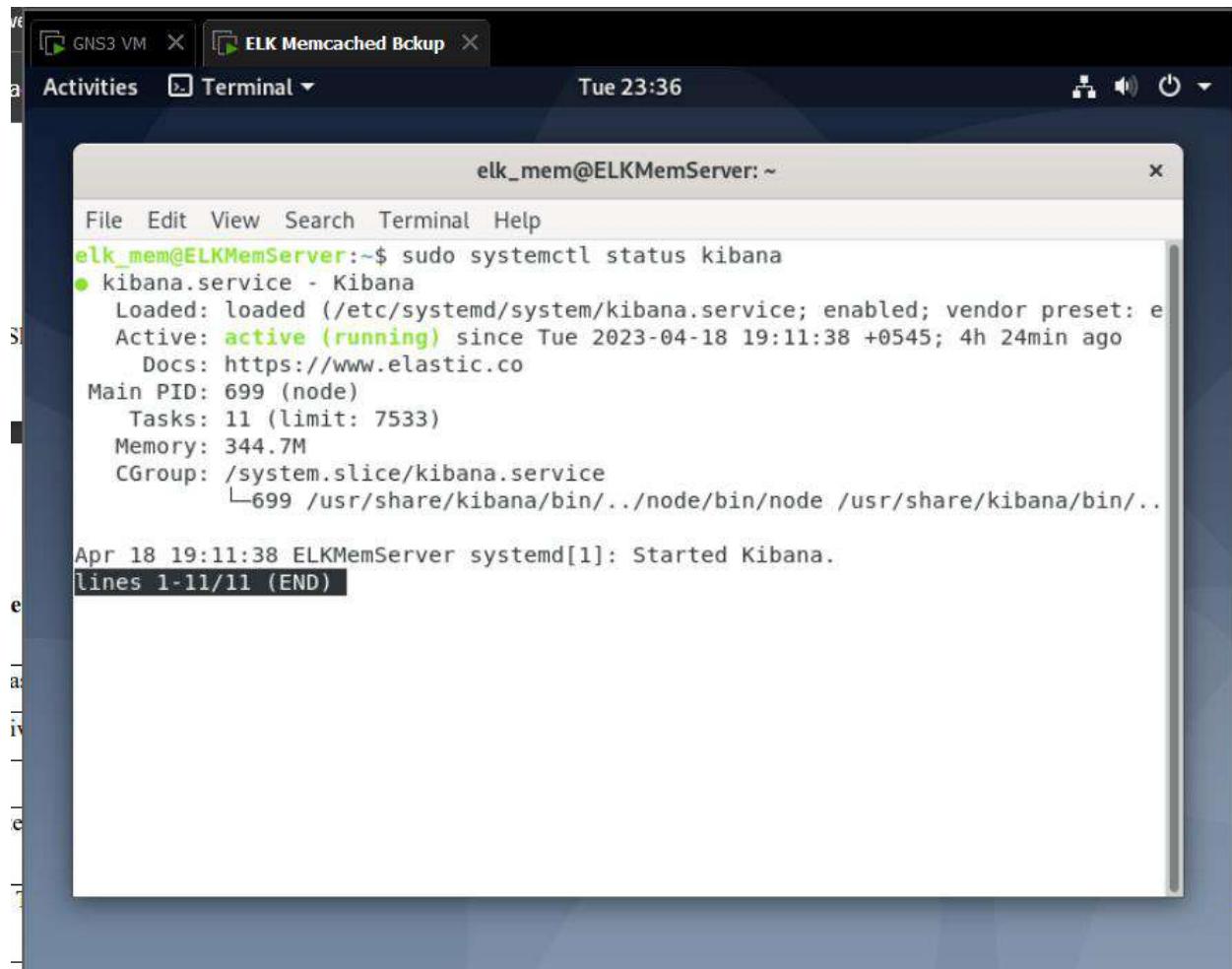
Figure 118: Elasticsearch was active and running.



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window contains the following text:

```
File Edit View Search Terminal Help
elk_mem@ELKMemServer:~$ curl -u elastic:root123 -XGET "10.10.30.3:9200"
{
  "name" : "ELKMemServer",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "z17KbuWBT-mhpfe16x_4QQ",
  "version" : {
    "number" : "7.17.9",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "ef4822227ee6b9e70e502f0f0daa52435ee634d",
    "build_date" : "2023-01-31T05:34:43.305517834Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
elk_mem@ELKMemServer:~$
```

Figure 119: Testing Elasticsearch configuration.



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window contains the following text:

```
File Edit View Search Terminal Help
elk_mem@ELKMemServer:~$ sudo systemctl status kibana
● Kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: e
   Active: active (running) since Tue 2023-04-18 19:11:38 +0545; 4h 24min ago
     Docs: https://www.elastic.co
   Main PID: 699 (node)
      Tasks: 11 (limit: 7533)
     Memory: 344.7M
        CPU: 699 /system.slice/kibana.service
              └─ 699 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/..
Apr 18 19:11:38 ELKMemServer systemd[1]: Started Kibana.
lines 1-11/11 (END)
```

Figure 120: Kibana services was active and running.

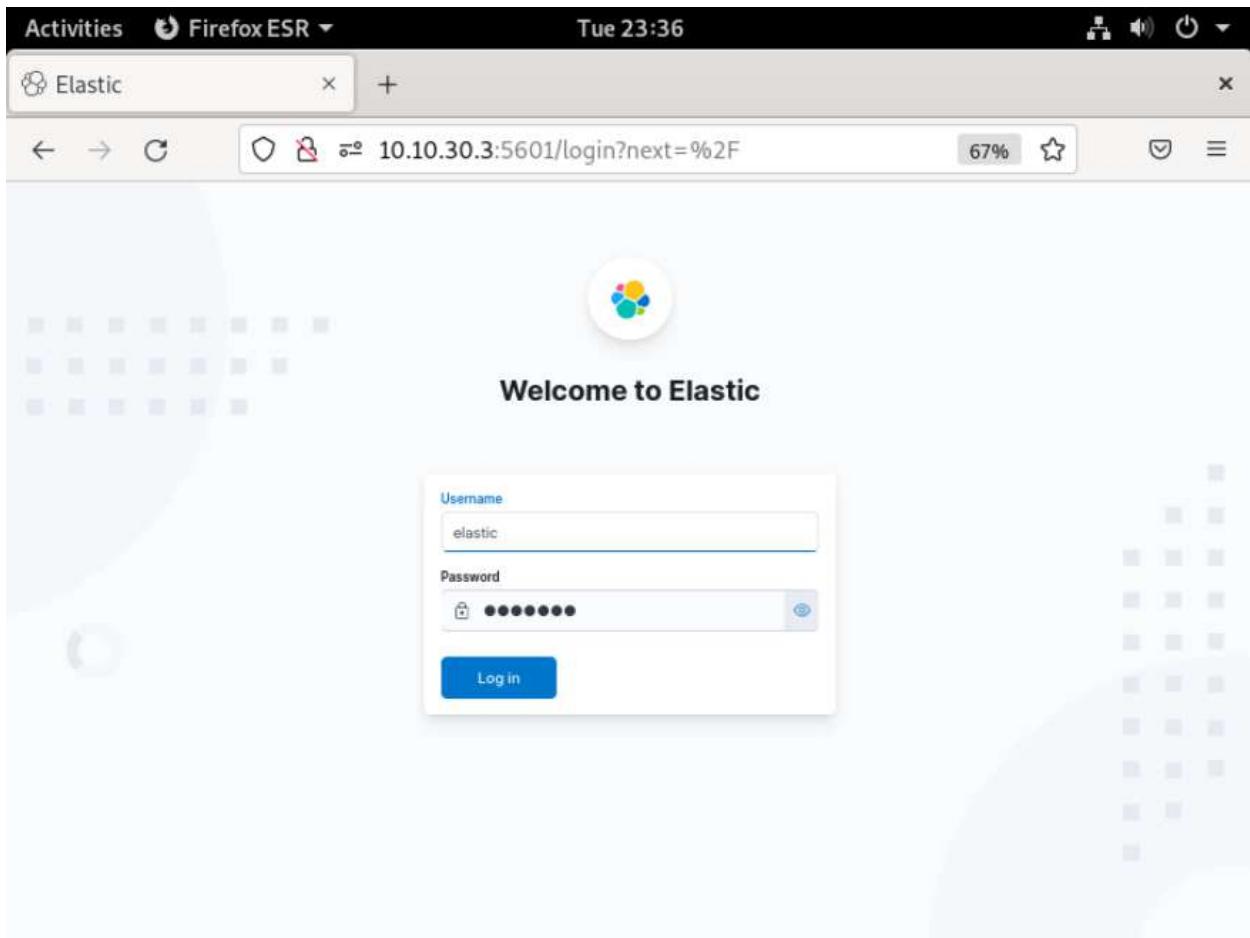


Figure 121: Kibana user interface.

The screenshot shows a terminal window with the title 'elk_mem@ELKMemServer: ~'. The window contains the following text:

```
File Edit View Search Terminal Help
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: enabled)
   Drop-In: /etc/systemd/system/logstash.service.d
     └─synlite_suricata.conf
     Active: active (running) since Tue 2023-04-18 19:11:36 +0545; 4h 25min ago
       Main PID: 502 (java)
          Tasks: 116 (limit: 7533)
         Memory: 1.3G
        CGroup: /system.slice/logstash.service
                  └─502 /usr/share/logstash/jdk/bin/java -Xms3g -Xmx3g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75

Apr 18 19:12:30 ELKMemServer logstash[502]: [2023-04-18T19:12:30,049][INFO ][logstash.javapipeline      ][win] Pipeline started
Apr 18 19:12:30 ELKMemServer logstash[502]: [2023-04-18T19:12:30,052][INFO ][logstash.javapipeline      ][suricata] Pipeline starte
Apr 18 19:12:30 ELKMemServer logstash[502]: [2023-04-18T19:12:30,076][INFO ][logstash.inputs.beats      ][apache] Starting inpu
Apr 18 19:12:30 ELKMemServer logstash[502]: [2023-04-18T19:12:30,083][INFO ][logstash.javapipeline      ][audit] Pipeline start
Apr 18 19:12:30 ELKMemServer logstash[502]: [2023-04-18T19:12:30,096][INFO ][logstash.javapipeline      ][apache] Pipeline star
Apr 18 19:12:30 ELKMemServer logstash[502]: [2023-04-18T19:12:30,308][INFO ][org.logstash.beats.Server][audit][873cc325a701cf
Apr 18 19:12:30 ELKMemServer logstash[502]: [2023-04-18T19:12:30,310][INFO ][org.logstash.beats.Server][win][fa0eld3356ebb05f
Apr 18 19:12:30 ELKMemServer logstash[502]: [2023-04-18T19:12:30,322][INFO ][org.logstash.beats.Server][suricata][0e3bf7d212b27a5
Apr 18 19:12:30 ELKMemServer logstash[502]: [2023-04-18T19:12:30,325][INFO ][org.logstash.beats.Server][apache][2c5e4319b9fd2
Apr 18 19:12:30 ELKMemServer logstash[502]: [2023-04-18T19:12:30,418][INFO ][logstash.agent              ] Pipelines running {:c
~
~
lines 1-21/21 (END)
```

Figure 122: Logstash was active and running.

4.4 Critical Analysis

Over the time of testing stages, the project encountered several technical problems. However, the issues and mistakes were being resolved, and the system was being developed accordingly. Overall, the project's components passed all its self-conducted unit and system tests. The system is fully functional and incorporates all the functionality specified in the proposal. The prototype version of the project underwent user acceptance testing, in which potential end users were encouraged to interact with the system and provide feedback. The user feedback showed a few strengths and weaknesses in the system. All of the flaws are the result of time constraints that might have been overcome to improve this project. The second survey result indicate the most like feature of this system was MISP lookup which help to match to data in MISP server to detect threat.

This main goal of this project was developing a system that will keep the network or end devices safe by detecting threat real time. For this ELK stack was selected as SIEM solution where logs of suicata, apache web server and windows log are centralized with help of log agents and to integrate MISP platform for threat intel, Memcached for cached server and telegram for alerting function. Initially, by looking youtube videos, readymade script and blog related to development of system similar to his project, it was assume the development procedure for this project will be easy and smooth but that was not a case. At the time development of this system, several problems and difficulties were encountered like script for fetching IOCs was not working as it should be, configuration and dependencies error with ElastAlert , Suricata, MISP server and ELK stack, log parsing error. After many debugging and testing of script of MISP was able to fetch IOCs data from local MISP server. When getting log from different log agent like winlogbeat, filebeat and auditbeat at the same time was very difficult than get log from only one agent. Many logs error was encountered at that time during that time, as all Logstash code was written in one single conf file. Later after many research and tutorial video it was solved by separating the config file for each log agent and creating the multiple pipeline that point config file of each log agent. At the time of installing suricata , problem was encounter where suricata was installed but its "emerging threat " rules was not found, so after many research found the "emerging threat " rules can be installed manually and it was solved. ELK stack also encountered error while placing private ip 10.10.30.3 in place localhost where only using localhost can

access kibana dashboard. This problem was solved by chaining network.host to 0.0.0.0 in elasticsearch..yml and 10.10.30.3 kibana.yml file. During the installation of ElastAlert many python dependencies error was to be found even though official page of ElastAlert2 page instruction was followed. After many hit and trial test, it was found that each dependencies for the python to use ElastAlert needed to be manually install one by one so that one package error cannot affect other. Some issues was also encountered when install and configuring pfSense on vm and running it on GNS3 where only one interface will work, it was solve by using GNS3 VM and other many small problem were encountered which were solved by watching tutorial video, YouTube, bogland official page. All problem encountered was resolve in order to complete project in time.

Chapter 5: Conclusion

As our reliance on computer network and other computer devices rises, rigorous network and end devices security becomes more critical. Many attacks are being carried out to compromise devices and network of an organization. Every day new threats are created and causes many losses to bank, hospital, government. To keep up with such new threat intel, idea of integration of MISP with ELK was born to detect new threats match data in its database. The propose of the project was to implement a system that can detect new threats and generate alert to user.

The MISP-based real-time threat detection initiative was a success. The implementation of MISP has provided a robust platform for threat intelligence sharing and facilitated real-time threat detection. The application of machine learning algorithms has enabled the automatic classification of hazards and drastically reduced the amount of time required for manual analysis.

The initiative illustrated the significance of collaboration in the battle against cyber threats. By exchanging threat intelligence between organizations, the efficacy of threat detection and response can be significantly enhanced. Utilizing MISP has made this collaboration feasible and enabled organizations to work more efficiently and effectively together.

Overall, the initiative demonstrated that real-time threat detection with MISP is an effective cyber security strategy. By leveraging the power of machine learning and collaboration, organizations can secure their networks and data from malicious actors and remain ahead of the constantly evolving threat landscape

5.1 Legal, Social and Ethical Issues

5.1.1 Legal Issues

The collection, processing, and storage of personal information may be required when using real-time threat detection with threat intelligence. It is critical to ensure that individuals' privacy is maintained and that their data is utilized legitimately and responsibly. It is critical to follow applicable data protection rules and regulations. The data collected by the real-time threat detection system must be saved and kept for a set length of time. It is critical to ensure that the data retention rules adhere to all applicable laws and

regulations. The usage of threat intelligence feeds may necessitate the use of intellectual property or proprietary data. It is critical that the project does not violate any third-party intellectual property rights.

5.1.2 Social Issues

The deployment of a real-time threat detection system that gathers data from multiple sources, including social media, may pose privacy concerns. People may be uneasy about their personal data being used for threat detection, and there is a possibility that it will be compromised or utilized for inappropriate purposes. The threat detection system could be abused for non-threat detection reasons. For example, the information could be utilized for surveillance or tracking, which could have detrimental social consequences.

.5.1.3 Ethical Issues

If personal information is acquired and processed without consent, the use of threat intelligence may infringe on people's privacy. Threat detection systems are not always accurate, and false positives or negatives can have major repercussions, such as false allegations or missing threats. The threat detection system could be used for personal advantage or other harmful goals, such as spying on individuals or groups.

5.2 Advantages

- Real-time threat detection can assist in detecting threats before they cause substantial damage.
- It allows you to monitor end devices and network activity around the clock and keep logs.
- It speeds up and improves the threat lookup function by utilizing the Memcached server.
- It provides an alarm mechanism to the system administrator or user.
- Real-time threat detection allows enterprises to gain a contextual understanding of the threat landscape and bring changes to their defences accordingly.
- Implementing real-time threat detection can assist in lowering the risk of data breaches, financial loss, and reputational harm.
- Real-time threat detection can help organizations improve their incident response capabilities, allowing them to respond to threats more swiftly and efficiently.

5.3 Limitations

- This system needs the use of qualified professionals to configure and manage it properly.
- While MISP offers comprehensive threat intelligence, it can only detect known threats in the MISP server's database and may not cover all threats or attacks.
- Real-time threat detection systems can be sophisticated and difficult to configure, necessitating extensive technical expertise.
- Real-time threat detection might result in false positives, which can cause alarm fatigue and diminish system efficacy.
- For managing big amounts of logs, this system required more hardware resources such as RAM, storage, and CPU.

APPENDIX G : FUTURE WORK

6. References

Computer Hope, 2022. *Microsoft Windows | History, Versions, & Facts | Britannica*. [Online] Available at: <https://www.computerhope.com/jargon/w/windows.htm> [Accessed 27 Novemeber 2022].

Digité, 2022. *What Is Scrum Methodology? & Scrum Project Management*. [Online] Available at: <https://www.digité.com/agile/scrum-methodology/> [Accessed 23 November 2022].

Elasticsearch B.V., 2022. *Logstash: Collect, Parse, Transform Logs | Elastic*. [Online] Available at: <https://www.elastic.co/logstash/> [Accessed 27 November 2022].

Elasticsearch B.V., 2022. *What is Elasticsearch? | Elastic*. [Online] Available at: <https://www.elastic.co/what-is/elasticsearch> [Accessed 27 November 2022].

Elasticsearch B.V., 2023. *Winlogbeat Overview | Winlogbeat Reference [8.7] | Elastic*. [Online] Available at: [https://www.elastic.co/guide/en/beats/winlogbeat/current/_winlogbeat_overview.html#:~:text=Winlogbeat%20ships%20Windows%20event%20logs.outputs%20\(Elasticsearch%20or%20Logstash\)](https://www.elastic.co/guide/en/beats/winlogbeat/current/_winlogbeat_overview.html#:~:text=Winlogbeat%20ships%20Windows%20event%20logs.outputs%20(Elasticsearch%20or%20Logstash)). [Accessed 12 January 2023].

Harappa Learning Private Limited, 2021. *Advantages & Disadvantages Of The Waterfall Model - Harappa*. [Online] Available at: <https://harappa.education/harappa-diaries/advantages-and-disadvantages-of-waterfall-model/#:~:text=Perhaps%20one%20of%20the%20biggest,project%20can%20easily%20get%20muddled>. [Accessed 27 November 2022].

Planview, 2022. *Introduction to Kanban Guide – What is Kanban? | Planview*. [Online] Available at: <https://www.planview.com/resources/guide/introduction-to-kanban/> [Accessed 27 November 2022].

Python Software Foundation, 2022. *What is Python? Executive Summary | Python.org*. [Online] Available at: <https://www.python.org/doc/essays/blurb/> [Accessed 27 November 2022].

Statista, 2022. *Financial losses due to cyber attacks in the U.S. 2022 | Statista*. [Online] Available at: <https://www.statista.com/statistics/1334399/us-common-results-of-cyber-attacks/> [Accessed 26 November 2022].

Adobe Communications Team, 2022. *Waterfall Methodology: Project Management | Adobe Workfront*. [Online] Available at: <https://business.adobe.com/blog/basics/waterfall> [Accessed 23 November 2022].

Ajmal, S., 2022. *Pros & Cons of Scrum Methodology | QuickStart*. [Online] Available at: <https://www.quickstart.com/blog/pros-and-cons-of-scrum-methodology/> [Accessed 23 November 2022].

Anon., n.d. [Online] Available at: <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban>

Chandana, 2022. *Scrum Project Management: Advantages and Disadvantages [Updated]*. [Online] Available at: <https://www.simplilearn.com/scrum-project-management-article> [Accessed 27 Novemeber 2022].

Debian, 2022. *Debian -- Reasons to use Debian*. [Online] Available at: https://www.debian.org/intro/why_debian [Accessed 27 November 2022].

Dutta, B., 2021. *Waterfall Methodology: Working, Advantages & Disadvantages | Analytics Steps*. [Online] Available at: <https://www.analyticssteps.com/blogs/waterfall-methodology-working-advantages-disadvantages> [Accessed 23 November 2022].

ElastAlert, 2014. *ElastAlert - Easy & Flexible Alerting With Elasticsearch — ElastAlert 0.0.1 documentation*. [Online] Available at: <https://elastalert.readthedocs.io/en/latest/elastalert.html#overview> [Accessed 10 January 2023].

Elasticsearch B.V., 2022. *What is Kibana? | Elastic*. [Online] Available at: <https://www.elastic.co/what-is/kibana> [Accessed 27 November 2022].

Elasticsearch B.V., 2023. *Auditbeat: Lightweight Shipper for Audit Data | Elastic*. [Online] Available at: <https://www.elastic.co/beats/auditbeat> [Accessed 12 January 2023].

Elasticsearch B.V., 2023. *Filebeat: Lightweight Log Analysis & Elasticsearch | Elastic*. [Online] Available at: <https://www.elastic.co/beats/filebeat> [Accessed 12 January 2023].

Elasticsearch B.V., 2023. *The ELK Stack: From the Creators of Elasticsearch | Elastic*. [Online] Available at: <https://www.elastic.co/what-is/elk-stack> [Accessed 10 January 2023].

Frye, M.-K., 2023. *What is an API? (Application Programming Interface) | MuleSoft*. [Online] Available at: <https://www.mulesoft.com/resources/api/what-is-an-api> [Accessed 12 January 2023].

geeksforgeeks.org, 2022. *Software Engineering | Prototyping Model - GeeksforGeeks*. [Online] Available at: <https://www.geeksforgeeks.org/software-engineering-prototyping-model/> [Accessed 23 November 2022].

geeksforgeeks, 2023. *Intrusion Detection System (IDS) - GeeksforGeeks*. [Online] Available at: <https://www.geeksforgeeks.org/intrusion-detection-system-ids/> [Accessed 12 January 2023].

gov.uk, 2022. *National Cyber Strategy 2022 (HTML) - GOV.UK*. [Online] Available at: <https://www.gov.uk/government/publications/national-cyber-strategy-2022/national-cyber-security-strategy-2022#introduction> [Accessed 26 November 2022].

Hamad AL-Mohannadi, I. A. J. A. H. A. C., 2018. *Cyber Threat Intelligence from Honeypot Data Using Elasticsearch*. Krakow, Poland, IEEE.

Indeed Editorial Team, 2021. *List of Scrum Advantages and Disadvantages | Indeed.com*. [Online] Available at: <https://www.indeed.com/career-advice/career-development/disadvantages-of-scrum> [Accessed 23 November 2022].

intland software, 2021. *Why Upgrade from Waterfall to Evolutionary Development (EVO)*. [Online] Available at: <https://content.intland.com/blog/why-upgrade-from-waterfall-to-evolutionary-development-evo> [Accessed 27 November 2022].

Javed, R., 2022. *Kanban - definition, explanation, advantages and disadvantages | Accounting For Management*. [Online] Available at: <https://www.accountingformanagement.org/kanban/> [Accessed 23 November 2022].

Kanbanize, 2022. *What Is Kanban? Explained in 10 Minutes | Kanbanize*. [Online] Available at: <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban> [Accessed 23 November 2022].

Kaspersky, 2023. *Threat Intelligence Definition | Cyber Threat Intelligence*. [Online] Available at: <https://www.kaspersky.com/resource-center/definitions/threat-intelligence> [Accessed 12 January 2023].

kissflow.com, 2022. *What is Kanban Methodology | Introduction to Kanban Framework*. [Online] Available at: <https://kissflow.com/project/agile/kanban-methodology/> [Accessed 26 November 2022].

Kris, 2018. *Difference Between Evolutionary Prototyping and Throw-away Prototyping - prototypeinfo.com*. [Online] Available at: <https://prototypeinfo.com/evolutionary-prototyping-and-throw-away-prototyping/> [Accessed 26 November 2022].

LAL, R., 2022. *What Is Software Prototyping And Why Do You Need It?*. [Online] Available at: <https://stackify.com/what-is-software-prototyping-and-why-do-you-need-it/> [Accessed 23 November 2022].

Lutkevich, B., 2023. *What is Malware? Definition, Types, Prevention - TechTarget*. [Online] Available at: <https://www.techtarget.com/searchsecurity/definition/malware#:~:text=Malware%2C%20or%20malicious%20software%2C%20is,Trojan%20horses%2C%20ransomware%20and%20spyware>. [Accessed 11 January 2023].

Mezmo Inc, 2022. *Logging Agents vs. Logging Libraries: Which Should You Use? | Mezmo*. [Online] Available at: <https://www.mezmo.com/blog/logging-agents-vs-logging-libraries-which-should-you-use> [Accessed 12 January 2023].

MISP project, 2023. *MISP features and functionalities*. [Online] Available at: <https://www.misp-project.org/features/> [Accessed 2 January 2023].

misp-project.org, 2022. *MISP Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing*. [Online] Available at: <https://www.misp-project.org/> [Accessed 27 November 2022].

O'Neill, R., 2019. *CERT NZ reports record quarterly losses from cyber attacks - Reseller News*. [Online] Available at: <https://www.reseller.co.nz/article/666038/cert-nz-reports-record-quarterly-losses-from-cyber->

attacks/

[Accessed 26 November 2022].

Passeri, P., 2022. Q1 2022 Cyber Attacks Statistics – HACKMAGEDDON. [Online]
Available at: <https://www.hackmageddon.com/2022/04/19/q1-2022-cyber-attacks-statistics/>
[Accessed 26 November 2022].

Prasanna, 2022. Waterfall Model Advantages and Disadvantages | What is Waterfall Model? Advantages and Disadvantages of Waterfall Model - A Plus Topper. [Online]
Available at: <https://www.aplustopper.com/waterfall-model-advantages-and-disadvantages/>
[Accessed 23 November 2022].

Rock Content, 2021. Real-time threat detection: why this is the future of cybersecurity. [Online]
Available at: <https://rockcontent.com/blog/real-time-threat-detection/>
[Accessed 12 January 2023].

scrum.org, 2022. What is Scrum? | Scrum.org. [Online]
Available at: <https://www.scrum.org/resources/what-is-scrum>
[Accessed 26 November 2022].

SecurityScorecard, 2020. Message from Security Scorecard. [Online]
Available at: <https://securityscorecard.com/blog/what-is-cyber-threat-intelligence-3-types-and-examples/>
[Accessed 12 January 2023].

Sharif, A., 2023. Log Files: Definition, Types, and Importance | CrowdStrike. [Online]
Available at: <https://www.crowdstrike.com/cybersecurity-101/observability/log-file#:~:text=A%20log%20file%20is%20an,as%20transactions%2C%20errors%20and%20intrusions.>
[Accessed 12 January 2023].

Sharif, A., 2023. What is an Event Log? Contents and Use | CrowdStrike. [Online]
Available at: <https://www.crowdstrike.com/cybersecurity-101/observability/event-log/>
[Accessed 12 January 2023].

simplilearn, 2023. Learning Consultant says.... [Online]
Available at: <https://www.simplilearn.com/tutorials/cyber-security-tutorial/what-is-firewall#:~:text=Types%20of%20Firewalls,the%20gateway%20and%20your%20network.>
[Accessed 11 January 2023].

Sornalakshmi.K, 2017. Detection of DoS attack and Zero Day Threat with. Madurai, India, IEEE.

Telegram, 2022. Telegram FAQ. [Online]
Available at: <https://telegram.org/faq#q-what-is-telegram-what-do-i-do-here>
[Accessed 27 November 2022].

Telegram, 2023. Telegram Messenger. [Online]
Available at: <https://telegram.org/>
[Accessed 12 January 2023].

Tim LaueA, T. K. C. K. K.-O. D., 2022. A SIEM Architecture for Advanced Anomaly Detection. 1(1), pp. 26-42.

tutorialspoint, 2022. Memcached Tutorial. [Online]
Available at: <https://www.tutorialspoint.com/memcached/index.htm>
[Accessed 27 November 2022].

tutorialspoint, 2022. *Network Security – Overview*. [Online]

Available at: https://www.tutorialspoint.com/network_security/network_security_overview.htm
[Accessed 26 November 2022].

VMware Inc, 2022. *What is VMware Workstation | FAQ | LATAM*. [Online]

Available at: <https://www.vmware.com/latam/products/workstation-pro/faq.html>
[Accessed 27 November 2022].

WisdomPlexus, 2022. *What are the Pros and Cons of Kanban? | WisdomPlexus*. [Online]

Available at: <https://wisdomplexus.com/blogs/pros-cons-kanban/>
[Accessed 23 November 2022].

7.Bibliography

R. Stoleriu, A. Puncioiu and I. Bica, "Cyber Attacks Detection Using Open Source ELK Stack," 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 2021, pp. 1-6, doi: 10.1109/ECAI52376.2021.9515120.

B. AlSabbagh and S. Kowalski, "A Framework and Prototype for A Socio-Technical Security Information and Event Management System (ST-SIEM)," 2016 European Intelligence and Security Informatics Conference (EISIC), Uppsala, Sweden, 2016, pp. 192-195, doi: 10.1109/EISIC.2016.049.

A. Serckumecka, I. Medeiros and A. Bessani, "Low-Cost Serverless SIEM in the Cloud," 2019 38th Symposium on Reliable Distributed Systems (SRDS), Lyon, France, 2019, pp. 381-3811, doi: 10.1109/SRDS47363.2019.00057.

M. Cinque, D. Cotroneo and A. Pecchia, "Challenges and Directions in Security Information and Event Management (SIEM)," 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Memphis, TN, USA, 2018, pp. 95-99, doi: 10.1109/ISSREW.2018.00-24.

W. Qingrong Jason Wu, X. Zhu Sherry Zhu, K. -C. Kuo Eric Guo and C. Lu Max Lu, "Light SIEM for semiconductor industry," 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 2017, pp. 2331-2335, doi: 10.1109/IEEM.2017.8290308.

M. Schölzel, E. Eren and K. -O. Detken, "A viable SIEM approach for Android," 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Warsaw, Poland, 2015, pp. 803-807, doi: 10.1109/IDAACS.2015.7341414.

K. Sornalakshmi, "Detection of DoS attack and zero day threat with SIEM," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2017, pp. 1-7, doi: 10.1109/ICCONS.2017.8250515.

S. S. Sekharan and K. Kandasamy, "Profiling SIEM tools and correlation engines for security analytics," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2017, pp. 717-721, doi: 10.1109/WiSPNET.2017.8299855.

M. Hristov, M. Nenova, G. Iliev and D. Avresky, "Integration of Splunk Enterprise SIEM for DDoS Attack Detection in IoT," 2021 IEEE 20th International Symposium on Network Computing and Applications (NCA), Boston, MA, USA, 2021, pp. 1-5, doi: 10.1109/NCA53618.2021.9685977.

J. -H. Lee, Y. S. Kim, J. H. Kim and I. K. Kim, "Toward the SIEM architecture for cloud-based security services," 2017 IEEE Conference on Communications and Network Security (CNS), Las Vegas, NV, USA, 2017, pp. 398-399, doi: 10.1109/CNS.2017.8228696.

CHAPTER 8: APPENDIX

8.1. Appendix A: PRE-SURVEY

8.1.1. Pre-Survey Form

Full Name	Short answer text
Email Address	Short answer text
1.Which version of windows operating system do you use often?	
<input type="radio"/> Windows 11	
<input type="radio"/> Windows 10	
<input type="radio"/> Windows 8	
<input type="radio"/> Windows 7	
2.How often do you update windows OS?	
<input type="radio"/> Everyday	
<input type="radio"/> Sometimes	
<input type="radio"/> Never	

Figure 123: Pre-Survey form 1.

3.Which web browser do you use often?

- Chrome
- Microsoft edge
- Firefox
- Safari
- Other

4.Which anti-virus software do you use for pc?

- Windows Defender
- Norton
- Kaspersky
- Bitdefender
- McAfee
- Other

Figure 124: Pre-Survey form 2.

5.How often do you update your anti-virus software?

- Everyday
- Sometimes
- Never

6.Which types of cyberattacks you have heard most?

- Malware
- Phishing
- Distributed Denial-of-Service (DDoS)
- Man-in-the-middle attack
- Zero-day exploit
- SQL injection

7.Do you ever have experienced any type of cyber-attack?

- Yes
- No

Figure 125: Pre-Survey form 3.

8. Do you know about malware attacks?

- Yes
- No

9. What type of malware attack do you have heard most?

- Ransomware
- Viruses
- Trojans
- Adware
- Spyware
- Worms
- Rootkit
- Other

10. Have you heard about SIEM security solutions?

- Yes
- No

Figure 126: Pre-Survey form 4.

11.Do you think real-time monitoring and threat detection is important for System security?

- Extremely important
- Very important
- Moderately important
- Not very important

12.Have you heard about the Cyber Threat Intelligence Sharing Platform ?

- Yes
- No
- Add option or [add "Other"](#)



Multiple choice



Required



13.Do you think such a Cyber Threat Intelligence Sharing Platform can help in the identification of the threat effectively?

- Yes, very much
- Yes, but not very much

Figure 127: Pre-Survey form 5.

14. Do you think with the integration of Cyber Threat Intelligence Sharing Platform in SIEM solution can help in better monitoring and threat detection?

- Yes
- No
- Depends upon types of attacks

15. What do you think about the importance of Cyber security in your daily life?

- Extremely important
- Very important
- Moderately important
- Not very important

Any feedback or suggestions if you have ?

Long answer text

Figure 128: Pre-Survey form 6.

8.1.2 Filled Pre-Survey Sample

The screenshot shows a survey form with three sections. The first section contains two text input fields: 'Hari Rai' and 'Klejan BC'. The second section contains two email input fields: 'hari@gmail.com' and 'abjc12335@gmail.com'. The third section is a question with four options for Windows versions, where 'Windows 10' is selected.

Hari Rai
Klejan BC

hari@gmail.com
abjc12335@gmail.com

1.Which version of windows operating system do you use often?

Windows 11
 Windows 10
 Windows 8
 Windows 7

Figure 129: Filled Pre-Survey Sample 1

2. How often do you update windows OS?

- Everyday
- Sometimes
- Never

3. Which web browser do you use often?

- Chrome
- Microsoft edge
- Firefox
- Safari
- Other

4. Which anti-virus software do you use for pc?

- Windows Defender
- Norton
- Kaspersky
- Bitdefender
- McAfee
- Other

Figure 130: Filled Pre-Survey Sample 2.

5. How often do you update your anti-virus software?

- Everyday
- Sometimes
- Never

6. Which types of cyberattacks you have heard most?

- Malware
- Phishing
- Distributed Denial-of-Service (DDoS)
- Man-in-the-middle attack
- Zero-day exploit
- SQL injection

7. Do you ever have experienced any type of cyber-attack?

- Yes
- No

Figure 131: Filled Pre-Survey Sample 3.

8. Do you know about malware attacks?

- Yes
- No

9. What type of malware attack do you have heard most?

- Ransomware
- Viruses
- Trojans
- Adware
- Spyware
- Worms
- Rootkit
- Other

10. Have you heard about SIEM security solutions?

- Yes
- No

Figure 132: Filled Pre-Survey Sample 4.

11.Do you think real-time monitoring and threat detection is important for System security?

- Extremely important
- Very important
- Moderately important
- Not very important

12.Have you heard about the Cyber Threat Intelligence Sharing Platform ?

- Yes
- No

13.Do you think such a Cyber Threat Intelligence Sharing Platform can help in the identify of the threat effectively?

- Yes, very much
- Yes, but not very much
- No, it does not

Figure 133: Filled Pre-Survey Sample 5.

14. Do you think with the integration of Cyber Threat Intelligence Sharing Platform in SIEM solution can help in better monitoring and threat detection?

- Yes
- No
- Depends upon types of attacks

15. What do you think about the importance of Cyber security in your daily life?

- Extremely important
- Very important
- Moderately important
- Not very important

Any feedback or suggestions if you have ?

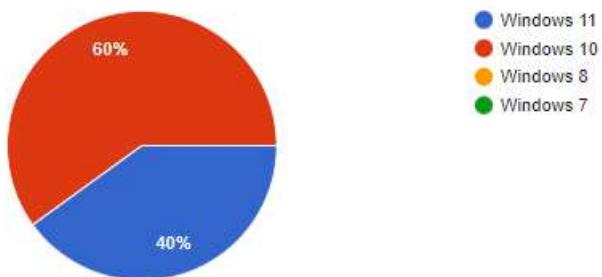
Figure 134: Filled Pre-Survey Sample 6.

8.1.3. Pre-Survey Result

1.Which version of windows operating system do you use often?

 Copy

15 responses



2.How often do you update windows OS?

 Copy

15 responses

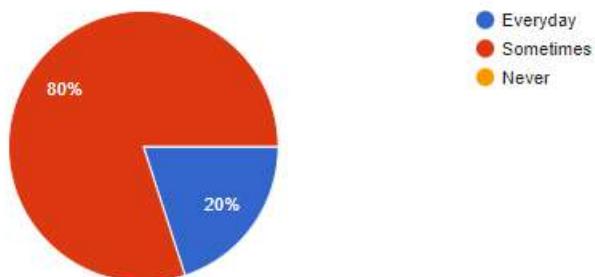


Figure 135: Pre-survey Result 1.

3.Which web browser do you use often?

 Copy

15 responses

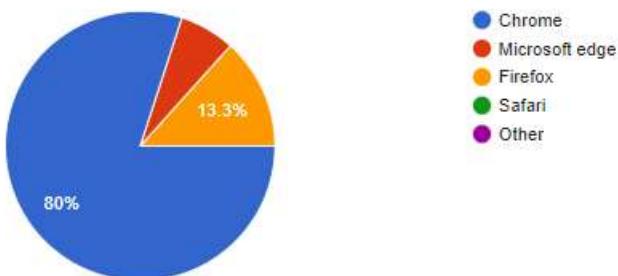
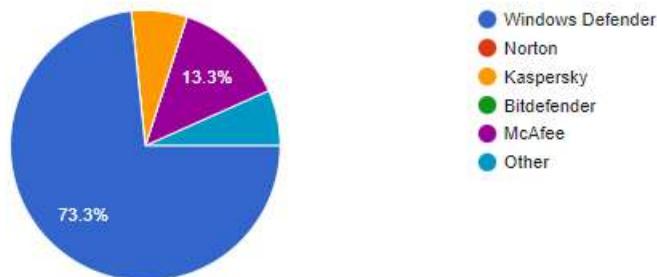


Figure 136: Pre-survey Result 2.

4.Which anti-virus software do you use for pc?

 Copy

15 responses

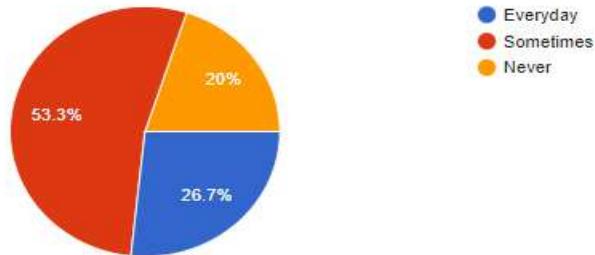


- Windows Defender
- Norton
- Kaspersky
- Bitdefender
- McAfee
- Other

5.How often do you update your anti-virus software?

 Copy

15 responses



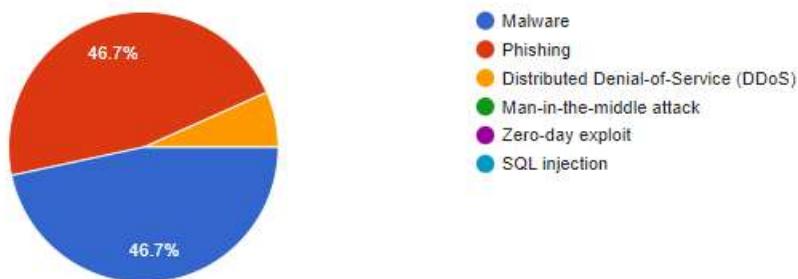
- Everyday
- Sometimes
- Never

Figure 137: Pre-survey Result 3.

6.Which types of cyberattacks you have heard most?

 Copy

15 responses



- Malware
- Phishing
- Distributed Denial-of-Service (DDoS)
- Man-in-the-middle attack
- Zero-day exploit
- SQL injection

Figure 138: Pre-survey Result 4.

7.Do you ever have experienced any type of cyber-attack?

 Copy

15 responses

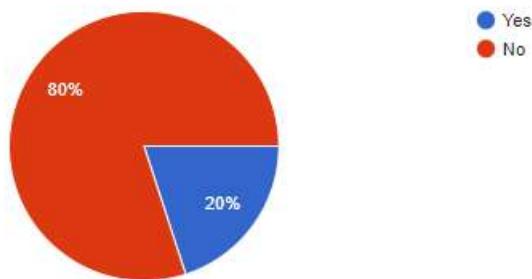


Figure 139: Pre-survey Result 5.

8.Do you know about malware attacks?

 Copy

15 responses

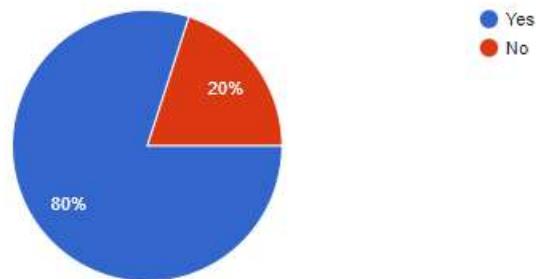


Figure 140: Pre-survey Result 6.

9.What type of malware attack do you have heard most?

15 responses

 Copy

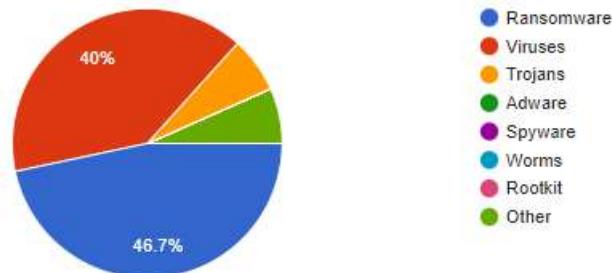


Figure 141: Pre-survey Result 7.

10.Have you heard about SIEM security solutions?

14 responses

 Copy

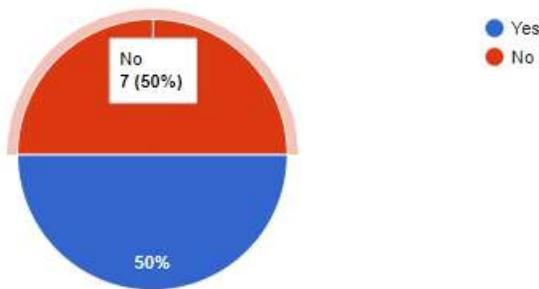


Figure 142: Pre-survey Result 8.

11.Do you think real-time monitoring and threat detection is important for System security?

Copy

15 responses

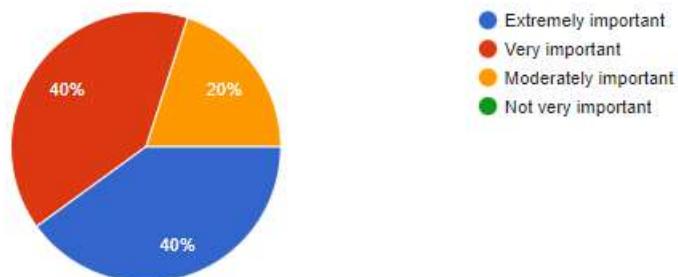


Figure 143: Pre-survey Result 9.

12.Have you heard about the Cyber Threat Intelligence Sharing Platform ?

Copy

15 responses

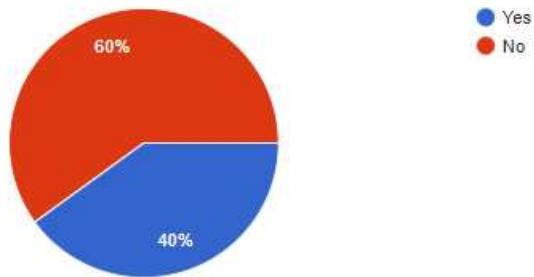


Figure 144: Pre-survey Result 10.

13. Do you think such a Cyber Threat Intelligence Sharing Platform can help in the identity of the threat effectively?

 Copy

15 responses

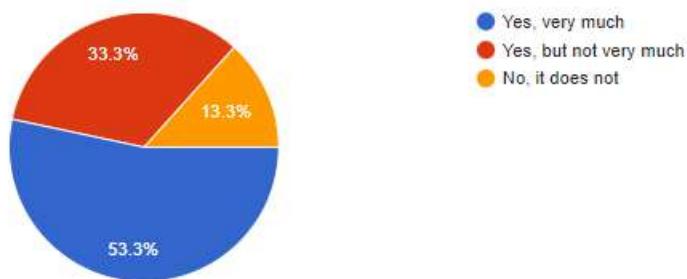


Figure 145: Pre-survey Result 11.

14. Do you think with the integration of Cyber Threat Intelligence Sharing Platform in SIEM solution can help in better monitoring and threat detection?

 Copy

15 responses

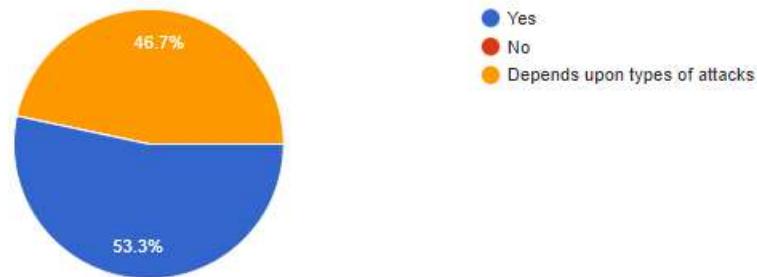


Figure 146: Pre-survey Result 12.

15. What do you think about the importance of Cyber security in your daily life?

 Copy

15 responses



Figure 147: Pre-survey Result 13.

Any feedback or suggestions if you have ?

5 responses

:)

Best of luck 

interested , Goodluck

Good work

Good job

Figure 148: Pre-survey Result 14.

[GO TO CHAPTER 3](#)

8.2 APPENDIX B: POST SURVEY

8.2.1 Post-Survey Form

Post-Survery question

Real Time Threat Detection System with Threat Intelligence is security solution for any organization which monitors real time traffic of data flow, changes in systems, malicious activities, and behavior's of different types of systems and devices used by an organization. This purpose system will collect logs from different network devices, end devices, IDS system etc and store in centralized storage where those logs will be analyzed and compared with help of a Threat Intel Platform for known threat and rule based detection will be used for unknown threat to produce an alert when any malicious or intrusion activities are detected.

This survey is conducted to determine how significant SIEM security solution are important in everyday life in order to detect threats in system or network to improve security. Additionally, this survey encourages me to develop more project features.

This is a Post survey for Final Year Project. Your feedback will be extremely welcomed and helpful in completing my project. All of the responses will remain private.

Email *

Short-answer text

Figure 149: Post survey question form 1.

1. How effective was the real-time threat detection system in identifying potential threats?

- Effective
- Less Effective
- More Effective

2. Which feature do you think is the best in this project ?

- MISP lookup
- Visualization
- File integrity check
- Telegram alert

Figure 150: Post survey question form 2.

3. Were there any false positives or false negatives in the system's detection of threats?

- Many
- Less
- No

4. How satisfied were you with the quality of the threat intelligence used by the system?

- Very satisfied
- Just ok
- Not satisfied

Figure 151: Post survey question form 3.

5. How quickly were threats identified and responded to by the system?

- Fast
- Medium
- Low

6. How user-friendly was the interface for monitoring and analyzing threats?

- Yes
- No

Figure 152: Post survey question form 5.

9. Were there any technical issues or challenges in implementing or using the system?

- Yes
- No

10. How helpful were the system's reports and analytics in identifying trends and improving overall security?

- More
- Medium
- Less

Figure 153: Post survey question form 6.

11. Would you recommend this real-time threat detection system to others in your industry or organization?

Yes

No

Figure 154: Post survey question form 7.

8.2.2 Filled Post-Survey Sample

Real Time Threat Detection System with Threat Intelligence is security solution for any organization which monitors real time traffic of data flow, changes in systems, malicious activities, and behavior's of different types of systems and devices used by an organization. This purpose system will collect logs from different network devices, end devices, IDS system etc and store in centralized storage where those logs will be analyzed and compared with help of a Threat Intel Platform for known threat and rule based detection will be used for unknown threat to produce an alert when any malicious or intrusion activities are detected.

This survey is conducted to determine how significant SIEM security solution are important in everyday life in order to detect threats in system or network to improve security. Additionally, this survey encourages me to develop more project features.

This is a Post survey for Final Year Project. Your feedback will be extremely welcomed and helpful in completing my project. All of the responses will remain private.

* Indicates required question

Email *

rabinjoshi97@gmail.com

Figure 155: Filled Form of Post survey 1.

1. How effective was the real-time threat detection system in identifying potential threats?

- Effective
- Less Effective
- More Effective

2. Which feature do you think is the best in this project ?

- MISP lookup
- Visualization
- File integrity check
- Telegram alert

Figure 156: Filled Form of Post survey 2.

3. Were there any false positives or false negatives in the system's detection of threats?

- Many
- Less
- No

4. How satisfied were you with the quality of the threat intelligence used by the system?

- Very satisfied
- Just ok
- No satisfied

Figure 157: Filled Form of Post survey 3.

5. How quickly were threats identified and responded to by the system?

- Fast
- Medium
- Low

6. How user-friendly was the interface for monitoring and analyzing threats?

- Yes
- No

Figure 158: Filled Form of Post survey 4.

7. Did the system provide sufficient context and information about identified threats?

- Yes
- No

8. How well did the system integrate with your existing security infrastructure?

- Fully
- Partially
- Not well

Figure 159: Filled Form of Post survey 5.

9. Were there any technical issues or challenges in implementing or using the system?

Yes

No

10. How helpful were the system's reports and analytics in identifying trends and improving overall security?

More

Medium

Less

Figure 160: Filled Form of Post survey 6.

11. Would you recommend this real-time threat detection system to others in your industry or organization?

Yes

No

Figure 161: Filled Form of Post survey 7.

8.2.3 Post-Survey Result

1. How effective was the real-time threat detection system in identifying potential threats?

 Copy

20 responses

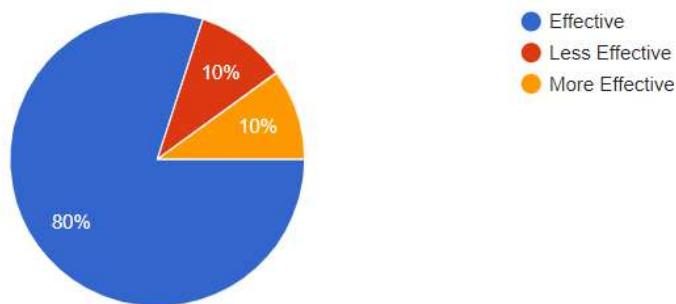


Figure 162: Post survey Result 1.

2. Which feature do you think is the best in this project ?

 Copy

20 responses

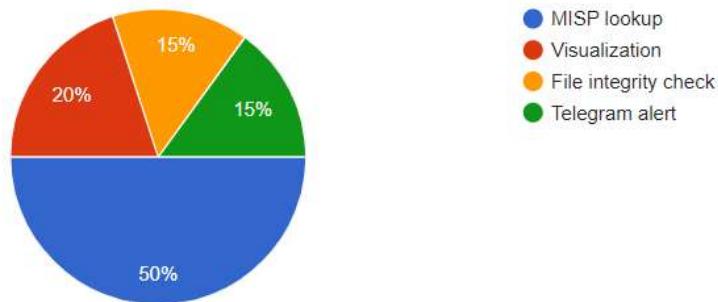


Figure 163: Post survey Result 2.

3. Were there any false positives or false negatives in the system's detection of threats?

 Copy

20 responses

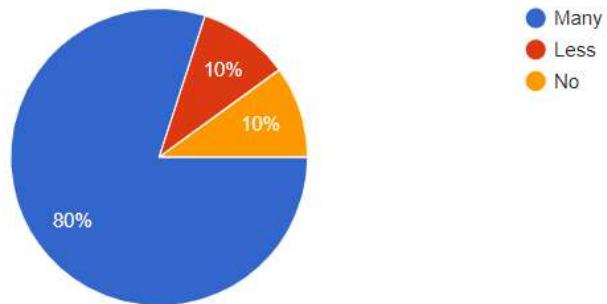


Figure 164: Post survey Result 3.

4. How satisfied were you with the quality of the threat intelligence used by the system?

 Copy

20 responses

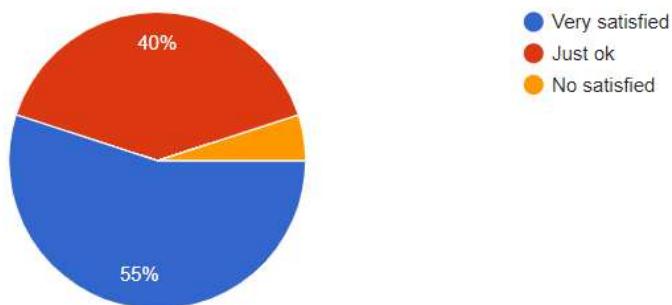


Figure 165: Post survey Result 4.

5. How quickly were threats identified and responded to by the system?

Copy

20 responses

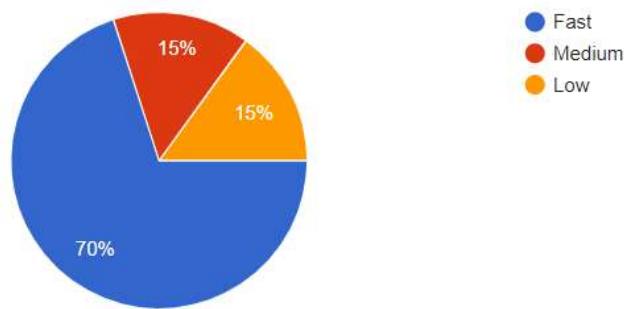


Figure 166: Post survey Result 5.

6. How user-friendly was the interface for monitoring and analyzing threats?

Copy

18 responses

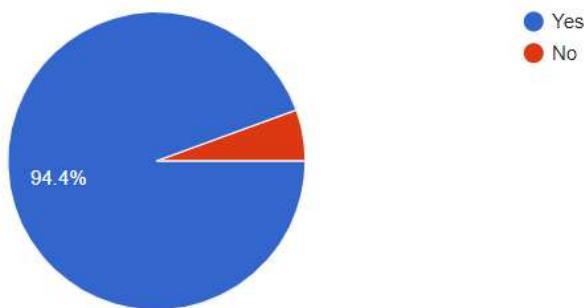


Figure 167:Post survey Result 6

7. Did the system provide sufficient context and information about identified threats?

Copy

20 responses

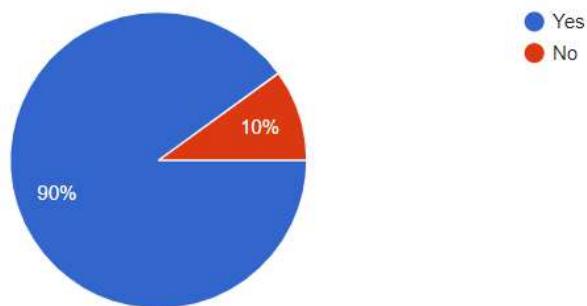


Figure 168: Post survey Result 7.

8. How well did the system integrate with your existing security infrastructure?

Copy

20 responses

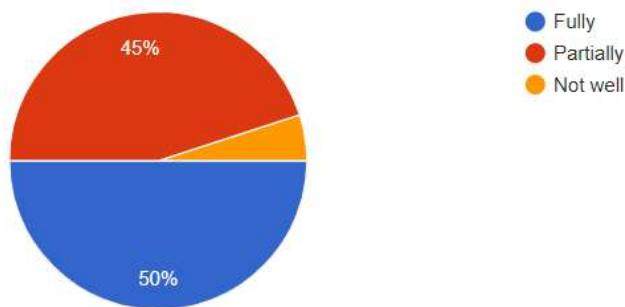


Figure 169: Post survey Result 8.

9. Were there any technical issues or challenges in implementing or using the system?

Copy

19 responses

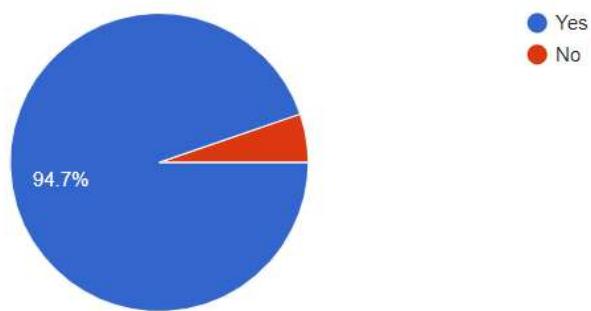


Figure 170: Post survey Result 9.

10. How helpful were the system's reports and analytics in identifying trends and improving overall security?

Copy

20 responses

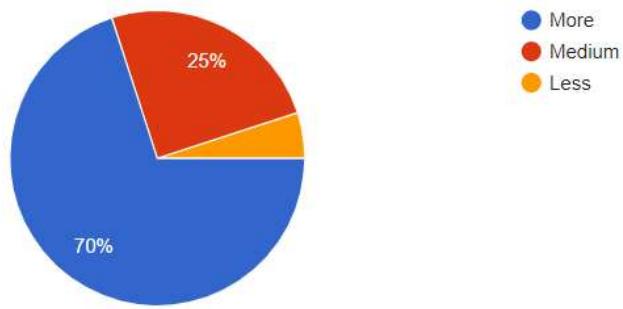


Figure 171: Post survey Result 10.

11. Would you recommend this real-time threat detection system to others in your industry or organization?

 Copy

20 responses

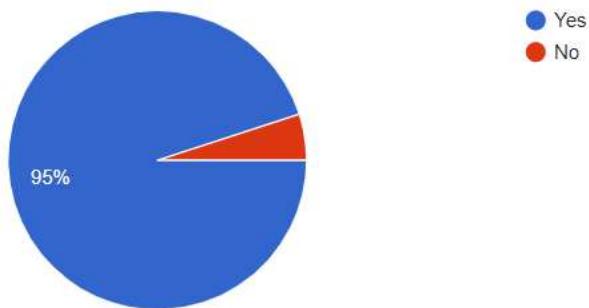


Figure 172: Post survey Result 11.

[GO TO CHAPTER 3](#)

8.3. Appendix C: RESOURCE REQUIREMENT

8.3.1 Software

Software requirement	Description
1. VMware workstation pro (v16.2.4.28481)	VMware Workstation is a desktop hypervisor that allows users to operate virtual machines on top of host Operating System. (VMware Inc, 2022) It is used to run ELK stack, windows, Misp server, Memcached server virtually on host OS.
2. Linux Debian 10 (64 bit)	Linux Debian 10 is a Linux-based operating system that may be used on a variety of devices such as laptops, desktop computers, and servers. (Debian, 2022) It is used to as end devices, run ELK stack, Misp server, Memcached server.
3. Elasticsearch (v7.17)	Elasticsearch is a decentralized, free, and open search and analytic engine that can handle all forms of data which include structured, and unstructured textual, quantitative, geographic, data. (Elasticsearch B.V., 2022) It is used as logs storage, search and analytical engine.
4. Logstash (v7.17)	Logstash is an open-source server-side data processing pipeline which collects data from many sources, modifies, filters it, and then sends it to prefer storage. (Elasticsearch B.V., 2022) It is used to collect, filter, enrich all the logs from different sources and forward to elastic search. It uses different filter plugin like ruby, Memcached, grok etc.

5. Kibana (v.7.17)	<p>Kibana is an open-source web portal interface application that runs on top of the Elastic Stack which provides searching, querying and data visualization features for Elasticsearch-indexed data. (Elasticsearch B.V., 2022)</p> <p>It is used for searching, querying and data visualization.</p>
6. Windows 10 (64 bit)	<p>Microsoft Windows, often known as Windows and Windows OS, is a personal computer operating system (OS) developed by Microsoft Corporation which has the graphical user interface (GUI). (Computer Hope, 2022)</p> <p>It is used as End devices and host OS on which VMs will run.</p>
7. Python	<p>python is a programming language that is object oriented, high level and interpreted. (Python Software Foundation, 2022)</p> <p>It is used to pull data from MISP server and push to Memcached server.</p>
8. MISP	<p>MISP is open-source threat intelligence and sharing platform for sharing, storing, correlating, and analysing cyber threat and malicious activity. (misp-project.org, 2022)</p> <p>It is used to feed threat intelligence data to enrich log data in Logstash.</p>
9. Memcached	<p>It is a high-performance, shared memory object caching technology that is open source. (tutorialspoint, 2022)</p> <p>It is used to provide temporary storage for threat intel data and provide fast access of data to Logstash.</p>

10. Telegram	<p>It is a messaging application that concentrates on speed and security which is extremely fast, easy and free that can be used on several devices at the same time.</p> <p>(Telegram, 2022)</p> <p>It is used as platform to receive alerting messages when threats are detected in system.</p>
11. GNS3	It is used in the project to create LAN topology with combination of virtual router, firewall, switch, VMs to depict or stimulate real scenario.
12. Mikrotik Router	It is used to route packet from internal LAN to another network. It also used here to provide internet access to internal network.
13. Cisco Switch	It is used to connect multiples Virtual Machines together to network.
14. PF Sense	It is used as firewall to monitor and filter incoming and outgoing network traffic.
15. Suricata	It is used as Network Intrusion Detection System to detect abnormal activities in the network as well as Host Intrusion Detection System to detect abnormal activities in host machine in this project.
16. ElastAlert	It is used to query with Elasticsearch for triggering alerts and sending messages to telegram.

Table 16: Software requirements.

8.3.2 Hardware

Hardware requirements	Description
Laptop	It is used to run all Virtual Machine, GNS3 which are need in this project and for writing python scripts.

Table 17: Hardware Requirement.

8.4. APPENDIX D: CONSIDER METHODOLOGY FOR DEVELOPMENT

8.4.1. Water Fall Methodology

The Waterfall methodology which is also known as the Waterfall model that is a step wise procedure for development which flows like a waterfall through all phases of a project for example, analysis, design, development, and testing with each phase that will be completely wrapped up before moving on to the next phase. (Adobe Communications Team, 2022)



Figure 173: Water Fall Methodology. (Adobe Communications Team, 2022)

Advantage	Disadvantage
1. This methodology has clear and exact structure, that specifies the function that must be perform at each stage of the project. (Harappa Learning Private Limited, 2021)	1. In this methodology, it might be quite difficult to revert and change anything that wasn't carefully thought out at the design phase, once an application is in the testing phase. (Dutta, 2021)

8.4.2. Scrum

Scrum is an agile framework for software development that is built on iterative and incremental processes which is also adaptable, rapid, flexible, and effective in delivering value to the client throughout the project's development. (Digité, 2022)

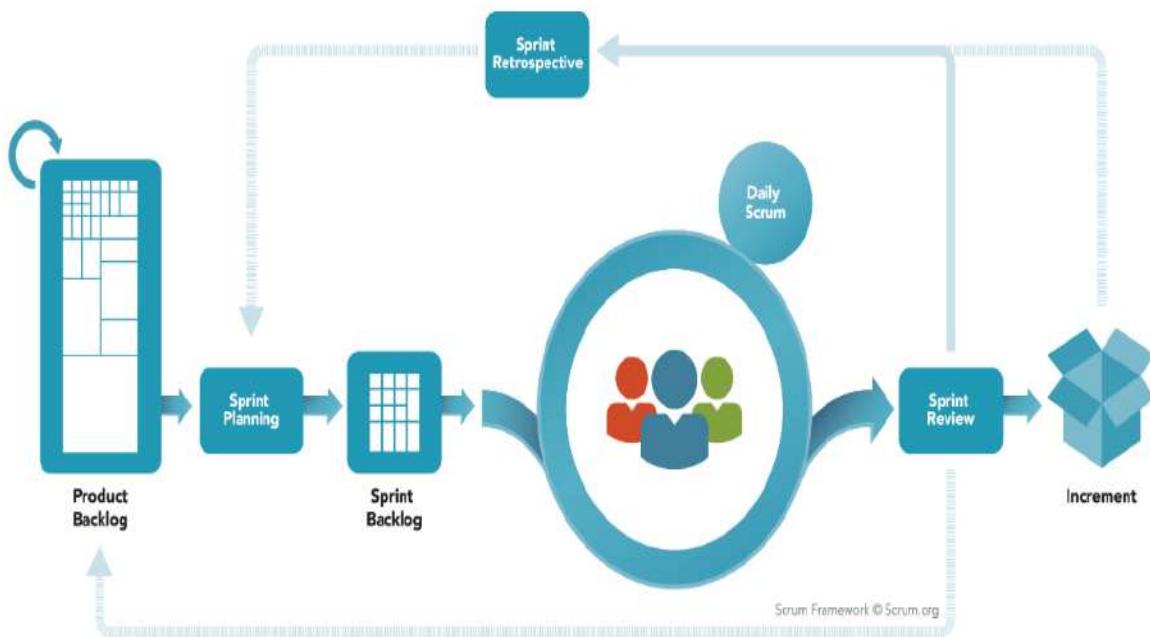


Figure 174: Scrum Methodology. (scrum.org, 2022)

Advantage	Disadvantage
1. This methodology can assist teams in completing project milestones in a timely and effective manner. (Chandana, 2022)	1. To properly implement this methodology a well-trained and skilled manpower is needed. (Indeed Editorial Team, 2021)

8.4.3. Kanban

Kanban is a common Lean workflow management strategy for establishing, managing, and optimizing services which deliver knowledge work that enables to visualize user work, increase productivity, and keep improving. (Kanbanize, 2022)

Kanban project management framework

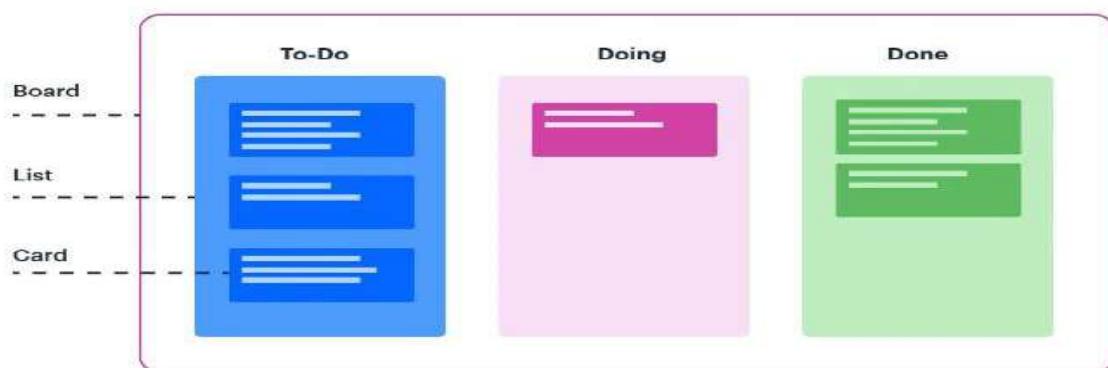


Figure 175: Kanban Methodology. (kissflow.com, 2022)

Advantage	Disadvantage
<p>1. This methodology employs a visual work and process model to manage workflow at the individual, team, and even organizational levels. (Planview, 2022)</p>	<p>1. Kanban will become exceedingly challenging to apply when a system has an excessive number of interconnected tasks or activities. (Javed, 2022)</p>

[GO TO TOP CHAPTER 3](#)

8.5 APPENDIX E: Implementation Screenshot of the System

8.5.1 VMware Workstation 16 Pro

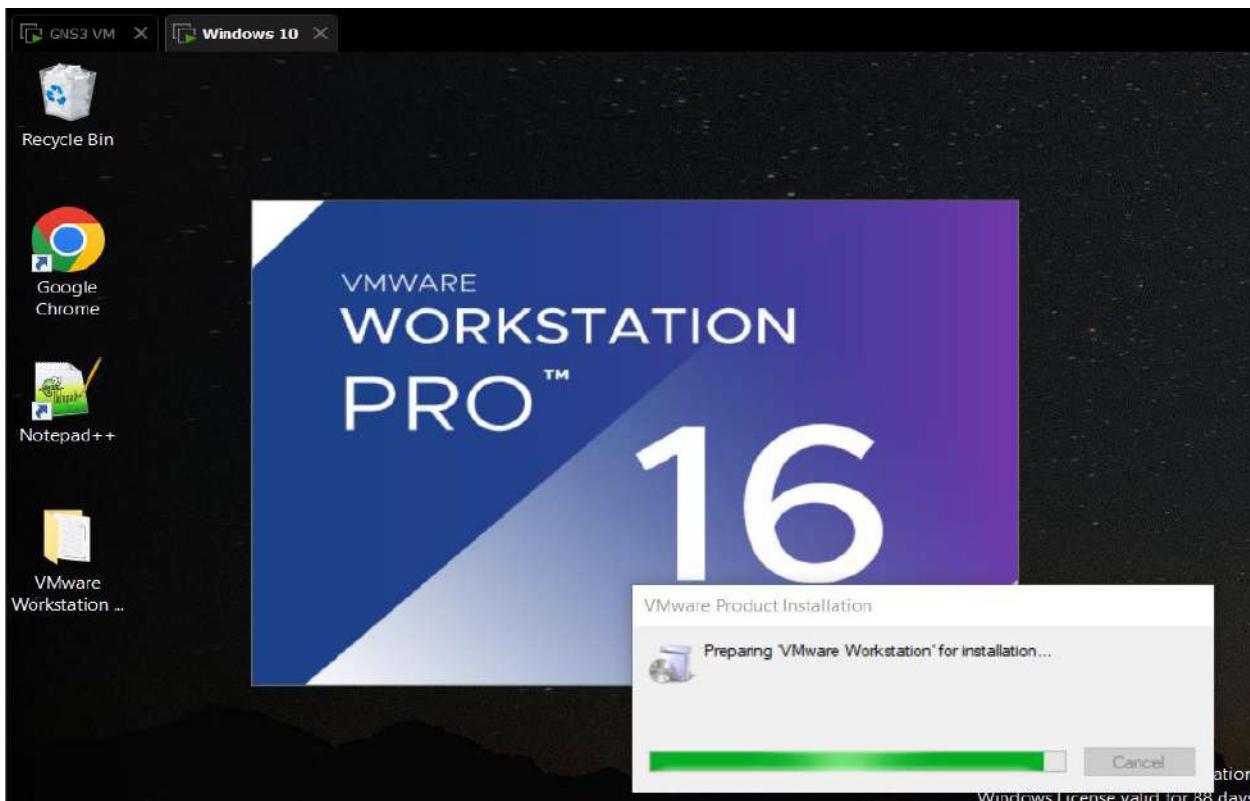


Figure 176: Installation of VM workstation 1.

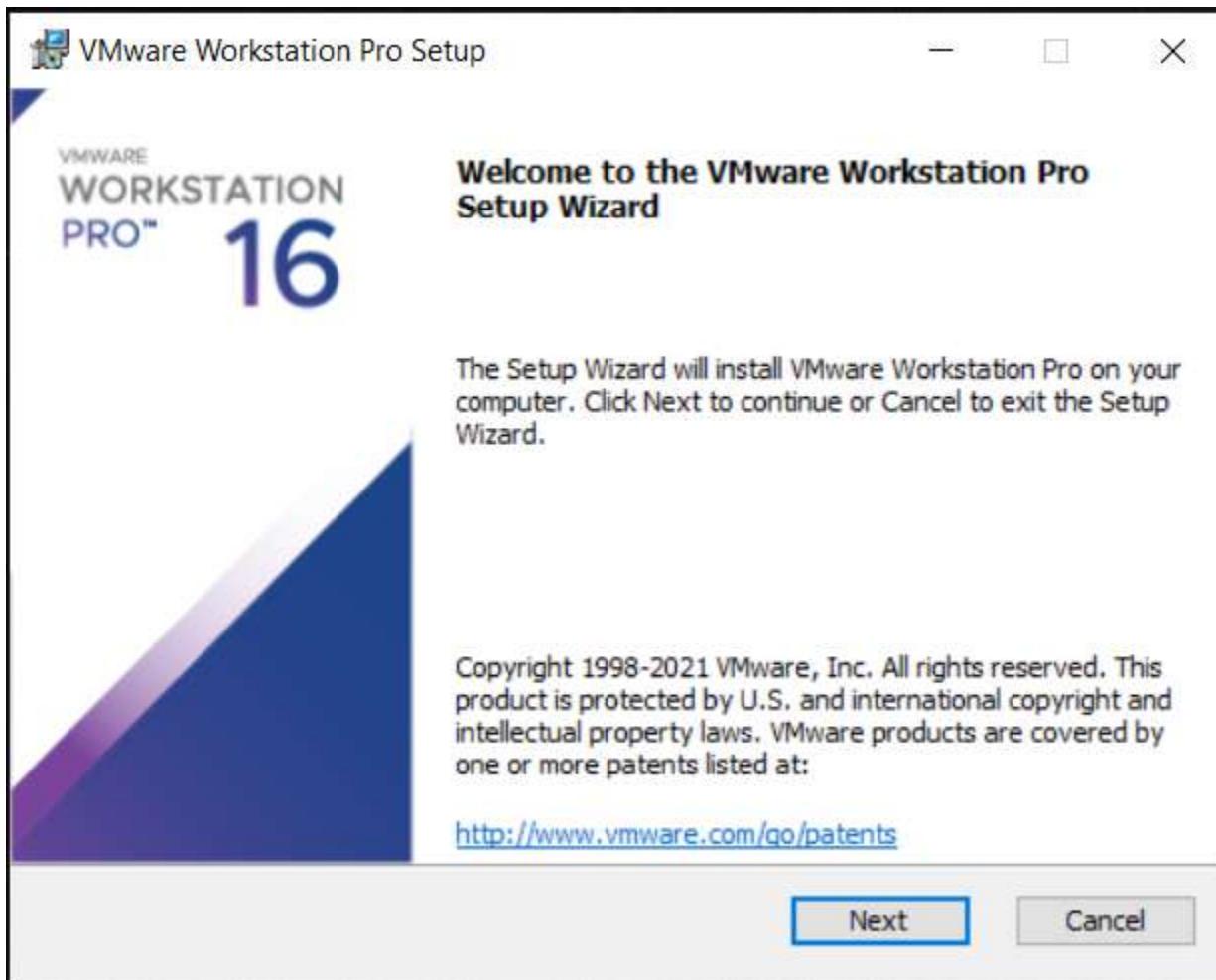


Figure 177: Installation of VM workstation 2.



Figure 178: Installation of VM workstation 3.

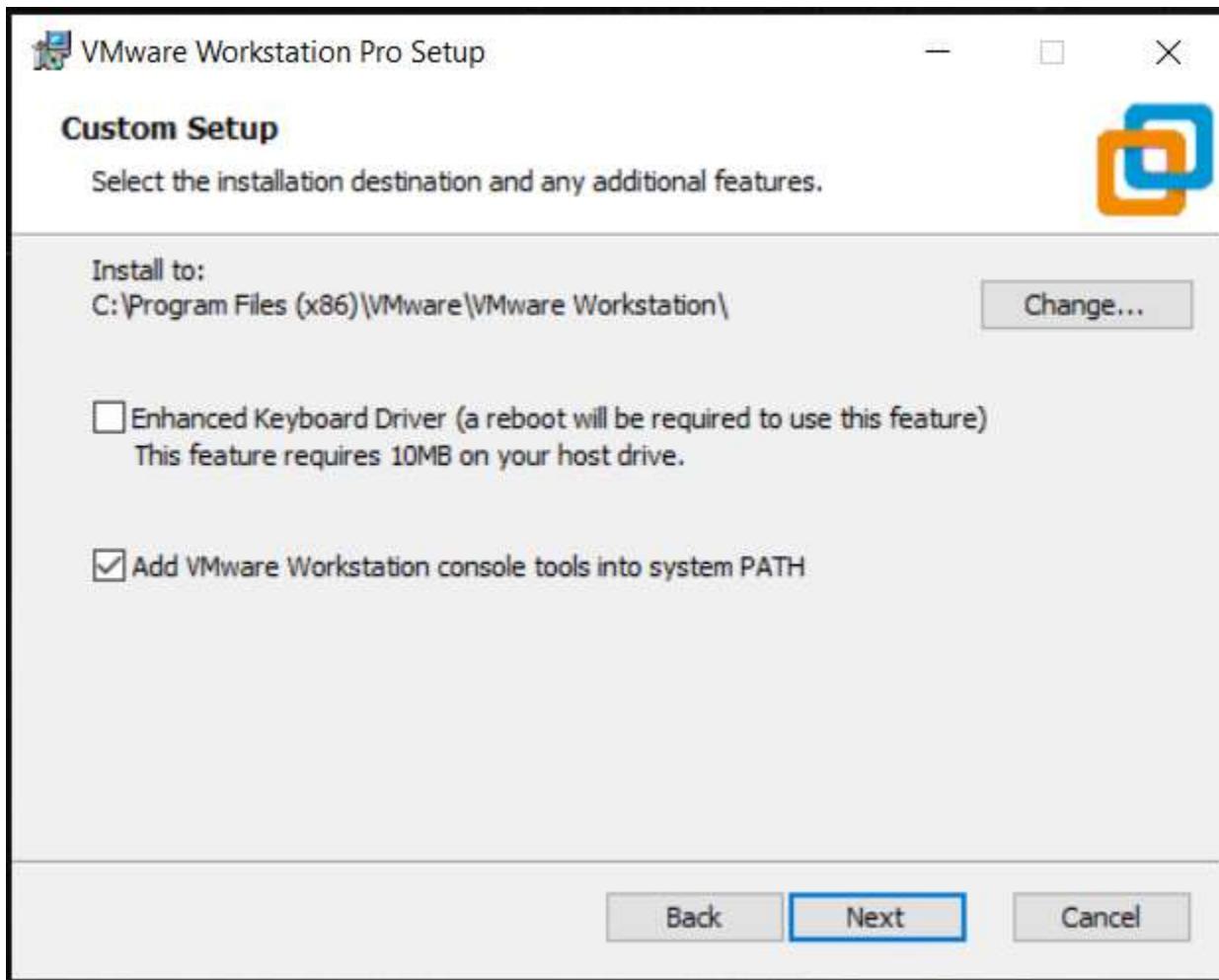


Figure 179: Installation of VM workstation 4.

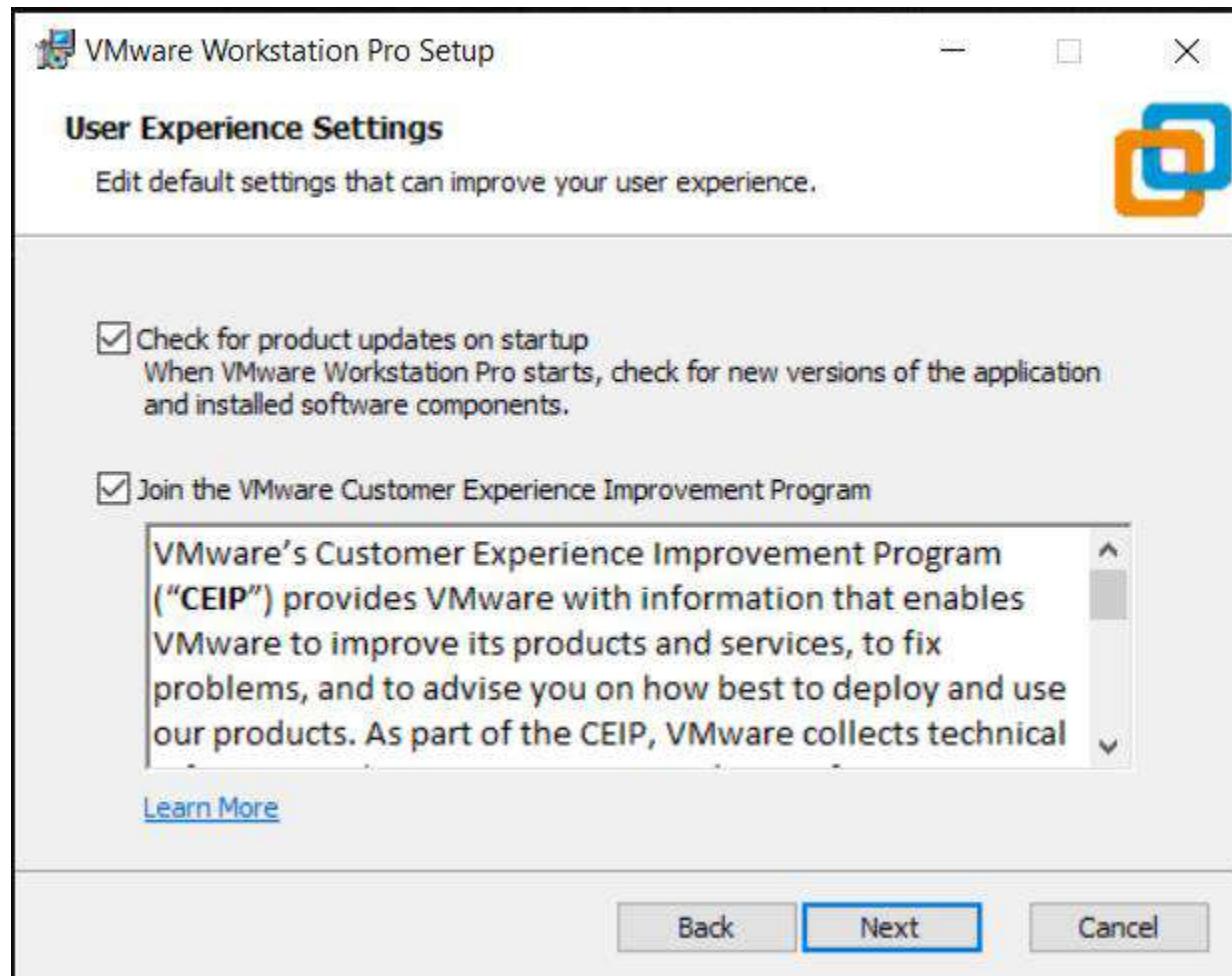


Figure 180: Installation of VM workstation 5.

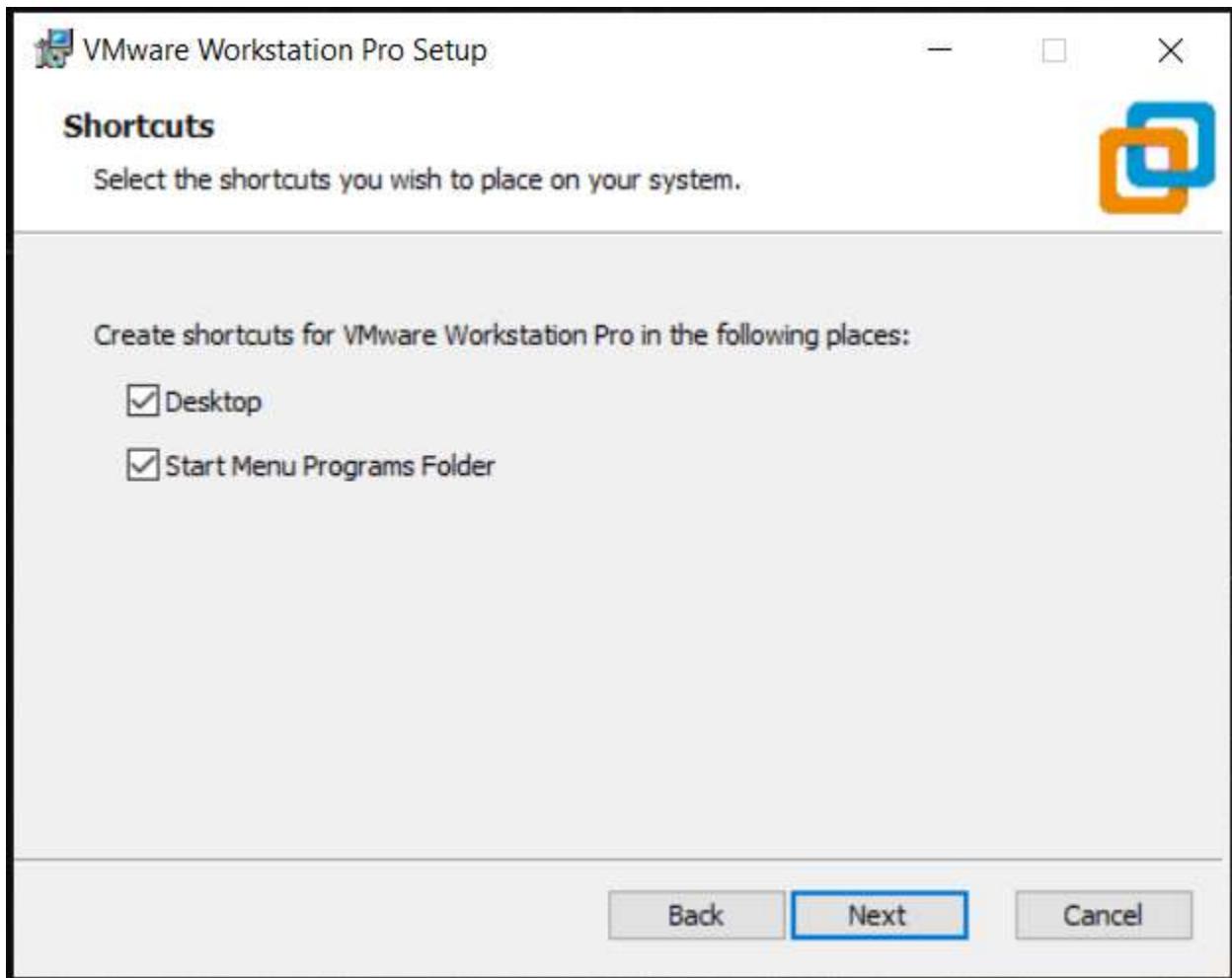


Figure 181: Installation of VM workstation 6.

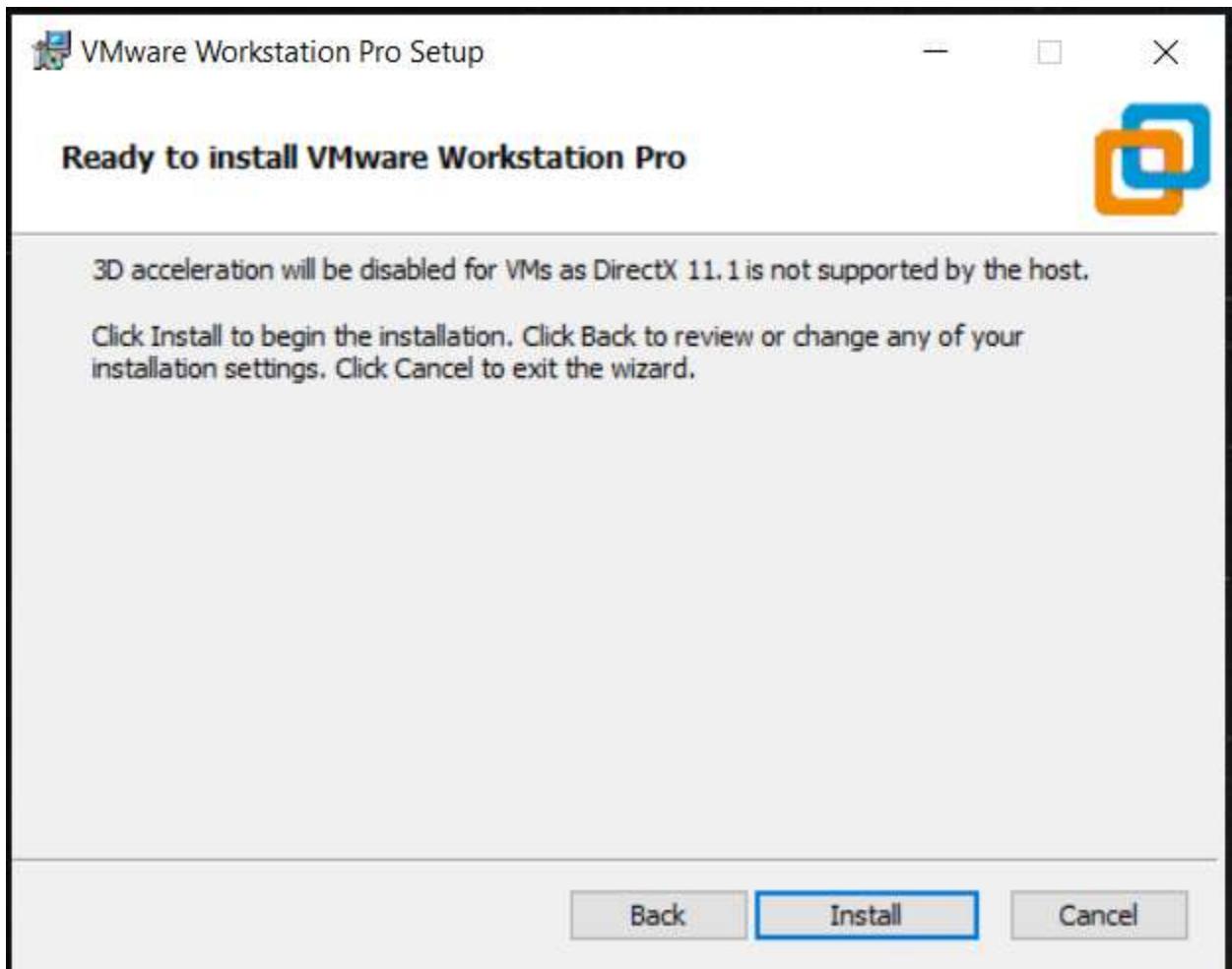


Figure 182: Installation of VM workstation 7.

