

PENERAPAN ALGORITMA *EXTENDED TINY ENCRYPTION ALGORITHM*
(XTEA) PADA *PORT KNOCKING* UNTUK PENINGKATAN
KEAMANAN JARINGAN



SKRIPSI

Diajukan Sebagian salah Satu syarat untuk menyelesaikan
Pendidikan Diploma Empat (D-4) Program Studi Teknik Komputer dan
Jaringan Teknik Elektro
Politeknik Negeri Ujung Pandang

MUHAMMAD RIFQI MUWAFFAQ
42519013

PROGRAM STUDI D-4 TEKNIK KOMPUTER DAN JARINGAN
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI UJUNG PANDANG
MAKASSAR
2023

HALAMAN PENGESAHAN

Proposal skripsi ini dengan judul **PENERAPAN ALGORITMA XTEA (EXTENDED TINY ENCRYPTION ALGORITHM) PADA PORT KNOCKING UNTUK PENINGKATAN KEAMANAN JARINGAN** oleh Muhammad Rifqi Muwaffaq NIM 425 19 013 dinyatakan layak untuk diseminarkan.

Makassar, 19 Juni 2023

Mengesahkan,

Pembimbing I

Irfan Syamsuddin, S.T., M.Com, ISM, Ph.D.
NIP. 19820503 201404 2 002

Pembimbing II

Muh. Fajri Raharjo, S.T., M.T.
NIP. 197005211996011001

Mengetahui,
Ketua Program Studi
Teknik Komputer dan Jaringan









Eddy Turigadi, S.T., M.T.
NIP. 19790823 201012 1 001

HALAMAN PENERIMAAN

Pada hari ini, Senin tanggal 14 Agustus 2023 Tim Penguji Ujian Sidang Skripsi telah menerima dengan baik skripsi oleh mahasiswa: **MUHAMMAD RIFQI MUWAFFAQ NIM 425 19 013** dengan judul **“PENERAPAN ALGORITMA *EXTENDED TINY ENCRYPTION ALGORITHM (XTEA)* PADA *PORT KNOCKING* UNTUK PENINGKATAN KEAMANAN JARINGAN”**.

Makassar, 14 Agustus 2023

Tim Penguji Ujian Sidang Skripsi:

- | | | |
|---|------------|---|
| 1. Iin Karmila Yusri, S.ST., M.Eng., Ph.D.. | Ketua | () |
| 2. Tantri Indrabulan, S.T, M.T. | Sekretaris | () |
| 3. Rini Nur, S.T., M.T. | Anggota | () |
| 4. Muhammad Nur Yasir Utomo, S.ST., M.Eng. | Anggota | () |
| 5. Irfan Syamsuddin, S.T, M.Com, ISM, Ph.D. | Anggota | () |
| 6. Muh. Fajri Raharjo, S.T, M.T. | Anggota | () |

KATA PENGANTAR

Alhamdulillah puji syukur atas segala nikmat dan karunia tak terhitung yang diberikan oleh sang Maha Esa, Allah SWT, sehingga penulis mampu menyelesaikan skripsi ini dengan baik. Shalawat serta salam banyak tercurah kepada Rasulullah SAW sebagai sebaik-baik panutan bagi seluruh umat manusia.

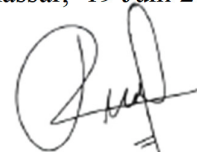
Sebagai salah satu syarat untuk menyelesaikan studi serta dalam rangka memperoleh gelar diploma IV (D-4/S1 Terapan) pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang, maka skripsi ini disusun dengan sebaik-baiknya. Penulis tentunya menyadari bahwa keberhasilan skripsi ini tidak lepas dari bantuan berbagai pihak baik secara langsung maupun tidak langsung. Oleh karenanya, penulis menyampaikan apresiasi dengan menghaturkan terima kasih yang sebesar-besarnya kepada:

1. Orang tua penulis yakni Bapak Munarsin Amry. dan Nurhayati H. Sabani yang sampai saat ini senantiasa memberikan doa terbaik, memberikan semangat, motivasi dan dukungan kepada penulis.
2. Bapak Prof. Ir. Ilyas Mansur, M.T selaku Direktur Politeknik Negeri Ujung Pandang.
3. Bapak Ahmad Rizal Sultan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.
4. Bapak Eddy Tungadi, S.T., M.T. selaku Koordinator Program Studi Teknik Komputer dan Jaringan.

5. Irfan Syamsuddin,S.T. M.Com.ISM.,Ph.D. selaku pembimbing I dan Bapak Muh. Fajri Raharjo,S.T, M.T. selaku pembimbing II atas segala ilmu, motivasi, nasehat, arahan, pandangan, bantuan dan kesedian waktu dan kesabarannya dalam membimbing penulis hingga terselesaikannya penelitian ini.
6. Seluruh dosen dan Staf Jurusan Teknik Elektro, khususnya Program Studi D4 Teknik Komputer dan Jaringan.
7. Teman-teman seperjuangan di Program Studi Teknik Komputer dan Jaringan Angkatan 2019 yang telah berjuang bersama selama 4 tahun, mengajarkan berbagai banyak hal baik dari segi akademik maupun non akademik.
8. Semua pihak yang telah memberikan bantuan moril maupun materi yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, sehingga penulis mengharap kritik dan saran yang membangun demi perbaikan dimasa mendatang. Semoga tulisan ini bermanfaat.

Makassar, 19 Juni 2023



Penulis

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PENERIMAAN	iii
KATA PENGANTAR	ii
DAFTAR ISI.....	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN.....	xvii
SURAT PERNYATAAN.....	xviii
RINGKASAN	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Ruang Lingkup Penelitian	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Jaringan Komputer	4
2.2 <i>Firewall</i>	6
2.3 <i>Web Server</i>	7
2.4 Secure Shell (SSH).....	8
2.5 <i>File Transfer Protocol (FTP)</i>	8

2.6	Telnet (<i>Telecommunication Network</i>)	9
2.7	SMTP (<i>Simple Mail Transport Potocol</i>)	9
2.8	<i>Port Scanning Attack</i>	10
2.9	<i>Sniffing Attack</i>	10
2.10	<i>Port Knocking</i>	11
2.11	Algoritma Simetri.....	19
2.11.1	Algoritma TEA.....	20
2.11.2	Algoritma XTEA.....	21
BAB III METODE PENELITIAN.....		23
3.1	Tempat dan Waktu Penelitian	23
3.2	Prosedure Penelitian	24
3.2.1	Identifikasi Masalah.....	25
3.2.2	Analisis Kebutuhan	27
3.2.3	Perancangan Sistem	28
3.2.4	Skenario Pengujian.....	33
3.2.5	Pengujian Sistem.....	34
3.2.6	Hasil Penelitian	36
3.2.7	Evaluasi.....	36
BAB IV HASIL DAN PEMBAHASAN		38
4.1	Pengujian <i>Server</i>	38
4.1.1	Pengujian <i>Server</i> Tidak Ada Sistem Keamanan	37
4.1.2	Pengujian <i>Server</i> Menggunakan <i>Port Knocking</i>	62
4.1.3	Pengujian <i>Server</i> Menggunakan <i>Port Knocking</i> dan Algoritma XTEA	107

4.2	Tabel Pengujian	159
4.3.1	Tabel Pengujian <i>Server</i>	159
BAB V PENUTUP.....		166
5.1	Kesimpulan	166
5.2	Saran.....	167
DAFTAR PUSTAKA		168
LAMPIRAN.....		172



DAFTAR GAMBAR

Gambar 2. 1 Satu putaran enkripsi dalam Jaringan Feistel.....	22
Gambar 3. 1 Diagram Alir Prosedur Perancangan.....	24
Gambar 3. 2 Admin Mengakses Port Keadaan Normal.....	25
Gambar 3. 3 Penerapan Port knocking Pada Server	26
Gambar 3. 4 Arsitektur Sistem.....	29
Gambar 3. 5 Arsitektur Program.....	30
Gambar 3. 6 Diagram Port Knocking	31
Gambar 3. 7 Diagram Port Knocking dan Algoritma XTEA	32
Gambar 3. 8 Usecase Admin.....	32
Gambar 3. 9 Skenario Pengujian.....	33
Gambar 3. 10 Pengujian Keadaan Normal.....	34
Gambar 3. 11 Pengujian Penerapan Port knocking.....	35
Gambar 3. 12 Pengujian Penerapan Port knocking dan Algoritma XTEA pada Server	35
Gambar 4. 1 Konfigurasi Putty	37
Gambar 4. 2 Proses Login Putty	38
Gambar 4. 3 Proses Login Telah Berhasil Dilakukan.....	38
Gambar 4. 4 Masuk ke Super User	38
Gambar 4. 5 Proses PING	39
Gambar 4. 6 Proses Login Admin.....	39
Gambar 4. 7 Mask Super User	39
Gambar 4. 8 SSH Berhasil Diakses.....	40
Gambar 4. 9 Konfigurasi Putty	41
Gambar 4. 10 Proses Login Putty	41
Gambar 4. 11 Proses Login Telah Berhasil Dilakukan.....	42
Gambar 4. 12 Masuk Super User	42
Gambar 4. 13 Proses PING	42
Gambar 4. 14 Proses Login Admin.....	43
Gambar 4. 15 Masuk Super User	43
Gambar 4. 16 TELNET Berhasil Diakses.....	44
Gambar 4. 17 Konfigurasi Putty	45
Gambar 4. 18 Proses Login Putty	45
Gambar 4. 19 Proses Login Telah Berhasil Dilakukan.....	46
Gambar 4. 20 Masuk Super User	46
Gambar 4. 21 Proses PING	46
Gambar 4. 22 Proses Login Admin.....	47
Gambar 4. 23 Masuk Super User	47
Gambar 4. 24 HTTP Berhasil Diakses	48
Gambar 4. 25 Konfigurasi Putty	48
Gambar 4. 26 Proses Login Putty	48