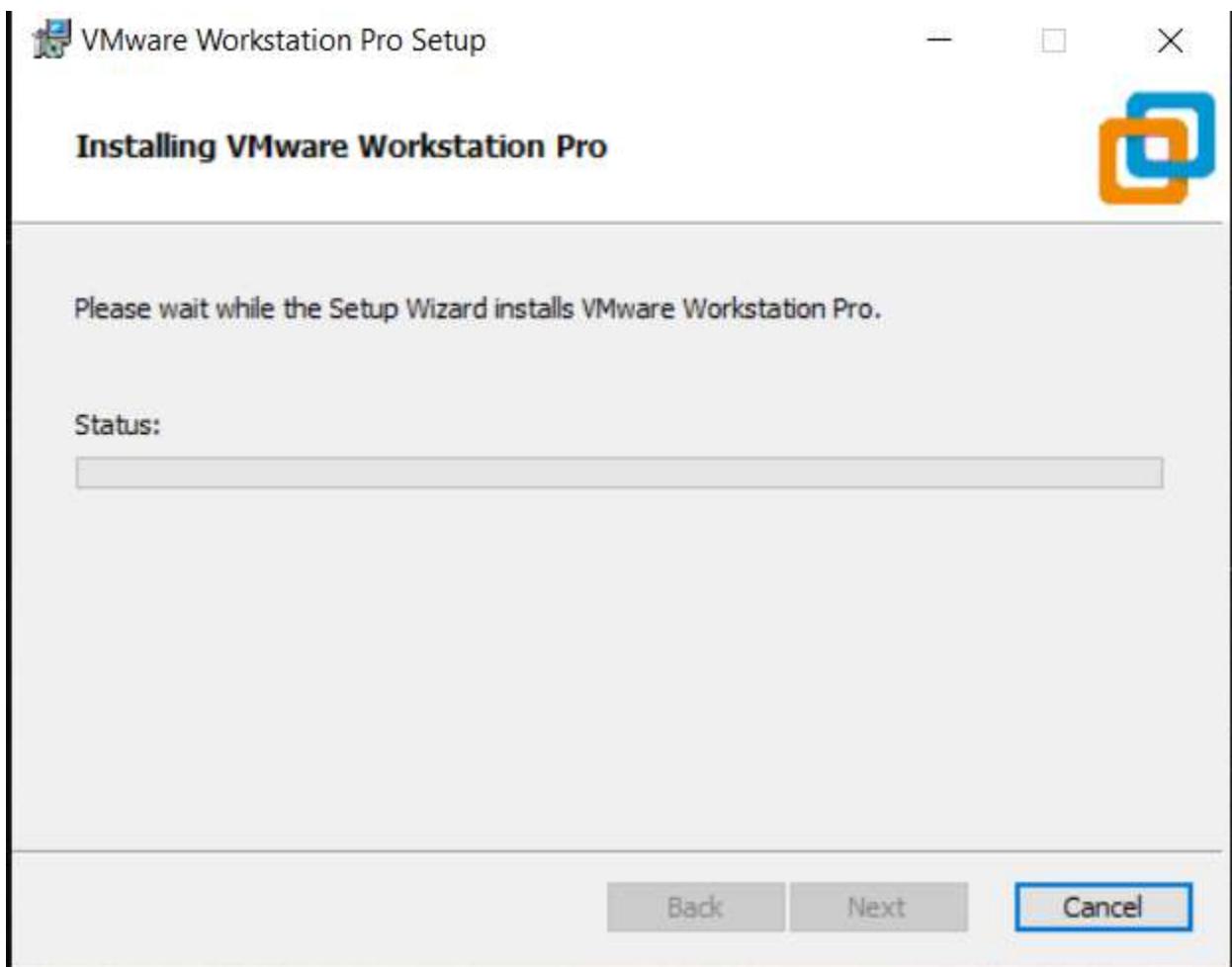


Figure 183: Installation of VM workstation 8.

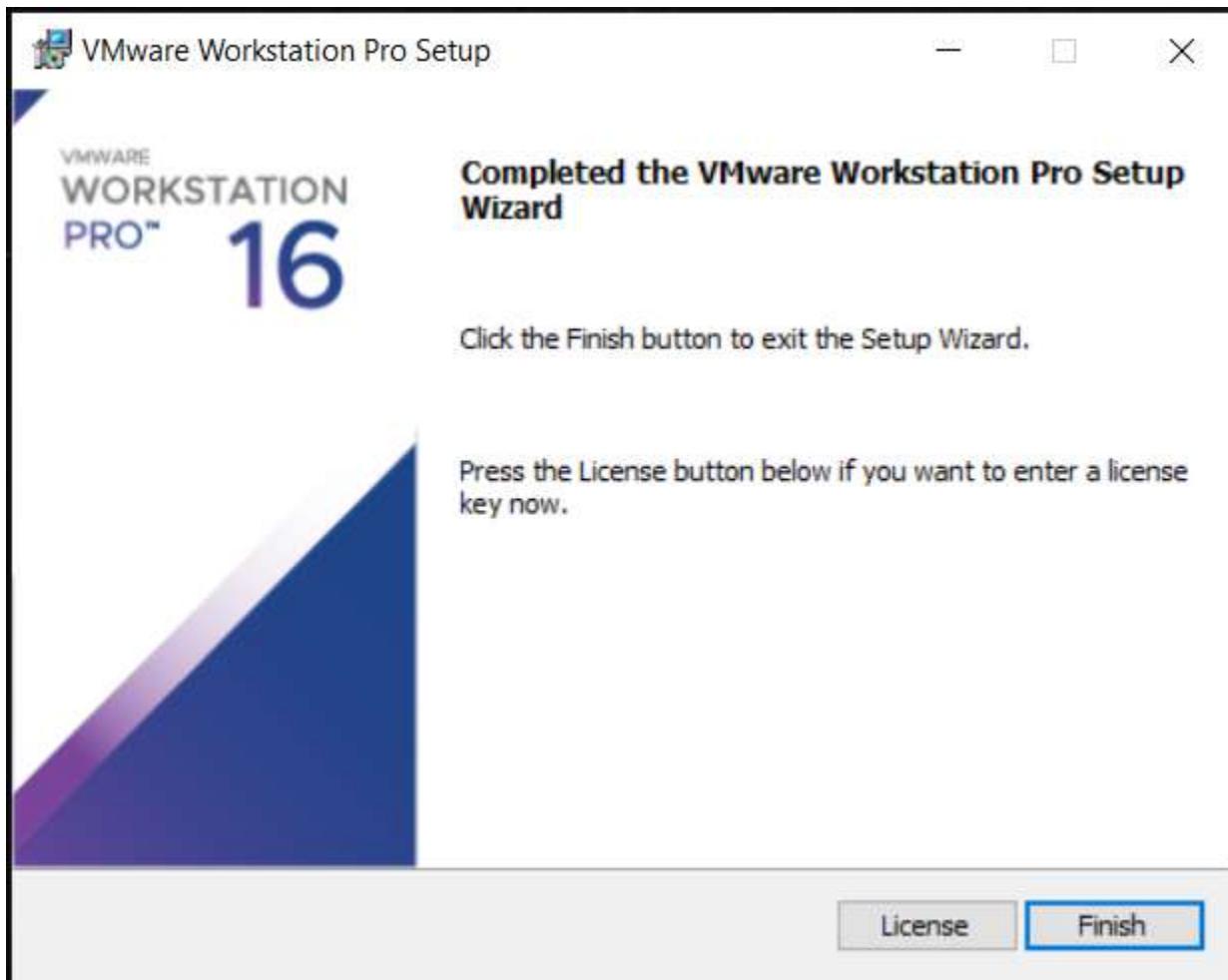


Figure 184: Installation of VM workstation 9.

8.5.2 Window 10

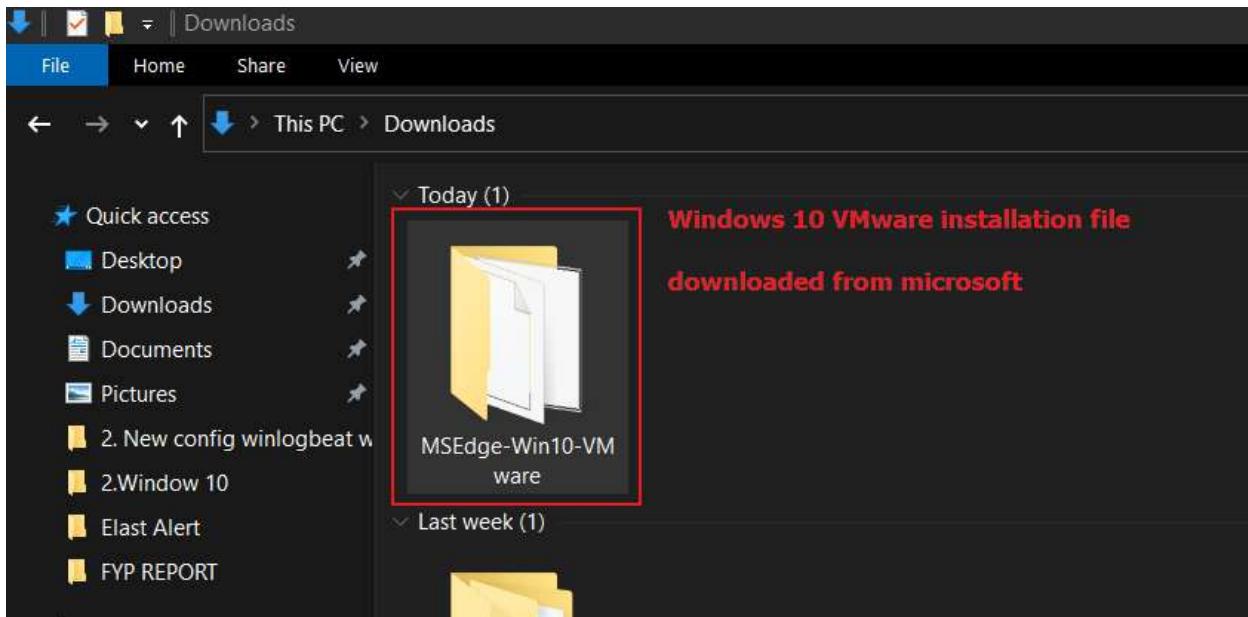


Figure 185: Installing windows 10 VM image 1.

Downloads > MSEdge-Win10-VMware				
Name	Date modified	Type	Size	
MSEdge-Win10-VMware.mf	3/19/2019 7:25 PM	MF File	1 KB	
MSEdge-Win10-VMware	3/19/2019 7:25 PM	Open Virtualizatio...	6 KB	
MSEdge-Win10-VMware-disk1	3/19/2019 7:25 PM	VMware virtual dis...	7,122,448 ...	

Figure 186: Installing windows 10 VM image 2.

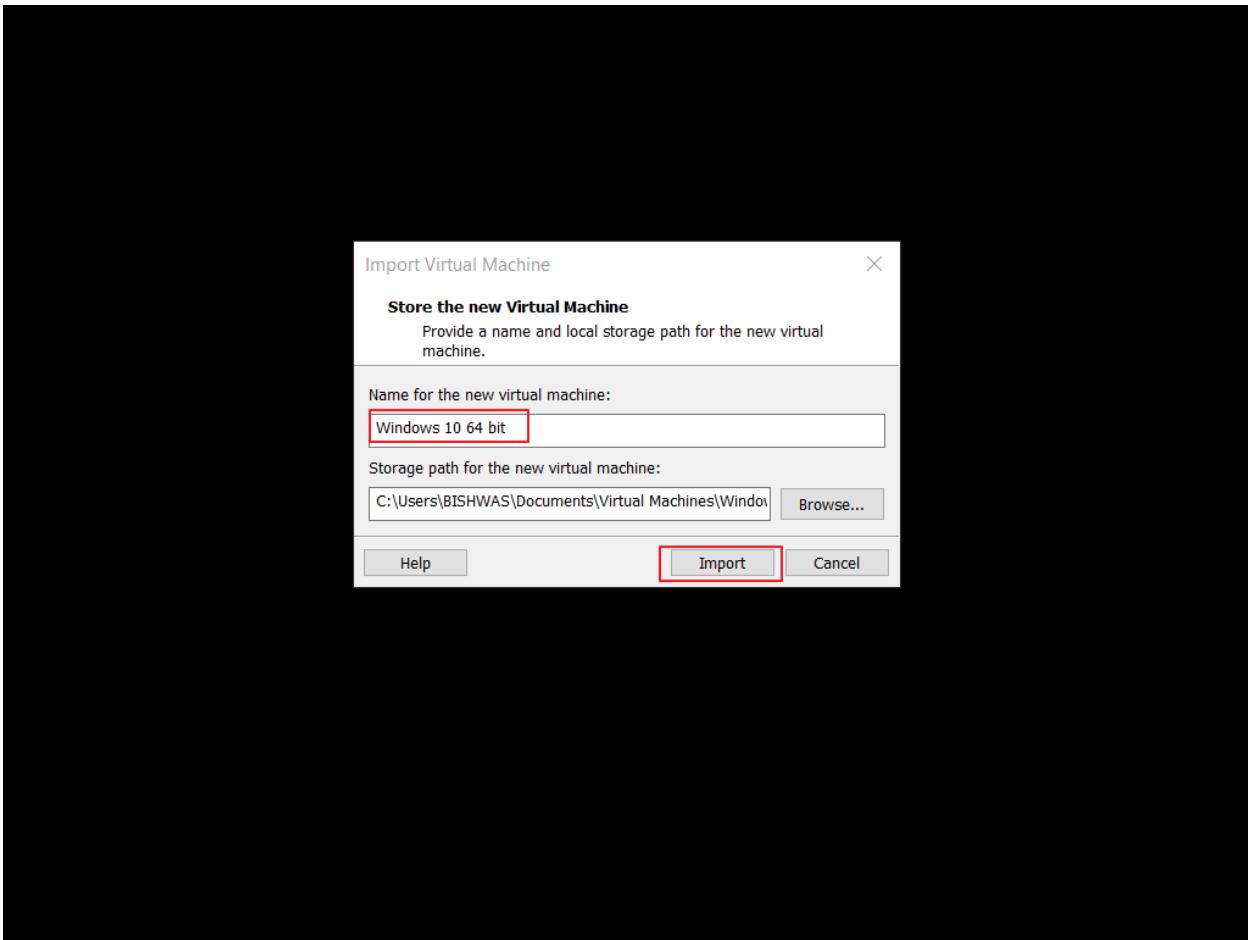


Figure 187: Installing windows 10 VM image 3.

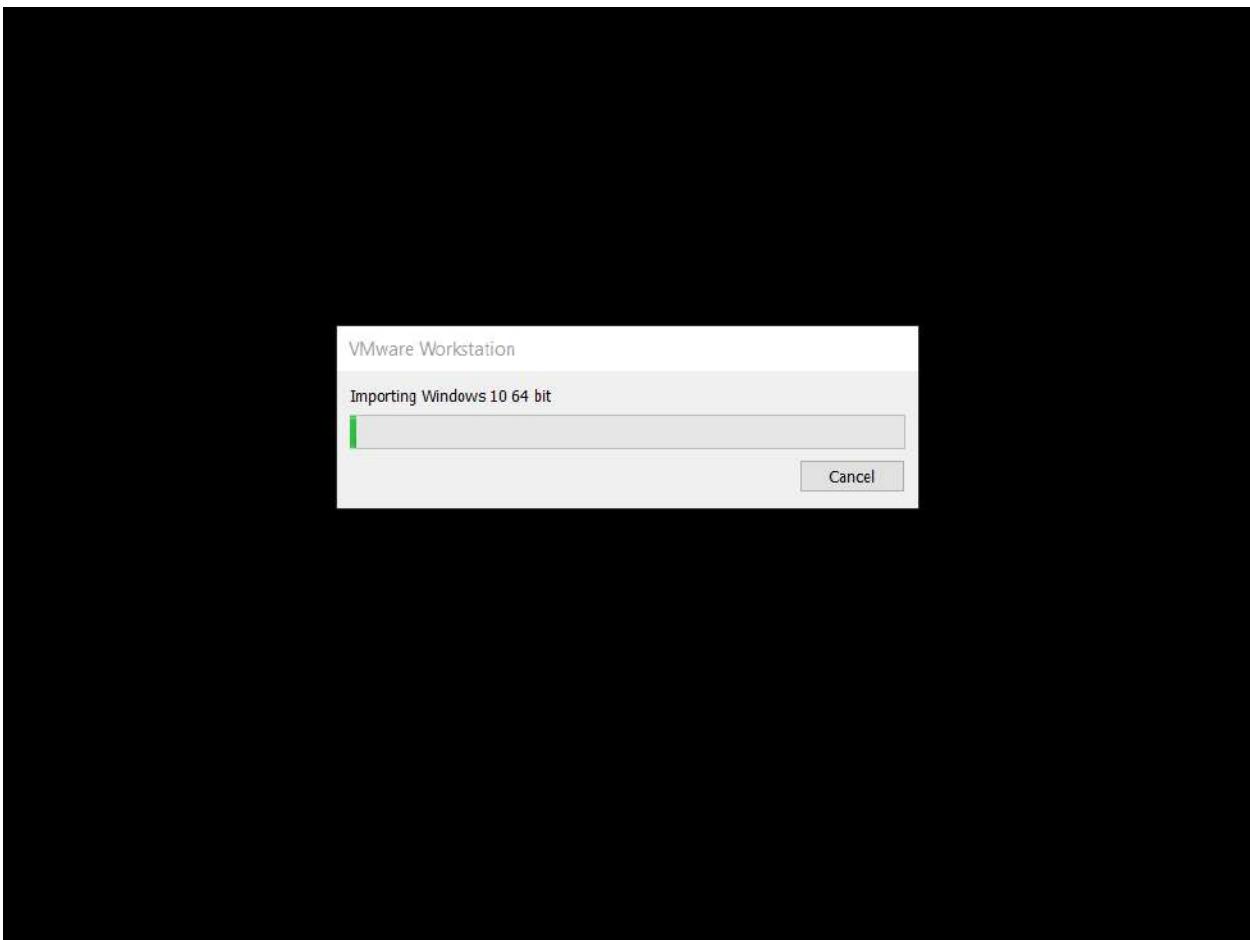


Figure 188: Installing windows 10 VM image 4.



Figure 189: Installing windows 10 VM image 5.

Installation and configuration of winlogbeat

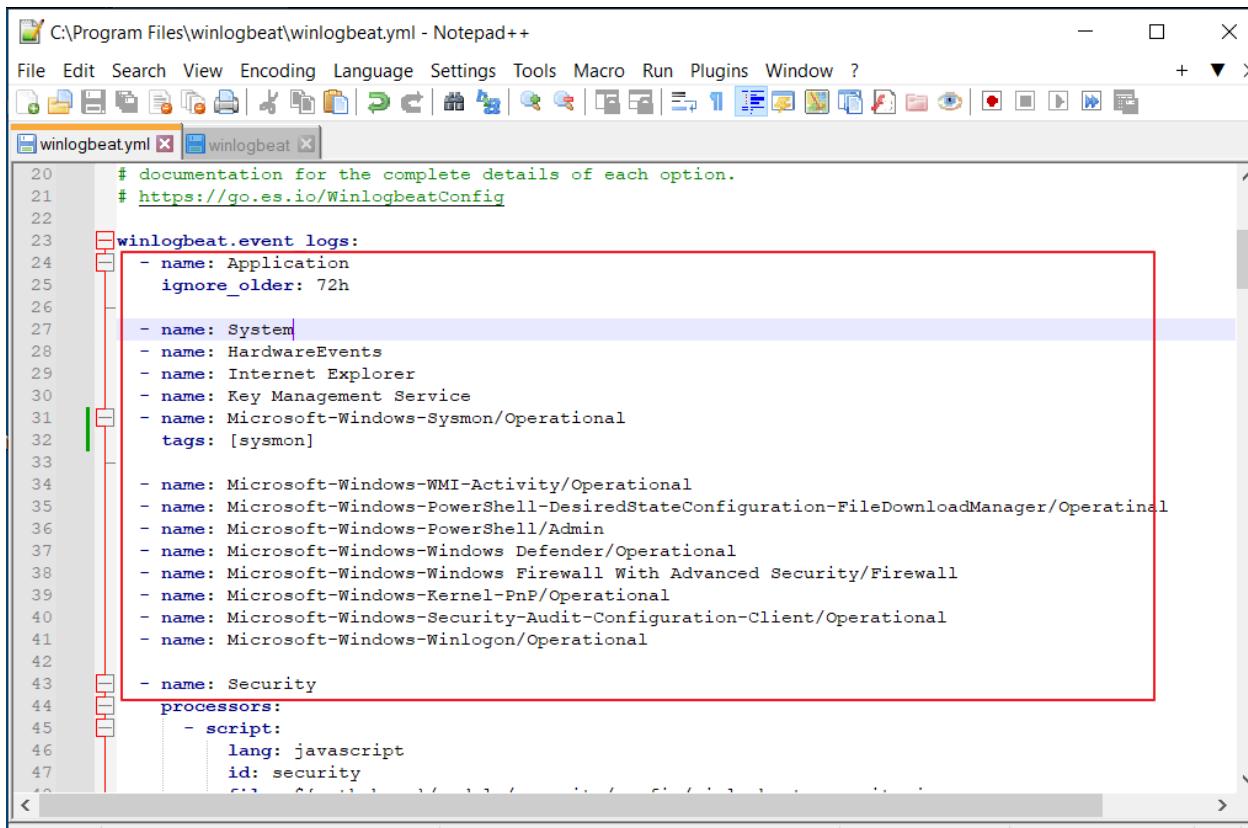
The screenshot shows an Administrator Windows PowerShell window. The command `Get-ExecutionPolicy -List` is run, showing the execution policy settings for MachinePolicy, UserPolicy, Process, CurrentUser, and LocalMachine. The LocalMachine policy is set to RemoteSigned. The command `Set-ExecutionPolicy unrestricted` is run to change the execution policy. A security warning follows, asking if the user wants to run the script. The user selects 'Run once' (R). Finally, the command `.\install-service-winlogbeat.ps1` is run, which installs the service. The status of the service 'winlogbeat' is shown as Stopped.

```
Administrator: Windows PowerShell
PS C:\Program Files\winlogbeat> Get-ExecutionPolicy -List
Scope ExecutionPolicy
-----
MachinePolicy      Undefined
UserPolicy         Undefined
Process           Undefined
CurrentUser        Undefined
LocalMachine       RemoteSigned

PS C:\Program Files\winlogbeat> Set-ExecutionPolicy unrestricted
Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about_Execution_Policies help topic at
https://go.microsoft.com/fwlink/?LinkId=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A
PS C:\Program Files\winlogbeat> .\install-service-winlogbeat.ps1
Security warning
Run only scripts that you trust. While scripts from the internet can be useful, this script can potentially harm your computer. If you trust this script, use the Unblock-File cmdlet to allow the script to run without this warning message. Do you want to run C:\Program Files\winlogbeat\install-service-winlogbeat.ps1?
[D] Do not run [R] Run once [S] Suspend [?] Help (default is "D"): R
Status   Name            DisplayName
-----   --   -----
Stopped  winlogbeat      winlogbeat

PS C:\Program Files\winlogbeat>
```

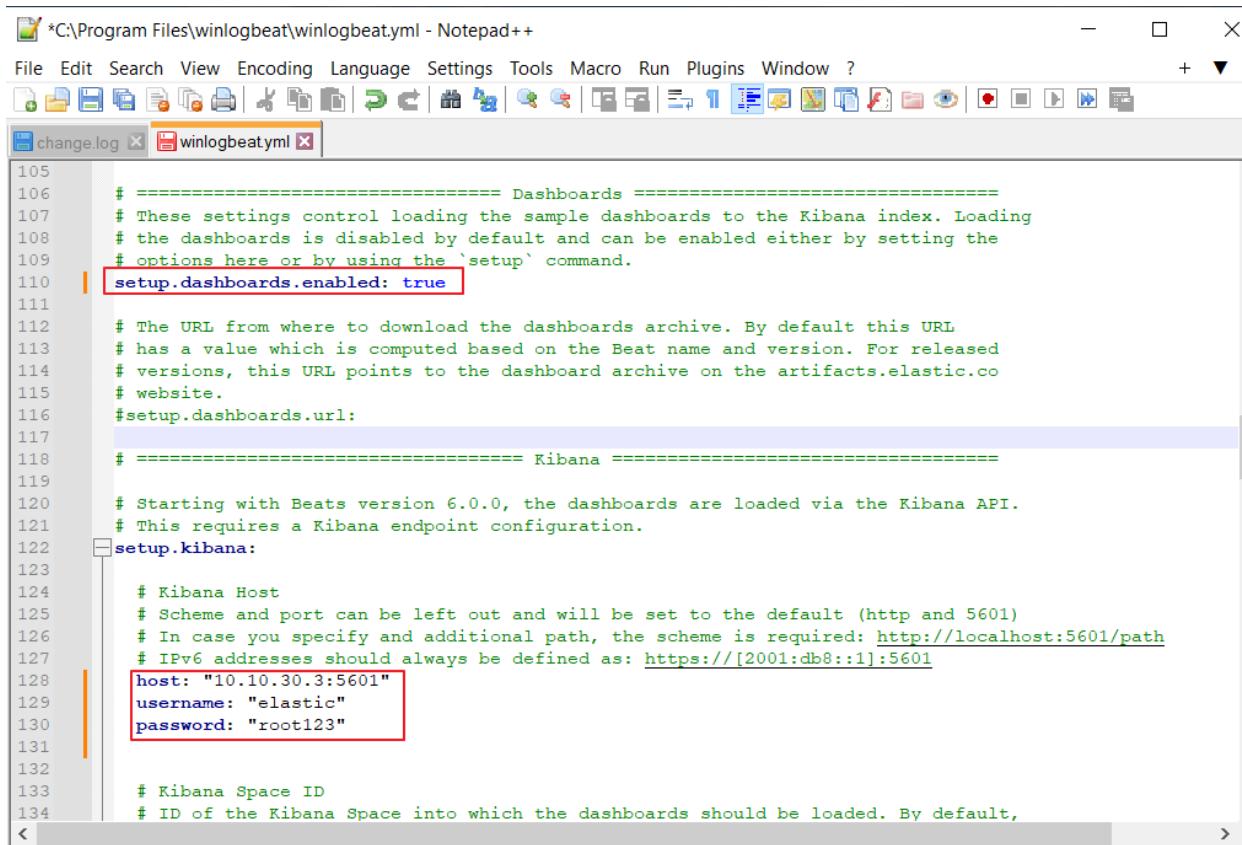
Figure 190: Installing winlogbeat .



The screenshot shows the Notepad++ application window with the file `C:\Program Files\winlogbeat\winlogbeat.yml` open. The window title bar reads "C:\Program Files\winlogbeat\winlogbeat.yml - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and Help. The toolbar below the menu bar contains various icons for file operations like Open, Save, Print, and Find. Below the toolbar, two tabs are visible: "winlogbeat.yml" (which is active) and "winlogbeat". The main code editor area displays the following YAML configuration:

```
20  # documentation for the complete details of each option.
21  # https://go.es.io/WinlogbeatConfig
22
23  winlogbeat.event_logs:
24      - name: Application
25          ignore_older: 72h
26
27      - name: System
28      - name: HardwareEvents
29      - name: Internet Explorer
30      - name: Key Management Service
31      - name: Microsoft-Windows-Sysmon/Operational
32          tags: [sysmon]
33
34      - name: Microsoft-Windows-WMI-Activity/Operational
35      - name: Microsoft-Windows-PowerShell-DesiredStateConfiguration-FileDownloadManager/Operational
36      - name: Microsoft-Windows-PowerShell/Admin
37      - name: Microsoft-Windows-Windows Defender/Operational
38      - name: Microsoft-Windows-Windows Firewall With Advanced Security/Firewall
39      - name: Microsoft-Windows-Kernel-PnP/Operational
40      - name: Microsoft-Windows-Security-Audit-Configuration-Client/Operational
41      - name: Microsoft-Windows-Winlogon/Operational
42
43      - name: Security
44  processors:
45      - script:
46          lang: javascript
47          id: security
48
```

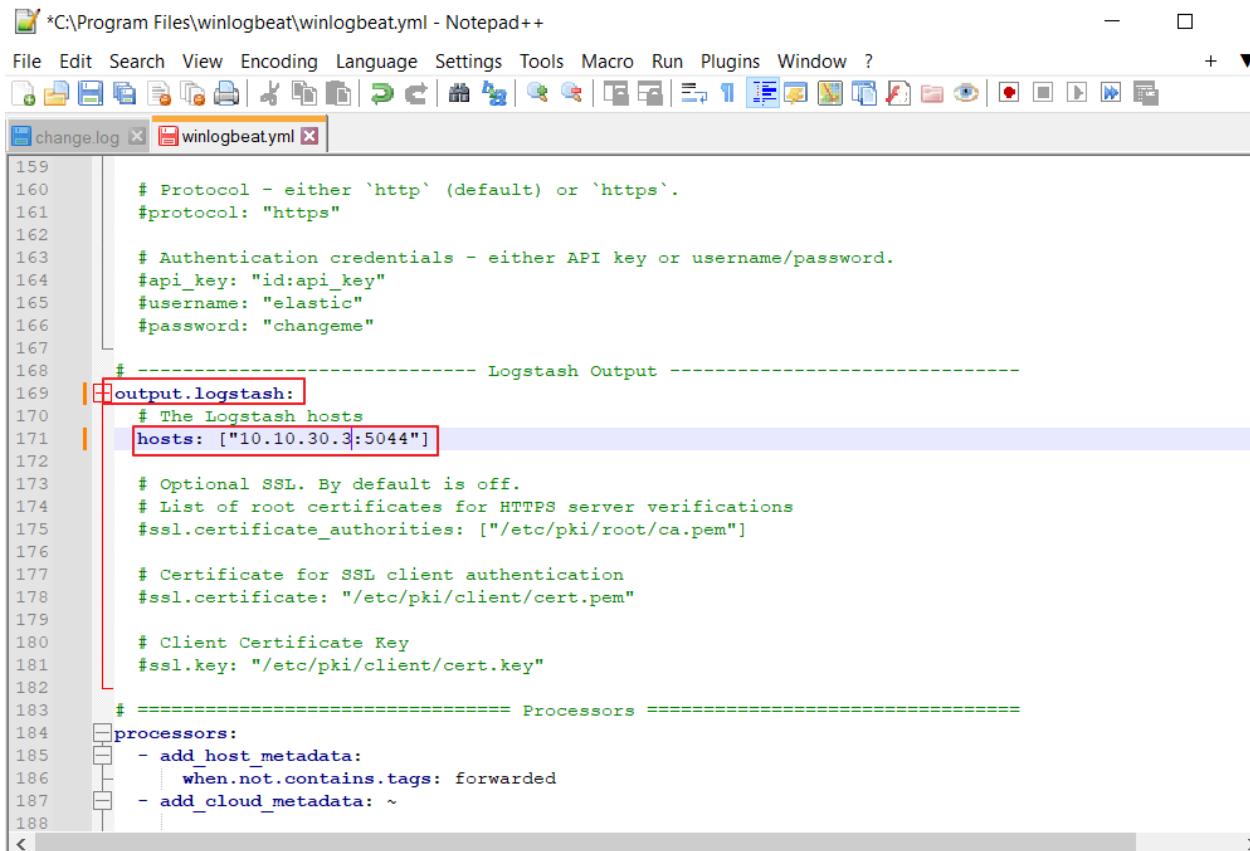
Figure 191: Configuring winlogbeat yml file 1.



The screenshot shows a Notepad++ window with the file `winlogbeat.yml` open. The code is a configuration file for Winlogbeat, specifically for setting up dashboards and Kibana. Several lines of code are highlighted with red boxes:

```
105
106      # ===== Dashboards =====
107      # These settings control loading the sample dashboards to the Kibana index. Loading
108      # the dashboards is disabled by default and can be enabled either by setting the
109      # options here or by using the `setup` command.
110      setup.dashboards.enabled: true
111
112      # The URL from where to download the dashboards archive. By default this URL
113      # has a value which is computed based on the Beat name and version. For released
114      # versions, this URL points to the dashboard archive on the artifacts.elastic.co
115      # website.
116      #setup.dashboards.url:
117
118      # ===== Kibana =====
119
120      # Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
121      # This requires a Kibana endpoint configuration.
122      setup.kibana:
123
124          # Kibana Host
125          # Scheme and port can be left out and will be set to the default (http and 5601)
126          # In case you specify and additional path, the scheme is required: http://localhost:5601/path
127          # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
128          host: "10.10.30.3:5601"
129          username: "elastic"
130          password: "root123"
131
132
133          # Kibana Space ID
134          # ID of the Kibana Space into which the dashboards should be loaded. By default,
```

Figure 192: Configuring winlogbeat yml file 2.



The screenshot shows the Notepad++ application window with two tabs open: "change.log" and "winlogbeat.yml". The "winlogbeat.yml" tab contains the configuration code. A red box highlights the "output.logstash:" section, specifically the "hosts" field which is set to ["10.10.30.3:5044"]. The code is a YAML file with the following content:

```
159
160     # Protocol - either 'http' (default) or 'https'.
161     #protocol: "https"
162
163     # Authentication credentials - either API key or username/password.
164     #api_key: "id:api_key"
165     #username: "elastic"
166     #password: "changeme"
167
168     # ----- Logstash Output -----
169     output.logstash:
170         # The Logstash hosts
171         hosts: ["10.10.30.3:5044"]
172
173         # Optional SSL. By default is off.
174         # List of root certificates for HTTPS server verifications
175         #ssl.certificateAuthorities: ["/etc/pki/root/ca.pem"]
176
177         # Certificate for SSL client authentication
178         #ssl.certificate: "/etc/pki/client/cert.pem"
179
180         # Client Certificate Key
181         #ssl.key: "/etc/pki/client/cert.key"
182
183     # ===== Processors =====
184     processors:
185         - add_host_metadata:
186             when.not.contains.tags: forwarded
187         - add_cloud_metadata: ~
```

Figure 193: Configuring winlogbeat yml file 3.

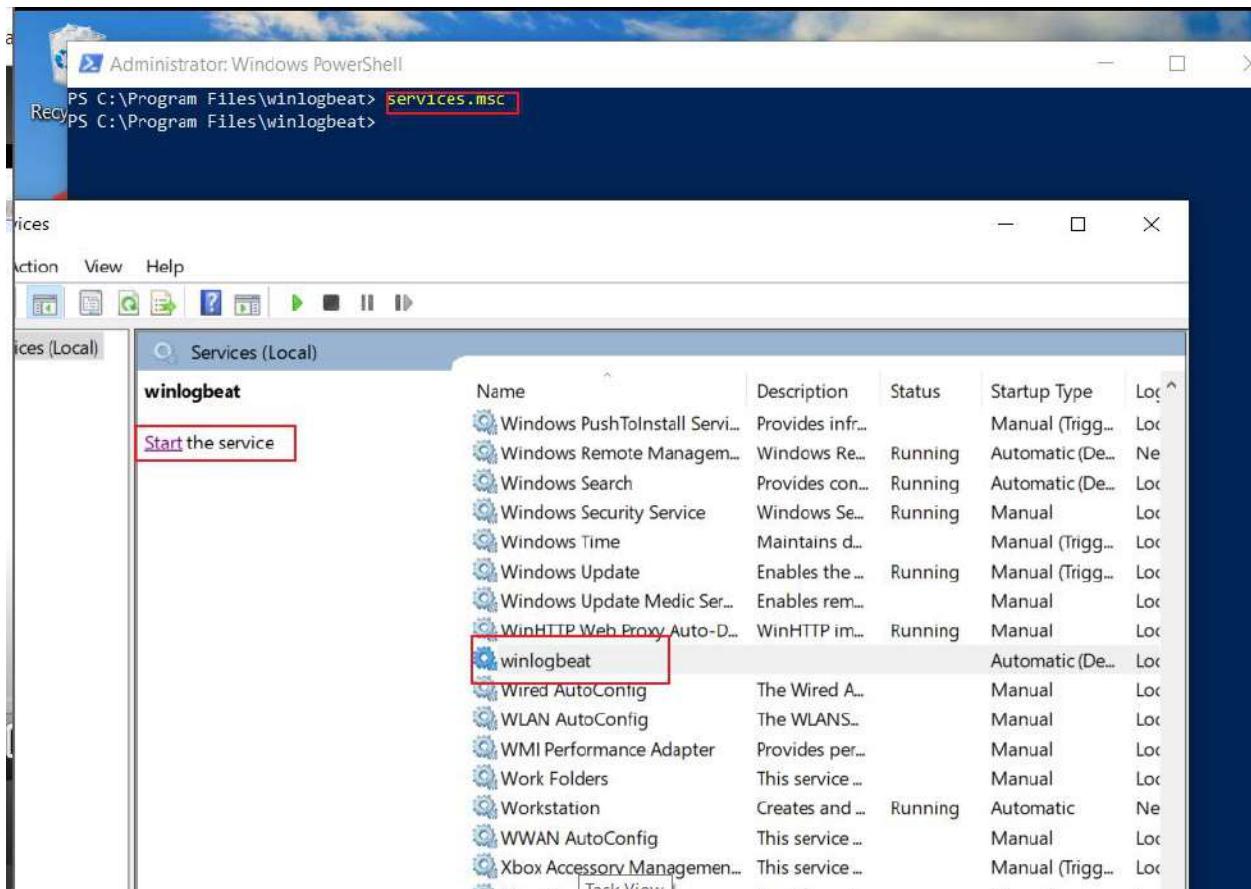


Figure 194: Starting winlogbeat service.

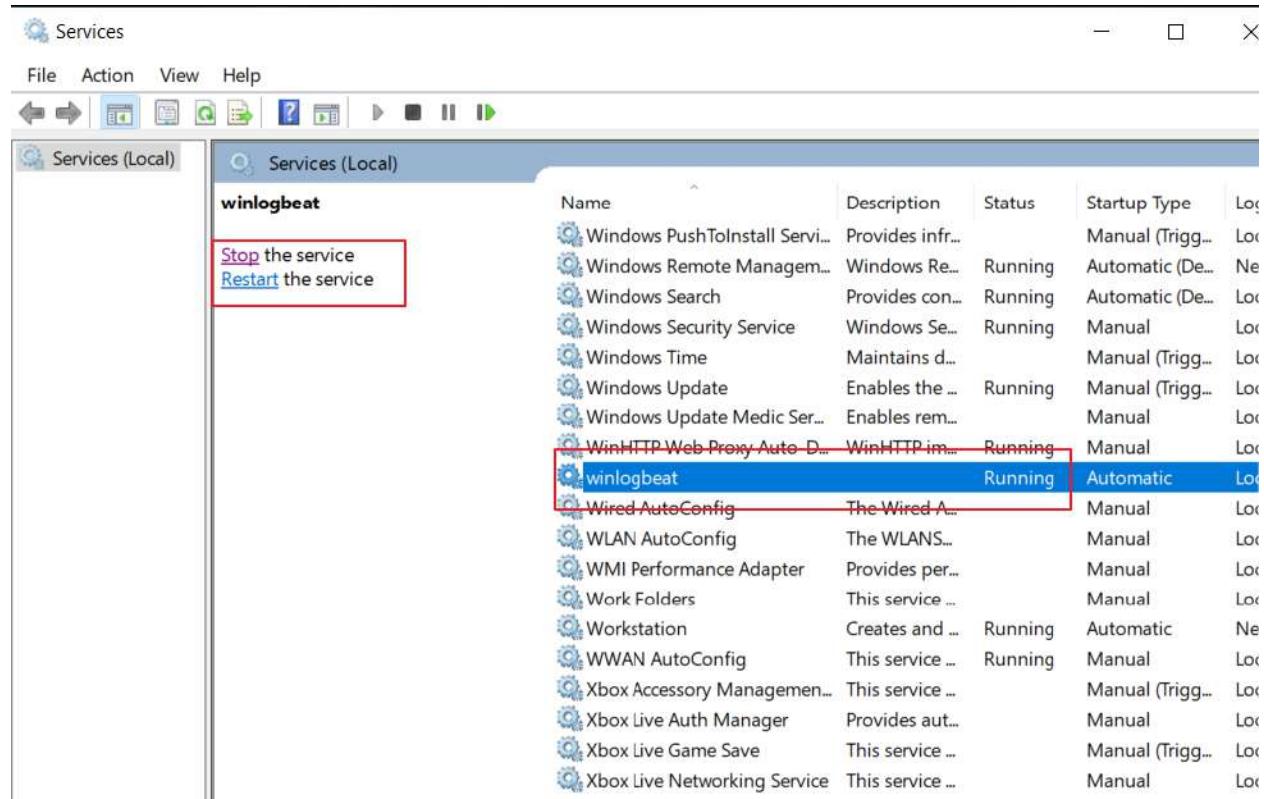


Figure 195: Winlogbeat started.

Installation and configuration of auditbeat

The screenshot shows a Windows PowerShell window titled "Administrator: Windows PowerShell". The session starts with the standard Microsoft copyright notice. The user runs several commands to manage execution policies and install the auditbeat service:

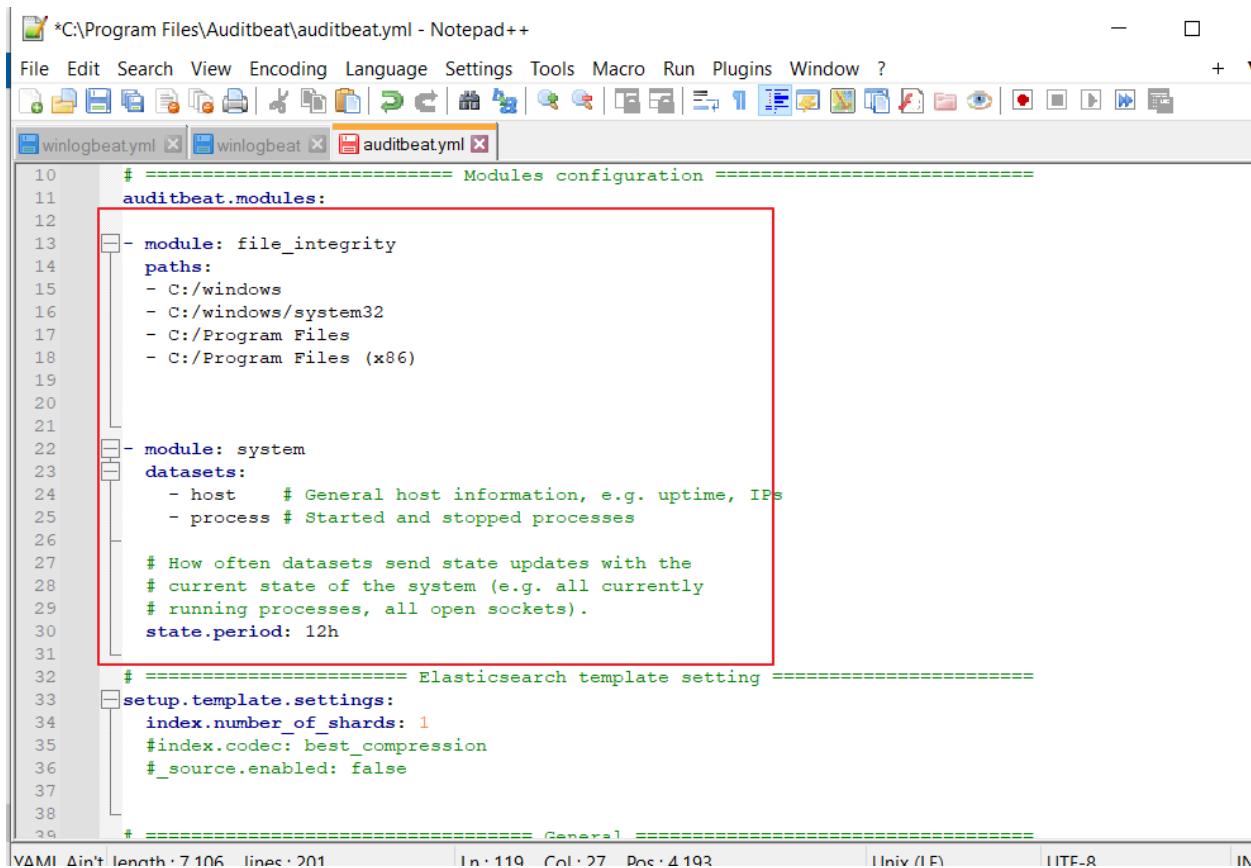
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> cd 'C:\Program Files\Auditbeat'
PS C:\Program Files\Auditbeat> Get-ExecutionPolicy -List
Scope ExecutionPolicy
-----
MachinePolicy      Undefined
UserPolicy         Undefined
Process           Undefined
CurrentUser        Undefined
LocalMachine       Unrestricted

PS C:\Program Files\Auditbeat> Set-ExecutionPolicy Unrestricted
Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N") A
PS C:\Program Files\Auditbeat>
PS C:\Program Files\Auditbeat> .\install-service-auditbeat.ps1
Status    Name          DisplayName
-----  -----
Stopped  auditbeat     auditbeat

PS C:\Program Files\Auditbeat>
```

Figure 196: Installing auditbeat in win 10.



```
# ===== Modules configuration =====
auditbeat.modules:
  - module: file_integrity
    paths:
      - C:/windows
      - C:/windows/system32
      - C:/Program Files
      - C:/Program Files (x86)

  - module: system
    datasets:
      - host # General host information, e.g. uptime, IPs
      - process # Started and stopped processes

    # How often datasets send state updates with the
    # current state of the system (e.g. all currently
    # running processes, all open sockets).
    state.period: 12h

# ===== Elasticsearch template setting =====
setup.template.settings:
  index.number_of_shards: 1
  #index.codec: best_compression
  #_source.enabled: false

# ===== General =====
```

Figure 197: Configuring auditbeat yml file 1.

```
*C:\Program Files\Auditbeat\auditbeat.yml - Notepad++  
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?  
winlogbeat.yml winlogbeat auditbeat.yml  
52 # env: staging  
53  
54 # ===== Dashboards =====  
55 # These settings control loading the sample dashboards to the Kibana index. Loading  
56 # the dashboards is disabled by default and can be enabled either by setting the  
57 # options here or by using the 'setup' command.  
58 setup.dashboards.enabled: true  
59  
60 # The URL from where to download the dashboards archive. By default this URL  
61 # has a value which is computed based on the Beat name and version. For released  
62 # versions, this URL points to the dashboard archive on the artifacts.elastic.co  
63 # website.  
64 $setup.dashboards.url:  
65  
66 # ===== Kibana =====  
67  
68 # Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.  
69 # This requires a Kibana endpoint configuration.  
70 setup.kibana:  
71  
72     # Kibana Host  
73     # Scheme and port can be left out and will be set to the default (http and 5601)  
74     # In case you specify an additional path, the scheme is required: http://localhost:5601/path  
75     # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601  
76     host: "10.10.30.3:5601"  
77     username: "elastic"  
78     password: "root123"  
79  
80     # Kibana Space ID  
81     # ID of the Kibana space into which the dashboards should be loaded. By default+  
YAML Ain't length: 7,104 lines: 201 Ln:60 Col:60 Pos:2,019 Unix (LF) UTF-8
```

Figure 198: Configuring auditbeat yml file 2.

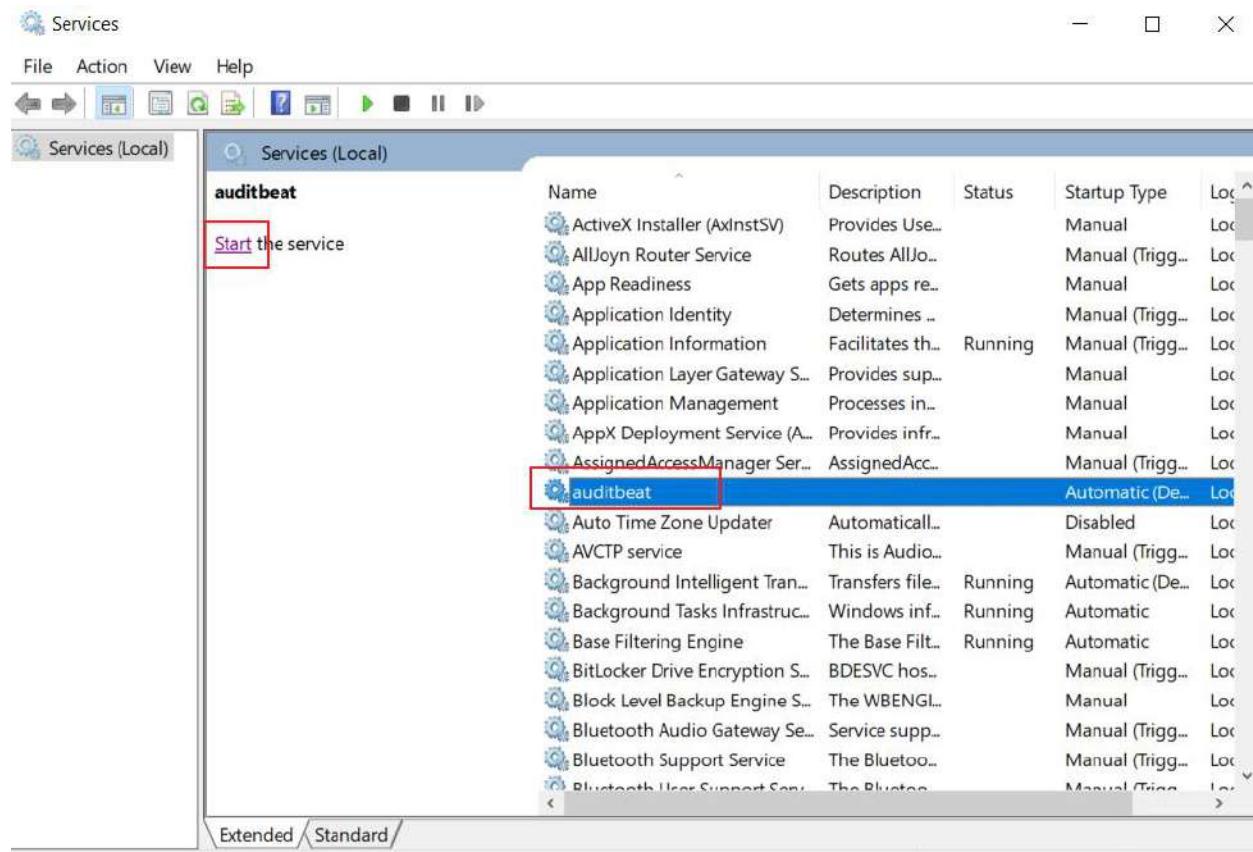


Figure 199: Starting auditbeat service.

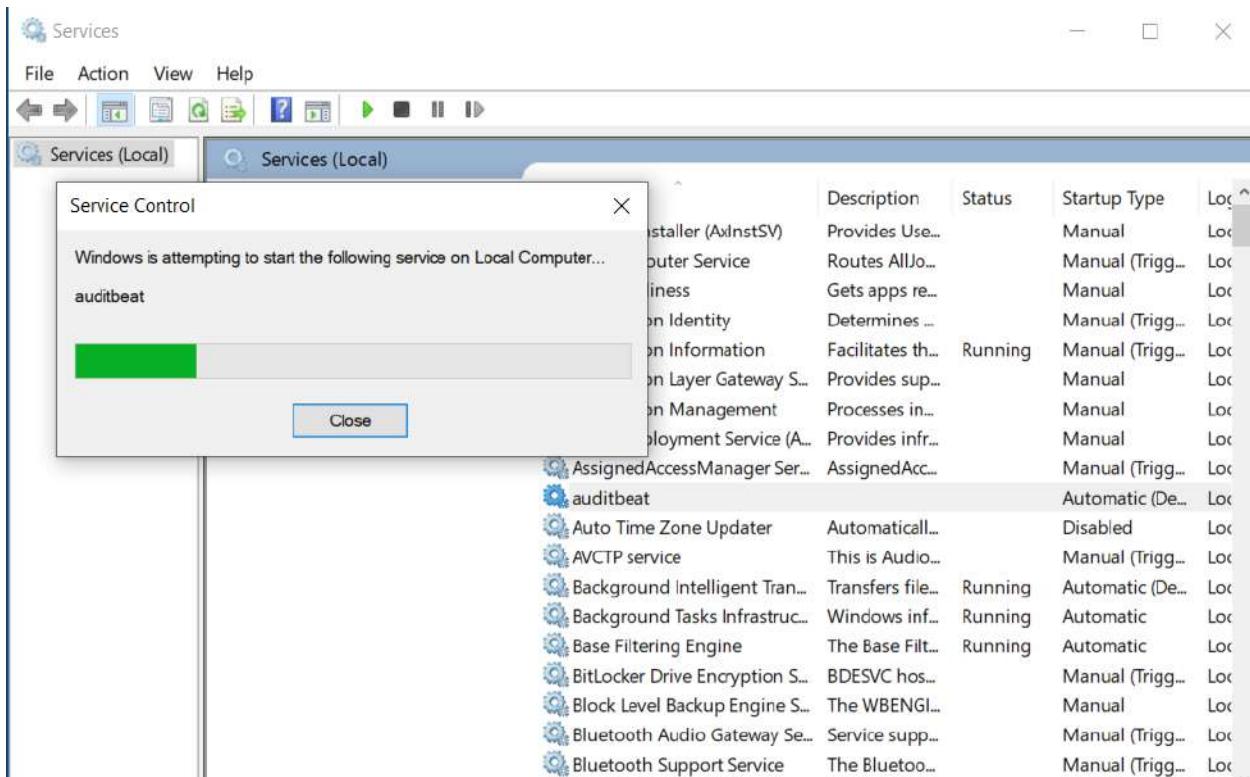


Figure 200: Auditbeat service started 1.

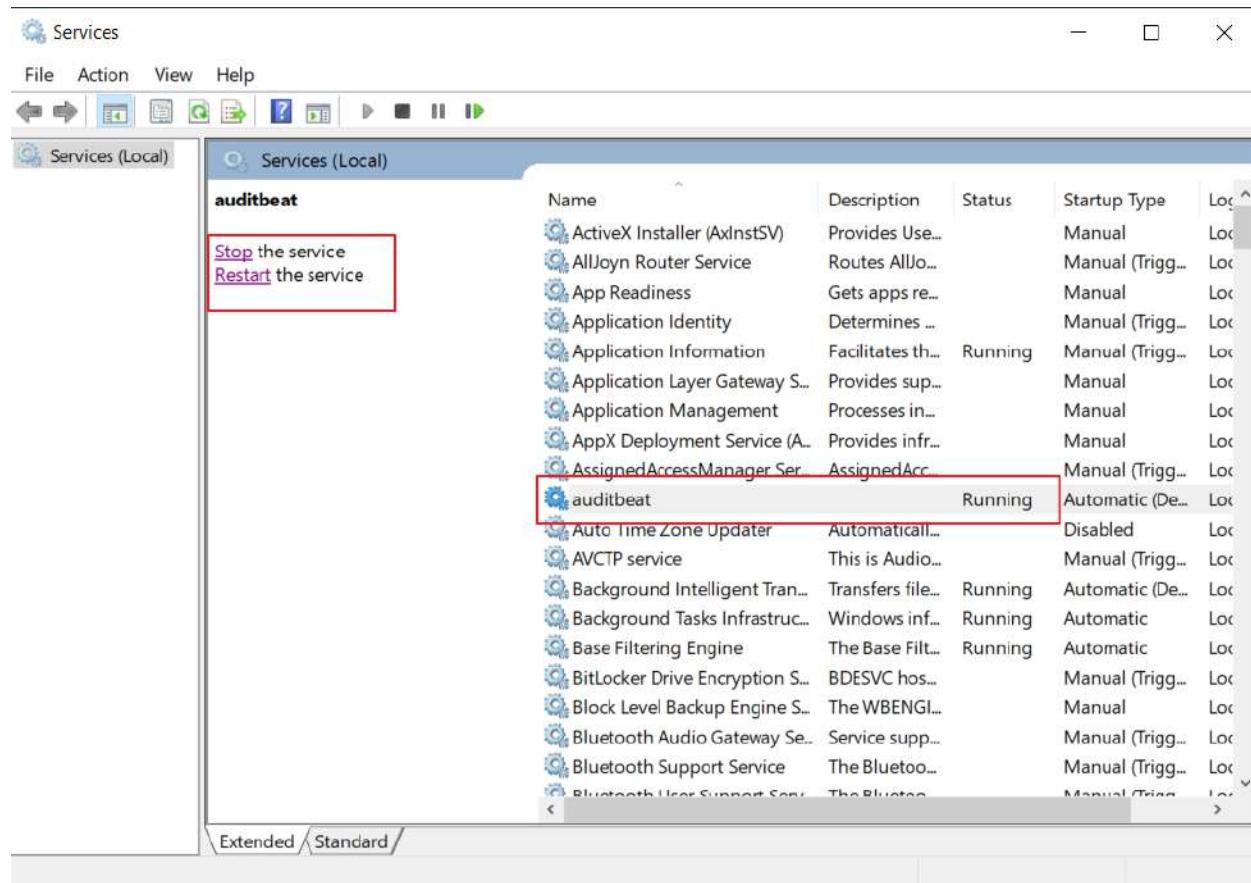


Figure 201: Auditbeat service started 2.

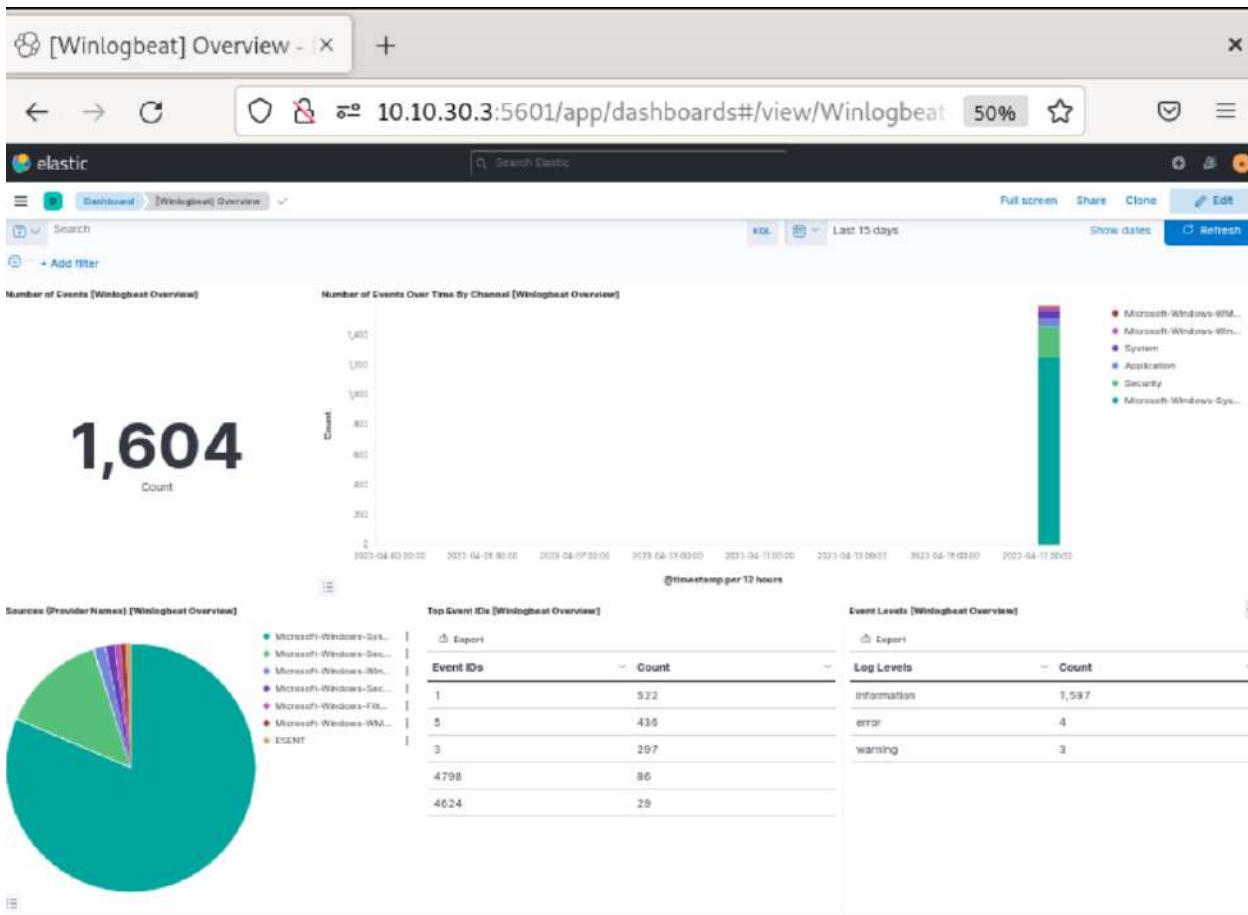


Figure 202: Winlogbeat Dashboard.

8.5.3 Debian 10 Linux

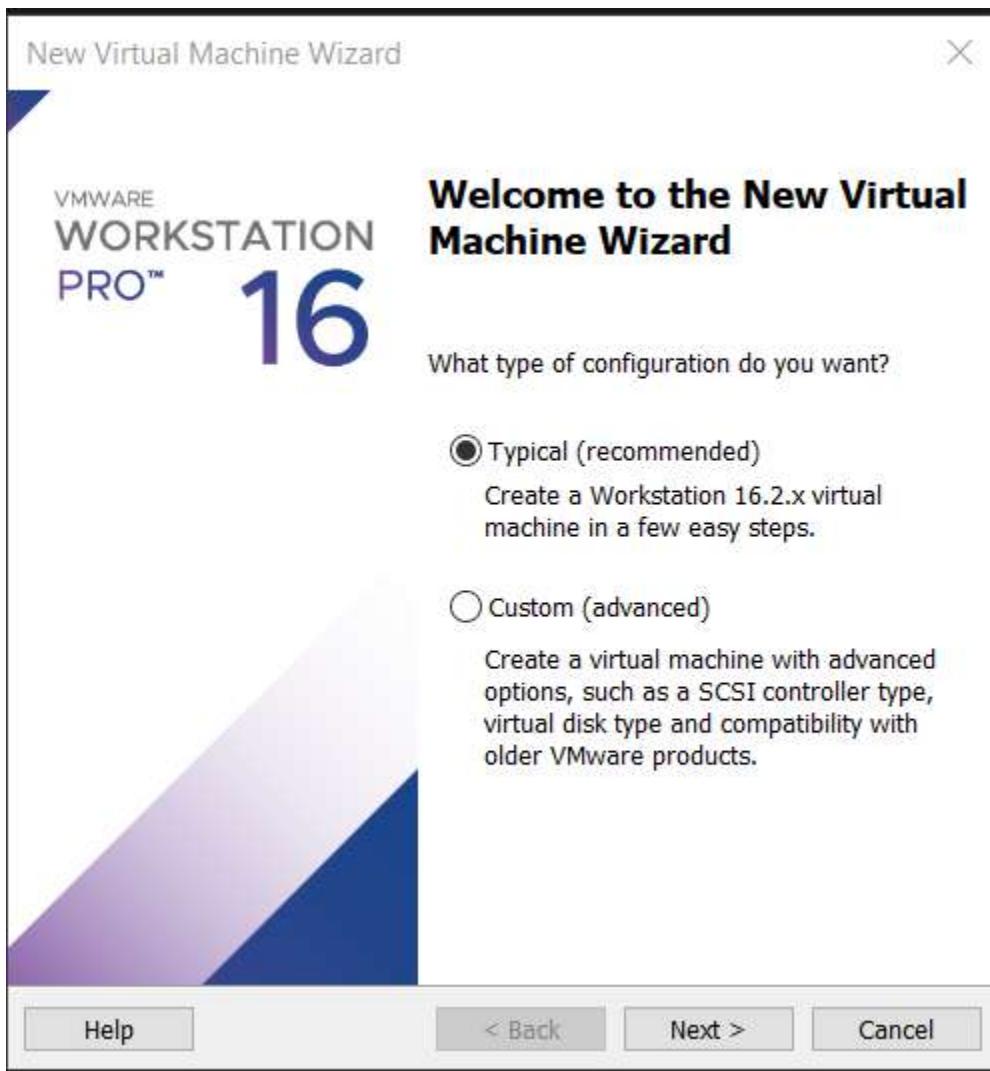


Figure 203: Installing Debian 10 on VM 1.

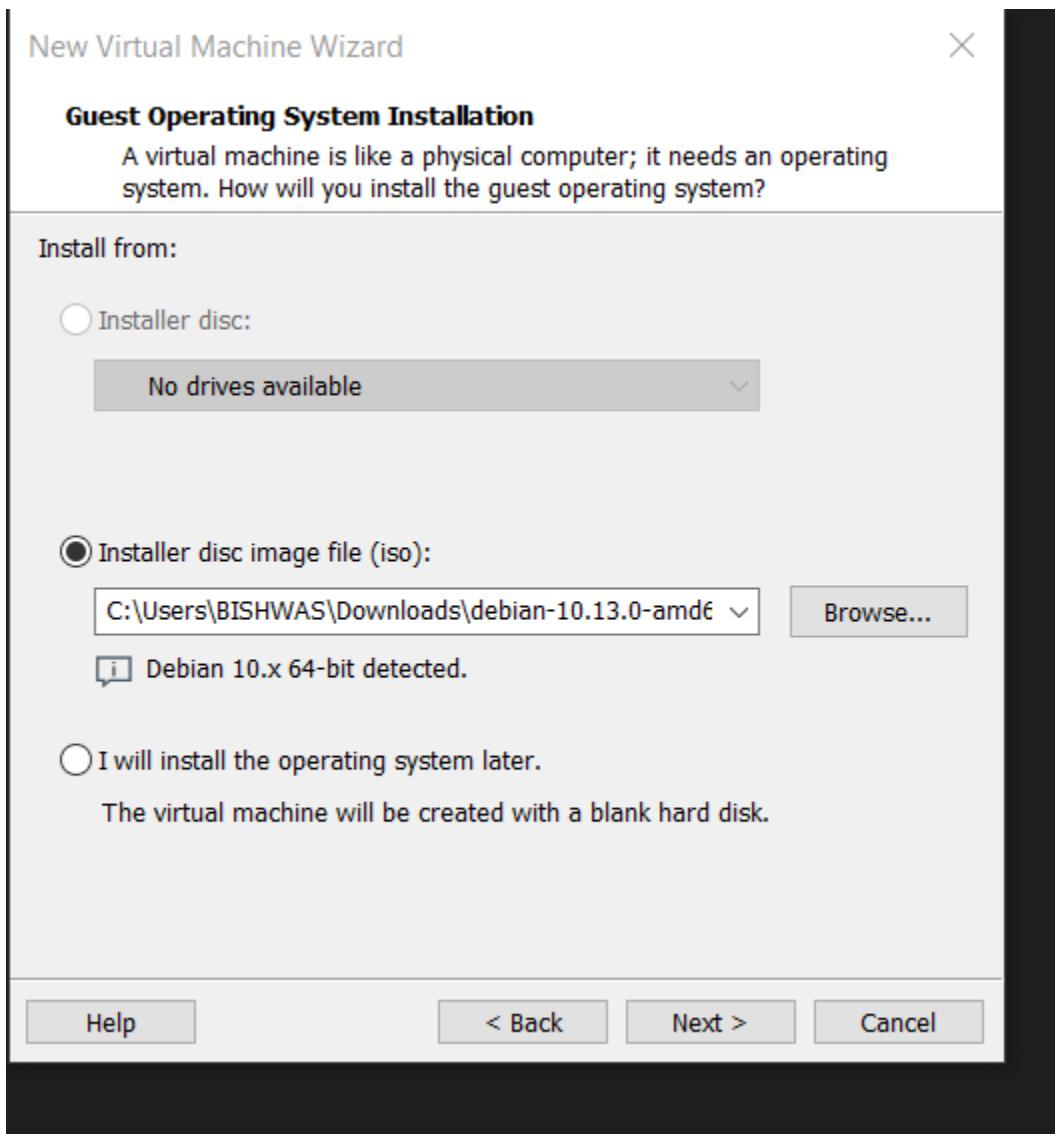


Figure 204: Installing Debian 10 on VM 2.

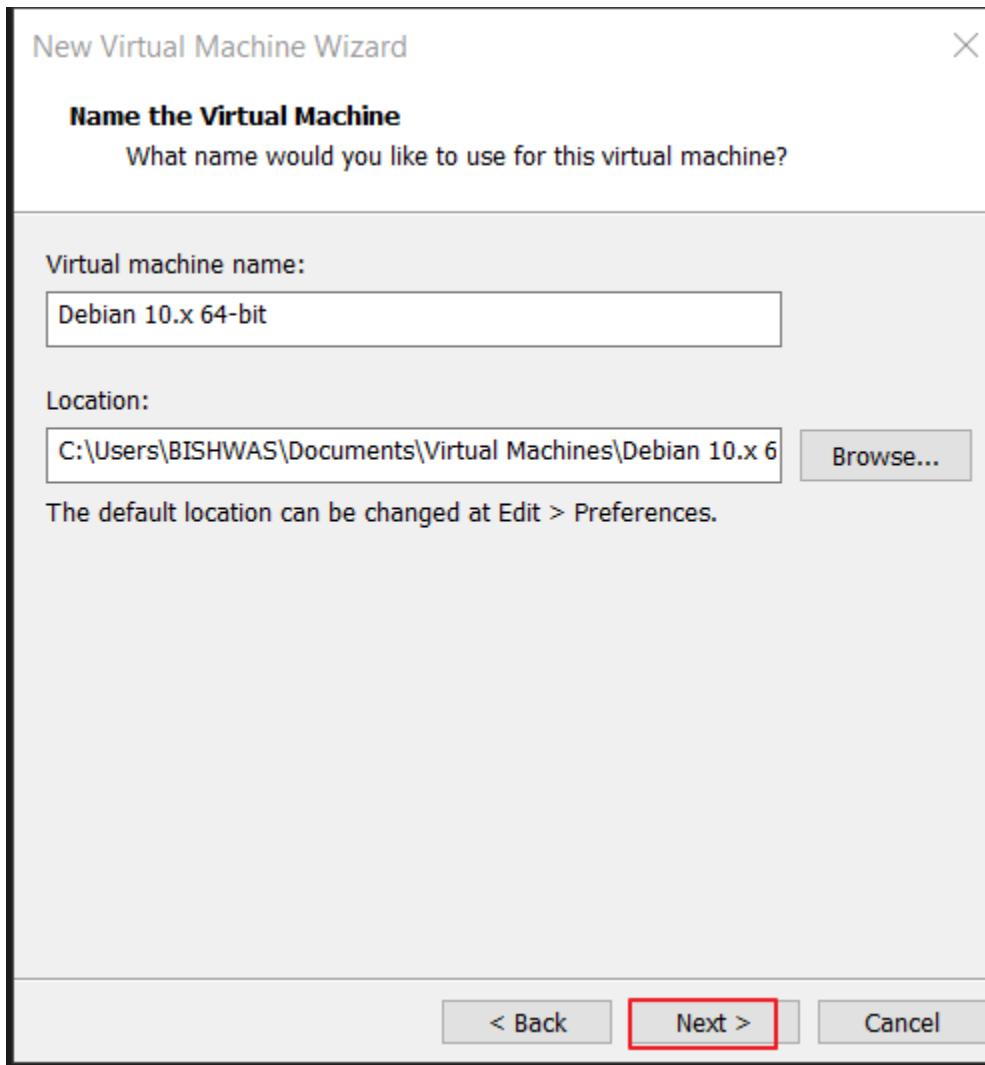


Figure 205: Installing Debian 10 on VM 3.

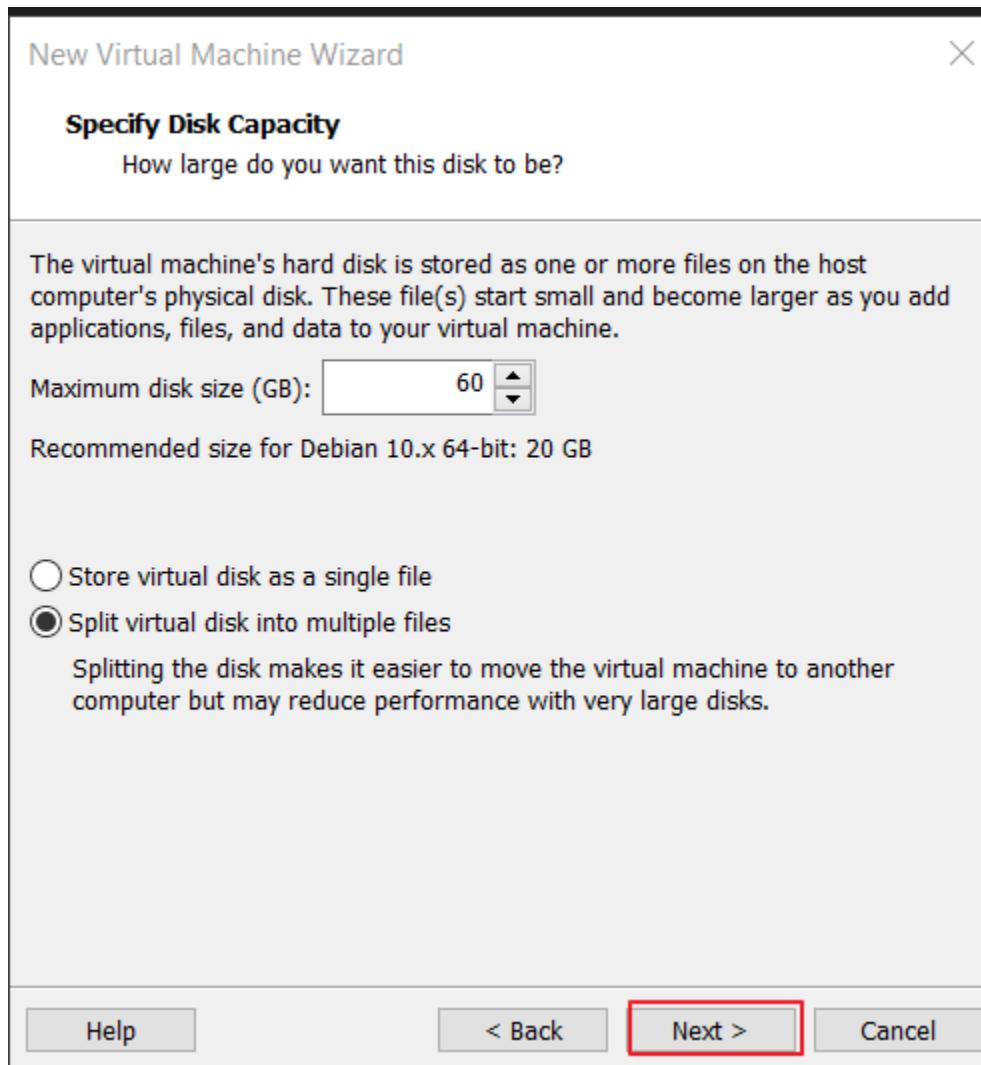


Figure 206: Installing Debian 10 on VM 4.

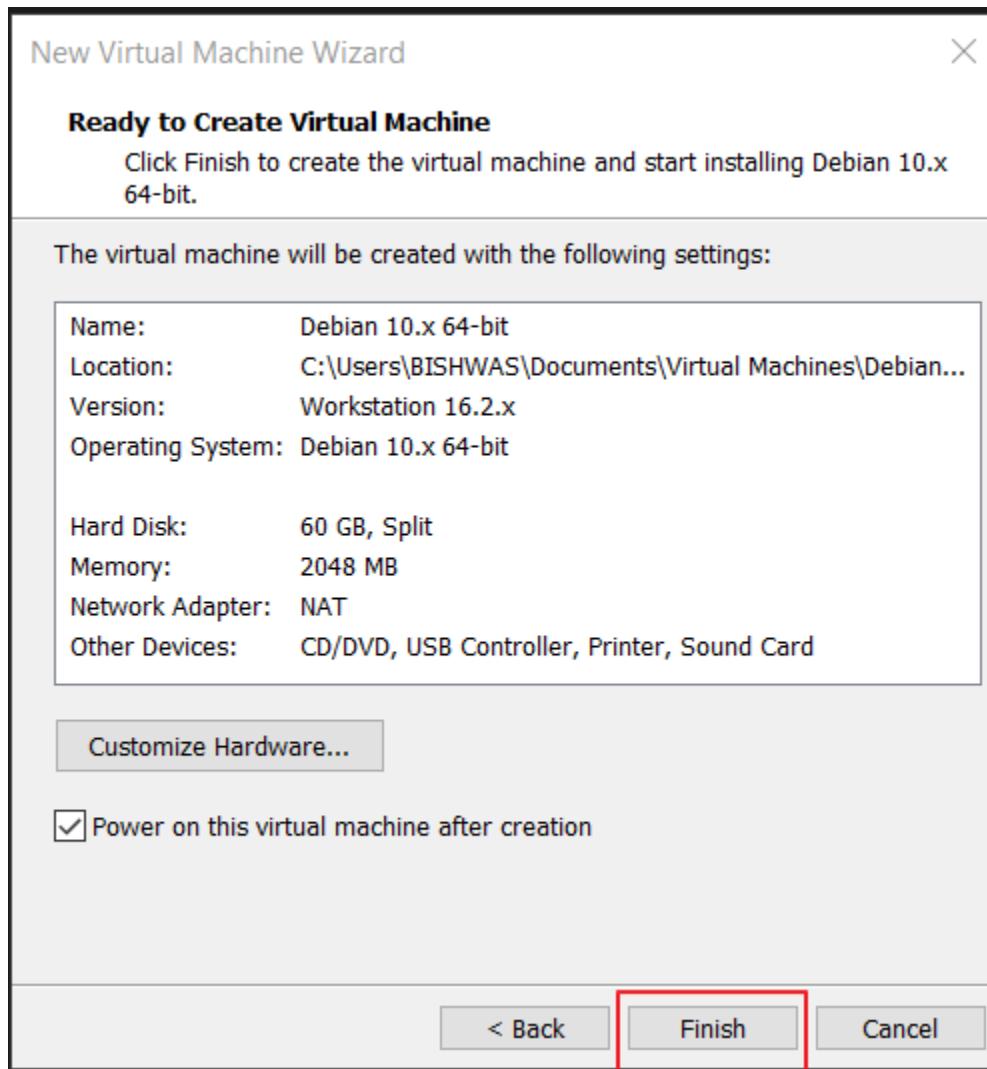


Figure 207: Installing Debian 10 on VM 5.

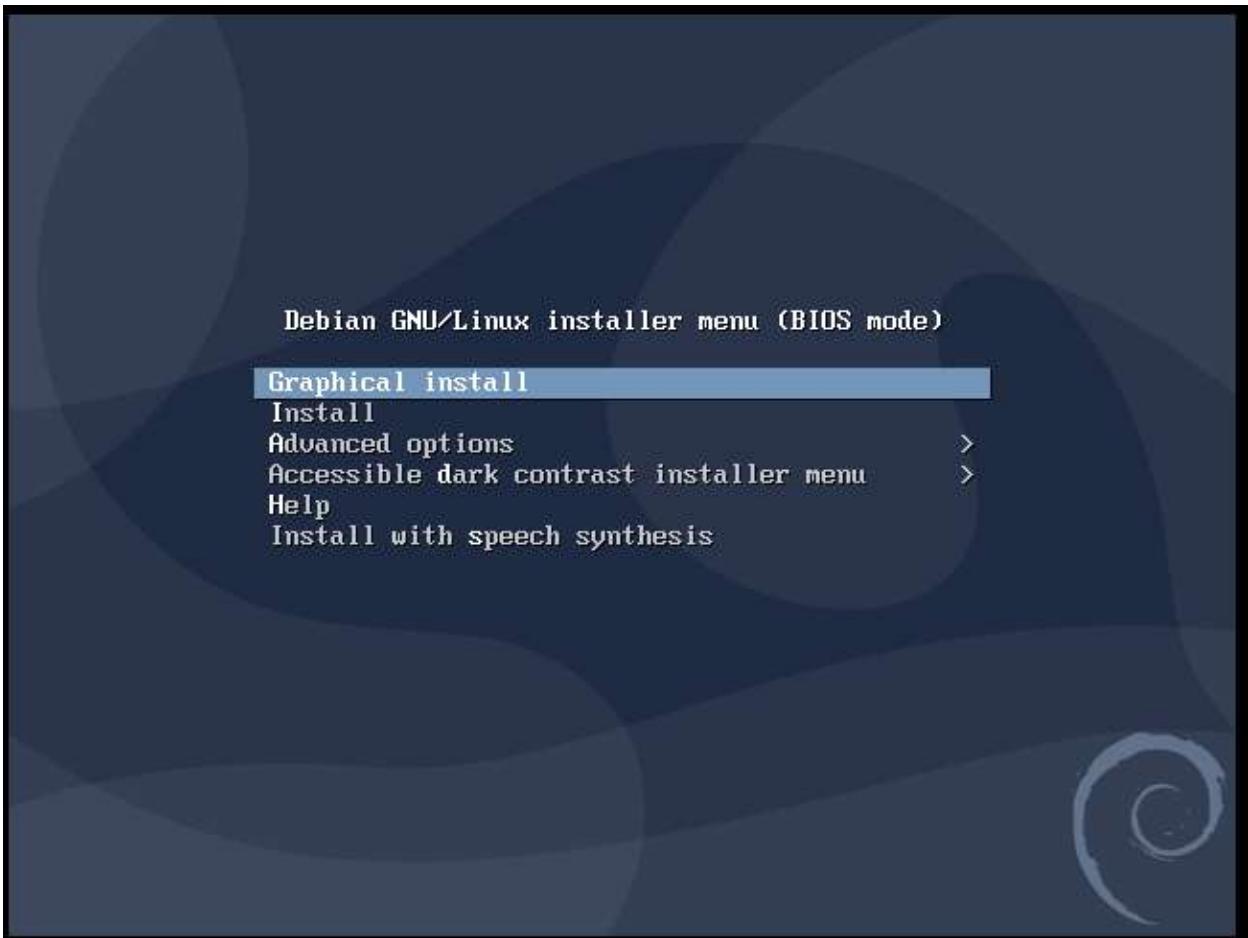


Figure 208: Installing Debian 10 on VM 6.

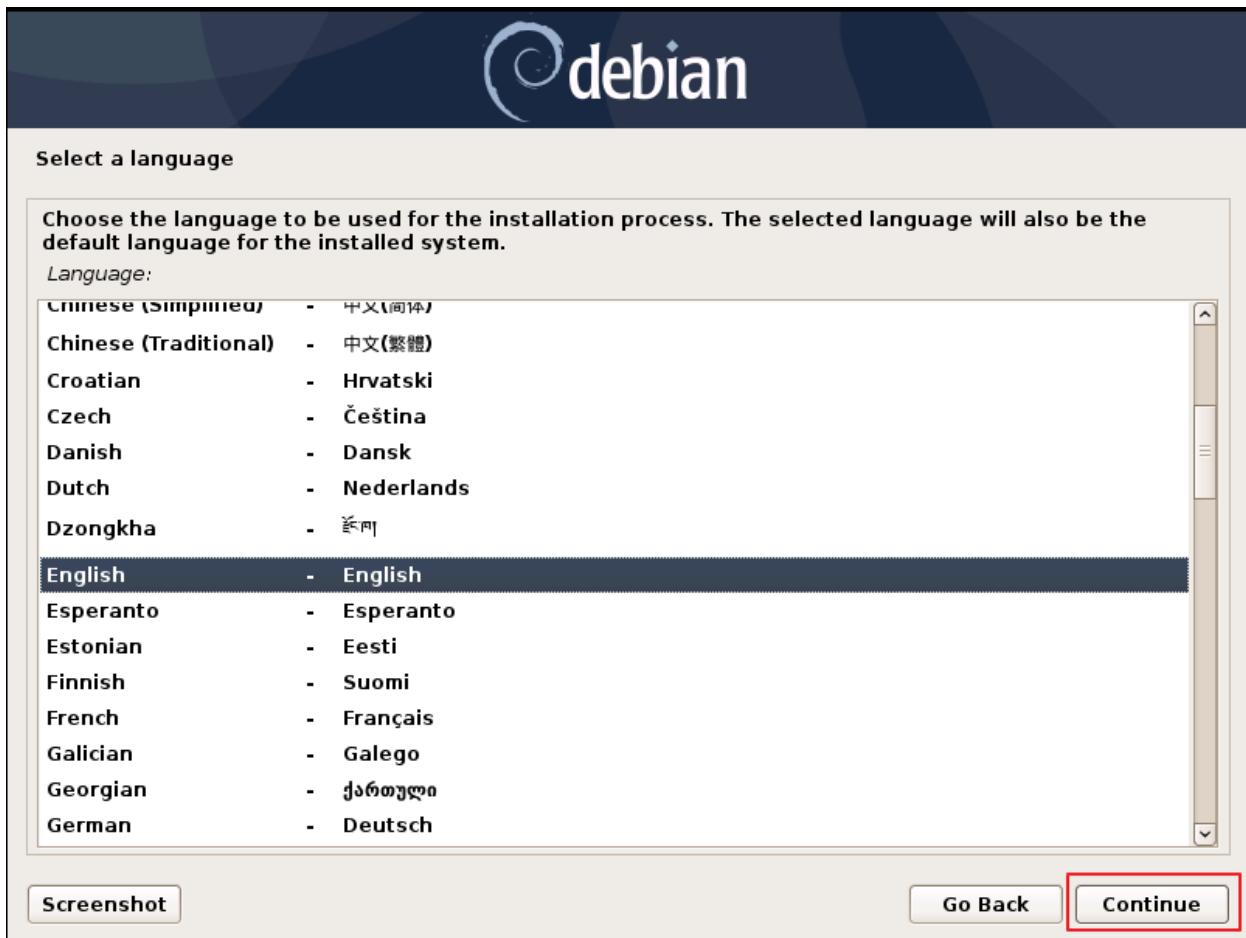


Figure 209: Installing Debian 10 on VM 7.

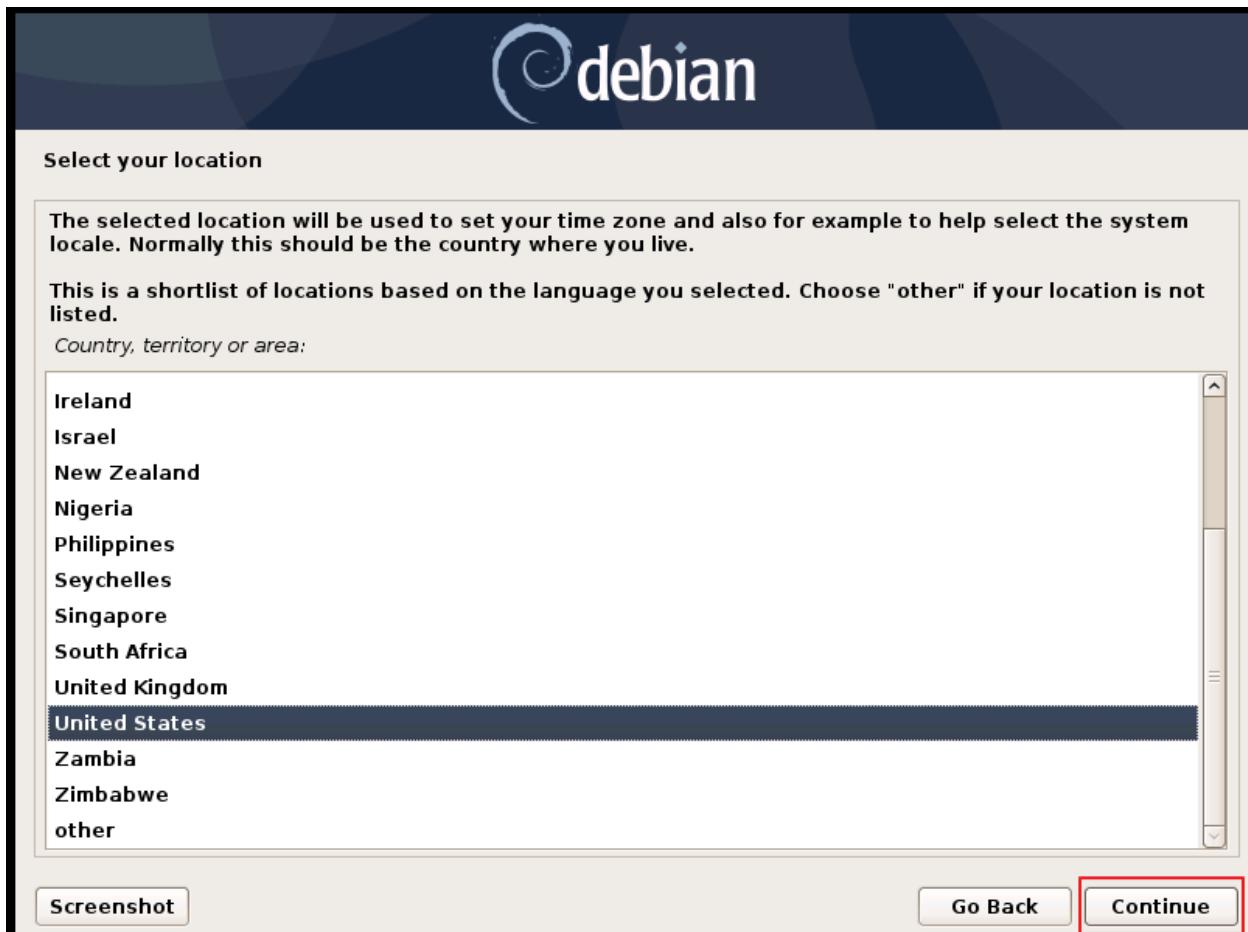


Figure 210: Installing Debian 10 on VM 8.

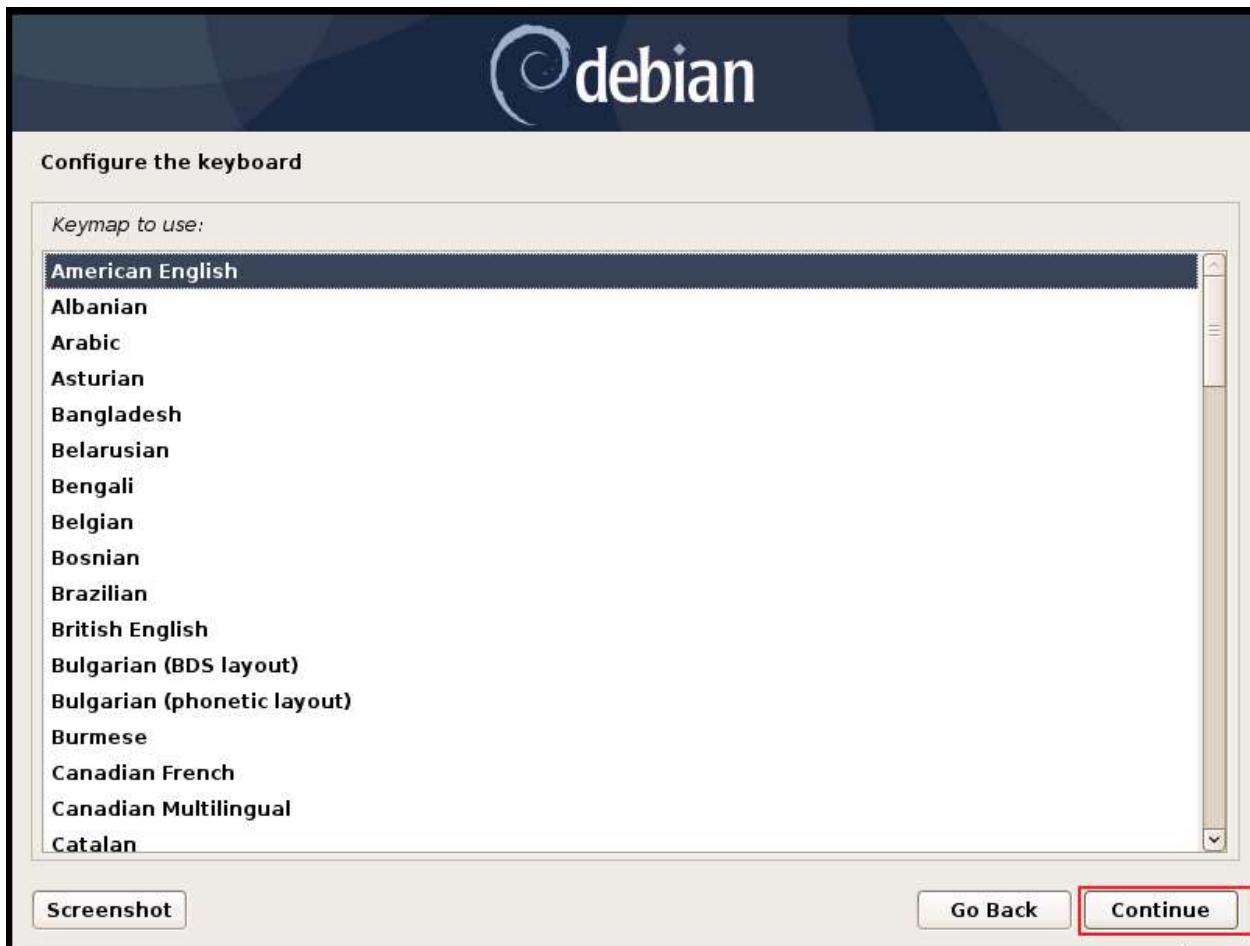


Figure 211: Installing Debian 10 on VM 9.



Figure 212: Installing Debian 10 on VM 10.



Figure 213: Installing Debian 10 on VM 11.



Figure 214: Installing Debian 10 on VM 12.

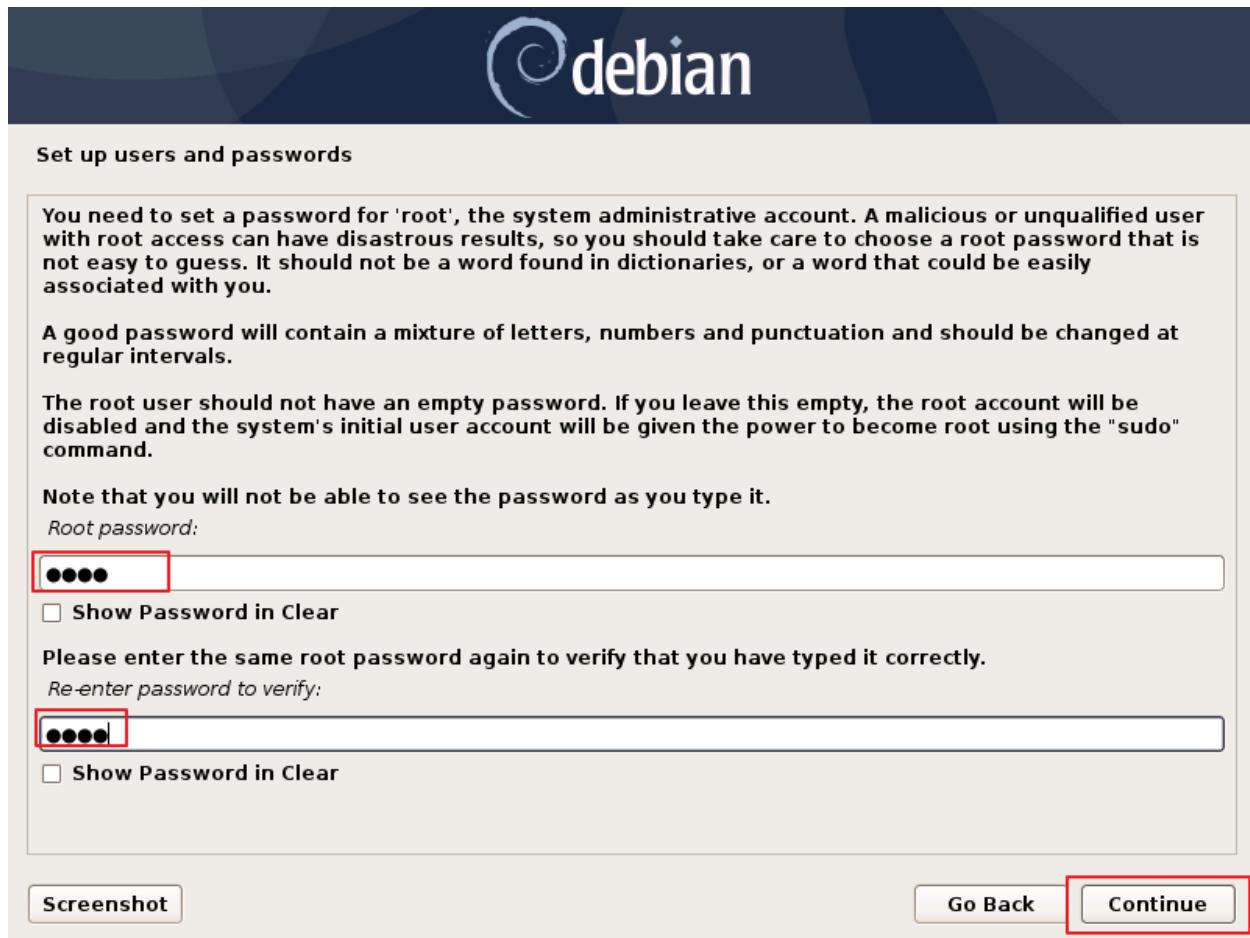


Figure 215: Installing Debian 10 on VM 13.



Figure 216: Installing Debian 10 on VM 14.



Figure 217: Installing Debian 10 on VM 15.

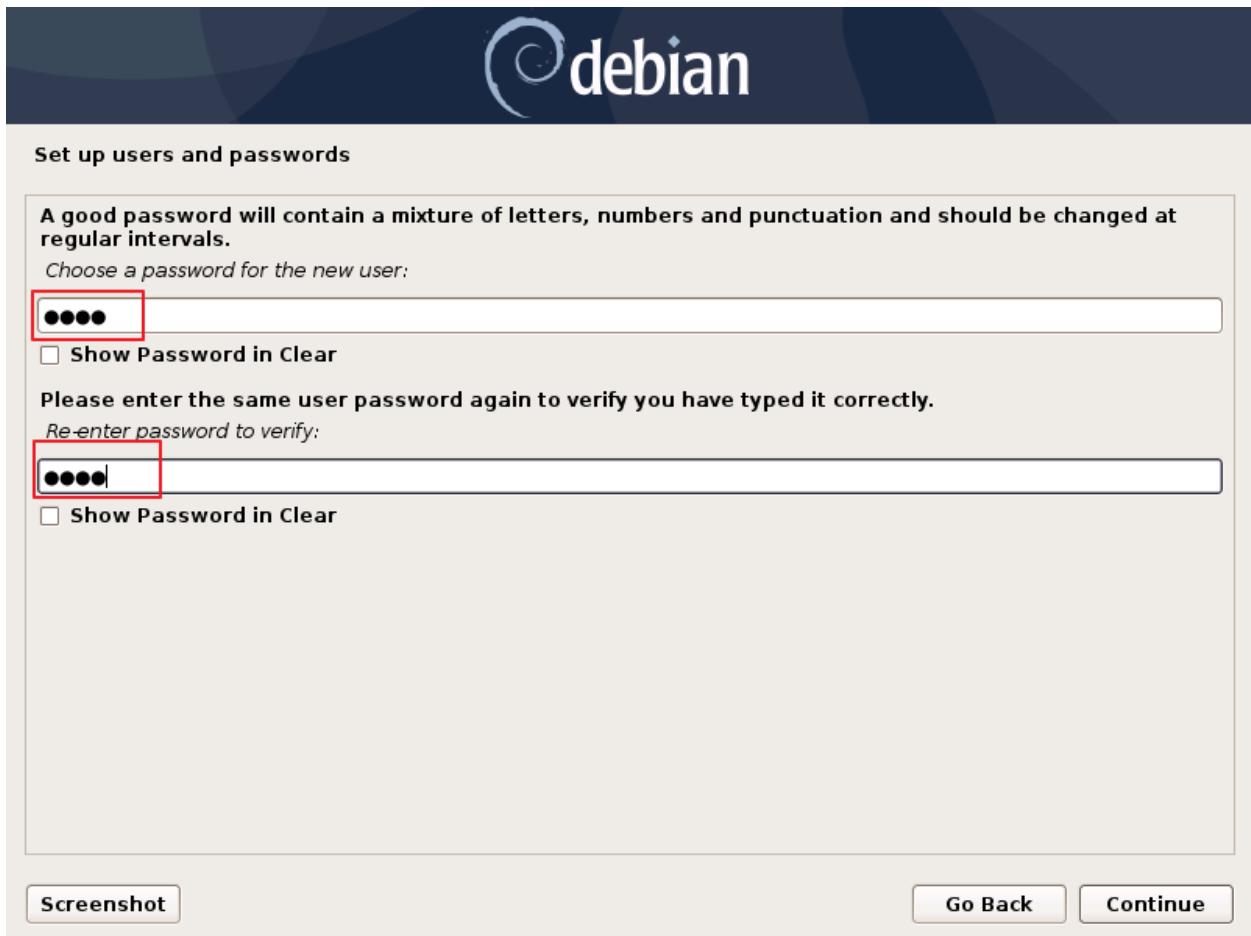


Figure 218: Installing Debian 10 on VM 16.



Figure 219: Installing Debian 10 on VM 17.

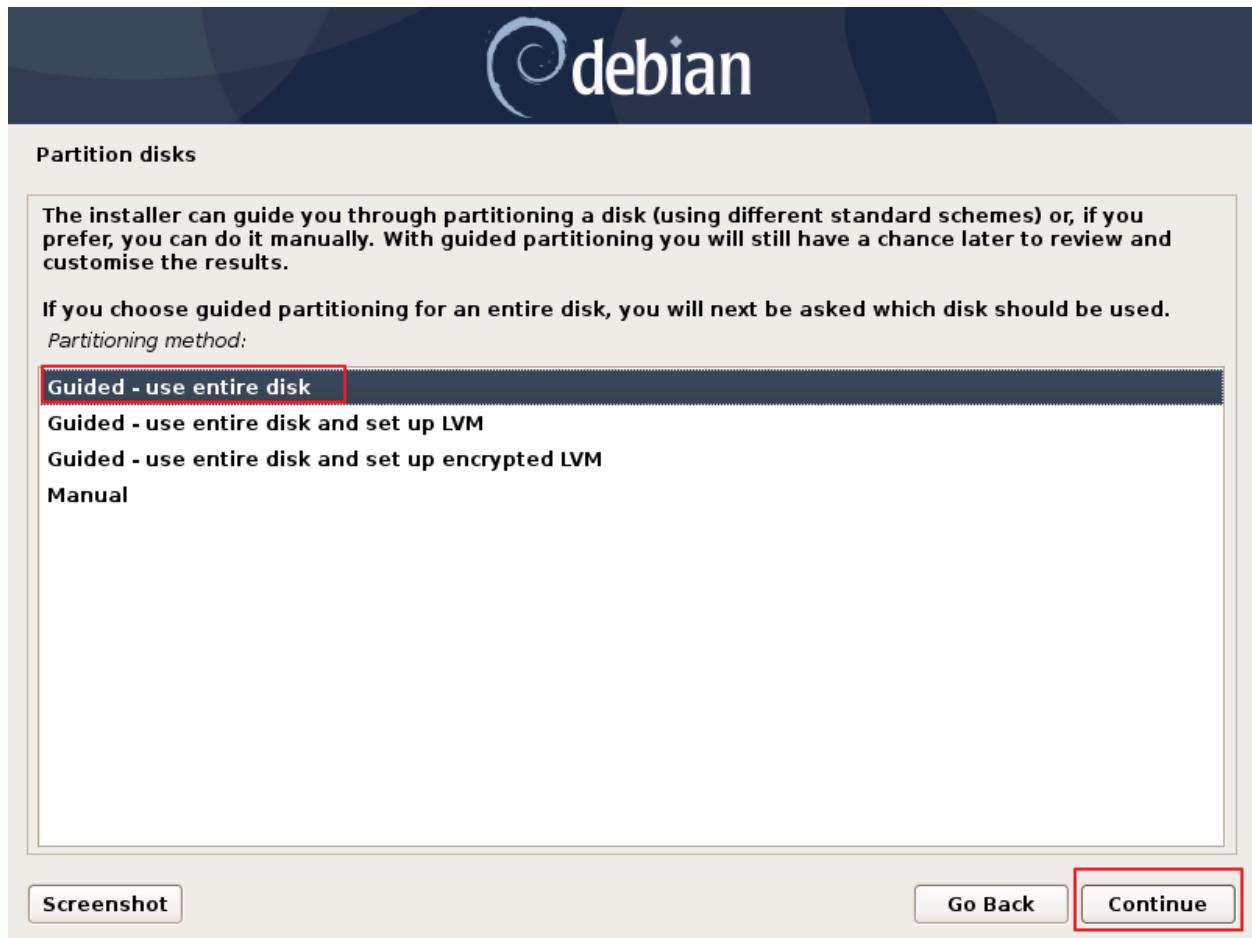


Figure 220: Installing Debian 10 on VM 18.



Figure 221: Installing Debian 10 on VM 19.

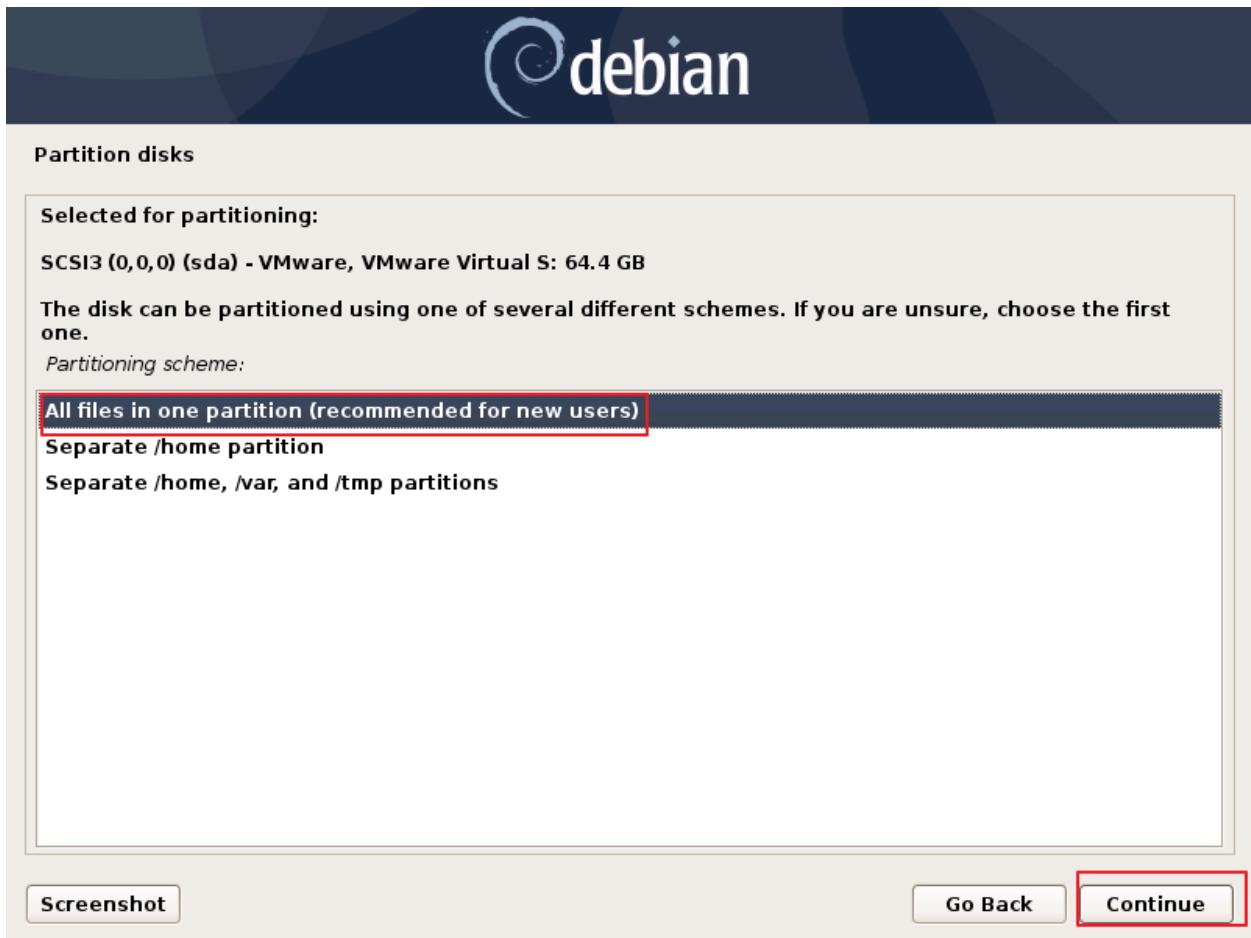


Figure 222: Installing Debian 10 on VM 20.



Figure 223: Installing Debian 10 on VM 21.

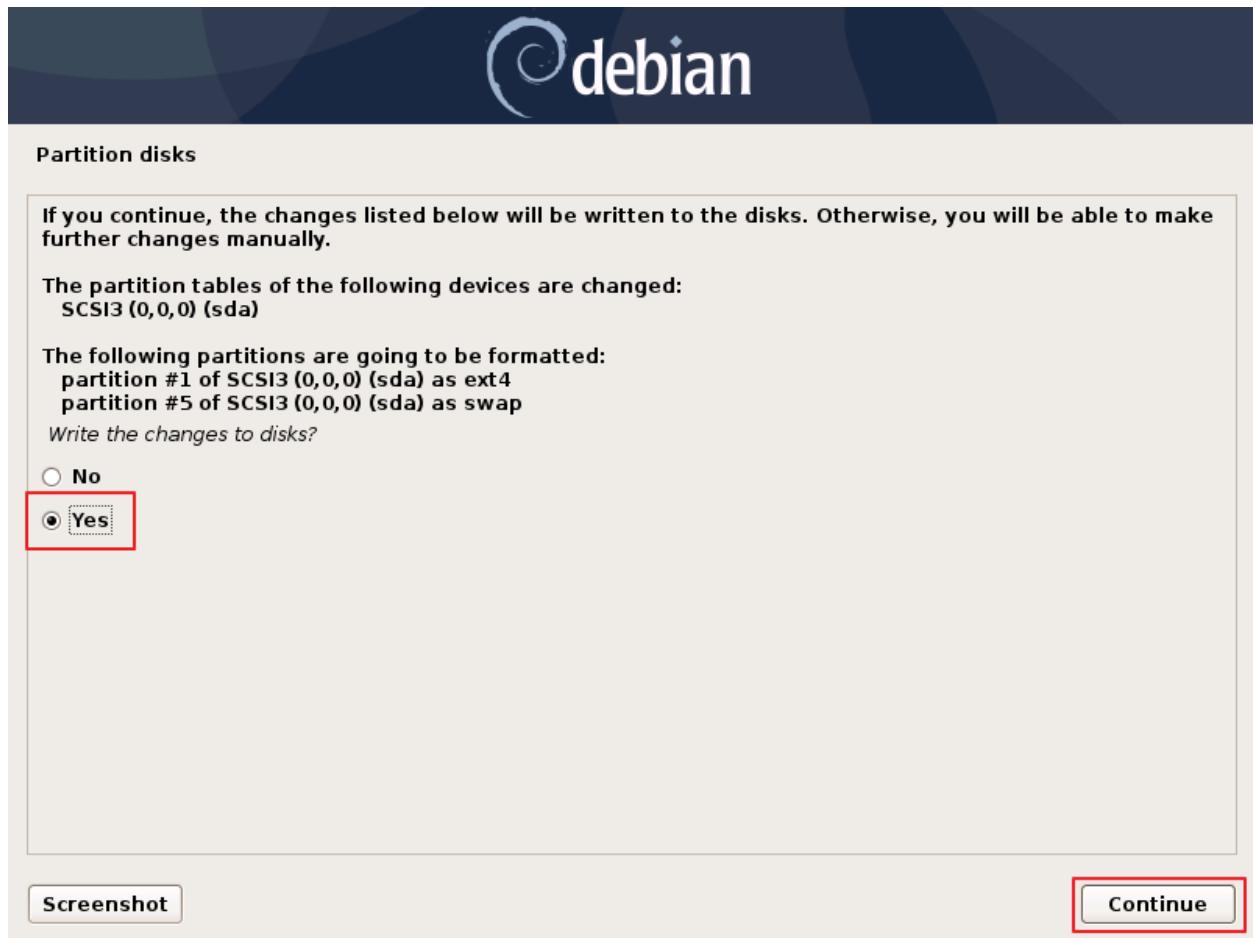


Figure 224: Installing Debian 10 on VM 22.



Figure 225: Installing Debian 10 on VM 23.

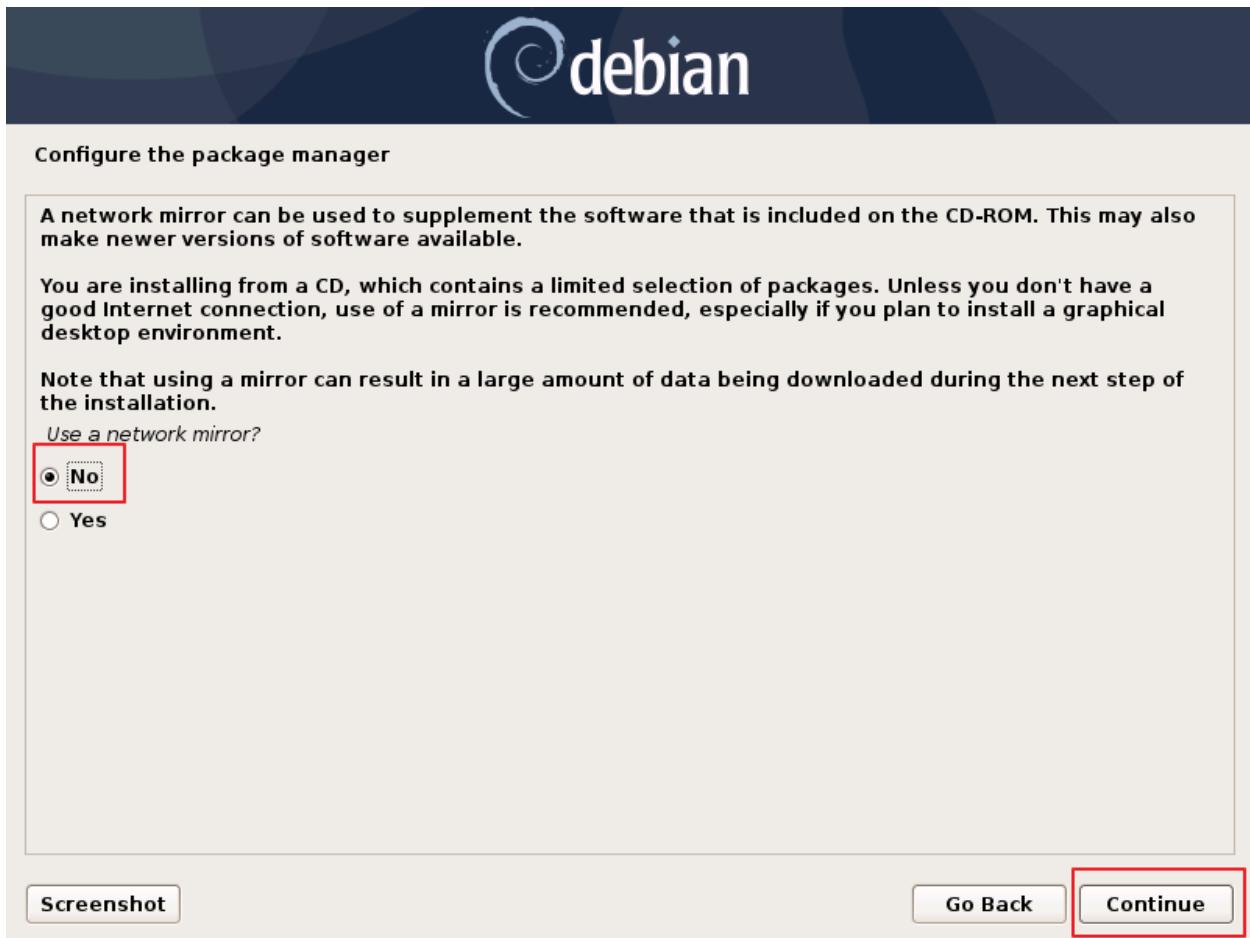


Figure 226: Installing Debian 10 on VM 24.



Figure 227: Installing Debian 10 on VM 25.

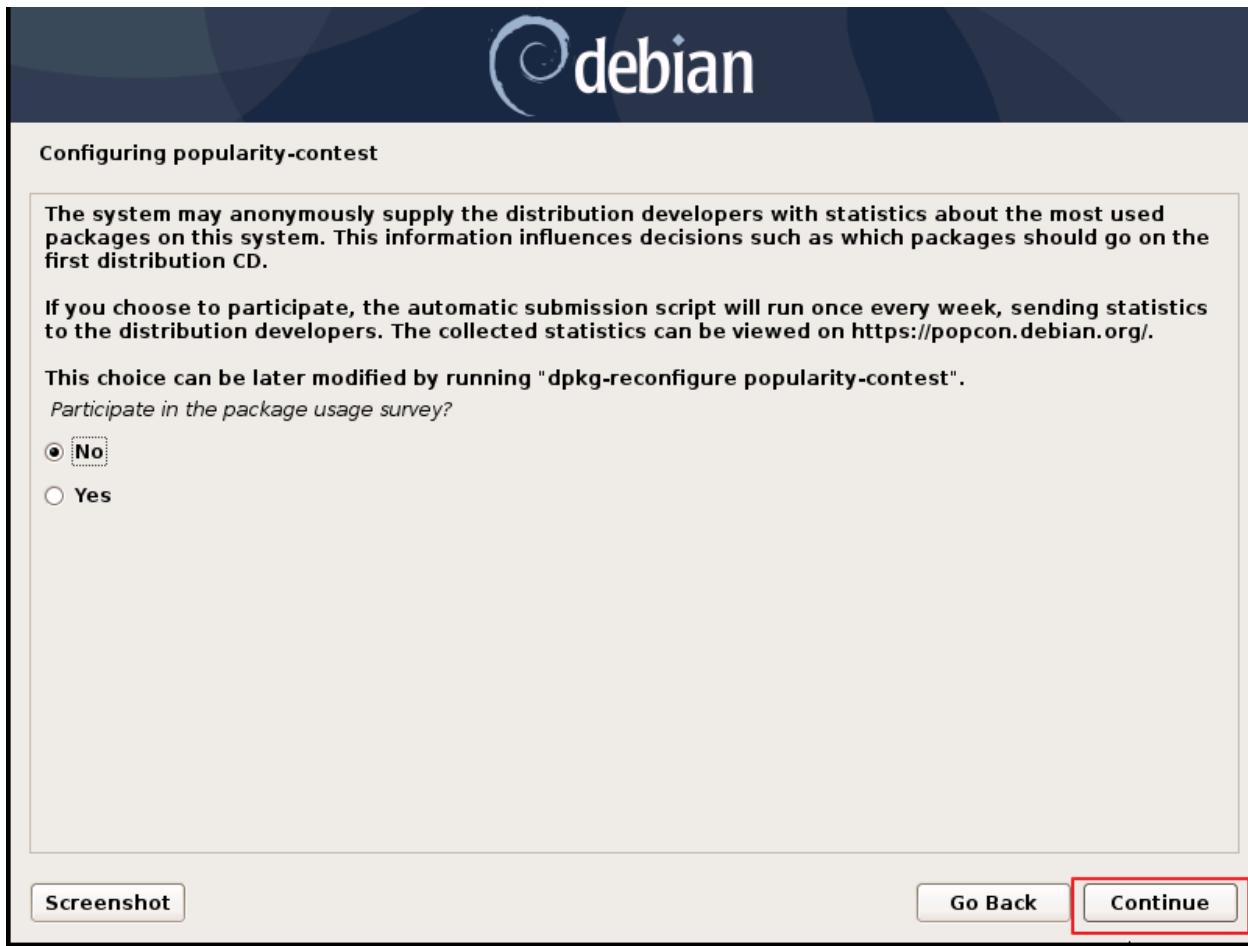


Figure 228: Installing Debian 10 on VM 26.

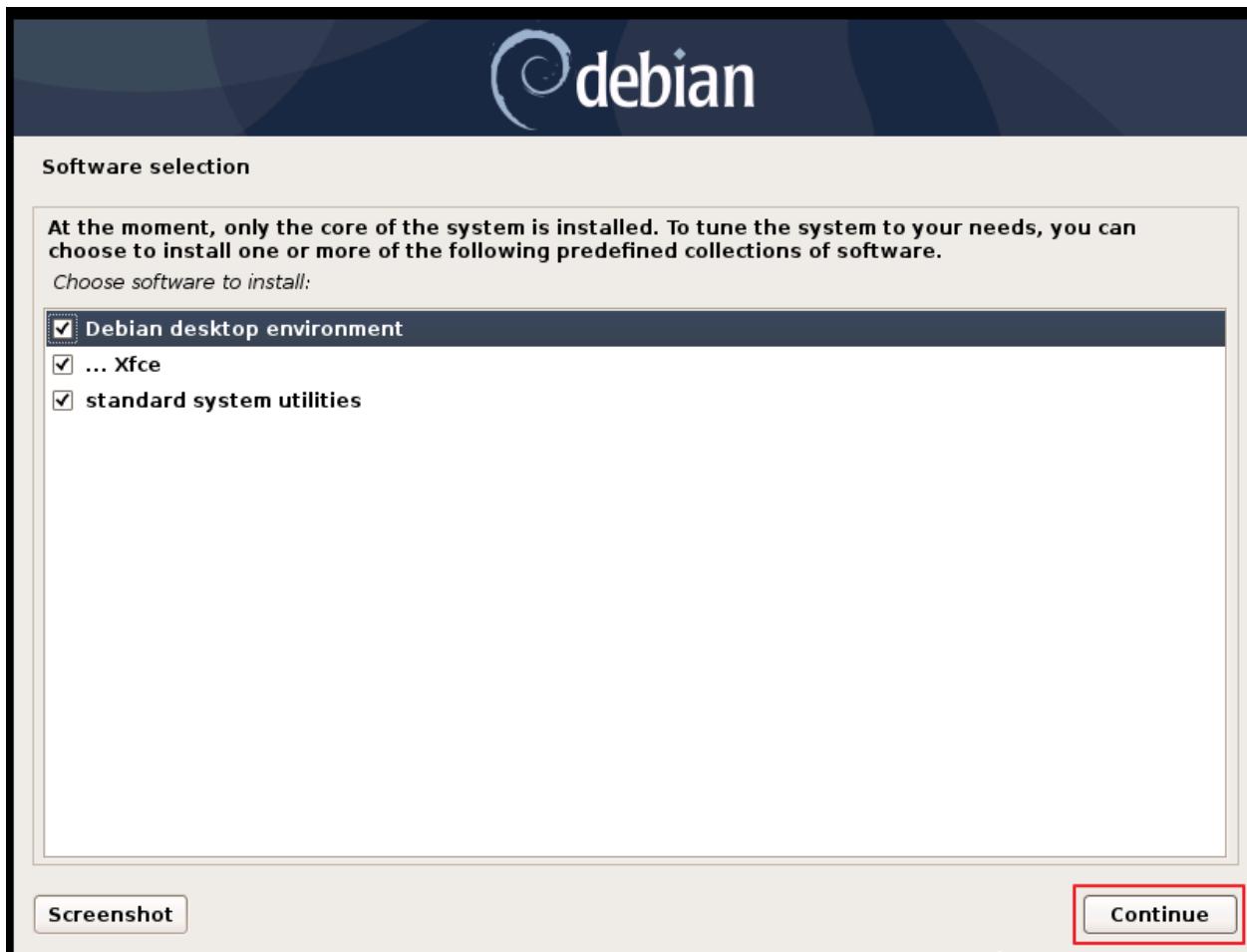


Figure 229: Installing Debian 10 on VM 27.



Figure 230: Installing Debian 10 on VM 28.



Figure 231: Installing Debian 10 on VM 29.



Figure 232: Installing Debian 10 on VM 30.



Figure 233. Installing Debian 10 on VM 31.

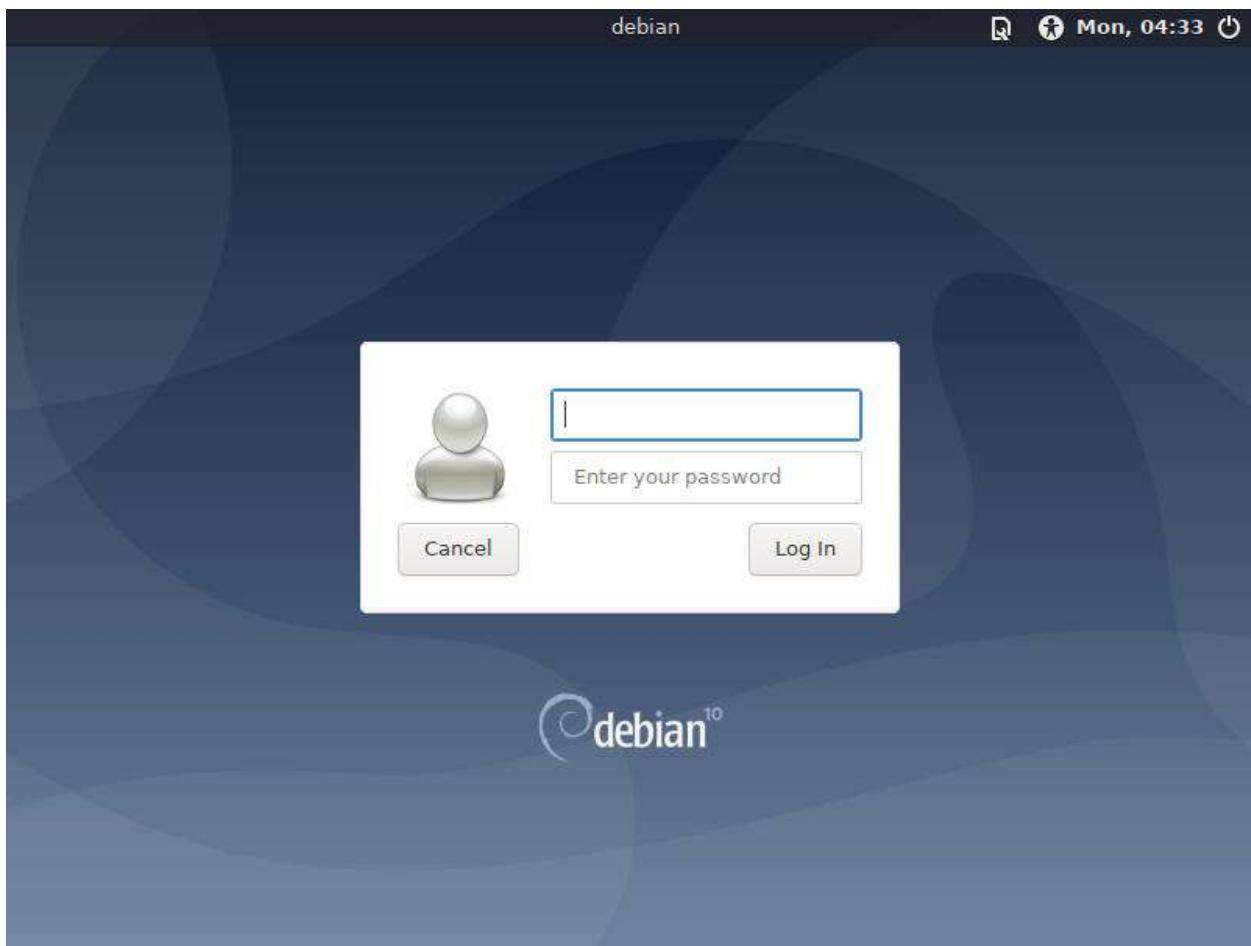


Figure 234: Installing Debian 10 on VM 32.

8.5.4 GNS3



Figure 235: Installing GNS3 1.

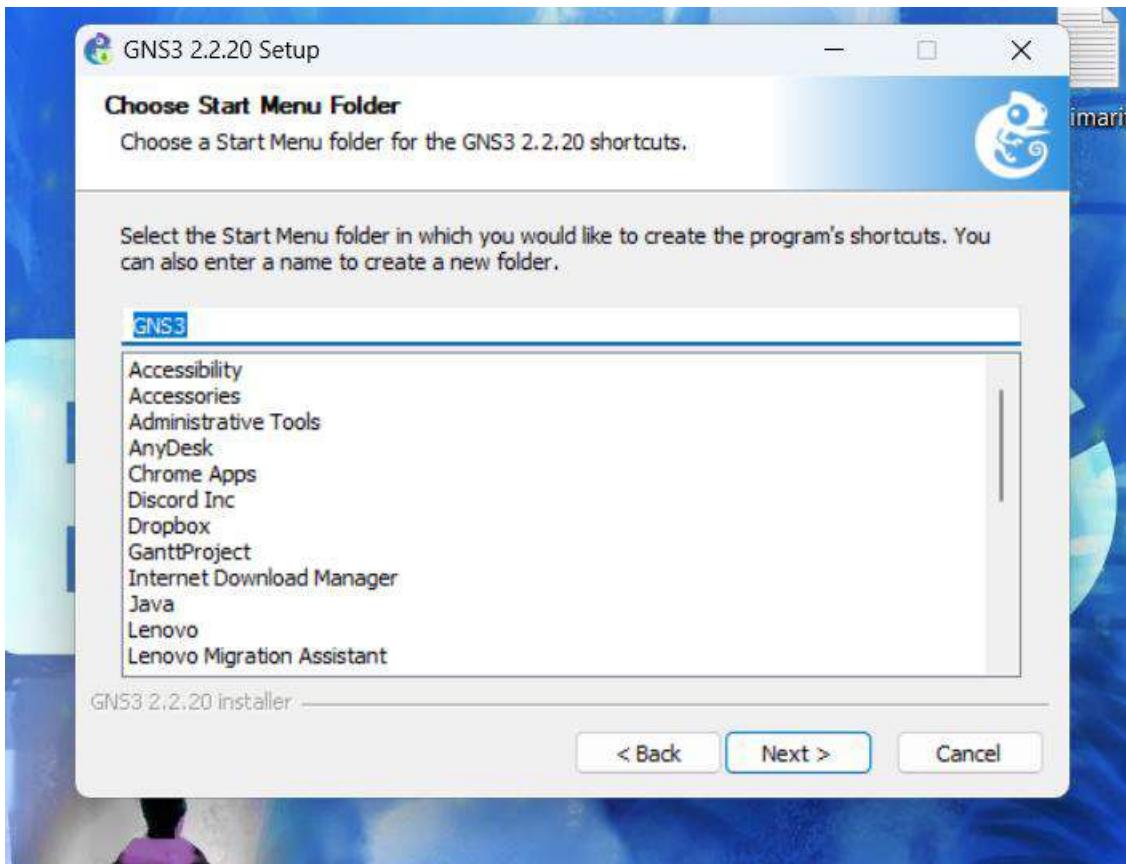


Figure 236: Installing GNS3 2.

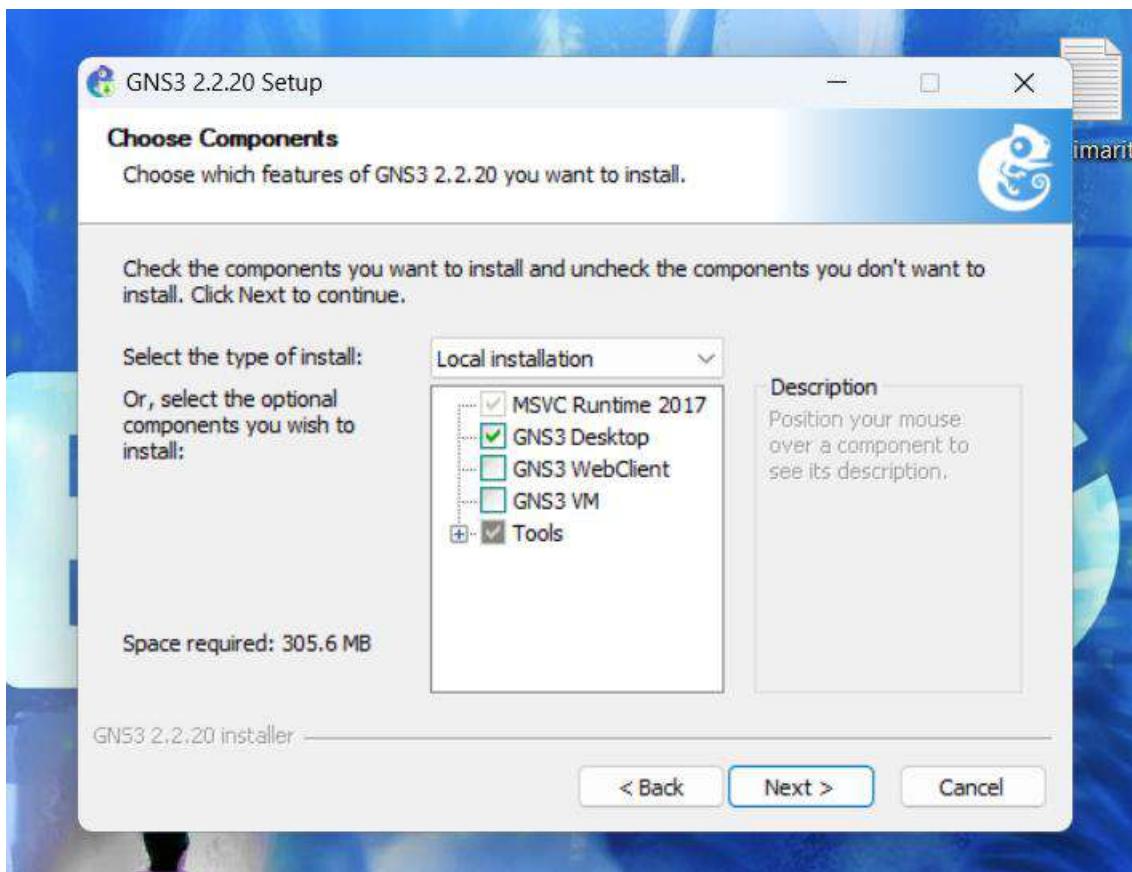


Figure 237: Installing GNS3 3.

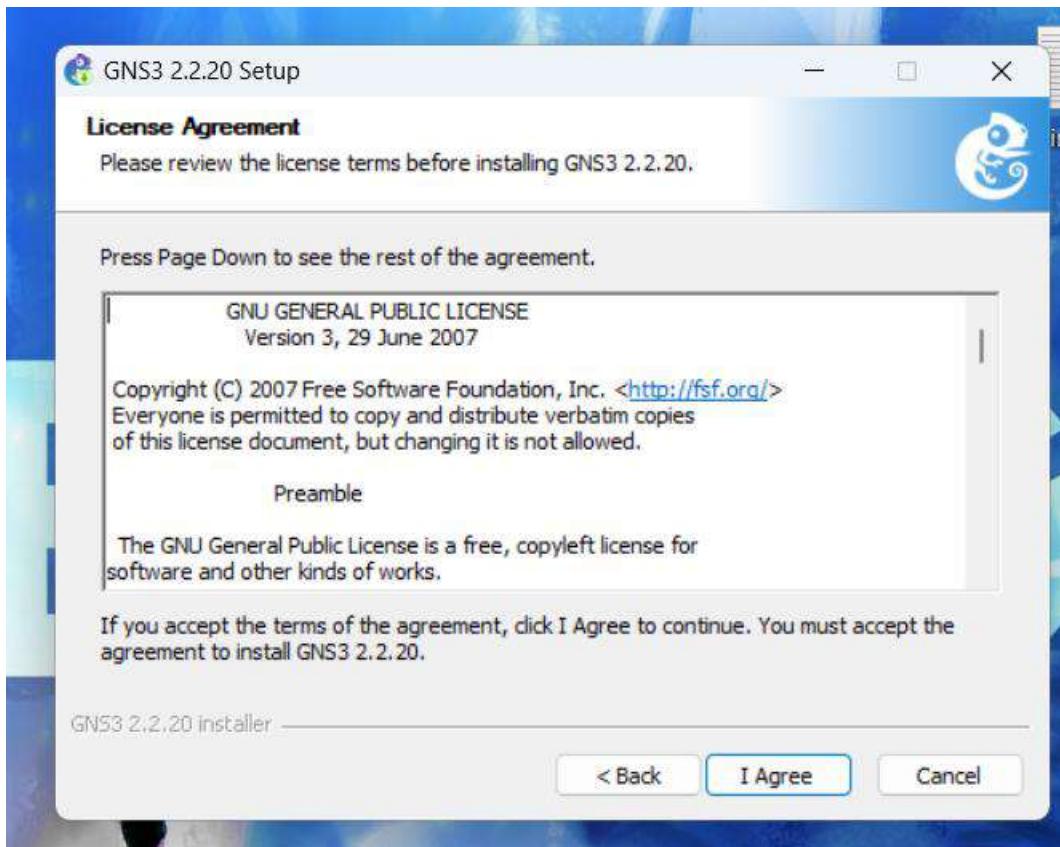


Figure 238: Installing GNS3 5.

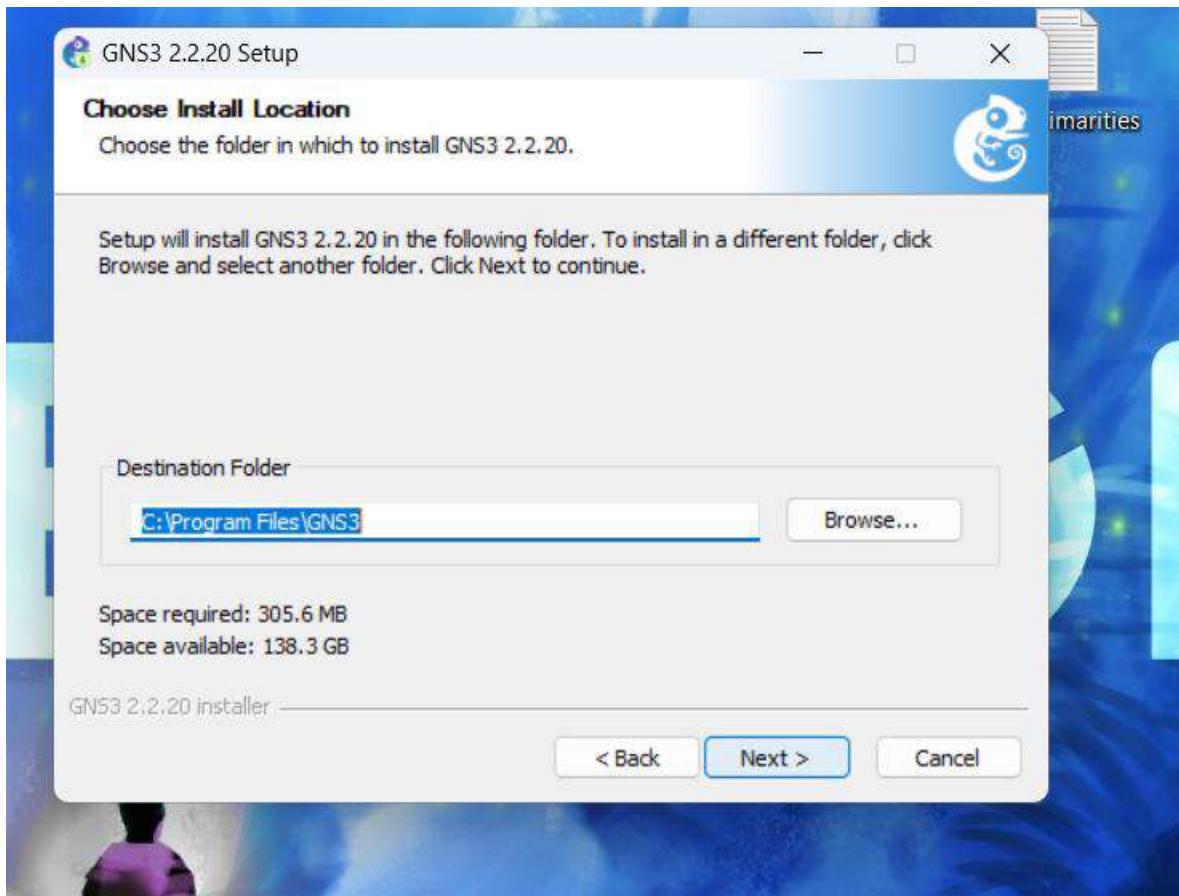


Figure 239: Installing GNS3 6.

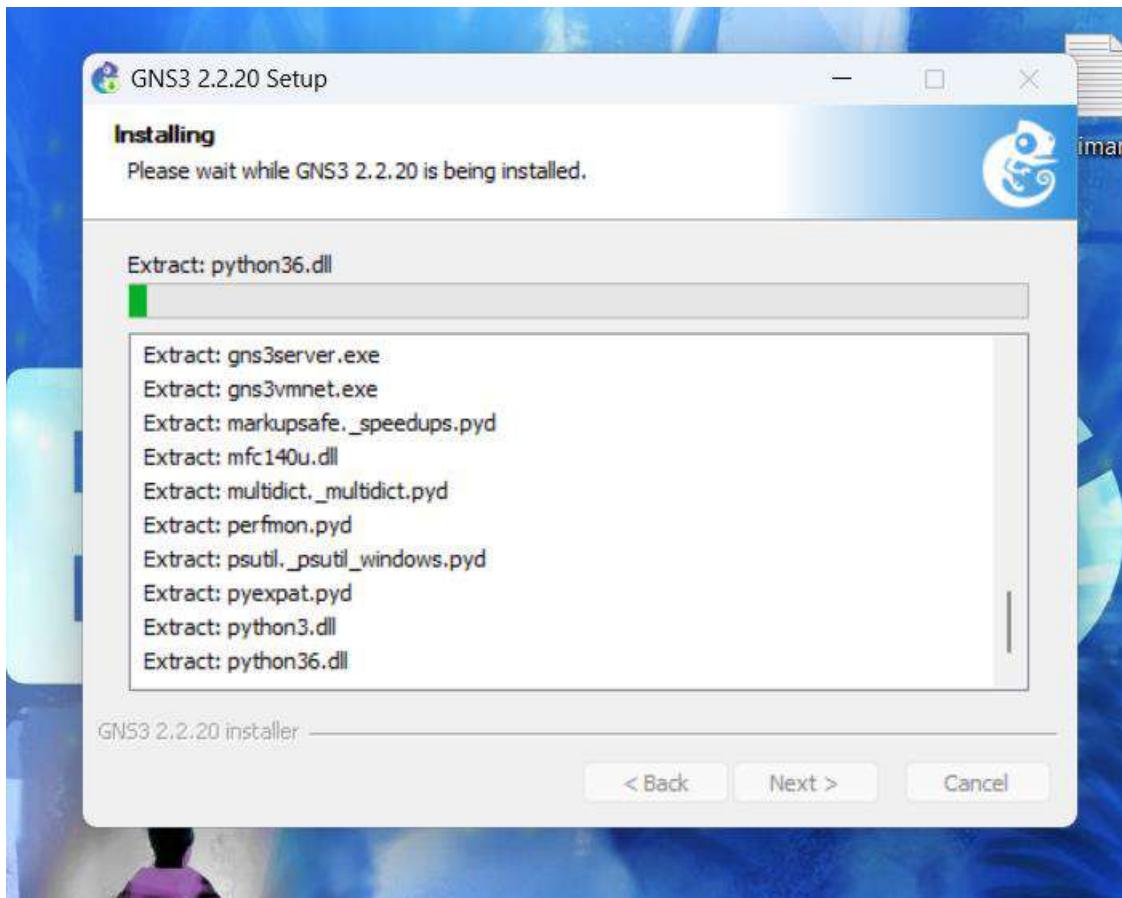