Gambar 4. 27 Login Telah Berhasil Dilakukan	49
Gambar 4. 28 Masuk Super User	49
Gambar 4. 29 Proses PING	49
Gambar 4. 30 Proses Login Admin.....	50
Gambar 4. 31 Masuk Super User	50
Gambar 4. 32 FTP Berhasil Diakses	51
Gambar 4. 33 Konfigurasi Putty	51
Gambar 4. 34 Proses Login Putty	51
Gambar 4. 35 Proses Login Telah Berhasil Dilakukan.....	52
Gambar 4. 36 Masuk Super User	52
Gambar 4. 37 Proses PING	52
Gambar 4. 38 Proses Login Admin.....	53
Gambar 4. 39 Proses Masuk Super User.....	53
Gambar 4. 40 SMTP Berhasil Diakses	53
Gambar 4. 41 Penyerangan Port Scanning SSH Sebelum Port knocking	54
Gambar 4. 42 Penyerangan Port Scanning TELNET Sebelum Port knocking.....	55
Gambar 4. 43 Penyerangan Port Scanning HTTP Sebelum Port knocking	56
Gambar 4. 44 Penyerangan Port Scanning FTP Sebelum Port knocking	56
Gambar 4. 45 Penyerangan Port Scanning SMTP Sebelum Port knocking.....	57
Gambar 4. 46 Port SSH.....	58
Gambar 4. 47 IP Server.....	58
Gambar 4. 48 Port TELNET	59
Gambar 4. 49 IP Server.....	59
Gambar 4. 50 Port HTTP	60
Gambar 4. 51 IP Server.....	60
Gambar 4. 52 Port FTP	61
Gambar 4. 53 IP Server.....	61
Gambar 4. 54 Port SMTP.....	61
Gambar 4. 55 IP SMTP.....	62
Gambar 4. 56 Konfigurasi Port knocking	63
Gambar 4. 57 Konfigurasi Putty	64
Gambar 4. 58 Proses Login Putty	64
Gambar 4. 59 Proses Login Telah Berhasil Dilakukan.....	65
Gambar 4. 60 Masuk ke Super User	65
Gambar 4. 61 Proses PING	65
Gambar 4. 62 Mendrop Akses Server	66
Gambar 4. 63 Akses Server di Drop	66
Gambar 4. 64 Konfigurasi Putty	67
Gambar 4. 65 Server Tidak Dapat di Akses.....	67
Gambar 4. 66 Pengecekan Status Knockd	68
Gambar 4. 67 Proses Login Admin.....	68
Gambar 4. 68 Masuk ker Super User	69
Gambar 4. 69 Proses Membuka SSH Ketukan Salah	69

Gambar 4. 70 Proses Membuka SSH Ketukan Benar.....	70
Gambar 4. 71 Proses Menutup SSH Ketukan Salah	71
Gambar 4. 72 Proses Menutup SSH Ketukan Benar	71
Gambar 4. 73 Konfigurasi Putty	72
Gambar 4. 74 Proses Login Putty	72
Gambar 4. 75 Proses Login Telah Berhasil Dilakukan.....	73
Gambar 4. 76 Masuk ke Super User	73
Gambar 4. 77 Proses PING	73
Gambar 4. 78 Mendrop Akses Server	74
Gambar 4. 79 Akses Server di Drop	74
Gambar 4. 80 Konfigurasi Putty	75
Gambar 4. 81 Server Tidak Dapat di Akses.....	75
Gambar 4. 82 Pengecekan Status Knockd	76
Gambar 4. 83 Proses Login Admin.....	76
Gambar 4. 84 Masuk ke Super <i>User</i>	76
Gambar 4. 85 Proses Membuka TELNET Ketukan Salah.....	77
Gambar 4. 86 Proses Membuka TELNET Ketukan Benar	77
Gambar 4. 87 Proses Menutup TELNET Ketukan Salah	78
Gambar 4. 88 Proses Menutup TELNET Ketukan Benar.....	78
Gambar 4. 89 Konfigurasi Putty	79
Gambar 4. 90 Proses Login Putty	79
Gambar 4. 91 Proses Login Telah Berhasil Dilakukan.....	80
Gambar 4. 92 Masuk ke Super User	80
Gambar 4. 93 Proses PING	80
Gambar 4. 94 Mendrop Akses Server	81
Gambar 4. 95 Server Tidak Dapat Diakses	81
Gambar 4. 96 Pengecekan Status Knockd	82
Gambar 4. 97 Proses Login Admin.....	82
Gambar 4. 98 Masuk ke Super User	82
Gambar 4. 99 Proses Membuka HTTP Ketukan Salah.....	83
Gambar 4. 100 HTTP Tidak Dapat Diakses	83
Gambar 4. 101 Gambar Proses Membuka HTTP Ketukan Benar	83
Gambar 4. 102 HTTP Dapat Diakses.....	84
Gambar 4. 103 Proses Menutup HTTP Ketukan Salah	84
Gambar 4. 104 HTTP Masih Dapat Diakses.....	84
Gambar 4. 105 Proses Menutup HTTP Ketukan Benar.....	85
Gambar 4. 106 HTTP Tidak Dapat Diakses	85
Gambar 4. 107 Konfigurasi Putty	86
Gambar 4. 108 Proses Login Putty	86
Gambar 4. 109 Proses Login Telah Berhasil Dilakukan.....	87
Gambar 4. 110 Masuk ke Super User	87
Gambar 4. 111 Proses PING	87
Gambar 4. 112 Mendrop Akses <i>Server</i>	88

Gambar 4. 113 Server Tidak Dapat Diakses.....	88
Gambar 4. 114 Pengecekan Satus Knockd	89
Gambar 4. 115 Proses Login Admin.....	89
Gambar 4. 116 Masuk ke Super User	89
Gambar 4. 117 Proses Membuka FTP Ketukan Salah.....	90
Gambar 4. 118 Proses Membuka FTP Ketukan Benar	90
Gambar 4. 119 Proses Menutup FTP Ketukan Salah.....	91
Gambar 4. 120 Proses Menutup FTP Ketukan Benar	91
Gambar 4. 121 Konfigurasi Putty	92
Gambar 4. 122 Proses Login Putty	92
Gambar 4. 123 Proses Login Telah Berhasil	93
Gambar 4. 124 Masuk ke Super User	93
Gambar 4. 125 Proses PING	93
Gambar 4. 126 Mendrop Akses Server	94
Gambar 4. 127 Akses Server di Drop	94
Gambar 4. 128 Pengecekan Status Knockd	95
Gambar 4. 129 Proses Login Admin.....	95
Gambar 4. 130 Masuk Super User.....	95
Gambar 4. 131 Proses Membuka SMTP Ketukan Salah	96
Gambar 4. 132 Proses Membuka SMTP Ketukan Benar.....	96
Gambar 4. 133 Proses Menutup SMTP Ketukan Salah	97
Gambar 4. 134 Proses Membuka SMTP Ketukan Benar.....	97
Gambar 4. 135 Penyerangan Port Scanning SSH Setelah Port knocking	98
Gambar 4. 136 Penyerangan Port Scanning TELNET Setelah Port knocking	99
Gambar 4. 137 Penyerangan Port Scanning HTTP Setelah Port knocking	99
Gambar 4. 138 Penyerangan Port Scanning FTP Setelah Port knocking	100
Gambar 4. 139 Penyerangan Port Scanning SMTP Setelah Port knocking	101
Gambar 4. 140 Sequence SSH	102
Gambar 4. 141 IP Server	102
Gambar 4. 142 Port SSH.....	102
Gambar 4. 143 Sequence TELNET	103
Gambar 4. 144 IP Server	103
Gambar 4. 145 Port TELNET	103
Gambar 4. 146 Sequence HTTP.....	104
Gambar 4. 147 IP Server	104
Gambar 4. 148 Port HTTP	104
Gambar 4. 149 Sequence FTP.....	105
Gambar 4. 150 IP Server	105
Gambar 4. 151 Port FTP	105
Gambar 4. 152 Sequence SMTP	106
Gambar 4. 153 IP Server	106
Gambar 4. 154 Port SMTP.....	107
Gambar 4. 155 Konfigurasi Port knocking	109

Gambar 4. 156 Konfigurasi Putty	110
Gambar 4. 157 Proses Login Putty	110
Gambar 4. 158 Proses Login Telah Berhasil Dilakukan.....	111
Gambar 4. 159 Masuk Super User	111
Gambar 4. 160 Proses PING	111
Gambar 4. 161 Mendrop Akses Server	112
Gambar 4. 162 Akses Server di Drop	112
Gambar 4. 163 Konfigurasi Putty	113
Gambar 4. 164 Server Tidak Dapat Diakses	113
Gambar 4. 165 Pengecekan Status Knockd	114
Gambar 4. 166 Proses Login Admin.....	114
Gambar 4. 167 Masuk ke Super User	114
Gambar 4. 168 Proses Membuka SSH Ketukan Salah	115
Gambar 4. 169 Membuka SSH Ketukan Benar	116
Gambar 4. 170 Menutup SSH Ketukan Salah.....	116
Gambar 4. 171 Menutup SSH Ketukan Benar	117
Gambar 4. 172 Konfigurasi Putty	117
Gambar 4. 173 Proses Login Putty	118
Gambar 4. 174 Proses Login Telah Berhasil Dilakukan.....	118
Gambar 4. 175 Masuk ke Super User	118
Gambar 4. 176 Proses PING	119
Gambar 4. 177 Mendrop Akses Server	119
Gambar 4. 178 Akses Server di Drop	119
Gambar 4. 179 Konfigurasi Putty	120
Gambar 4. 180 Server Tidak Dapat Diakses	120
Gambar 4. 181 Pengecekan Status Knockd	121
Gambar 4. 182 Proses Login Admin.....	121
Gambar 4. 183 Masuk ke Super User	122
Gambar 4. 184 Membuka TELNET Ketukan Salah	122
Gambar 4. 185 Membuka TELNET Ketukan Benar	123
Gambar 4. 186 Menutup TELNET Ketukan Salah.....	124
Gambar 4. 187 Menutup TELNET Ketukan Benar	125
Gambar 4. 188 Konfigurasi Putty	125
Gambar 4. 189 Proses Login Putty	126
Gambar 4. 190 Proses Login Telah Berhasil Dilakukan.....	126
Gambar 4. 191 Masuk ke Super User	126
Gambar 4. 192 Proses PING	127
Gambar 4. 193 Mendrop Akses Server	127
Gambar 4. 194 Akses Server di Drop	127
Gambar 4. 195 Pengecekan Status Knockd	128
Gambar 4. 196 Proses Login Admin.....	128
Gambar 4. 197 Masuk Super User	129
Gambar 4. 198 Proses Membuka HTTP Ketukan Salah.....	129

Gambar 4. 199 HTTP Tidak Dapat Diakses	130
Gambar 4. 200 Membuka HTTP Ketukan Benar.....	130
Gambar 4. 201 HTTP Dapat Diakses.....	130
Gambar 4. 202 Menutup HTTP Ketukan Salah	131
Gambar 4. 203 HTTP Dapat Diakses.....	131
Gambar 4. 204 Menutup HTTP Ketukan Benar	132
Gambar 4. 205 HTTP Tidak Dapat Diakses	132
Gambar 4. 206 Konfigurasi Putty	133
Gambar 4. 207 Proses Login Putty	133
Gambar 4. 208 Proses Login Telah Berhasil Dilakukan.....	134
Gambar 4. 209 Masuk ke Super User	134
Gambar 4. 210 Proses PING	134
Gambar 4. 211 Mendrop Akses Server	135
Gambar 4. 212 Pengecekan Status Knockd	135
Gambar 4. 213 Proses Login Admin.....	136
Gambar 4. 214 Masuk ke Super User	136
Gambar 4. 215 Proses Membuka FTP Ketukan Salah.....	137
Gambar 4. 216 Proses Membuka FTP Ketukan Benar	138
Gambar 4. 217 Proses Menutup FTP Ketukan Salah.....	139
Gambar 4. 218 Proses Menutup FTP Ketukan Benar.....	139
Gambar 4. 219 Konfigurasi Putty	140
Gambar 4. 220 Proses Login.....	140
Gambar 4. 221 Proses Login Telah Berhasil Dilakukan.....	141
Gambar 4. 222 Masuk Super User	141
Gambar 4. 223 Proses PING	141
Gambar 4. 224 Mendrop Akses Server	142
Gambar 4. 225 Server Tidak Dapat Diakses.....	142
Gambar 4. 226 Pengecekan Status Knockd	143
Gambar 4. 227 Proses Login Admin.....	143
Gambar 4. 228 Masuk ke Super User	144
Gambar 4. 229 Proses Membuka SMTP Ketukan Salah	144
Gambar 4. 230 Proses Membuka SMTP Ketukan Benar.....	145
Gambar 4. 231 Menutup Server Ketukan Salah.....	146
Gambar 4. 232 Menutup Server Ketukan Benar	147
Gambar 4. 233 Penyerangan Port Scanning SSH Setelah Port knocking	148
Gambar 4. 234 Penyerangan Port Scanning TELNET Setelah Port knocking ...	149
Gambar 4. 235 Penyerangan Port Scanning HTTP Setelah Port knocking	149
Gambar 4. 236 Penyerangan Port Scanning FTP Setelah Port knocking	150
Gambar 4. 237 Penyerangan Port Scanning SMTP Setelah Port knocking	150
Gambar 4. 238 Sequence Enkripsi SSH.....	151
Gambar 4. 239 IP Server	152
Gambar 4. 240 Port SSH.....	152
Gambar 4. 241 Sequence Enkripsi TELNET	153

Gambar 4. 242 IP Server	153
Gambar 4. 243 Port Server	153
Gambar 4. 244 Sequence Enkripsi HTTP	154
Gambar 4. 245 IP Server	154
Gambar 4. 246 Port HTTP	154
Gambar 4. 247 Enkripsi Sequence FTP	155
Gambar 4. 248 IP Server	156
Gambar 4. 249 Port FTP	156
Gambar 4. 250 Enkripsi Sequence SMTP	157
Gambar 4. 251 IP Server	157
Gambar 4. 252 Port SMTP	158



DAFTAR TABEL

Tabel 4. 1 Pengujian <i>Server</i>	159
--	-----



DAFTAR LAMPIRAN

Lampiran 1: <i>Attacker</i> Melakukan Penyadapan Pada <i>Server</i>	172
Lampiran 2: Penjelasan Script	181



SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama: Muhammad Rifqi Muwaffaq

NIM: 42519013

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam skripsi ini yang berjudul **“PENERAPAN ALGORITMA *EXTENDED TINY ENCRYPTION ALGORITHM* (XTEA) PADA *PORT KNOCKING* UNTUK PENINGKATAN KEAMANAN JARINGAN”** merupakan gagasan dan hasil karya saya sendiri dengan arahan komisi pembimbing, dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi dan instansi manapun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam skripsi ini.

Jika pernyataan saya tersebut diatas tidak benar, saya siap menanggung resiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, 19 Juni 2023



Muhammad Rifqi Muwaffaq
19013

**PENERAPAN ALGORITMA *EXTENDED TINY ENCRYPTION*
ALGORITHM (XTEA) PADA *PORT KNOCKING*
UNTUK PENINGKATAN KEAMANAN**

RINGKASAN

Berbagai jenis serangan dapat terjadi pada jaringan internet, salah satu jenis ancaman dapat terjadi pada *server* yaitu serangan yang ditargetkan pada suatu *port* dalam kondisi terbuka. Terdapat metode yang dapat mengatasi masalah *port* dalam kondisi terbuka yaitu dengan menggunakan metode *port knocking*, dengan menggunakan *port knocking* maka *port-port* akan terlihat tertutup, namun *port knocking* masih terdapat masalah dalam sistem keamanan karena saat seorang *admin* jaringan melakukan proses *knocking*, *sequence port* yang sedang terbuka berbentuk *plaint text* yang mudah untuk dipahami.

Penelitian ini memberikan tingkat keamanan tambahan di model *port knocking* yang ada dengan meningkatkan kerumitan penyerang dalam menemukan *sequence port knocking* yang benar dengan melakukan *enkripsi* terhadap *sequence port* dengan menggunakan Algoritma *Extended Tiny Encryption Algorithm* (XTEA) sehingga *sequence port* yang digunakan untuk mengetuk *port* yang terdapat pada *server* berbentuk *chipertext* yang sulit untuk dipahami.

Berdasarkan implementasi dan pengujian yang dilakukan untuk penerapan *port knocking* pada *server* ketika *admin* melakukan *remote server* dan penerapan algoritma XTEA pada *port knocking* disimpulkan bahwa penerapan *port knocking* dapat memberikan keamanan pada *server* jika menerima serangan *port scanning* tetapi tidak dapat mengamankan *server* dari serangan *sniffing*. Penerapan algoritma XTEA pada *port knocking* menambah tingkat keamanan pada *server* terutama jika *attacker* melakukan serangan *sniffing*.

Kata kunci : Sistem Keamanan, *Firewall*, *Port Knocking*, Algoritma XTEA



BAB I PENDAHULUAN

1.1 Latar Belakang

Berbagai jenis serangan yang dapat terjadi pada jaringan internet menjadikan alasan penting adanya sebuah sistem keamanan. Adapun salah satu jenis ancaman yang terjadi pada *server* adalah serangan yang ditargetkan pada suatu *port* yang berada dalam kondisi terbuka, sehingga dapat menjadikan orang yang tidak mempunyai hak akses dapat melakukan *port scanning* untuk menyusup ke dalam *server* (Suhendar, Sajati, and Astuti 2013). *Port-port* yang terbuka ini rawan terhadap eksploitasi dari akses yang tidak diinginkan, untuk itu dibutuhkan suatu sistem yang dapat menangkal masalah tersebut (Hasbi Muhammad, I Wayan Agus Rimbawa and Andi Hidayat Jatmika 2019).

Untuk mengatasi jenis ancaman tersebut maka dalam mengamankan *server* seorang *admin* dituntut untuk bekerja lebih keras. Berbagai cara telah diterapkan misalnya menggunakan *firewall* sebagai dinding penghalang pembatasan akses. Dalam penggunaan *firewall* sendiri masih terdapat kekurangan dikarenakan menutup semua akses tanpa memperhatikan siapapun yang sedang terkoneksi dalam jaringan. Suatu metode keamanan yang dapat menutup celah dan masalah pembatasan hak akses berdasarkan *firewall* yaitu dengan menggunakan metode *port knocking*, metode ini dapat digunakan dalam proses mengamankan *server* (Linux dan Unix) dan melakukan monitoring jaringan melalui pembatasan akses *blocking* pada *port* yang terdapat dalam jaringan (Iqbal, Arini, and Bayu Suseno 2020).

Port knocking adalah sebuah metode autentikasi yang dapat diterapkan untuk menyembunyikan *service port*, serta dapat juga diterapkan untuk membuka akses pada *service port* tertutup harus menggunakan ketukan *port sequence* (Andreatos 2017). Dengan cara ini, perangkat jaringan misalnya *router* akan lebih aman, karena *admin* jaringan dapat melakukan filtering terhadap *port-port* yang rentan terhadap serangan. Jika melakukan proses *port scanning* maka *port-port* tersebut terlihat tertutup. Dari sisi *admin* jaringan tetap dapat melakukan konfigurasi dan monitoring akan tetapi dengan menggunakan metode autentikasi agar dapat diijinkan oleh *firewall* untuk mengakses *port*. Pada *port knocking* terdapat istilah *knocking* atau disebut autentikasi merupakan usaha untuk membuka *port* yang dalam kondisi tertutup dengan cara mengakses beberapa *port* komunikasi ketika beberapa *port* komunikasi diakses dengan kombinasi tertentu, maka akan terbuka sebuah *port* (Devie Ryana Suchendra¹, Alfian Fitra Rahman² 2017).

Dalam proses melakukan *knocking* terhadap suatu *server* masih terdapat suatu masalah dalam sistem keamanan karena saat seorang *admin* jaringan melakukan proses *knocking*, *sequence port* yang sedang terbuka berbentuk *plaintext* yang mudah untuk dipahami sehingga terdapat kemungkinan seseorang yang melakukan penyadapan dapat dengan mudah mengetahui *sequence port* yang terpasang pada *server*.

Dari berbagai uraian permasalahan keamanan serta cara penanganannya maka penelitian ini membahas masalah ini dengan mengusulkan kerangka kerja yang akan memberikan tingkat keamanan tambahan di model *port knocking*

yang ada. Kerangka kerja ini akan meningkatkan kerumitan penyerang dalam menemukan *sequence port knocking* yang benar dengan melakukan *enkripsi* terhadap *sequence port* dengan menggunakan Algoritma *Extended Tiny Encryption Algorithm (XTEA)* sehingga *sequence port* yang digunakan untuk mengetuk *port* yang terdapat pada *server* berbentuk *chipertext* yang sulit untuk dipahami.