Figure 240: Installing GNS3 7.



Figure 241:Installing GNS3 8.

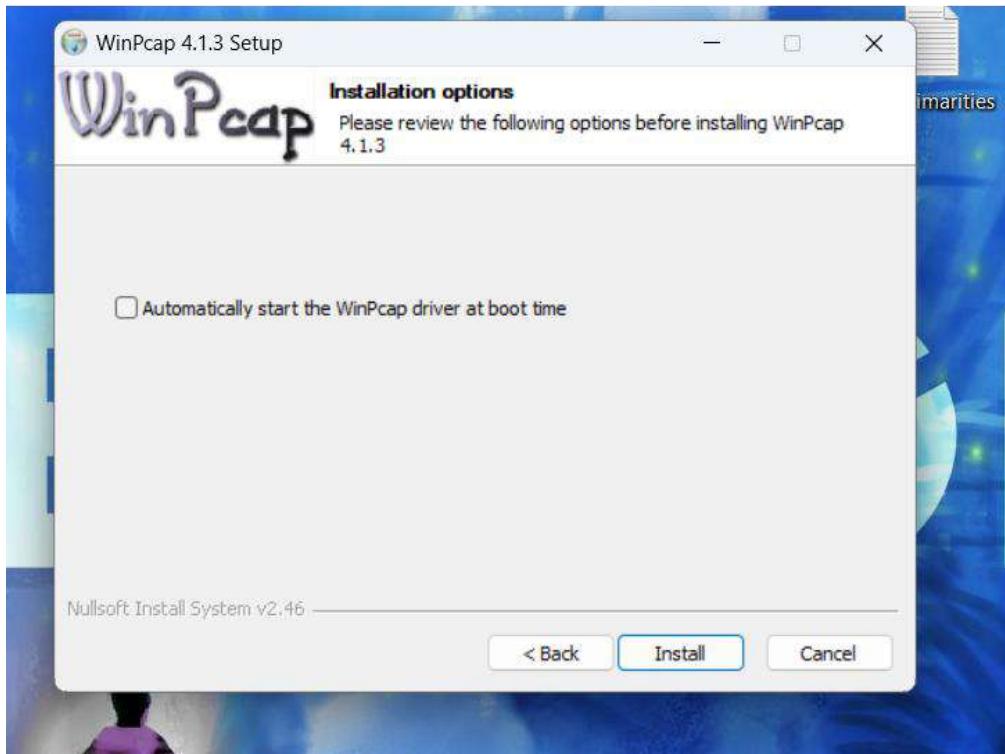


Figure 242: Installing GNS3 9.

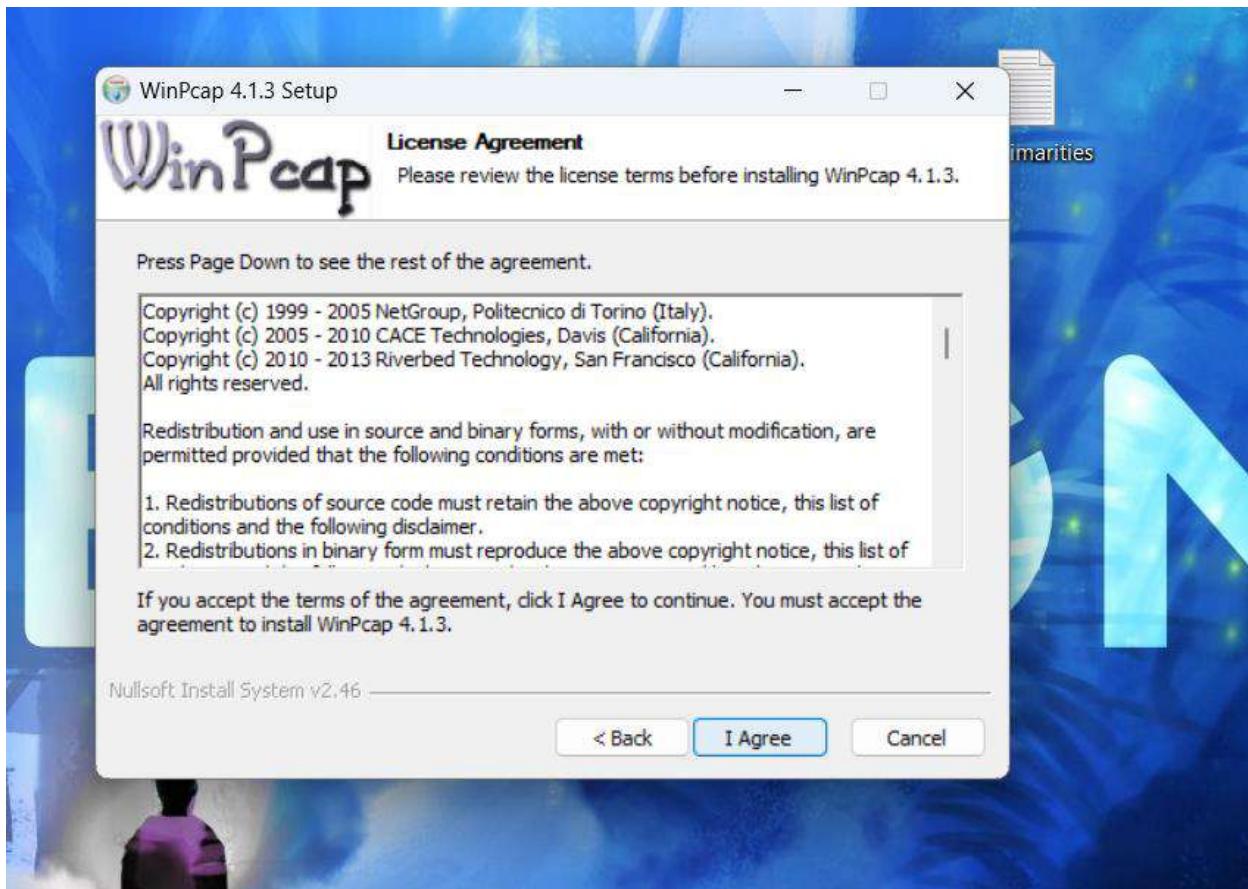


Figure 243: Installing GNS3 11.

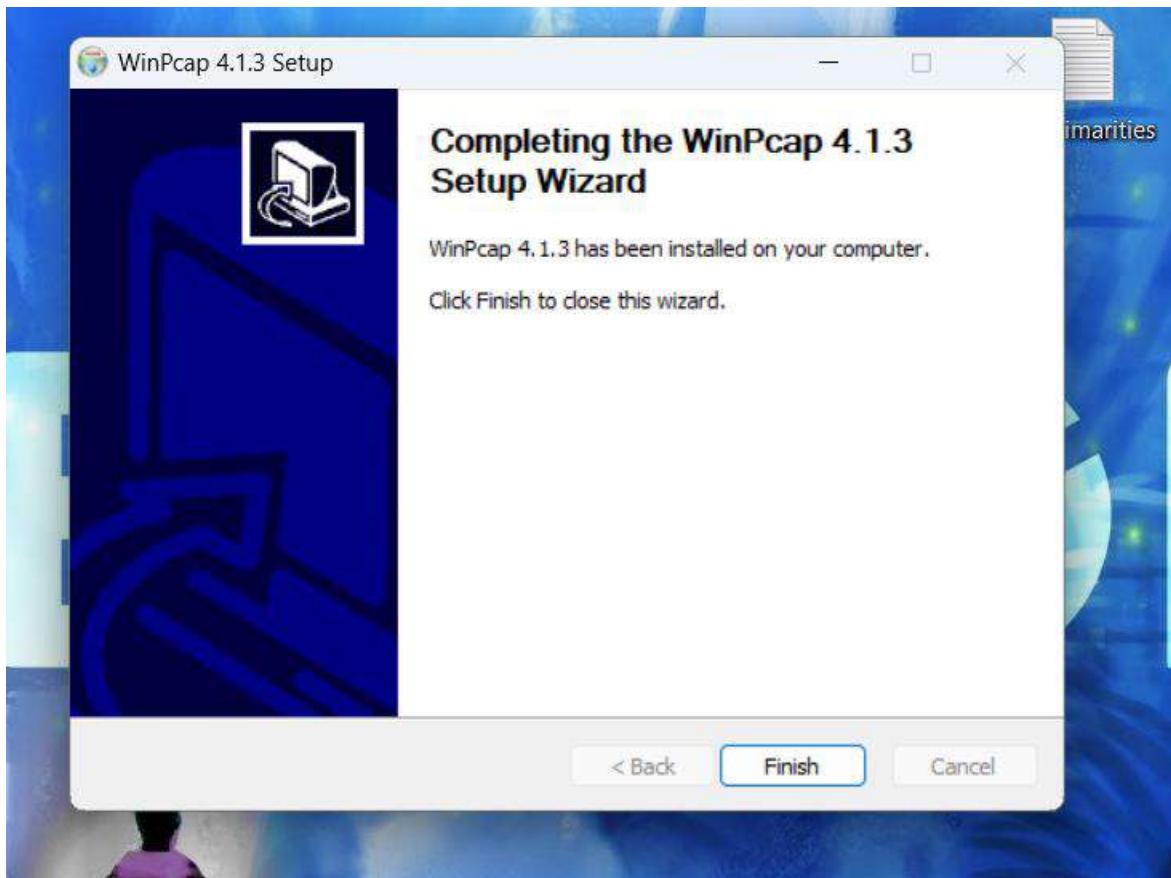


Figure 244: Installing GNS3 12.

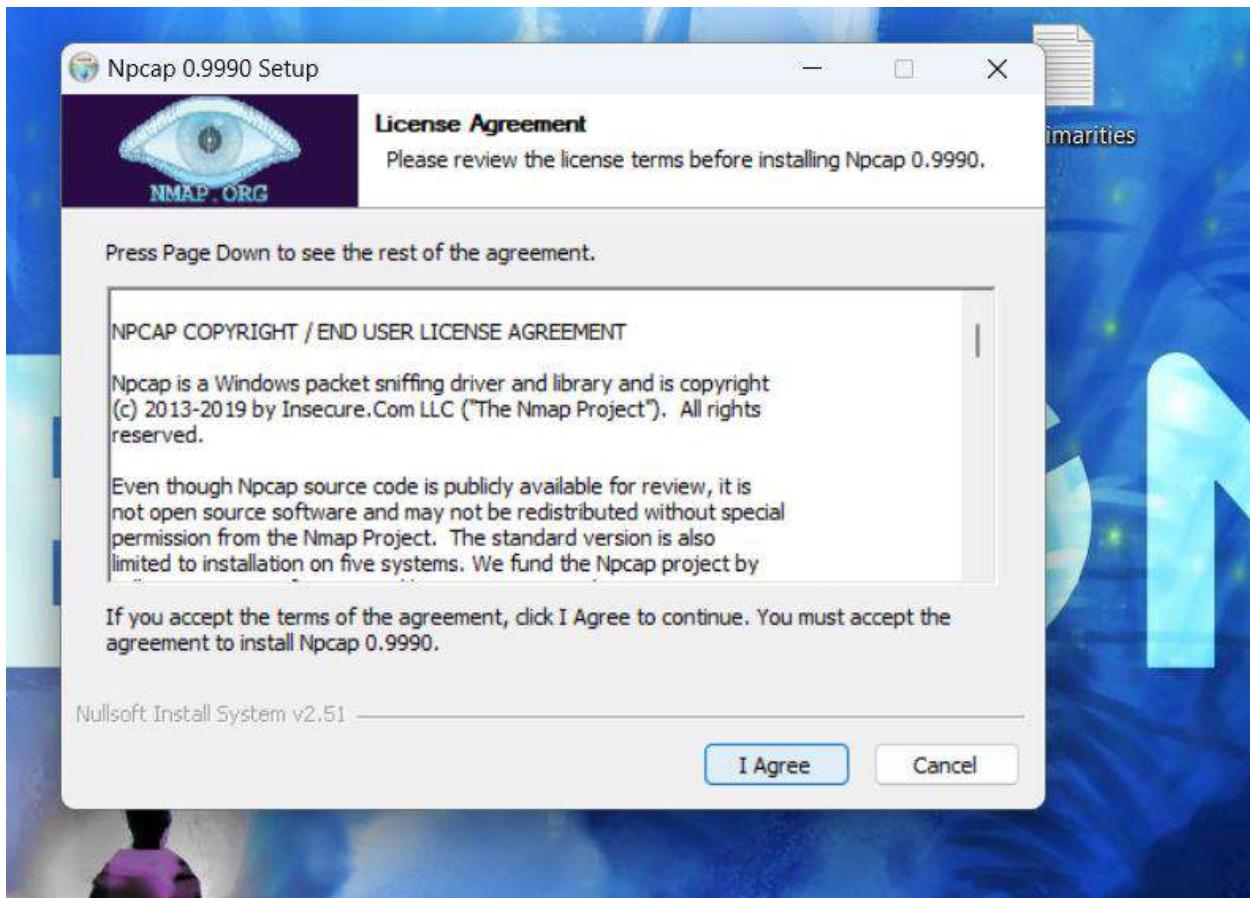


Figure 245: Installing GNS3 13.

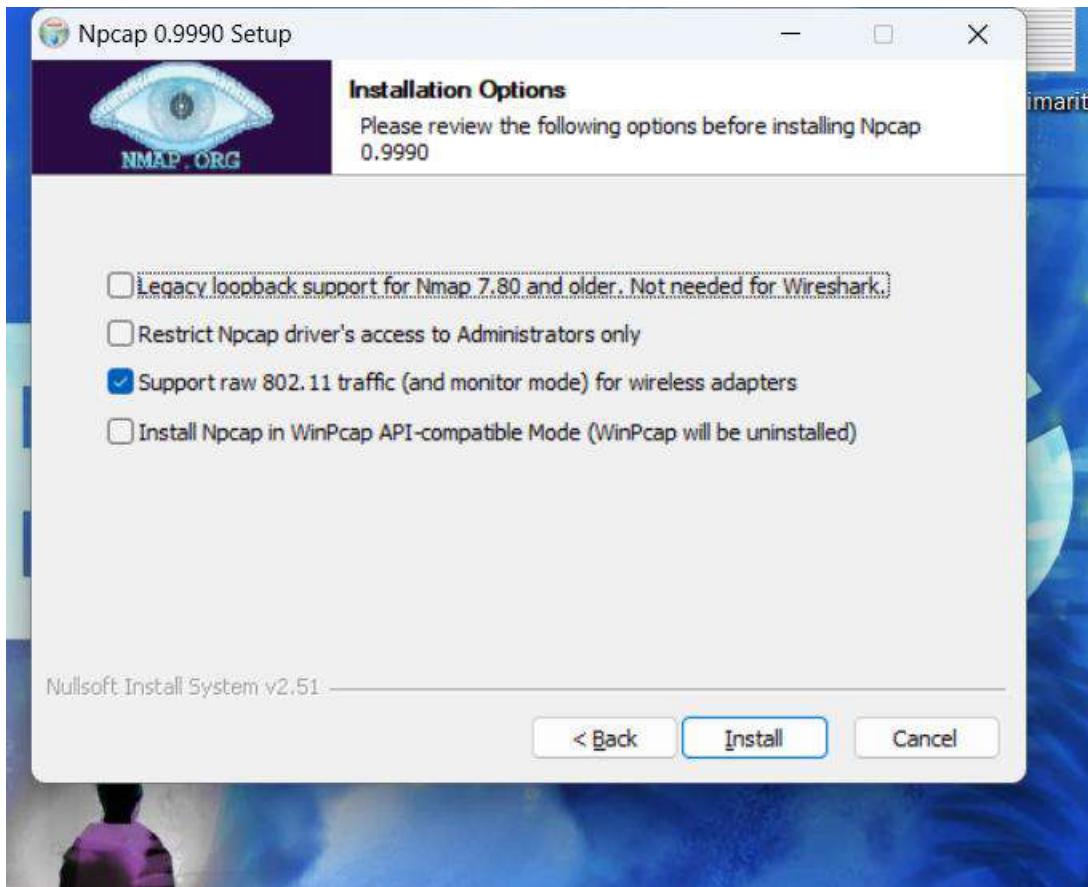


Figure 246: Installing GNS3 14.

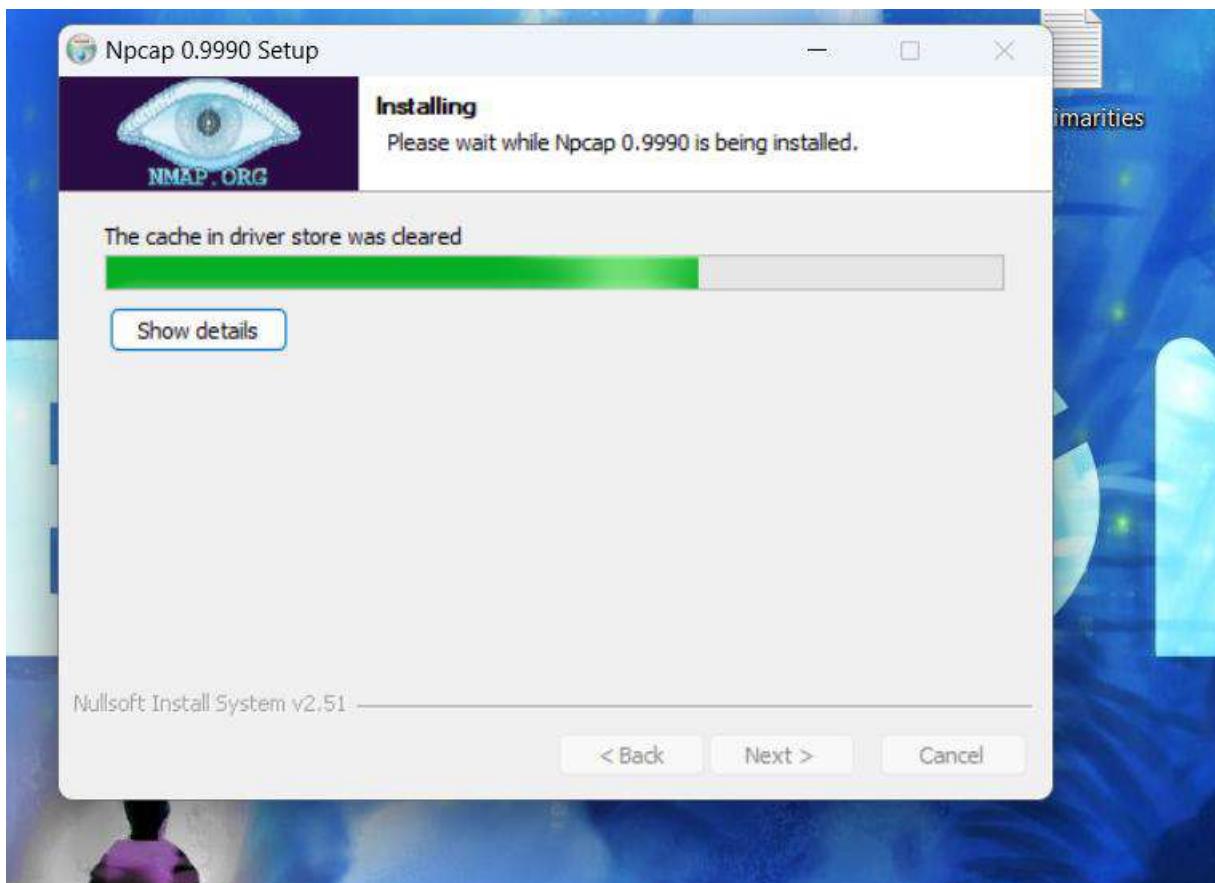


Figure 247: Installing GNS3 15.

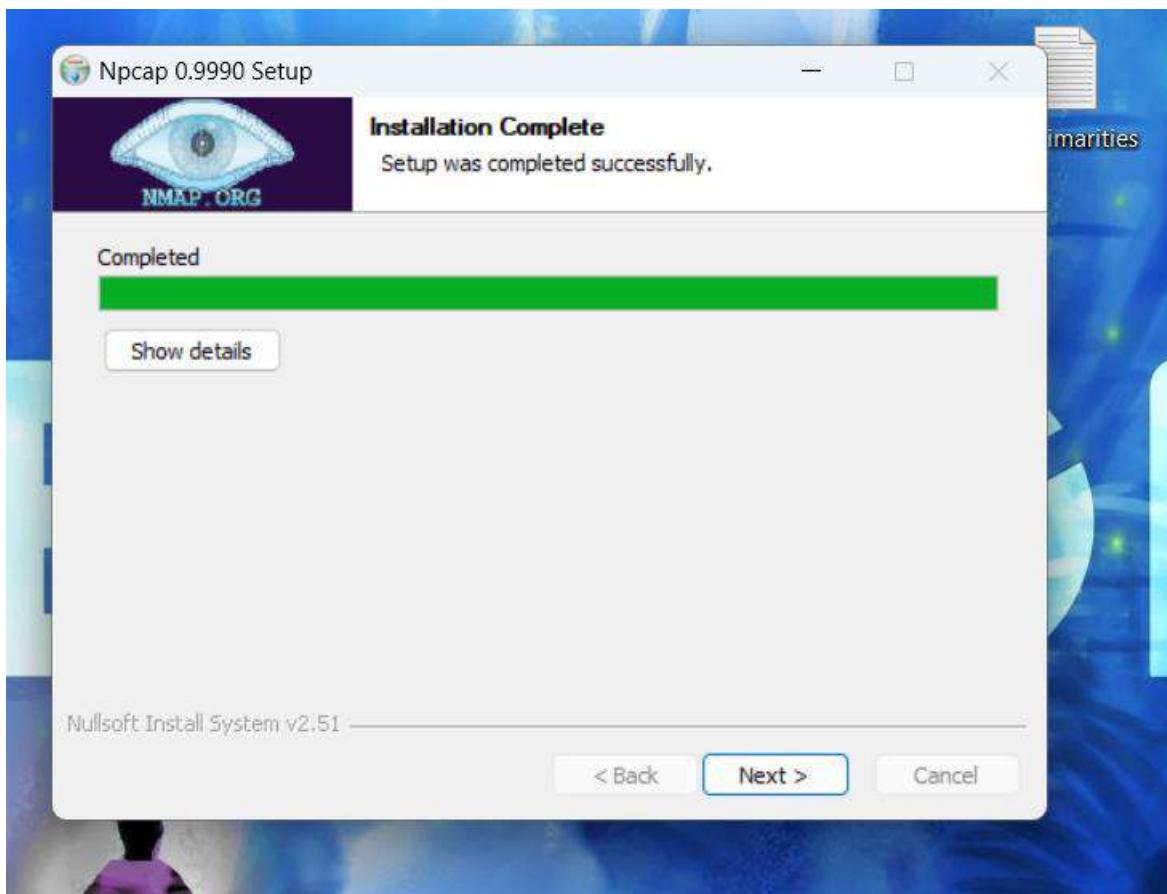


Figure 248: Installing GNS3 16.

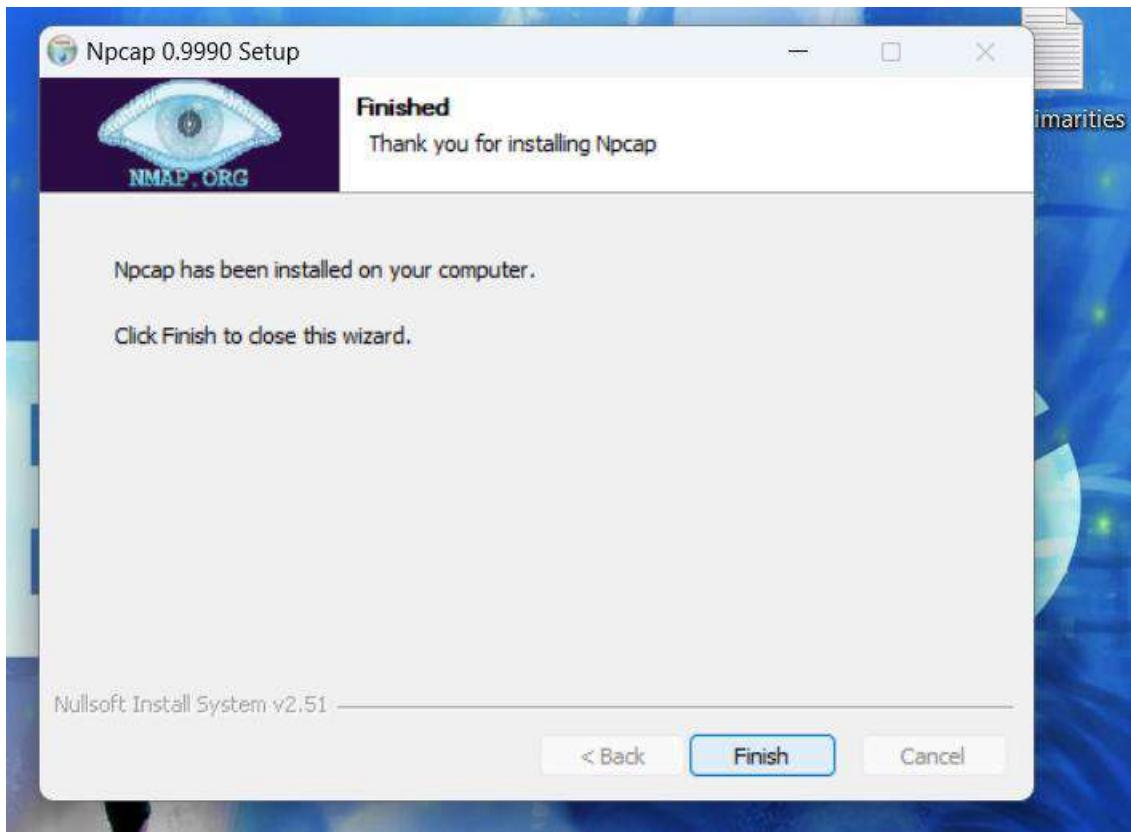


Figure 249: Installing GNS3 17.

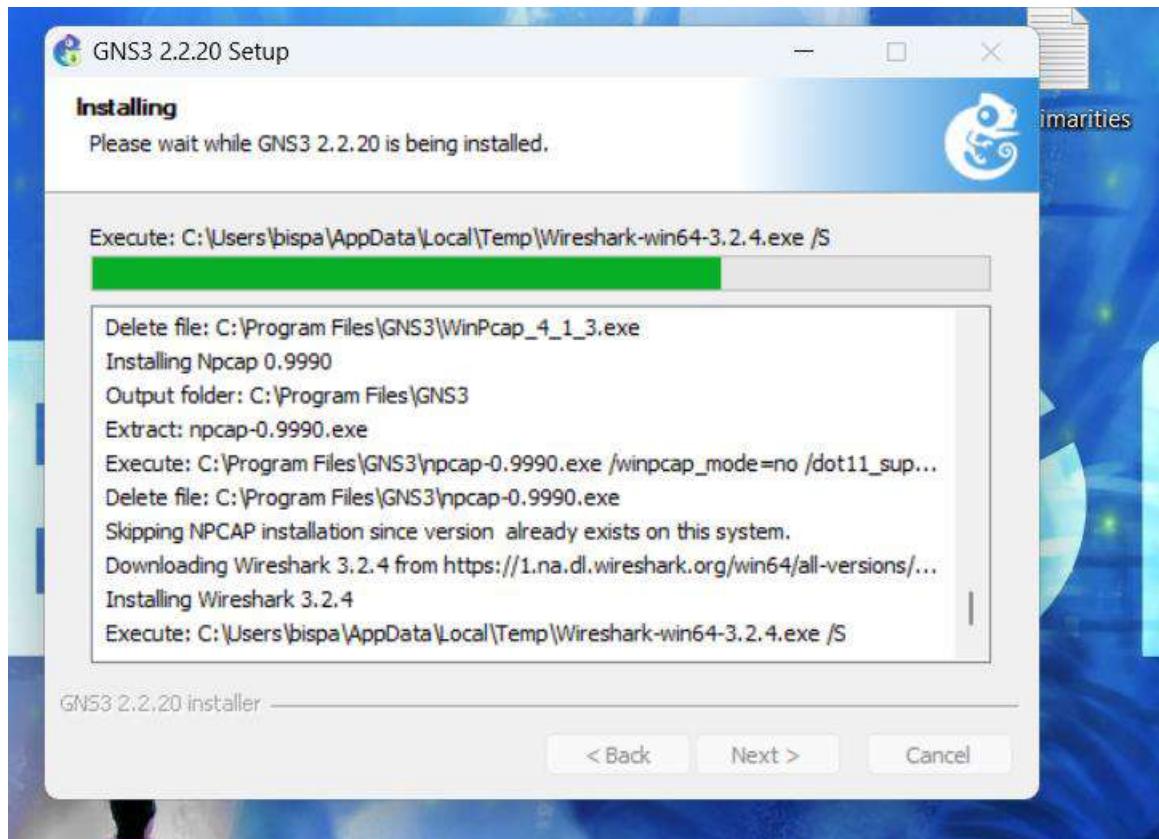


Figure 250: Installing GNS3 18.

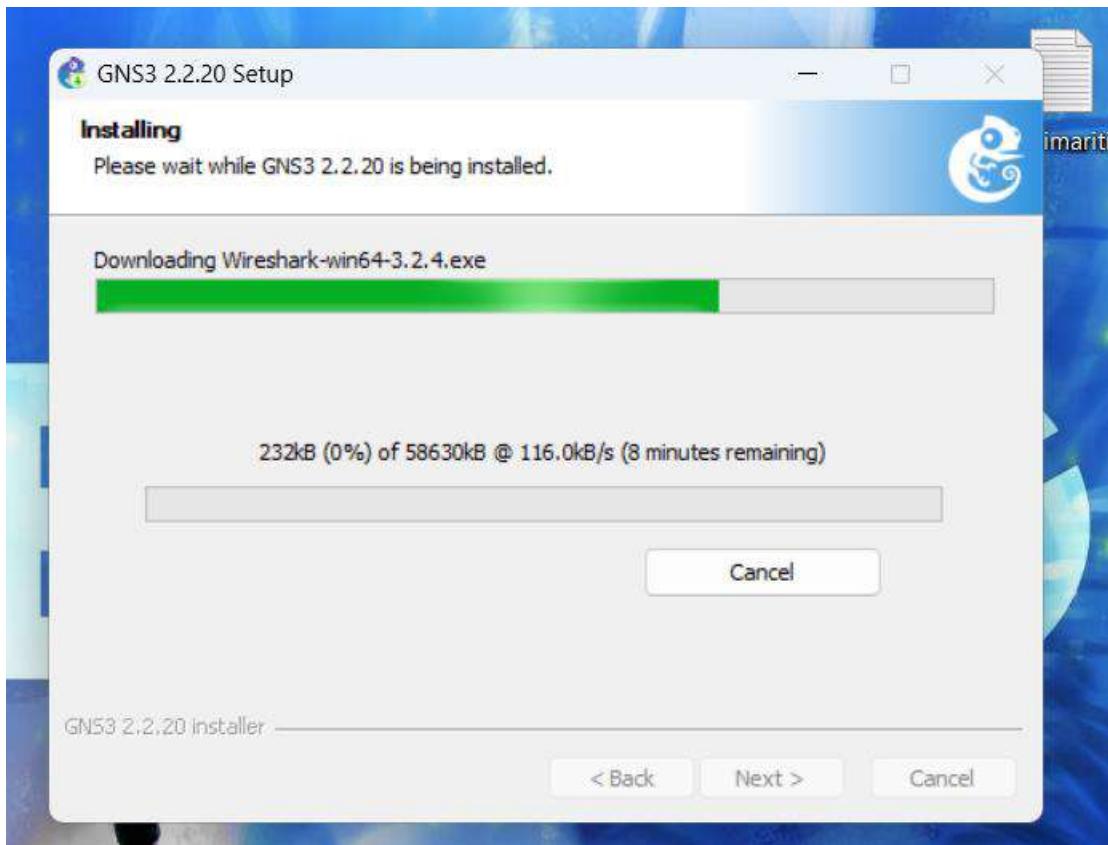


Figure 251: Installing GNS3 19.

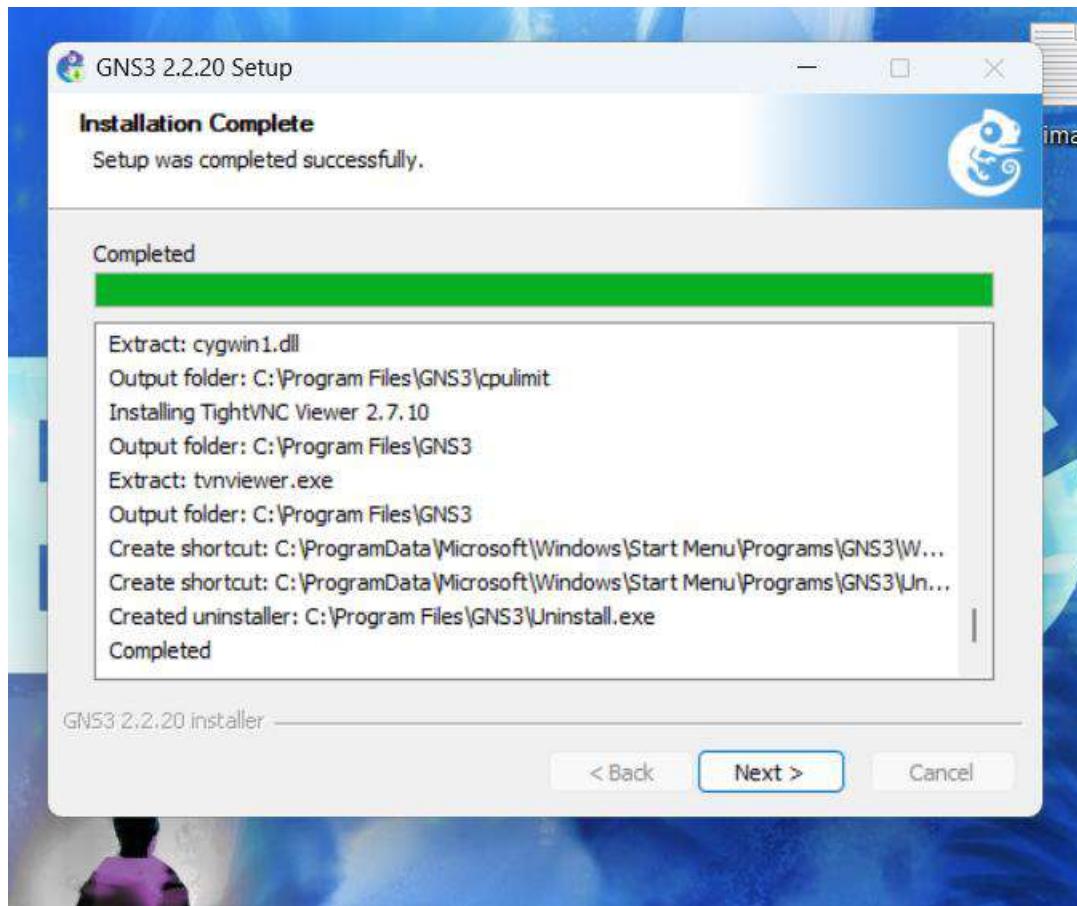


Figure 252: Installing GNS3 2.1.

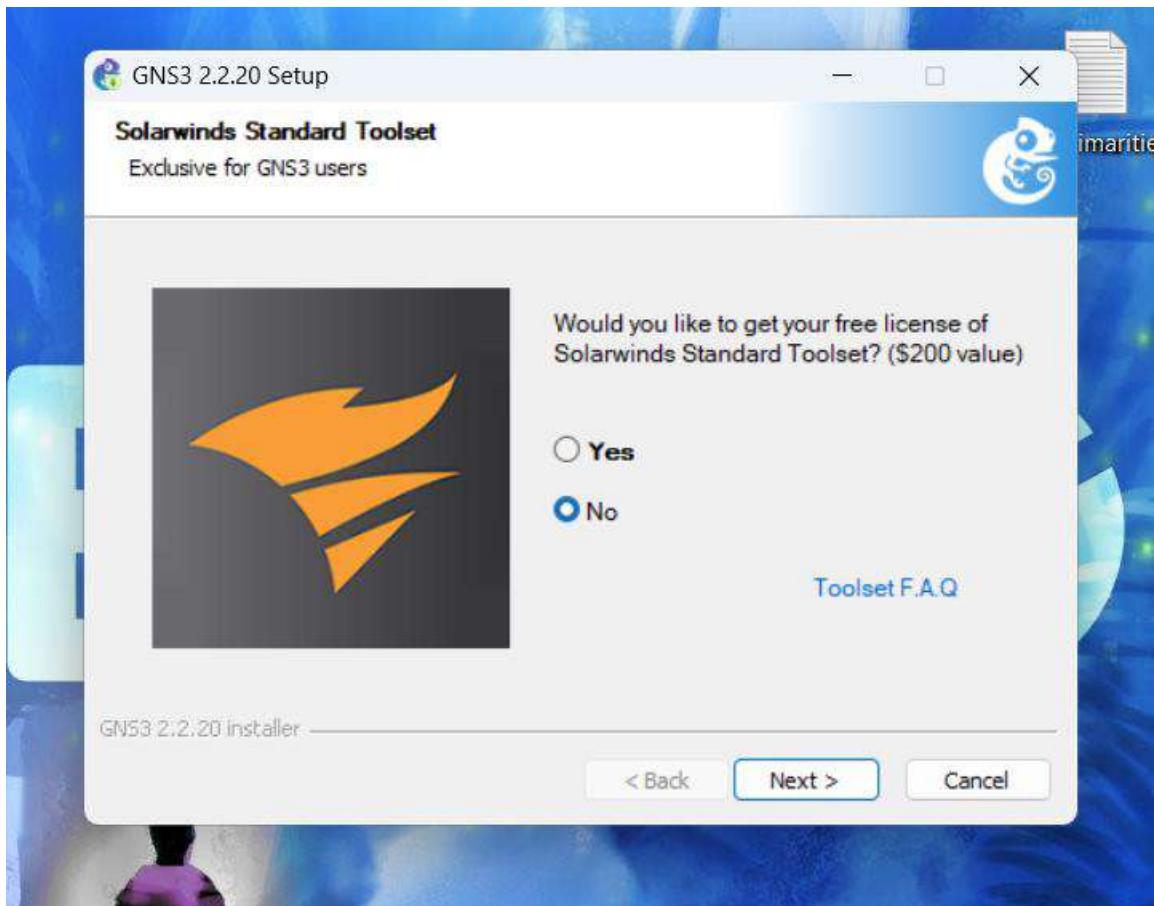


Figure 253: Installing GNS3 22.



Figure 254: Installing GNS3 23.

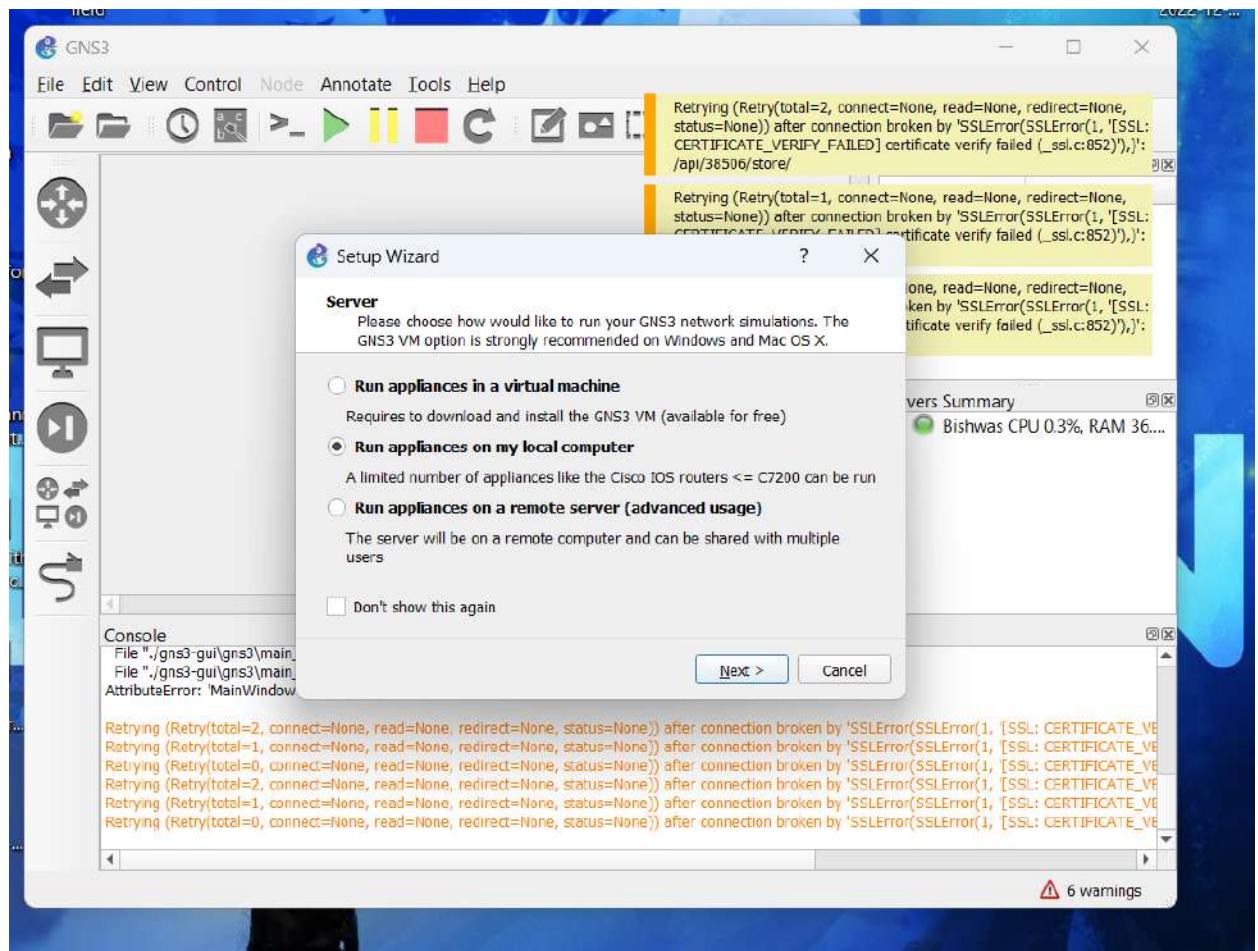


Figure 255: Installing GNS3 24.

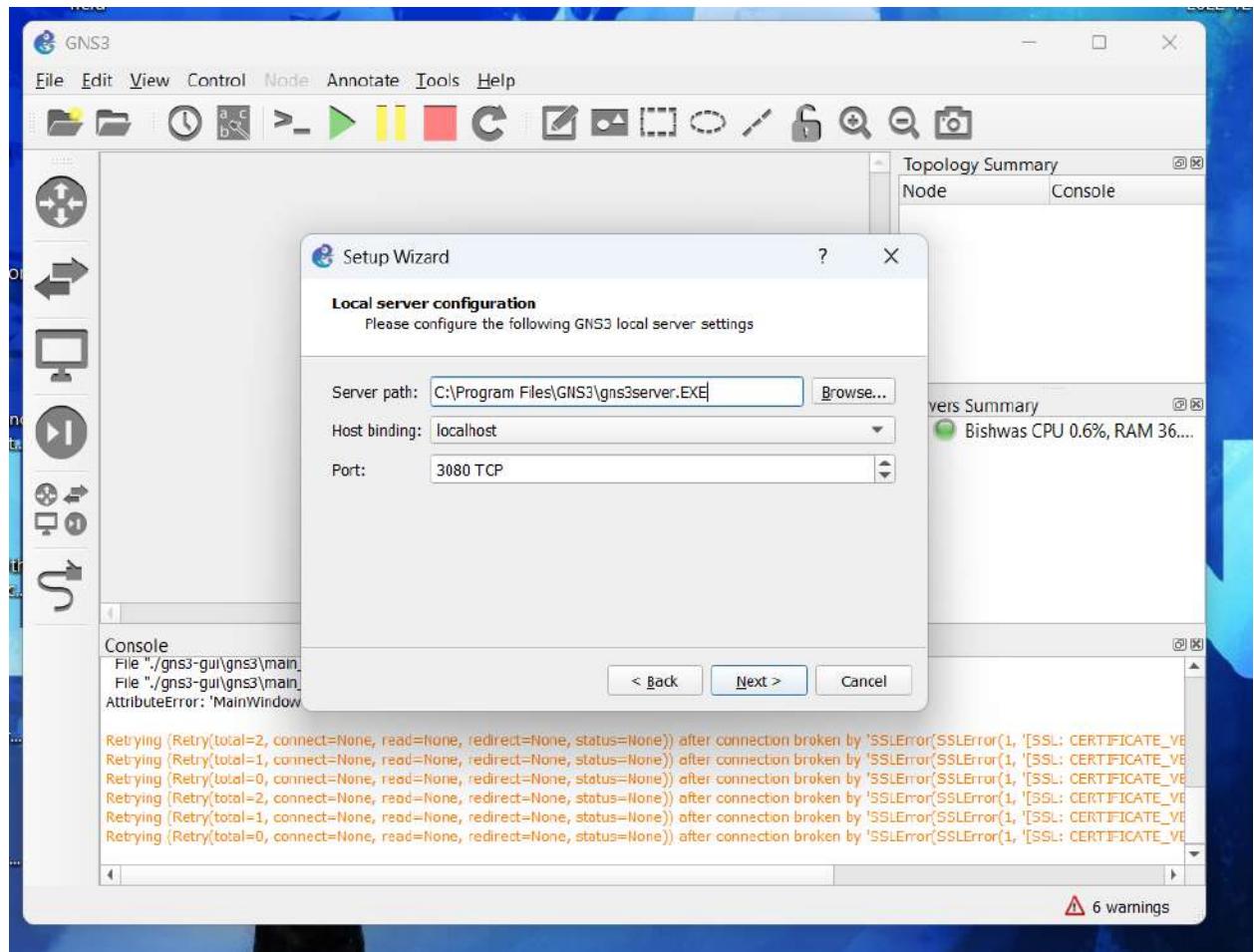


Figure 256: Installing GNS3 25.

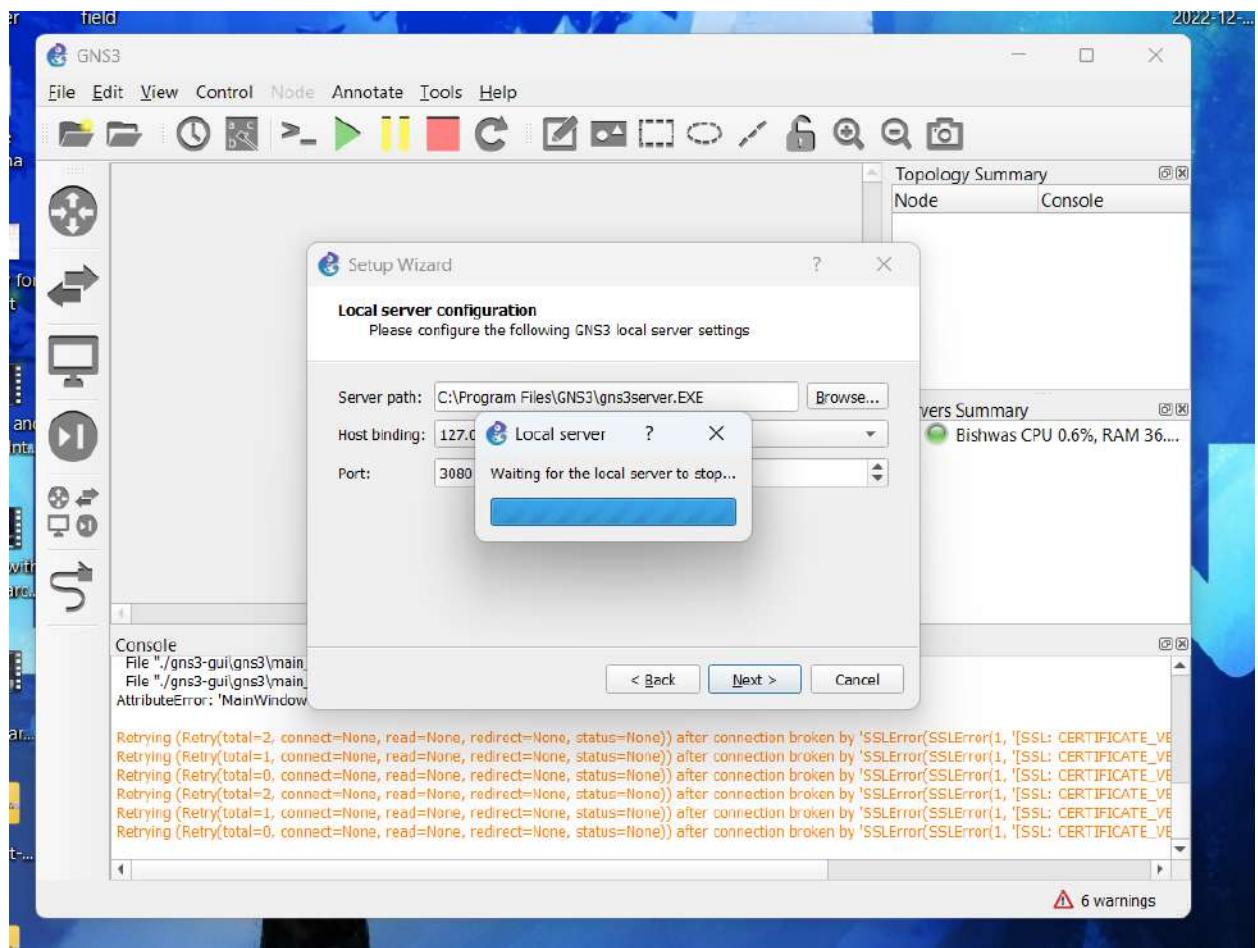


Figure 257: Installing GNS3 26.

Configuration for Cloud interface in GNS3

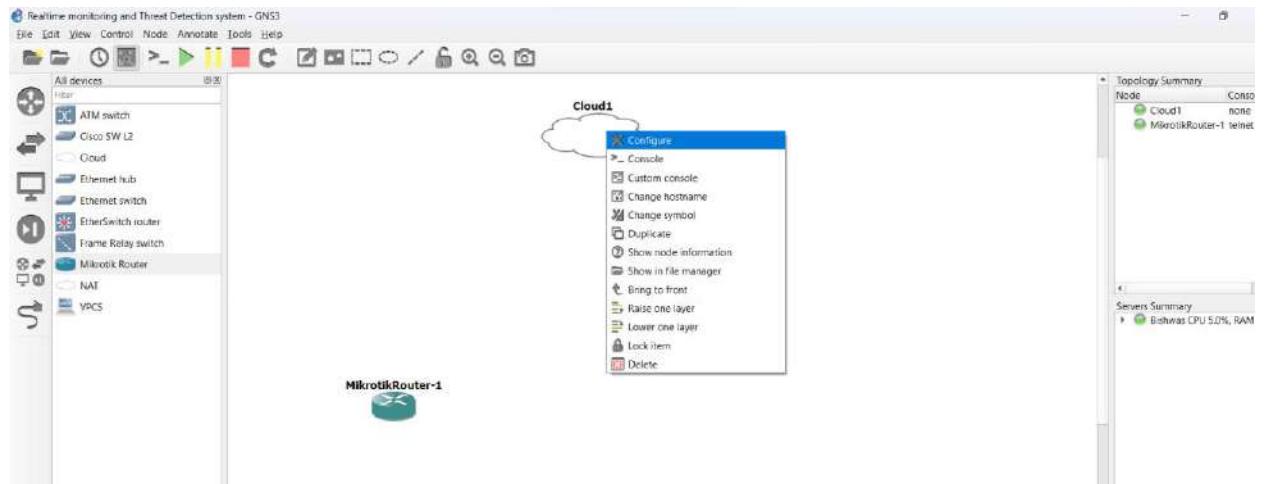
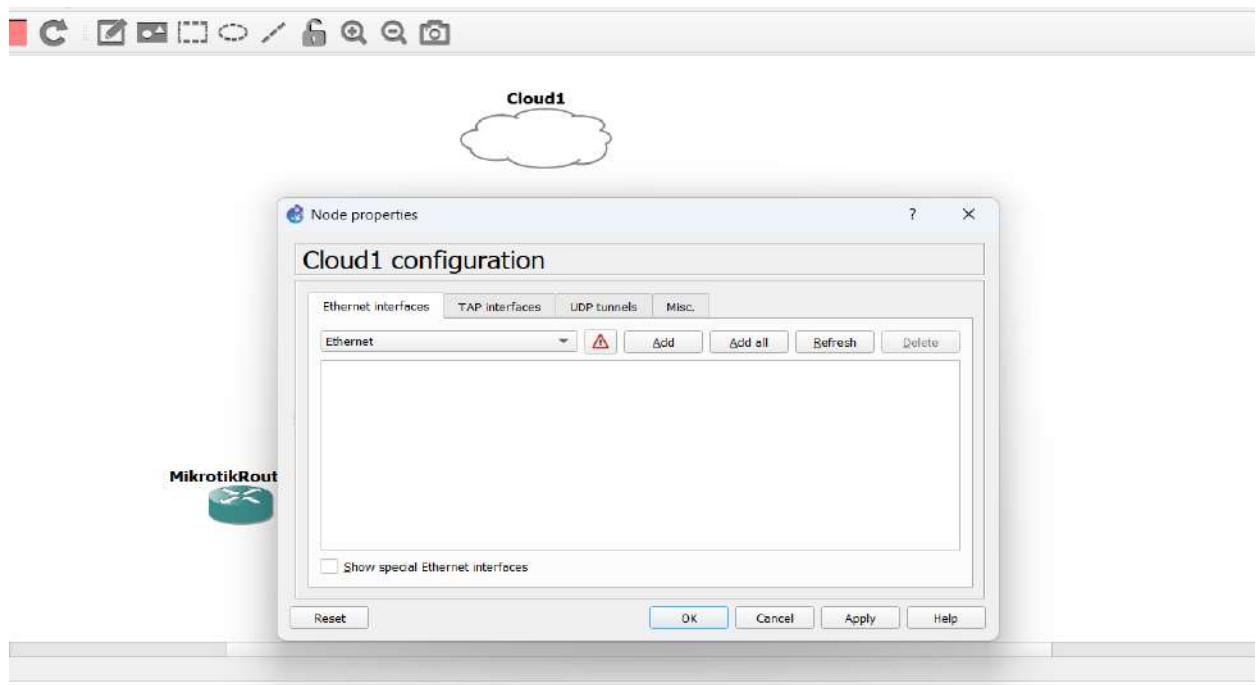


Figure 258: configuration for cloud interface 1.



It) with Python 3.7.5 Qt 5.15.2 and PyQt 5.15.4.

Figure 259: configuration for cloud interface 2.

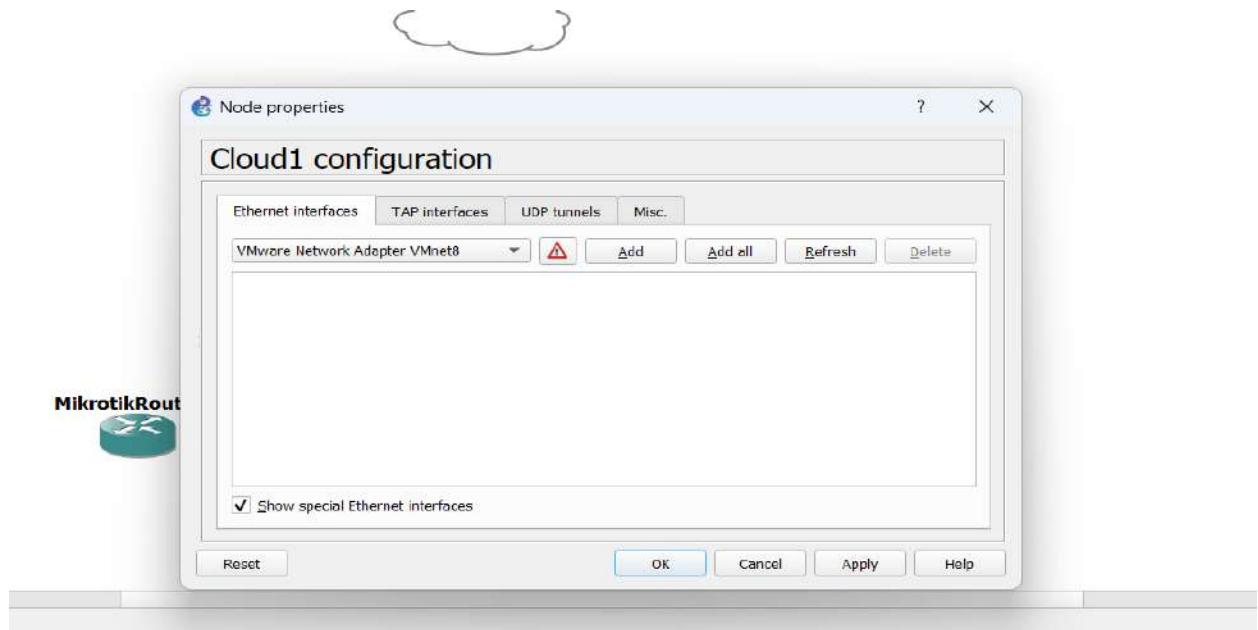
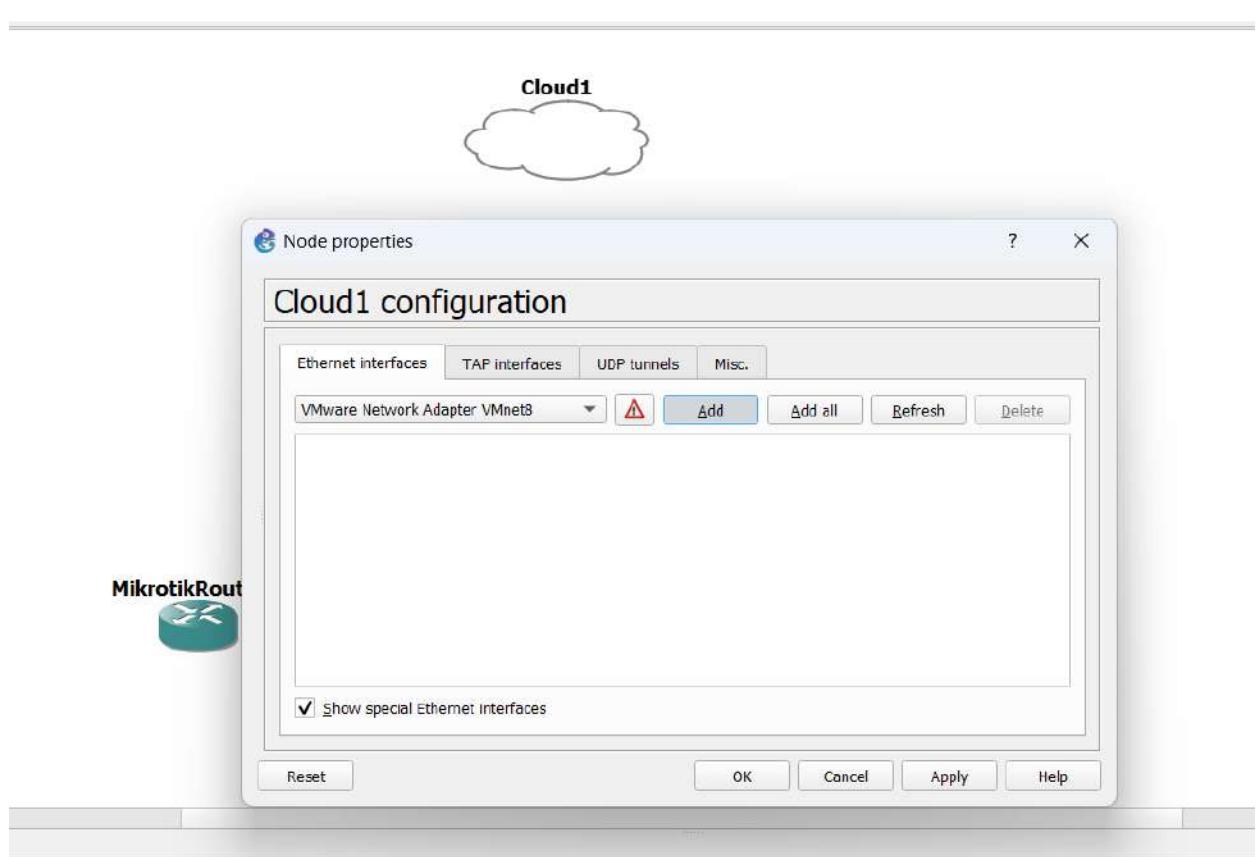


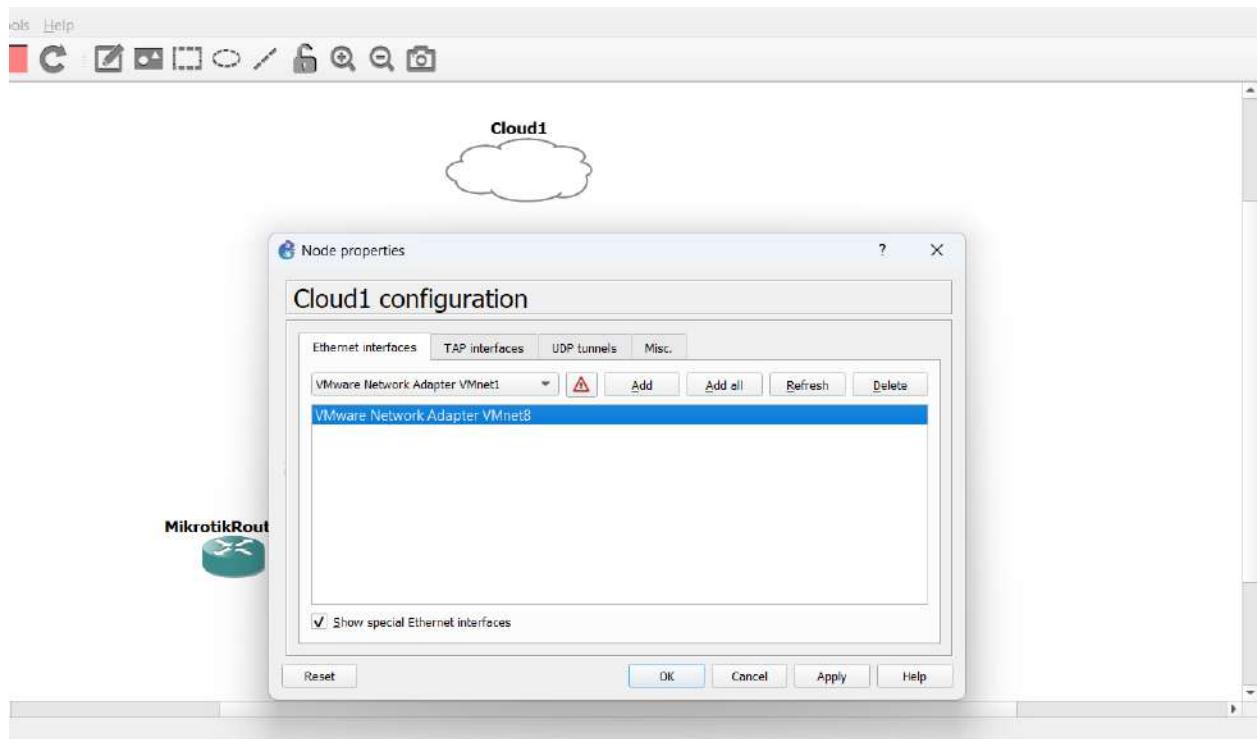
Figure 260: configuration for cloud interface 3.



in 3.7.5 Qt 5.15.2 and PyQt 5.15.4.

Want to use the CMAQ2 VM

Figure 261: configuration for cloud interface 4.

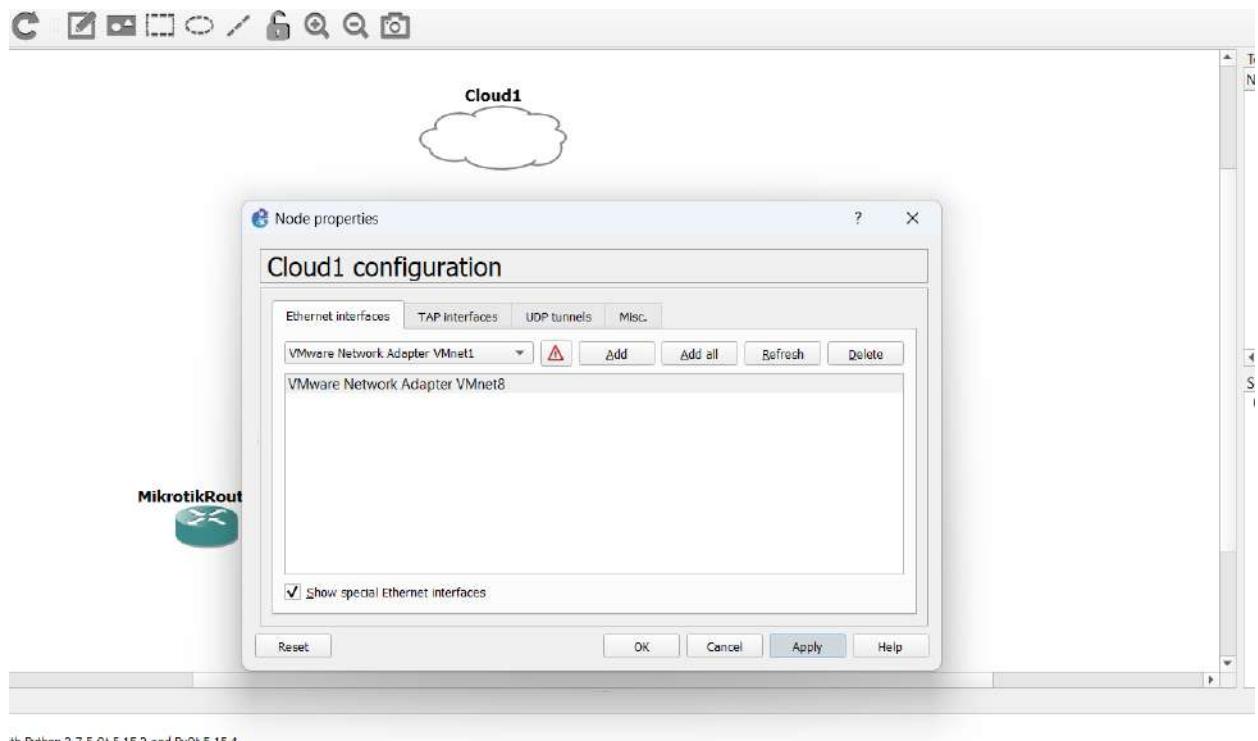


It) with Python 3.7.5 Qt 5.15.2 and PyQt 5.15.4.

jes.

indows and OSX is to use the GNS3 VM
ys and OSX is to use the GNS3 VM
is host

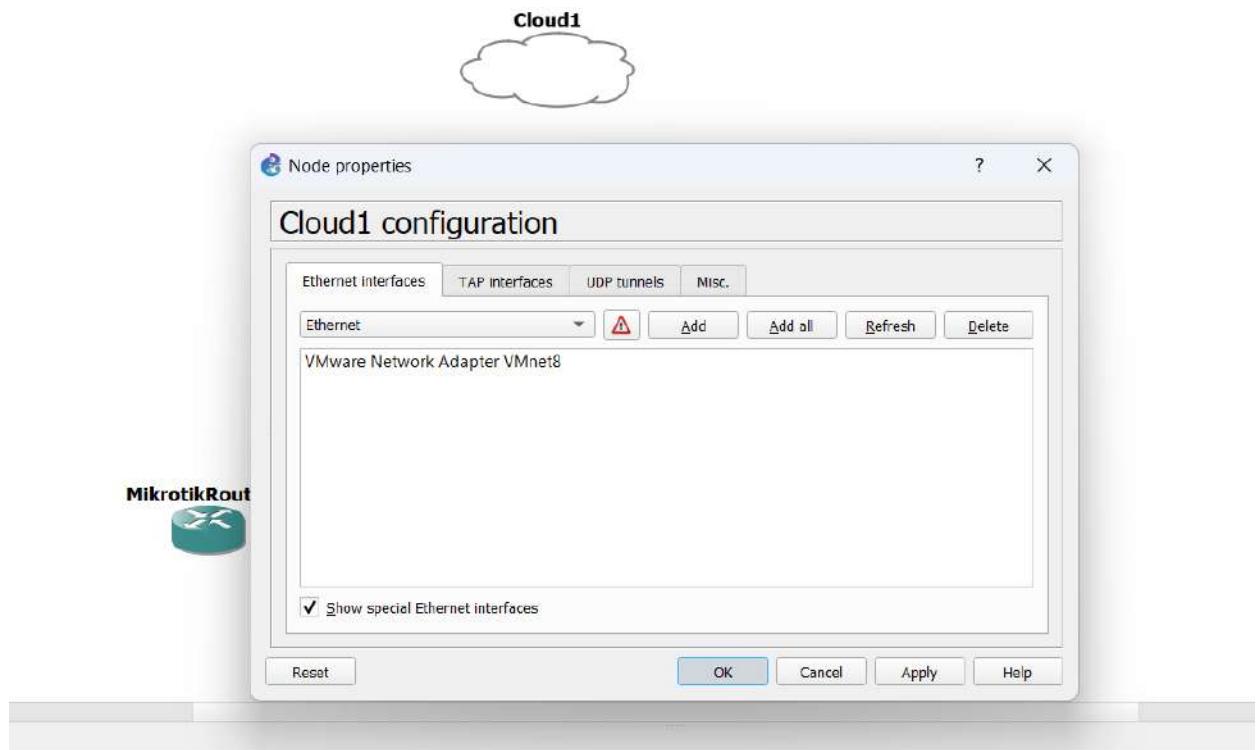
Figure 262: configuration for cloud interface 5.



th Python 3.7.5 Qt 5.15.2 and PyQt 5.15.4.

s and OSX is to use the GNS3 VM
d OSX is to use the GNS3 VM
t

Figure 263: configuration for cloud interface 6.



thon 3.7.5 Qt 5.15.2 and PyQt 5.15.4.

Figure 264: configuration for cloud interface 7.

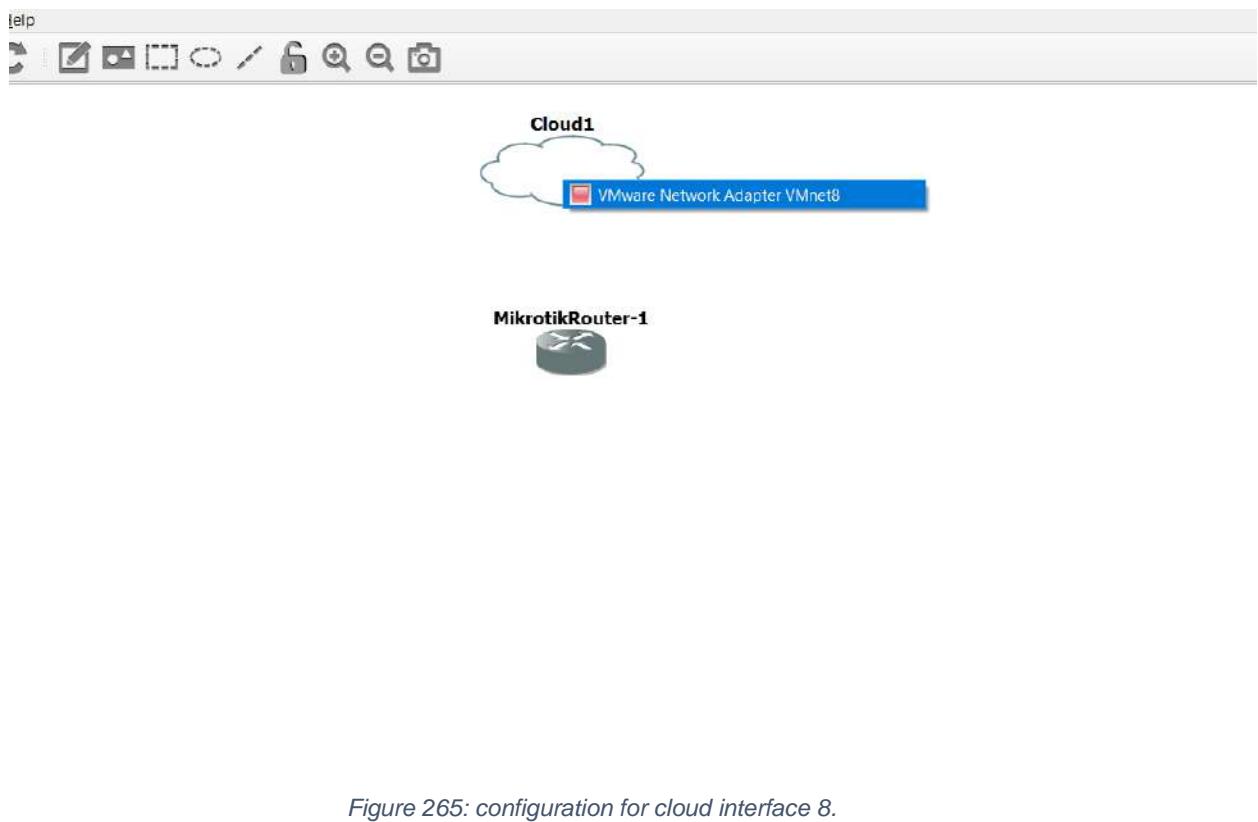


Figure 265: configuration for cloud interface 8.

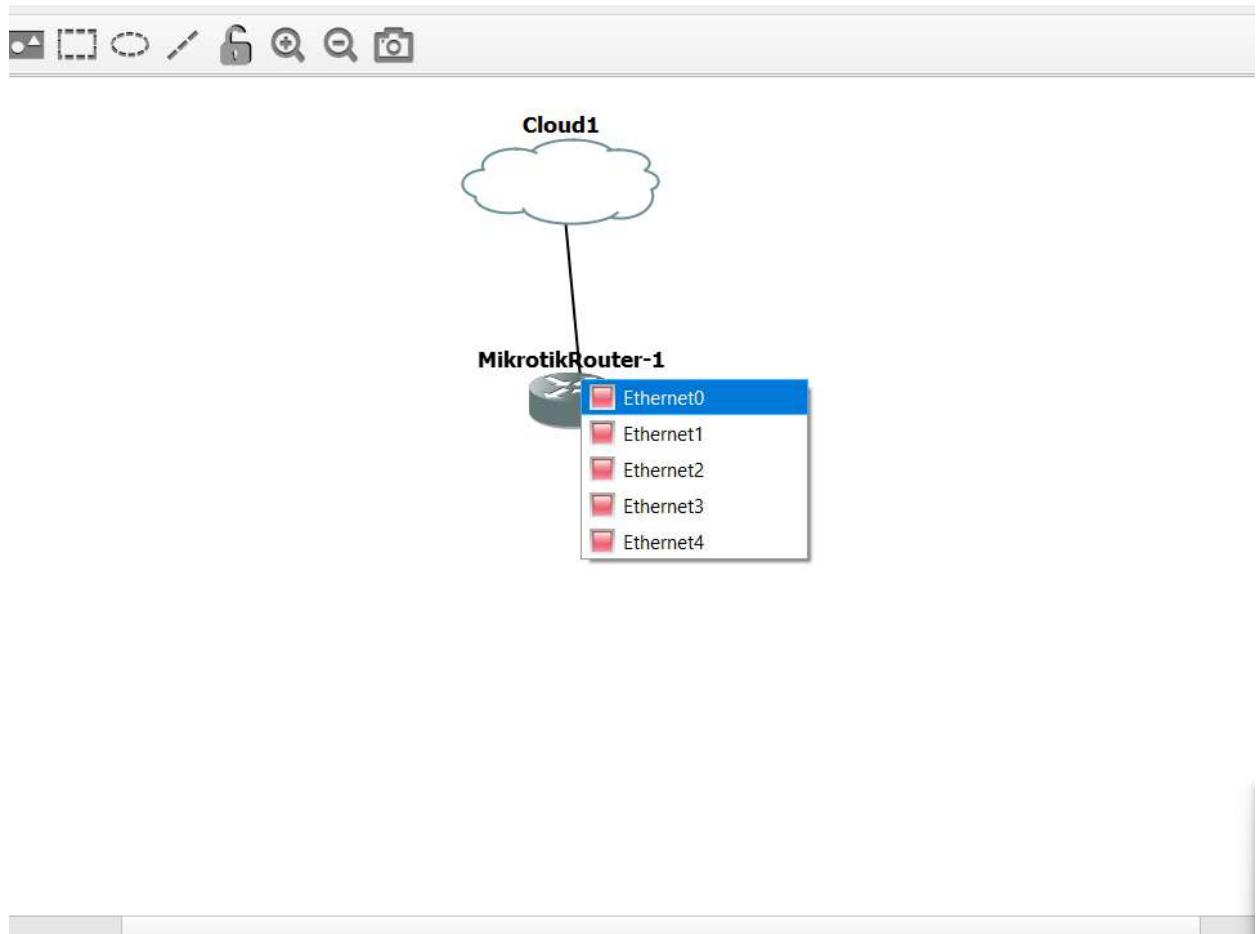


Figure 266: configuration for cloud interface 9.

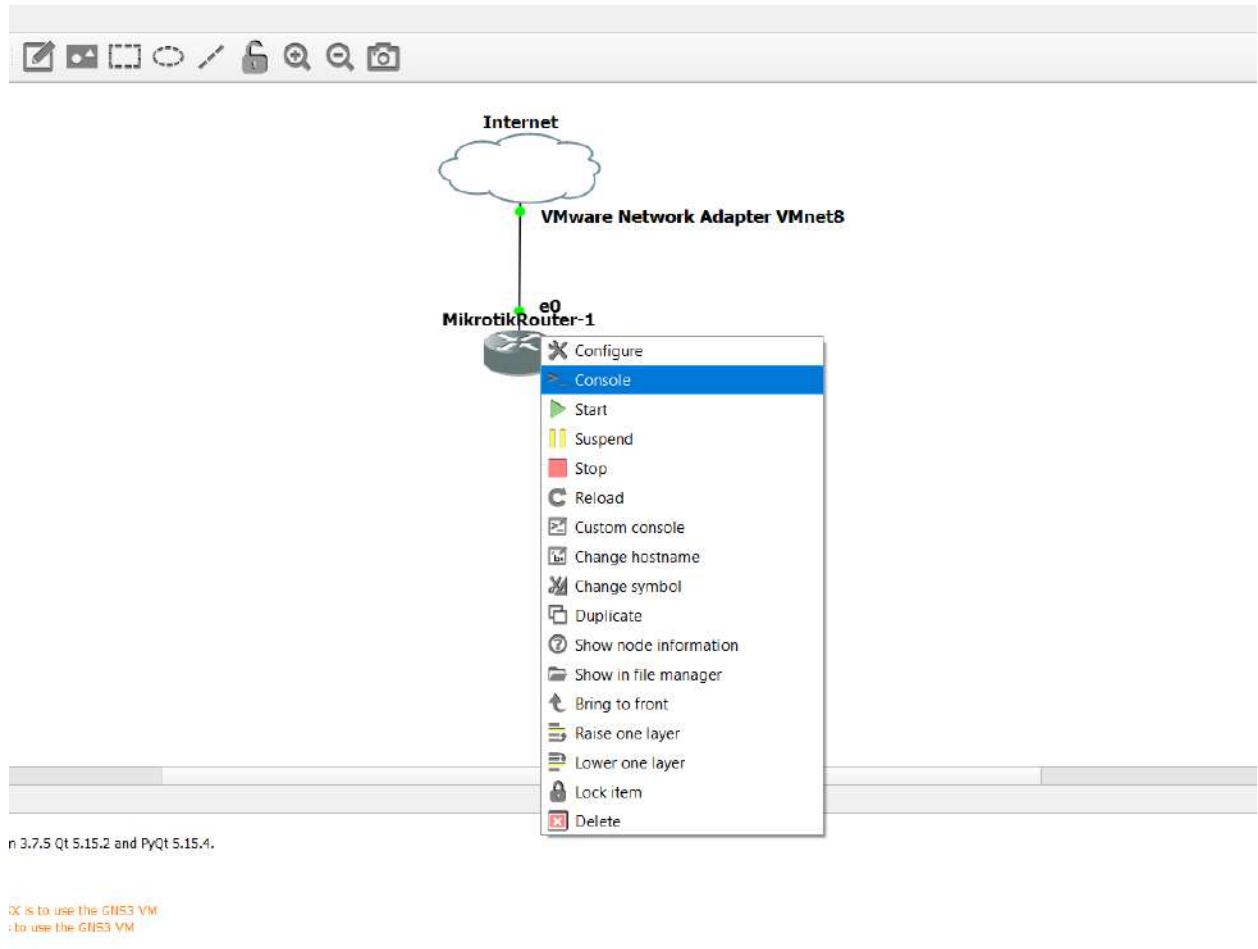


Figure 267: configuration for cloud interface 10.

Importing appliances for GNS3

Mikrotik Router

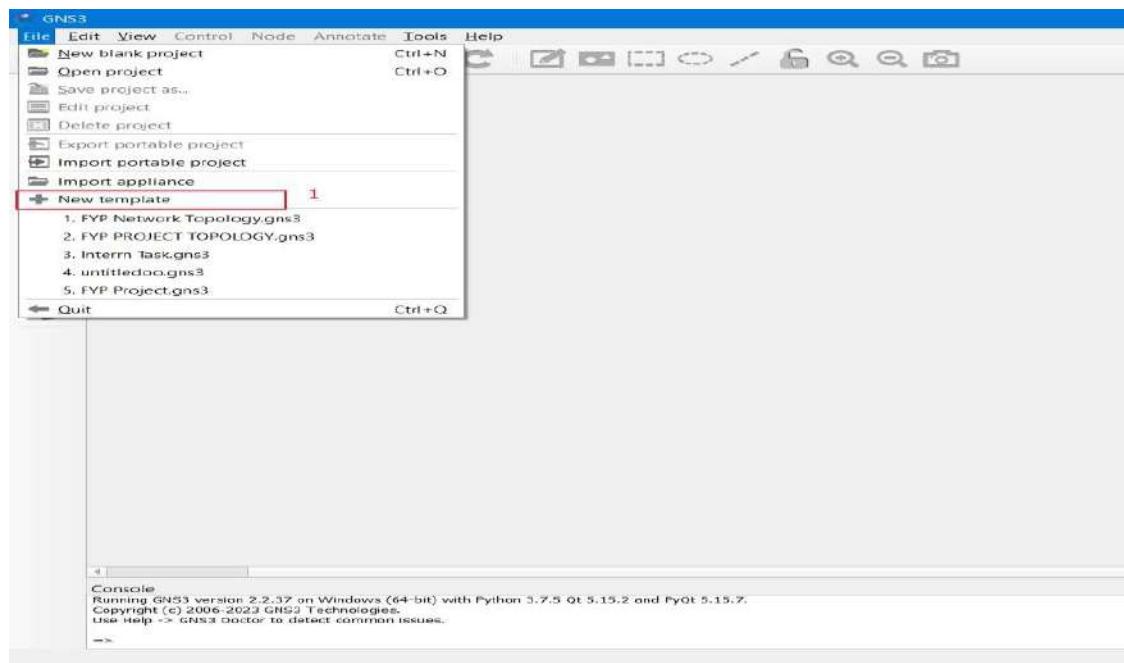


Figure 268: Importing Mikrotik router 1.

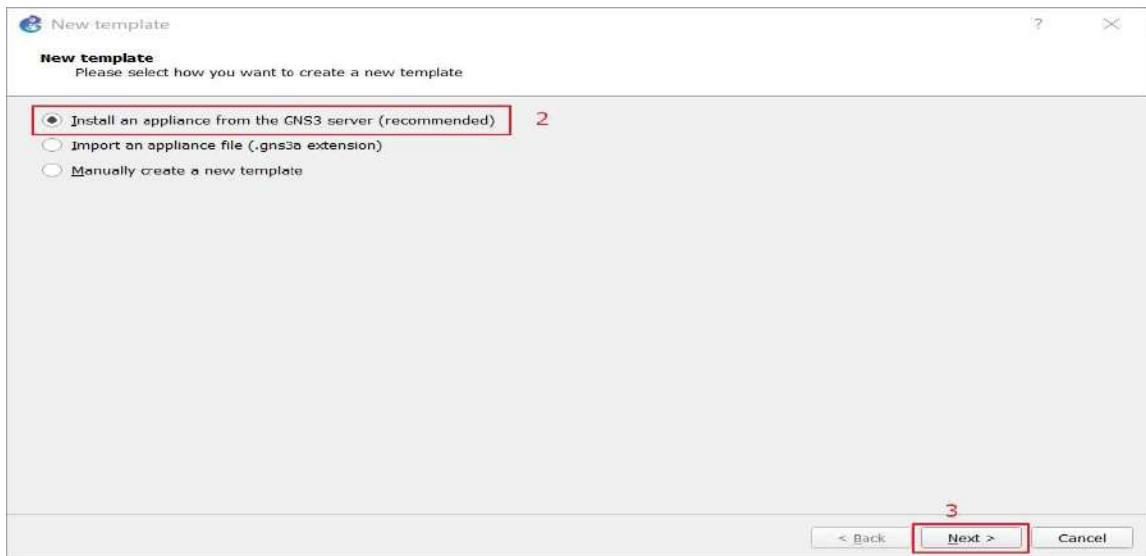


Figure 269: Importing Mikrotik router 2.

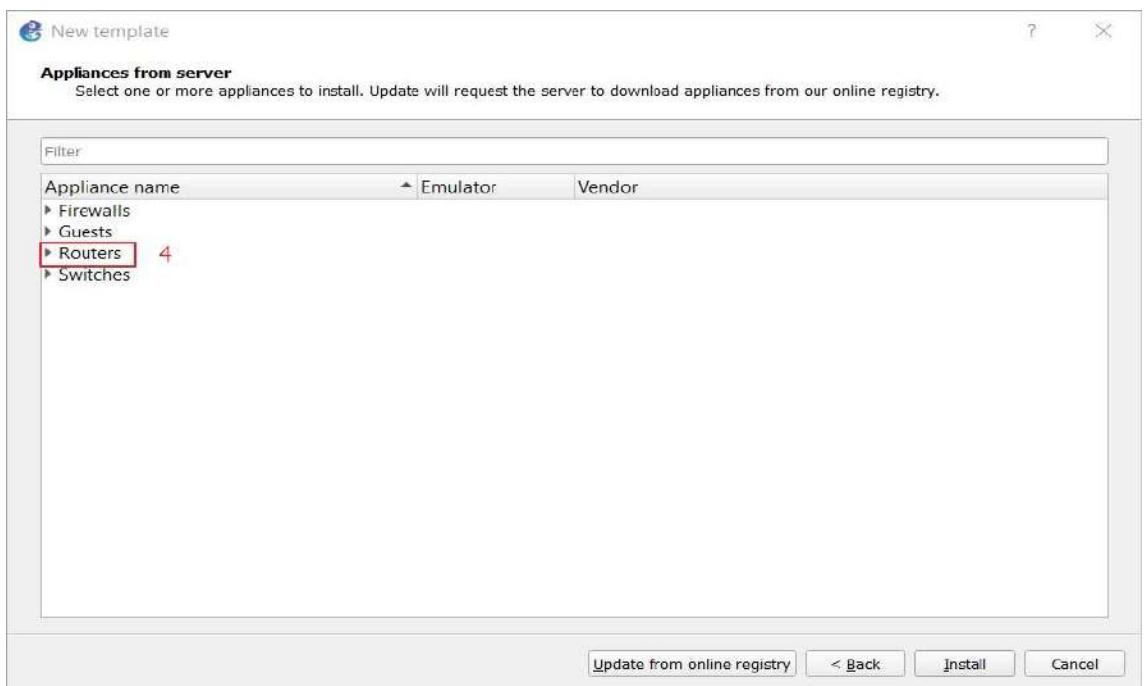


Figure 270: Importing Mikrotik router 3.

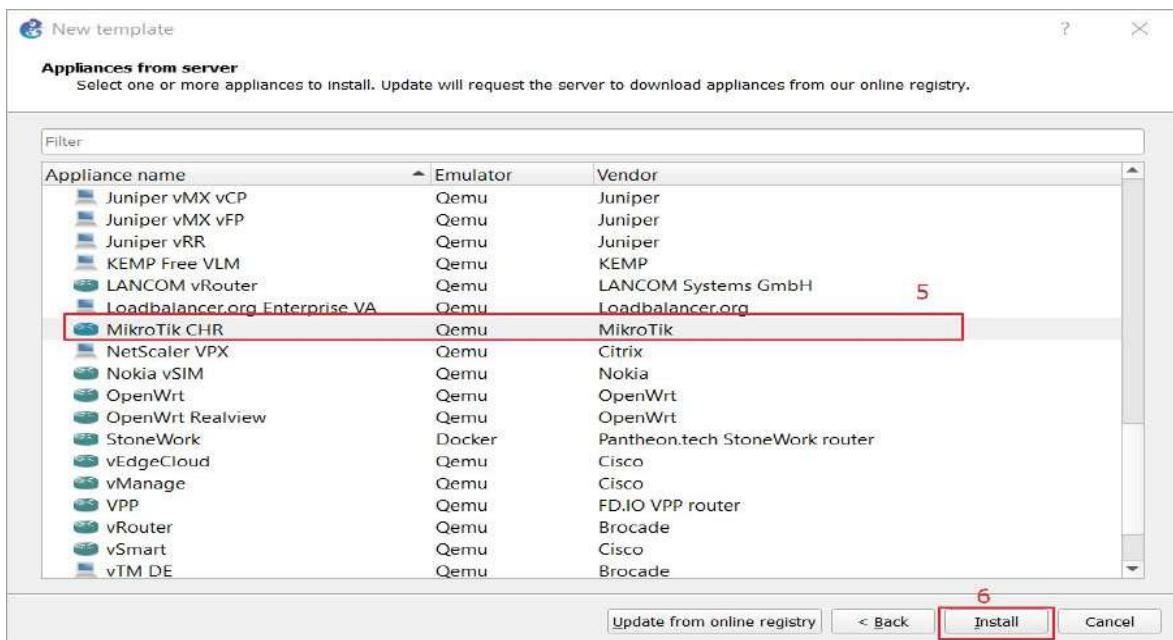


Figure 271: Importing Mikrotik router 4.



Figure 272: Importing Mikrotik router 5.



Figure 273: Importing Mikrotik router 6.

Install MikroTik CHR appliance

Required files
Please select one version of MikroTik Cloud Hosted Router and import the required files. Files are searched in your downloads and GNS3 images directories by default

Appliance version and files	Size	Status
✓ MikroTik Cloud Hosted Router version 7.7 chr-7.7.img	128.0 MB	Missing
✓ MikroTik Cloud Hosted Router version 7.6 chr-7.6.img	128.0 MB	Missing
✓ MikroTik Cloud Hosted Router version 7.3.1 chr-7.3.1.img	128.0 MB	Missing
✓ MikroTik Cloud Hosted Router version 7.1.5 chr-7.1.5.img	128.0 MB	Missing
✓ MikroTik Cloud Hosted Router version 6.49.6 chr-6.49.6.img	64.0 MB	Missing
✓ MikroTik Cloud Hosted Router version 6.48.6 chr-6.48.6.img	64.0 MB	10 Found on GNS3 VM (GNS3 VM)

11

This image file was downloaded before hand and save in folder.

Allow custom files Create a new version Refresh

Appliance info < Back **Next >** Cancel

Figure 274: Importing Mikrotik router 7.

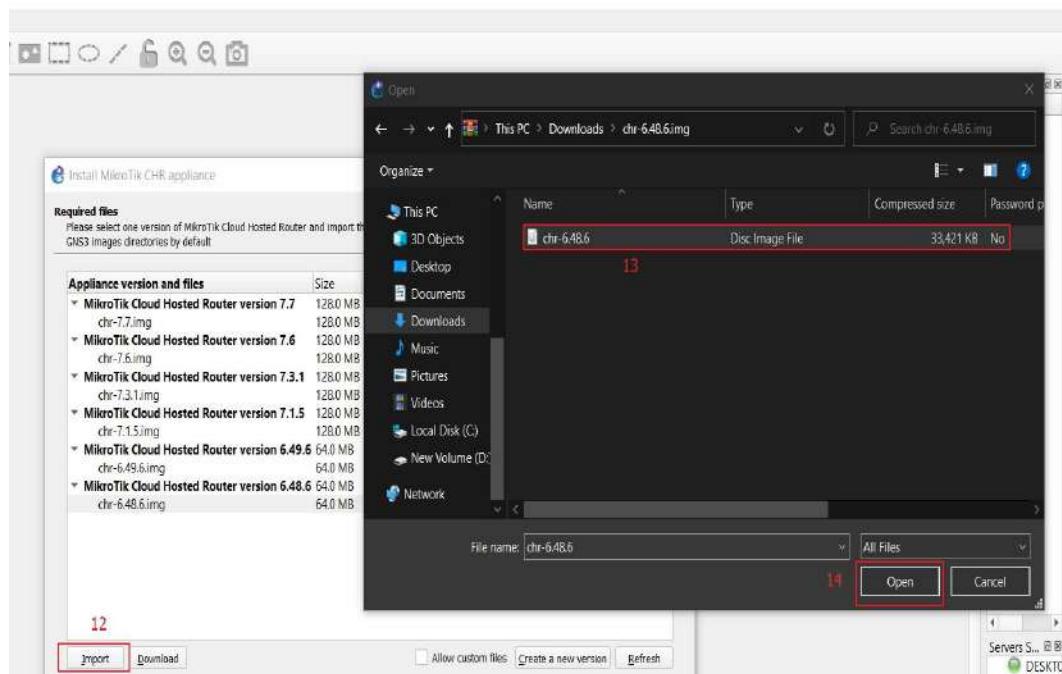


Figure 275: Importing Mikrotik router 8.

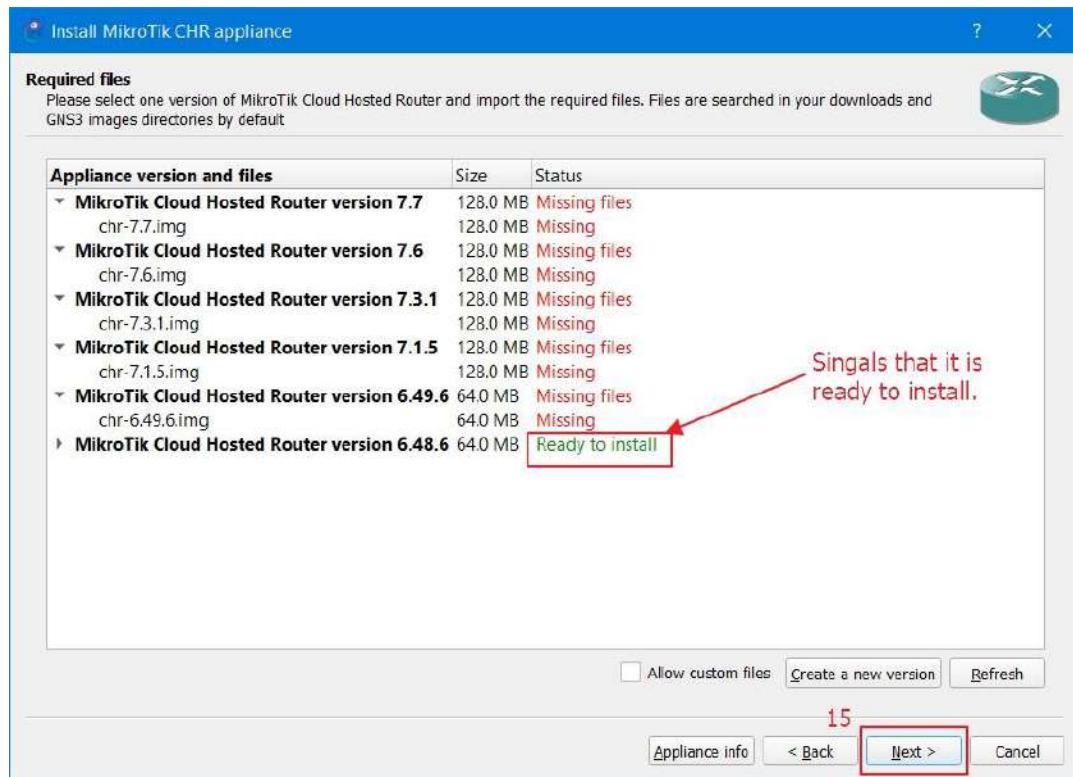


Figure 276: Importing Mikrotik router 9.

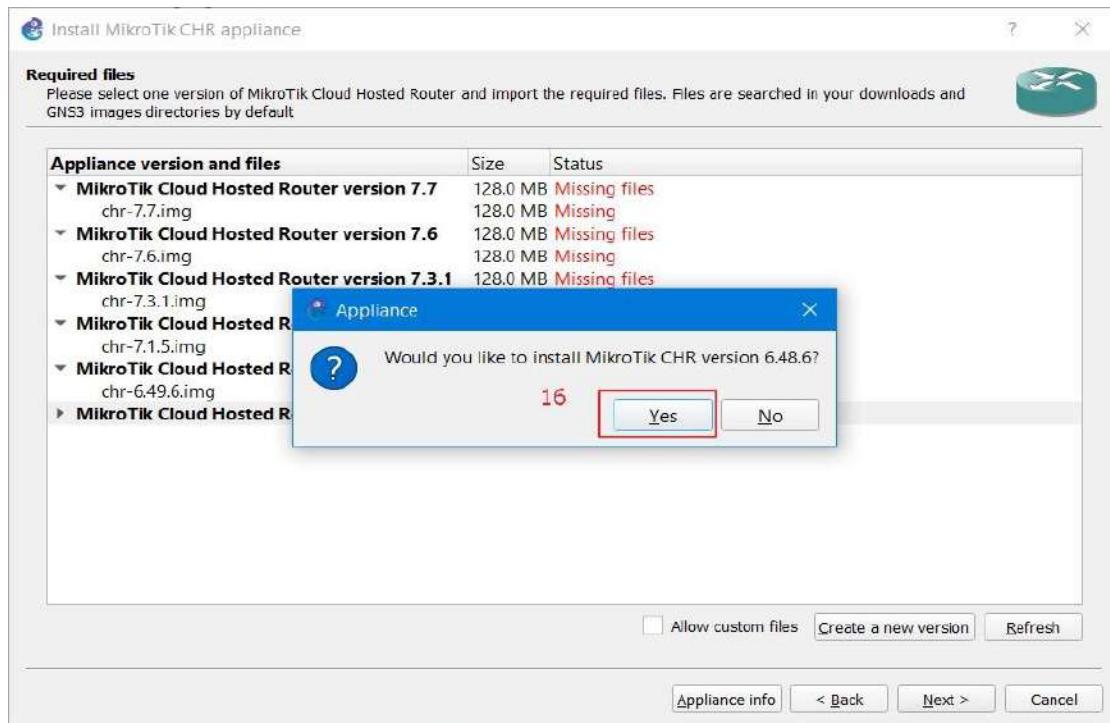


Figure 277: Importing Mikrotik router 10.

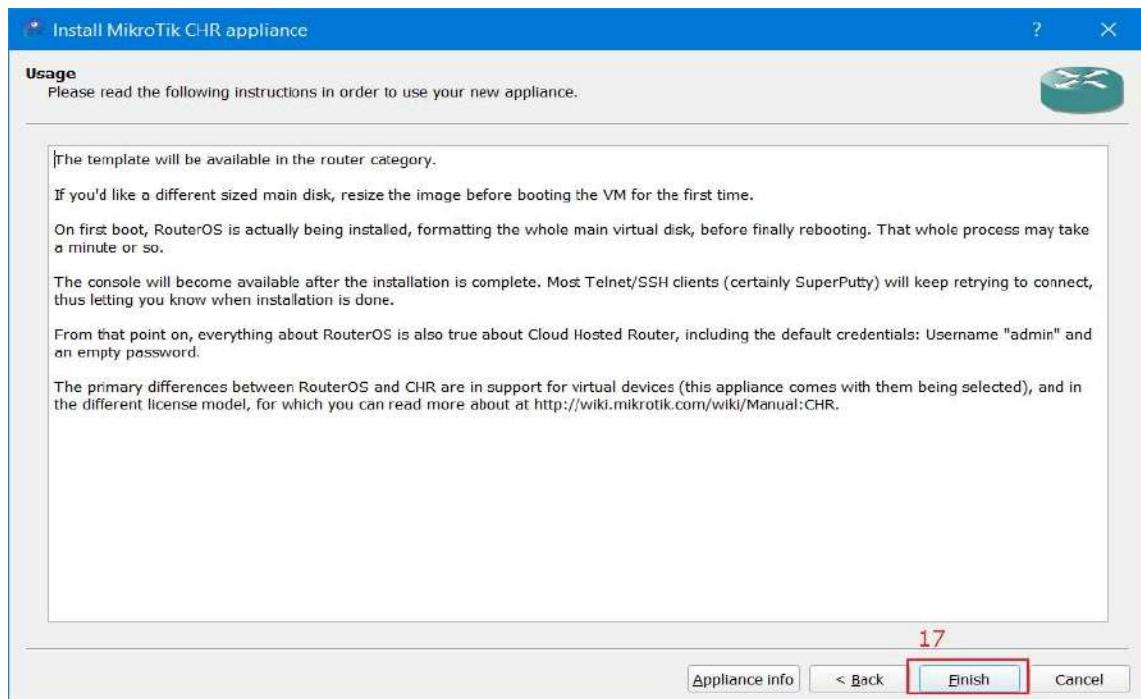


Figure 278: Importing Mikrotik router 11.

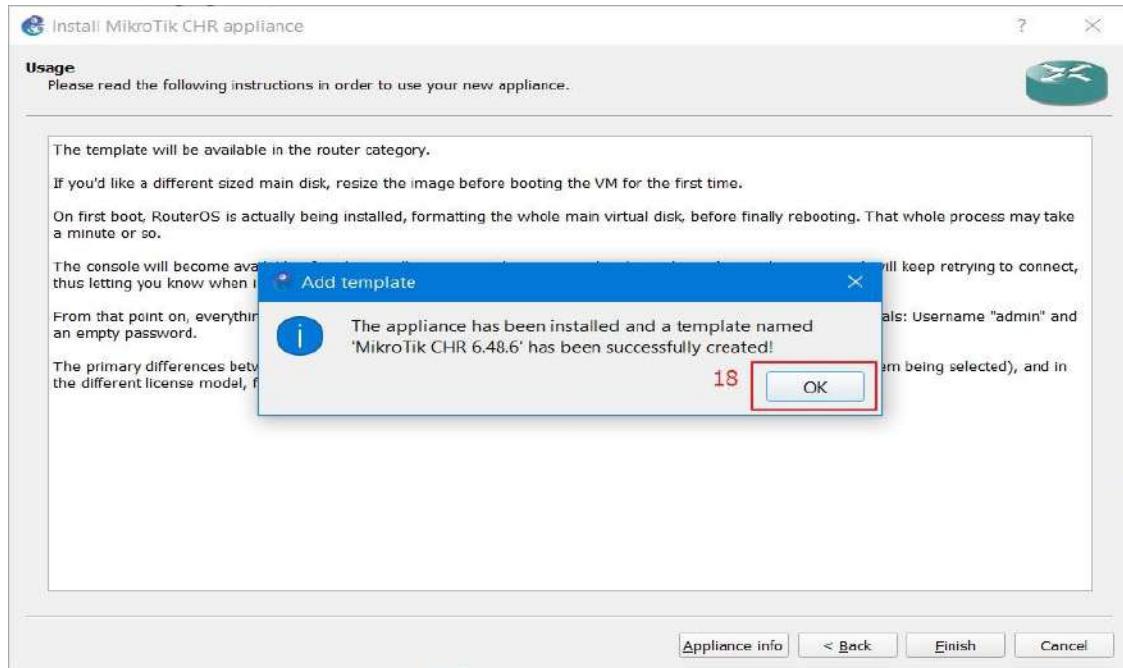


Figure 279: Importing Mikrotik router 12.

PfSense

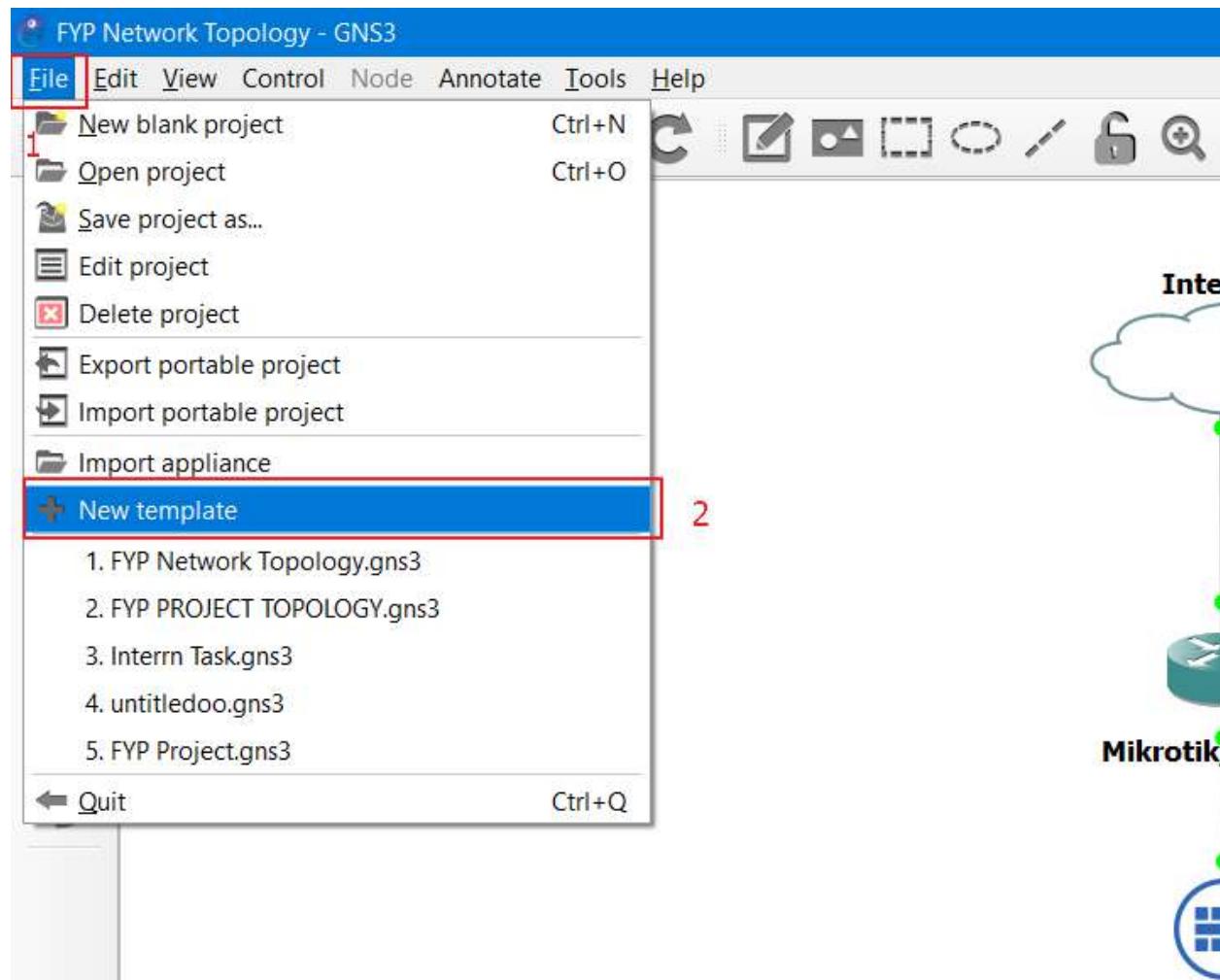


Figure 280: Importing PfSense 1.

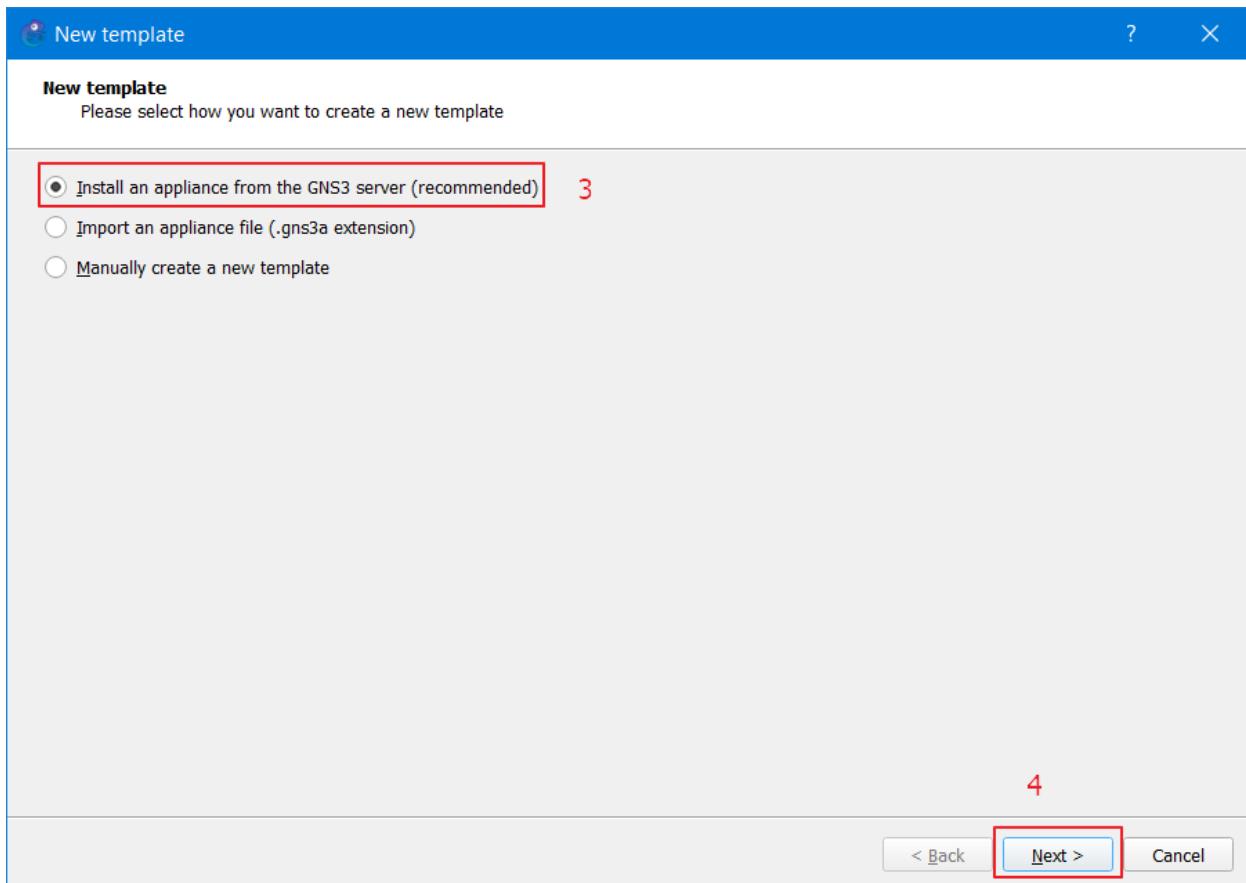


Figure 281: Importing PfSense 2.

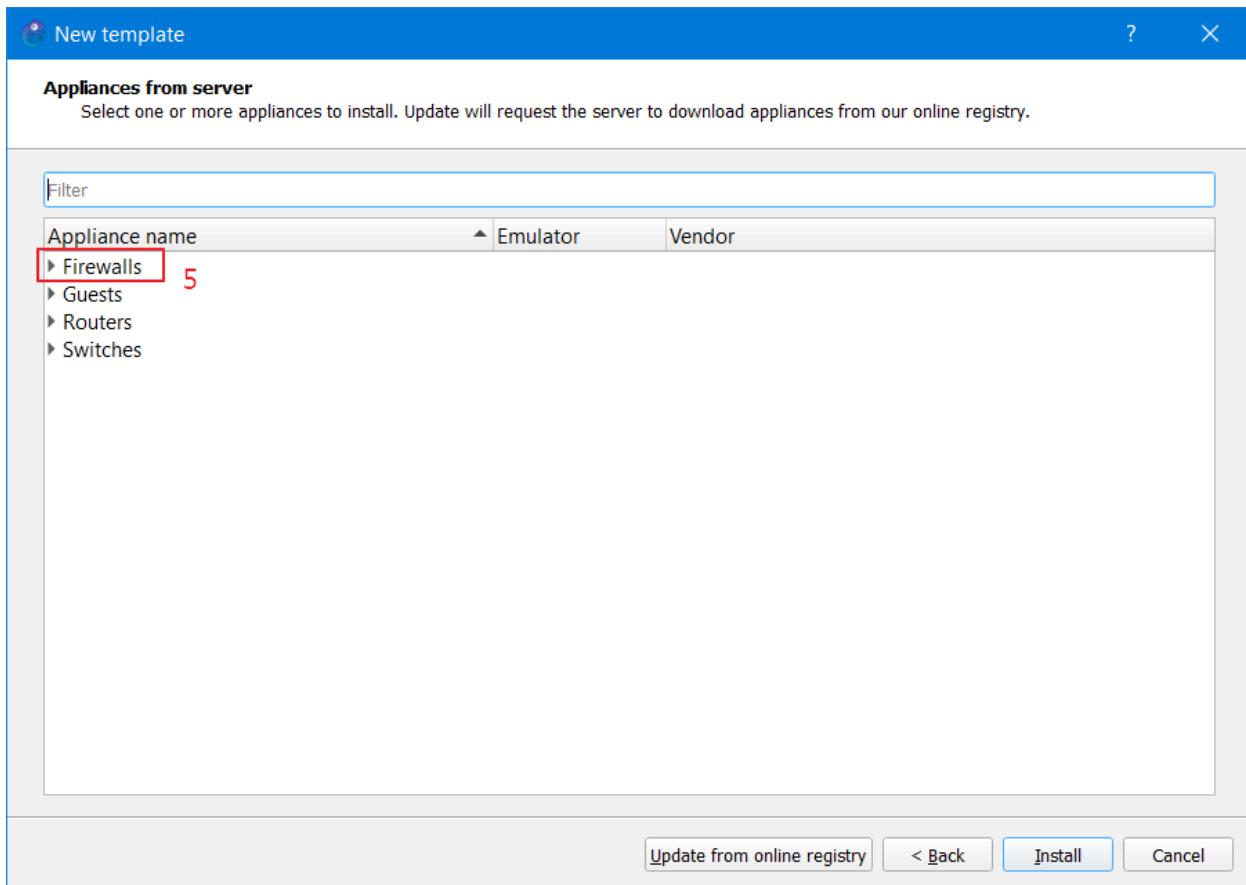


Figure 282: Importing Pfsense 3.

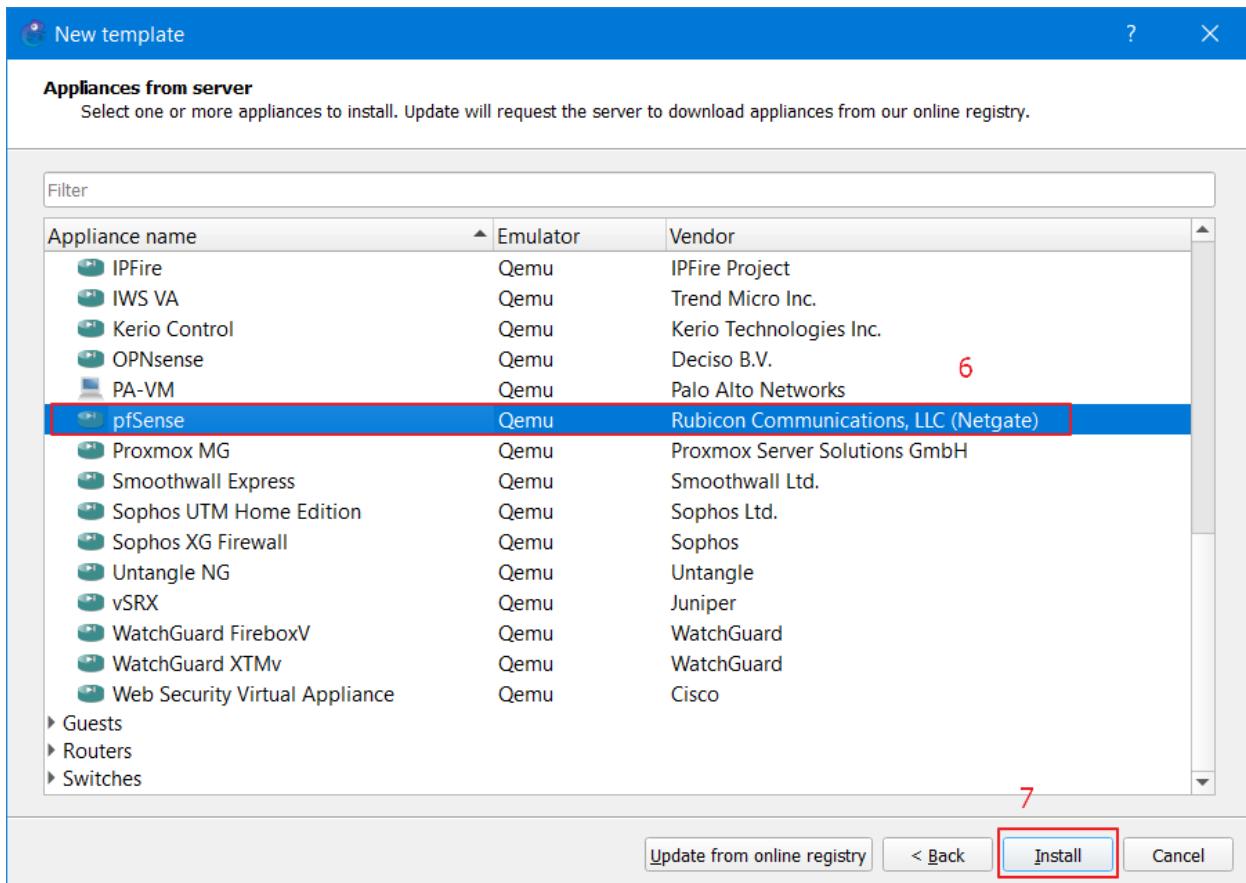


Figure 283: Importing PfSense 4.

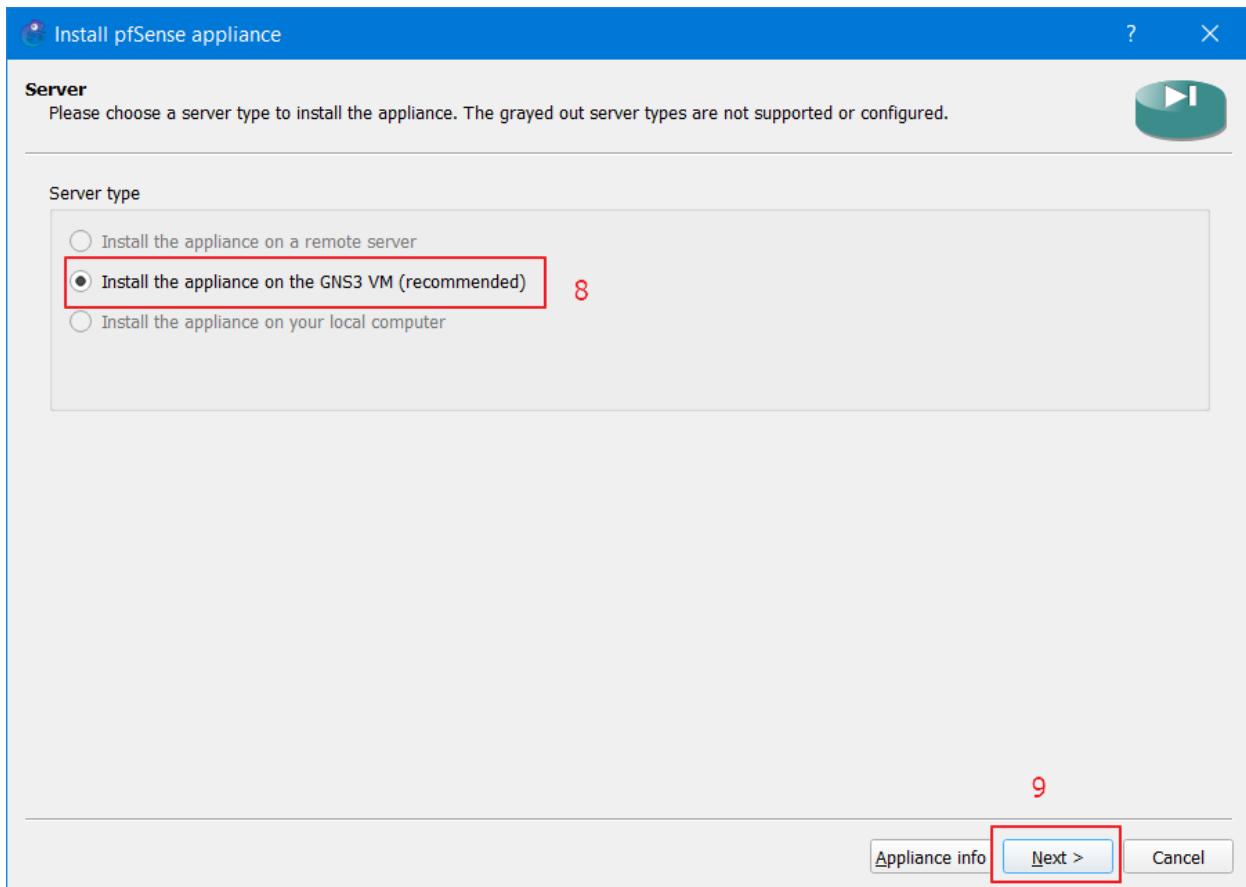


Figure 284: Importing Pfsense 5.

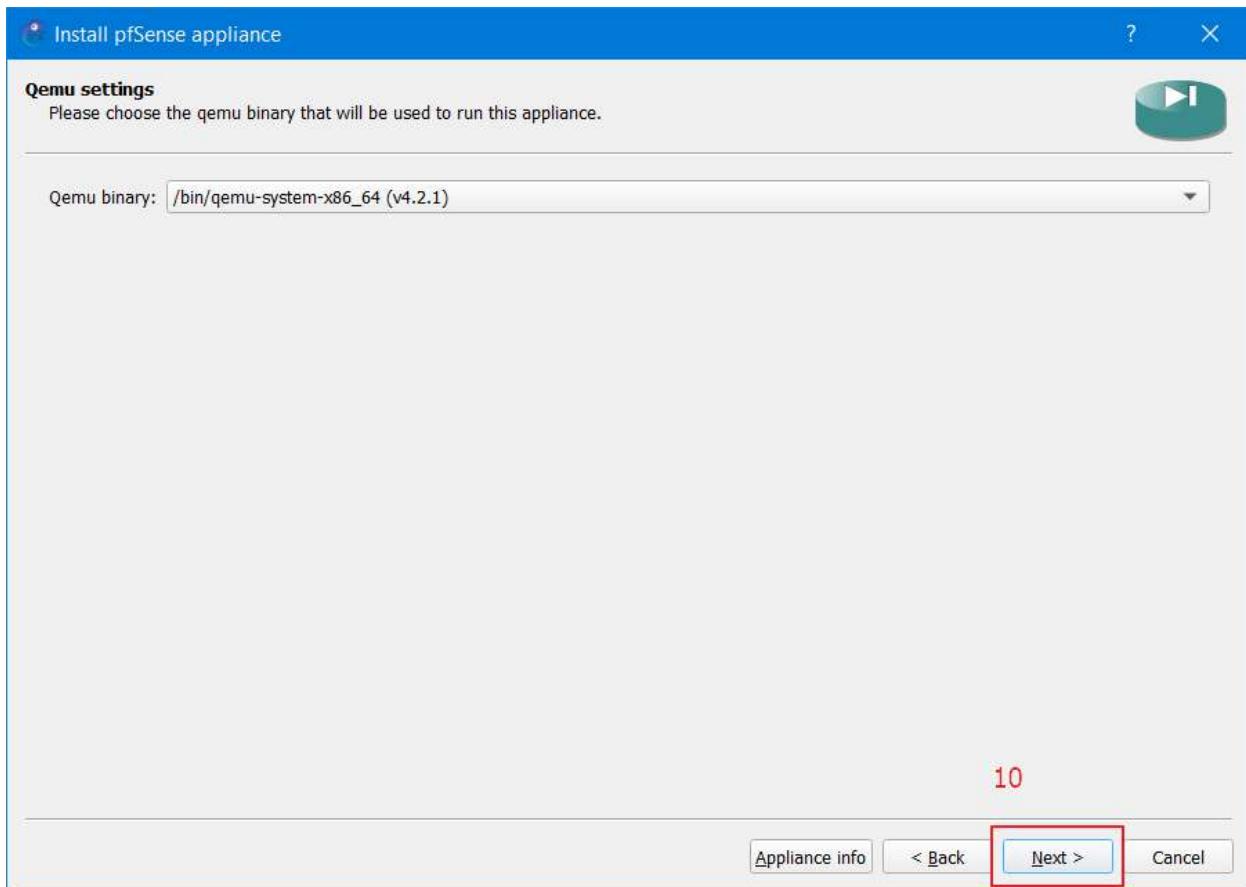


Figure 285: Importing Pfsense 6.

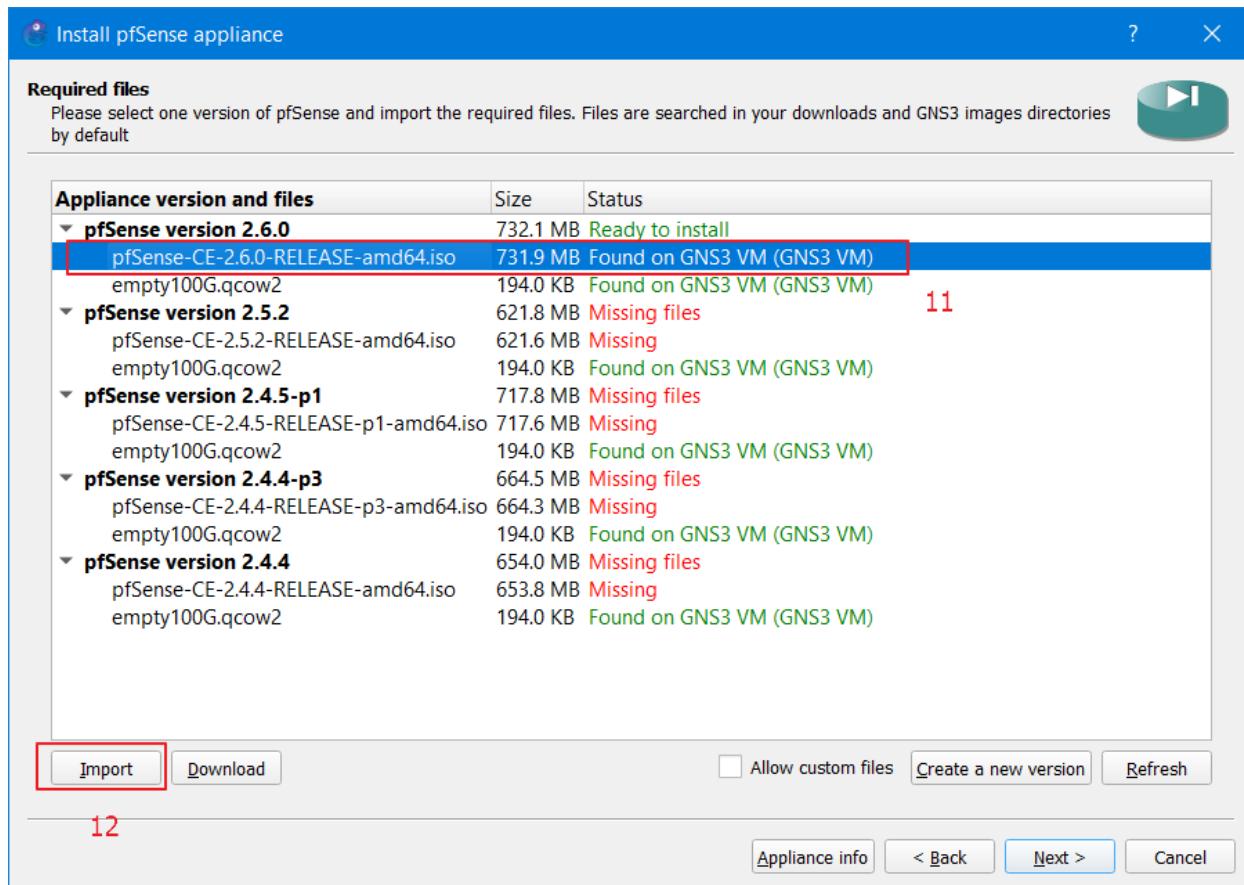


Figure 286: Importing Pfsense 7.

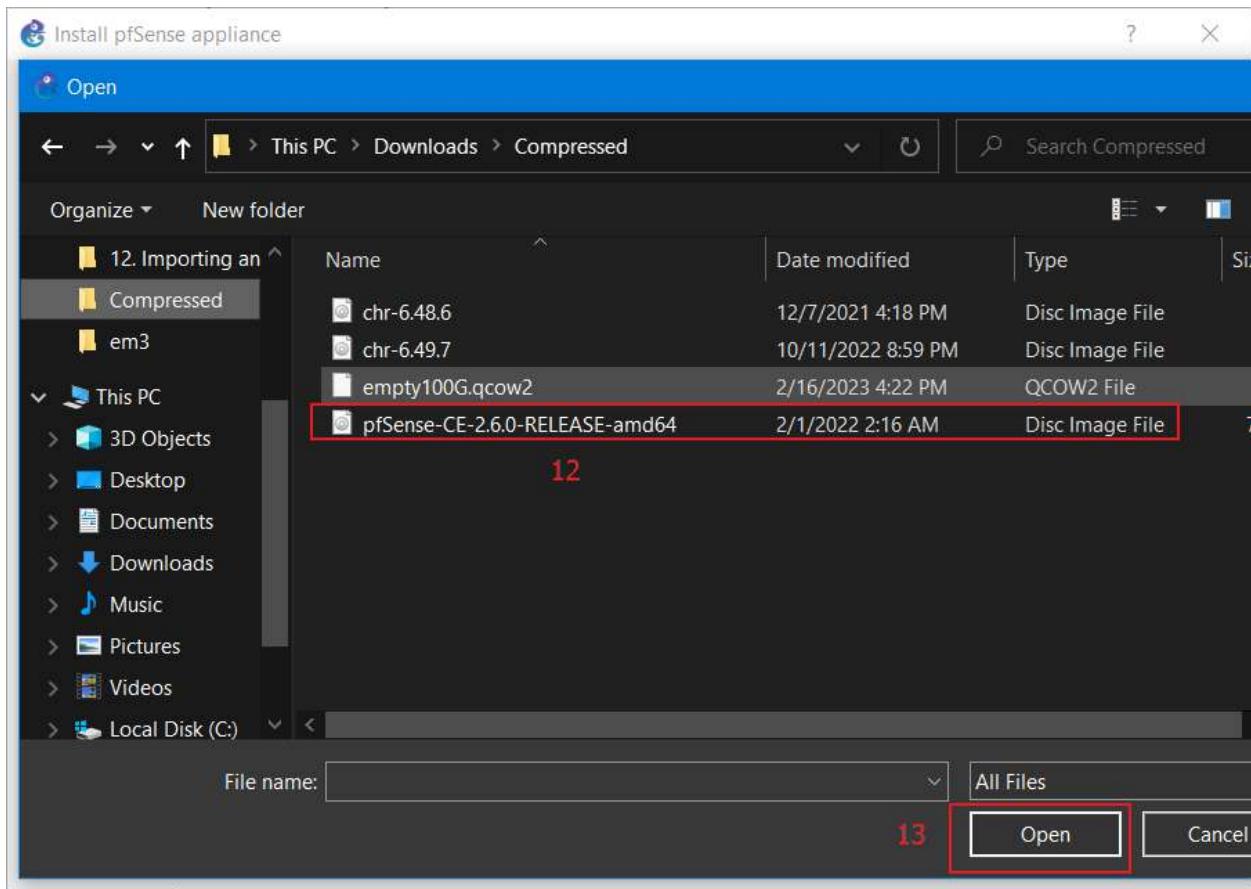


Figure 287: Importing Pfsense 8.

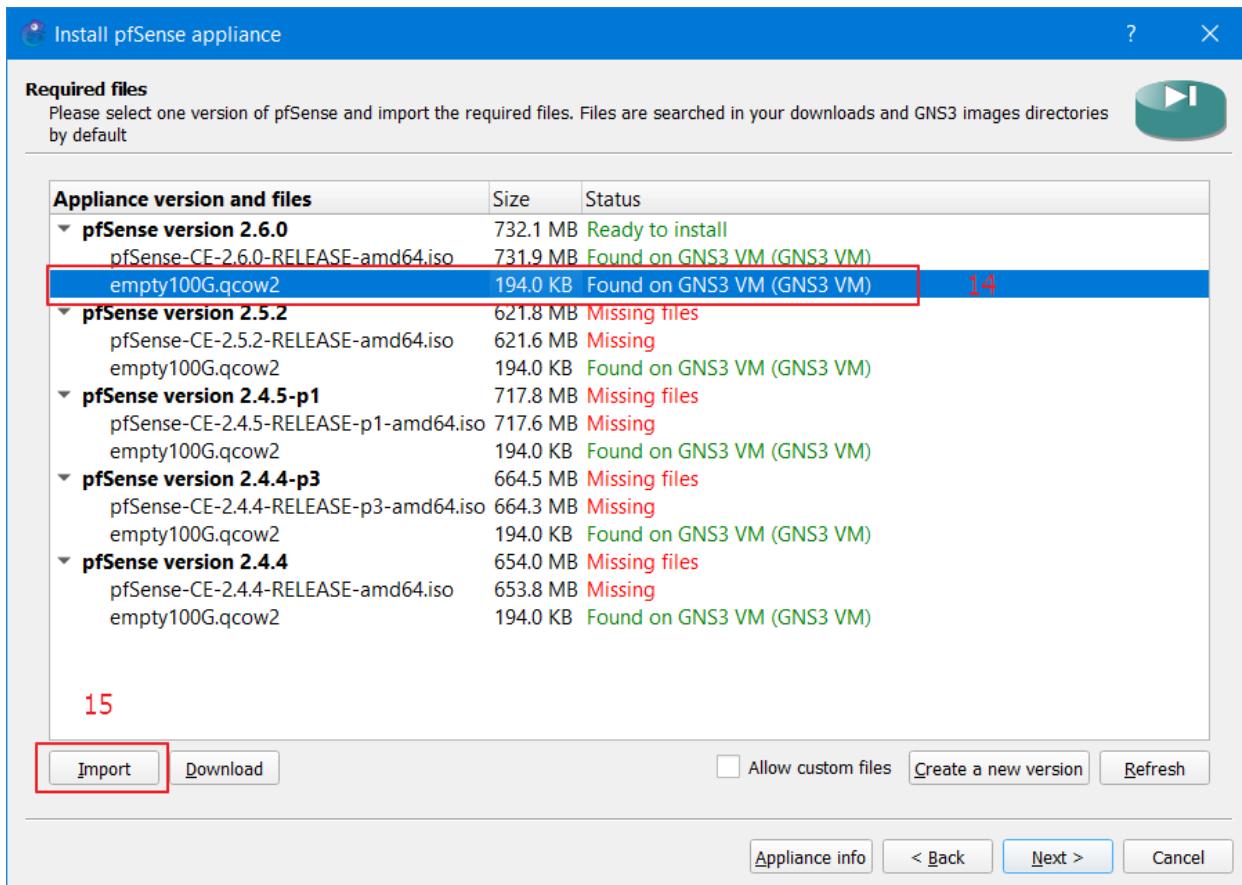


Figure 288: Importing Pfsense 9.

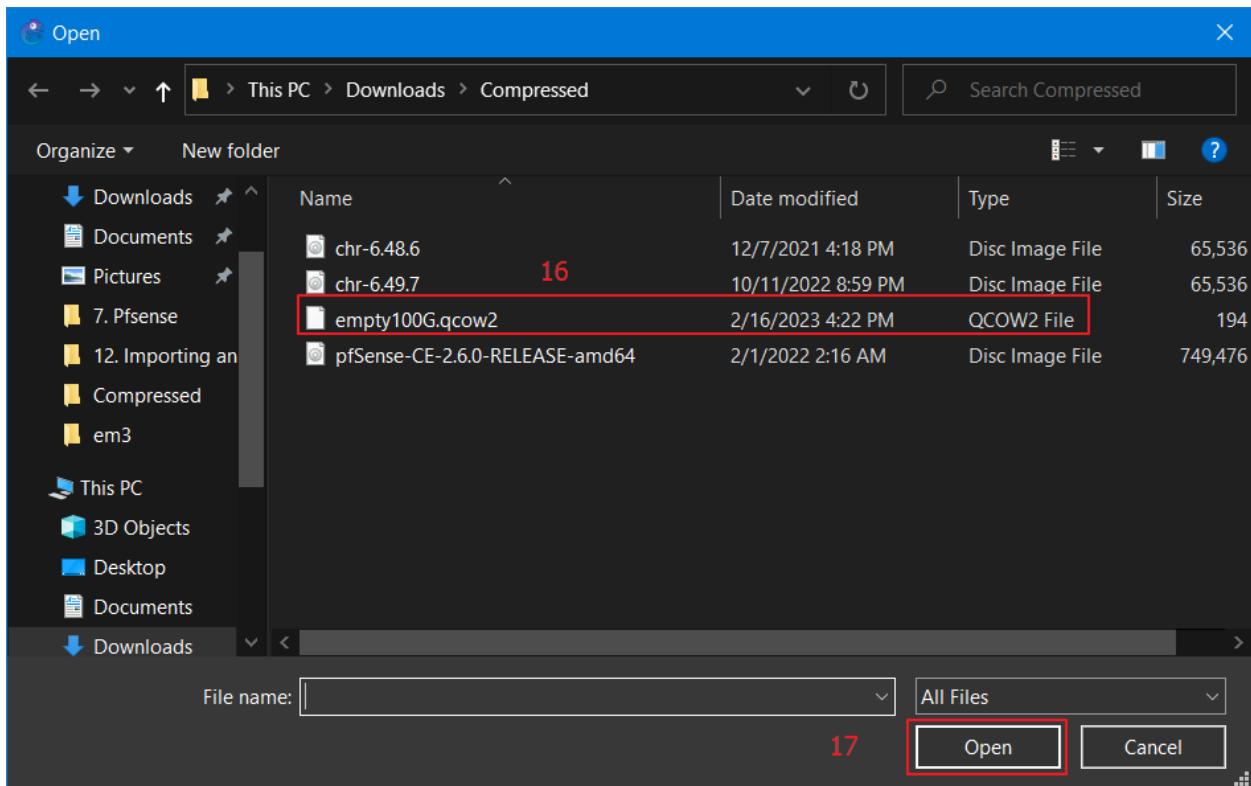


Figure 289: Importing PfSense 10.

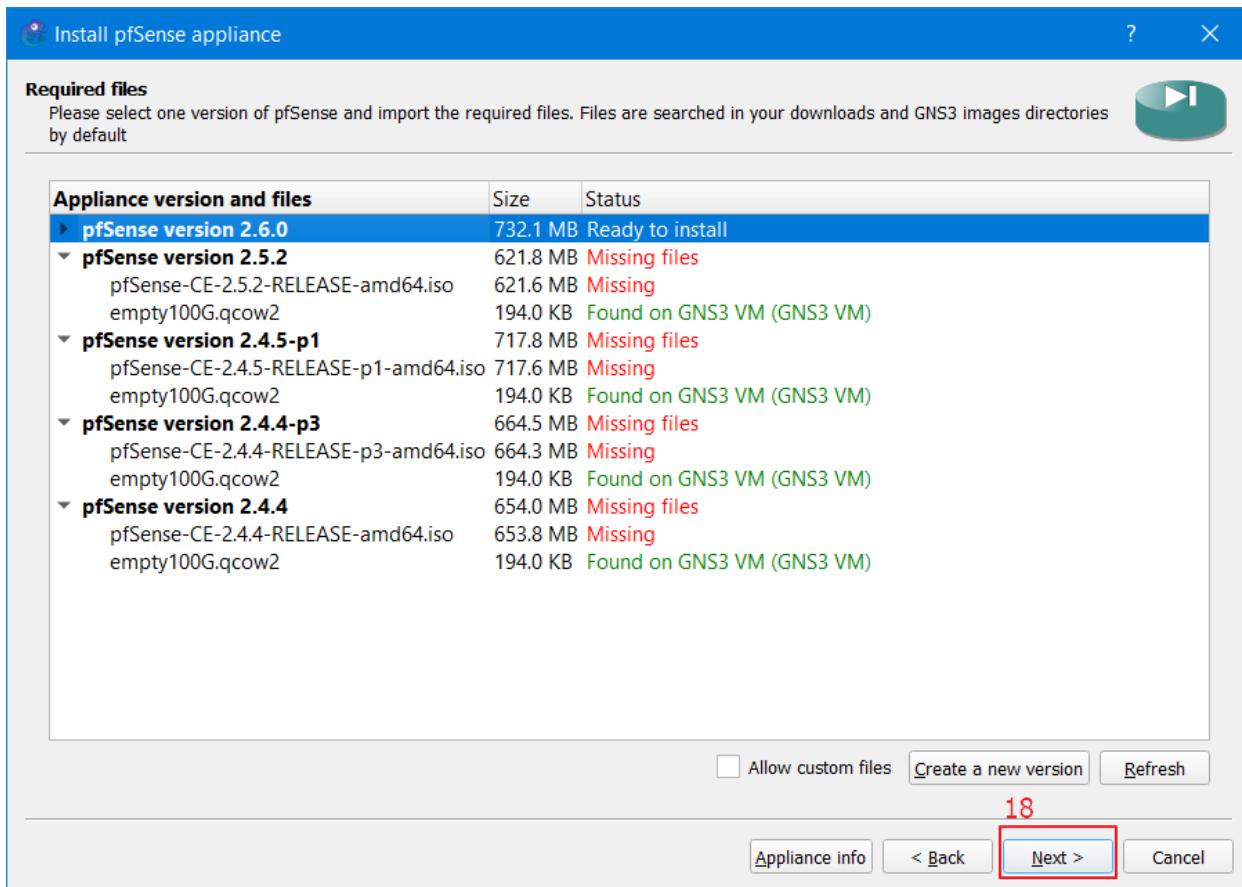


Figure 290: Importing Pfsense 13.

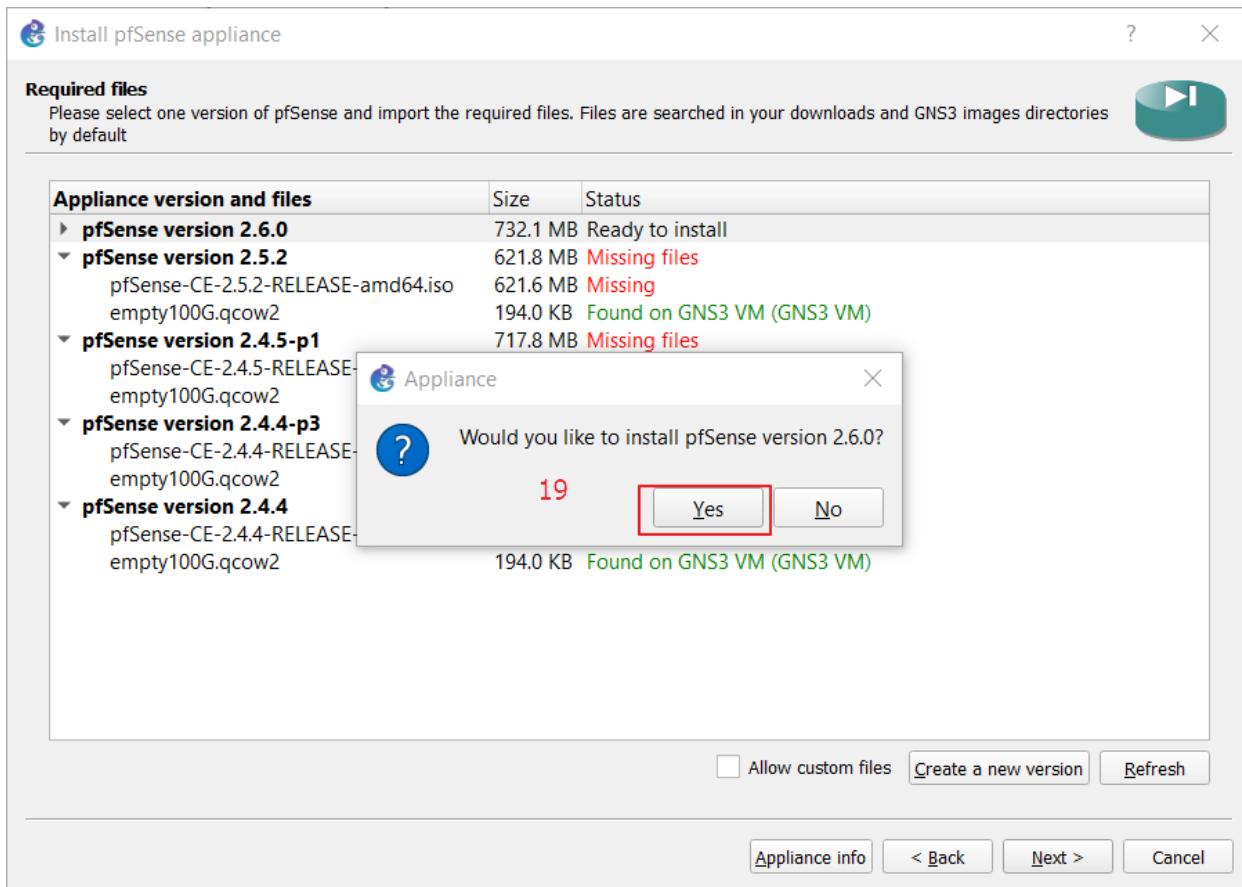


Figure 291: Importing Pfsense 14.

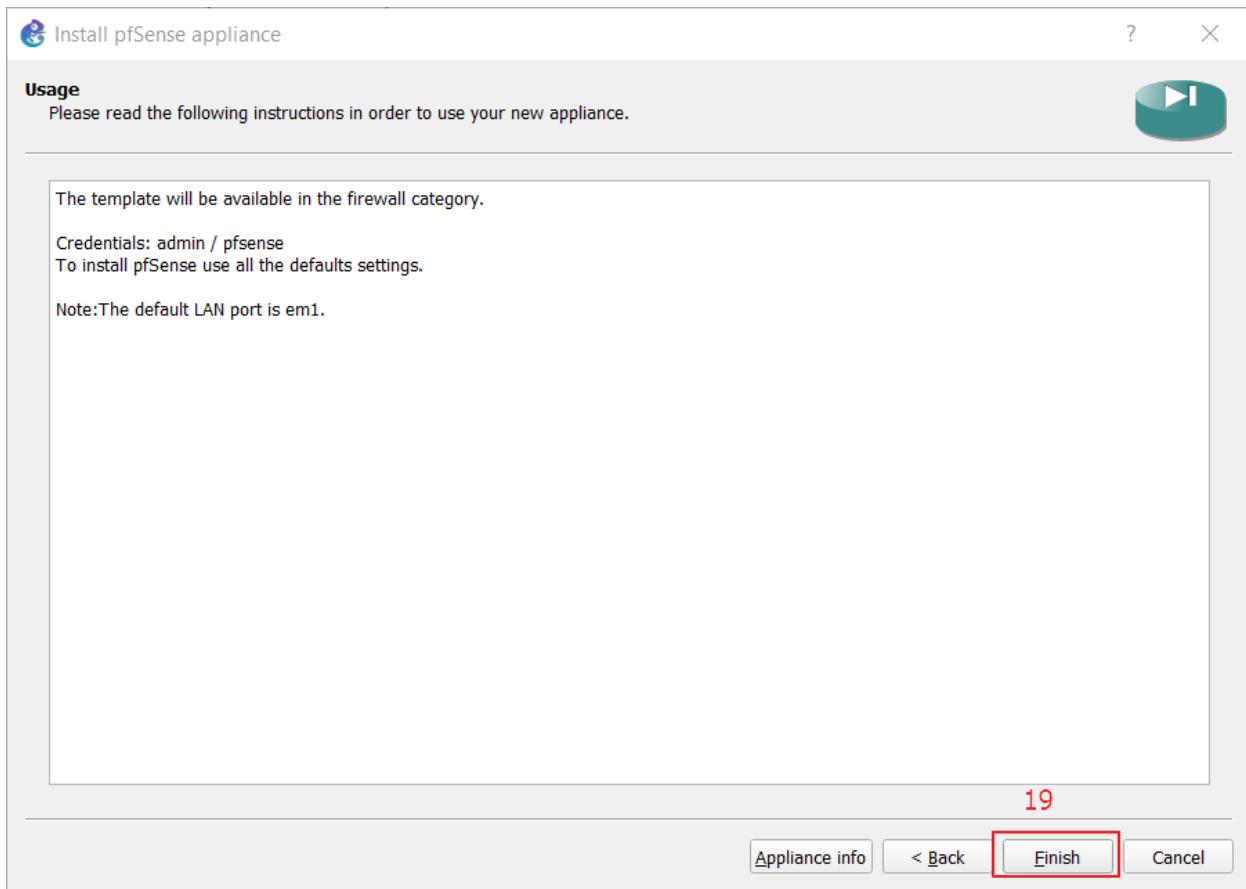


Figure 292: Importing Pfsense 15.

ELKMemcached Server

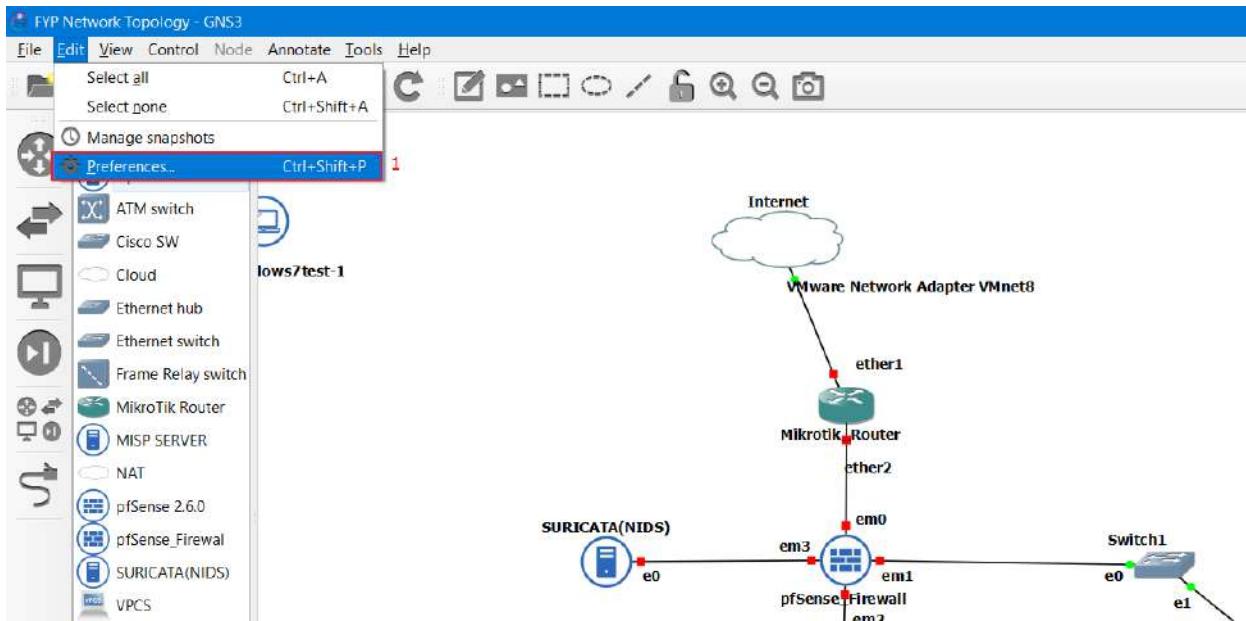


Figure 293: Importing ELKMemcached server 1.

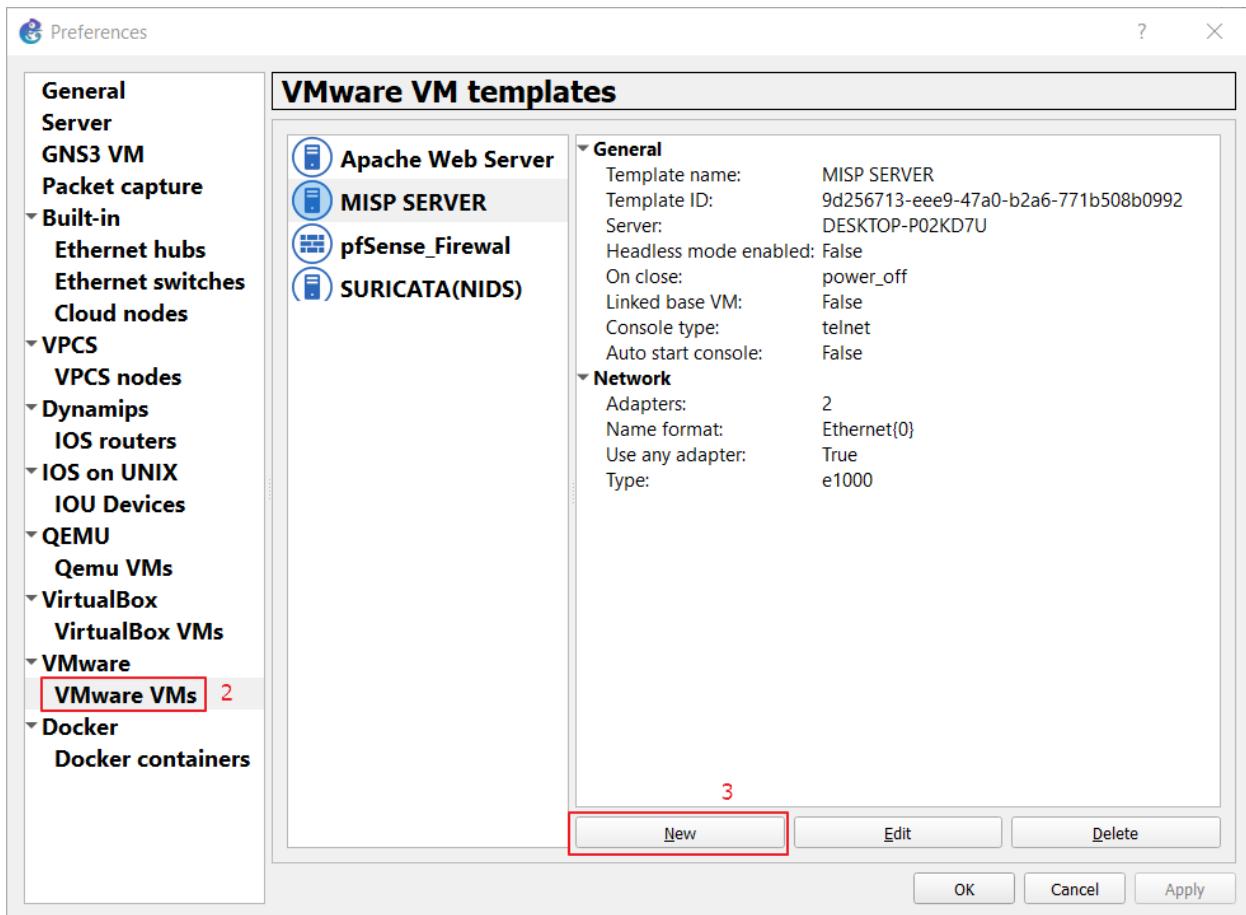


Figure 294: Importing ELKMemcached server 2.

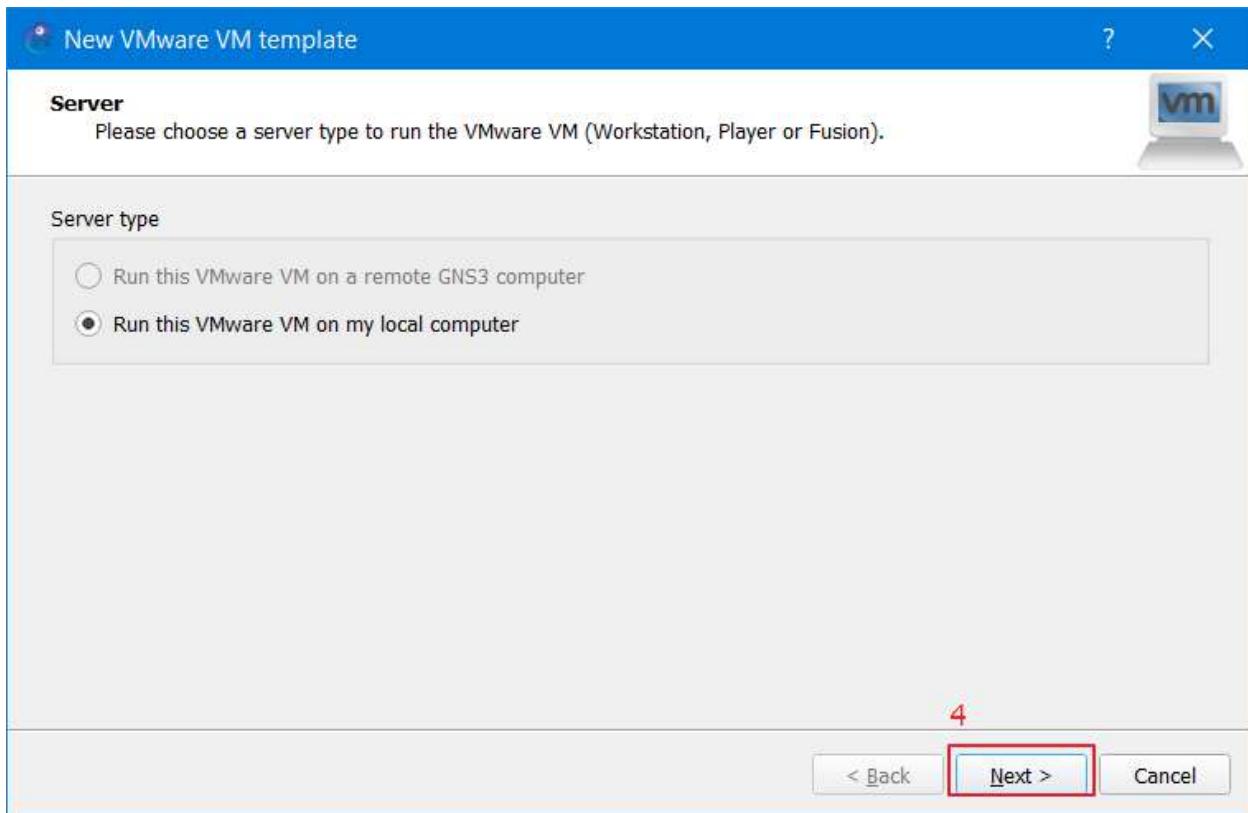


Figure 295: Importing ELKMemcached server 3.

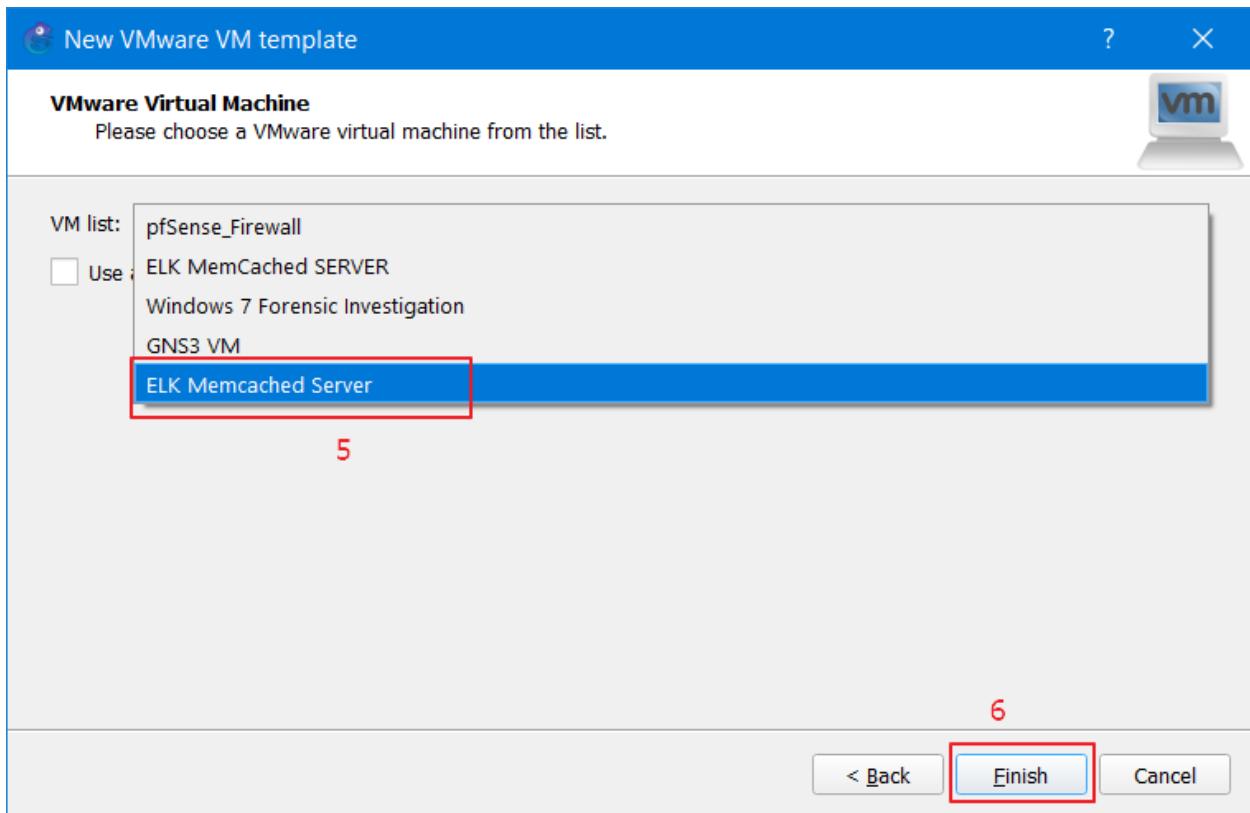


Figure 296: Importing ELKMemcached server 4.

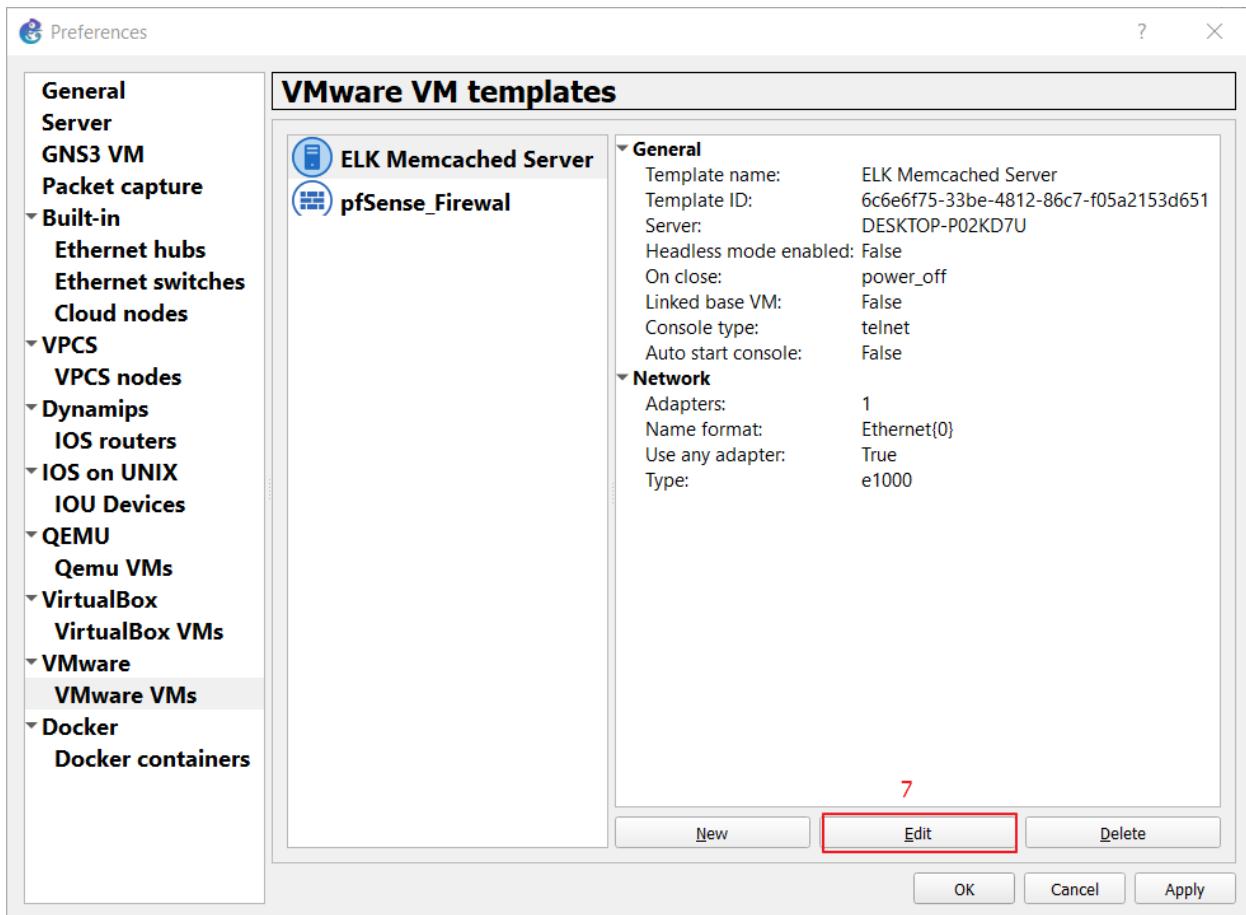


Figure 297: Importing ELKMemcached server 5.

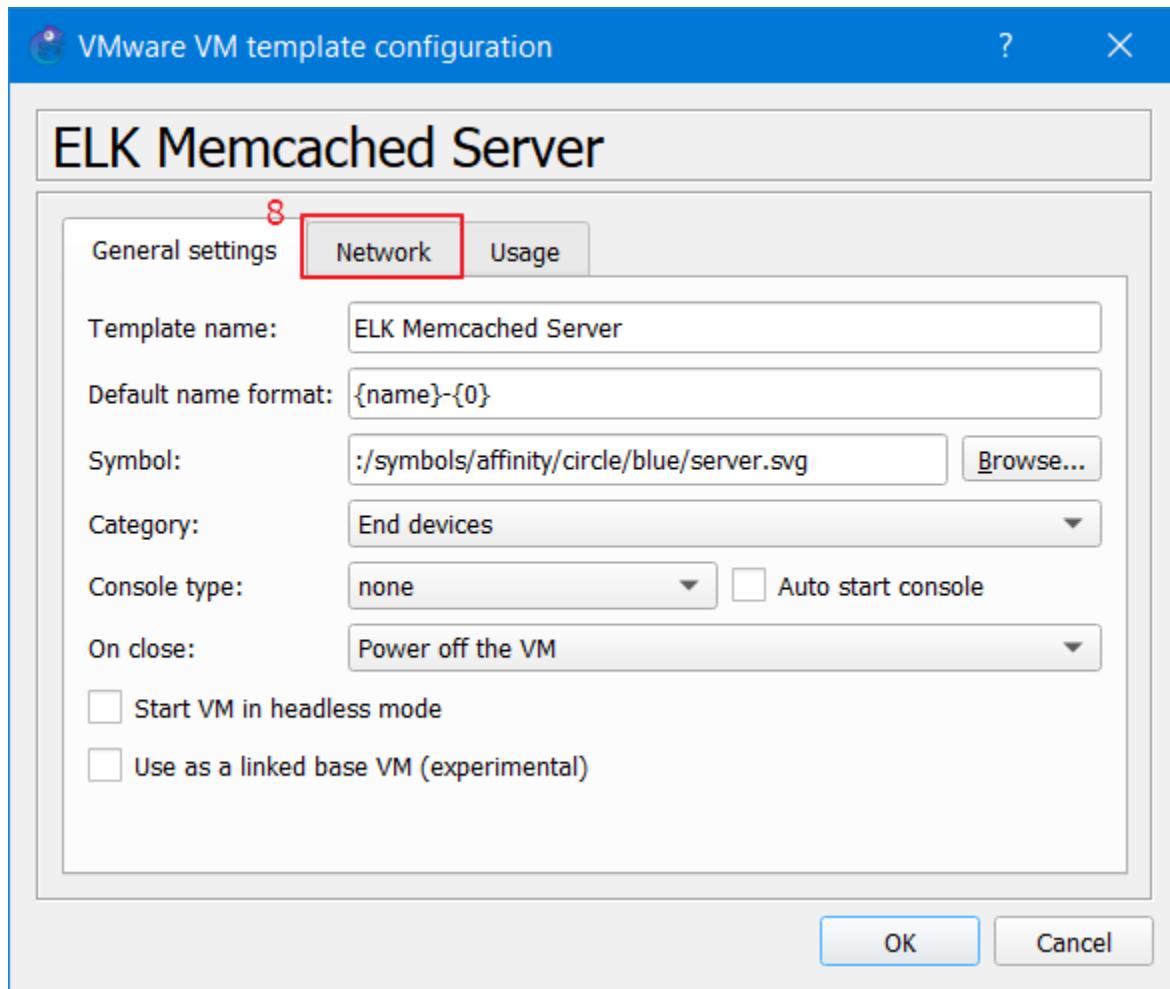


Figure 298: Importing ELKMemcached server 6.

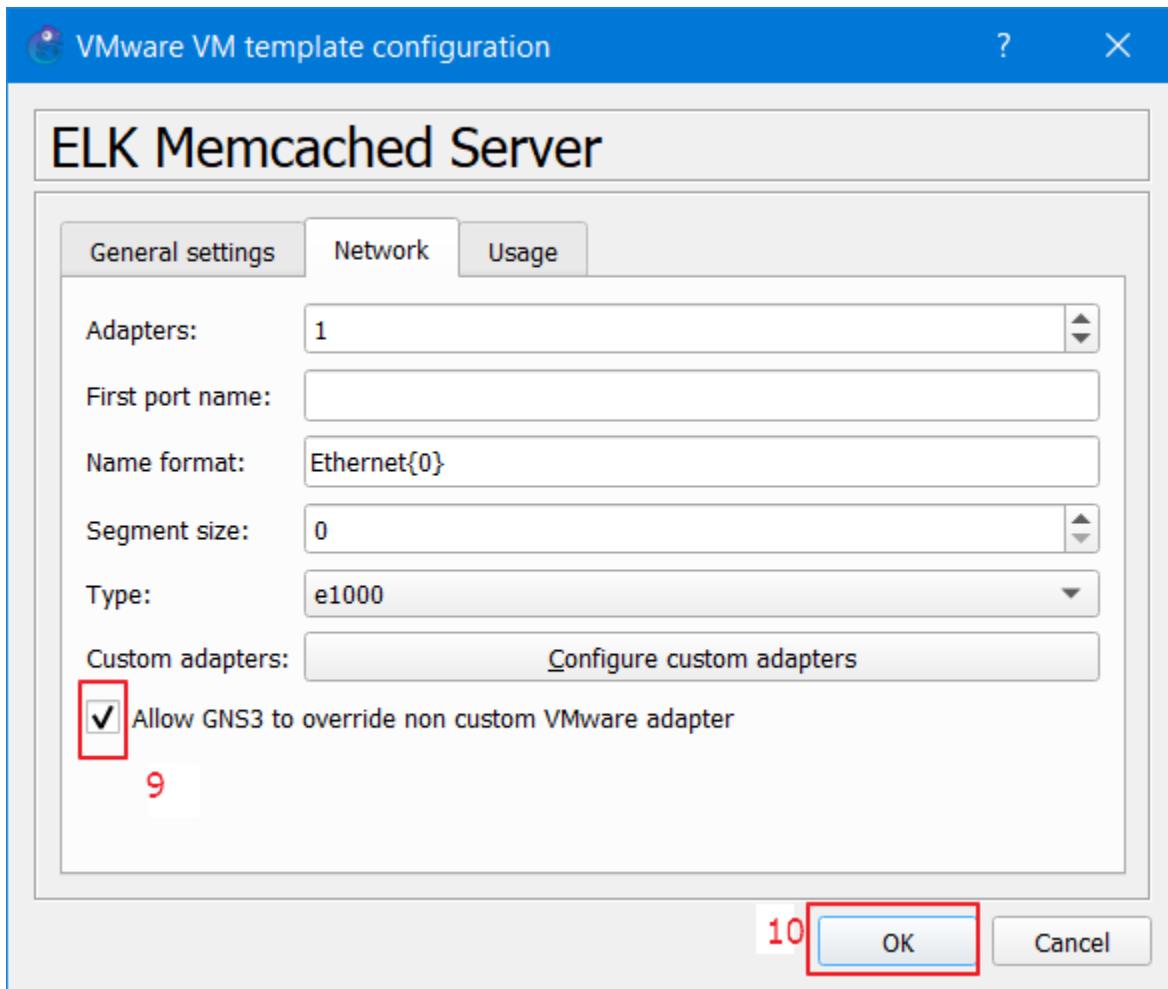


Figure 299: Importing ELKMemcached server 7.

Suricata NIDS

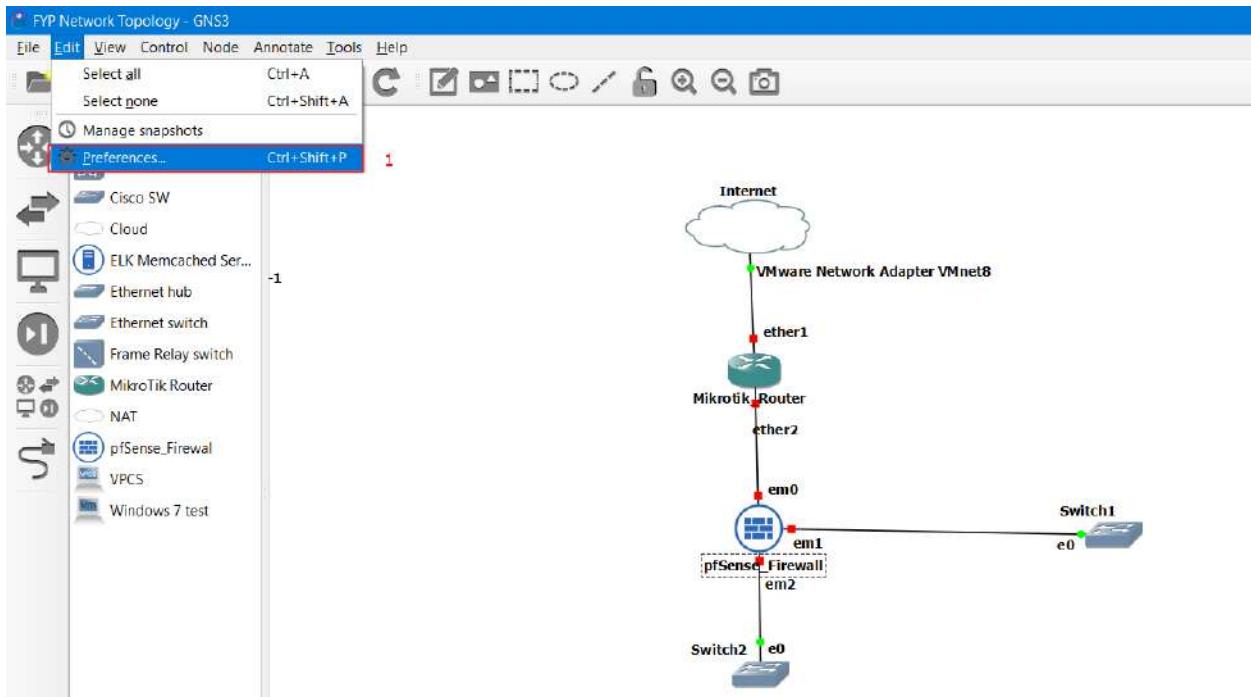


Figure 300: Importing suricata VM server 1.

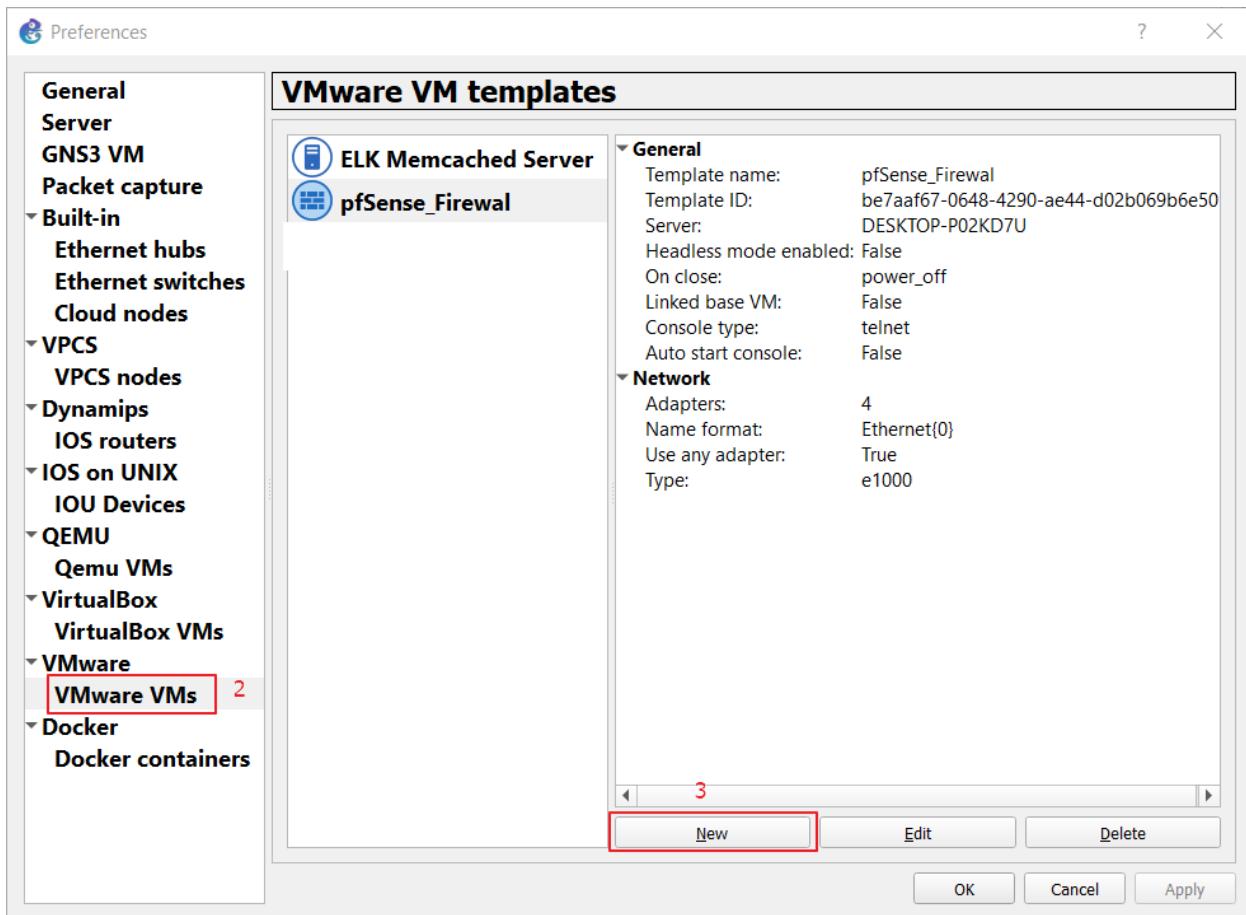


Figure 301: Importing suricata VM server 2.

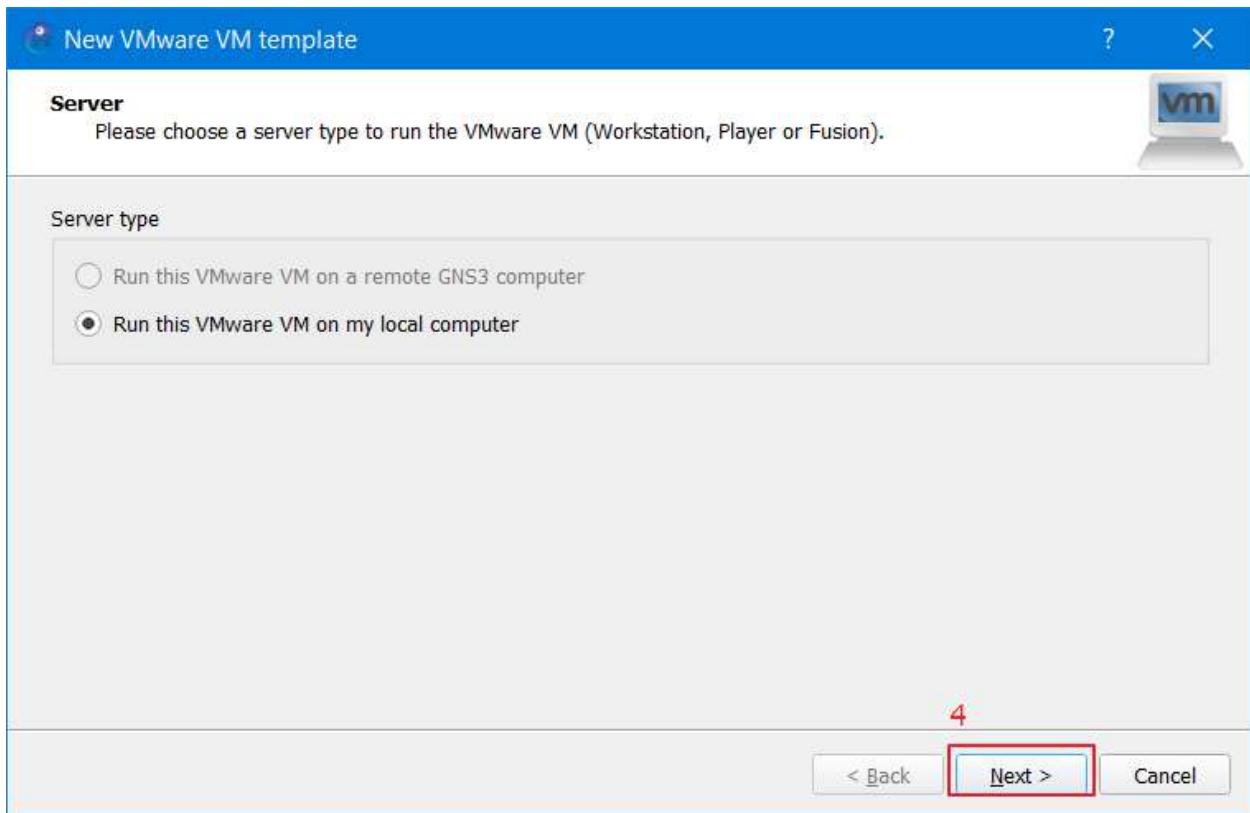


Figure 302: Importing suricata VM server 3.

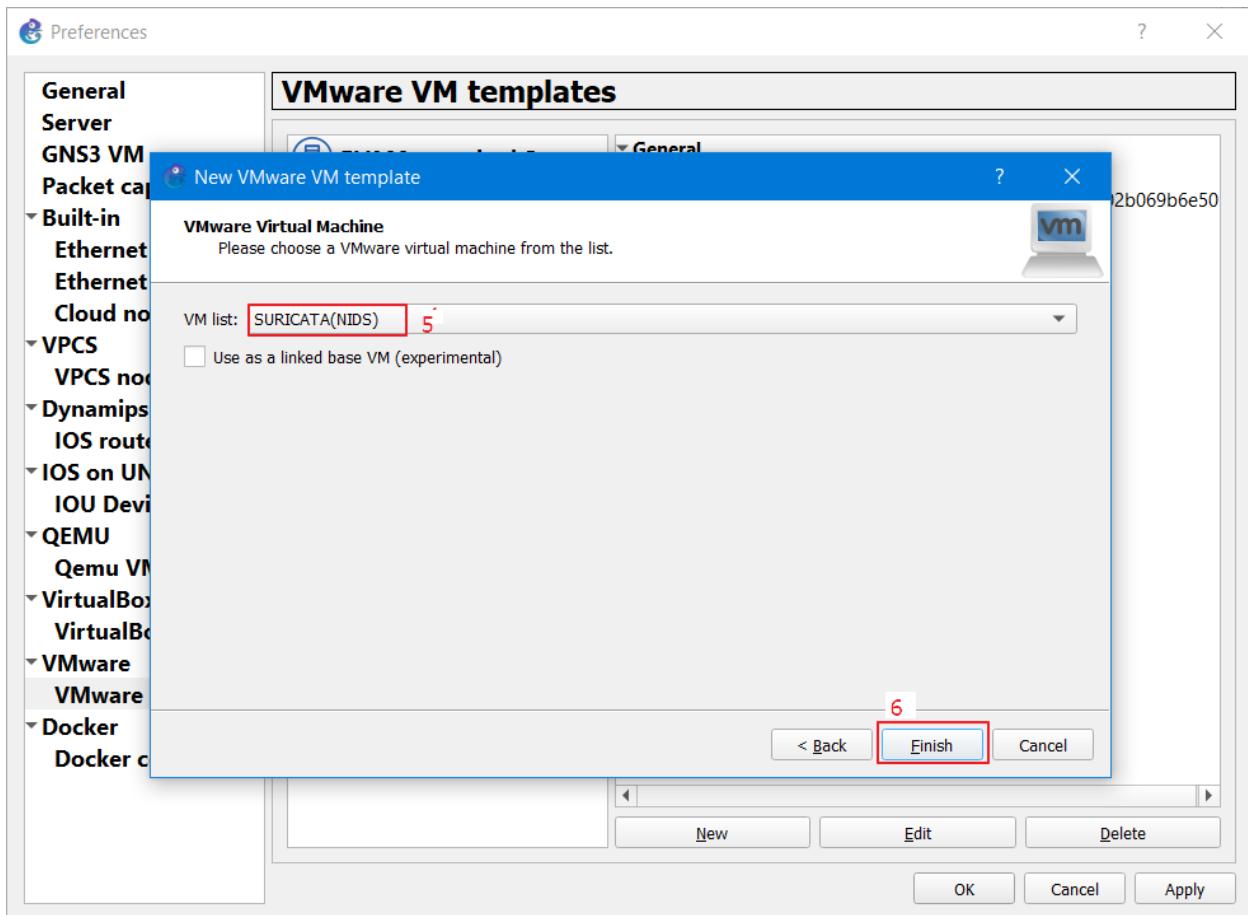


Figure 303: Importing suricata VM server 4.

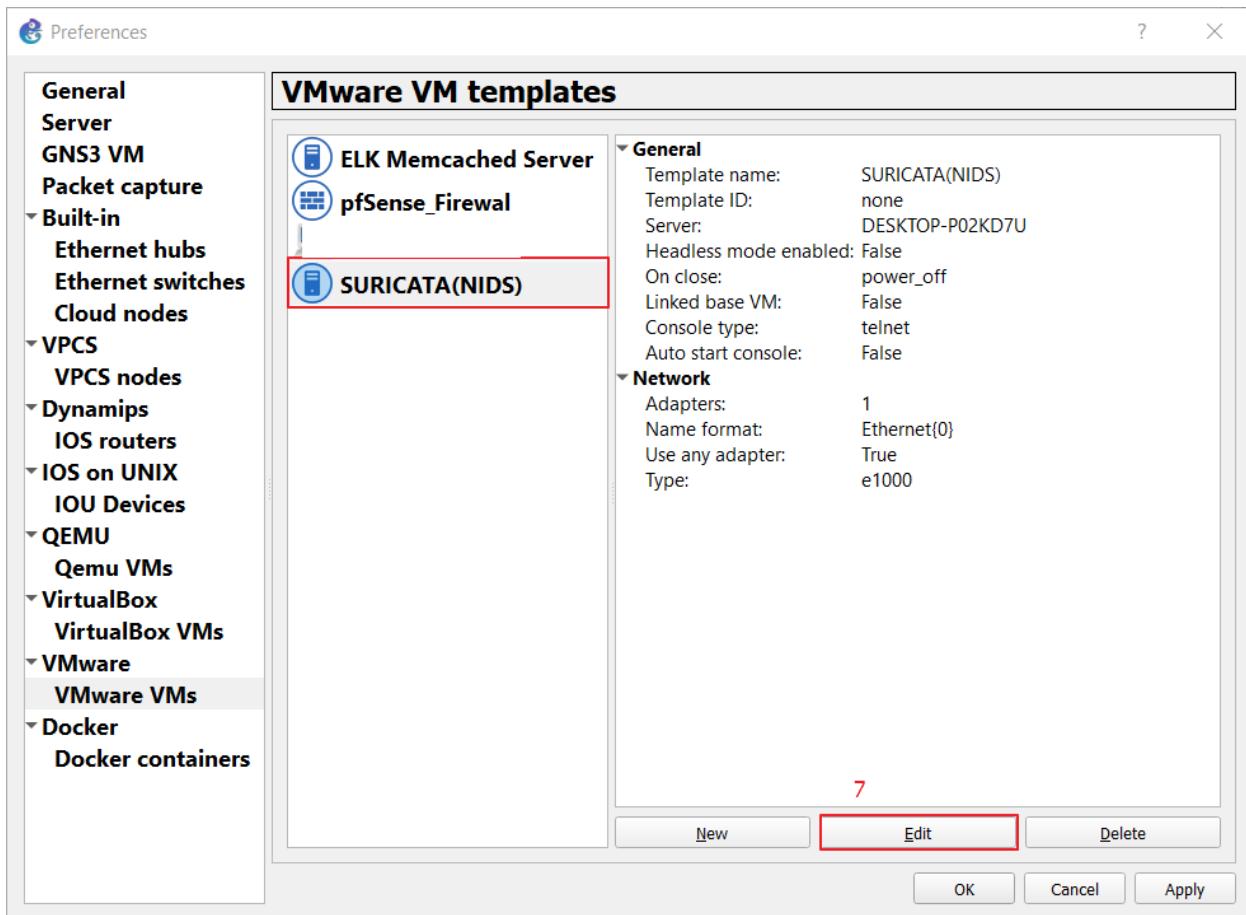


Figure 304: Importing suricata VM server 5.

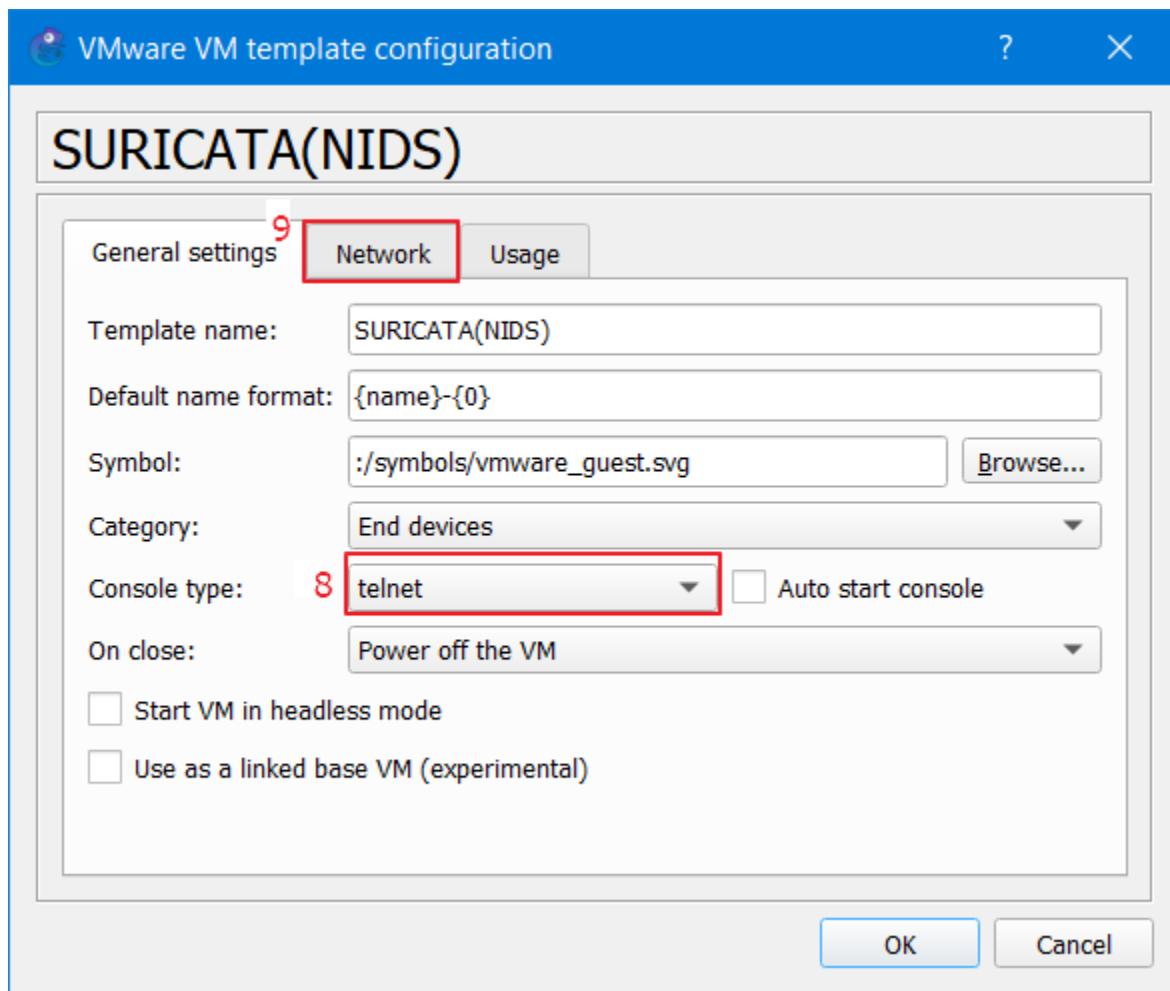


Figure 305: Importing suricata VM server 6.

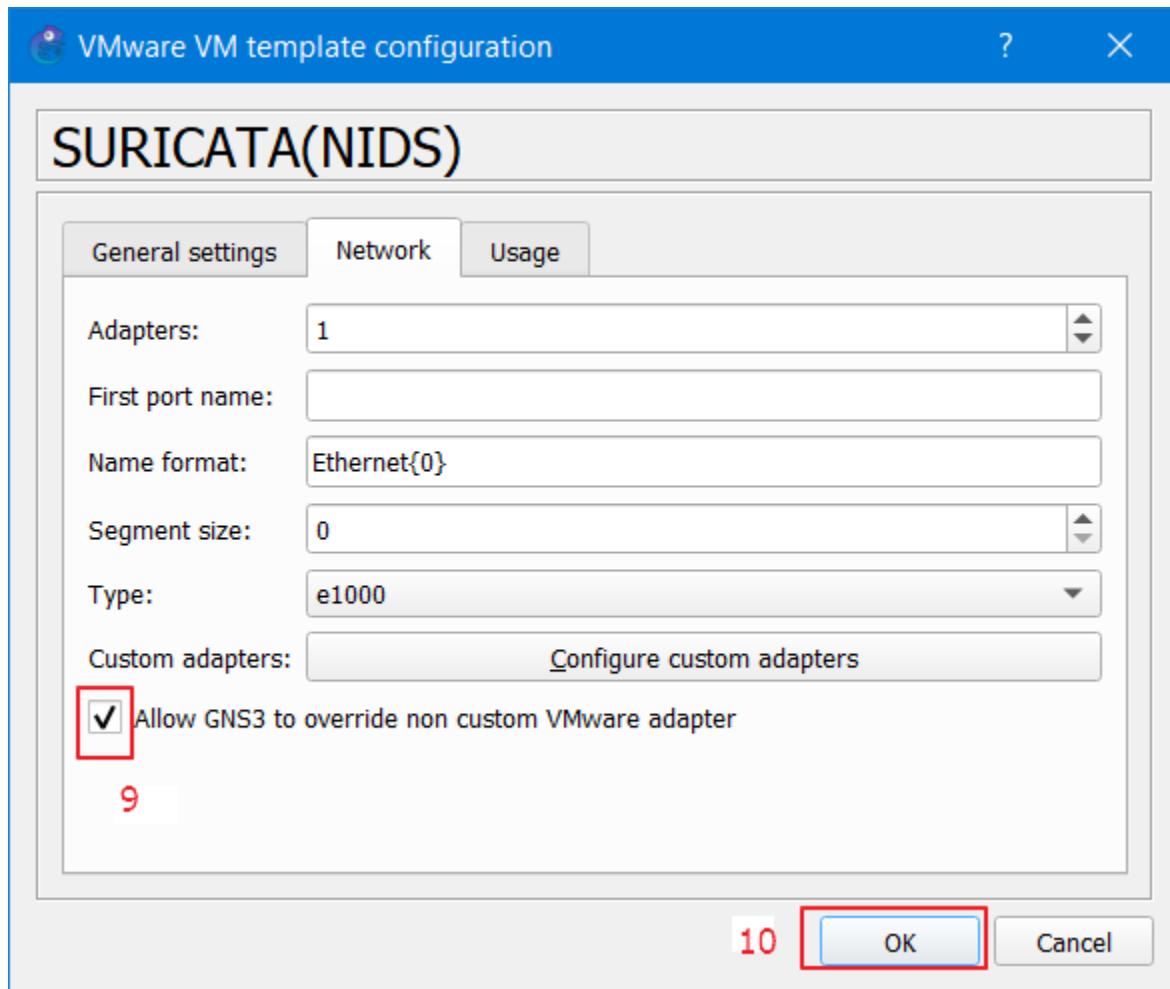


Figure 306: Importing suricata VM server 7.

MISP Server

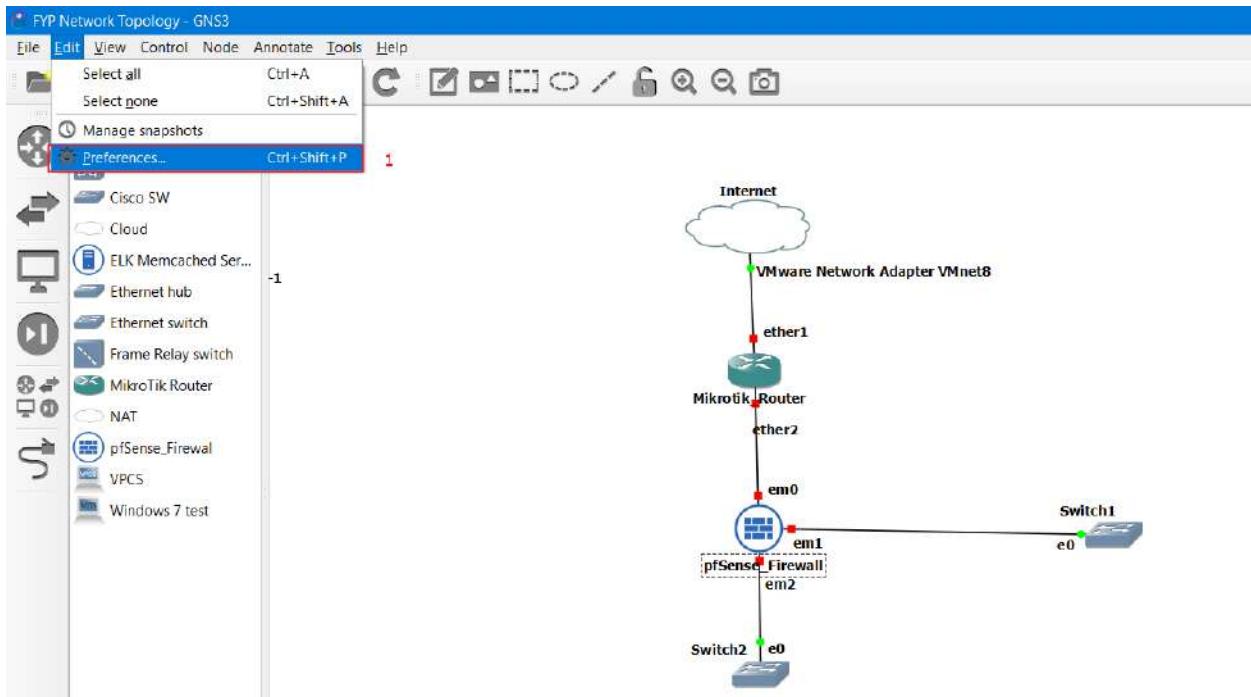


Figure 307: Importing MISP VM server 1.

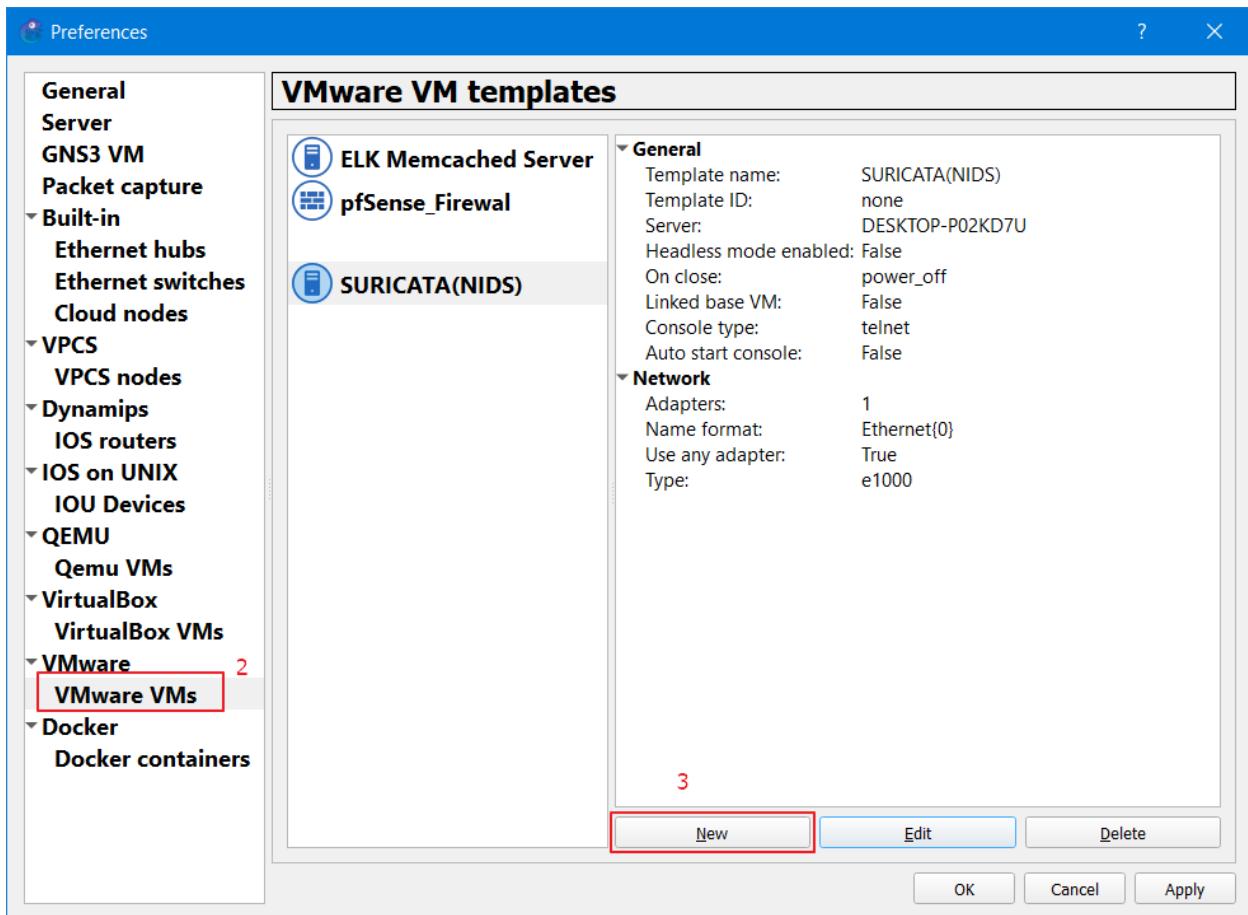


Figure 308: Importing MISP VM server 2..

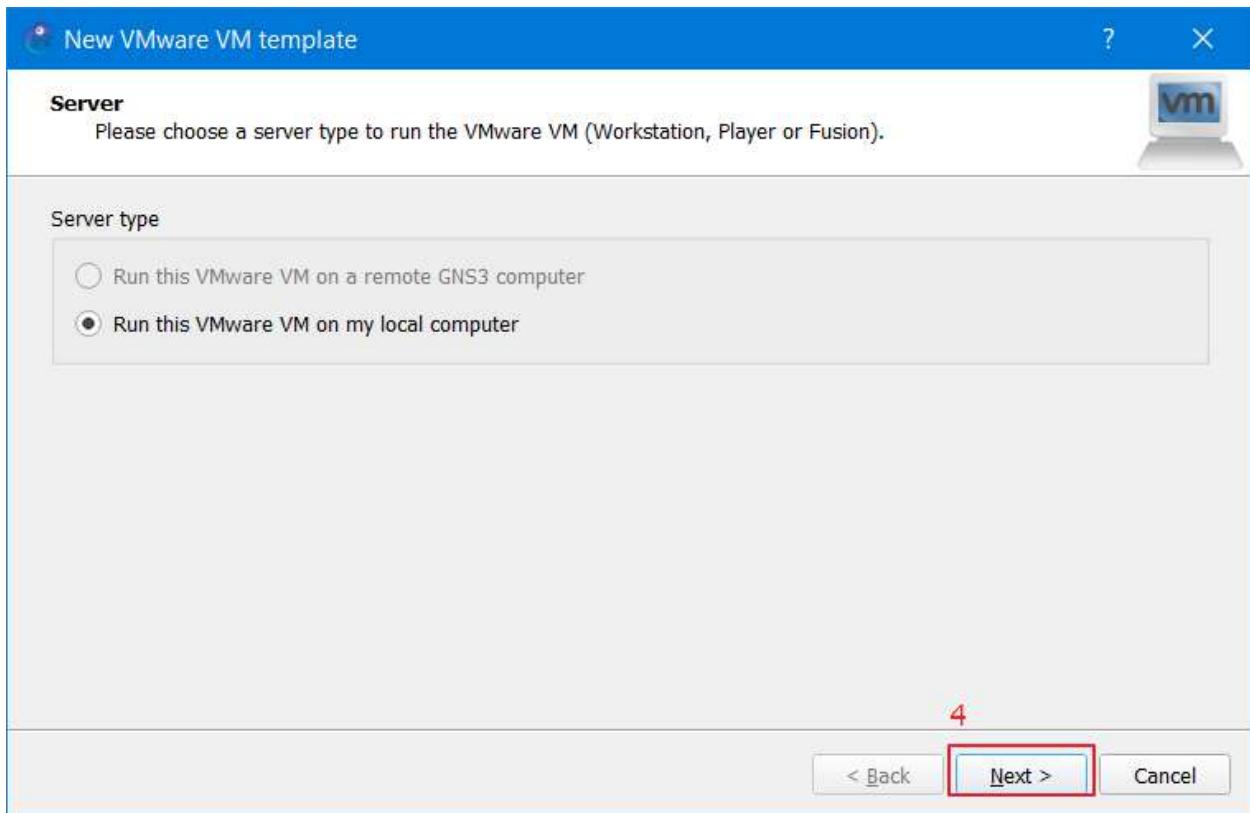


Figure 309: Importing MISP VM server 3.

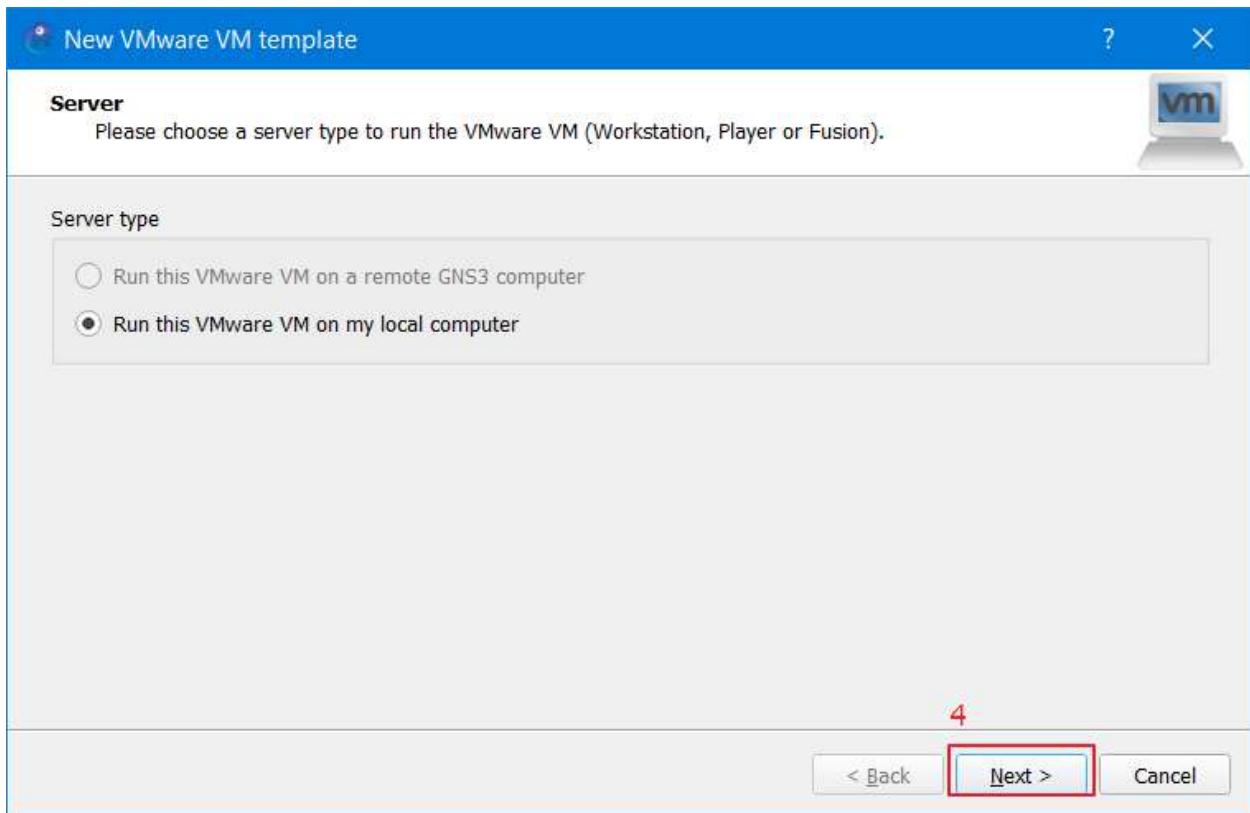


Figure 310: Importing MISP VM server 4.

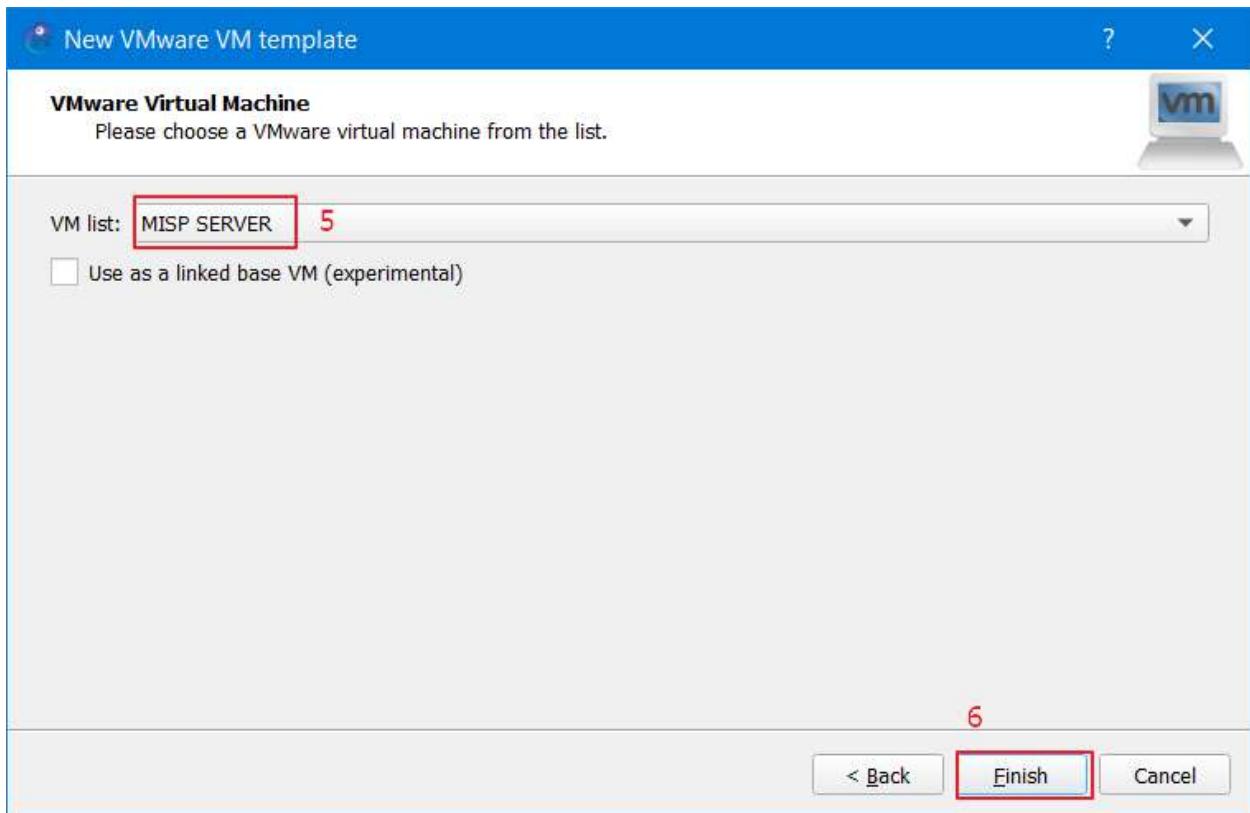


Figure 311: Importing MISP VM server 5.

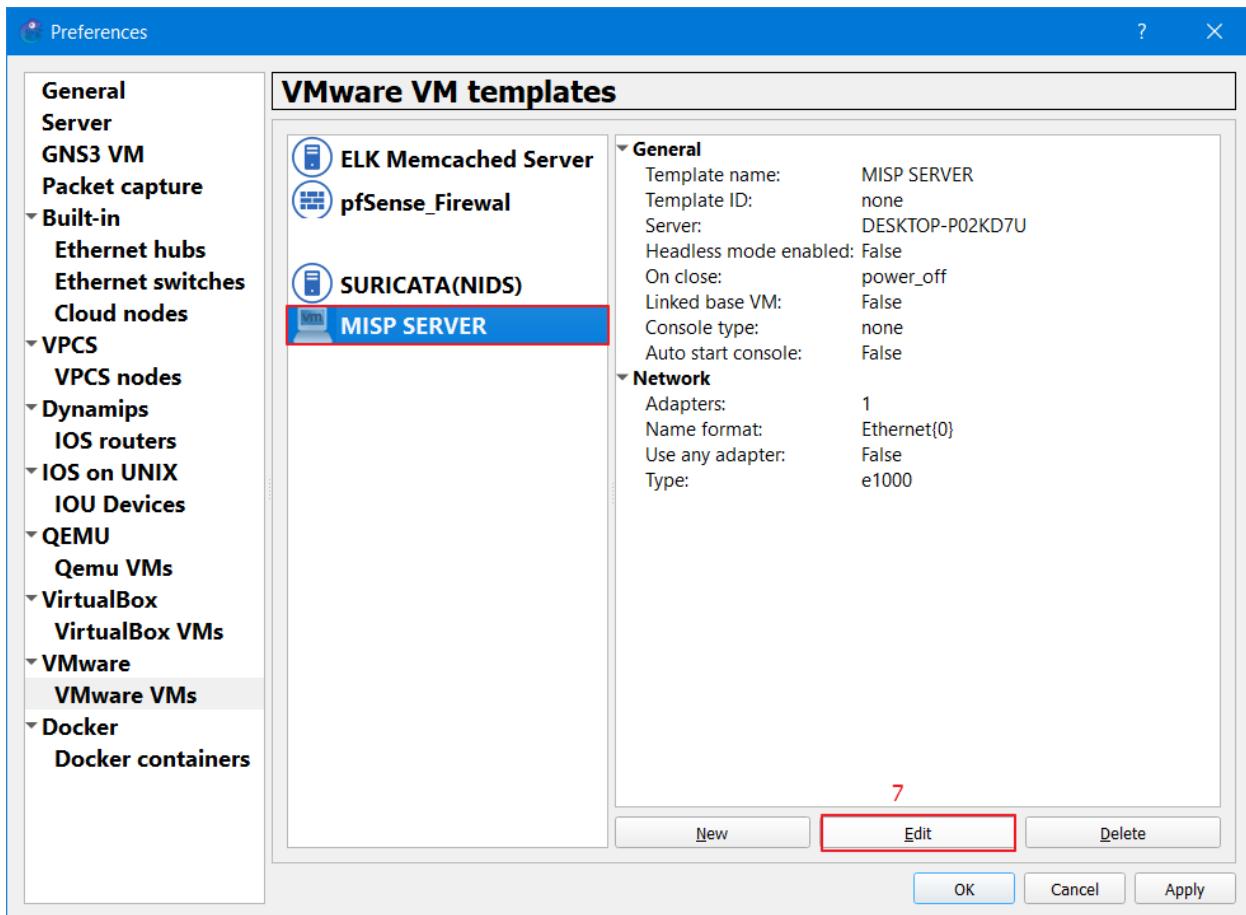


Figure 312: Importing MISP VM server 6.

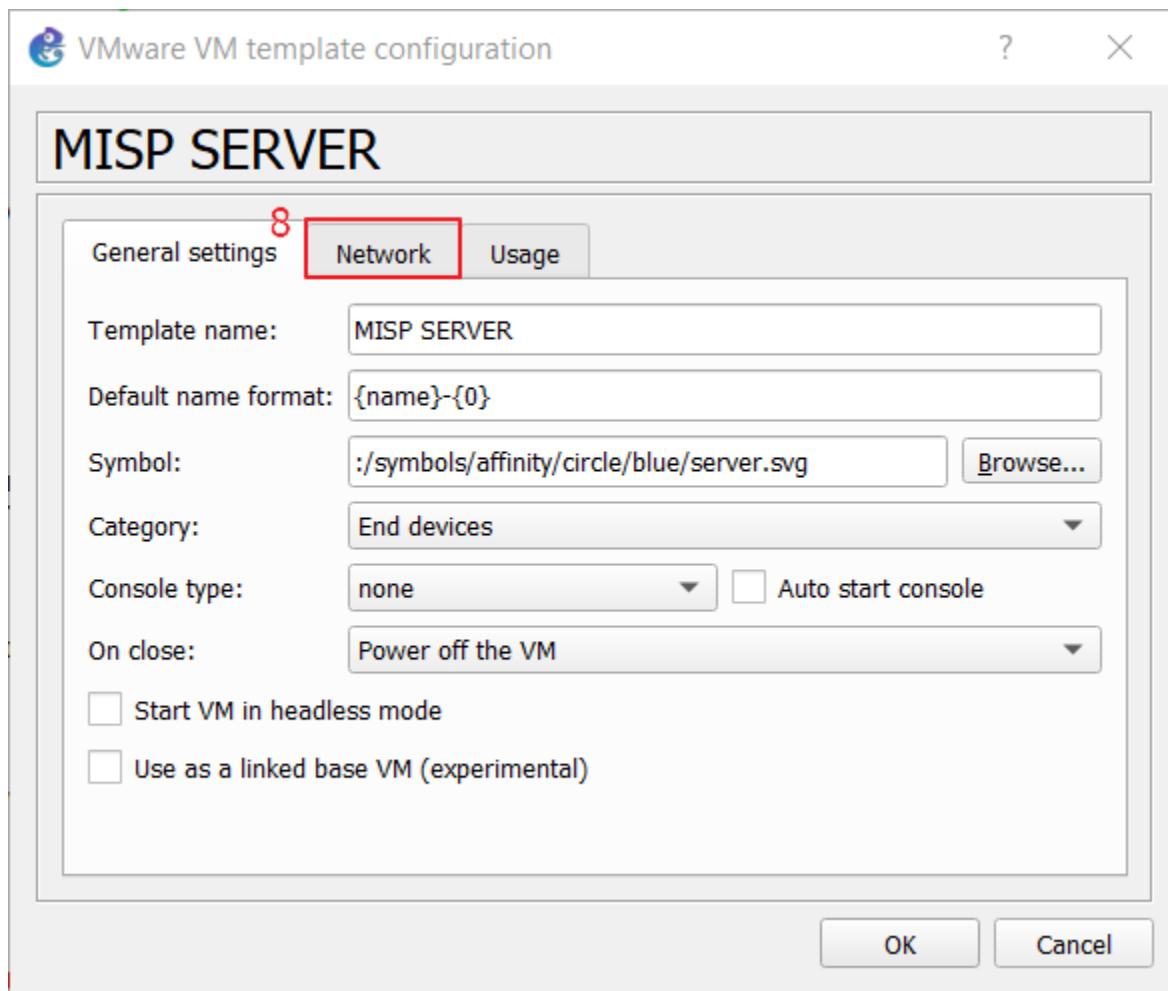


Figure 313: Importing MISP VM server 7.

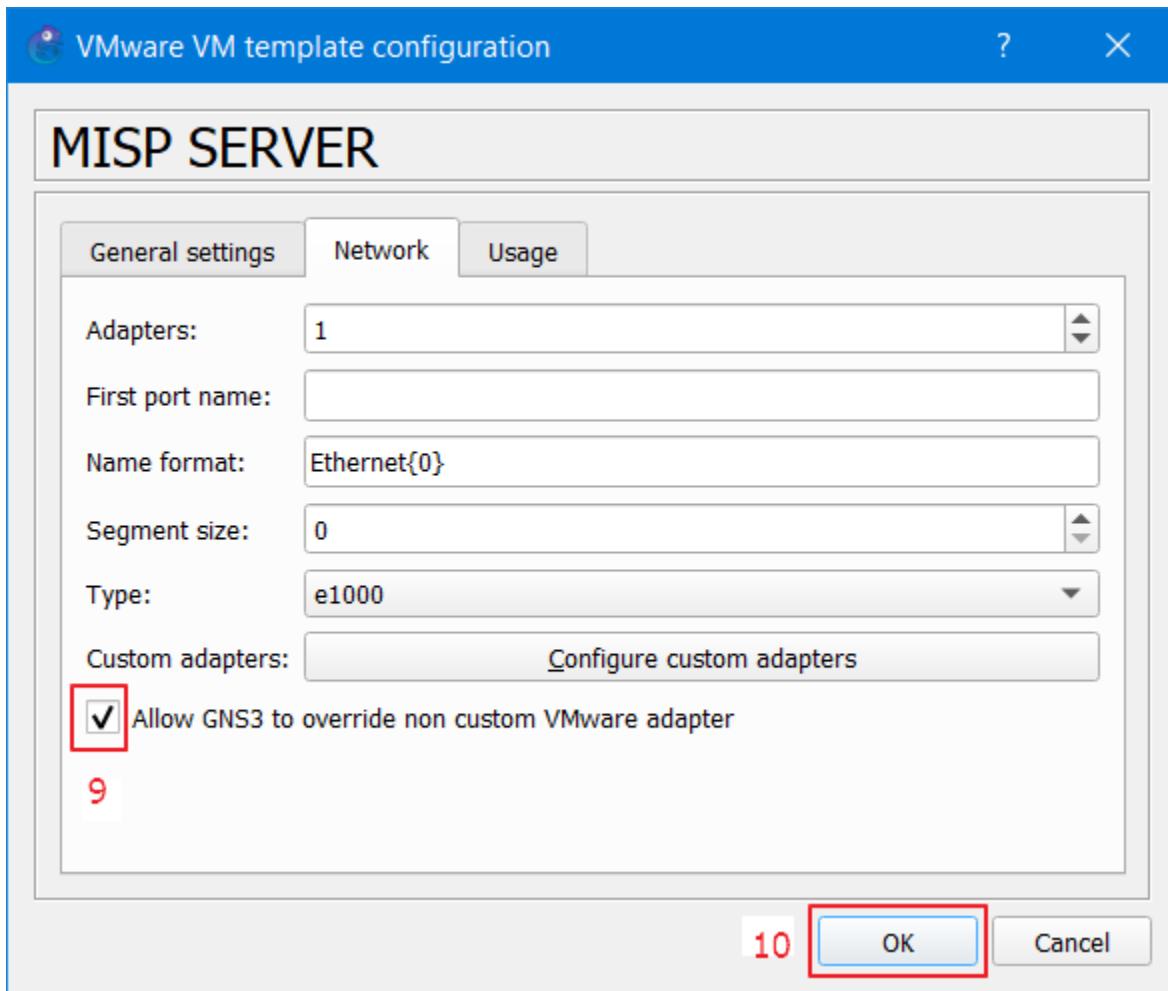


Figure 314: Importing MISP VM server 8.

Apache Web server

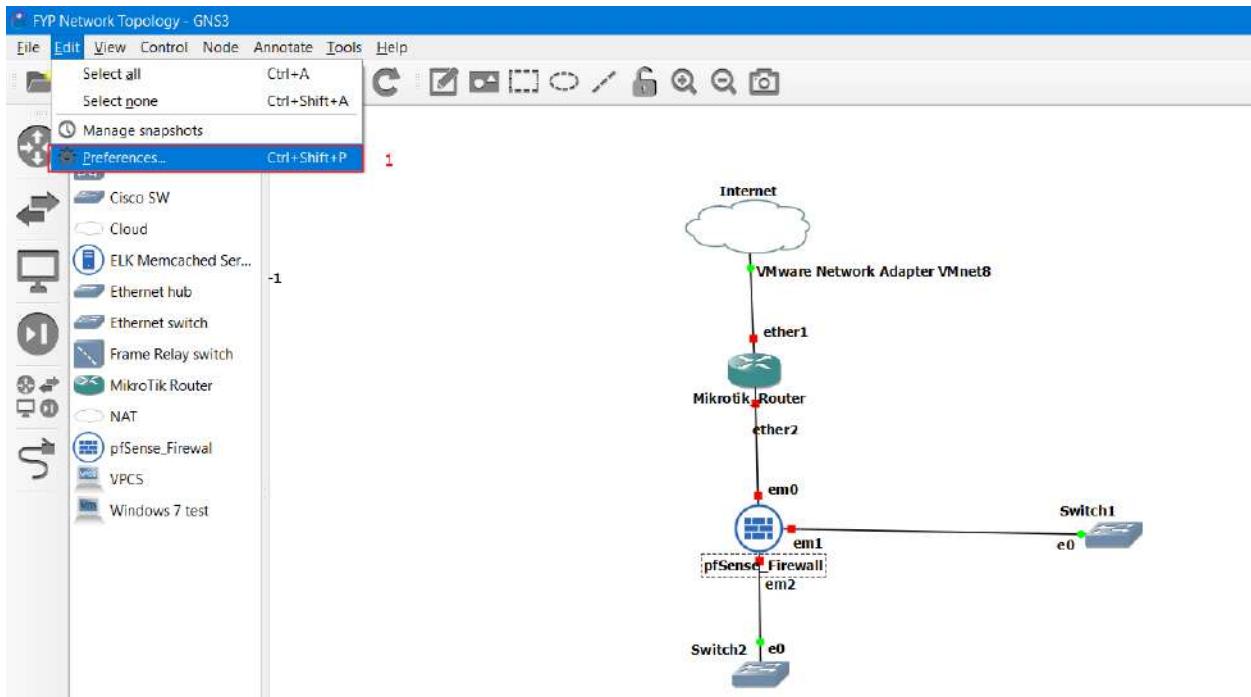


Figure 315: Import Apache server 1.

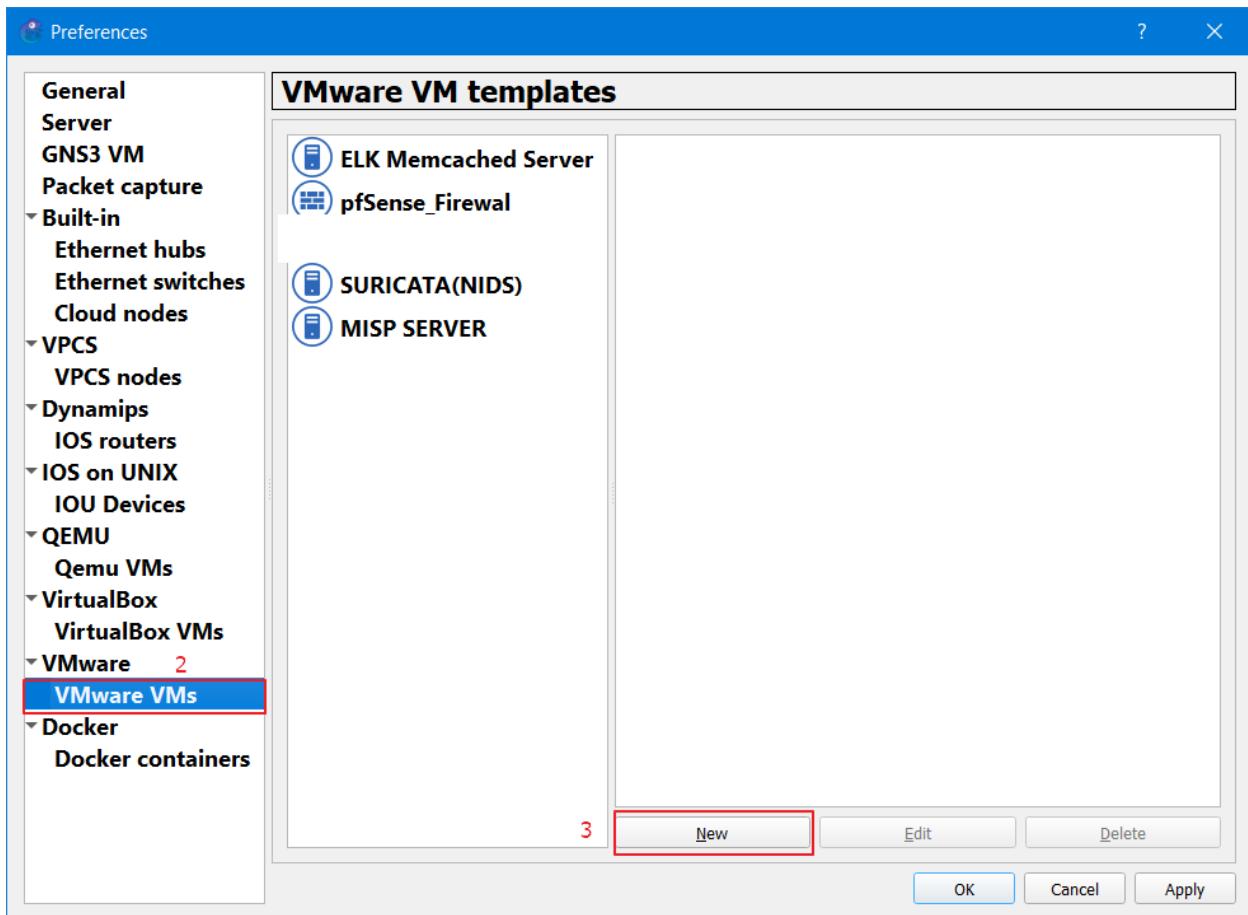


Figure 316: Import Apache server 2.

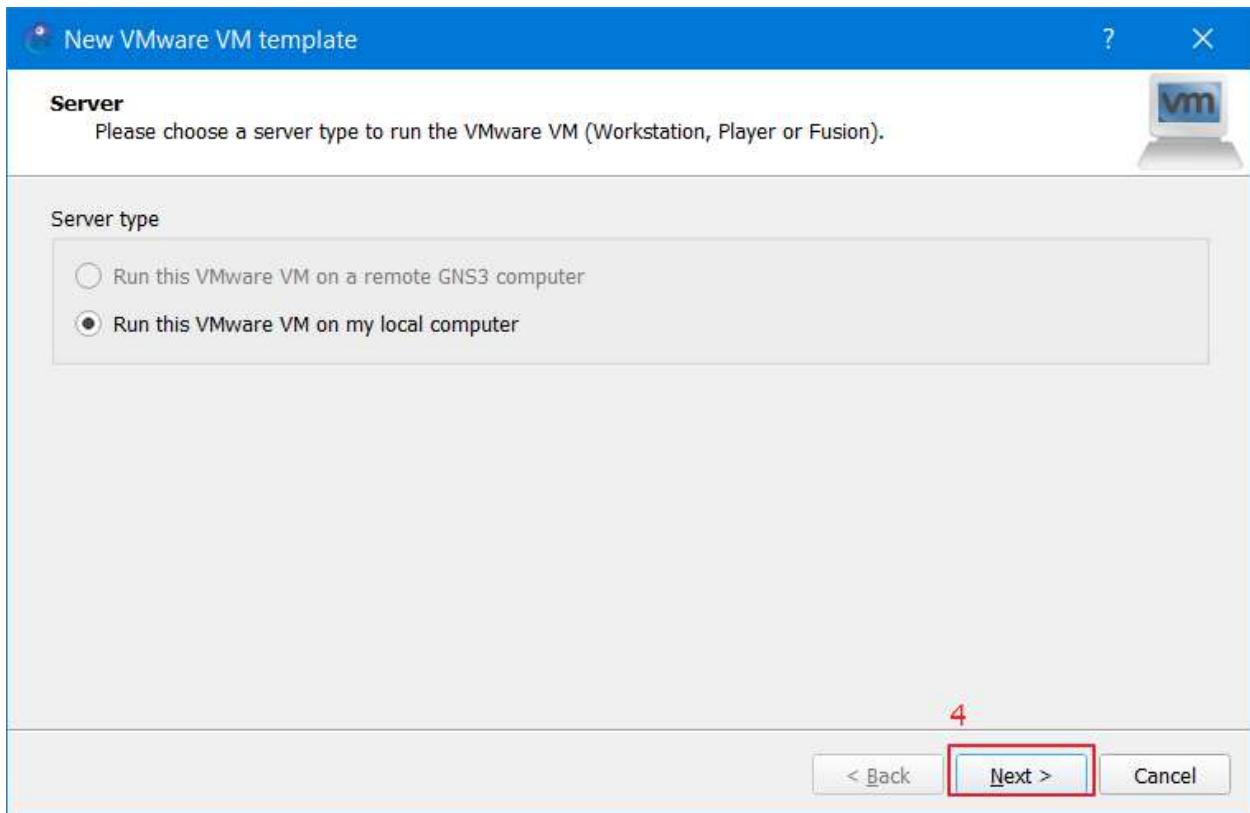


Figure 317: Import Apache server 3.

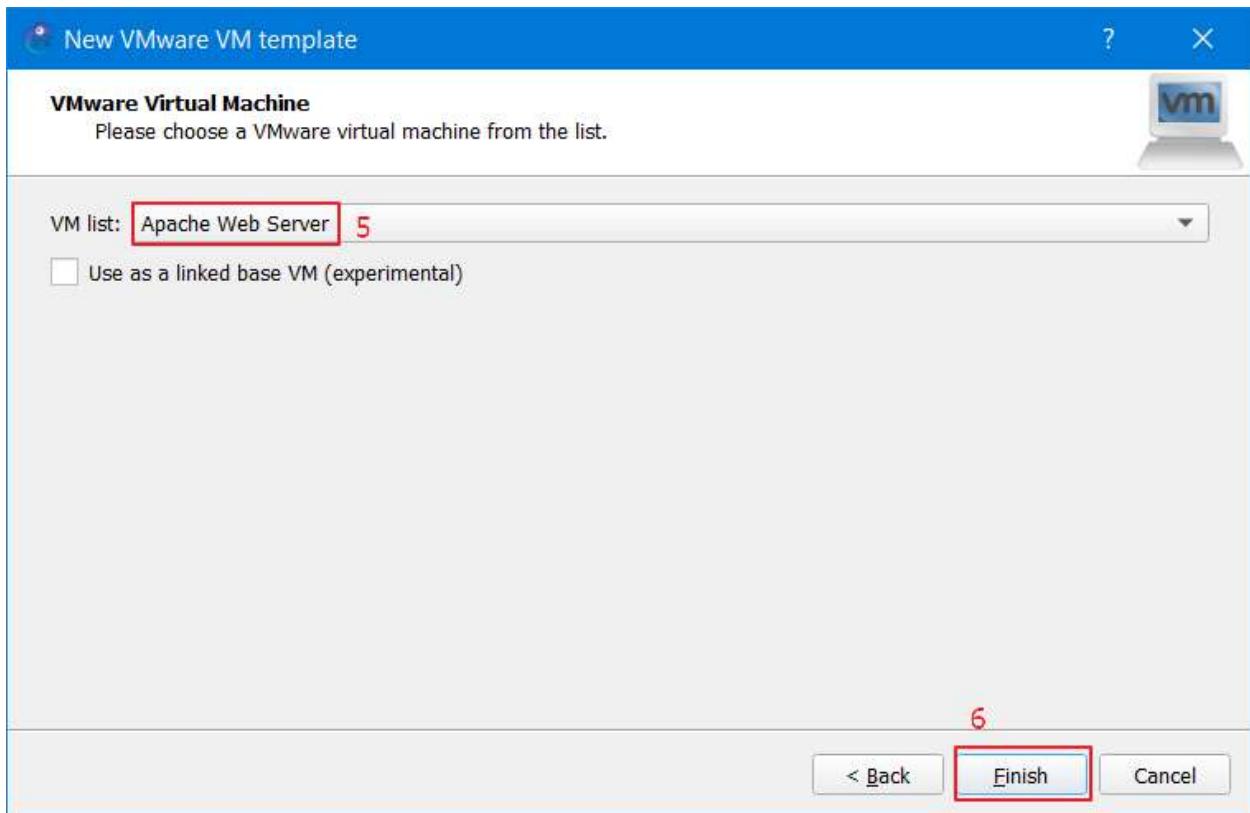


Figure 318: Import Apache server 4.

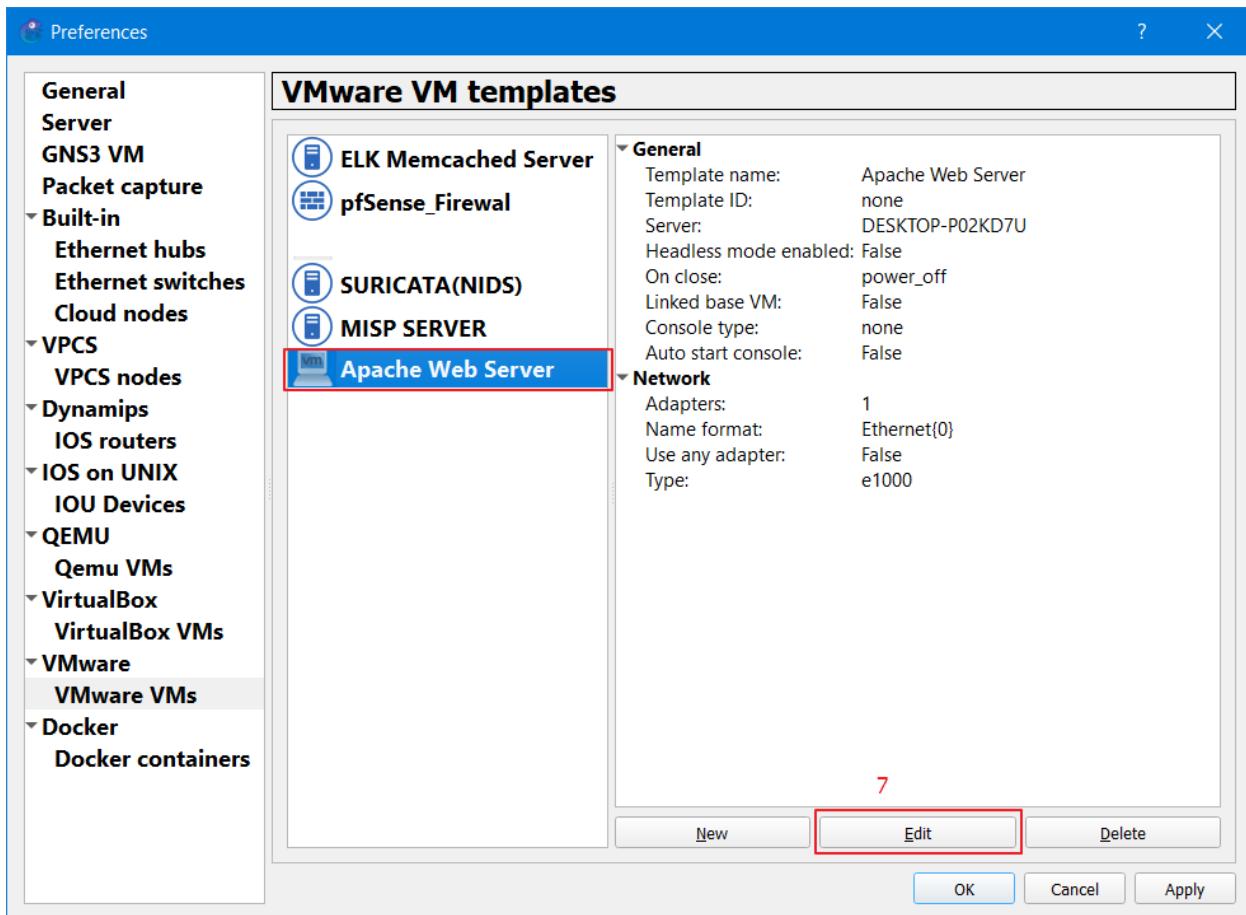


Figure 319: Import Apache server 7.

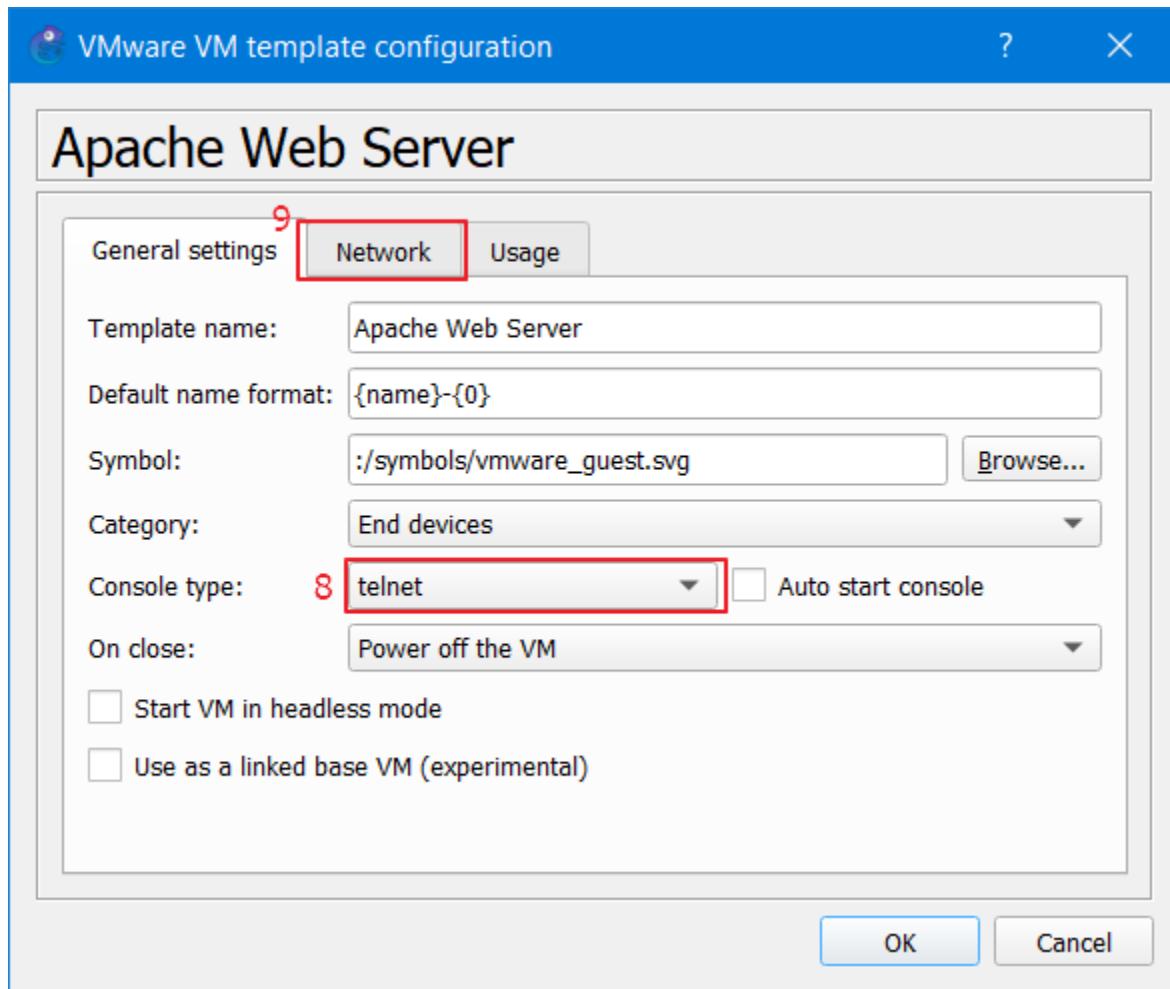


Figure 320: Import Apache server 8.

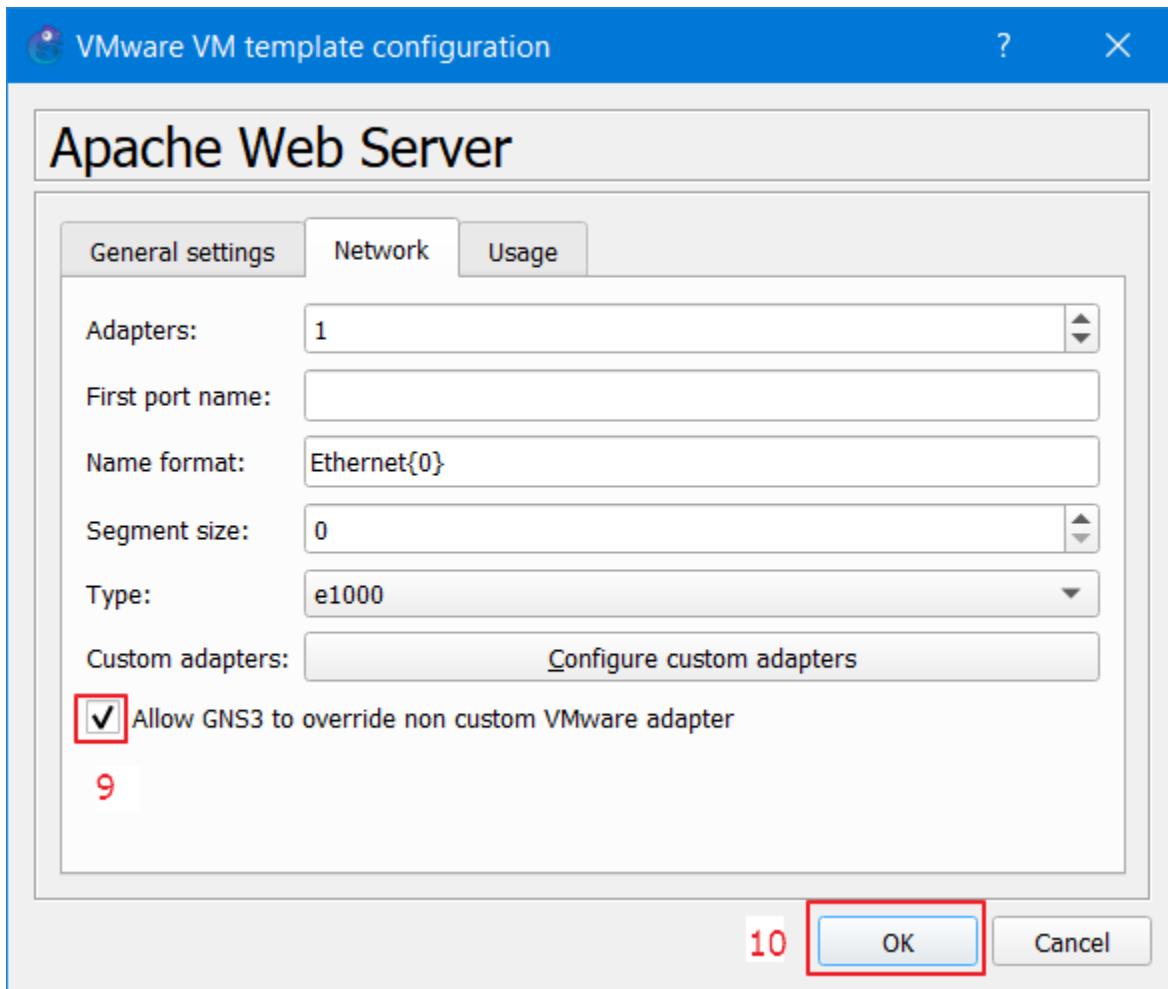


Figure 321: Import Apache server 9.

Windows 10

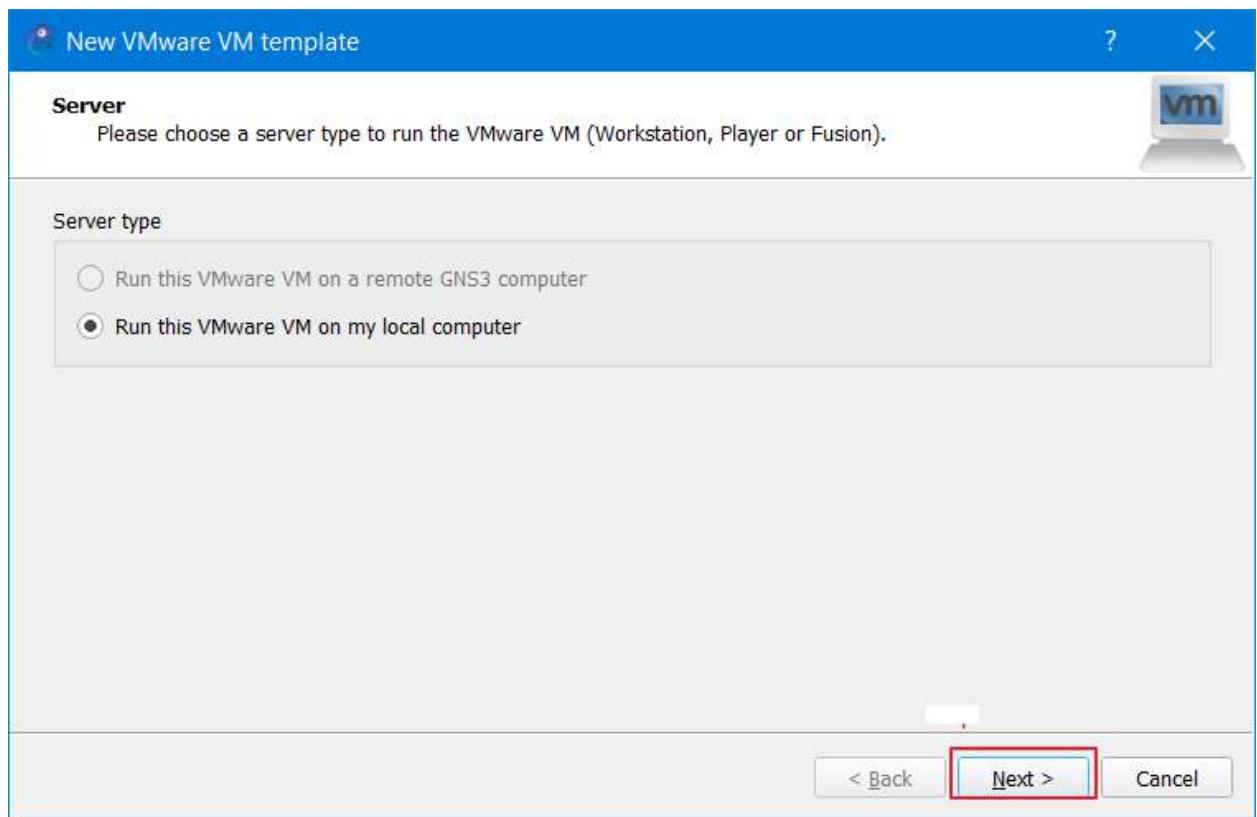


Figure 322: Importing windows 10 1.

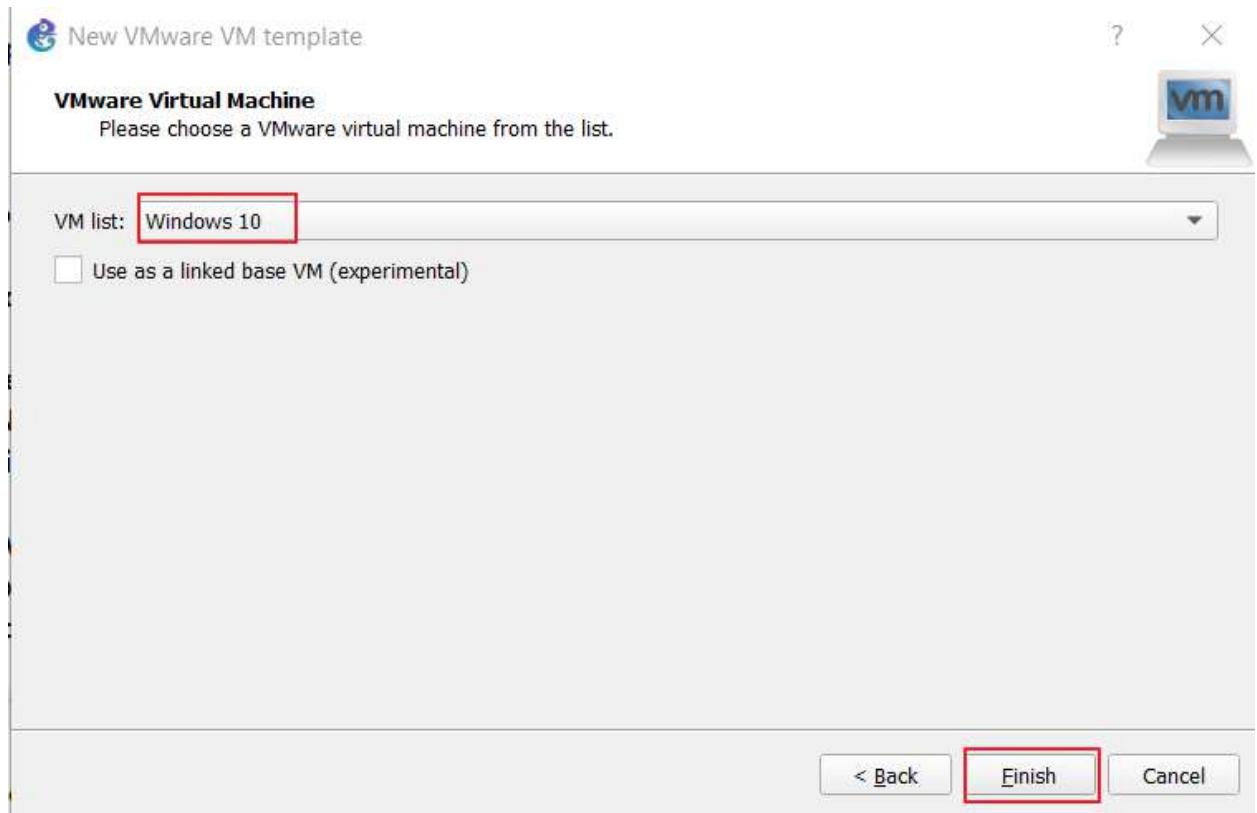


Figure 323: Importing windows 10 2.

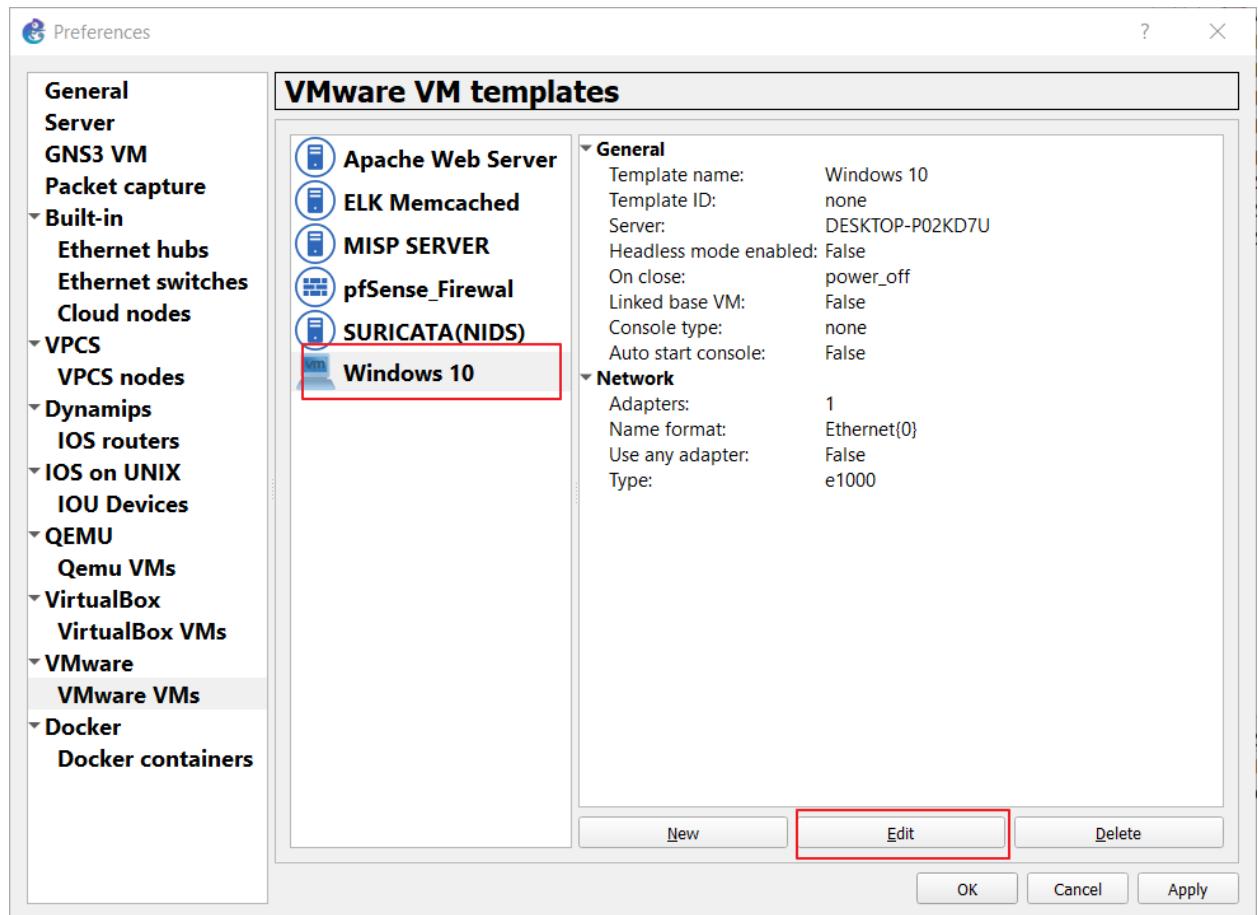


Figure 324: Importing windows 10 3.

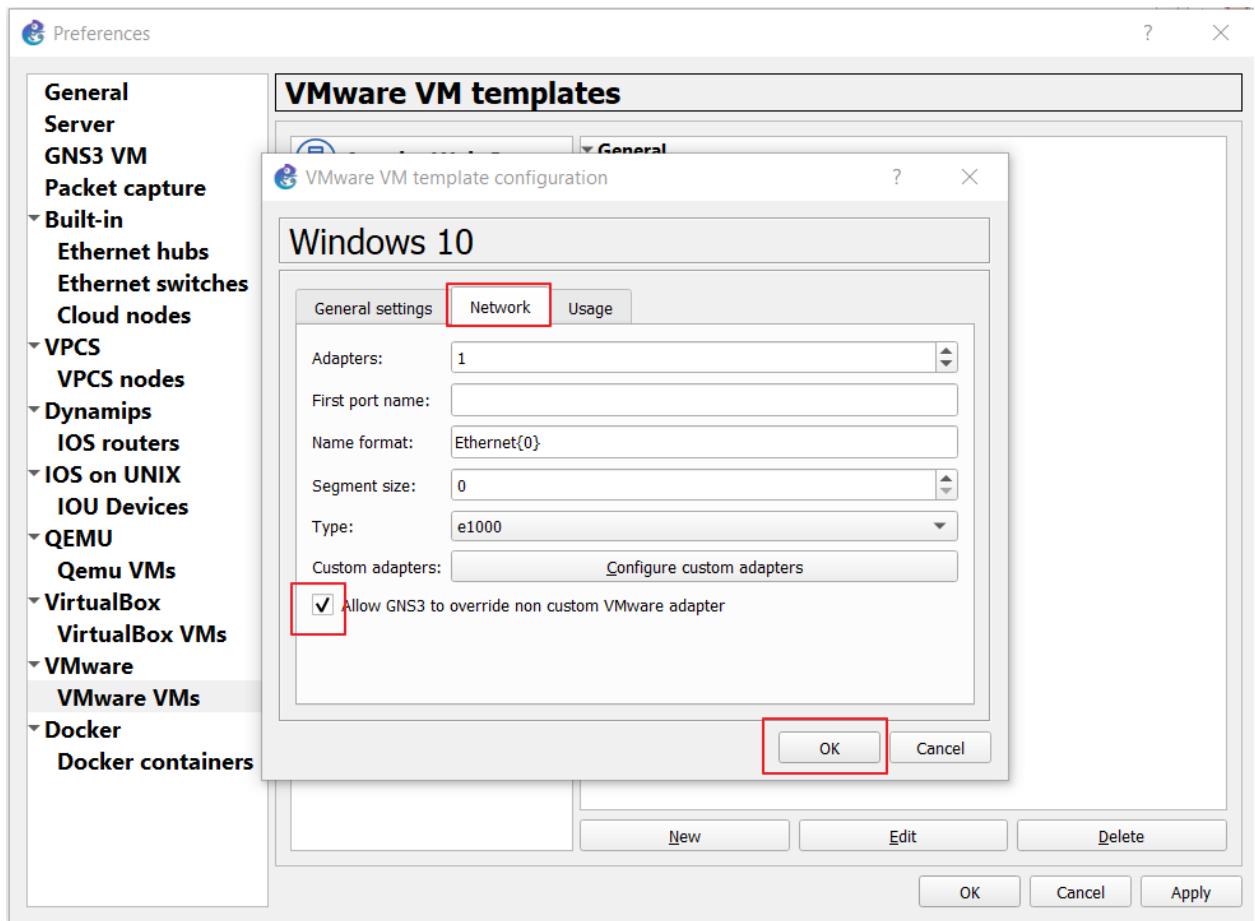
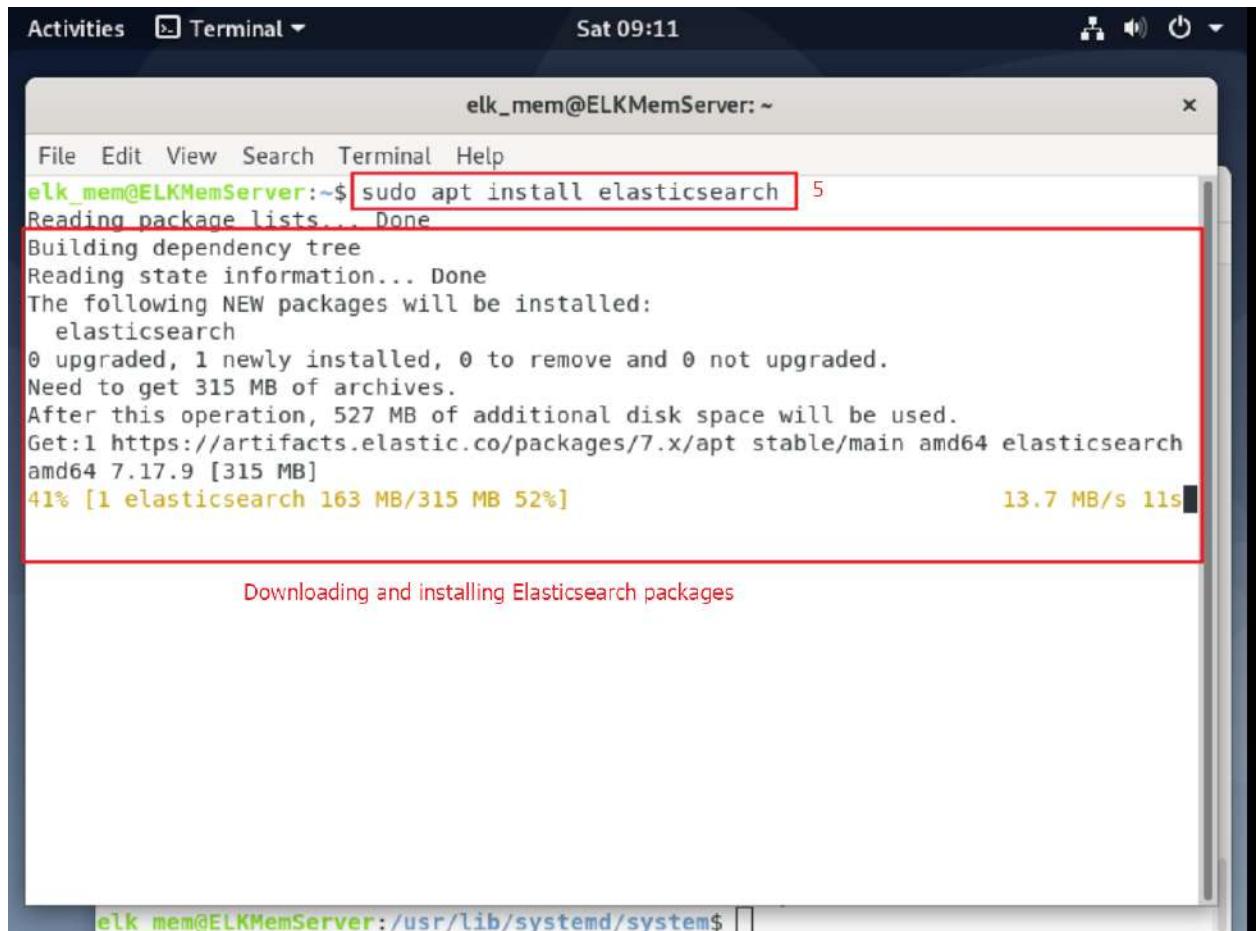


Figure 325: Importing windows 10 4.

8.6.5 ELKMemcached Server

Elasticsearch



The screenshot shows a terminal window titled "Activities Terminal" running on a Linux system. The terminal session is for user "elk_mem" on "ELKMemServer". The command entered is "sudo apt install elasticsearch". The output shows the package lists being read, dependencies being built, and the state information being updated. It indicates that one package will be installed: "elasticsearch". The operation is noted as upgrading 0 packages, installing 1 new package, removing 0 packages, and not upgrading 0 packages. A total of 315 MB of disk space is required. The package is being downloaded from "https://artifacts.elastic.co/packages/7.x/apt/stable/main" and is currently at 41% download progress, with a speed of 13.7 MB/s. The file size is 163 MB.