1.2 Rumusan Masalah

- a. Bagaimana menerapkan *port knocking* ketika *admin* melakukan *remote server*?
- b. Bagaimana menerapkan Algoritma XTEA pada *port knocking*?

1.3 Ruang Lingkup Penelitian

- a. Implementasi *port knocking* ketika *admin* melakukan *remote server*.
- b. Implementasi algoritma XTEA dalam enkripsi ketika melakukan *knocking*.

1.4 Tujuan Penelitian

- a. Mengamankan *server* ketika *admin* melakukan *remote* dengan metode *port knocking*.
- b. Memberikan tingkat keamanan tambahan pada *port knocking*.

1.5 Manfaat Penelitian

- a. Membantu mengamankan *server* ketika *admin* melakukan *remote server*.
- b. Meningkatkan kerumitan penyerang dalam menemukan *port* yang sedang di *remote* oleh *admin*.

BAB II TINJAUAN PUSTAKA

2.1 Jaringan Komputer

Jaringan komputer merupakan dua atau lebih perangkat komputer yang saling terhubung antar satu dengan yang lainnya serta digunakan untuk berbagi sumber data. Jaringan komputer dibangun dengan menggunakan dua kombinasi yaitu *hardware* dan *software*. Saat membangun jaringan komputer, *switch* dan *router* menggunakan *protocol* dan algoritma untuk bertukar informasi sesuai kebutuhan sehingga data dapat ditransfer ke titik akhir (*Admin Koinfo 2020*).

Setiap titik akhir, terkadang disebut *host* pada jaringan, yang memiliki pengidentifikasi unik yaitu alamat IP maupun alamat *Media Access Control* yang digunakan untuk mengidentifikasi sumber atau tujuan. Titik akhir dapat mencakup *server*, telepon, PC, dan jenis perangkat keras lainnya pada jaringan (*Admin Koinfo 2020*).

Jaringan dapat bersifat *private* atau *public*. Adanya jaringan *private*, pengguna biasanya harus memasukkan kredensial untuk mengakses jaringan. Ini biasanya disediakan secara manual oleh *admin* jaringan atau diperoleh langsung dari pengguna melalui kata sandi atau kredensial lainnya. Lain halnya untuk jaringan jaringan *public* misalnya, berbagai akses tidak dibatasi di Internet. (*Admin Koinfo 2020*).

Terdapat beberapa jenis jaringan komputer yang dapat diklasifikasikan berdasarkan cakupan areanya:

1) *Personal Area Network* (PAN)

Personal Area Network atau PAN adalah jenis jaringan yang digunakan untuk menghubungkan berbagai perlengkapan elektronik milik pribadi yang hanya dikelola oleh orang itu sendiri. Misalnya untuk penggunaan jaringan jenis PAN yaitu menghubungkan perangkat printer ke komputer atau contoh lainnya yaitu menghubungkan komputer dengan *speaker Bluetooth* (Admin Kominfo 2020).

2) *Local Area Network* (LAN)

Local Area Network atau LAN adalah penghubung antara perangkat jaringan yang saling berdekatan. Gedung sekolah, kantor, dan bahkan rumah termasuk pada jaringan LAN. Dalam beberapa kasus, LAN dapat menjangkau sekelompok bangunan di sekitarnya. Pada kasus lain LAN juga dapat diterapkan pada warnet yang menghubungkan banyak komputer pada satu *server* (Admin Kominfo 2020).

3) *Metropolitan Area Network* (MAN)

Metropolitan Area Network atau MAN adalah jaringan komputer yang digunakan untuk menghubungkan berbagai perangkat dalam area sebuah kota. Untuk area jangkauan pada jaringan MAN berkisar 10-50 km. MAN hanya terdapat satu atau dua kabel yang tidak dilengkapi dengan elemen *switching* yang dapat berfungsi untuk membuat rancangan menjadi *simple* (Dinda Febrianti 2017).

4) *Wide Area Network* (WAN)

Wide Area Network atau WAN adalah jaringan komputer yang digunakan untuk menghubungkan beberapa perangkat komputer atau berbagai macam tipe jaringan lainnya dalam jangkauan cukup jauh bahkan *dapat* antar negara. Dengan menggunakan jenis jaringan WAN, misalnya terdapat data di dalam komputer yang

terletak di Indonesia dapat di transfer dengan mudah dan cepat ke komputer di negara lainnya, misalnya ke Amerika Serikat (*Admin Kominfo* 2020).

2.2 Firewall

Firewall didefinisikan sebagai sebuah komponen atau kumpulan komponen yang membatasi akses antara sebuah jaringan yang diproteksi dan internet, atau antara kumpulan-kumpulan jaringan lainnya. *Firewall* merupakan solusi untuk mengatasi keamanan di dalam dunia internet baik itu keamanan komputer maupun keamanan jaringan yang banyak dipenuhi dengan berbagai ancaman baik dari dalam maupun dari luar (Basten 2009). *Firewall* dibagi menjadi beberapa jenis yaitu:

1) Packet-filtering firewalls

Firewall jenis ini diletakkan di beberapa tempat seperti *router* atau *switch*. Mempunyai beberapa aturan yang dikonfigurasi berdasarkan alamat IP sumber, *no port*, dan lain-lain. Jika sebuah paket tidak cocok dengan aturan yang dibuat, paket tersebut akan dibuang (Santoso, Noertjahyana, and Andjarwirawan).

2) Circuit-level Gateways

Firewall jenis ini adalah *firewall* sederhana yang tidak memakan banyak sumber daya. *Firewall* ini menerima atau menolak paket berdasarkan lalu lintas pada saat TCP *handshake* *Stateful Inspection Firewall*. Jenis *firewall* ini menggabungkan 2 *firewall* sebelumnya agar mempunyai keamanan yang lebih kuat (Santoso, Noertjahyana, and Andjarwirawan).

3) *Application-level gateways (Proxy firewalls)*

Firewall ini beroperasi di tingkat aplikasi. Menyaring lalu lintas antar jaringan dan sumber. *Firewall Proxy* membuat koneksi dengan sumber dan kemudian memeriksa isi paket (Santoso, Noertjahyana, and Andjarwirawan).

4) *Next-generation Firewalls*

Firewall ini menggabungkan metode seperti *Deep packet inspection*, *Intrusion Prevention System*, *Bandwidth management*, *URL filtering* *Antivirus* dan *Malware detection*. Kelebihan *firewall* ini dapat dapat mencegah ancaman terhadap jaringan dan memberikan tingkat keamanan yang lebih tinggi (Santoso, Noertjahyana, and Andjarwirawan)

5) *Cloud Firewalls*

Firewall ini dapat diimplementasikan dengan bantuan *cloud*. *Firewall Cloud* dianggap sama dengan *firewall proxy* (Santoso, Noertjahyana, and Andjarwirawan n.d.).

2.3 **Web Server**

Web server adalah *software* yang digunakan untuk menerima dan melayani permintaan yang dikirimkan *user* melalui *browser* kemudian ditampilkan kepada *user* sesuai dengan permintaan yang dikirimkan ke *server* (Ali, Muhammad Rasyid, Muhamad Anda Falahuddin, Susilawati St, and M Eng. 2021). Penggunaan paling umum *web server* adalah untuk menempatkan situs *web*. Fungsi utama sebuah *web server* untuk mentransfer berkas atas permintaan pengguna melalui *protocol* komunikasi yang telah ditentukan dan mentransfer ke dalam sebuah halaman *web* (Prisma, Supramana and I Gusti Lanang Putra Eka. 2016). *Web Server* selalu

terhubung ke internet, setiap *web server* yang terhubung ke internet menggunakan alamat unik yang disusun dengan serangkaian empat nomor antara 0 dan 255 yang dipisahkan oleh periode (Haynes 2018).

2.4 Secure Shell (SSH)

Secure Shell atau SSH merupakan *protocol* jaringan yang dapat melakukan pertukaran data melalui saluran aman antar dua perangkat yang banyak digunakan pada sistem berbasis Linux dan Unix (Sakti, Aziz, and Doewes 2016). SSH dirancang sebagai pengganti dari *protocol* telnet dan FTP tetapi tetap pada tujuan utamanya adalah untuk mengamankan komunikasi internet. SSH dapat menjalankan dua hal penting yaitu *console login* (menggantikan telnet) dan *secure filetransfer* (menggantikan FTP), juga memperoleh kemampuan membentuk source tunnel untuk melewati HTTP, FTP, POP3, dan apa pun lainnya melalui SSH tunnel (Cahyani 2011).

2.5 File Transfer Protocol (FTP)

File Transfer Protocol merupakan sebuah *protocol* internet yang digunakan untuk melakukan proses pertukaran *file* antar komputer dalam sebuah jaringan. FTP juga merupakan lapisan *application layer protocol* yang biasa digunakan untuk mentransfer data. Kemudian FTP dikembangkan agar dapat melakukan transfer data di antara FTP *client* dan FTP *server*.

FTP *client* merupakan aplikasi yang terdapat di dalam komputer untuk melakukan proses *download* dan *upload file* dari FTP *server* (Khadafi, Nurmuslimah, and Anggakusuma 2019). FTP *server* merupakan sebuah *Windows*

Service atau daemon yang aktif di dalam komputer untuk melakukan respon perintah dari *client* FTP (Kurniawan and Herryanto 2017).

2.6 Telnet (*Telecommunication Network*)

Telnet (*Telecommunication Network*) adalah protocol *client-server* yang digunakan untuk melakukan *remote login* ke komputer target dalam jaringan yang terhubung ke internet maupun secara local (Gatra 2015). Komunikasi dengan menggunakan protokol telnet dapat bekerja antar komputer dan sistem operasi yang berbeda karena semua sistem tersebut memakai protokol yang sama. Telnet menggunakan dua program yaitu salah satunya adalah *client* dan *server* (RochimahS and Bowo 2006).

2.7 SMTP (*Simple Mail Transport Potocol*)

SMTP (*Simple Mail Transport Protocol*) salah protokol *email* dapat digunakan untuk mengirim *email* secara *online* (Syahrir, Najoan, and Sinsuw 2018). Email yang akan dikirim ada di program email *client* (komputer yang mengirim email) dan kemudian dikirim menuju *server* SMTP dan terhubung ke Internet. Kemudian program email lainnya (target pengiriman Email) mengambil email dari Internet melalui *server* POP3, penyedia email penerima (Putra Perdana, Wawan and Noptin Harpawai 2013).