```
elk_mem@ELKMemServer:~$ sudo apt install elasticsearch
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 315 MB of archives.
After this operation, 527 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt/stable/main amd64 elasticsearch
  amd64 7.17.9 [315 MB]
41% [1 elasticsearch 163 MB/315 MB 52%] 13.7 MB/s 11s
```

Downloading and installing Elasticsearch packages

```
elk_mem@ELKMemServer:/usr/lib/systemd/system$
```

Figure 326: Installing Elasticsearch 1.

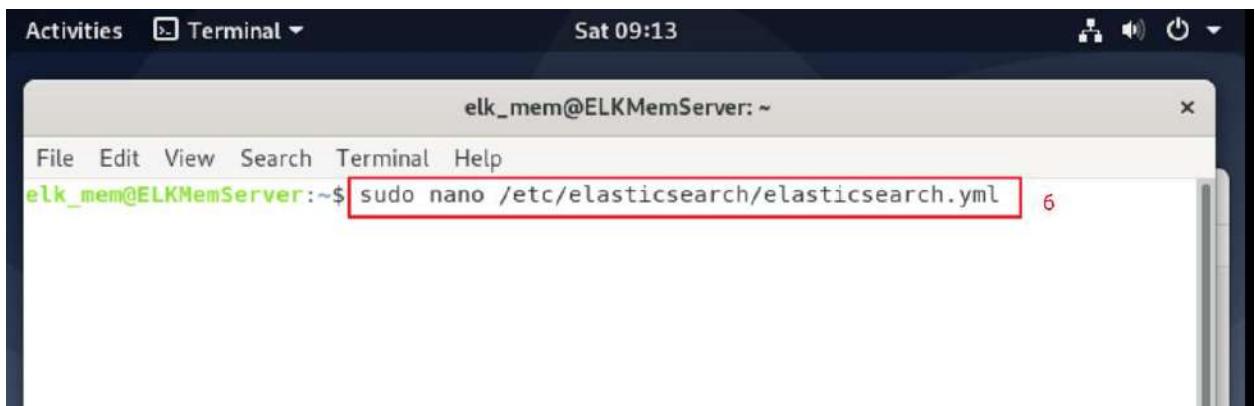
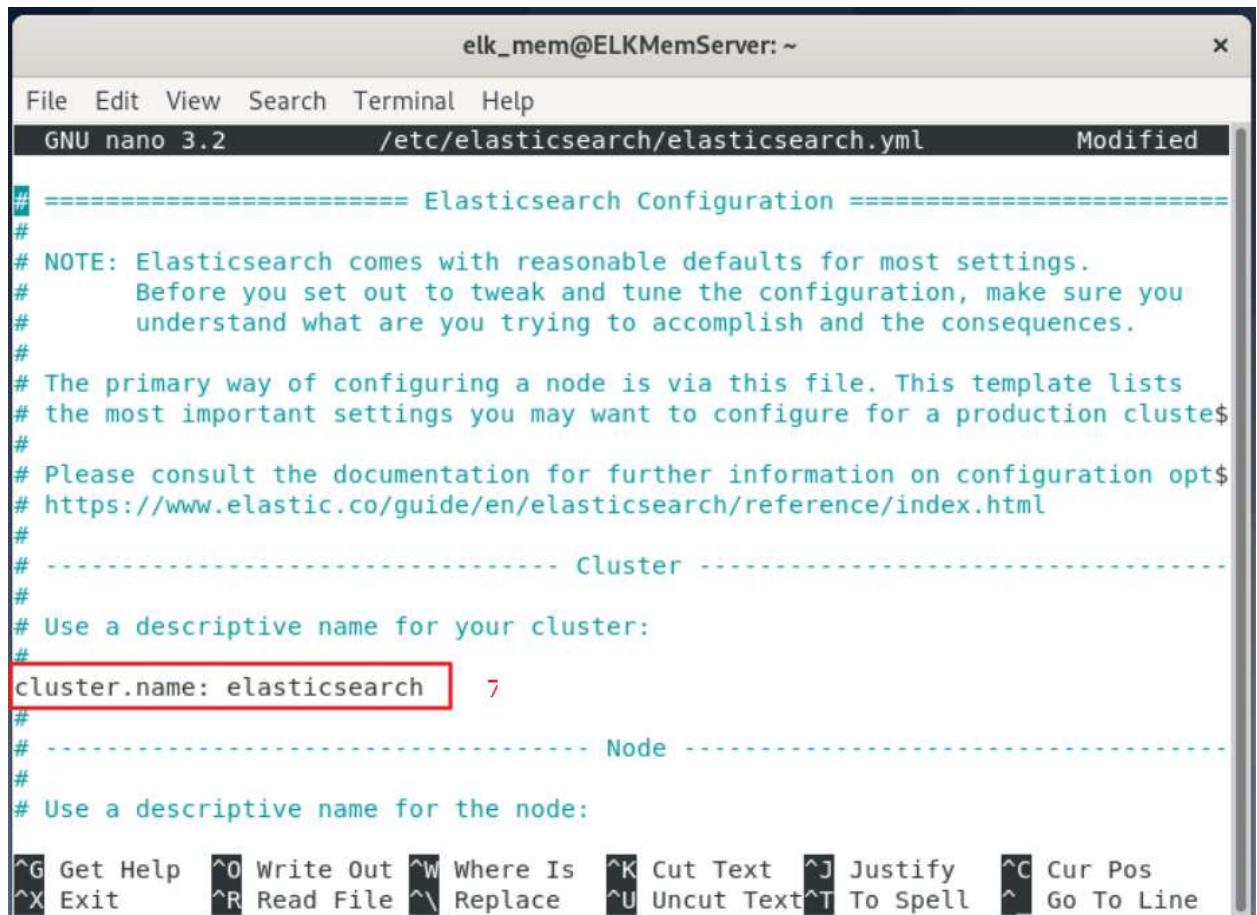


Figure 327: Configuration of Elasticsearch yml 1.



```
elk_mem@ELKMemServer: ~
File Edit View Search Terminal Help
GNU nano 3.2          /etc/elasticsearch/elasticsearch.yml      Modified

# ===== Elasticsearch Configuration =====
#
# NOTE: Elasticsearch comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluste$#
#
# Please consult the documentation for further information on configuration opt$#
# https://www.elastic.co/guide/en/elasticsearch/reference/index.html
#
# ----- Cluster -----
#
# Use a descriptive name for your cluster:
#
cluster.name: elasticsearch    7
#
# ----- Node -----
#
# Use a descriptive name for the node:

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Figure 328: Configuration of Elasticsearch yml 2.

The screenshot shows a terminal window titled "Terminal" with the command "elk_mem@ELKMemServer: ~". The file being edited is "/etc/elasticsearch/elasticsearch.yml". The text in the editor is the configuration for Elasticsearch, specifically for the network and discovery modules. Two lines of code are highlighted with red boxes: "network.host: 0.0.0.0" and "http.port: 9200". The terminal window includes a menu bar with File, Edit, View, Search, Terminal, Help, and a toolbar with various editing functions like Get Help, Write Out, Cut Text, Justify, Cur Pos, Undo, etc.

```
# ----- Network -----
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 0.0.0.0
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# ----- Discovery -----
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
```

Figure 329: Configuration of Elasticsearch yml 3.

Activities Terminal ▾ Mon 13:47

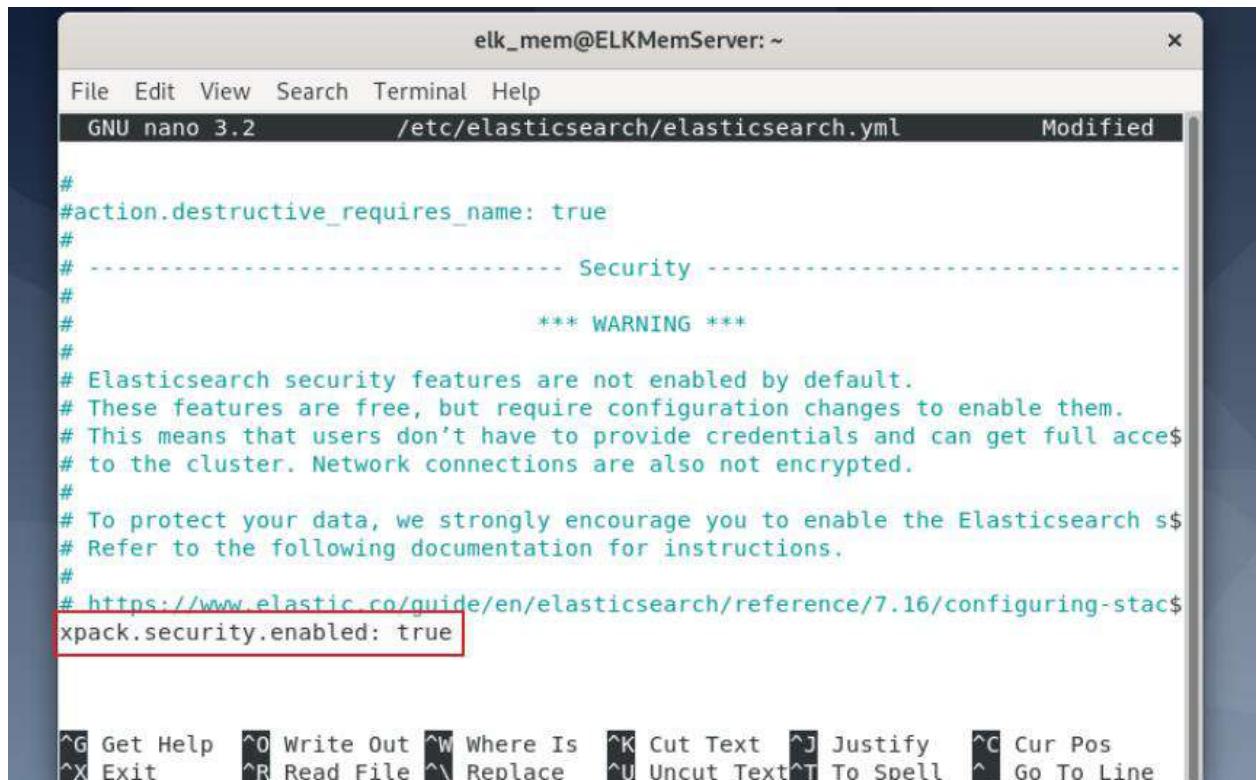
elk_mem@ELKMemServer: ~

File Edit View Search Terminal Help

GNU nano 3.2 /etc/elasticsearch/elasticsearch.yml Modified

```
# Pass an initial list of hosts to perform discovery when this node is started:  
# The default list of hosts is ["127.0.0.1", "[::1]"]  
#  
discovery.seed_hosts: []  
#  
# Bootstrap the cluster using an initial set of master-eligible nodes:  
#  
#cluster.initial_master_nodes: ["node-1", "node-2"]  
#  
# For more information, consult the discovery and cluster formation module docu$  
discovery.type: single-node  
# ----- Various -----  
#  
# Require explicit names when deleting indices:  
#  
#action.destructive_requires_name: true  
#  
# ----- Security -----  
#  
# *** WARNING ***  
#  
#  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^L Go To Line
```

Figure 330: Configuration of Elasticsearch yml 4.



```
elk_mem@ELKMemServer: ~
File Edit View Search Terminal Help
GNU nano 3.2          /etc/elasticsearch/elasticsearch.yml      Modified
#
#action.destructive_requires_name: true
#
# ----- Security -----
#
#           *** WARNING ***
#
# Elasticsearch security features are not enabled by default.
# These features are free, but require configuration changes to enable them.
# This means that users don't have to provide credentials and can get full access
# to the cluster. Network connections are also not encrypted.
#
# To protect your data, we strongly encourage you to enable the Elasticsearch security features.
# Refer to the following documentation for instructions.
#
# https://www.elastic.co/guide/en/elasticsearch/reference/7.16/configuring-security.html
xpack.security.enabled: true

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text^T To Spell  ^L Go To Line
```

Figure 331: Configuration of Elasticsearch yml 6.



```
File Edit View Search Terminal Help
elk_mem@ELKMemServer:/$ sudo nano /etc/elasticsearch/jvm.options 7
elk_mem@ELKMemServer:/$
```

Figure 332: Configuration of jvm.options.

The screenshot shows a terminal window titled "elk_mem@ELKMemServer: /etc/elasticsearch". The window displays the contents of the "jvm.options" file, which is currently being edited with the nano text editor. The file contains configuration options for the JVM heap size. A red box highlights the line "-Xms512m". The terminal interface includes a menu bar with File, Edit, View, Search, Terminal, and Help, and a status bar at the bottom showing keyboard shortcuts for various functions like Get Help (^G), Write Out (^O), Cut Text (^K), Justify (^J), Exit (^X), Read File (^R), Replace (^V), Uncut Tex (^U), and To Spell (^T).

```
elk_mem@ELKMemServer: /etc/elasticsearch
File Edit View Search Terminal Help
GNU nano 3.2          jvm.options          Modified

#####
## IMPORTANT: JVM heap size
#####
## The heap size is automatically configured by Elasticsearch
## based on the available memory in your system and the roles
## each node is configured to fulfill. If specifying heap is
## required, it should be done through a file in jvm.options.d,
## and the min and max should be set to the same value. For
## example, to set the heap to 4 GB, create a new file in the
## jvm.options.d directory containing these lines:
##
-Xms512m
-Xmx512m

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit      ^R Read File ^V Replace ^U Uncut Tex ^T To Spell
```

Figure 333: Configuration of jvm.options 1.

Figure 334: Installation complete.

The terminal window title is "elk_mem@ELKMemServer: ~". The command run is "/usr/share/elasticsearch/bin/elasticsearch-setup-passwords interactive". The output shows the setup of passwords for reserved users: elastic, apm_system, Kibana, Kibana_system, logstash_system, beats_system, remote_monitoring_user. It prompts for confirmation with "Please confirm that you would like to continue [y/N]Y". The password entry section is highlighted with a red box and labeled 14. The password "root123" is noted as being used for all users. The terminal window has a red border.

```
File Edit View Search Terminal Help
root@ELKMemServer:~# /usr/share/elasticsearch/bin/elasticsearch-setup-passwords
interactive
initiating the setup of passwords for reserved users elastic,apm_system,Kibana,K
ibana_system,logstash_system,beats_system,remote_monitoring_user.
You will be prompted to enter passwords as the process progresses.
Please confirm that you would like to continue [y/N]Y
Enter password for [elastic]:
Reenter password for [elastic]:
Enter password for [apm_system]:
Reenter password for [apm_system]:
Enter password for [kibana_system]:
Reenter password for [kibana_system]:
Passwords do not match.
Try again.
Enter password for [kibana_system]:
Reenter password for [kibana_system]:
Enter password for [logstash_system]:
Reenter password for [logstash_system]:
Enter password for [beats_system]:
Reenter password for [beats_system]:
Enter password for [remote_monitoring_user]:
Reenter password for [remote monitoring user]:
```

12

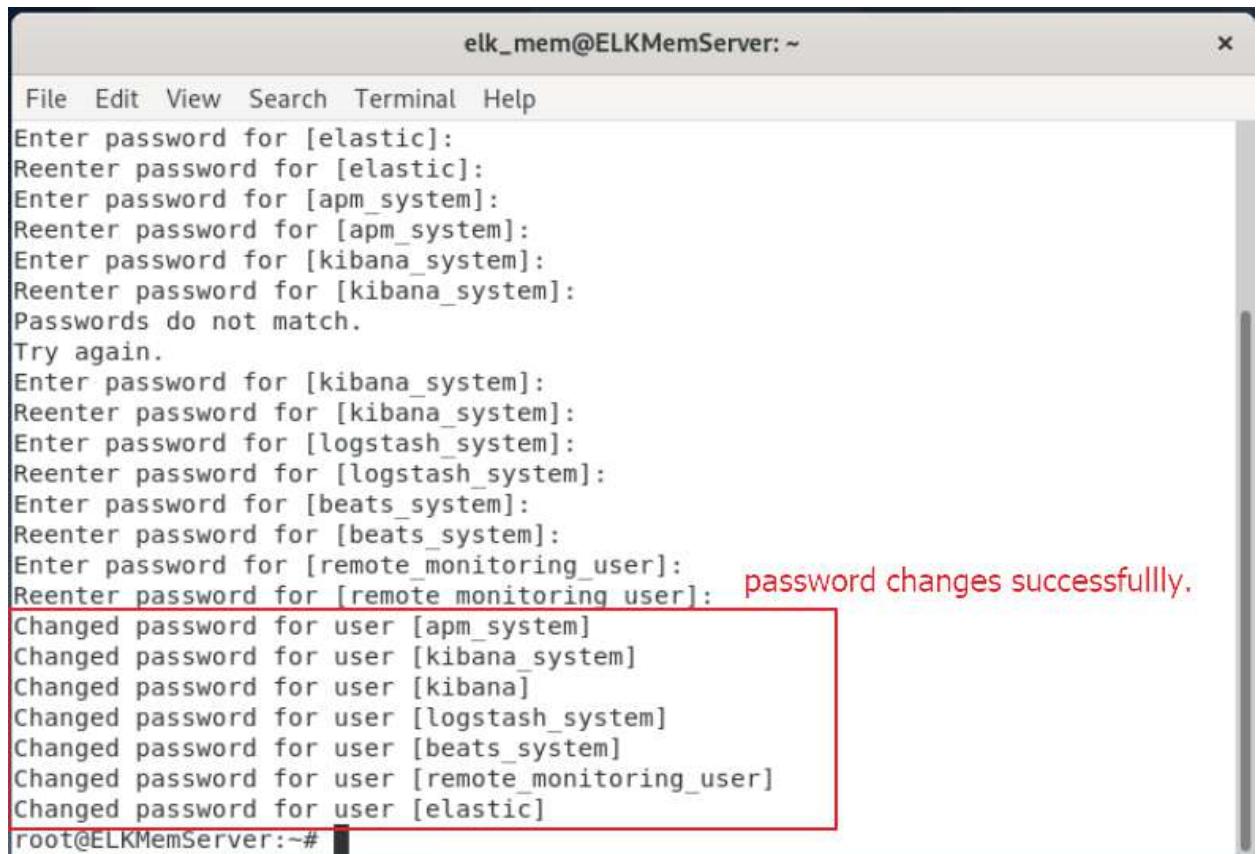
13

14

password = root123

FOR ALL

Figure 335: setting password 1.



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". It displays a process of setting passwords for multiple system users. The user is prompted to enter and re-enter passwords for elastic, apm_system, kibana_system, logstash_system, beats_system, and remote_monitoring_user. There is a noticeable error where the password for remote_monitoring_user is entered twice, resulting in a red message: "Reenter password for [remote monitoring user]: password changes successfully." A red rectangular box highlights the successful password change message and the subsequent log entries for all users: apm_system, kibana_system, kibana, logstash_system, beats_system, remote_monitoring_user, and elastic.

```
File Edit View Search Terminal Help
Enter password for [elastic]:
Reenter password for [elastic]:
Enter password for [apm_system]:
Reenter password for [apm_system]:
Enter password for [kibana_system]:
Reenter password for [kibana_system]:
Passwords do not match.
Try again.
Enter password for [kibana_system]:
Reenter password for [kibana_system]:
Enter password for [logstash_system]:
Reenter password for [logstash_system]:
Enter password for [beats_system]:
Reenter password for [beats_system]:
Enter password for [remote_monitoring_user]:
Reenter password for [remote monitoring user]: password changes successfully.
Changed password for user [apm_system]
Changed password for user [kibana_system]
Changed password for user [kibana]
Changed password for user [logstash_system]
Changed password for user [beats_system]
Changed password for user [remote_monitoring_user]
Changed password for user [elastic]
root@ELKMemServer:~#
```

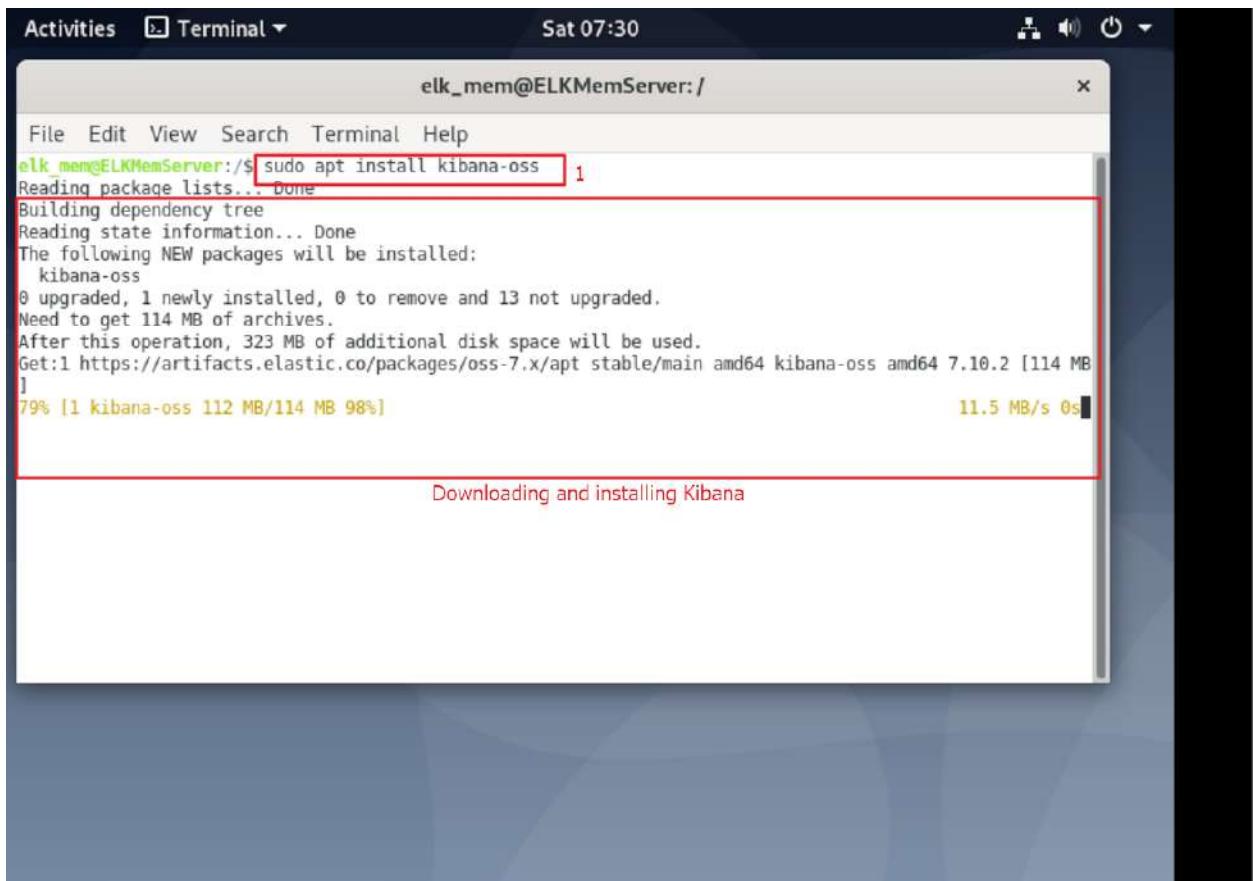
Figure 336: setting password 2.

The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window contains several lines of text output from a curl command. A red box highlights the central portion of the output, which displays the Elasticsearch configuration object. The word "OUTPUT" is written in red capital letters to the right of this highlighted area. The terminal prompt "root@ELKMemServer:~# " appears at the bottom twice.

```
File Edit View Search Terminal Help
Changed password for user [logstash_system]
Changed password for user [beats_system]
Changed password for user [remote_monitoring_user]
Changed password for user [elastic]
root@ELKMemServer:~# curl -X GET -u elastic:root123 "localhost:9200" 15
{
  "name" : "ELKMemServer",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "z17KbuWBT-mhpfe16x_4QQ",
  "version" : {
    "number" : "7.17.9",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "ef4822227ee6b9e70e502f0f0daa52435ee634d",
    "build_date" : "2023-01-31T05:34:43.305517834Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
root@ELKMemServer:~#
root@ELKMemServer:~#
```

Figure 337: Accessing Elasticsearch.

Kibana



A screenshot of a Linux desktop environment showing a terminal window titled "Terminal". The window title bar also shows "Activities" and the date "Sat 07:30". The terminal window contains the following text:

```
elk_mem@ELKMemServer:/$ sudo apt install kibana-oss
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  kibana-oss
0 upgraded, 1 newly installed, 0 to remove and 13 not upgraded.
Need to get 114 MB of archives.
After this operation, 323 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/oss-7.x/apt/stable/main amd64 kibana-oss amd64 7.10.2 [114 MB]
100% [1 kibana-oss 112 MB/114 MB 98%] 11.5 MB/s 0s
```

The terminal output is highlighted with a red rectangle, and the status bar at the bottom of the terminal window also has a red rectangle around it. The status bar text reads "Downloading and installing Kibana".

Figure 338: Installing Kibana.



Figure 339: configuring yml files for kibana 1.

Kibana is served by a back end server. This setting specifies the port to use.
#server.port: 5601
server.port: 5601
Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
The default is 'localhost', which usually means remote machines will not be able to connect.
To allow connections from remote users, set this parameter to a non-loopback address.
#server.host: "localhost"
server.host: "0.0.0.0"
Enables you to specify a path to mount Kibana at if you are running behind a proxy.
Use the 'server.rewriteBasePath' setting to tell Kibana if it should remove the basePath
from requests it receives, and to prevent a deprecation warning at startup.
This setting cannot end in a slash.
#serverbasePath: ""

Specifies whether Kibana should rewrite requests that are prefixed with
`server.basePath` or require that they are rewritten by your reverse proxy.
This setting was effectively always 'false' before Kibana 6.3 and will
default to 'true' starting in Kibana 7.0.
#server.rewriteBasePath: false

Specifies the public URL at which Kibana is available for end users. If
`server.basePath` is configured this URL should end with the same basePath.
#server.publicBaseUrl: ""

The maximum payload size in bytes for incoming server requests.
#server.maxPayload: 1048576

The Kibana server's name. This is used for display purposes.
#server.name: "your-hostname"

The URLs of the Elasticsearch instances to use for all your queries.
#elasticsearch.hosts: ["http://localhost:9200"]
elasticsearch.hosts: ["http://10.10.30.3:9200"]
Kibana uses an index in Elasticsearch to store saved searches, visualizations and

Figure 340: configuring yml files for kibana 2.

The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window title bar includes "File Edit View Search Terminal Help" and a status bar showing "GNU nano 3.2 /etc/kibana/kibana.yml Modified". The main area of the terminal displays the contents of the /etc/kibana/kibana.yml file. A red box highlights the following configuration block:

```
# If your Elasticsearch is protected with basic authentication, these settings $  
# the username and password that the Kibana server uses to perform maintenance $  
# index at startup. Your Kibana users still need to authenticate with Elasticsearch$  
# is proxied through the Kibana server.  
xpack.security.enabled: true  
elasticsearch.username: "elastic"  
elasticsearch.password: "root123"
```

Below this, the file continues with comments about service account tokens and SSL settings. At the bottom of the terminal window, there is a standard nano editor key mapping legend.

Figure 341: configuring yml files for kibana 3.

The screenshot shows a terminal window titled "Terminal" with the command-line interface "elk_mem@ELKMemServer:/". The terminal displays the following sequence of commands and their outputs:

```
File Edit View Search Terminal Help
elk_mem@ELKMemServer:/$ sudo systemctl start kibana      3
elk_mem@ELKMemServer:/$ sudo systemctl enable kibana      4
Synchronizing state of kibana.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable kibana
elk_mem@ELKMemServer:/$ sudo systemctl status kibana      5
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-02-11 09:45:18 GMT; 19s ago
     Docs: https://www.elastic.co
 Main PID: 6193 (node)
    Tasks: 11 (limit: 7024)
   Memory: 635.4M
      CGroup: /system.slice/kibana.service
              └─6193 /usr/share/kibana/bin/.../node/bin/node /usr/share/kibana/bin/.../src/c

Feb 11 09:45:18 ELKMemServer systemd[1]: Started Kibana.
(lines 1-11/11 (END))
```

At the bottom right of the terminal window, the message "Kibana Successfully Running." is displayed.

Figure 342: installation complete.

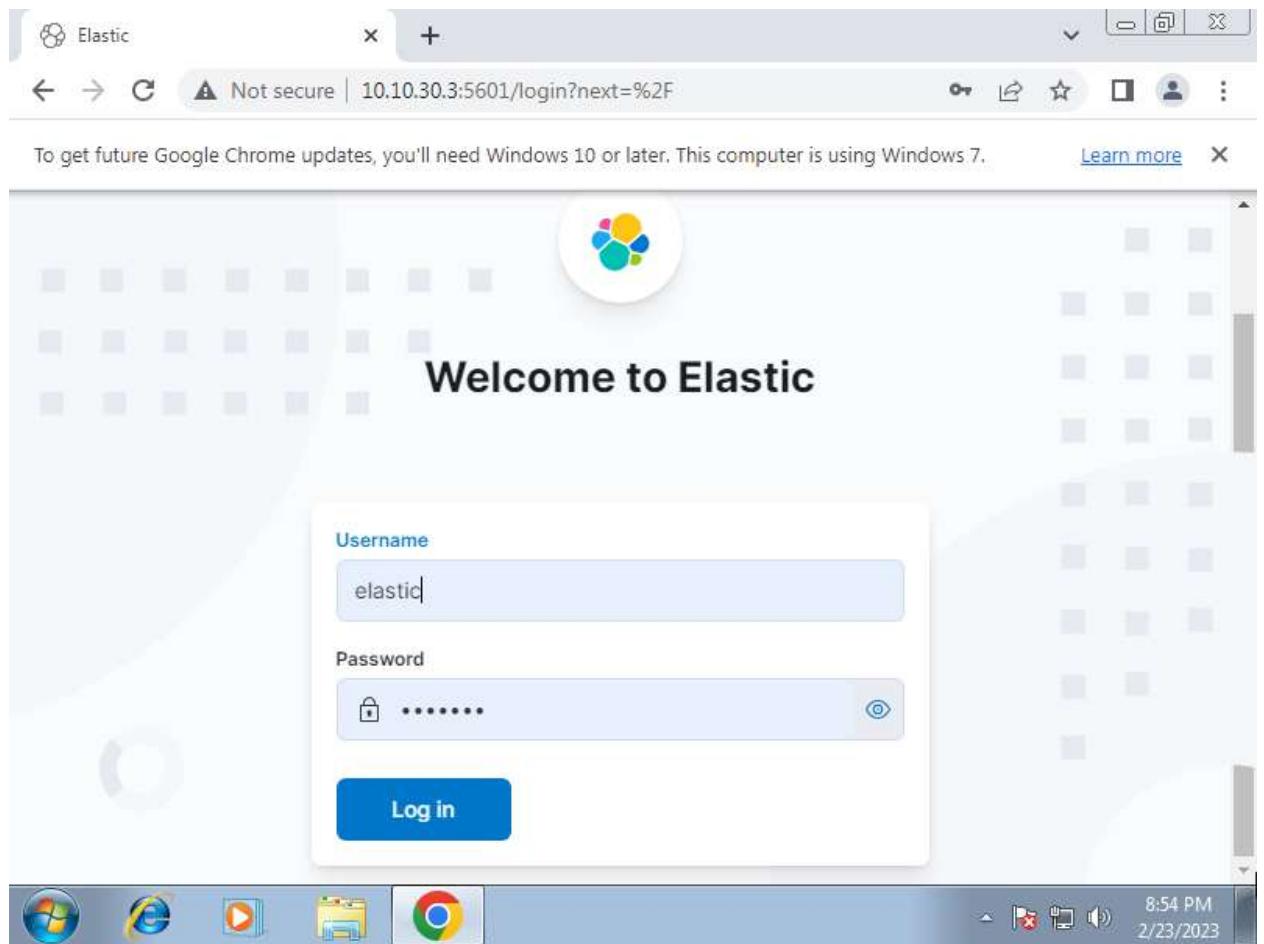
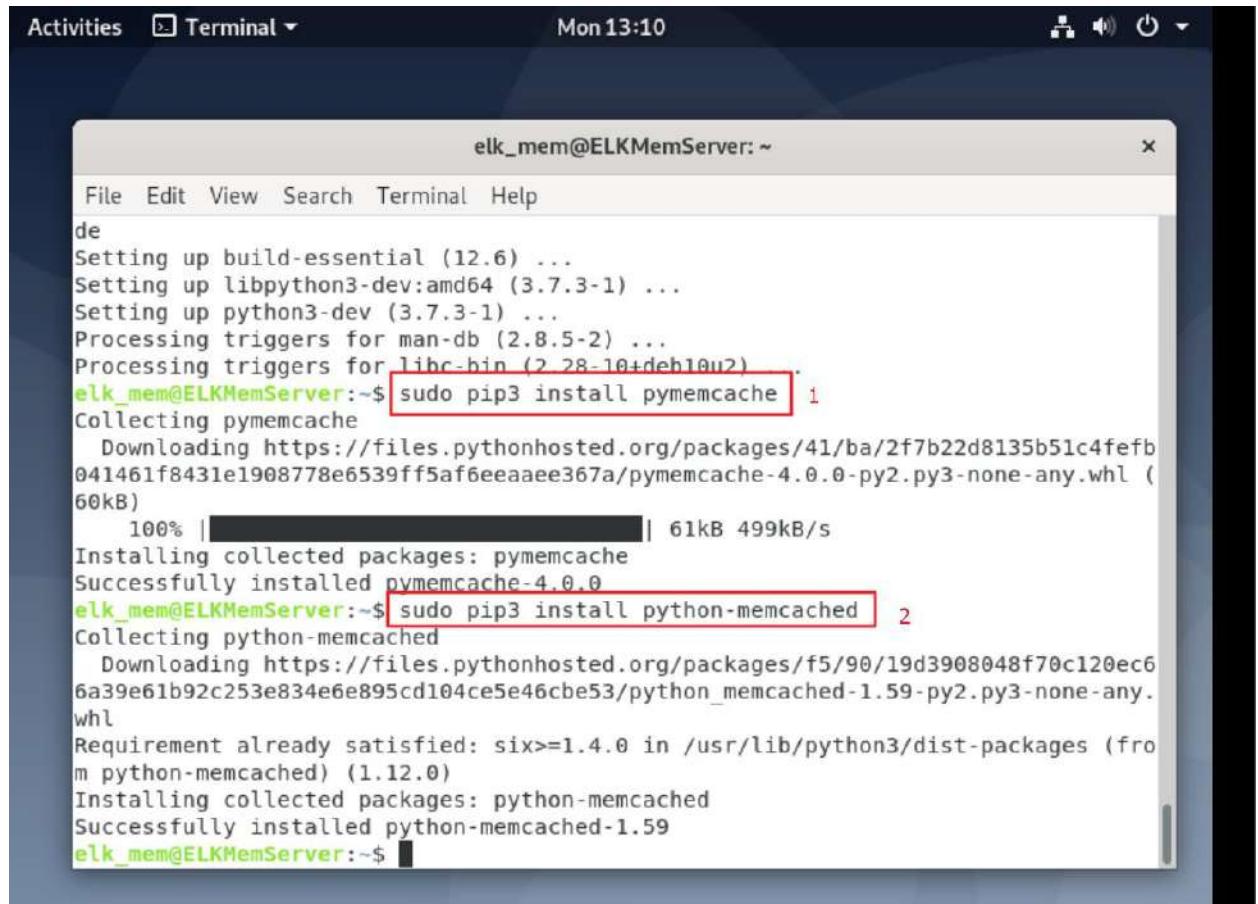


Figure 343: Dashboard of Kibana.

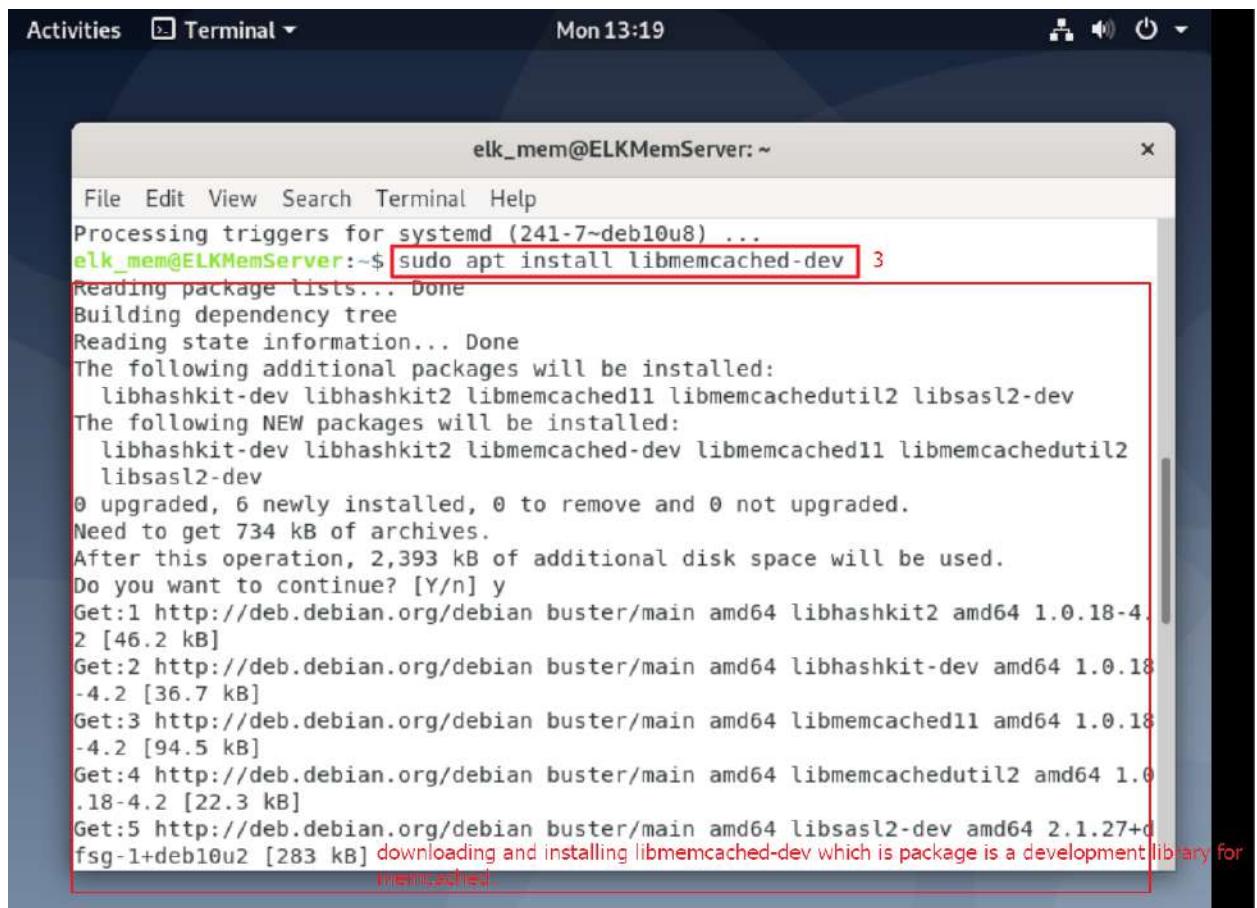
Memcached



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The terminal is displaying the output of a pip3 install command. The command is broken into two parts: "sudo pip3 install pymemcache" (marked with a red box and number 1) and "sudo pip3 install python-memcached" (marked with a red box and number 2). The terminal shows the progress of the download for pymemcache, which is 100% complete at 61kB/s. Both packages are successfully installed.

```
File Edit View Search Terminal Help
de
Setting up build-essential (12.6) ...
Setting up libpython3-dev:amd64 (3.7.3-1) ...
Setting up python3-dev (3.7.3-1) ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for libc-bin (2.28-10+deb10u2).
elk_mem@ELKMemServer:~$ sudo pip3 install pymemcache 1
Collecting pymemcache
  Downloading https://files.pythonhosted.org/packages/41/ba/2f7b22d8135b51c4fefb041461f8431e1908778e6539ff5af6eeaaee367a/pymemcache-4.0.0-py2.py3-none-any.whl (60kB)
    100% |██████████| 61kB 499kB/s
Installing collected packages: pymemcache
Successfully installed pymemcache-4.0.0
elk_mem@ELKMemServer:~$ sudo pip3 install python-memcached 2
Collecting python-memcached
  Downloading https://files.pythonhosted.org/packages/f5/90/19d3908048f70c120ec66a39e61b92c253e834e6e895cd104ce5e46cbe53/python_memcached-1.59-py2.py3-none-any.whl
Requirement already satisfied: six>=1.4.0 in /usr/lib/python3/dist-packages (from python-memcached) (1.12.0)
Installing collected packages: python-memcached
Successfully installed python-memcached-1.59
elk_mem@ELKMemServer:~$
```

Figure 344: Installing dependencies 1.

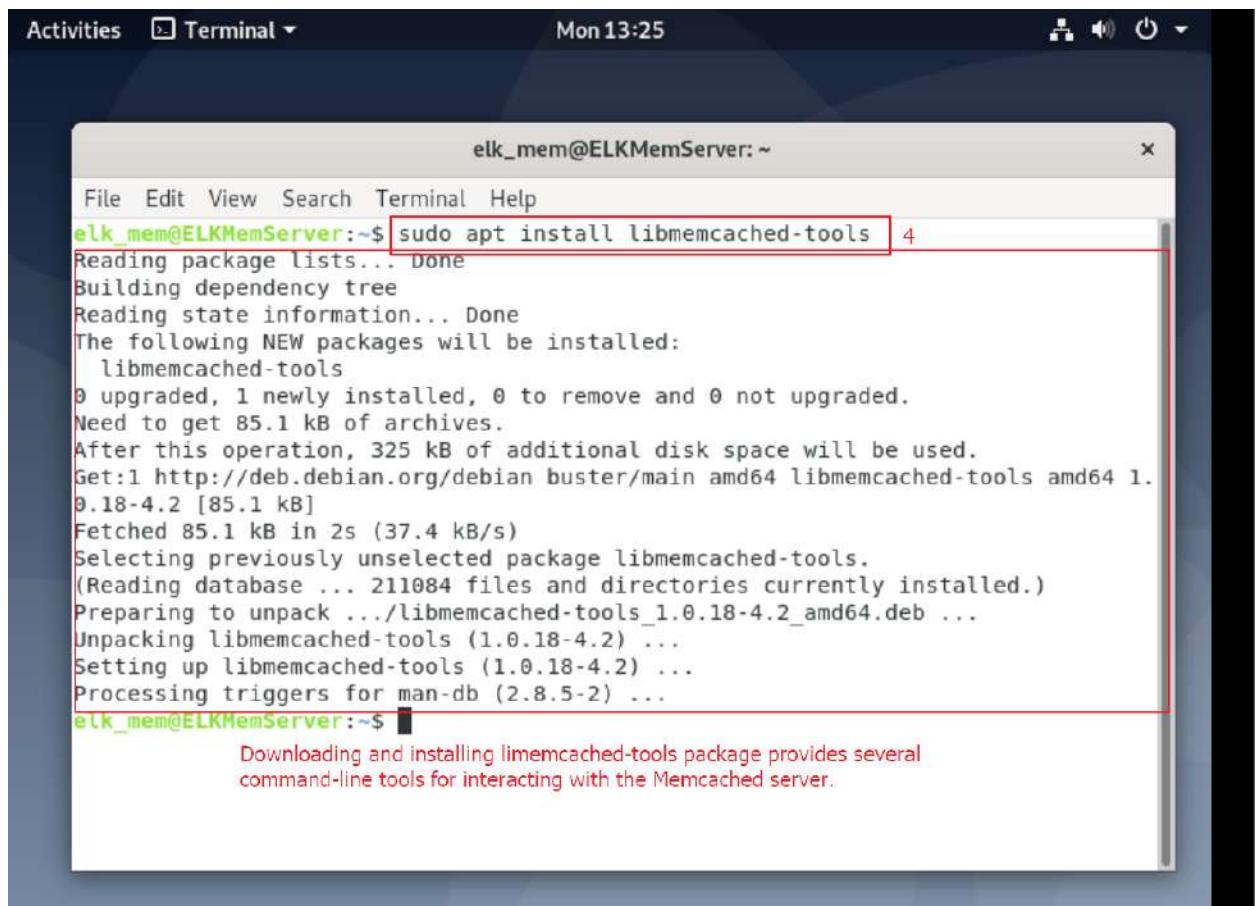


The screenshot shows a terminal window titled "Activities Terminal" with the status bar indicating "Mon 13:19". The terminal window title is "elk_mem@ELKMemServer: ~". The terminal content shows the following command being run:

```
elk_mem@ELKMemServer:~$ sudo apt install libmemcached-dev
```

The output of the command is displayed in the terminal window, showing the package manager processing triggers for systemd, reading package lists, building dependency trees, and determining packages to install. It lists several packages to be installed, including libhashkit-dev, libhashkit2, libmemcached11, libmemcachedutil2, libsasl2-dev, libmemcached-dev, libmemcached11, libmemcachedutil2, and libsasl2-dev. It shows 0 upgraded, 6 newly installed, 0 to remove, and 0 not upgraded, requiring 734 kB of additional disk space. The user is prompted with "Do you want to continue? [Y/n] y" and the download and installation process begins, listing five GET requests for the packages from deb.debian.org.

Figure 345: Installing dependencies 2.

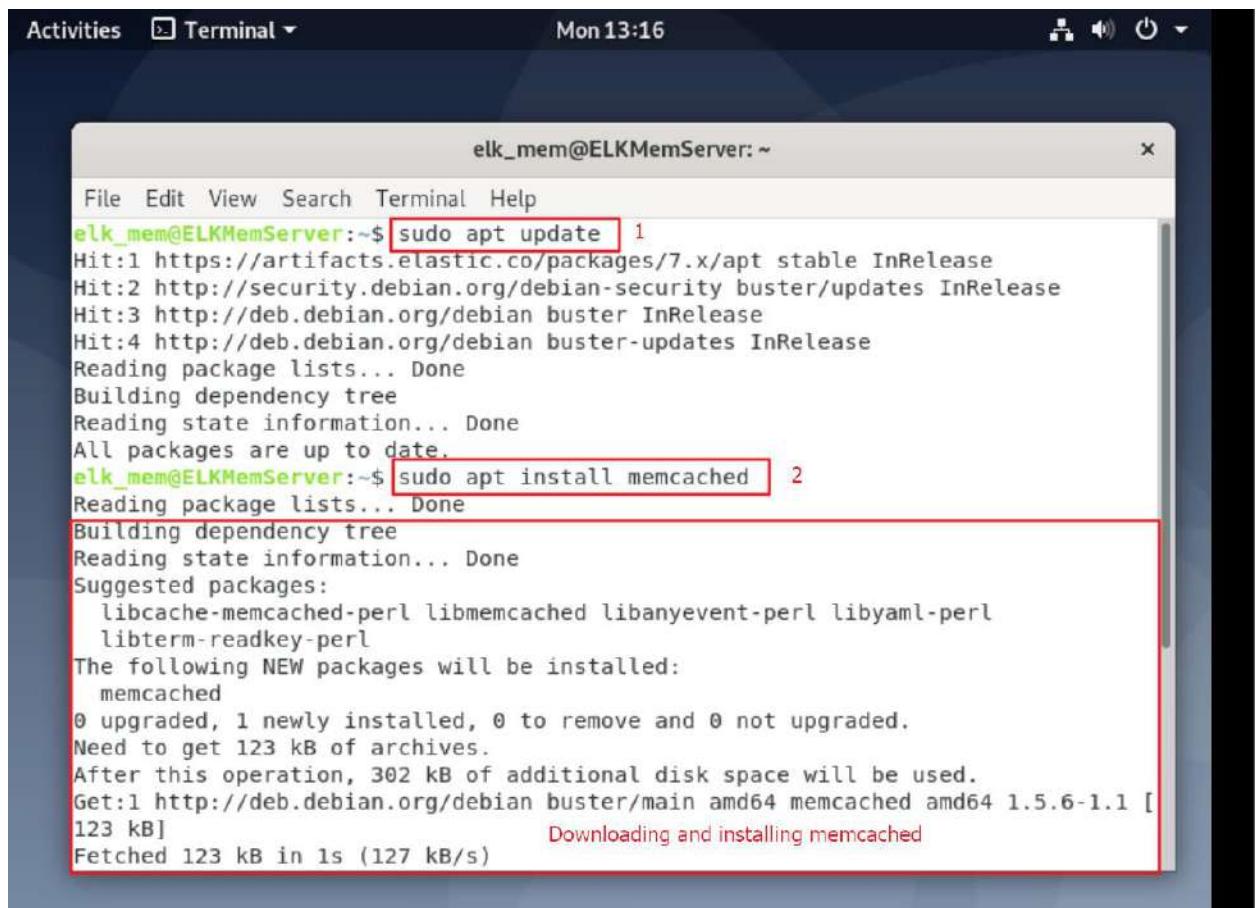


A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "elk_mem@ELKMemServer: ~". The window contains a command-line session where the user runs "sudo apt install libmemcached-tools". The output shows the package being downloaded from "http://deb.debian.org/debian" and installed. A red box highlights the command "sudo apt install libmemcached-tools". Below the terminal window, a tooltip provides a brief description of the package: "Downloading and installing libmemcached-tools package provides several command-line tools for interacting with the Memcached server."

```
File Edit View Search Terminal Help
elk_mem@ELKMemServer:~$ sudo apt install libmemcached-tools 4
Reading package lists... done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  libmemcached-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 85.1 kB of archives.
After this operation, 325 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian buster/main amd64 libmemcached-tools amd64 1.0.18-4.2 [85.1 kB]
Fetched 85.1 kB in 2s (37.4 kB/s)
Selecting previously unselected package libmemcached-tools.
(Reading database ... 211084 files and directories currently installed.)
Preparing to unpack .../libmemcached-tools_1.0.18-4.2_amd64.deb ...
Unpacking libmemcached-tools (1.0.18-4.2) ...
Setting up libmemcached-tools (1.0.18-4.2) ...
Processing triggers for man-db (2.8.5-2) ...
elk_mem@ELKMemServer:~$ 
```

Download and installing libmemcached-tools package provides several command-line tools for interacting with the Memcached server.

Figure 346: Installing dependencies 3.



A screenshot of a Linux desktop environment showing a terminal window titled "elk_mem@ELKMemServer: ~". The terminal is displaying the output of a command-line session. The user has run "sudo apt update" (line 1) and "sudo apt install memcached" (line 2). The terminal shows standard package manager output, including dependency trees, state information, and download/installation details for the "memcached" package.

```
File Edit View Search Terminal Help
elk_mem@ELKMemServer:~$ sudo apt update 1
Hit:1 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Hit:2 http://security.debian.org/debian-security buster/updates InRelease
Hit:3 http://deb.debian.org/debian buster InRelease
Hit:4 http://deb.debian.org/debian buster-updates InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
elk_mem@ELKMemServer:~$ sudo apt install memcached 2
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  libcache-memcached-perl libmemcached libanyevent-perl libyaml-perl
  libterm-readkey-perl
The following NEW packages will be installed:
  memcached
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 123 kB of archives.
After this operation, 302 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian buster/main amd64 memcached amd64 1.5.6-1.1 [123 kB]                                     Downloading and installing memcached
Fetched 123 kB in 1s (127 kB/s)
```

Figure 347: Installing Memcached.

The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The terminal displays the following command sequence:

```
elk_mem@ELKMemServer:~$ sudo systemctl start memcached 5
elk_mem@ELKMemServer:~$ sudo systemctl enable memcached 6
Synchronizing state of memcached.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable memcached
elk_mem@ELKMemServer:~$ sudo systemctl status memcached 7
```

Output of the status command:

```
* memcached.service - memcached daemon
   Loaded: loaded (/lib/systemd/system/memcached.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-02-13 13:15:54 GMT; 12min ago
     Docs: man:memcached(1)
     Main PID: 9833 (memcached)
        Tasks: 10 (limit: 7024)
       Memory: 3.9M
          CGroup: /system.slice/memcached.service
                  └─9833 /usr/bin/memcached -m 64 -p 11211 -u memcache -l 127.0.0.1 -P
```

A red box highlights the output of the status command, with the text "Successfully Running Memcached." appearing in red at the end of the list of metrics.

Log entries from the terminal window:

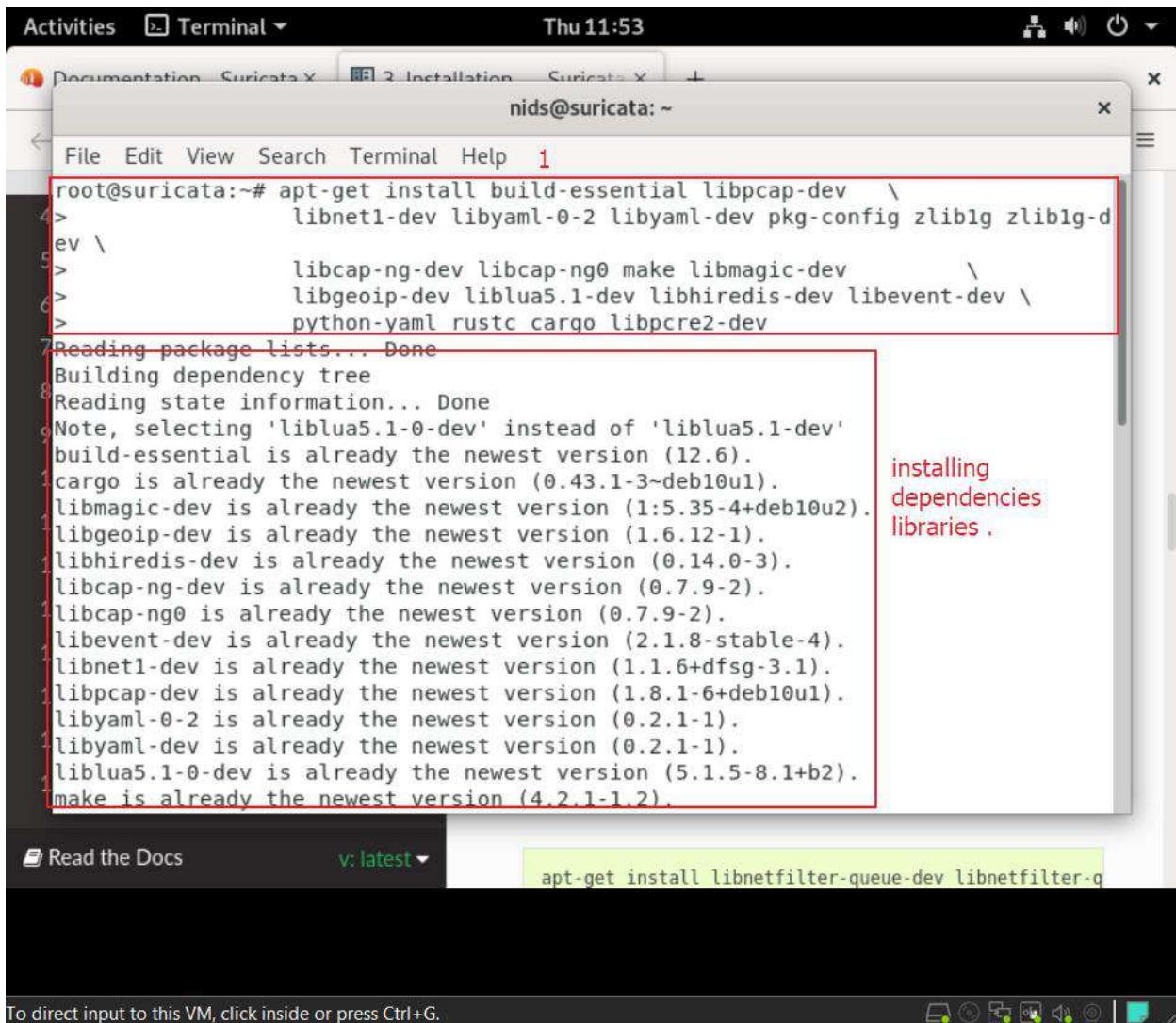
```
Feb 13 13:15:54 ELKMemServer systemd[1]: Started memcached daemon.
Feb 13 13:15:54 ELKMemServer systemd[1]: /lib/systemd/system/memcached.service:1
Feb 13 13:28:36 ELKMemServer systemd[1]: /lib/systemd/system/memcached.service:1
Feb 13 13:28:36 ELKMemServer systemd[1]: /lib/systemd/system/memcached.service:1
Feb 13 13:28:37 ELKMemServer systemd[1]: /lib/systemd/system/memcached.service:1
```

Bottom of the terminal window:

```
lines 1-15/15 (END)
```

Figure 348: Started Memcached.

8.4.6 Suricata



```
root@suricata:~# apt-get install build-essential libpcap-dev \
4>           libnet1-dev libyaml-0-2 libyaml-dev pkg-config zlib1g zlib1g-d
ev \
5>           libcap-ng-dev libcap-ng0 make libmagic-dev \
6>           libgeoip-dev liblua5.1-dev libhiredis-dev libevent-dev \
7>           python-yaml rustc cargo libpcre2-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'liblua5.1-0-dev' instead of 'liblua5.1-dev'
build-essential is already the newest version (12.6).
cargo is already the newest version (0.43.1-3~deb10u1).
libmagic-dev is already the newest version (1:5.35-4+deb10u2).
libgeoip-dev is already the newest version (1.6.12-1).
libhiredis-dev is already the newest version (0.14.0-3).
libcap-ng-dev is already the newest version (0.7.9-2).
libcap-ng0 is already the newest version (0.7.9-2).
libevent-dev is already the newest version (2.1.8-stable-4).
libnet1-dev is already the newest version (1.1.6+dfsg-3.1).
libpcap-dev is already the newest version (1.8.1-6+deb10u1).
libyaml-0-2 is already the newest version (0.2.1-1).
libyaml-dev is already the newest version (0.2.1-1).
liblua5.1-0-dev is already the newest version (5.1.5-8.1+b2).
make is already the newest version (4.2.1-1.2).
```

To direct input to this VM, click inside or press Ctrl+G.

installing dependencies libraries .

Figure 349: Installing suricata 1.

The screenshot shows a terminal window titled "nids@suricata: ~". The terminal is displaying the output of the command "sudo apt-get install suricata". The output includes package lists, dependency building, state information, additional packages to be installed (libhttp2, libhyperscan5, libluajit-5.1-2, libluajit-5.1-common, libprelude23, prelude-utils, python-simplejson, snort-rules-default, suricata-oinkmaster), suggested packages (snort, snort-pgsql, snort-mysql, libtcmalloc-minimal4), and NEW packages to be installed (libhttp2, libhyperscan5, libluajit-5.1-2, libluajit-5.1-common, libprelude23, prelude-utils, python-simplejson, snort-rules-default, suricata, suricata-oinkmaster). It also shows that 0 packages were upgraded, 10 were newly installed, 0 were removed, and 10 were not upgraded. A message indicates that 5,849 KB of archives need to be downloaded, and 25.9 MB of additional disk space will be used. The user is prompted with "Do you want to continue? [Y/n]". The letter "y" is highlighted with a red box, and the number "3" is also highlighted with a red box at the bottom right of the terminal window.

Figure 350: Installing suricata 2.

```
root@suricata:~# apt-get install libnetfilter-queue-dev libnetfilter-queue1 \
> libnetfilter-log-dev libnetfilter-log1 \
> libnfnetwork-dev libnfnetwork0
Reading package lists... Done
Building dependency tree
Reading state information... Done
libnetfilter-log-dev is already the newest version (1.0.1-1.1+b1).
libnetfilter-log1 is already the newest version (1.0.1-1.1+b1).
libnetfilter-queue-dev is already the newest version (1.0.3-1).
libnetfilter-queue1 is already the newest version (1.0.3-1).
libnfnetwork-dev is already the newest version (1.0.1-3+b1).
libnfnetwork-dev set to manually installed.
libnfnetwork0 is already the newest version (1.0.1-3+b1).
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
root@suricata:~#
```

7. `apt-get install rustc cargo` 5 For Rust support.

8. Reading package lists... Done
Building dependency tree
Reading state information... Done
cargo is already the newest version (0.43.1-3~deb10u1).
rustc is already the newest version (1.41.1+dfsg1-1~deb10u1).
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded. 6

root@suricata:~# cargo install --force --debug --version 0.14.1 cbindgen
Updating crates.io index
Fetch [=====] 28.74%, 897.77KiB/s
13. Setting up nftables for Linux

Figure 351: Installing Suricata 3.

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Activities Terminal" and the date and time are "Thu 12:05". The user is root and the command being run is:

```
root@suricata:~# echo "deb http://http.debian.net/debian buster-backports main" > /etc/apt/sources.list.d/backports.list
```

Line 14 shows the command "apt-get update" being run, which is highlighted with a red box. The output of the update command shows various package downloads from the Debian repositories.

Below the terminal window, there is a tooltip with the text:

The following is an example of installing Suricata 6.0 on Fedora. If you wish to install 5.0 instead, change the

At the bottom of the screen, there is a status bar with the text "To direct input to this VM, click inside or press Ctrl+G." and a set of icons for VM control.

Figure 352: Installing Suricata 4.

```
root@suricata:~# apt-get install suricata -t buster-backports
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libprelude23 prelude-utils python-simplejson suricata-oinkmaster
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  libbpf4.19 libhtp2 python3-simplejson python3-yaml suricata-update
Suggested packages:
  libtcmalloc-minimal4
The following NEW packages will be installed:
  libbpf4.19 python3-simplejson python3-yaml suricata-update
The following packages will be upgraded:
  libhtp2 suricata
2 upgraded, 4 newly installed, 0 to remove and 169 not upgraded.
Need to get 2,877 kB of archives.
After this operation, 3,025 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:4 http://security.debian.org/debian-security buster/updates/main amd64 libbpf4.19 amd64 4.19.269-1 [656 kB]
Get:5 http://deb.debian.org/debian buster/main amd64 python3-simplejson amd64 3.16.0-1 [60.7 kB]
25. Suricata Developer Guide
The following is an example of installing Suricata 6.0 on
CentOS. If you wish to install 5.0 instead, change the
To direct input to this VM, click inside or press Ctrl+G.
```

Figure 353: Installing suricata 5.

```
root@suricata:/etc/suricata/rules# suricata-update |10
23/2/2023 -- 12:15:24 - <Info> -- Using data-directory /var/lib/suricata.
23/2/2023 -- 12:15:24 - <Info> -- Using Suricata configuration /etc/suricata/suricata.yaml
23/2/2023 -- 12:15:24 - <Info> -- Using /etc/suricata/rules for Suricata provided rules.
23/2/2023 -- 12:15:24 - <Info> -- Found Suricata version 6.0.1 at /usr/bin/suricata.
23/2/2023 -- 12:15:24 - <Info> -- Loading /etc/suricata/suricata.yaml
23/2/2023 -- 12:15:24 - <Info> -- Disabling rules for protocol http2
23/2/2023 -- 12:15:24 - <Info> -- Disabling rules for protocol modbus
23/2/2023 -- 12:15:24 - <Info> -- Disabling rules for protocol dnp3
23/2/2023 -- 12:15:24 - <Info> -- Disabling rules for protocol enip
23/2/2023 -- 12:15:24 - <Info> -- No sources configured, will use Emerging Threats Open
23/2/2023 -- 12:15:24 - <Info> -- Fetching https://rules.emergingthreats.net/open/suricata-6.0.1/emerging.rules.tar.gz.
6.9 100% - 3740080/3740080                                Updating suricata rules
23/2/2023 -- 12:15:30 - <Info> -- Done.
23/2/2023 -- 12:15:30 - <Info> -- Loading distribution rule file /etc/suricata/rules/app-layer-events.rules
23/2/2023 -- 12:15:30 - <Info> -- Loading distribution rule file /etc/suricata/rules/decoder-events.rules
23/2/2023 -- 12:15:30 - <Info> -- Loading distribution rule file /etc/suricata/rule-sets/Using other rulesets
```

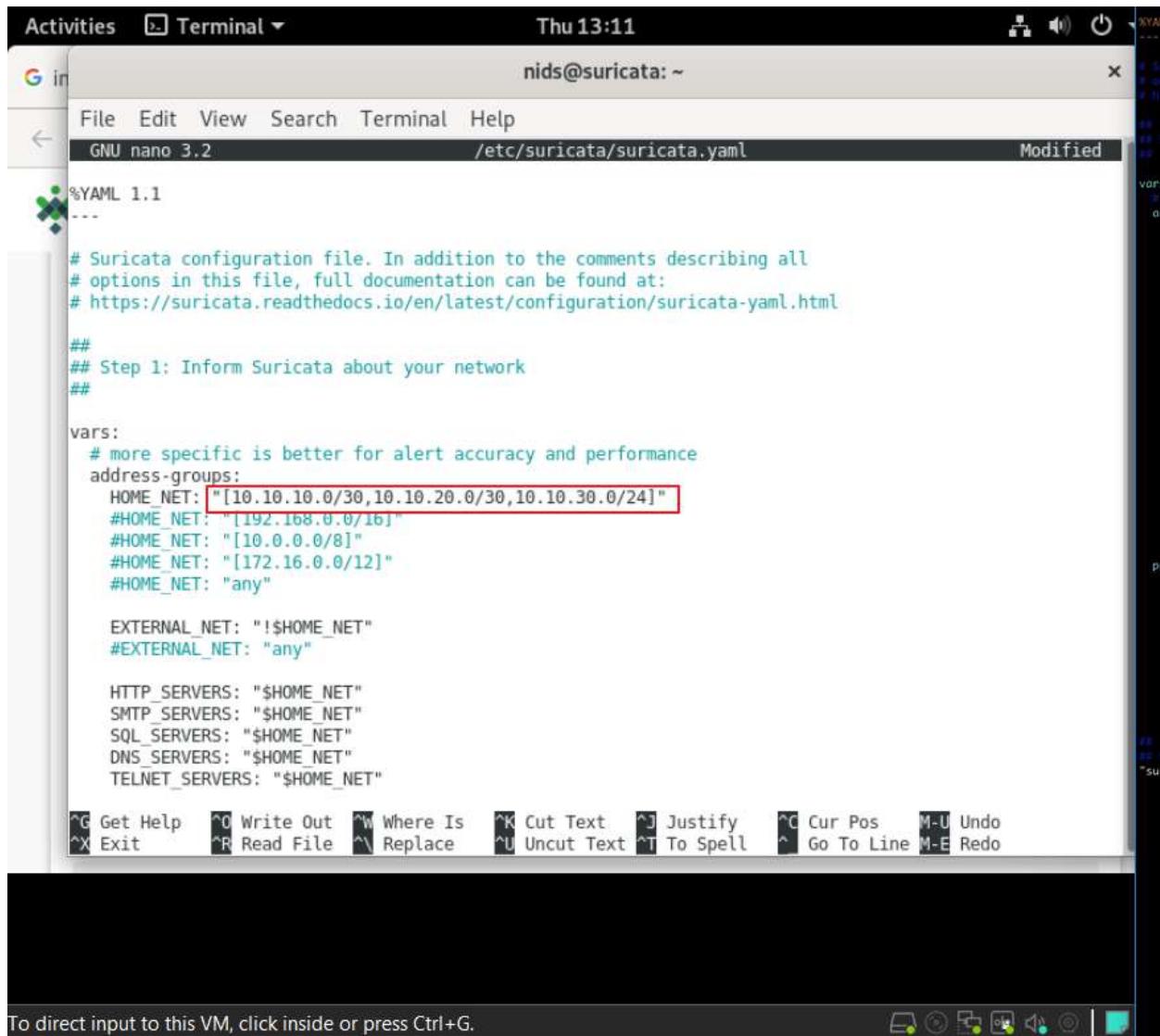
Figure 354: Installing suricata 6.

Configuration of suricata .yml

The screenshot shows a terminal window titled "Terminal" with the command "nids@suricata: ~". The terminal displays the output of the "ls -l" command, listing several files in the "/etc/suricata/" directory. A red box highlights the command "nano suricata.yaml", which is being edited. Red annotations provide instructions: "go to directory cd /etc/suricata/" points to the directory listing, and "editing suricata yaml" points to the command being edited.

```
File Edit View Search Terminal Help
root@suricata:/etc/suricata# ls -l
total 92
-rw----- 1 root root 3327 Dec 21 04:12 classification.config
-rw----- 1 root root 1375 Dec 21 04:12 reference.config
drwxr-xr-x 2 root root 4096 Dec 21 06:48 rules
-rw----- 1 root root 74802 Dec 21 04:12 suricata.yaml
-rw----- 1 root root 1644 Dec 21 04:12 threshold.config
root@suricata:/etc/suricata#
root@suricata:/etc/suricata# nano suricata.yaml
```

Figure 355: Configuration of suricata yml.



```
%YAML 1.1
---
# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://suricata.readthedocs.io/en/latest/configuration/suricata-yaml.html

##
## Step 1: Inform Suricata about your network
##

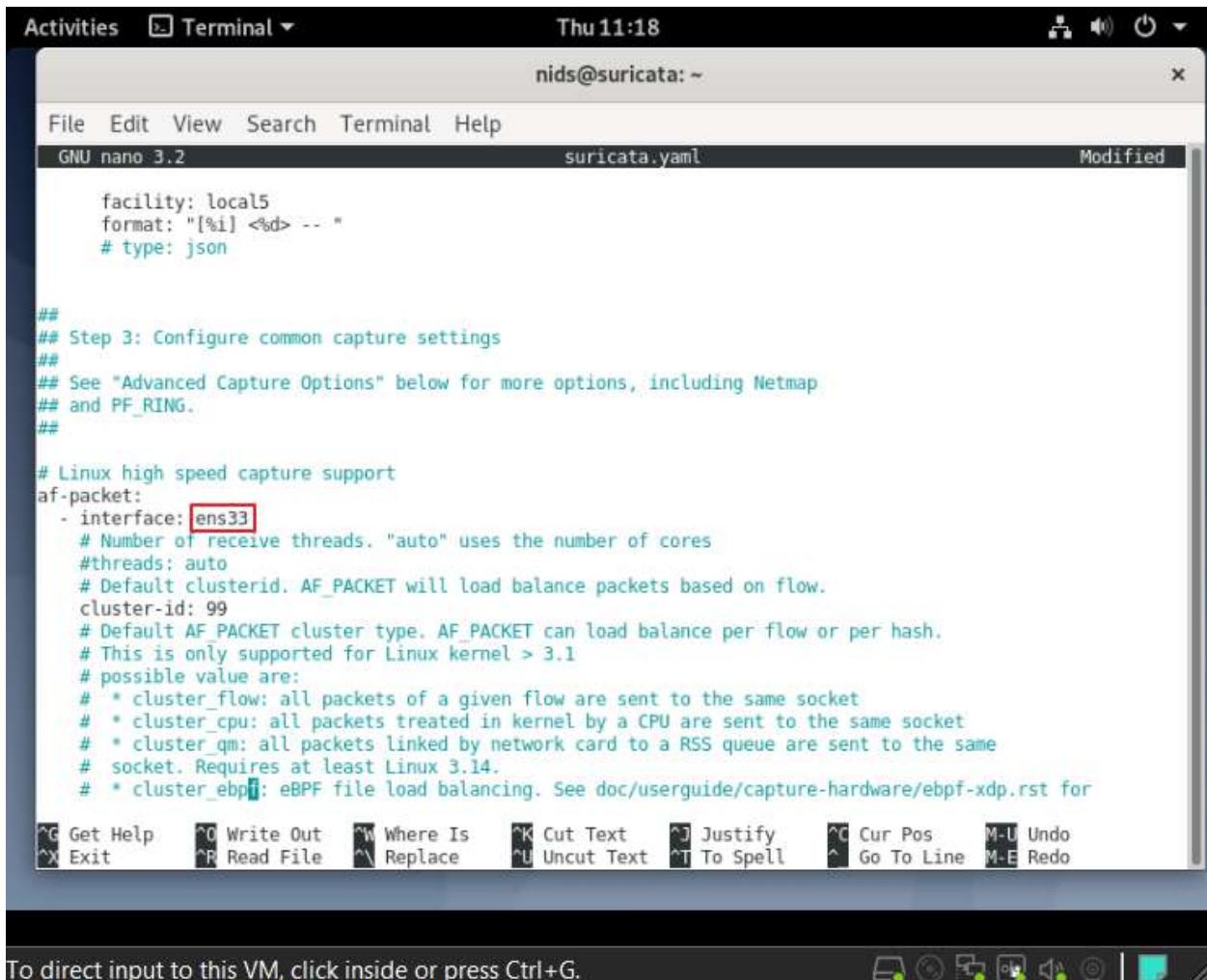
vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    HOME_NET: "[10.10.10.0/30,10.10.20.0/30,10.10.30.0/24]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"

    EXTERNAL_NET: "!$HOME_NET"
    #EXTERNAL_NET: "any"

    HTTP_SERVERS: "$HOME_NET"
    SMTP_SERVERS: "$HOME_NET"
    SQL_SERVERS: "$HOME_NET"
    DNS_SERVERS: "$HOME_NET"
    TELNET_SERVERS: "$HOME_NET"

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^ Go To Line M-E Redo
```

Figure 356: Configuration of suricata.yml.



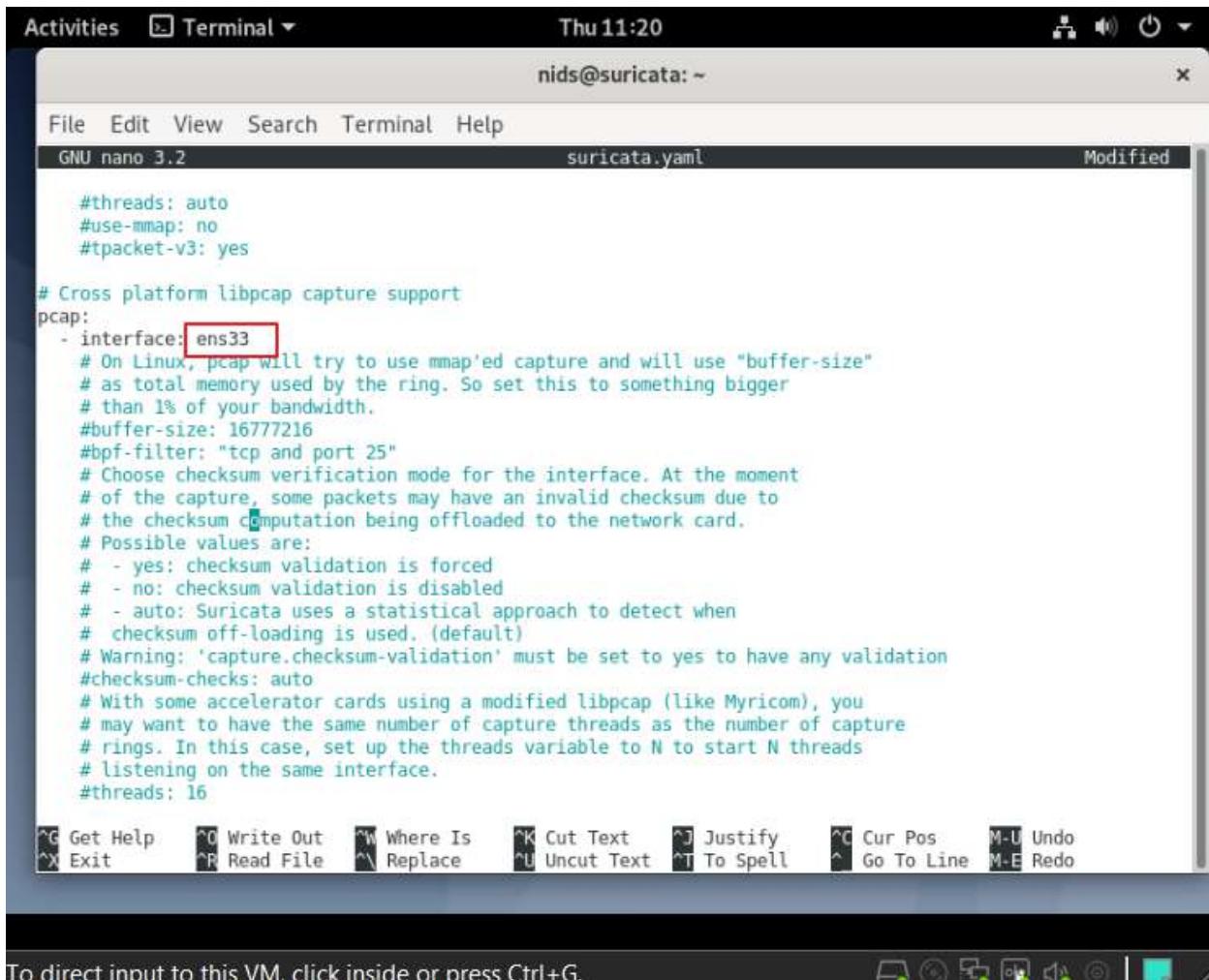
```
facility: local5
format: "[%i] <%d> -- "
# type: json

##
## Step 3: Configure common capture settings
##
## See "Advanced Capture Options" below for more options, including Netmap
## and PF_RING.
##

# Linux high speed capture support
af-packet:
  - interface: ens33
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
    # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per hash.
    # This is only supported for Linux kernel > 3.1
    # possible value are:
    # * cluster_flow: all packets of a given flow are sent to the same socket
    # * cluster_cpu: all packets treated in kernel by a CPU are sent to the same socket
    # * cluster_qm: all packets linked by network card to a RSS queue are sent to the same
    #   socket. Requires at least Linux 3.14.
    # * cluster_ebpf: eBPF file load balancing. See doc/userguide/capture-hardware/ebpf-xdp.rst for
      # details.
```

To direct input to this VM, click inside or press Ctrl+G.

Figure 357: Configuration of suricata.yml



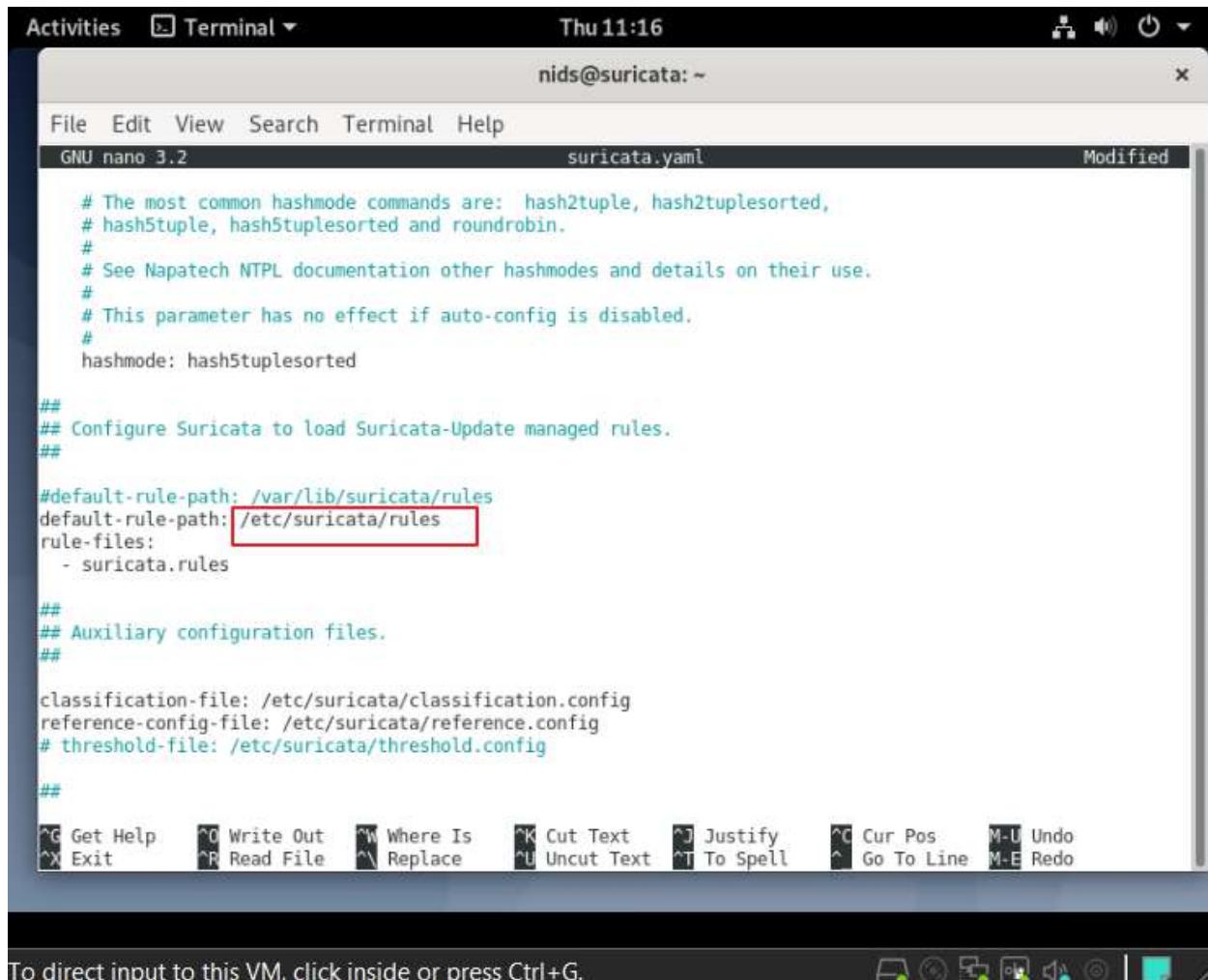
```
#threads: auto
#use-mmap: no
#tpacket-v3: yes

# Cross platform libpcap capture support
pcap:
  - interface: ens33
    # On Linux, pcap will try to use mmap'ed capture and will use "buffer-size"
    # as total memory used by the ring. So set this to something bigger
    # than 1% of your bandwidth.
    #buffer-size: 16777216
    #bpf-filter: "tcp and port 25"
    # Choose checksum verification mode for the interface. At the moment
    # of the capture, some packets may have an invalid checksum due to
    # the checksum computation being offloaded to the network card.
    # Possible values are:
    # - yes: checksum validation is forced
    # - no: checksum validation is disabled
    # - auto: Suricata uses a statistical approach to detect when
    # checksum off-loading is used. (default)
    # Warning: 'capture.checksum-validation' must be set to yes to have any validation
    #checksum-checks: auto
    # With some accelerator cards using a modified libpcap (like Myricom), you
    # may want to have the same number of capture threads as the number of capture
    # rings. In this case, set up the threads variable to N to start N threads
    # listening on the same interface.
    #threads: 16

^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos   M-U Undo
^X Exit      ^R Read File   ^\ Replace    ^U Uncut Text ^T To Spell   ^  Go To Line M-E Redo
```

To direct input to this VM, click inside or press Ctrl+G.

Figure 358: Configuration of suricata.yml



```
# The most common hashmode commands are: hash2tuple, hash2tuplesorted,
# hash5tuple, hash5tuplesorted and roundrobin.
#
# See Napatech NTPL documentation other hashmodes and details on their use.
#
# This parameter has no effect if auto-config is disabled.
#
hashmode: hash5tuplesorted

##
## Configure Suricata to load Suricata-Update managed rules.
##

#default-rule-path: /var/lib/suricata/rules
default-rule-path: /etc/suricata/rules
rule-files:
  - suricata.rules

##
## Auxiliary configuration files.
##
classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config

##
```

Figure 359: Configuration of suricata.yml

The screenshot shows a terminal window titled "Activities Terminal" running on a Linux system. The terminal session is as follows:

```
root@suricata:~# systemctl start suricata 11
root@suricata:~# systemctl status suricata 12
● suricata.service - Suricata IDS/IDP daemon
   Loaded: loaded (/lib/systemd/system/suricata.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-02-23 13:19:13 EST; 10s ago
     Docs: man:suricata(8)
           man:suricatasc(8)
           https://suricata-ids.org/docs/
   Process: 6416 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid
   Main PID: 6417 (Suricata-Main)
     Tasks: 7 (limit: 1736)
    Memory: 55.4M
      CGroup: /system.slice/suricata.service
              └─6417 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricata.yaml --pidfile /run/suricata.pid

Feb 23 13:19:13 suricata systemd[1]: Starting Suricata IDS/IDP daemon...
Feb 23 13:19:13 suricata suricata[6416]: 23/2/2023 -- 13:19:13 - <Notice> - This is Suricata version 6.0
Feb 23 13:19:13 suricata systemd[1]: Started Suricata IDS/IDP daemon.

suricata successfully running.

root@suricata:~# systemctl enable suricata 13
Synchronizing state of suricata.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
root@suricata:~#
```

A red box highlights the output of the "systemctl status suricata" command, which includes the message "suricata successfully running.". Another red box highlights the command "systemctl enable suricata". A third red box highlights the number "11" next to the first command, "12" next to the second, and "13" next to the third.

Figure 360: Configuration of suricata .yml

Installing and configuring filebeat for Suricata

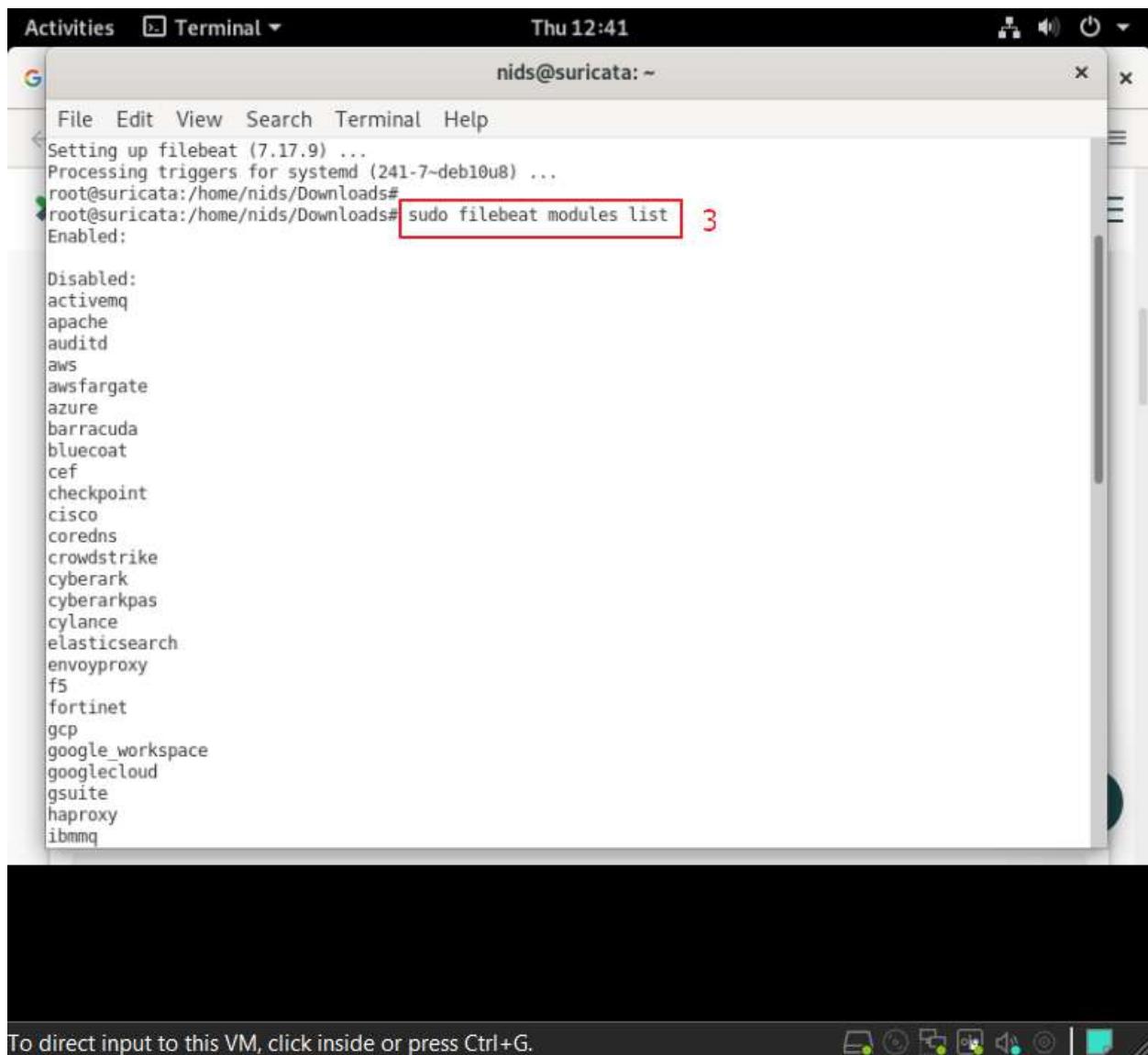
The screenshot shows a terminal window titled "Terminal" with the command "nids@suricata: ~". The terminal displays the following commands and output:

```
root@suricata:/home/nids/Downloads# curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.17.9-amd64.deb
% total % Received % Xferd Average Speed   Time   Time   Time Current
          Dload Upload Total Spent   Left Speed
100 33.8M 100 33.8M    0     0 18.4M      0:00:01 0:00:01 --:--:-- 18.4M
root@suricata:/home/nids/Downloads# ls
emerging.rules  filebeat-7.17.9-amd64.deb
root@suricata:/home/nids/Downloads# sudo dpkg -i filebeat-7.17.9-amd64.deb
Selecting previously unselected package filebeat.
(Reading database ... 145736 files and directories currently installed.)
Preparing to unpack filebeat-7.17.9-amd64.deb ...
Unpacking filebeat (7.17.9) ...
Setting up filebeat (7.17.9) ...
Processing triggers for systemd (241-7~deb10u8) ...
root@suricata:/home/nids/Downloads#
```

The command "curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.17.9-amd64.deb" is highlighted with a red box labeled "1". The command "sudo dpkg -i filebeat-7.17.9-amd64.deb" is highlighted with a red box labeled "2".

To direct input to this VM, click inside or press Ctrl+G.

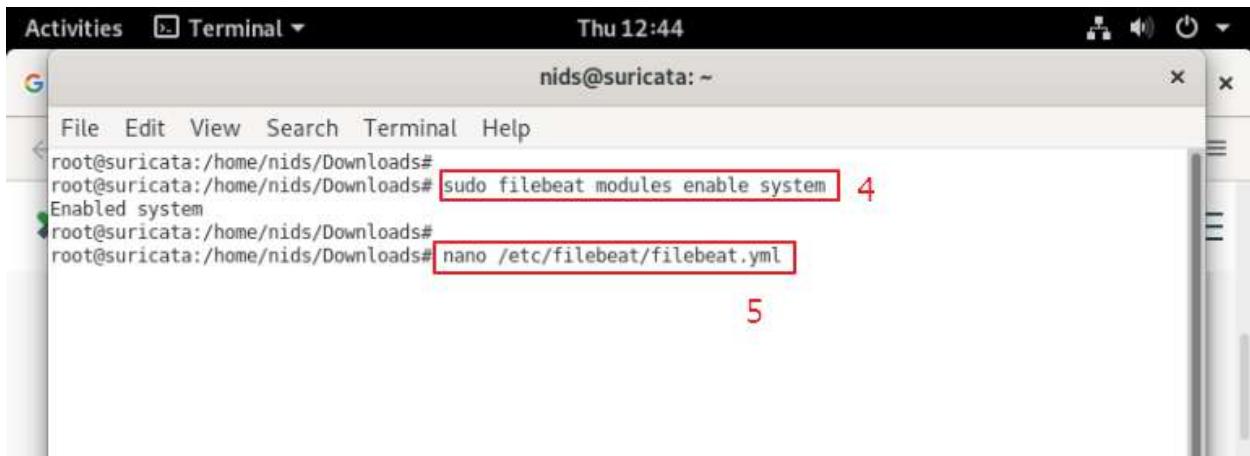
Figure 361: Installing and configuring filebeat for Suricata.



```
Setting up filebeat (7.17.9) ...
Processing triggers for systemd (241-7~deb10u8) ...
root@suricata:/home/nids/Downloads# sudo filebeat modules list 3
Enabled:

Disabled:
activemq
apache
auditd
aws
awsfargate
azure
barracuda
bluecoat
cef
checkpoint
cisco
coredns
crowdstrike
cyberark
cyberarkpas
cylance
elasticsearch
envoyproxy
f5
fortinet
gcp
google_workspace
googlecloud
gsuite
haproxy
ibmmq
```

Figure 362: Installing and configuring filebeat for Suricata.



The screenshot shows a terminal window titled "Terminal" with the command "nids@suricata: ~". The terminal displays the following commands:

```
File Edit View Search Terminal Help
root@suricata:/home/nids/Downloads#
root@suricata:/home/nids/Downloads# sudo filebeat modules enable system 4
Enabled system
root@suricata:/home/nids/Downloads# nano /etc/filebeat/filebeat.yml
5
```

Two lines of the command history are highlighted with red boxes: "sudo filebeat modules enable system" and "nano /etc/filebeat/filebeat.yml". A red number "4" is placed next to the second command, and a red number "5" is placed next to the third command.

Figure 363; Installing and configuring filebeat for Suricata.

The screenshot shows a terminal window titled "Activities Terminal" running on a Linux system. The window title bar indicates the date and time as "Thu 12:51". The terminal prompt is "nids@suricata: ~". The file being edited is "/etc/filebeat/filebeat.yml", which is marked as "Modified". The configuration file contains the following code:

```
filebeat.inputs:
  # Each - is an input. Most options can be set at the input level, so
  # you can use different inputs for various configurations.
  # Below are the input specific configurations.

  # filestream is an input for collecting log messages from files.
  - type: log

    # Unique ID among all inputs, an ID is required.
    #id: my-filestream-id
    id: suricata-id
    # Change to true to enable this input configuration.
    enabled: false

    # Paths that should be crawled and fetched. Glob based paths.
    paths:
      - /var/log/suricata/eve.json
      # - /var/log/*.log
      # - c:\programdata\elasticsearch\logs\*
    fields:
      event.type: suricata
    # Exclude lines. A list of regular expressions to match. It drops the lines that are
    # matching any regular expression from the list.
    #exclude_lines: ['^DBG']

    # Include lines. A list of regular expressions to match. It exports the lines that are
    # matching any regular expression from the list.
```

The terminal window includes a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The bottom of the window shows a toolbar with various icons and a status bar with keyboard shortcuts like "Get Help", "Write Out", "Where Is", "Cut Text", "Justify", "Cur Pos", "Undo", "Exit", "Read File", "Replace", "Uncut Text", "To Spell", "Go To Line", and "Redo".

Figure 364: Installing and configuring filebeat for Suricata.

The screenshot shows a terminal window titled "nids@suricata: ~" running on a Linux system. The window title bar also displays "Thu 12:58". The terminal is displaying the contents of the file "/etc/filebeat/filebeat.yml". The configuration file is in YAML format and includes settings for hosts, protocol, authentication, and Logstash output. A red rectangular box highlights the "Logstash Output" section, which contains the "output.logstash:" configuration block.

```
# hosts: ["localhost:9200"]
# Protocol - either 'http' (default) or 'https'.
#protocol: "https"

# Authentication credentials - either API key or username/password.
#api_key: "id:api_key"
#username: "elastic"
#password: "changeme"

# Logstash Output -----
#output.logstash:
#  # The Logstash hosts
hosts: ["10.10.30.3:5044"]
loadbalance: true
ssl.enabled: true
# optional SSL. By default is off.
# List of root certificates for HTTPS server verifications
#ssl.certificateAuthorities: ["/etc/pki/root/ca.pem"]

# Certificate for SSL client authentication
#ssl.certificate: "/etc/pki/client/cert.pem"

# Client Certificate Key
#ssl.key: "/etc/pki/client/cert.key"
```

Figure 365: Installing and configuring filebeat for Suricata.

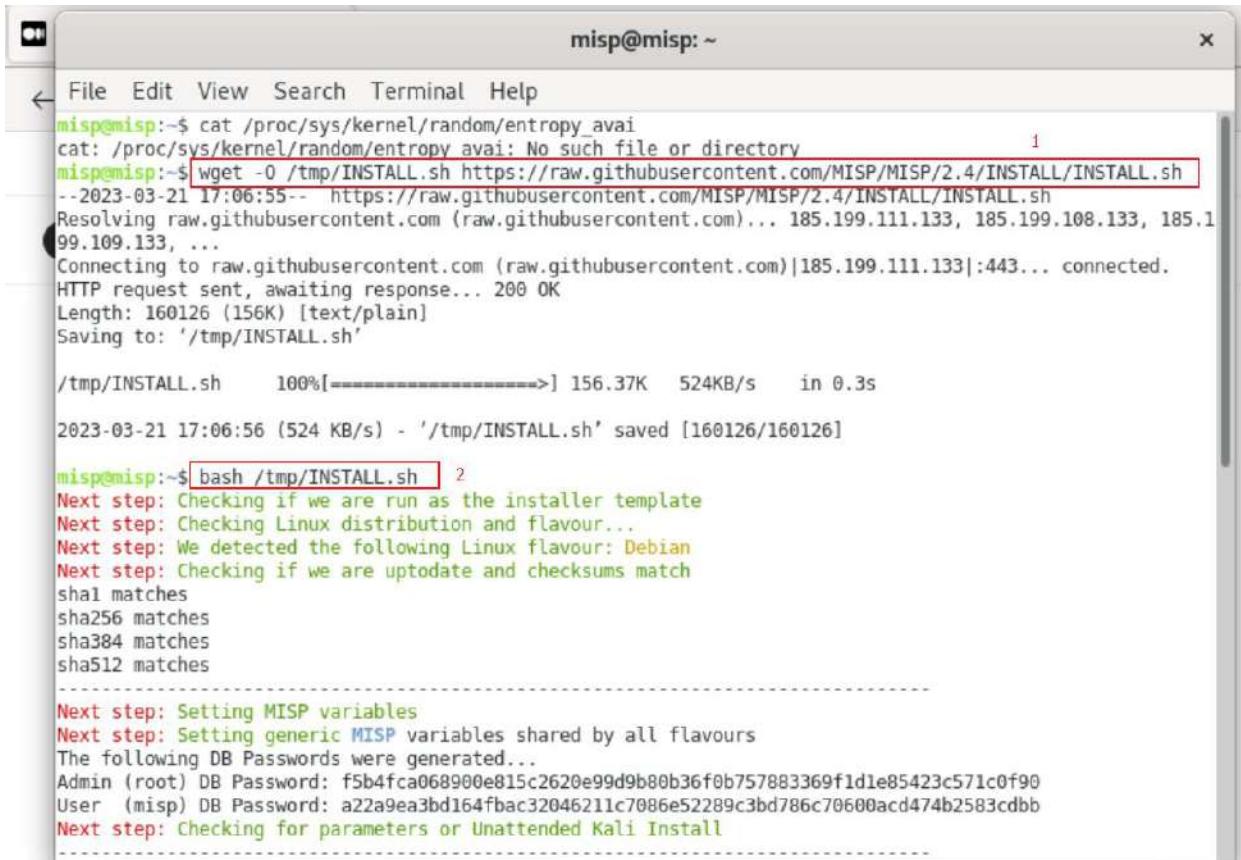
The screenshot shows a terminal window titled "Activities Terminal" with the command "nids@suricata: ~". The terminal displays the following sequence of commands and their outputs:

- File Edit View Search Terminal Help
- root@suricata:/home/nids/Downloads# systemctl enable filebeat **6**
- Synchronizing state of filebeat.service with SysV service script with /lib/systemd/systemd-sysv-install.
- Executing: /lib/systemd/systemd-sysv-install enable filebeat
- Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /lib/systemd/system/filebeat.service.
- root@suricata:/home/nids/Downloads# systemctl start filebeat **7**
- root@suricata:/home/nids/Downloads# systemctl status filebeat **8**
- filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
 - Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
 - Active: **active (running)** since Thu 2023-02-23 13:04:45 EST; 9s ago
 - Docs: <https://www.elastic.co/beats/filebeat>
 - Main PID: 6302 (filebeat)
 - Tasks: 7 (limit: 1736)
 - Memory: 65.1M
 - CGroup: /system.slice/filebeat.service
 - └─6302 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/filebeat/filebeat.yml -
- filebeat successfully running.**
- Feb 23 13:04:48 suricata filebeat[6302]: 2023-02-23T13:04:48.554-0500 INFO [input.harvesie]
- Feb 23 13:04:48 suricata filebeat[6302]: 2023-02-23T13:04:48.555-0500 INFO [input.harvesie]
- Feb 23 13:04:48 suricata filebeat[6302]: 2023-02-23T13:04:48.555-0500 INFO [input.harvesie]
- Feb 23 13:04:48 suricata filebeat[6302]: 2023-02-23T13:04:48.555-0500 INFO [input.harvesie]
- Feb 23 13:04:48 suricata filebeat[6302]: 2023-02-23T13:04:48.555-0500 INFO [input.harvesie]
- Feb 23 13:04:48 suricata filebeat[6302]: 2023-02-23T13:04:48.556-0500 INFO [input.harvesie]
- Feb 23 13:04:48 suricata filebeat[6302]: 2023-02-23T13:04:48.597-0500 INFO [add_cloud_meta]
- Feb 23 13:04:48 suricata filebeat[6302]: 2023-02-23T13:04:48.637-0500 INFO [publisher_pipe]
- Feb 23 13:04:48 suricata filebeat[6302]: 2023-02-23T13:04:48.639-0500 INFO [publisher]
- Feb 23 13:04:48 suricata filebeat[6302]: 2023-02-23T13:04:48.639-0500 INFO [publisher]

lines 1-20/20 (END)

Figure 366: Installing and configuring filebeat for Suricata.

8.4.7 MISP Server



The screenshot shows a terminal window titled "misp@misp: ~". The terminal is displaying the process of installing the MISP server. It starts with a check for the "/proc/sys/kernel/random/entropy_avail" file, which is not found. Then, it uses "wget" to download the "INSTALL.sh" script from GitHub. The download is successful at 160126 bytes. After saving the script, it is executed with "bash /tmp/INSTALL.sh". The script performs several steps: checking if it's run as the installer template, detecting the Linux distribution (Debian), checking for updates and checksums, setting MISP variables, and generating database passwords for Admin and User. Finally, it checks for parameters or performs an unattended Kali install.

```
misp@misp:~$ cat /proc/sys/kernel/random/entropy_avail
cat: /proc/sys/kernel/random/entropy_avail: No such file or directory
misp@misp:~$ wget -O /tmp/INSTALL.sh https://raw.githubusercontent.com/MISP/MISP/2.4/INSTALL/INSTALL.sh
--2023-03-21 17:06:55-- https://raw.githubusercontent.com/MISP/MISP/2.4/INSTALL/INSTALL.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 160126 (156K) [text/plain]
Saving to: '/tmp/INSTALL.sh'

/tmp/INSTALL.sh      100%[=====] 156.37K   524KB/s   in 0.3s

2023-03-21 17:06:56 (524 KB/s) - '/tmp/INSTALL.sh' saved [160126/160126]

misp@misp:~$ bash /tmp/INSTALL.sh
Next step: Checking if we are run as the installer template
Next step: Checking Linux distribution and flavour...
Next step: We detected the following Linux flavour: Debian
Next step: Checking if we are uptodate and checksums match
sha1 matches
sha256 matches
sha384 matches
sha512 matches
-----
Next step: Setting MISP variables
Next step: Setting generic MISP variables shared by all flavours
The following DB Passwords were generated...
Admin (root) DB Password: f5b4fca068900e815c2620e99d9b80b36f0b757883369f1d1e85423c571c0f90
User (misp) DB Password: a22a9ea3bd164fbac32046211c7086e52289c3bd786c70600acd474b2583cdbb
Next step: Checking for parameters or Unattended Kali Install
```

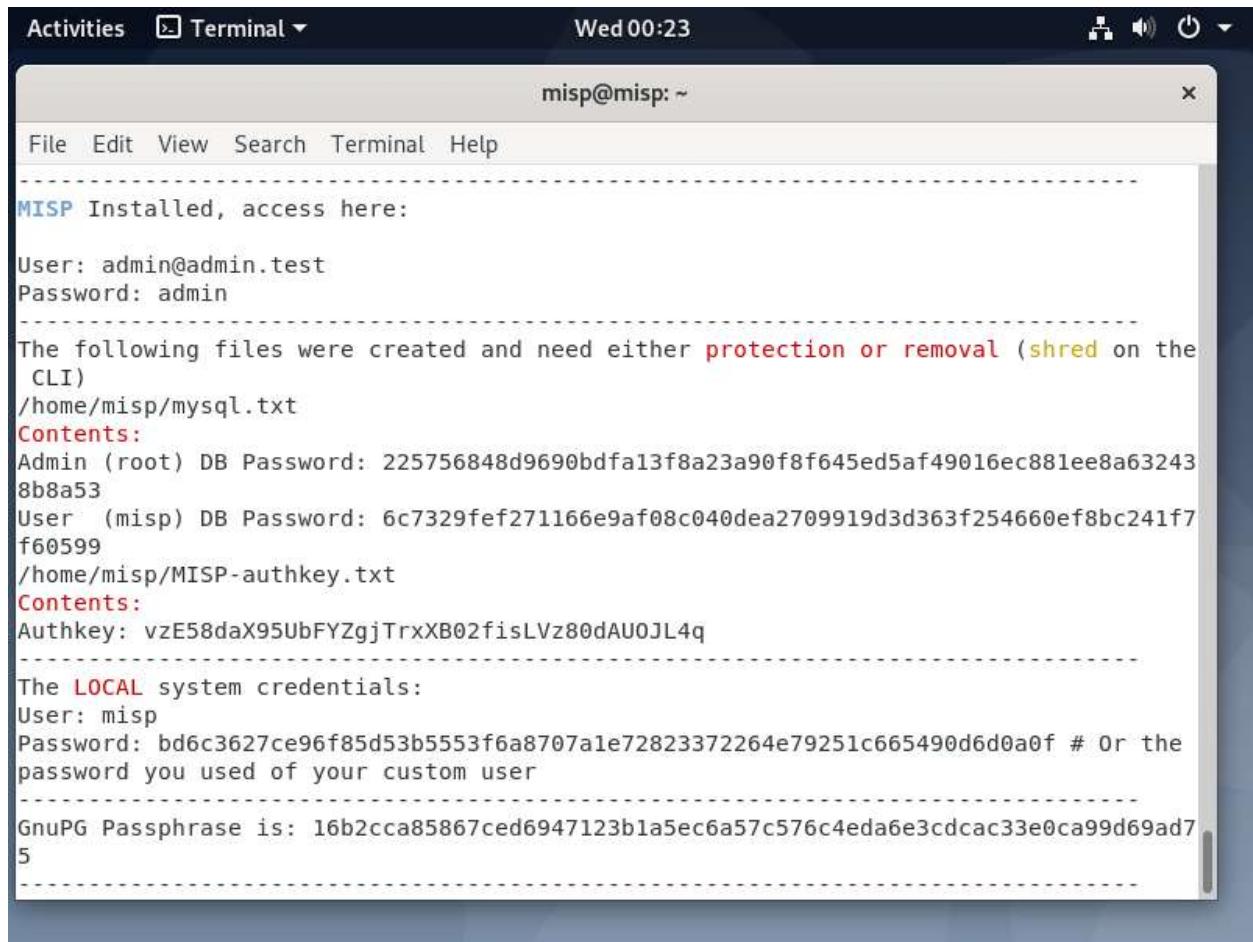
Figure 367: Installing MISP.

```
misp@misp: ~
File Edit View Search Terminal Help
DBUSER_ADMIN/DBPASSWORD ADMIN # MySQL admin user, default: root/opensslGeneratedPassword
DBUSER_MISP/DBPASSWORD_MISP # MISP database user, default: misp/opensslGeneratedPassword

You need to export the variable(s) to be taken into account. (or specified in-line when invoking INSTALL.sh)
-----
nisp@misp:~$ bash /tmp/INSTALL.sh -A | tee log.txt 3
Next step: Checking if we are run as the installer template
Next step: Checking Linux distribution and flavour...
Next step: We detected the following Linux flavour: Debian
Next step: Checking if we are uptodate and checksums match
sha1 matches
sha256 matches
sha384 matches
sha512 matches
-----
Next step: Setting MISP variables
Next step: Setting generic MISP variables shared by all flavours
The following DB Passwords were generated...
Admin (root) DB Password: 965d4b7b1ff871fc753847f4880b2479b8a097555f35c0153f3a769842c0c0cb
User (misp) DB Password: 62e0e62bcb7299cd99cbe951f99223af90310327bf3292b58f5ab0003353168d
Next step: Checking for parameters or Unattended Kali Install
Next step: Setting install options with given parameters.
all
Install on Debian testing fully supported.
Please report bugs/issues here: https://github.com/MISP/MISP/issues
-----
Proceeding with the installation of MISP core
-----
Checking for sudo and installing etckeeper

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
```

Figure 368: Installing MISP.



The screenshot shows a terminal window titled "misp@misp: ~" running on a Linux desktop environment. The terminal displays the output of a script or command related to the MISP installation process. Key text visible in the terminal includes:

- "MISP Installed, access here:"
- User: admin@admin.test
Password: admin
- The following files were created and need either protection or removal (shred on the CLI)
/home/misp/mysql.txt
- Contents:
Admin (root) DB Password: 225756848d9690bdःa13f8a23a90f8f645ed5af49016ec881ee8a63243
8b8a53
User (misp) DB Password: 6c7329fef271166e9af08c040dea2709919d3d363f254660ef8bc241f7
f60599
- /home/misp/MISP-authkey.txt
- Contents:
Authkey: vzE58daX95UbFYZgjTrxXB02fisLVz80dAU0JL4q
- The LOCAL system credentials:
User: misp
Password: bd6c3627ce96f85d53b5553f6a8707a1e72823372264e79251c665490d6d0a0f # Or the
password you used of your custom user
- GnuPG Passphrase is: 16b2cca85867ced6947123b1a5ec6a57c576c4eda6e3cdcac33e0ca99d69ad7
5

Figure 369: Installing MISP.

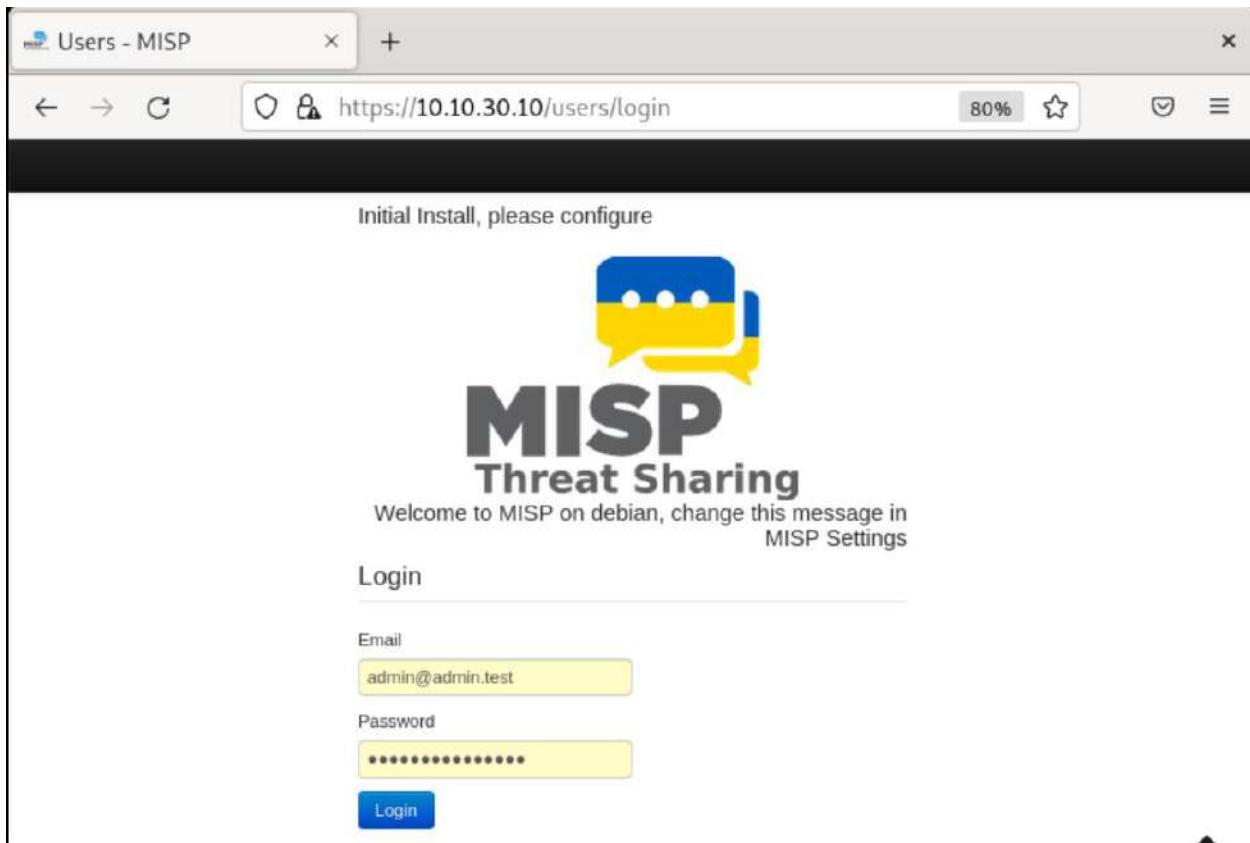


Figure 370: MISP web interface.

The screenshot shows the 'Events' index page of the MISP web interface. The left sidebar has a blue header 'List Events' and includes links for 'Add Event', 'Import from...', 'REST client', 'List Attributes', 'Search Attributes', 'View Proposals', 'Events with proposals', 'View delegation requests', 'View periodic summary', 'Export', and 'Automation'. The main content area is titled 'Events' and contains a search bar and navigation buttons ('previous', 'next'). A table lists five events:

	Creator org	Owner org	ID	Clusters	Tags	#Attr.	#Corr.	#Sightings	Creator user	Date	Info	Distribu...
<input type="checkbox"/>	X ORGNAME	ORGNAME	1612			8			admin@admin.test	2023-04-14	Domain	Communi...
<input type="checkbox"/>	X ORGNAME	ORGNAME	1613			9			admin@admin.test	2023-04-15	hash sha256	Communi...
<input type="checkbox"/>	X ORGNAME	ORGNAME	1610			5			admin@admin.test	2023-04-14	soc2	Communi...
<input type="checkbox"/>	X ORGNAME	ORGNAME	1615			4			admin@admin.test	2023-04-15	HASH SH1	Communi...
<input type="checkbox"/>	X ORGNAME	ORGNAME	1614			3			admin@admin.test	2023-04-15	Destination ip	Communi...

A red box highlights the fifth event entry. The text 'custom created events for testing purposes.' is displayed in red at the bottom of the table. Below the table, a message says 'Page 1 of 1, showing 5 records out of 5 total, starting on record 1, ending on 5'.

Figure 371: Event in MISP

8.4.8 Apache Server

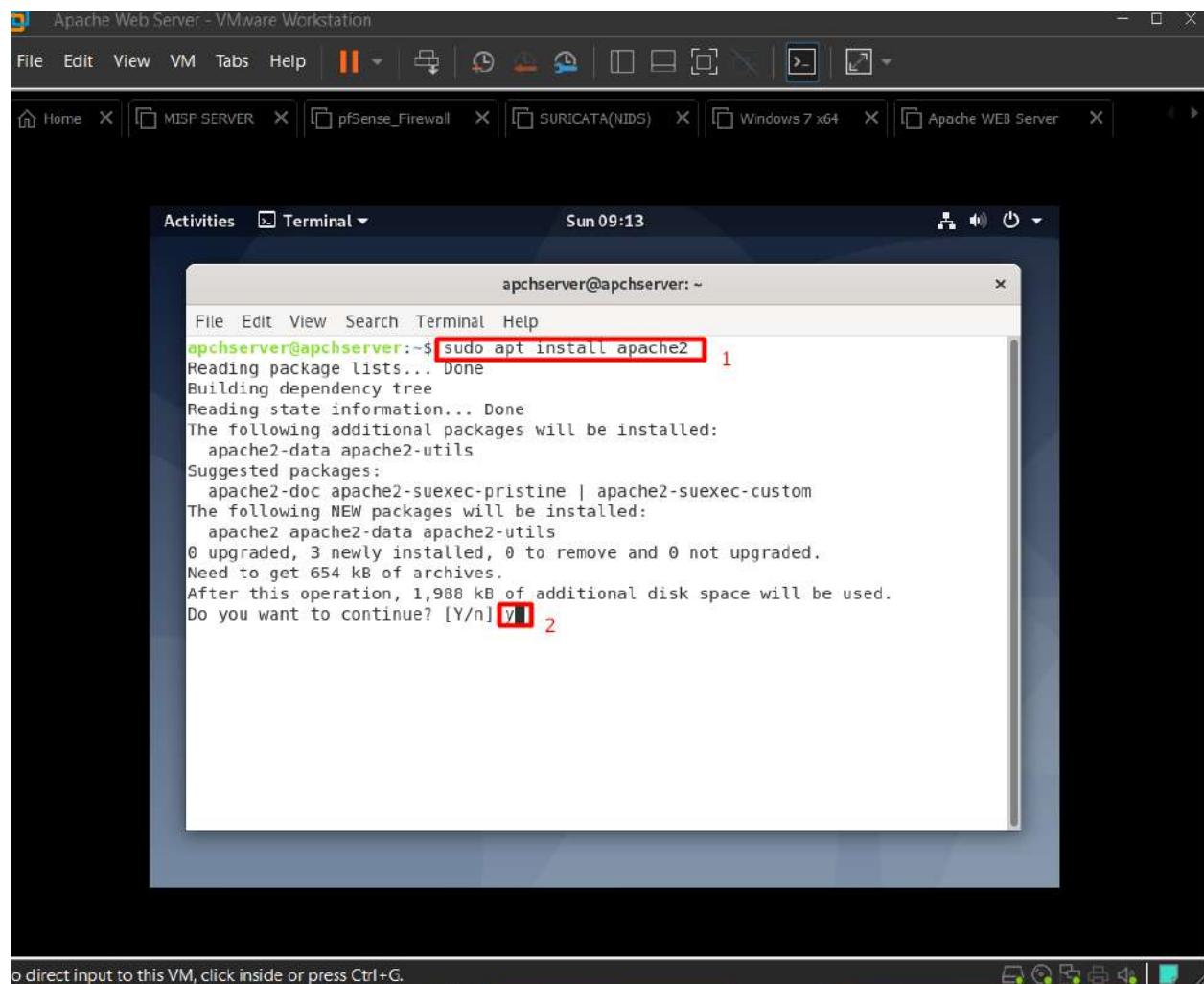


Figure 372: Installing Apache server.

```
apchserver@apchserver:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-data apache2-utils
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-data apache2-utils
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 654 kB of archives.
After this operation, 1,988 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://security.debian.org/debian-security buster/updates/main amd64 apache2-data all 2.4.38-3+deb10u7 [165 kB]
Get:2 http://security.debian.org/debian-security buster/updates/main amd64 apache2-utils amd64 2.4.38-3+deb10u7 [237 kB]
Get:3 http://security.debian.org/debian-security buster/updates/main amd64 apache2 amd64 2.4.38-3+deb10u7 [252 kB]
Fetched 654 kB in 2s (417 kB/s)
Selecting previously unselected package apache2-data.
(Reading database ... 139865 files and directories currently installed.)
Preparing to unpack .../apache2-data_2.4.38-3+deb10u7_all.deb ...
```

Figure 373: Installing Apache server.

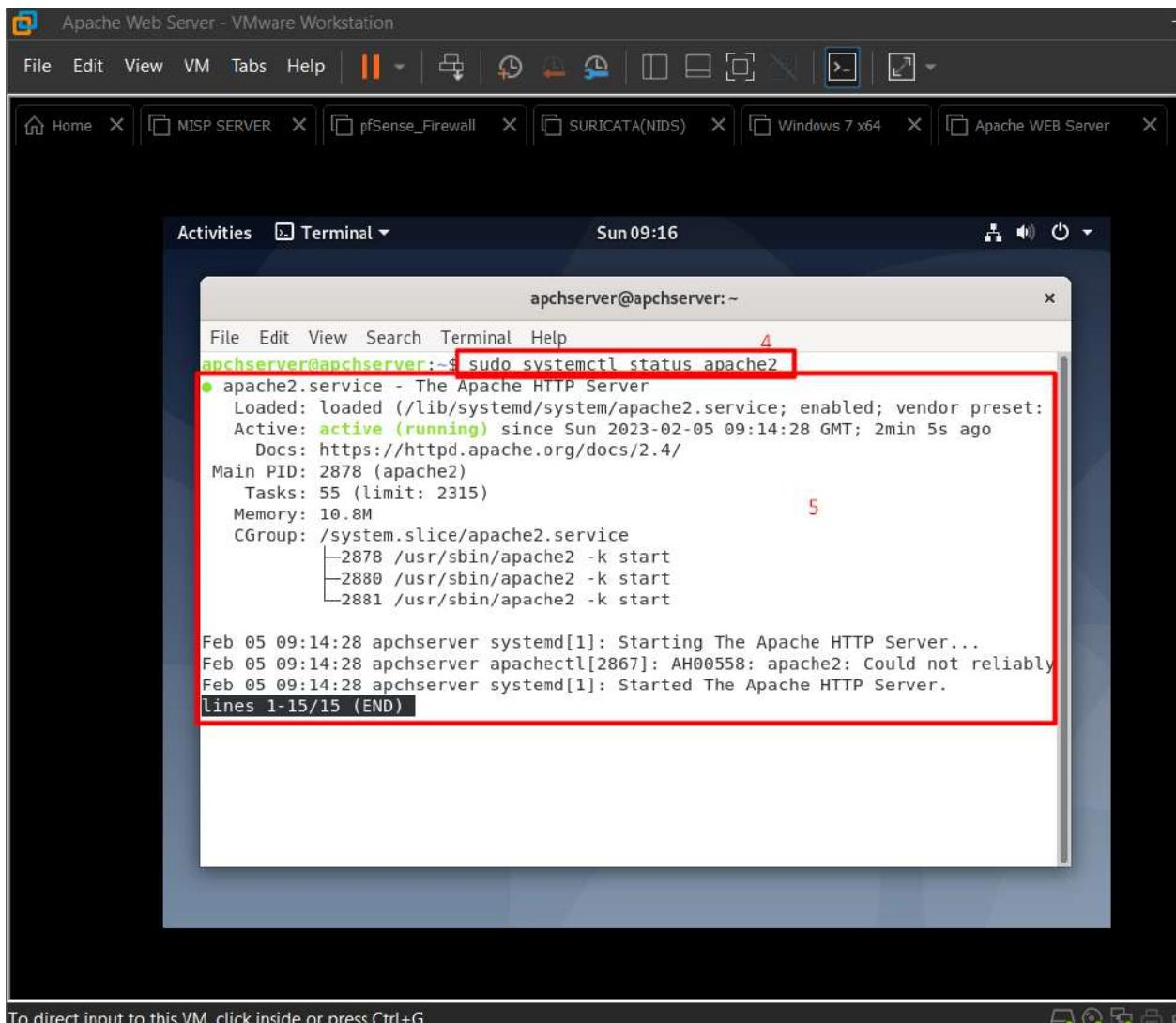


Figure 374: Installing Apache server.

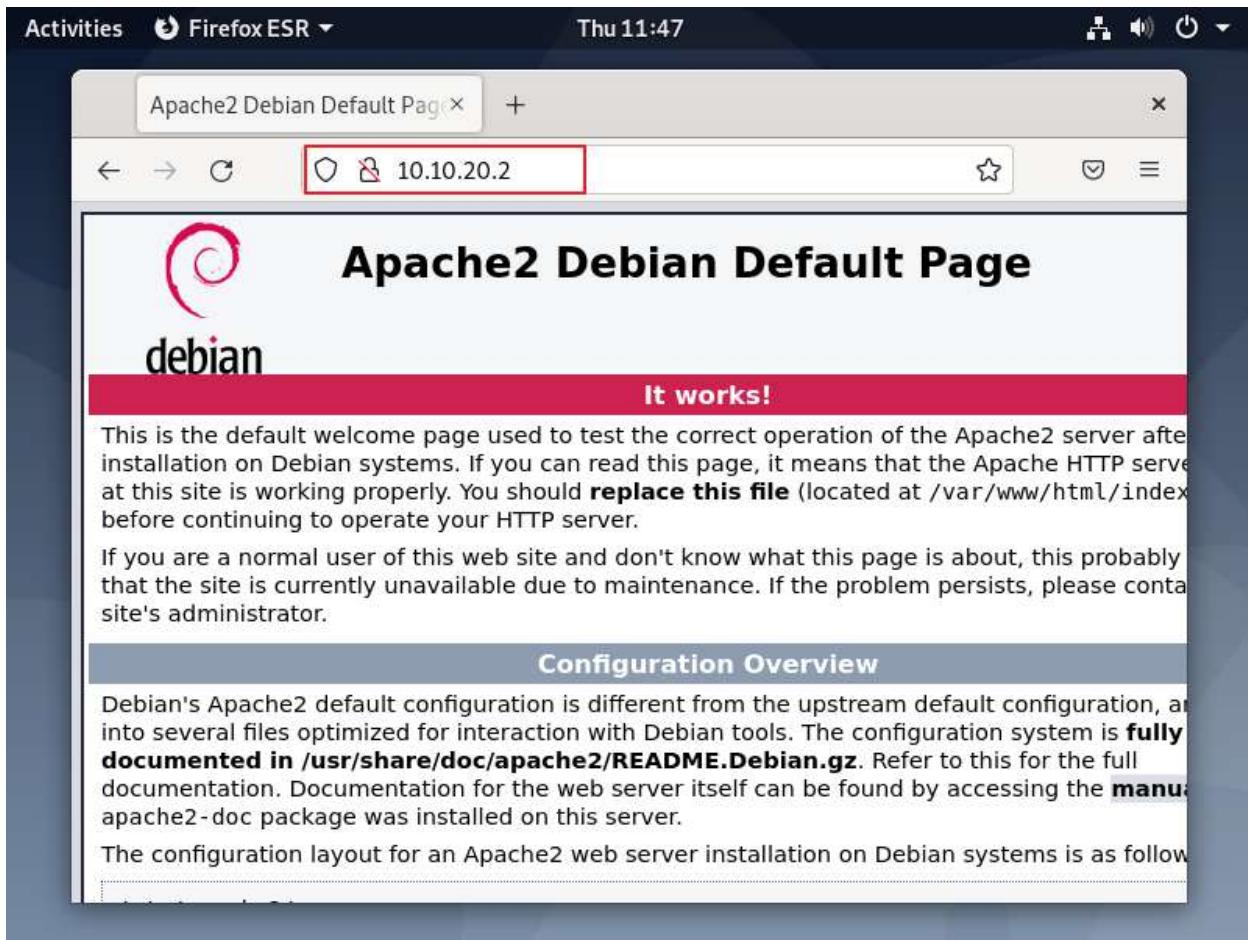


Figure 375: Apache server web page.

Installing and configuring filebeat for Apache web server

The screenshot shows a terminal window with two tabs. The current tab is titled 'apchserver@apchserver: ~/Downloads'. The command history in the terminal shows:

```
apchserver@apchserver:~/Downloads$ curl -L -o https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.17.9-amd64.deb
apchserver@apchserver:~/Downloads$ ls -l
total 39496
-rw-r--r-- 1 apchserver apchserver 35503128 Apr 16 21:27 filebeat-7.17.9-amd64.deb
-rw-r--r-- 1 apchserver apchserver 3772416 Apr 16 21:21 filebeat-8.7.0-amd64.deb
drwxr-xr-x 5 apchserver apchserver 4096 Apr 16 18:21 rpds_py-0.7.1
-rw-r--r-- 1 apchserver apchserver 1146491 Apr 16 18:23 rpds_py-0.7.1-pp39-pypy39_pp73-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
-rw-r--r-- 1 apchserver apchserver 15355 Apr 16 18:21 rpds_py-0.7.1.tar.gz
apchserver@apchserver:~/Downloads$ sudo dpkg -i filebeat-7.17.9-amd64.deb
```

The password entry 'apchserver' is highlighted with a red box. The command 'sudo dpkg -i filebeat-7.17.9-amd64.deb' is also highlighted with a red box.

Figure 376: Installing and configuring filebeat for Apache web server.

The screenshot shows a terminal window titled "apchserver@apchserver: ~/Downloads". The terminal has two tabs open: "apchserver@apchserver: ~/elastalert/e..." and "apchserver@apchserver: ~/Downloads". The user is performing the following steps:

- curl -L -o https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.17.9-amd64.deb
- ls -l
- sudo dpkg -i filebeat-7.17.9-amd64.deb
- [sudo] password for apchserver: (redacted)
- Selecting previously unselected package filebeat.
- (Reading database ... 149931 files and directories currently installed.)
- Preparing to unpack filebeat-7.17.9-amd64.deb ...
- Unpacking filebeat (7.17.9) ...
- Setting up filebeat (7.17.9) ...
- Processing triggers for systemd (243-7 deb30u0) ...
- sudo filebeat modules enable apache
- Enabled apache
- sudo filebeat modules enable system
- Enabled system
- sudo nano /etc/filebeat/filebeat.yml

Figure 377: Installing and configuring filebeat for Apache web server.

The screenshot shows a terminal window titled "apchserver@apchserver: ~/Downloads". The window has two tabs: "apchserver@apchserver: ~/elastal..." and "apchserver@apchserver: ~/Downlo...". The active tab displays the file "/etc/filebeat/filebeat.yml" using the "GNU nano 3.2" editor. The configuration file contains YAML code for setting up Filebeat inputs. A red box highlights the "type: log" line under the "filest" section. Another red box highlights the "id: apache-id" line under the "Unique ID among all inputs" section. A third red box highlights the list of log files under the "paths:" section, which includes "/var/log/apache2/access.log", "/var/log/apache2/other_vhosts_access.log", "/var/log/httpd/access_log", "/var/log/apache2/error.log", and "/var/log/httpd/error_log".

```
# you can use different inputs for various configurations.
# Below are the input specific configurations.

# filest is an input for collecting log messages from files.
- type: log

# Unique ID among all inputs, an ID is required.
id: apache-id

# Change to true to enable this input configuration.
enabled: false

# Paths that should be crawled and fetched. Glob based paths.
paths:
  #- /var/log/*.log
  #- c:\programdata\elasticsearch\logs\*
  - /var/log/apache2/access.log*
  - /var/log/apache2/other_vhosts_access.log*
  - /var/log/httpd/access_log*
  - /var/log/apache2/error.log*
  - /var/log/httpd/error_log*
exclude_files: [".gz$"]

# Exclude lines. A list of regular expressions to match. It drops the lines that are
# matching any regular expression from the list.
#exclude_lines: ['^DBG']

# Include lines. A list of regular expressions to match. It exports the lines that are
# matching any regular expression from the list.
#include_lines: ['^ERR', '^WARN']
```

Figure 378: Installing and configuring filebeat for Apache web server.

The screenshot shows a terminal window with two tabs. The active tab is titled 'apchserver@apchserver: ~/Download...' and contains the configuration file for filebeat. The configuration file is as follows:

```
output.logstash:
  # The Logstash hosts
  hosts: ["10.10.30.3:5104"]

  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  #ssl.certificateAuthorities: ["/etc/pki/root/ca.pem"]

  # Certificate for SSL client authentication
  #ssl.certificate: "/etc/pki/client/cert.pem"

  # Client Certificate Key
  #ssl.key: "/etc/pki/client/cert.key"

  # ===== Processors =====
processors:
  - add_host_metadata:
      when.not.contains.tags: forwarded
  - add_cloud_metadata: ~
  - add_docker_metadata: ~
  - add_kubernetes_metadata: ~

  # ===== Logging =====
  # Sets log level. The default log level is info.
  # Available log levels are: error, warning, info, debug
  #logging.level: debug

  # At debug level, you can selectively enable logging only for some components.

  ^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos      M-U Undo
  ^X Exit        ^R Read File     ^V Replace       ^U Uncut Text    ^T To Spell      ^L Go To Line   M-B Redo
```

Figure 379: Installing and configuring filebeat for Apache web server.

The screenshot shows a terminal window titled "apchserver@apchserver: ~/Downloads". It contains two tabs: "apchserver@apchserver: ~/elastal..." and "apchserver@apchserver: ~/Downlo...". The content of the terminal is as follows:

```
apchserver@apchserver:~/Downloads$ sudo nano /etc/filebeat/filebeat.yml
[sudo] password for apchserver:
apchserver@apchserver:~/Downloads$ sudo systemctl enable filebeat
Synchronizing state of filebeat.service with sysv service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable filebeat
Created symlink /etc/systemd/system/multi-user.target.wants/filebeat.service → /lib/systemd/system/filebeat.service

apchserver@apchserver:~/Downloads$ systemctl start filebeat
apchserver@apchserver:~/Downloads$ apchserver@apchserver:~/Downloads$ sudo systemctl stauts flebeat
Unknown operation stauts.
apchserver@apchserver:~/Downloads$ sudo systemctl stauts flebeat
Unknown operation stauts.
apchserver@apchserver:~/Downloads$ 
apchserver@apchserver:~/Downloads$ sudo systemctl status flebeat
Unit flebeat.service could not be found.
apchserver@apchserver:~/Downloads$ sudo systemctl status filebeat.service
* filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
  Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2023-04-16 22:01:24 BST; 45s ago
    Docs: https://www.elastic.co/beats/filebeat
   Main PID: 4939 (filebeat)
     Tasks: 8 (limit: 1194)
    Memory: 123.5M
      CGroup: /system.slice/filebeat.service
              └─4939 /usr/share/filebeat/bin/filebeat --environment systemd -c /etc/filebeat/filebeat.yml --path.home

Apr 16 22:01:28 apchserver filebeat[4939]: 2023-04-16T22:01:28.018+0100      INFO      [input.harvester]
Apr 16 22:01:30 apchserver filebeat[4939]: 2023-04-16T22:01:30.975+0100      INFO      [add_cloud_metadata]
Apr 16 22:01:31 apchserver filebeat[4939]: 2023-04-16T22:01:31.017+0100      INFO      [publisher_pipeline_out]
Apr 16 22:01:31 apchserver filebeat[4939]: 2023-04-16T22:01:31.018+0100      INFO      [publisher]          pipel
Apr 16 22:01:31 apchserver filebeat[4939]: 2023-04-16T22:01:31.018+0100      INFO      [publisher]          pipel
Apr 16 22:01:58 apchserver filebeat[4939]: 2023-04-16T22:01:58.002+0100      INFO      [monitoring]        log/
Apr 16 22:02:02 apchserver filebeat[4939]: 2023-04-16T22:02:02.029+0100      ERROR     [publisher_pipeline_out]
```

Figure 380: Installing and configuring filebeat for Apache web server.

8.4.9 ElastAlert

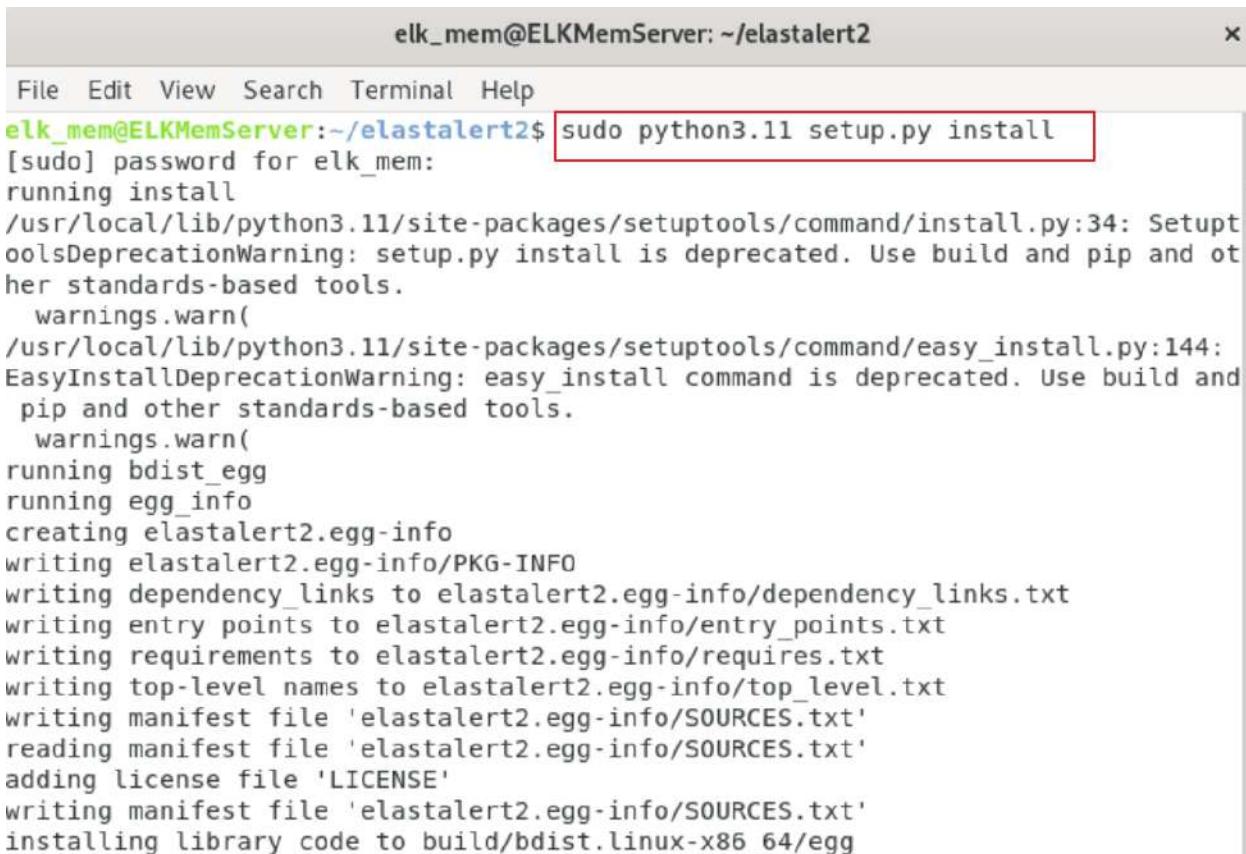
Installing Dependencies



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~/elastalert2". The terminal displays the following command-line session:

```
File Edit View Search Terminal Help
elk_mem@ELKMemServer:~/Python-3.11.15 sudo apt intall openssl
E: Invalid operation intall
elk_mem@ELKMemServer:~/Python-3.11.15 sudo apt install openssl
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssl is already the newest version (1.1.1n-0+deb10u4).
openssl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
N: Ignoring file 'elastic-7.17.9' in directory '/etc/apt/sources.list.d/' as it has an invalid filename extension
elk_mem@ELKMemServer:~/Python-3.11.15 cd ../
elk_mem@ELKMemServer:~$ git clone https://github.com/jertel/elastalert2.git
Cloning into 'elastalert2'...
remote: Enumerating objects: 16135, done.
remote: Counting objects: 100% (206/206), done.
remote: Compressing objects: 100% (113/113), done.
remote: Total 16135 (delta 105), reused 165 (delta 85), pack-reused 15929
Receiving objects: 100% (16135/16135), 5.23 MiB | 9.28 MiB/s, done.
Resolving deltas: 100% (11405/11405) done.
elk_mem@ELKMemServer:~$ cd elastalert2/
elk_mem@ELKMemServer:~/elastalert2$ sudo python3.11 -m pip install "setuptools>=11.3"#
Requirement already satisfied: setuptools >=11.3# in /usr/local/lib/python3.11/site-packages (65.5.0)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
elk_mem@ELKMemServer:~/elastalert2$
```

Figure 381: Installing dependencies.