Protokol SMTP cukup sederhana dengan berbasis teks dimana protokol menyebutkan satu atau lebih penerima email untuk ditinjau nanti. Jika penerima masukkan alamat email yang valid, email akan segera dikirim. SMTP menggunakan *port* 25 dan dapat dihubungi melalui program telnet. Untuk

menggunakan *server* SMTP dengan nama domain, lalu entri Domain Name *Server* (DNS). pada bagian Mail Exchange (MX).

2.8 Port Scanning Attack

Target serangan yang berbahaya dilakukan melalui proses mengintai *server* untuk mencari celah sehingga dapat dilancarkan serangan lanjutan. Salah satu serangan yang melakukan aktivitas mencari informasi ialah serangan *Port Scanning* (Rodney R Rohrmann, 2017). *Port scanning* adalah teknik mendeteksi *port-port* yang terbuka pada sebuah komputer. Tujuannya hanyalah untuk melihat *port-port* berapa saja yang terbuka pada komputer tersebut (Albar & Putra, 2022). *Port* yang terbuka rentan untuk diserang oleh *attacker*. Serangan terhadap *port* yang terbuka dapat dihindari dengan menerapkan metode *Port Knocking*. *Port Knocking* bekerja dengan menutup semua *port* yang ada pada sistem komputer dan hanya pengguna tertentu yang dapat mengakses sebuah *port* yang telah ditentukan yaitu dengan cara mengetuk terlebih dahulu. Permasalahan keamanan jaringan sering terjadi dikarenakan terdapat *port* yang terbuka dan secara autentikasi maupun otorisasi menyebabkan pengguna yang tidak valid dapat mengakses jaringan secara ilegal (Brades & Irwansyah, 2022). Maka dari itu serangan *Port Scanning* sangat berbahaya bagi suatu jaringan karena dapat memanfaatkan kelemahan *server* untuk melancarkan aksinya (Achmad R., Manullang, E. V., and Sanmas, 2020).

2.9 Sniffing Attack

Serangan *sniffing* adalah teknik pemantauan setiap paket yang melintasi jaringan, dan bagian dari perangkat lunak atau perangkat keras yang memonitor semua lalu lintas jaringan. Potensi bahaya packet *sniffing* adalah hilangnya privasi,

dan tercurinya informasi penting dan rahasia yang dimiliki oleh *user* (Dimas Perdana, Jayanta, 2023). *Sniffing* biasanya menyerang protokol-protokol seperti Telnet, HTTP, POP, IMAP, SBM, FTP, dan lain-lain. Dalam metode *hacking, sniffing* dibagi menjadi dua bagian *passive sniffing* dan *active sniffing* (Windi Agustiara & Satrio, 2022). *Sniffing attack*, teknik dimana data paket yang mengalir melalui jaringan terdeteksi dan diamati *administrator* jaringan menggunakan alat *sniffing* paket untuk memantau dan memvalidasi lalu lintas jaringan, sementara peretas dapat menggunakan alat serupa untuk tujuan jahat. Dapat disimpulkan bahwa *Sniffing attack* adalah teknik penyadapan melalui proses penangkapan aliran paket data yang melalui jaringan tertentu dengan menggunakan alat *sniffing* (Putranto Jayanta and Bayu Hananto, 2023).

2.10 Port Knocking

Metode yang dilakukan oleh *port knocking* adalah membuka akses ke *port* tertentu yang telah *diblock*. Koneksi pada *port knocking* dapat berupa TCP, UDP, maupun ICMP. Jika *host* telah mengirimkan koneksi yang telah sesuai dengan *rule knocking*, maka *port* yang sudah *diblock* akan diberikan akses secara dinamis oleh *firewall*. Maka *router* akan lebih aman, karena *admin* jaringan dapat melakukan blocking terhadap *port-port* yang rentan seperti Winbox (tcp 8291), SSH (tcp 22), Telnet (tcp 23) atau webfig (tcp 80). Jika proses *port scanning* dilakukan maka *port-port* akan terlihat tertutup. Di sisi *admin* jaringan, konfigurasi dan monitoring masih dapat dilakukan, tetapi langkah-langkah khusus (*knocking*) diperlukan untuk memungkinkan *firewall* mengakses *port* seperti Winbox dan SSH (Saputro, Andik, Nanang Saputro, Hendro Wijayanto, and Program Studi 2020).

Port knocking melakukan pengetukan terhadap *port-port* komunikasi yang terdapat pada sistem komunikasi data. Pada pengetukan tersebut dilakukan dengan kombinasi tertentu secara berurutan dalam rentan waktu tertentu. Apabila kombinasi dari pengetukan telah sesuai dengan yang ditentukan maka *port* komunikasi yang diinginkan akan terbuka. Setelah terbuka maka dapat dengan bebas mengakses apa yang ada di dalam jaringan tersebut melalui *port* komunikasi yang baru terbuka. Setelah selesai melakukan pekerjaan dan kepentingan maka *port* komunikasi yang sebelumnya terbuka dapat ditutup kembali dengan melakukan pengetukan sekuensialnya sekali lagi (Fatoni, Windu Farhan and Mustika 2022).

Pada penelitian yang dilakukan oleh Nursalim Yunus menerapkan teknik *port knocking* untuk memberikan tingkat keamanan pada *server* yang dikelola, dengan memberikan tingkat keamanan menggunakan metode *port knocking* pada *server* maka diperlukan suatu metode autentikasi yang mengizinkan *admin* sebagai *pemilik* akses sah untuk melakukan koneksi ke *server* dan mengakses *server* walaupun akses ke semua *port* di *server* ditutup (Nursalim Yunus. 2013).

Pada penelitian lainnya melakukan perancangan Algoritma Anggi (AA) untuk memberikan tingkat keamanan tambahan pada *port knocking*. Algoritma AA dapat digunakan untuk melakukan proses enkripsi dan dekripsi *keyport*. Hasil ujicoba diperoleh jika *keyport* tidak sesuai, maka terdapat peringatan bahwa proses *knocking* gagal dilakukan kemudian pada *logging* akan mendapatkan pesan “menerima *bad password*”, jika proses berhasil maka *server* akan menerima akses dari *client* yang melakukan proses *knocking* (Suhendar, Sajati, and Astuti 2013).

Pada penelitian yang dilakukan oleh S. Khadafi mengimplementasikan sistem keamanan komputer FTP *server* yang dapat digunakan untuk *unauthorized access* menggunakan rules *firewall* yang dikombinasikan dengan metode *port knocking*. *Port* FTP akan tetap berjalan dan terbuka meskipun tidak digunakan, hal ini dapat dijadikan target oleh peretas untuk melakukan *scanning port* untuk mengetahui *port* yang terbuka kemudian melakukan *sniffing* untuk mengetahui *username* dan *password*. Berdasarkan permasalahan tersebut setelah melakukan pengujian maka *the rule-set firewall* yang telah diimplementasikan pada FTP *server* membuat peretas tidak dapat mengetahui *port* mana saja yang dalam kondisi terbuka. Hasil lain yang didapatkan yaitu dengan menerapkan *port knocking* maka dapat melindungi akses pada FTP *server* (Khadafi, Nurmuslimah, and Anggakusuma 2019).

Pada penelitian yang dilakukan oleh R.Ernawati yaitu melakukan implementasikan *port knocking* pada *server* ubuntu virtual berbasis web monitoring. Berdasarkan hasil penelitian yang dilakukan dengan menerapkan metode *port knocking* maka dapat mengamankan *server* dengan memblok *port 22* dan melakukan *knock* pada *server* untuk membuka *port 22* jadi pada waktu dilakukan serangan *port scanning port 22* terlihat dalam keadaan tertutup. Dari hasil penelitian yang telah dilakukan maka dengan metode *port knocking* juga dapat mampu mengamankan *server* dari serangan DDOS *attack* dan brute force (Ernawati, Ruslianto, and Bahri 2022).

Pada penelitian ini dilakukan oleh Y.Inoue yaitu *Empowering Resource-Constraint IoT Gateways with Port Knocking Security* membahas membahas

tentang penggunaan metode keamanan yang disebut "*Port Knocking*" pada perangkat IoT *gateway* dengan sumber daya terbatas. Penelitian ini bertujuan untuk menjaga keamanan perangkat IoT serta mengurangi konsumsi daya yang digunakan. Metode yang digunakan berbasis skrip Python yang menggunakan algoritma *pseudorandom number generator* (PRNG) dan *chaotic random number generator* (CRNG), serta metode berbasis *stream cipher* yang menggunakan algoritma *authenticated encryption with associated data* (AEAD). Pengujian dilakukan pada perangkat IoT *gateway* dengan sumber daya terbatas. Hasil pengujian menunjukkan bahwa metode *port knocking* berbasis *stream cipher* efektif dalam menyembunyikan *port* SSH dan mengurangi jumlah percobaan *login* SSH yang tidak sah. Selain itu, metode ini juga mengurangi konsumsi daya perangkat IoT. Dalam pengujian yang dilakukan, perangkat dengan *port knocking* mengalami nol percobaan *login* SSH, sementara perangkat tanpa *port knocking* mengalami 1039 percobaan *login* SSH. Konsumsi daya perangkat dengan *port knocking* juga lebih rendah dibandingkan dengan perangkat tanpa *port knocking* karena jumlah paket yang diterima lebih sedikit (Yuta Inoue, Seiya Kato, Aamir 2020).

Pada penelitian yang dilakukan oleh Alaa Kamel Zidan yaitu *Enhanced User Authentication Based on Dynamic Port Knocking Technique* membahas teknik autentikasi pengguna yang ditingkatkan menggunakan teknik *Dynamic Port Knocking*. Teknik ini digunakan sebagai lapisan keamanan tambahan selain teknik autentikasi lainnya. Dalam penelitian ini, menggunakan algoritma AES. Algoritma ini menggunakan ID klien dan seed tambahan (misalnya nama layanan) untuk menghasilkan urutan *knocking*. Urutan ini dibagi menjadi blok-blok *knocking*, di

mana setiap blok memiliki jumlah *knocking* yang bervariasi dengan nomor *port* yang berbeda untuk setiap blok. Selain itu, penelitian ini juga mencakup kerja sama antara *firewall* dan *Intrusion Prevention System (IPS)* dalam sistem *port knocking*. *Firewall* berperan dalam mengontrol lalu lintas yang melewati dan menerapkan aturan yang sesuai berdasarkan urutan *knocking* yang benar. IPS digunakan sebagai bagian pendukung untuk mendeteksi dan mencegah serangan.(Major, Buchanan, and Ahmad 2020).