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~/elastalert2". The window contains a command-line session where the user runs "sudo python3.11 setup.py install". The output of the command is displayed, showing various warning messages from the setup tools about deprecated commands like "setup.py install" and "easy_install", and the creation of files such as "PKG-INFO", "dependency_links.txt", and "SOURCES.txt". The terminal interface includes a menu bar with File, Edit, View, Search, Terminal, and Help, and a standard window control bar with minimize, maximize, and close buttons.

```
elk_mem@ELKMemServer:~/elastalert2$ sudo python3.11 setup.py install
[sudo] password for elk_mem:
running install
/usr/local/lib/python3.11/site-packages/setuptools/command/install.py:34: SetuptoolsDeprecationWarning: setup.py install is deprecated. Use build and pip and other standards-based tools.
  warnings.warn(
/usr/local/lib/python3.11/site-packages/setuptools/command/easy_install.py:144: EasyInstallDeprecationWarning: easy_install command is deprecated. Use build and pip and other standards-based tools.
  warnings.warn(
running bdist_egg
running egg_info
creating elastalert2.egg-info
writing elastalert2.egg-info/PKG-INFO
writing dependency_links to elastalert2.egg-info/dependency_links.txt
writing entry points to elastalert2.egg-info/entry_points.txt
writing requirements to elastalert2.egg-info/requirements.txt
writing top-level names to elastalert2.egg-info/top_level.txt
writing manifest file 'elastalert2.egg-info/SOURCES.txt'
reading manifest file 'elastalert2.egg-info/SOURCES.txt'
adding license file 'LICENSE'
writing manifest file 'elastalert2.egg-info/SOURCES.txt'
installing library code to build/bdist.linux-x86_64/egg
```

Figure 382: Installing dependencies.

```
elk_mem@ELKMemServer: ~/elastalert2
File Edit View Search Terminal Help
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behavi
our with the system package manager. It is recommended to use a virtual environment instead: ht
tps://pip.pypa.io/warnings/venv
elk_mem@ELKMemServer:~/elastalert2$ sudo python3.11 -m pip install "elasticsearch==7.17.9"
Collecting.elasticsearch==7.17.9
  Downloading.elasticsearch-7.17.9-py2.py3-none-any.whl (385 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 386.0/386.0 kB 198.4 kB/s eta 0:00:00
Requirement already satisfied: urllib3<2,>=1.21.1 in /usr/local/lib/python3.11/site-packages (f
rom.elasticsearch==7.17.9) (1.26.15)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/site-packages (from.elasticsearch==7.17.9) (2022.12.7)
Installing collected packages: elasticsearch
  Attempting uninstall: elasticsearch
    Found existing installation: elasticsearch 8.7.0
    Uninstalling.elasticsearch-8.7.0:
      Successfully uninstalled.elasticsearch-8.7.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are
       installed. This behaviour is the source of the following dependency conflicts.
elastalert2 2.10.1 requires.elasticsearch==7.10.1, but you have.elasticsearch 7.17.9 which is i
ncompatible.
Successfully installed.elasticsearch-7.17.9
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behavi
our with the system package manager. It is recommended to use a virtual environment instead: ht
tps://pip.pypa.io/warnings/venv
```

Figure 383: Installing dependencies.

```
elk_mem@ELKMemServer: ~/elastalert2
File Edit View Search Terminal Help
our with the system package manager. It is recommended to use a virtual environment instead: ht
tps://pip.pypa.io/warnings/venv
elk_mem@ELKMemServer:~/elastalert2$ elastalert-create-index
Enter Elasticsearch host: 10.10.30.3
Enter Elasticsearch port: 9200
Use SSL? t/f: f
Enter optional basic-auth username (or leave blank): elastic
Enter optional basic-auth password (or leave blank):
Enter optional Elasticsearch URL prefix (prepends a string to the URL of every request):
New index name? (Default elastalert_status)
Name of existing index to copy? (Default None)
Reading Elastic 7 index mappings:
Reading index mapping 'es_mappings/7/silence.json'
Reading index mapping 'es_mappings/7/elastalert_status.json'
Reading index mapping 'es_mappings/7/elastalert.json'
Reading index mapping 'es_mappings/7/past_elastalert.json'
Reading index mapping 'es_mappings/7/elastalert_error.json'
WARNING:py.warnings:/usr/local/lib/python3.11/site-packages/elasticsearch/connection/base.py:20
0: ElasticsearchWarning: Camel case format name dateOptionalTime is deprecated and will be remo
ved in a future version. Use snake case name date_optional_time instead.
    warnings.warn(message, category= ElasticsearchWarning)

WARNING:py.warnings:/usr/local/lib/python3.11/site-packages/elasticsearch/connection/base.py:20
0: ElasticsearchWarning: [types removal] Using include_type_name in put mapping requests is dep
```

Figure 384: creating index in Elasticsearch.

The screenshot shows the Elasticsearch Management interface under the 'Indices' tab. On the left sidebar, there are sections for Management, Ingest, Data (Index Management, Index Lifecycle Policies, Snapshot and Restore, Rollup Jobs, Transformations, Remote Clusters), Alerts and Insights (Rules and Connectors, Reporting, Machine Learning Jobs), Security (Users, Roles, API keys), and Kibana (Index Patterns). The main area displays a table of indices. The first row, 'elast alert index', is expanded, revealing three sub-indices: 'elastalert_status', 'elastalert_status_silence', and 'elastalert_status_error'. These three sub-indices are highlighted with a red border. Other visible indices include 'filebeat-7.17.9-2023.04.15', 'filebeat-7.17.9-2023.04.14', 'filebeat-7.17.9-2023.04.16', 'auditbeat-7.17.9-2023.04.16', and 'rsyslog-2023.04.16'. The table includes columns for Name, Health, Status, Primaries, Replicas, Docs count, Storage size, and Data stream. At the bottom, there are pagination controls showing page 1 of 2.

Name	Health	Status	Primaries	Replicas	Docs count	Storage size	Data stream
elast alert index	yellow	open	1	1	0	226b	
elastalert_status	yellow	open	1	1	0	226b	
elastalert_status_silence	yellow	open	1	1	0	226b	
filebeat-7.17.9-2023.04.15	yellow	open	1	1	11946	8.6mb	
filebeat-7.17.9-2023.04.14	yellow	open	1	1	842	2.7mb	
filebeat-7.17.9-2023.04.16	yellow	open	1	1	33341	29.3mb	
elastalert_status_error	yellow	open	1	1	0	226b	
elastalert_status_status	yellow	open	1	1	0	226b	
elastalert_status_past	yellow	open	1	1	0	226b	
auditbeat-7.17.9-2023.04.16	yellow	open	1	1	68	271.4kb	
rsyslog-2023.04.16	yellow	open	1	1	307	82.9kb	

Figure 385: Index created in Elasticsearch.

8.4.10. Telegram

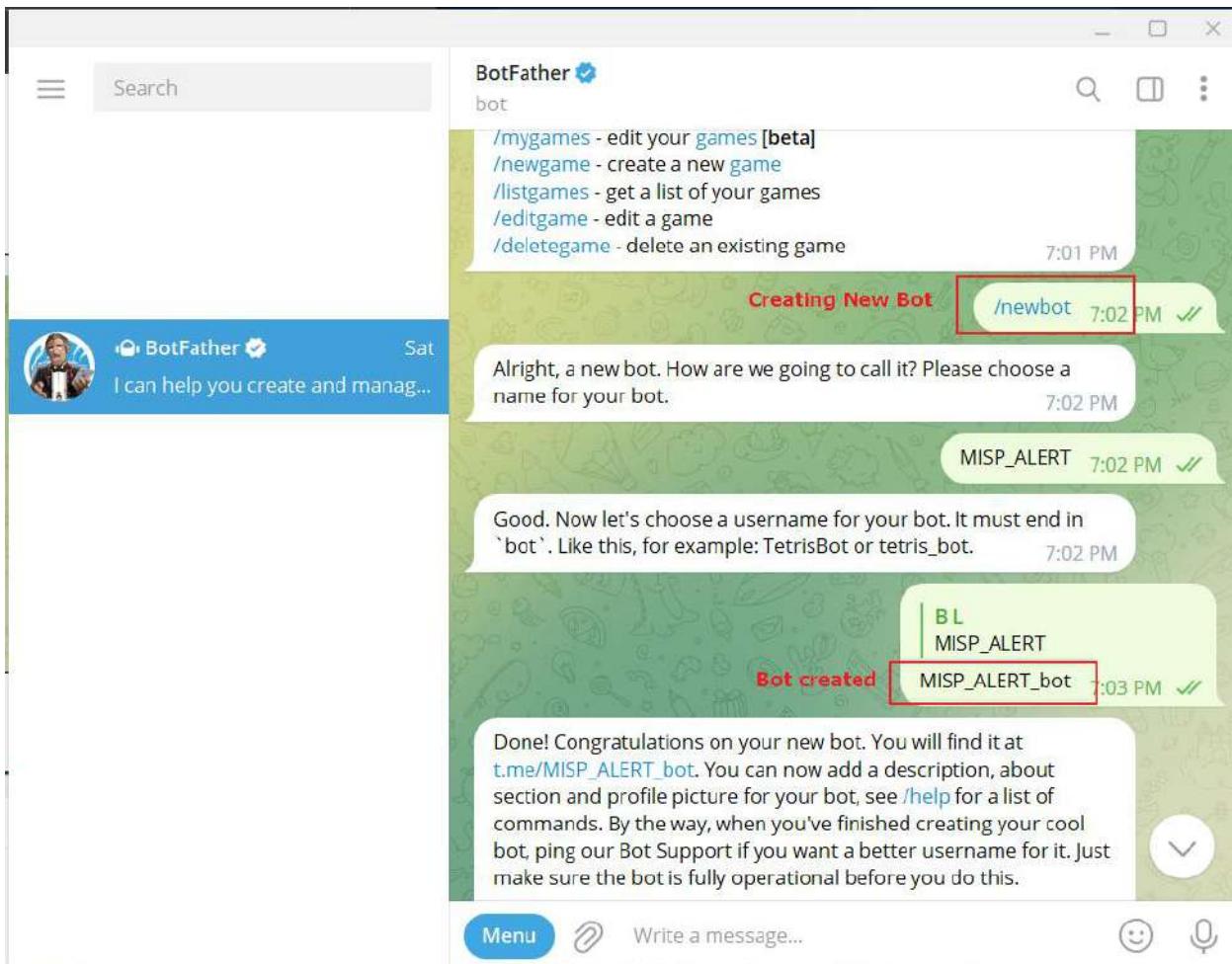


Figure 386: creating bot..

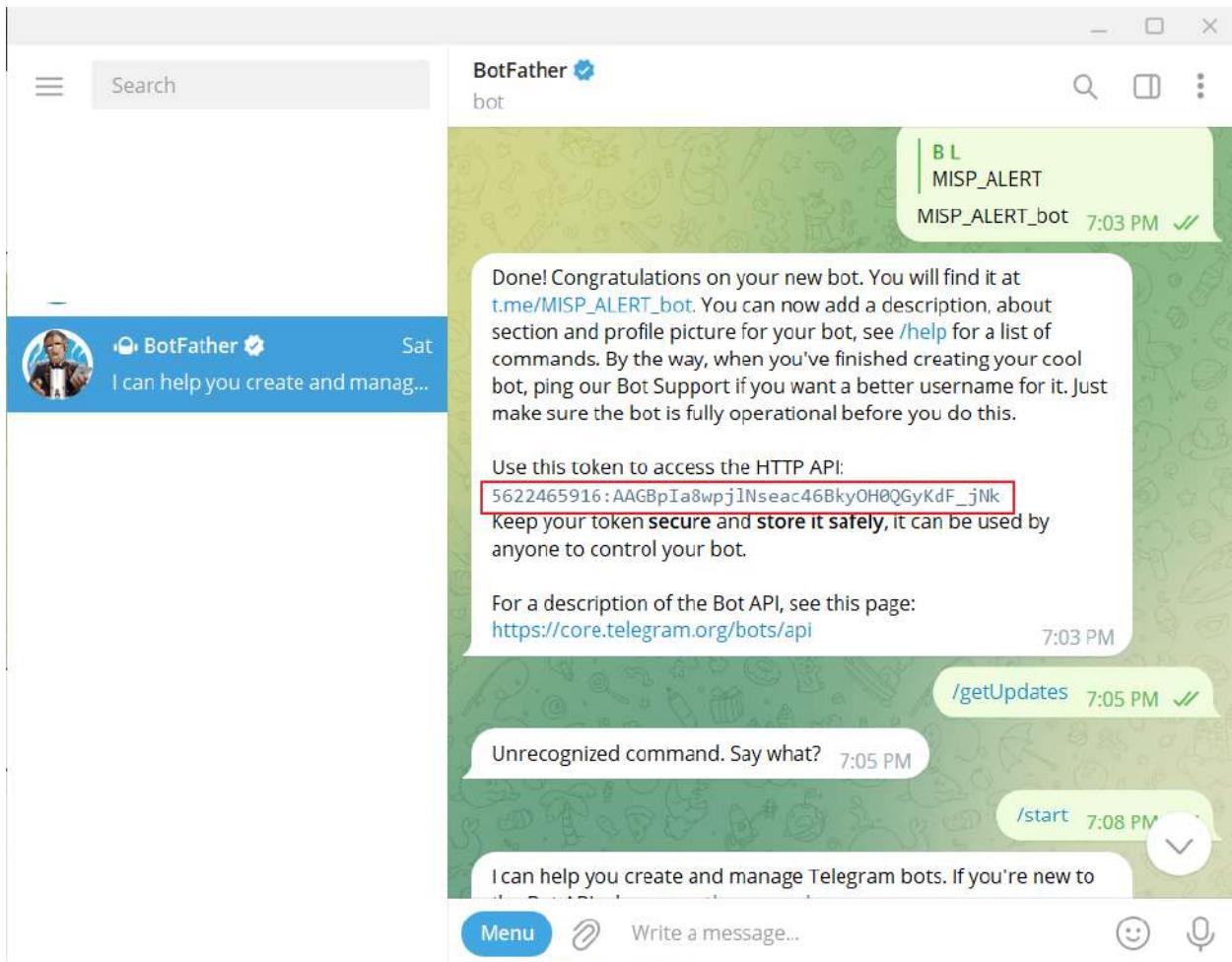


Figure 387: Generated token and API link. For telegram.

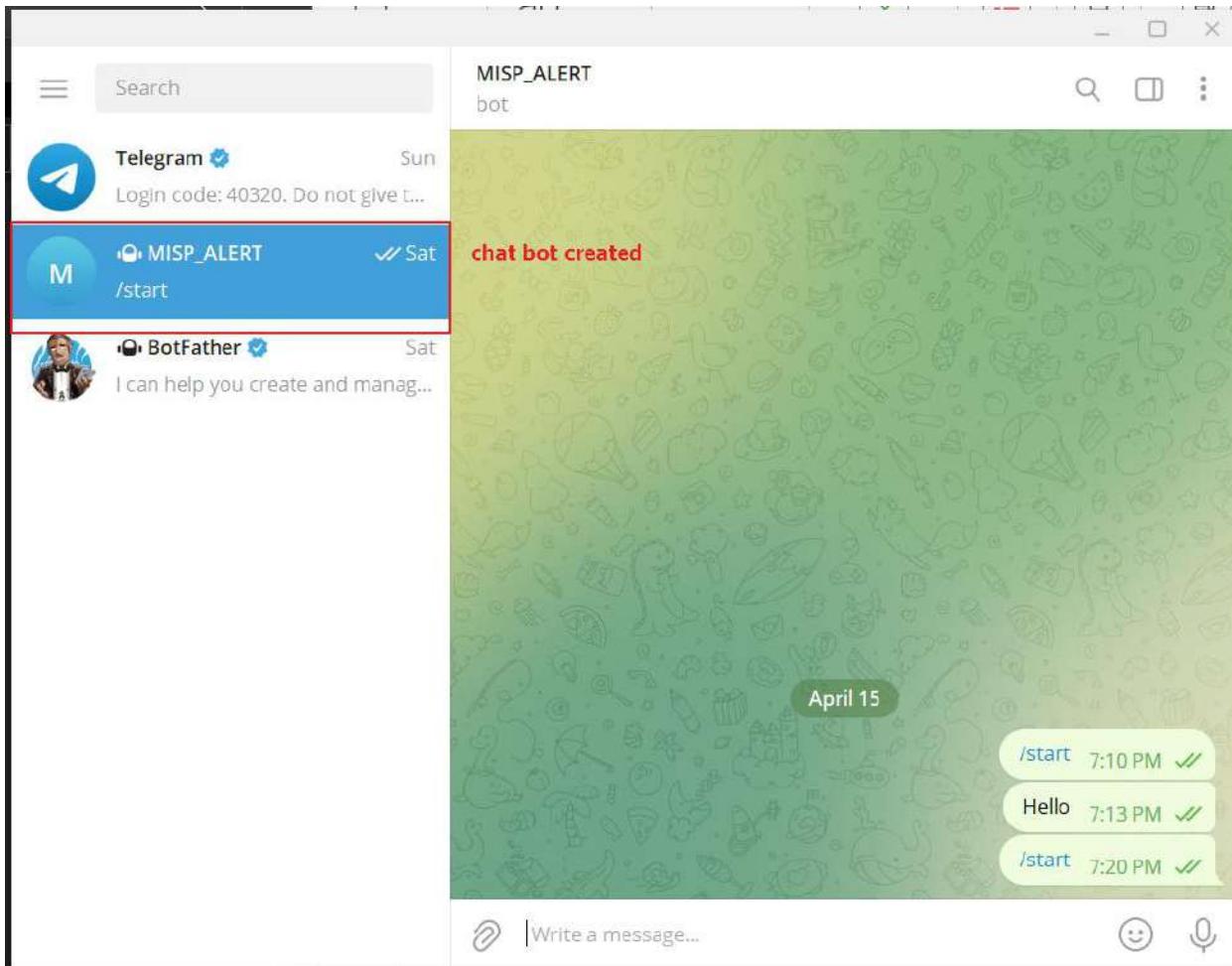


Figure 388: Chat room created.

```
example_frequency.yaml
~/elastalert2/examples/rules

#es_username: someusername
#es_password: somepassword

# (Required)
# Rule name, must be unique
name: Example frequency rule

# (Required)
# Type of alert.
# the frequency rule type alerts when num_events events occur with timeframe time
type: frequency

# (Required)
# Index to search, wildcard supported
index: filebeat-7.17.9-* filebeat-7.17.9-*

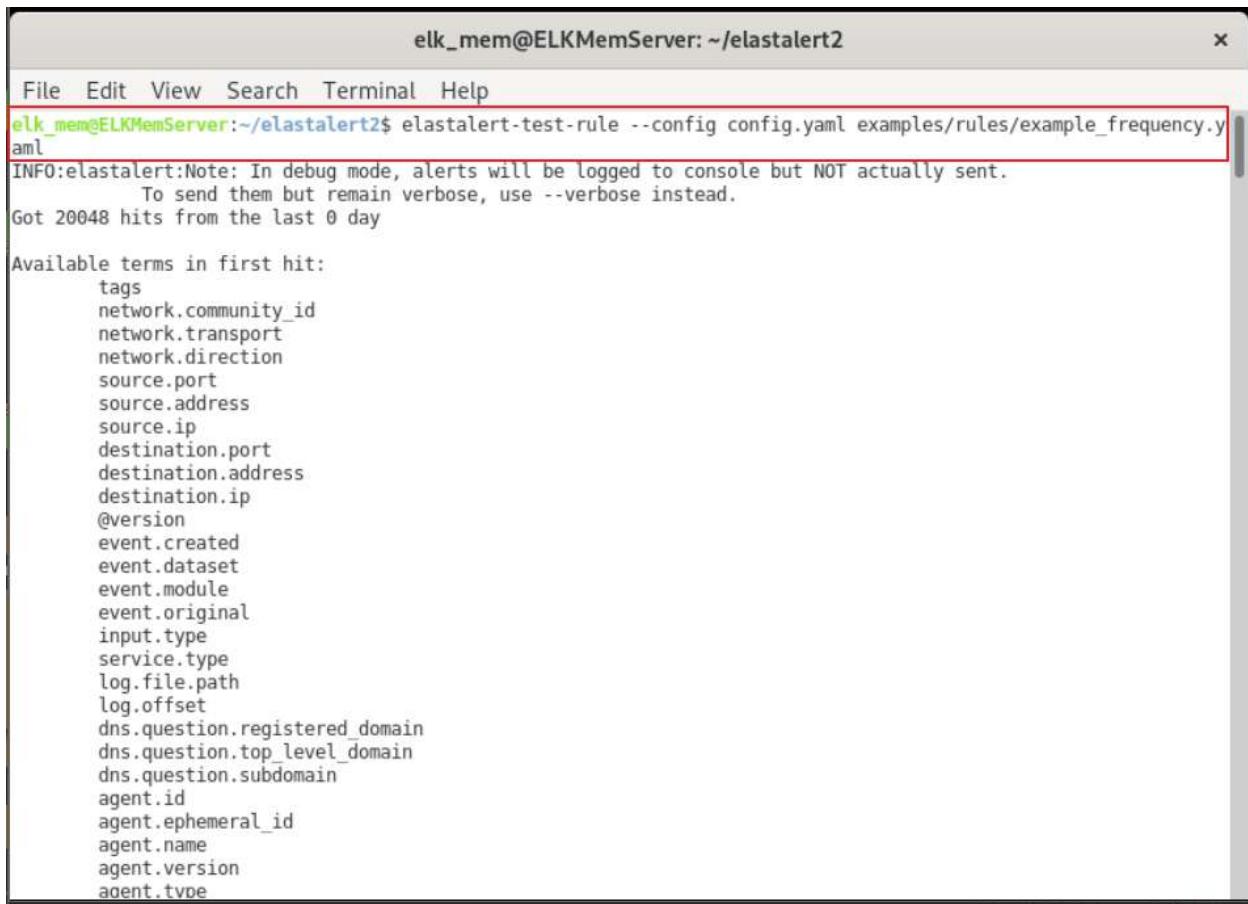
# (Required, frequency specific)
# Alert when this many documents matching the query occur within a timeframe
num_events: 2 2

# (Required, frequency specific)
# num events must occur within this amount of time to trigger an alert
timeframe:
  hours: 4

# (Required)
# A list of Elasticsearch filters used for find events
# These filters are joined with AND and nested in a filtered query
# For more info: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html
filter:
  - term:
    agent.hostname: "suricata" agent.hostname: "suricata" condition to be matched for alert

# The alert is use when a match is found
alert:
  - "telegram"
    telegram_bot_token: "5622465916:AAGBpIaBwpjLNseac46BkyOH0QGyKdF_jNk"
    telegram_room_id: "6267921323"
```

Figure 389: Configured yml file for alert generating in ElastAlert..

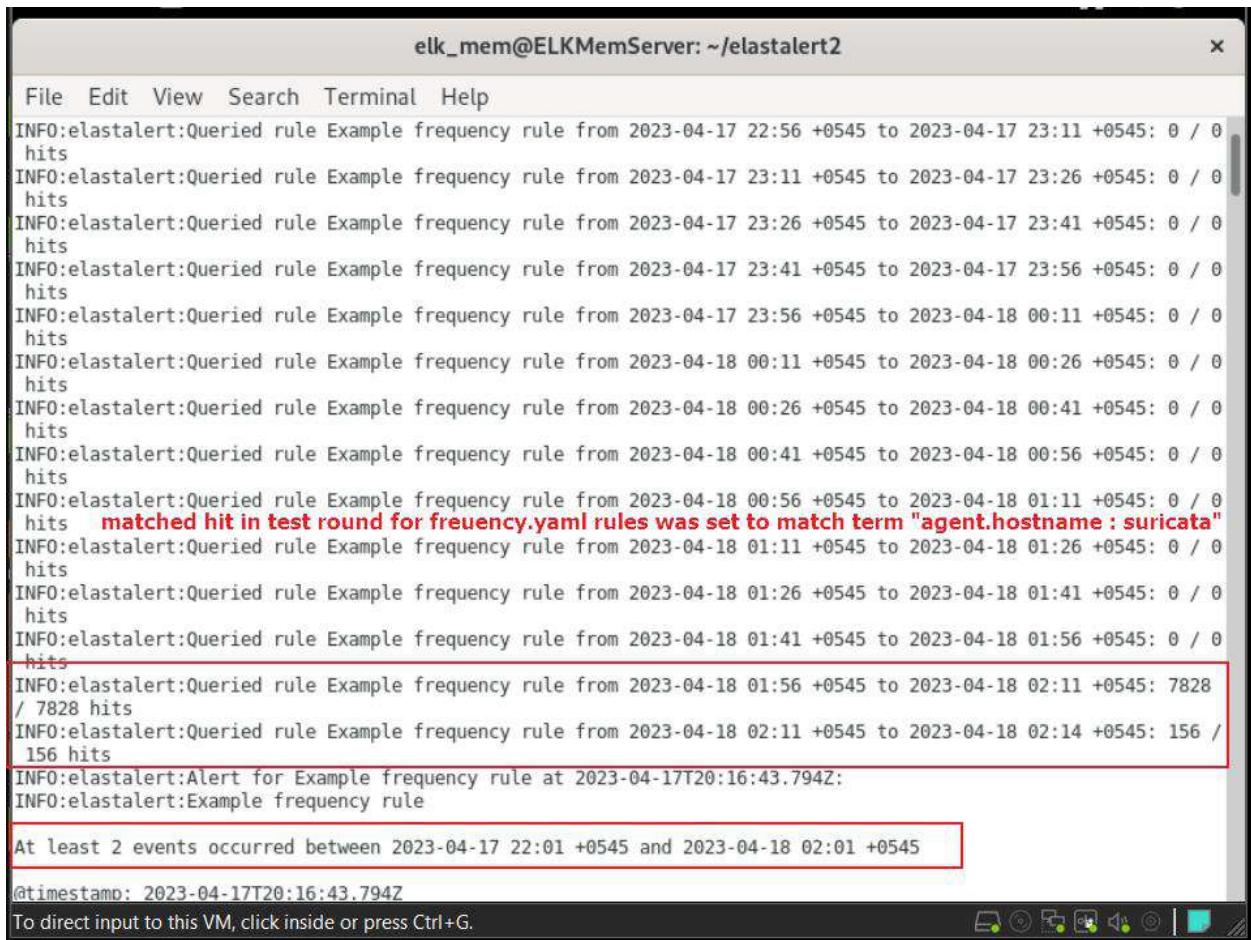


The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~/elastalert2". The window contains the following text:

```
File Edit View Search Terminal Help
elk_mem@ELKMemServer:~/elastalert2$ elastalert-test-rule --config config.yaml examples/rules/example_frequency.yaml
INFO:elastalert:Note: In debug mode, alerts will be logged to console but NOT actually sent.
      To send them but remain verbose, use --verbose instead.
Got 20048 hits from the last 0 day

Available terms in first hit:
tags
network.community_id
network.transport
network.direction
source.port
source.address
source.ip
destination.port
destination.address
destination.ip
@version
event.created
event.dataset
event.module
event.original
input.type
service.type
log.file.path
log.offset
dns.question.registered_domain
dns.question.top_level_domain
dns.question.subdomain
agent.id
agent.ephemeral_id
agent.name
agent.version
agent.type
```

Figure 390: Testing that yml file to generate alert.



```
elk_mem@ELKMemServer: ~/elastalert2
File Edit View Search Terminal Help
INFO:elastalert:Queried rule Example frequency rule from 2023-04-17 22:56 +0545 to 2023-04-17 23:11 +0545: 0 / 0
hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-17 23:11 +0545 to 2023-04-17 23:26 +0545: 0 / 0
hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-17 23:26 +0545 to 2023-04-17 23:41 +0545: 0 / 0
hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-17 23:41 +0545 to 2023-04-17 23:56 +0545: 0 / 0
hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-17 23:56 +0545 to 2023-04-18 00:11 +0545: 0 / 0
hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-18 00:11 +0545 to 2023-04-18 00:26 +0545: 0 / 0
hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-18 00:26 +0545 to 2023-04-18 00:41 +0545: 0 / 0
hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-18 00:41 +0545 to 2023-04-18 00:56 +0545: 0 / 0
hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-18 00:56 +0545 to 2023-04-18 01:11 +0545: 0 / 0
hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-18 01:11 +0545 to 2023-04-18 01:26 +0545: 0 / 0
hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-18 01:26 +0545 to 2023-04-18 01:41 +0545: 0 / 0
hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-18 01:41 +0545 to 2023-04-18 01:56 +0545: 0 / 0
hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-18 01:56 +0545 to 2023-04-18 02:11 +0545: 7828
/ 7828 hits
INFO:elastalert:Queried rule Example frequency rule from 2023-04-18 02:11 +0545 to 2023-04-18 02:14 +0545: 156 /
156 hits
INFO:elastalert:Alert for Example frequency rule at 2023-04-17T20:16:43.794Z:
INFO:elastalert:Example frequency rule

At least 2 events occurred between 2023-04-17 22:01 +0545 and 2023-04-18 02:01 +0545
@timestamp: 2023-04-17T20:16:43.794Z
To direct input to this VM, click inside or press Ctrl+G.
```

Figure 391: Matched hit in test round.

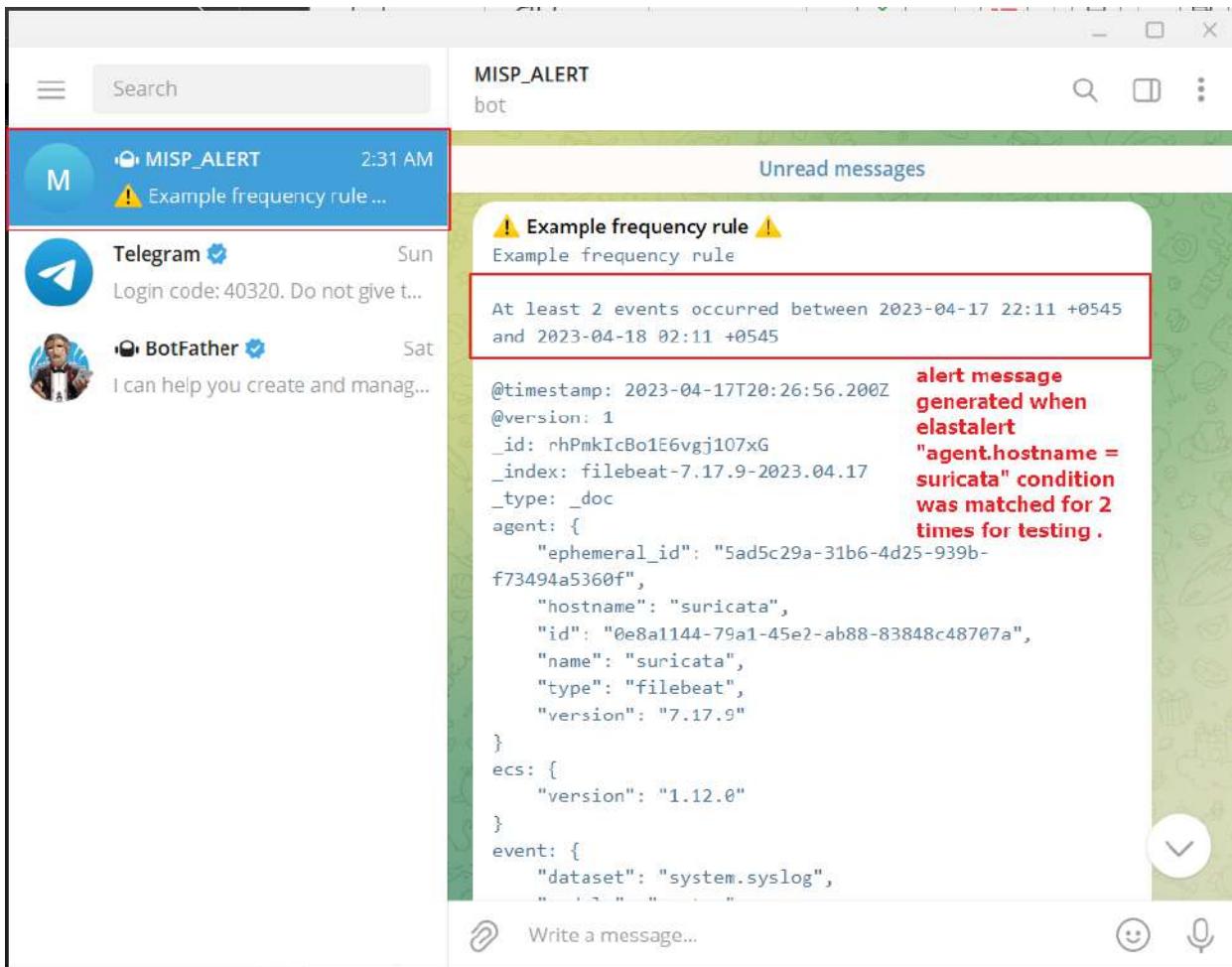
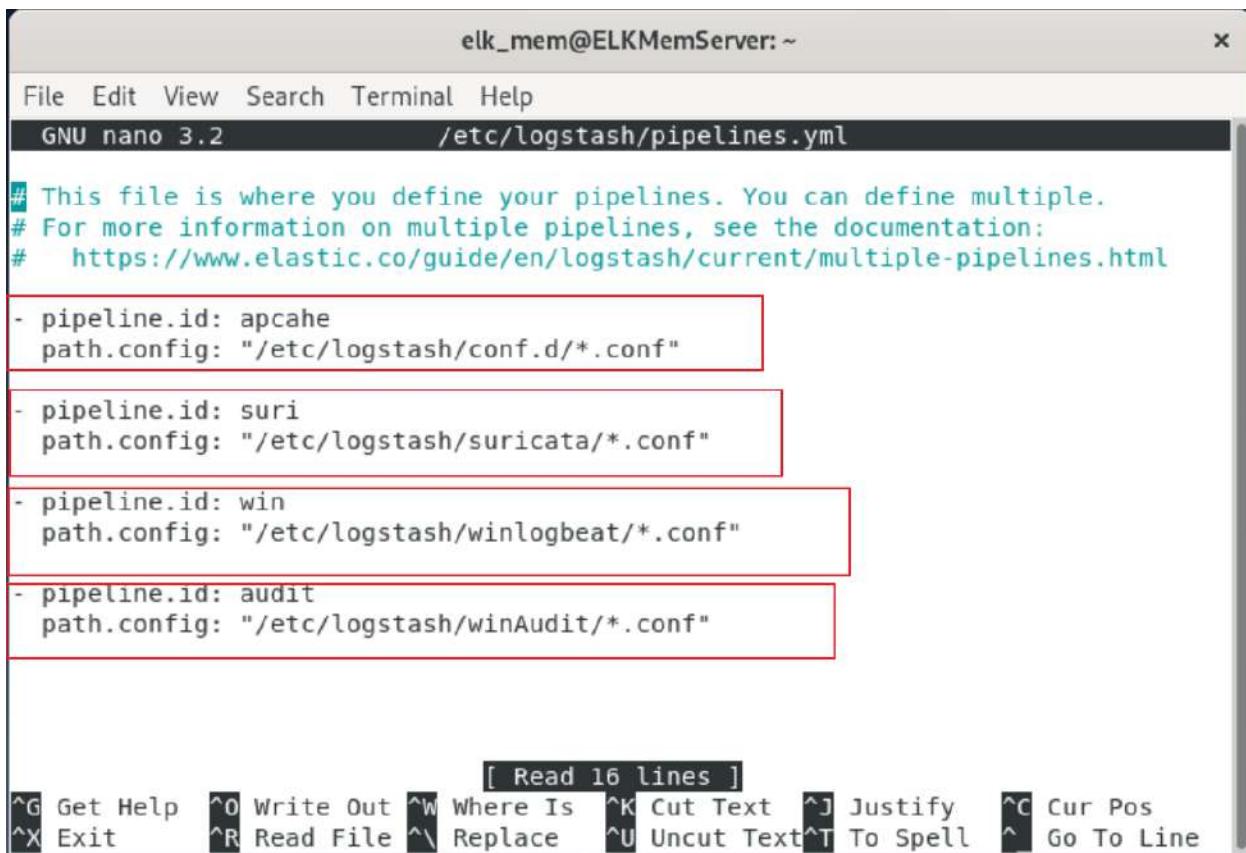


Figure 392: Log was sent to telegram.

8.4.11 Logstash config File



```
elk_mem@ELKMemServer: ~
File Edit View Search Terminal Help
GNU nano 3.2          /etc/logstash/pipelines.yml

# This file is where you define your pipelines. You can define multiple.
# For more information on multiple pipelines, see the documentation:
#   https://www.elastic.co/guide/en/logstash/current/multiple-pipelines.html

- pipeline.id: apcahe
  path.config: "/etc/logstash/conf.d/*.conf"

- pipeline.id: suri
  path.config: "/etc/logstash/suricata/*.conf"

- pipeline.id: win
  path.config: "/etc/logstash/winlogbeat/*.conf"

- pipeline.id: audit
  path.config: "/etc/logstash/winAudit/*.conf"

[ Read 16 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell ^L Go To Line
```

Figure 393: Logstash pipeline.

The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window contains a Logstash configuration file named "apache.conf" located at "/etc/logstash/conf.d/apache.conf". The configuration file uses the "nanomd" editor. The code defines an input section for Beats, a filter section for memcached entries from a specific IP, and a mutate section to remove the "enrich" field. The file ends with a closing brace for the main block. The terminal also displays a set of keyboard shortcuts at the bottom.

```
input{
    beats{
        port => 5042
    }
}

filter{
    if [dest_ip] {
        memcached {
            hosts => ["10.10.30.3:11211"]
            namespace => "misp-ip"
            get => [
                "%{[dest_ip]} => [enrich][tmp]"
            ]
        }
        if [enrich][tmp] {
            ruby {
                path => "/etc/logstash/process_ioc.rb"
            }
            mutate {
                remove_field => [ "[enrich]" ]
            }
        }
    }
    if [src_ip] {
        memcached {
            hosts => ["10.10.30.3:11211"]
            namespace => "misp-ip"
            get => [
                "%{[src_ip]} => [enrich][tmp]"
            ]
        }
        if [enrich][tmp] {
            ruby {
                path => "/etc/logstash/process_ioc.rb"
            }
        }
    }
}

^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos      M-U Undo
^X Exit         ^R Read File     ^\ Replace       ^U Uncut Text    ^T To Spell     ^_ Go To Line   M-E Redo
```

Figure 394: Logstash conf file for apache server 1.

The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The menu bar includes File, Edit, View, Search, Terminal, and Help. The title bar indicates the file is "/etc/logstash/conf.d/apache.conf" and the version is "GNU nano 3.2". The main content of the terminal is the Logstash configuration file:

```
remove_field => [ "[enrich]" ]
}
}
if [hash][sha256] {
memcached {
hosts => ["10.10.30.3:11211"]
namespace => "misp-sha256"
get => {
"%{[hash][sha256]}" => "[enrich][tmp]"
}
}
if [enrich][tmp] {
ruby {
path => "/etc/logstash/process_ioc.rb"
}
mutate {
remove_field => [ "[enrich]" ]
}
}
}
}
output{
elasticsearch{
hosts => ["10.10.30.3:9200"]
user => "elastic"
password => "root123"
index => "apachelog-%{+YYYY.MM.dd}"
}
}
```

Figure 395: Logstash conf file for apache server 2.

The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window title bar includes a close button ("x"). The menu bar contains "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the bottom shows "GNU nano 3.2", the file path "/etc/logstash/suricata/test6.conf", and "Modified". The main content area displays a Logstash configuration file:

```
input{
  beats{
    port => 5044
  }
}

filter {
  json{
    source => "message"
  }
  if [dest_ip] {
    memcached {
      hosts => ["10.10.30.3:11211"]
      namespace => "misp-ip"
      get => {
        "%{[dest_ip]}" => "[enrich][tmp]"
      }
    }
    if [enrich][tmp] {
      ruby {
        path => "/etc/logstash/process_ioc.rb"
      }
      mutate {
        remove_field => [ "[enrich]" ]
      }
    }
  }
  if [src_ip] {
    memcached {
      hosts => ["10.10.30.3:11211"]
      namespace => "misp-ip"
      get => {
        "%{[src_ip]}" => "[enrich][tmp]"
      }
    }
  }
}

^G Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos   M-U Undo
^X Exit       ^R Read File   ^\ Replace    ^U Uncut Text  ^T To Spell  ^_ Go To Line M-E Redo
```

Figure 396: Logstash conf file for Suricata VM.

The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window contains a nano editor session for a Logstash configuration file named "/etc/logstash/suricata/test6.conf". The file is marked as "Modified". The configuration code is as follows:

```
File Edit View Search Terminal Help
GNU nano 3.2 /etc/logstash/suricata/test6.conf Modified

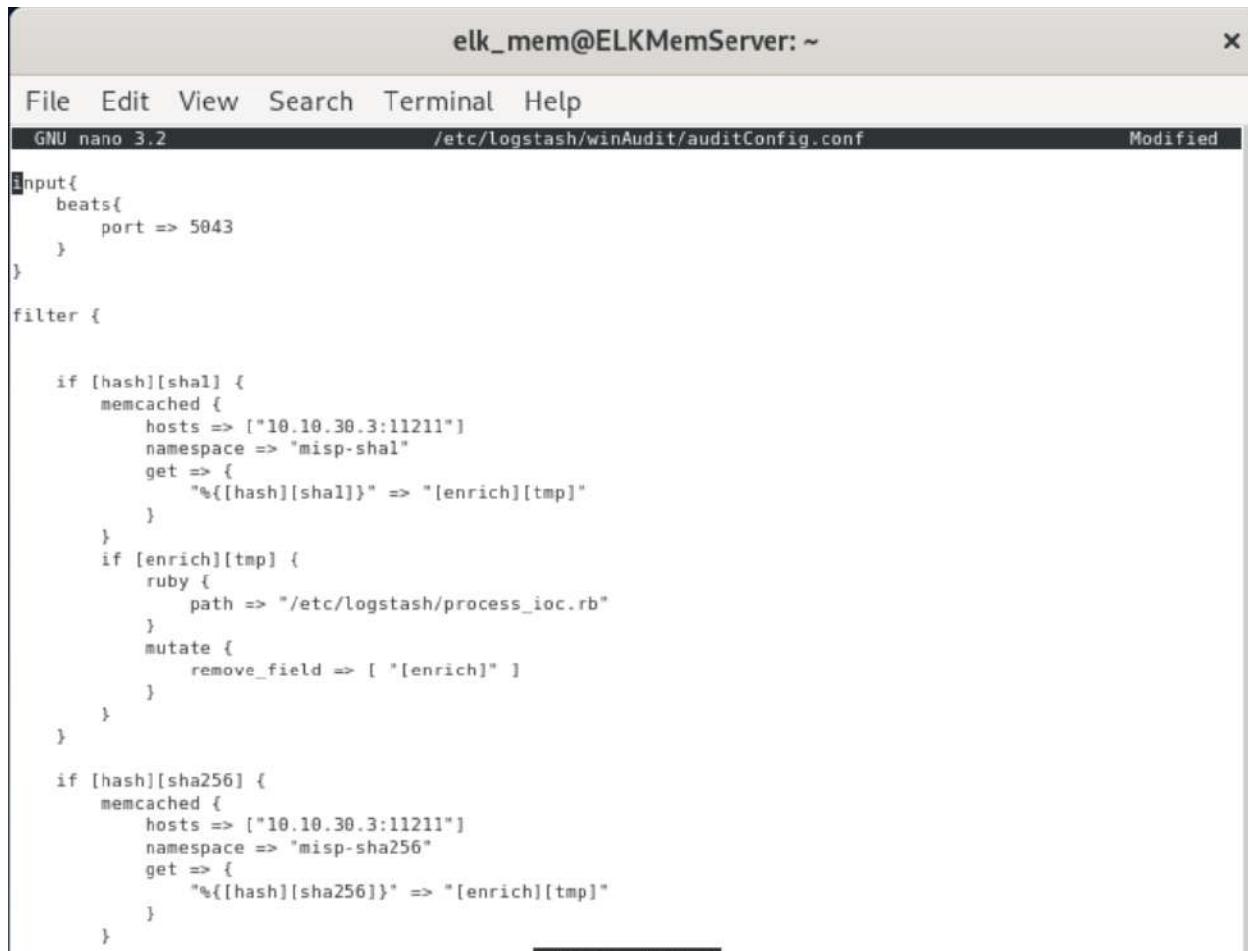
}

if [enrich][tmp] {
    ruby {
        path => "/etc/logstash/process_ioc.rb"
    }
    mutate {
        remove_field => [ "[enrich]" ]
    }
}

}

output{
    #stdout { codec => rubydebug }
    elasticsearch{
        hosts => ["10.10.30.3:9200"]
        user => "elastic"
        password => "root123"
        index => "filebeat-7.17.9-%{+yyyy.MM.dd}"
    }
}
```

Figure 397: Logstash conf file for Suricata VM.



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window contains a nano text editor displaying a Logstash configuration file. The file starts with an input section for beats on port 5043, followed by a filter section. The filter section includes logic for SHA-1 and SHA-256 hashes using memcached and ruby code to enrich and remove fields.

```
input{
  beats{
    port => 5043
  }
}

filter {

  if [hash][shal] {
    memcached {
      hosts => ["10.10.30.3:11211"]
      namespace => "misp-shal"
      get => {
        "%{[hash][shal]}" => "[enrich][tmp]"
      }
    }
    if [enrich][tmp] {
      ruby {
        path => "/etc/logstash/process_ioc.rb"
      }
      mutate {
        remove_field => [ "[enrich]" ]
      }
    }
  }

  if [hash][sha256] {
    memcached {
      hosts => ["10.10.30.3:11211"]
      namespace => "misp-sha256"
      get => {
        "%{[hash][sha256]}" => "[enrich][tmp]"
      }
    }
  }
}
```

Figure 398: Logstash conf file for windows 10 VM(auditbeat).



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window title bar also displays the file path "/etc/logstash/winAudit/auditConfig.conf" and the status "Modified". The terminal menu bar includes File, Edit, View, Search, Terminal, and Help. The main content area of the terminal shows the Logstash configuration file:

```
GNU nano 3.2 /etc/logstash/winAudit/auditConfig.conf Modified

ruby {
    path => "/etc/logstash/process_ioc.rb"
}
mutate {
    remove_field => [ "[enrich]" ]
}
}

output{
    #stdout { codec => rubydebug }
    elasticsearch{
        hosts => ["10.10.30.3:9200"]
        user => "elastic"
        password => "root123"
        index => "auditbeat-7.17.9-%{+yyyy.MM.dd}"
    }
}
```

Figure 399: Logstash conf file for windows 10 VM(auditbeat).

```
elk_mem@ELKMemServer: ~
File Edit View Search Terminal Help
GNU nano 3.2 /etc/logstash/winlogbeat/test7.conf

input{
  beats{
    port => 5045
  }
}

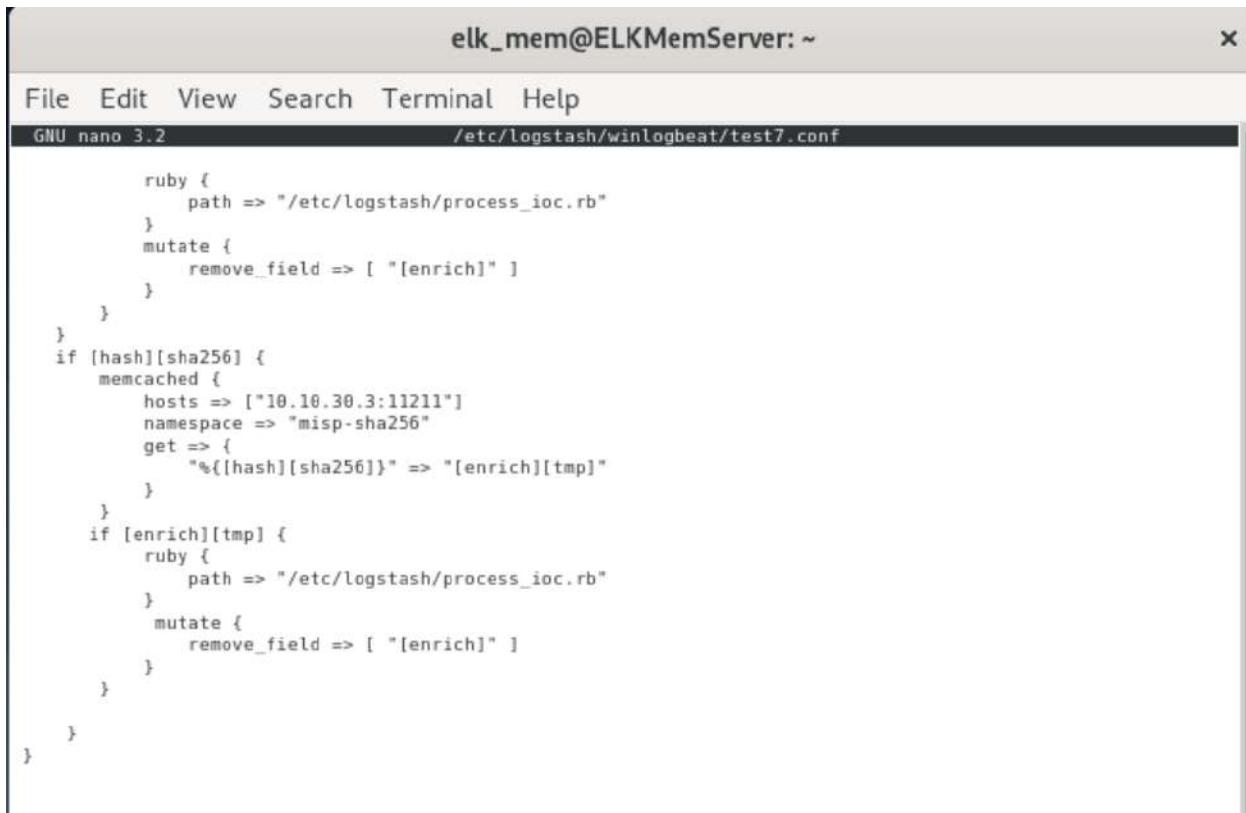
filter {

  if [destination][ip] {
    memcached {
      hosts => ["10.10.30.3:11211"]
      namespace => "misp-ip"
      get => {
        "%{[destination][ip]}" => "[enrich][tmp]"
      }
    }
    if [enrich][tmp] {
      ruby {
        path => "/etc/logstash/process_ioc.rb"
      }
      mutate {
        remove_field => [ "[enrich]" ]
      }
    }
  }

  if [source][ip] {
    memcached {
      hosts => ["10.10.30.3:11211"]
      namespace => "misp-ip"
      get => {
        "%{[winlog][event_data][SourceIp]}" => "[enrich][tmp]"
      }
    }
  }
}

output{
  elasticsearch{
    hosts => ["http://10.10.30.3:9200"]
    index => "winlogbeat-%{+YYYY.MM.dd}"
  }
}
```

Figure 400: Logstash conf file for windows 10 VM(winlogbeat).

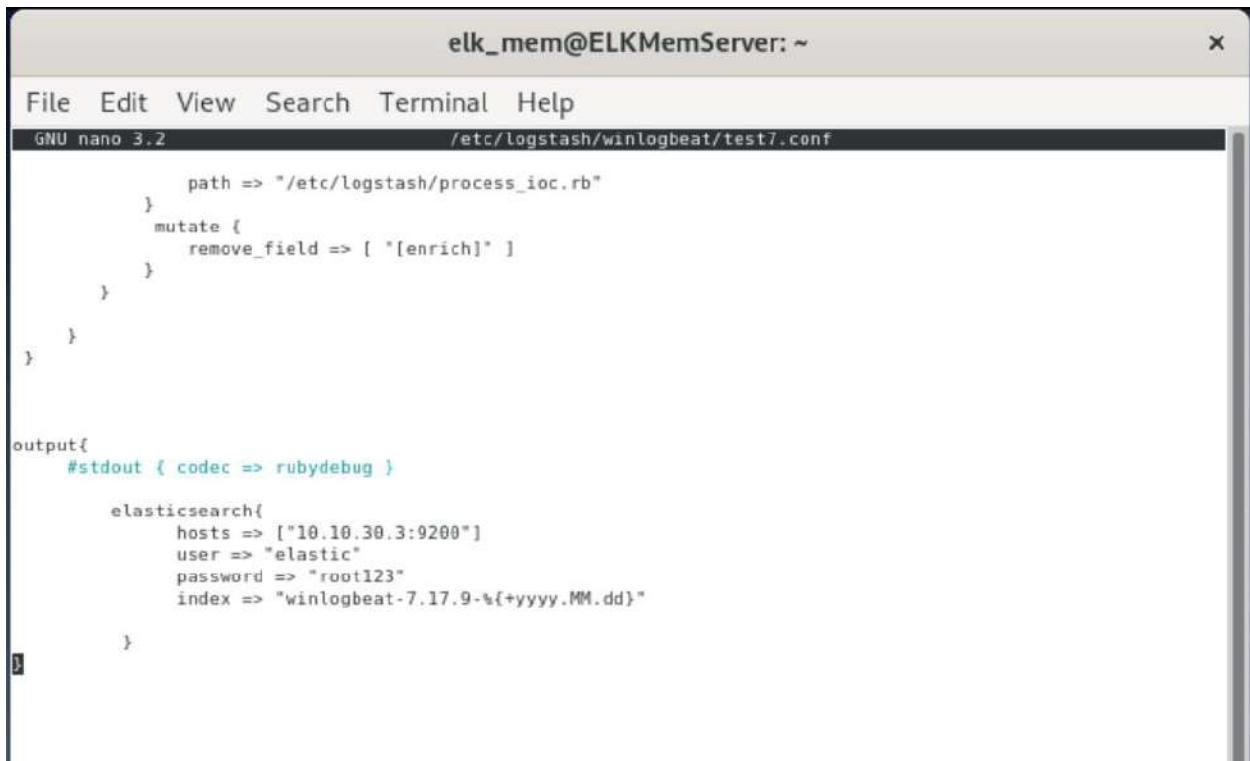


The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window contains a nano text editor displaying a Logstash configuration file named "test7.conf". The file uses the Ruby processor to read from "/etc/logstash/process_ioc.rb", mutate the data by removing the "[enrich]" field, and then checks if there is a hash entry with sha256. If found, it uses memcached to get the enriched data from a host at "10.10.30.3:11211" and adds it back to the log. Finally, it checks if the enriched data has a tmp entry and processes it similarly.

```
ruby {
    path => "/etc/logstash/process_ioc.rb"
}
mutate {
    remove_field => [ "[enrich]" ]
}
}

if [hash][sha256] {
    memcached {
        hosts => ["10.10.30.3:11211"]
        namespace => "misp-sha256"
        get => {
            "%{[hash][sha256]}" => "[enrich][tmp]"
        }
    }
    if [enrich][tmp] {
        ruby {
            path => "/etc/logstash/process_ioc.rb"
        }
        mutate {
            remove_field => [ "[enrich]" ]
        }
    }
}
```

Figure 401: Logstash conf file for windows 10 VM(winlogbeat).



The screenshot shows a terminal window titled "elk_mem@ELKMemServer: ~". The window contains a nano text editor displaying a Logstash configuration file named "test7.conf". The file content is as follows:

```
GNU nano 3.2 /etc/logstash/winlogbeat/test7.conf

    path => "/etc/logstash/process_ioc.rb"
}
mutate {
    remove_field => [ "[enrich]" ]
}
}

output{
    #stdout { codec => rubydebug }

    elasticsearch{
        hosts => ["10.10.30.3:9200"]
        user => "elastic"
        password => "root123"
        index => "winlogbeat-7.17.9-{+yyyy.MM.dd}"
    }
}
```

Figure 402: Logstash conf file for windows 10 VM(winlogbeat).

8.4.12 Configuration for Mikrotik router and PfSense

Mikrotik router

```
[admin@RouterOS] >
[admin@RouterOS] >
[admin@RouterOS] >
[admin@RouterOS] >
[admin@RouterOS] >
[admin@RouterOS] >
[admin@RouterOS] > /ip dhcp-client add interface=ether1 disabled=no 1
Failure: dhcp-client on that interface already exists
[admin@RouterOS] >
[admin@RouterOS] >
[admin@RouterOS] > interface ether1 2
Flags: D - dynamic, X - disabled, R - running, S - slave
#  NAME          TYPE      ACTUAL-MTU L2-MTU   MAX-L2-MTU MAC-ADDRESS
0  R  ether1      ether     1500      0C:E2:46:BF:00:00
1  ether2        ether     1500      0C:E2:46:BF:00:01
2  ether3        ether     1500      0C:E2:46:BF:00:02
3  ether4        ether     1500      0C:E2:46:BF:00:03
4  ether5        ether     1500      0C:E2:46:BF:00:04
5  ether6        ether     1500      0C:E2:46:BF:00:05
6  ether7        ether     1500      0C:E2:46:BF:00:06
7  ether8        ether     1500      0C:E2:46:BF:00:07
[admin@RouterOS] > ip address print 3
Flags: X - disabled, I - invalid, D - dynamic
#  ADDRESS          NETWORK           INTERFACE
0  0.0.0.0/0         0.0.0.0/0       ether1
[admin@RouterOS] >
[admin@RouterOS] > ip address add address=10.10.10.1/29 interface=ether2 4
[admin@RouterOS] >
[admin@RouterOS] > ip dhcp-server setup 5
Select interface to run DHCP server on
dhcp server interface: ether2
select network for DHCP addresses
dhcp address space: 10.10.10.0/29
Select gateway for given network
gateway for dhcp network: 10.10.10.1
Select pool of IP addresses given out by DHCP server
addresses to give out: 10.10.10.3-10.10.10.6
Select DNS servers
dns servers: 8.8.8.8
Select lease time
lease time: 60d
[admin@RouterOS] > 6
List of available interfaces
where "R" represent
running interface.

IP address received from cloud for internet connection.

Setting up dhcp server in ether2 interface.
```

Figure 403: Configuring Mikrotik router.

```
[admin@RouterOS] >
[admin@RouterOS] > /ip dhcp-server print 7
Successfully dhcp server setup.
Flags: X - disabled, X - invalid, I - invalid
#  NAME          INTERFACE          RELAY          ADDRESS-POOL          LEASE-TIME          ADD-ARP
0  dhcp1         ether2            dhcp_pool1          Bw4d
[admin@RouterOS] >
[admin@RouterOS] > ip address print 8
IP addresses of running interfaces.
Flags: X - disabled, I - invalid, D - dynamic
#  ADDRESS          NETWORK           INTERFACE
0  192.168.239.1/24 192.168.239.0  ether1
1  10.10.10.0/29    10.10.10.0    ether2
[admin@RouterOS] >
[admin@RouterOS] > ping 8.8.8.8 9
SEQ HOST          SIZE TTL TIME STATUS
0 8.8.8.8          56 128 56ms
1 8.8.8.8          56 128 55ms
2 8.8.8.8          56 128 55ms
sent=3 received=3 packet-loss=0% min-ttl=55ms avg-ttl=55ms max-ttl=55ms
[admin@RouterOS] > /ip firewall nat add chain=srcnat action=masquerade out-interface=ether1 10
[admin@RouterOS] >
[admin@RouterOS] >
[admin@RouterOS] > ip firewall nat print 11
Flags: X - disabled, I - invalid, D - dynamic
0  chain=srcnat action=masquerade out-interface=ether1
Successfully setup source nat for the all internal IP address to access outside
```

Figure 404: Configuring Mikrotik router.

PfSense

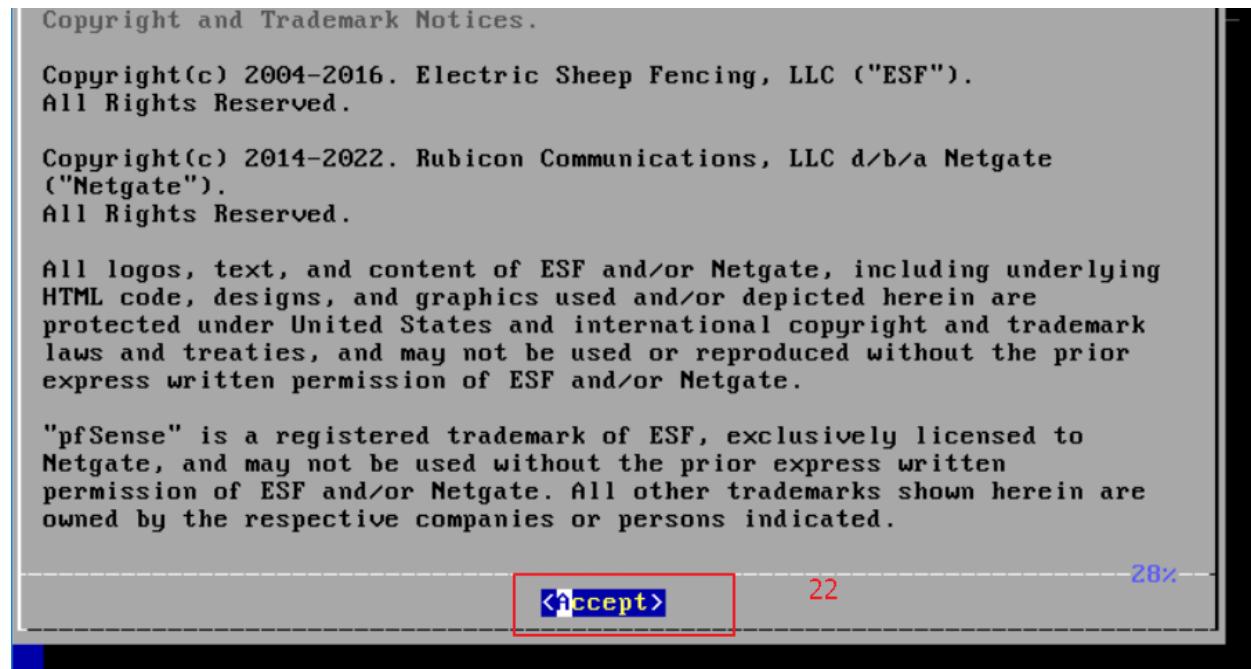


Figure 405: Configuring PfSense.

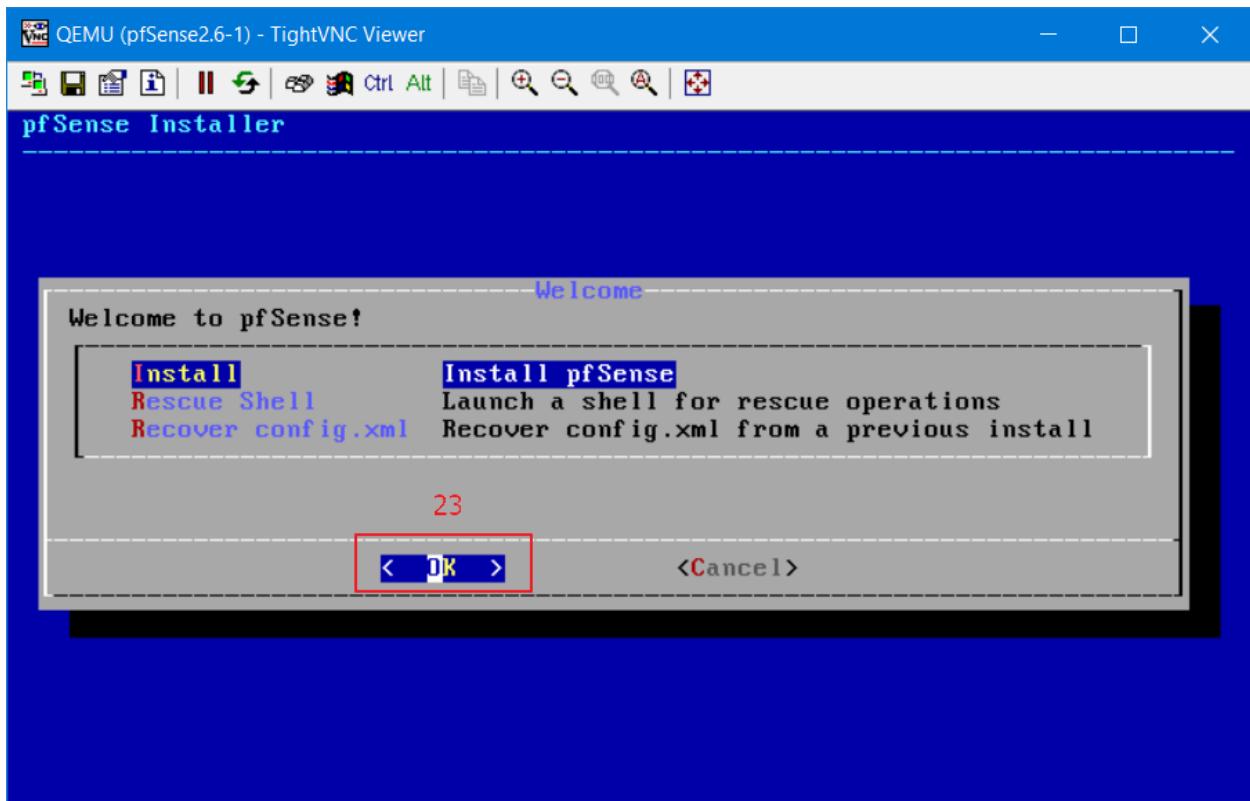


Figure 406: Configuring Pfsense.

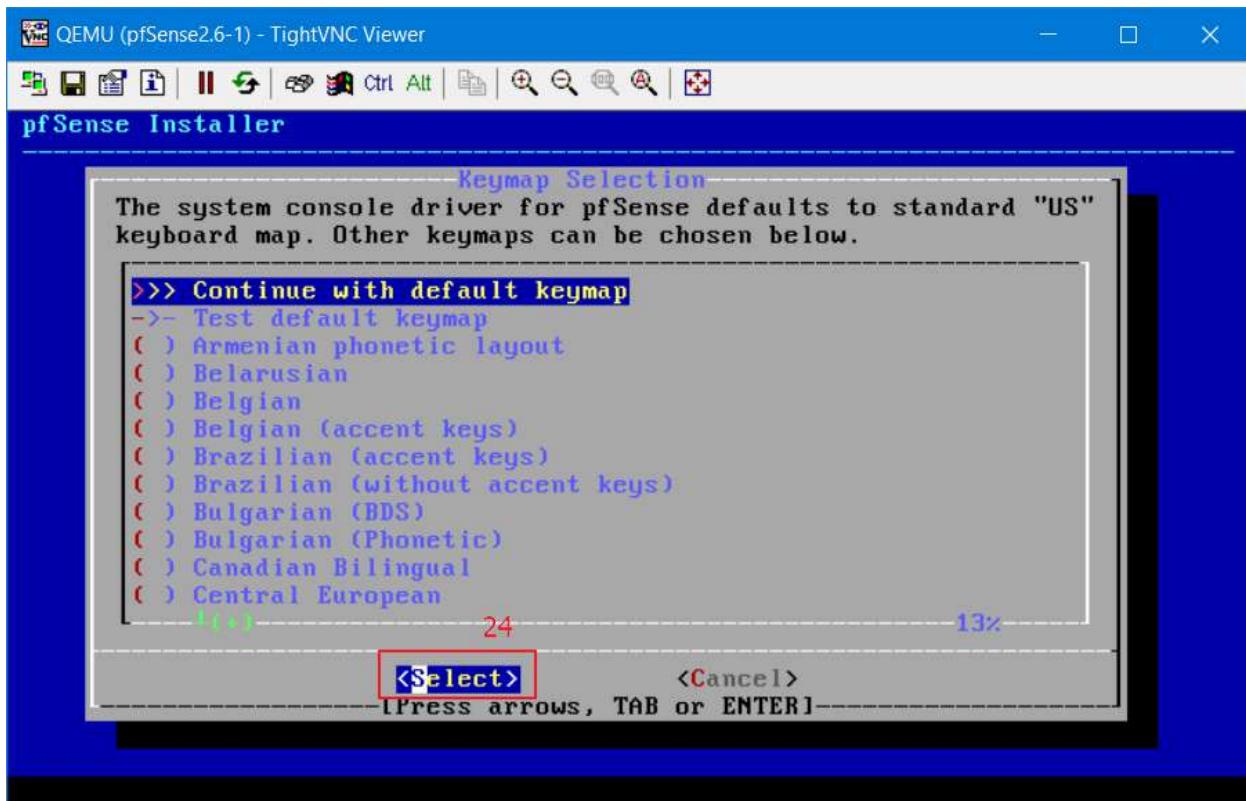


Figure 407: Configuring Pfsense.

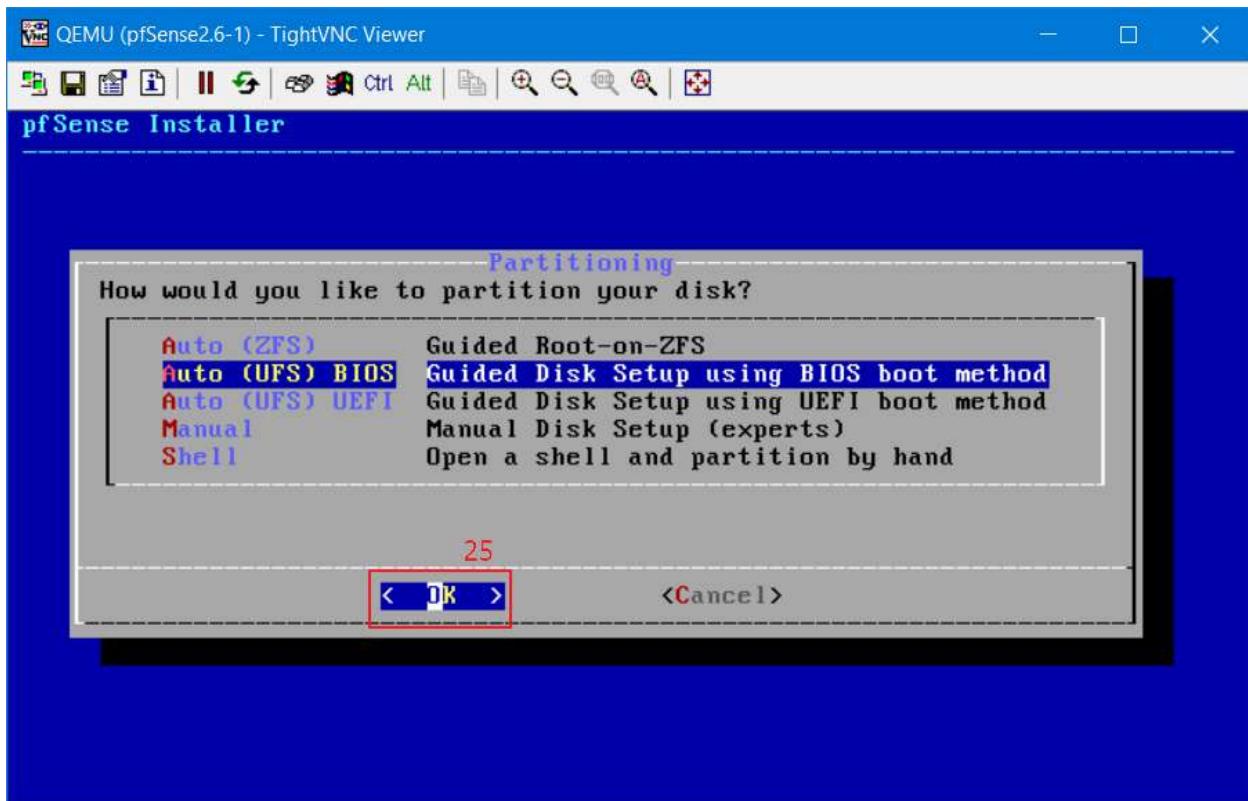


Figure 408: Configuring Pfsense.

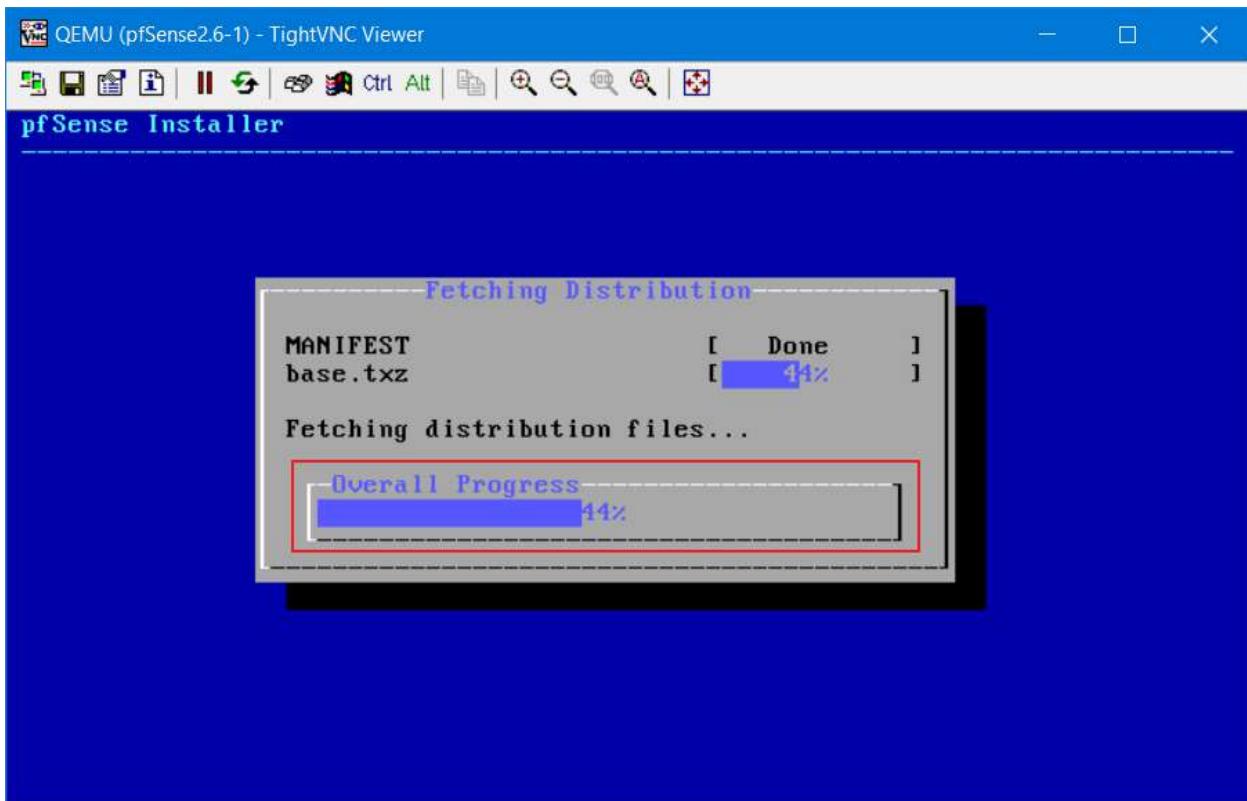


Figure 409: Configuring Pfsense.

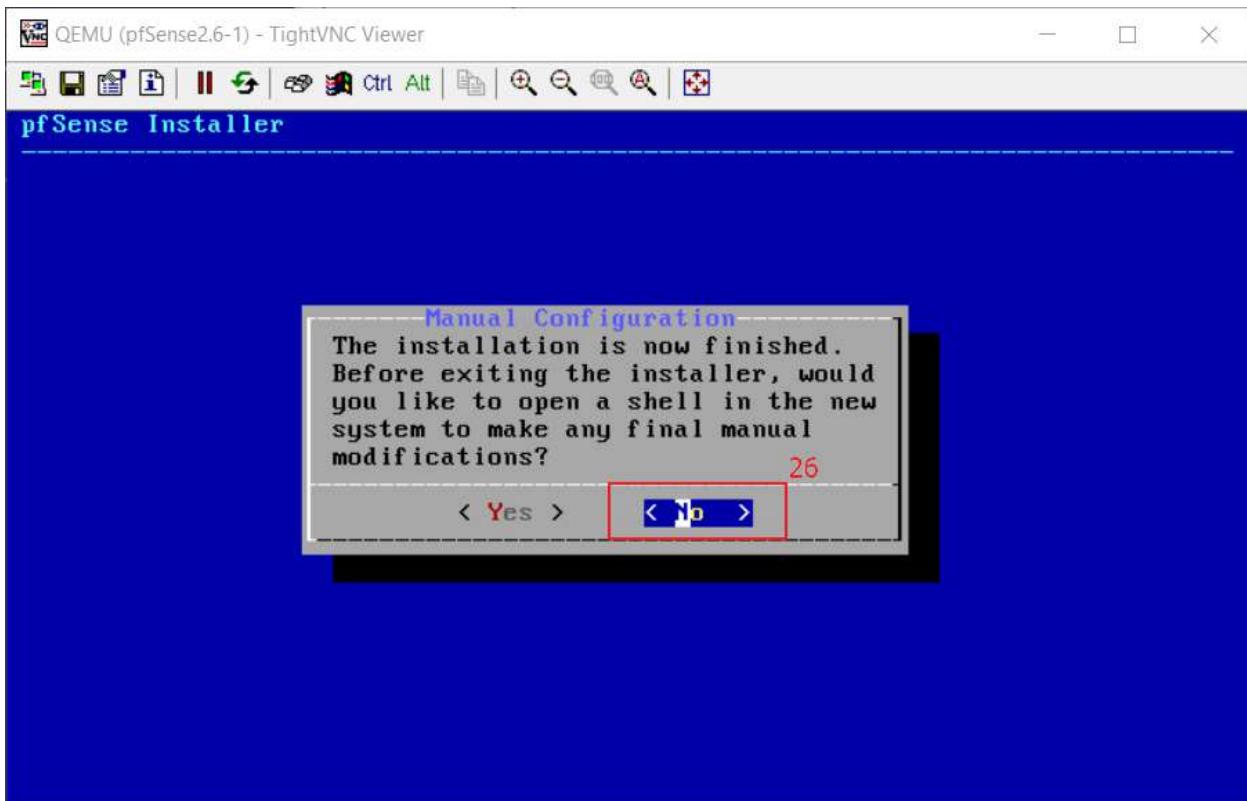


Figure 410:Configuring PfSense.

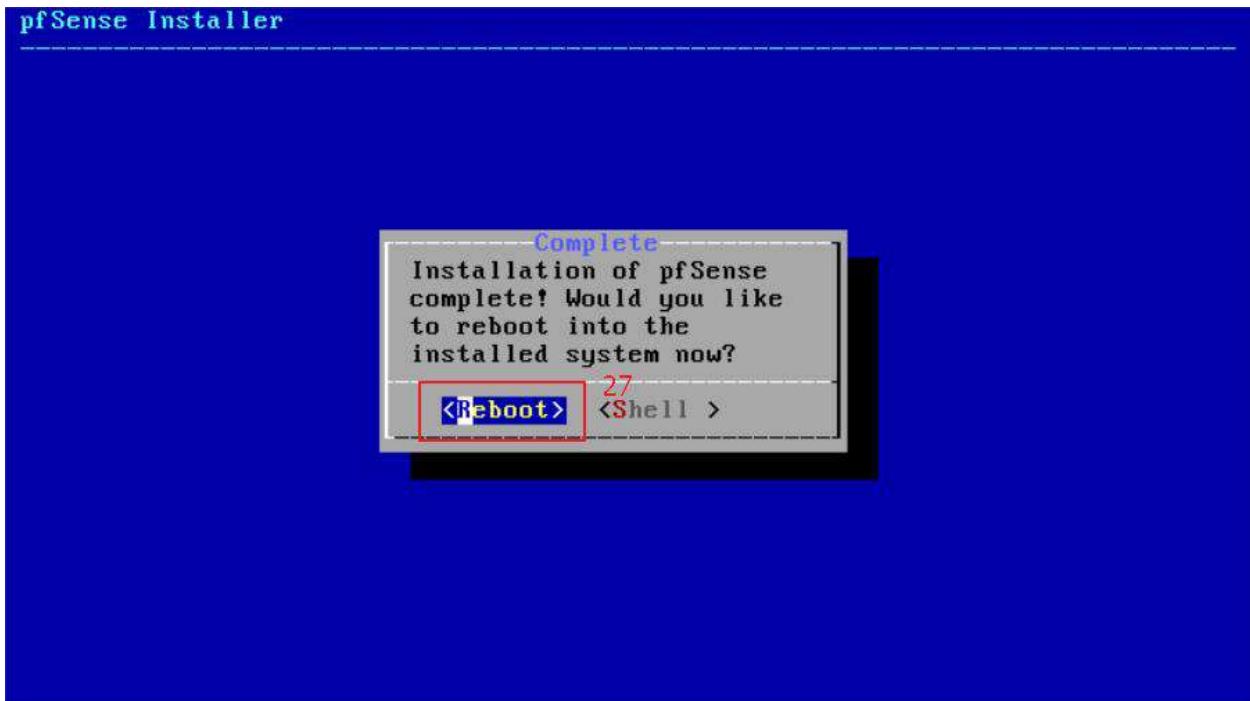


Figure 411: Configuring Pfsense.

Configured Interface in PfSense

If you decide to purchase a Netgate Global TAC Support subscription, you **MUST** have your **Netgate Device ID (NDI)** from your firewall in order to validate support for this unit. Write down your NDI and store it in a safe place. You can purchase TAC support [here](#).

Interface	Status	Description	IP Address
WAN	Up	1000baseT <full-duplex>	10.10.10.2
ELK	Up	1000baseT <full-duplex>	10.10.30.1
DMZ	Up	1000baseT <full-duplex>	10.10.20.1
SURICATA	Up	1000baseT <full-duplex>	n/a
SURICATA_1	Up	1000baseT <full-duplex>	10.10.40.1

Figure 412: configuration of interfaces.

Firewall Rules Configured

The screenshot shows the pfSense Firewall Rules configuration interface. The URL in the browser is 10.10.30.1/firewall_rules.php?if=opt1. The navigation bar includes links for System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, and Help. The main menu shows Firewall / Rules / DMZ. Below this, there are tabs for Floating, WAN, ELK, DMZ (which is selected), and SURICATA, SURICATA_1. The main content area displays a table titled "Rules (Drag to Change Order)" with the following data:

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0 / 0 B	IPv4 TCP/UDP	10.10.20.2	*	10.10.30.3	5601	*	none		kibana Access	🔗 ✍️ 🕒 📋
<input type="checkbox"/>	✓ 0 / 0 B	IPv4 TCP/UDP	10.10.20.2	*	10.10.30.3	9200	*	none		Elasticsearch Access	🔗 ✍️ 🕒 📋
<input type="checkbox"/>	✓ 0 / 2 KIB	IPv4 TCP/UDP	10.10.20.2	*	10.10.30.3	5042	*	none		Logstash Access	🔗 ✍️ 🕒 📋
<input type="checkbox"/>	✗ 0 / 0 B	IPv4 *	DMZ net	*	ELK net	*	*	none			🔗 ✍️ 🕒 📋
<input type="checkbox"/>	✓ 0 / 98 KIB	IPv4 *	DMZ net	*	*	*	*	none			🔗 ✍️ 🕒 📋

At the bottom right of the table are several icons for adding new rules and deleting existing ones.

Figure 413: Configuring firewall rules.

The screenshot shows the pfSense Firewall Rules configuration interface. The URL in the browser is 10.10.30.1/firewall_rules.php?if=lan. The navigation bar includes links for System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, and Help. A notification bar at the top right shows a red alert icon with the number 2. The main title is "Firewall / Rules / ELK". Below it, tabs for Floating, WAN, ELK (selected), DMZ, SURICATA, and SURICATA_1 are visible. The main content area is titled "Rules (Drag to Change Order)" and contains a table of rules:

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	2 / 708 KiB	*	*	*	ELK Address	80	*	*		Anti-Lockout Rule	
<input type="checkbox"/>	0 / 0 B	IPv4 TCP	10.10.20.2	*	10.10.30.3	5601	*	none			
<input type="checkbox"/>	0 / 0 B	IPv4 TCP/UDP	10.10.20.2	*	10.10.30.3	5042	*	none		Web server log	
<input type="checkbox"/>	0 / 0 B	IPv4 TCP/UDP	10.10.40.2	*	10.10.30.3	5044	*	none			
Internet Services											
<input type="checkbox"/>	0 / 0 B	IPv4 ICMP	ELK net	*	*	*	*	none			
<input type="checkbox"/>	9 / 4.53 MiB	IPv4 TCP/UDP	ELK net	*	*	ServiceAccess	*	none			

At the bottom of the table are buttons for Add, Delete, Save, and Revert. A help icon is located on the left side of the table.

Figure 414: Configuring firewall rules.

The screenshot shows the pfSense Firewall Rules configuration interface. The title bar indicates the URL is 10.10.30.1/firewall_rules.php. The top navigation bar includes links for System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, and Help. A notification icon shows 2 alerts.

The main title is "Firewall / Rules / WAN". Below it, tabs for Floating, WAN, ELK, DMZ, SURICATA, and SURICATA_1 are present, with "WAN" being the active tab. The title "Rules (Drag to Change Order)" is displayed above a table.

Rules (Drag to Change Order) Table:

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
0/0 B	*	RFC 1918 networks	*	*	*	*	*	*	Block private networks	
0/0 B	*	Reserved Not assigned by IANA	*	*	*	*	*	*	Block bogon networks	

Remote MGMT Section:

IPv4 TCP	192.168.239.0/24	*	This Firewall	RemoteMgmt	*	none					
IPv4 ICMP any	192.168.239.0/24	*	This Firewall	*	*	none					

Inbound NAT Rules Section:

IPv4 TCP	*	*	10.10.20.2	80 (HTTP)	*	none					
----------	---	---	------------	-----------	---	------	--	--	--	--	--

Action Buttons:

- Add
- Add
- Delete
- Save
- Separator

Figure 415: Configuring firewall rules.

The screenshot shows the pfSense Firewall Rules configuration interface. The title bar indicates the URL is 10.10.30.1/firewall_rules.php?if=opt3. The top navigation bar includes links for System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, and Help. A notification icon shows 2 alerts.

The main title is "Firewall / Rules / SURICATA_1". Below it, tabs for Floating, WAN, ELK, DMZ, SURICATA, and SURICATA_1 are present, with "SURICATA_1" being the active tab. The title "Rules (Drag to Change Order)" is displayed above a table.

Rules (Drag to Change Order) Table:

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions				
0/96 KIB	IPv4 *	*	*	*	*	*	*	none						

Action Buttons:

- Add
- Add
- Delete
- Save
- Separator

Figure 416: Configuring firewall rules.

8.6. APPENDIX F: USER FEEDBACK

8.6.1 USER FEEDBACK FROM

User's Feedback

Real Time Threat Detection System with Threat Intelligence is security solution for any organization which monitors real time traffic of data flow, changes in systems, malicious activities, and behavior's of different types of systems and devices used by an organization. This purpose system will collect logs from different network devices, end devices, IDS system etc and store in centralized storage where those logs will be analyzed and compared with help of a Threat Intel Platform for known threat and rule based detection will be used for unknown threat to produce an alert when any malicious or intrusion activities are detected.

This survey is conducted to determine how significant SIEM security solution are important in everyday life in order to detect threats in system or network to improve security. Additionally, this survey encourages me to develop more project features.

This is a Post survey for Final Year Project. Your feedback will be extremely welcomed and helpful in completing my project. All of the responses will remain private.

Email *

Short-answer text

Figure 417: User Feedback form 1.

1. How effective was the real-time threat detection system in identifying potential threats?

- Effective
- Less Effective
- More Effective

2. Which feature do you think is the best in this project ?

- MISP lookup
- Visualization
- File integrity check
- Telegram alert

Figure 418: User Feedback form 2.

3. Were there any false positives or false negatives in the system's detection of threats?

- Many
- Less
- No

4. How satisfied were you with the quality of the threat intelligence used by the system?

- Very satisfied
- Just ok
- Not satisfied

Figure 419: User Feedback form 3.

5. How quickly were threats identified and responded to by the system?

- Fast
- Medium
- Low

6. How user-friendly was the interface for monitoring and analyzing threats?

- Yes
- No

Figure 420: User Feedback form 4.

7. Did the system provide sufficient context and information about identified threats?

- Yes
- No

8. How well did the system integrate with your existing security infrastructure?

- Fully
- Partially
- Not Well

Figure 421: User Feedback form 5.

9. Were there any technical issues or challenges in implementing or using the system?

Yes

No

10. How helpful were the system's reports and analytics in identifying trends and improving overall security?

More

Medium

Less

Figure 422: User Feedback form 6.

11. Would you recommend this real-time threat detection system to others in your industry or organization?

Yes

No

12. Please provide your valuable Feedback *

Long-answer text

Figure 423: User Feedback form 7.

8.6.2 SAMPLE OF FILLED USER FEEDBACK FORMS

Responses cannot be edited

User's Feedback

Real Time Threat Detection System with Threat Intelligence is security solution for any organization which monitors real time traffic of data flow, changes in systems, malicious activities, and behavior's of different types of systems and devices used by an organization. This purpose system will collect logs from different network devices, end devices, IDS system etc and store in centralized storage where those logs will be analyzed and compared with help of a Threat Intel Platform for known threat and rule based detection will be used for unknown threat to produce an alert when any malicious or intrusion activities are detected.

This survey is conducted to determine how significant SIEM security solution are important in everyday life in order to detect threats in system or network to improve security. Additionally, this survey encourages me to develop more project features.

This is a Post survey for Final Year Project. Your feedback will be extremely welcomed and helpful in completing my project. All of the responses will remain private.

Email

bisparov@gmail.com

Figure 424: User Filled Feedback form 1.

1. How effective was the real-time threat detection system in identifying potential threats?

- Effective
- Less Effective
- More Effective

2. Which feature do you think is the best in this project ?

- MISP lookup
- Visualization
- File integrity check
- Telegram alert

Figure 425: User Filled Feedback form 2.

3. Were there any false positives or false negatives in the system's detection of threats?

- Many
- Less
- No

4. How satisfied were you with the quality of the threat intelligence used by the system?

- Very satisfied
- Just ok
- Not satisfied

Figure 426: User Filled Feedback form 3.

5. How quickly were threats identified and responded to by the system?

- Fast
- Medium
- Low

6. How user-friendly was the interface for monitoring and analyzing threats?

- Yes
- No

Figure 427: User Filled Feedback form 4.

7. Did the system provide sufficient context and information about identified threats?

Yes

No

8. How well did the system integrate with your existing security infrastructure?

Fully

Partially

Not Well

Figure 428:User Filled Feedback form 5.

9. Were there any technical issues or challenges in implementing or using the system?

Yes

No

10. How helpful were the system's reports and analytics in identifying trends and improving overall security?

More

Medium

Less

Figure 429: User Filled Feedback form 6.

11. Would you recommend this real-time threat detection system to others in your industry or organization?

Yes

No

12. Please provide your valuable Feedback

real-time threat detection with threat intelligence was quite interesting project and very helpful in threat detection for any organization that wants to protect its assets from cyber attacks. This helped to detect real time threat and have alert function which was awesome. Overall, this system provided reliable and timely threat detection to mitigate potential risks.

Figure 430: User Filled Feedback form 7.

8.7 APPENDIX G: FUTURE WORK

8.7.1 FUTURE WORK

There are many rooms left for future work for this project even though it fulfilled the client needs. Some are feature that can be integrated to this system to make it more effective and advance are:

User Entity Behaviour Analysis (UEBA): User Entity Behaviour Analysis (UEBA) technology can be integrated into current system so the detect limitation of current system can be broken. This technology uses advance analytics and machine learning methodology to detect and respond to threat in real time. This will be especially useful in detecting insider threat through user behaviour analyse technique.

Rootkit Detector: A rootkit detector can be integrated with this system to detects and removes rootkits that are malicious.

8.7.2. READING FOR FUTURE

- User Entity Behaviour Analysis (UEBA)

<https://www.proquest.com/openview/8e8bd52d36d02baa443a50561f3d5f32/1?pq-origsite=gscholar&cbl=18750&diss=y>

<https://ieeexplore.ieee.org/abstract/document/8855782>

- Rootkit Detector

<https://blackhat.com/presentations/bh-usa-05/bh-us-05-sparks.pdf>

https://link.springer.com/chapter/10.1007/978-3-642-24037-9_36