Pada penelitian yang dilakukan oleh S.Kato yaitu *Empirical Analysis of Security and Power-Saving Features of Port Knocking Technique Applied to an IoT Device* membahas tentang pentingnya menjaga keamanan perangkat IoT (Internet of Things) yang sering menjadi target serangan oleh agen yang tidak bertanggung jawab. Salah satu metode yang dikaji dalam jurnal ini adalah teknik "*Port Knocking*" yang dapat digunakan untuk mengamankan perangkat IoT yang memiliki keterbatasan sumber daya. Hasil eksperimen menunjukkan bahwa *Port Knocking* dapat memberikan keamanan tambahan tanpa mempengaruhi secara negatif sumber daya perangkat IoT. Penggunaan fitur *Port Knocking* menggunakan algoritma stream cipher dengan metode AEAD (Authenticated Encryption with Associated Data) menghasilkan overhead konsumsi daya CPU maksimum sebesar 15%, sedangkan penggunaan algoritma PRNG-CRNG menghasilkan overhead sebesar 50% pada konsumsi daya CPU.Selain itu, *Port Knocking* juga terbukti efektif dalam memblokir akses yang tidak diinginkan pada layanan yang dilindungi. Percobaan menunjukkan bahwa perangkat IoT dengan fitur *Port Knocking* dapat berhasil memblokir semua akses yang tidak diinginkan selama 42 hari, sementara

layanan yang tidak dilindungi menerima 431.142 permintaan dari 5.424 host selama periode tersebut (Aamir Bokhari, Yuta Inoue, Seiya Kato 2021).

Pada penelitian yang dilakukan oleh T. Popeea yaitu *Extension of a port knocking client-server architecture with NTP synchronization* membahas tentang komunikasi melalui *port* yang tertutup dapat dilakukan melalui *log firewall* yang mencatat semua upaya koneksi. Inisiasi komunikasi dilakukan oleh *client* dengan mengirimkan paket SYN ke *port* yang ditentukan dalam *knock*. Selama fase *knocking* ini, *server* tidak memberikan respons kepada *client* karena sedang "diam-diam" memproses urutan *port*. Ketika *server* berhasil mendekode *knock* yang valid, proses sisi *server* akan diaktifkan. Namun, mekanisme keamanan ini rentan terhadap serangan *brute force*, *eavesdropping*, dan serangan *replay* atau *man in the middle*. Untuk mengurangi kerentanan ini, penulis menggunakan sinkronisasi dan kriptografi untuk menghasilkan urutan *knock* yang unik dengan masa berlaku terbatas. Sinkronisasi waktu antara *client* dan *server* dilakukan menggunakan *Network Time Protocol (NTP)*. Selain itu, fungsi hash digunakan untuk menghasilkan urutan *knock* berdasarkan kunci yang telah dibagikan sebelumnya, waktu saat itu, alamat IP *client*, dan *port* tujuan (Popeea, Traian, Vladimir Olteanu, Laura Gheorghe, and Răzvan Rughiniș 2011).

Pada penelitian lain yang membahas mengenai *cryptography port knocking* dilakukan oleh Ms Pratiksha R yaitu *A Modified Hybrid Port Knocking Technique for Host Authentication: A Review* membahas tentang enkripsi yang digunakan pada teknik *hybrid port knocking (MHPK)* melibatkan empat konsep, yaitu *port knocking*, enkripsi/dekripsi kunci simetris, steganografi, dan *mutual authentication*.

Dalam MHPK, *knock sequence* yang dikirim oleh *client* dienkripsi menggunakan kunci simetris sebelum dikirimkan ke *server* yang digunakan untuk enkripsi dan dekripsi *knock sequence*. Dengan menggunakan enkripsi ini, *knock sequence* yang dikirimkan tidak terbaca oleh pihak yang tidak berwenang. Selain itu, steganografi juga digunakan dalam MHPK. Steganografi adalah teknik yang digunakan untuk menyembunyikan informasi dalam media yang tampaknya tidak memiliki informasi tambahan. *Mutual authentication* digunakan dalam MHPK untuk memastikan bahwa *client* dan *server* saling mengenali dan memverifikasi identitas masing-masing. Setelah *knock sequence* yang dienkripsi diterima oleh *server*, autentikasi saling digunakan untuk memverifikasi keaslian *client* dan memastikan bahwa *client* memiliki hak akses ke layanan yang dilindungi (Yewale 2014).

Kriptografi merupakan ilmu menjaga kerahasiaan pesan dengan menggunakan metode menyandikan dalam bentuk yang tidak dapat dipahami. *Kriptografi* mempunyai dua proses yaitu *enkripsi* dan *dekripsi* (Azhari, Muhammad, Dadang Iskandar Mulyana, Faizal Joko Perwitosari, and Firhan Ali 2022). Sebuah pesan asli yang disebut sebagai *plaintext* disandikan menjadi pesan yang tersandi yang disebut sebagai *ciphertext* melalui proses *enkripsi* dan *ciphertext* dipulihkan menjadi *plaintext* kembali melalui proses *dekripsi*. *Kriptografi* memiliki beragam algoritma yang telah banyak digunakan sebagai keamanan untuk informasi.(Yusfrizal 2019).

Terdapat beberapa istilah-istilah penting dalam *kriptografi* yang perlu diketahui:

- 1) Pesan (*Plaintext* dan *Ciphertext*): Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Pesan asli disebut plainteks

(*plaintext*) atau teks-jelas (*cleartext*). Sedangkan pesan yang sudah disandikan disebut cipherteks (*chipertext*) (Sitinjak and Fauziah 2010).

- 2) Pengirim dan Penerima: Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan (Sitinjak and Fauziah 2010).
- 3) Penyadap (*eavesdropper*) adalah orang yang mencoba menangkap pesan selama ditransmisikan (Sitinjak and Fauziah 2010).
- 4) Kriptanalisis dan Kriptologi: Kriptanalisis (*cryptanalysis*) adalah ilmu dan seni untuk memecahkan chiperteks menjadi plainteks tanpa mengetahui kunci yang digunakan. Pelakunya disebut kriptanalis. Kriptologi (*cryptography*) adalah studi mengenai kriptografi dan kriptanalisis (Sitinjak and Fauziah 2010).
- 5) *Enkripsi* dan *Dekripsi*: Proses menyandikan plainteks menjadi cipherteks disebut enkripsi (*encryption*) atau enciphering. Sedangkan proses mengembalikan cipherteks menjadi plainteks semula dinamakan dekripsi (*decryption*) atau deciphering (Sitinjak and Fauziah 2010).
- 6) Cipher dan Kunci: Algoritma *kriptografi* disebut juga *cipher* yaitu aturan untuk *enchipering* dan *dechipering*, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Kunci (*key*) adalah parameter yang digunakan untuk transformasi *enciphering* dan *dechipering*. Kunci biasanya berupa string atau deretan bilangan (Sitinjak and Fauziah 2010).

Ada empat tujuan mendasar dari ilmu *kriptografi* yang juga aspek keamanan informasi, yaitu:

- 1) *Confidentiality*, adalah layanan yang digunakan untuk menjaga kerahasiaan isi informasi dari pihak yang tidak bersangkutan atau kunci rahasia untuk membuka informasi yang telah dikodekan (Putri, Fitria Nova Hulu and Maharani 2019).
- 2) Integritas data, adalah menjaga pelestarian perubahan data yang tidak sah. Dalam menjaga integritas data, sistem wajib memiliki kemampuan untuk mendeteksi manipulasi data dari pihak yang tidak memiliki hak atas penyisipan, penghapusan, dan pemasukan data lainnya kedalam data yang sebenarnya (Putri, Fitria Nova Hulu and Maharani 2019).
- 3) Otentikasi, berkaitan dengan identifikasi atau pengenalan, baik sebagai keseluruhan sistem atau informasi itu sendiri. Dua pihak yang saling berkomunikasi wajib memperkenalkan diri. Informasi yang dikirim melalui kanal wajib diotentikasi keaslian isi data, waktu pengiriman, dan lain-lain (Putri, Fitria Nova Hulu and Maharani 2019)
- 4) *Non-repudiation* adalah sebuah proses yang dilakukan untuk mencegah penolakan penyampaian atau pembuatan suatu informasi oleh pihak – pihak yang mengirim atau membuat informasi tersebut (Putri, Fitria Nova Hulu and Maharani 2019).

2.11 Algoritma Simetri

Pada sistem *kriptografi* terdapat kunci-simetri karena memiliki kunci yang sama untuk kunci *enkripsi* dan kunci *deskripsi* (Sitinjak and Fauziah 2010). Algoritma ini memiliki kunci yang bersifat rahasia atau *private* sehingga algoritma ini disebut juga sebagai algoritma kunci rahasia (Halik and Prayudi 2005).

Keamanan algoritma simetris tergantung pada kunci, membocorkan kunci berarti menjadikan orang lain dapat mengenkripsi dan mendekripsi pesan. (Massandy 2009).

2.11.1 Algoritma TEA

David Wheeler dan Roger Needham dari Komputer Laboratory, Cambridge University, England pada bulan November tahun 1994 menciptakan algoritma sandi yaitu Algoritma TEA (*Tiny Encryption Algorithm*). Algoritma TEA adalah algoritma penyandian *block cipher* yang dibuat untuk penggunaan memori yang seminimal mungkin dengan kecepatan proses yang maksimal. Untuk tujuan pada algoritma TEA yaitu untuk menghemat memori sambil meningkatkan kecepatan (Yee Hunn, Siti, and Binti Idris 2012).

TEA memproses 64-bit input sekali waktu dan menghasilkan 64-bit output. TEA menyimpan 64-bit input kedalam L0 dan R0 masing masing 32-bit, sedangkan 128-bit kunci disimpan kedalam k[0], k[1], k[2], dan k[3] yang masing masing berisi 32-bit. Diharapkan teknik ini cukup dapat mencegah penggunaan *teknik exshautive search* secara efektif. Hasil outputnya akan disimpan dalam L16 dan R16 (Qamal 2014).

2.11.2 Algoritma XTEA

Algoritma XTEA (*Extended Tiny Encryption Algoritma*) merupakan salah satu algoritma enkripsi data sehingga data asli hanya dapat dibaca oleh seseorang yang memiliki kunci *enkripsi* tersebut. Adapun pada pengembangan algoritma XTEA dengan algoritma TEA yaitu menggunakan key yang lebih kompleks dan pengaturan urutan dari operasi shift, XOR dan penambahan.(Pandiangan 2020)

Untuk dapat melakukan enkripsi XTEA menggunakan data 64-bit dan untuk melakukan *deskripsi* dengan kunci simetris 128-bit, pada algoritma ini juga dikenal sebagai *block cipher* (Anusha and Veena Devi Shastrimath 2021). Dari 128 bit dibagi menjadi 4 blok masing-masing blok 32 bit, seperti berikut: (Sinaga, Sinurat, and Zebua 2021)

- Kunci [0] = Kunci blok 1
- Kunci [1] = Kunci blok 2
- Kunci [2] = Kunci blok 3
- Kunci [3] = Kunci blok 4

Bentuk jaringan feistel masih sama, yang membedakan adalah fungsi feistel dan penjadwalan kunci yang digunakan yaitu pada algoritma XTEA, ronde ganjil digunakan $K [\text{sum AND } 3]$, sedangkan pada ronde genap digunakan $K [\text{sum} \gg 11 \text{ AND } 3]$. Adapun setiap penjadwalan kunci untuk setiap ronde pada putaran *enkripsi* tetap menggunakan nilai Delta 9E3779B9(16). Berikut rumus dalam menentukan jadwal kunci untuk setiap ronde pada satu putaran *enkripsi* dan dekripsi XTEA (Sinaga, Sinurat, and Zebua 2021):

Ronde ganjil menggunakan sub kunci dengan rumus:

Kunci $[\text{sum} + \text{Delta AND } 3]$

Ronde genap menggunakan sub kunci dengan rumus:

Kunci $[\text{sum} + \text{Delta} \gg \text{AND } 3]$

Keterangan:

Sum = Jumlah putaran enkripsi.

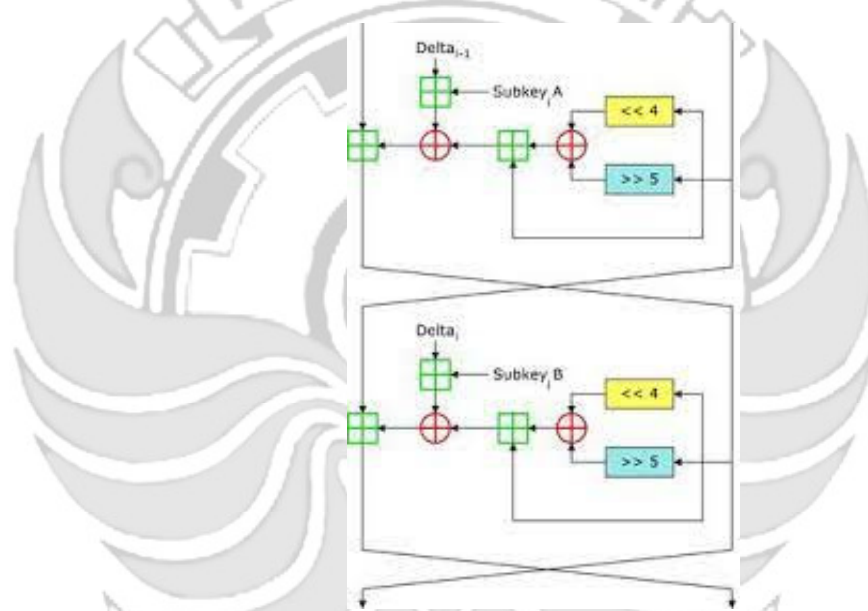
Delta = Nilai konstan dalam algoritma

XTEA yaitu 9E3779B9(16).

Delta >> = Nilai Delta yang dirubah kedalam biner dan dilakukan pergeseran 11bit biner delta kekanan.

AND = operator

Adapun satu putaran algoritma XTEA memiliki 2 ronde enkripsi yang dapat dilihat pada gambar 1 di bawah ini:



Gambar 2. 1 Satu putaran enkripsi dalam Jaringan Feistel

Sumber : (Sinaga, Sinurat, and Zebua 2021)

Pada penelitian yang dilakukan oleh Suhendra menerapkan teknik kriptografi moderen menggunakan algoritma XTEA dengan metode pembangkitan kunci acak Linear Congruential Generator (LCG) agar dapat mengamankan *file* dokumen rahasia yang belum di sandikan. *File* dokumen yang diamankan menghasilkan *chiperdokumen* yang tidak dapat dipahami maknanya jika *file*

tersebut dibuka, maka hasil dari modifikasi kunci XTEA dengan teknik LCG berhasil mengamankan *file chiperdokument* dari kebocoran (Sinaga, Sinurat, and Zebua 2021).

Pada penelitian lainnya menerapkan teknik kriptografi dengan algoritma yang sama yakni algoritma XTEA untuk aplikasi pengamanan data email. Berdasarkan hasil pengujian, format *file* akan berubah menjadi format.sc. dan untuk ukuran *file* meningkat menjadi ukuran *file* yang lebih besar setelah melalui proses enkripsi dengan rata-rata ukuran *file* yang telah melalui proses encrypt meningkat sekitar 22,221% dari ukuran *file* asli sebelum melalui proses encrypt. Menurut hasil uji coba format *file* .sc akan berubah ke ukuran *file* asli sebelum dienkripsi ukuran *file* akan berubah ke ukuran *file* asli sebelum dienkripsi. Perubahan ukuran *file* rata-rata akan meningkat sebesar 77,779% (penurunan 22,221%) sebagai rata-rata proses enkripsi. Dari hasil pengujian maka data *dapat* terlindungi dengan baik dengan algoritma kriptografi Data XTEA tidak dapat dibuka oleh orang yang tidak berkepentingan tidak diizinkan memiliki kunci untuk mendekripsi *file* (Anif, Siswanto, and Prasetyo, Basuki Hari 2020).

Pada penelitian yang dilakukan oleh Pandiangan menerapkan kriptografi dengan algoritma XTEA untuk melakukan pengamanan data *file* dokumen teks. Aspek kerahasiaan pada pengamanan data *file* dokumen terletak pada pengamanan data yaitu *password* maka aplikasi pengamanan *file* dokumen ini dapat melakukan enkripsi dan dekripsi *password* (Pandiangan 2020).

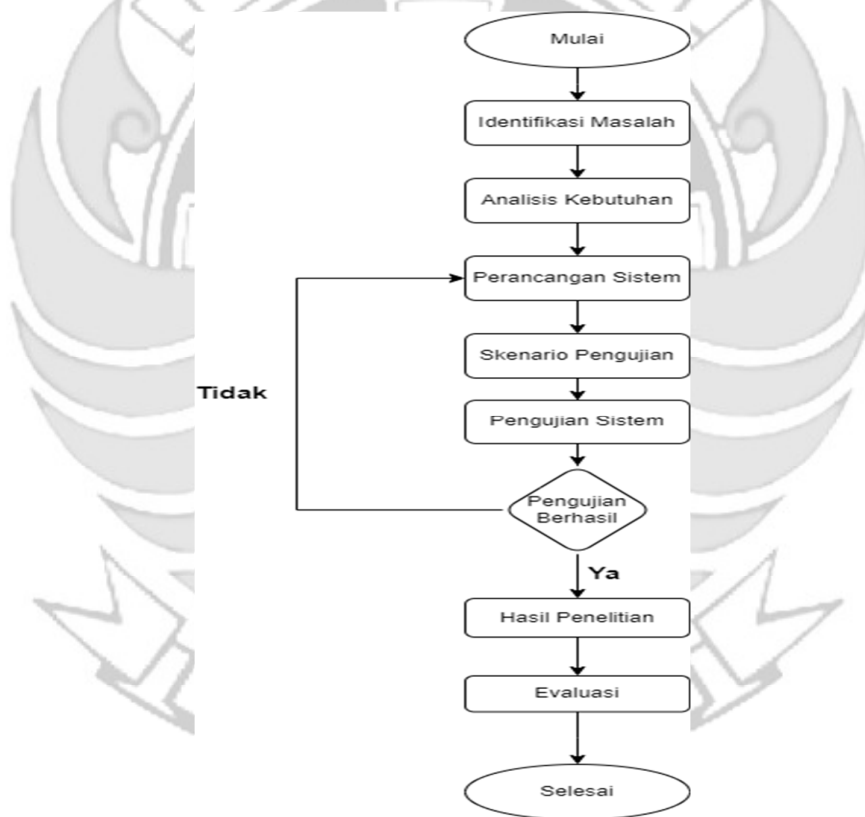
BAB III METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Pelaksanaan penelitian ini bertempat di Laboratorium Internet dan Mobile, Politeknik Negeri Ujung Pandang, Jl. Perintis Kemerdekaan KM 10 Kota Makassar. Dimulai pada bulan Januari 2023 sampai dengan Juni 2023.

3.2 Prosedur Penelitian

Prosedur penelitian diperlukan agar setiap proses yang dilakukan dapat terstruktur sehingga hasil yang diperoleh sesuai dengan tujuan pada penelitian. Adapun tahap prosedur penelitian seperti pada gambar 3.1



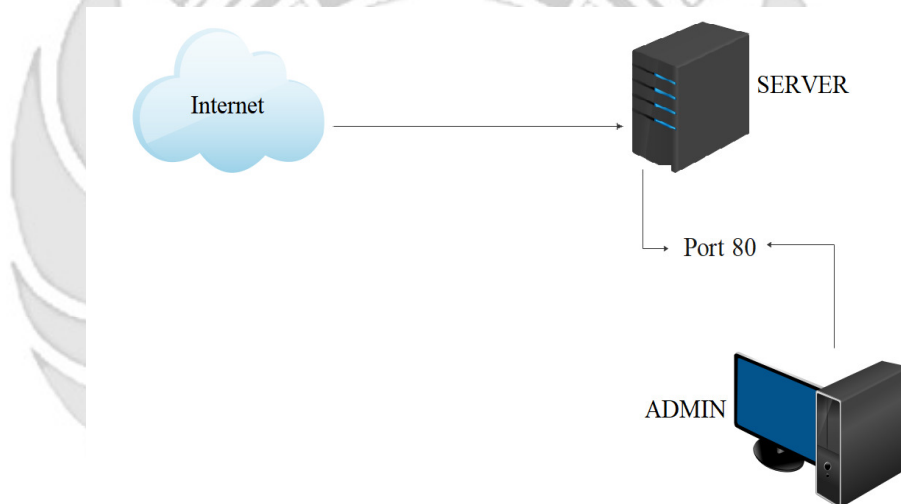
Gambar 3. 1 Diagram Alir Prosedur Perancangan

3.2.1 Identifikasi Masalah

Pada tahap ini dilakukan identifikasi masalah sehingga dapat menentukan cakupan sistem yang akan dibuat. Permasalahan yang ada adalah keamanan pada suatu *server* saat melakukan *knocking* terdapat kemungkinan seseorang melakukan proses penyadapan maka perlu adanya tingkat keamanan tambahan di model *port knocking* yaitu dengan melakukan proses enkripsi pada *sequence port*.

a. *Admin Mengakses Port Keadaan Normal*

Seorang *admin* mengakses *port 80* pada *server* secara langsung tanpa melakukan *remote server*, dapat dilihat pada gambar 3.2

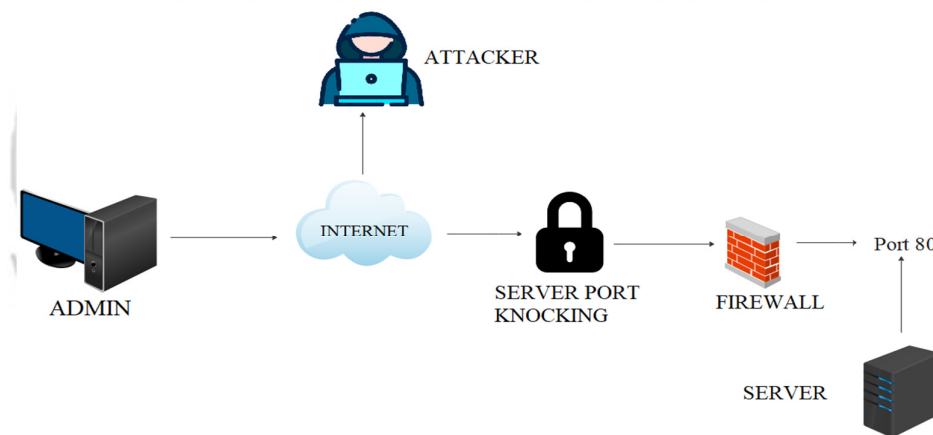


Gambar 3. 2 *Admin Mengakses Port Keadaan Normal*

Kegiatan rutin yang dilakukan oleh seorang *admin* jaringan yaitu mengakses *server* secara langsung, misalnya jika ingin mengakses *port 80* yang terdapat pada *server* maka seorang *admin* dapat mengakses *port* tersebut langsung pada *server*. Terdapat suatu kondisi seorang *admin* sedang diberikan tugas keluar kota akan tetapi *admin* tetap diharuskan mengakses *server* sehingga agar tetap dapat mengakses *server* seorang *admin* melakukan dengan cara *via remote*.

b. Penerapan *Port knocking* pada *Server*

Jika terdapat kondisi yang mengharuskan seorang *admin* melakukan *remote server* misalnya pada *port 80* maka *admin* menerapkan metode *port knocking*. Metode *port knocking* dapat digunakan untuk menjaga semua *port* yang ditutup sampai pengguna melakukan autentikasi dengan *knock port*. Jika urutan ketukan benar maka *server* memberikan ijin kepada *client* untuk dapat mengakses *port* tersebut, tetapi apabila urutan salah maka *client* tidak dapat mengakses *port* tersebut, dapat dilihat pada gambar 3.3.



Gambar 3. 3 Penerapan *Port knocking* Pada *Server*

Saat seorang *admin* melakukan *remote server* dengan menerapkan metode *port knocking* hal ini masih terdapat kelemahan karena *sequence port* yang sedang di *remote* oleh *admin* masih berbentuk *plaintext* sehingga jika terdapat kemungkinan terjadi penyadapan maka *attacker* dapat dengan mudah memahami *sequence port* yang sedang di *remote* oleh *admin*.

3.2.2 Analisis Kebutuhan

Dalam membangun sistem enkripsi pada *port knocking* dibutuhkan analisis kebutuhan. Pada tahap ini akan dilakukan analisis terhadap kebutuhan-kebutuhan sistem, kebutuhan perangkat keras (*Hardware*) dan perangkat lunak (*Software*). Analisis ini bertujuan untuk mengetahui sistem seperti apa yang akan diterapkan, serta kebutuhan perangkat keras dan lunak yang sesuai pada pembuatan sistem

a. Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan dalam membangun sebuah sistem “Penerapan Algoritma XTEA Pada *Port knocking* Untuk Peningkatan Keamanan Jaringan”. Adapun kebutuhan perangkat yang dibutuhkan sebagai berikut:

1. Prosesor Corei7/ AMD
2. RAM 8Gb (Gigabyte)
3. Harddisk 1Tb (Terrabyte)

b. Perangkat Lunak (*Software*)

Perangkat lunak yang dibutuhkan dalam membangun sistem Sebagai berikut:

1. Vmware
2. Ubuntu *Server*
3. Ubuntu Dekstop
4. Kali Linux
5. Namap
6. *Wireshark*
7. *Knockd*
8. Putty

9. SSH

10. Web Server

11. FTP Server

12. Telnet

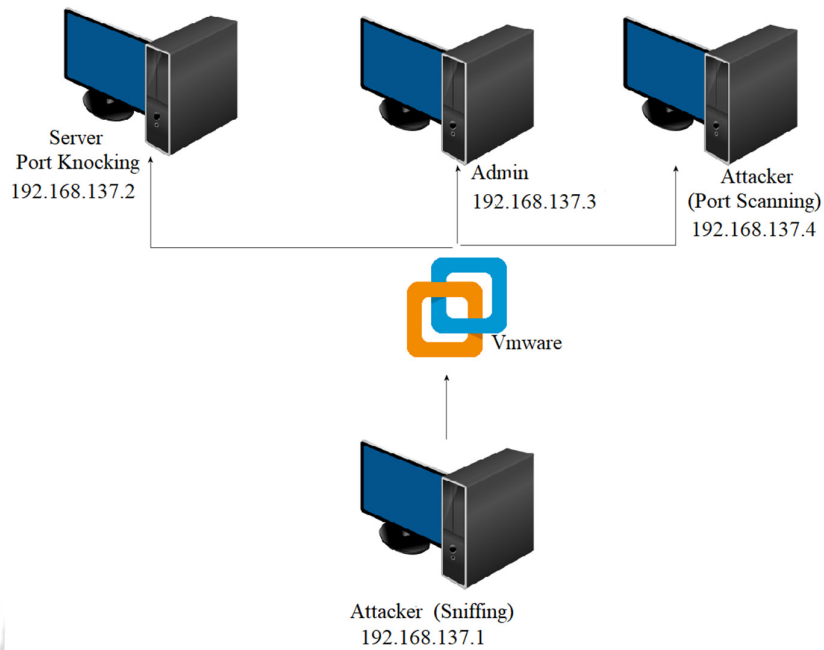
13. SMTP

3.2.3 Perancangan Sistem

Pada sistem perancangan ini dilakukan perancangan konseptual. Tujuan pembuatan sistem ini adalah pembuatan *server port knocking* dengan menerapkan algoritma XTEA untuk mengamankan sebuah *sequence port* sehingga *sequence port* tersebut tidak mudah untuk dibaca jika terdapat *attacker* melakukan penyadapan untuk. Untuk pemrograman Algoritma XTEA dalam mengamankan sebuah *sequence port* menggunakan bahasa *pyhton*. Adapun Proses utama pada sistem ini adalah melakukan enkripsi pada sebuah *sequence port* yang terbuka saat *admin* melakukan *remote* terhadap *server*. Adapun tujuan dari perancangan ini adalah untuk memberikan gambaran mengenai sistem yang akan dibuat sesuai dengan hasil analisis masalah dan analisis kebutuhan.

1. Arsitektur Sistem

Arsitektur Sistem merupakan penggambaran umum dari lalu lintas jaringan yang akan dibuat, dapat dilihat pada gambar 3.4.

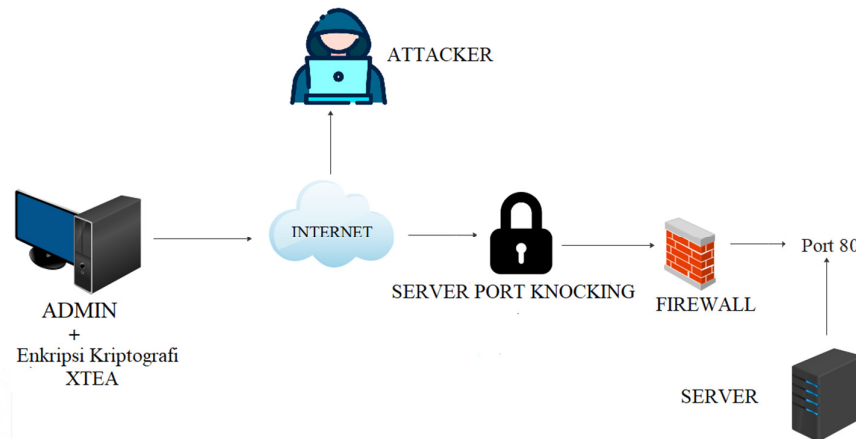


Gambar 3. 4 Arsitektur Sistem

Sebuah komputer membuat jaringan secara virtualisasi menggunakan vmware, yang dimana didalamnya terdapat satu buah *server* dengan sistem operasi ubuntu *server* yang akan digunakan untuk menjalankan sistem *port knocking*, kemudian terdapat juga komputer dengan sistem operasi ubuntu *dekstop* yang akan digunakan *admin* untuk melakukan *remote* pada *port* yang terdapat di *server* dan yang terakhir komputer sebagai *attacker* yang menggunakan sistem operasi kali linux untuk melakukan serangan dengan metode *port scanning* menggunakan *tool* nmap. Komputer yang melakukan jaringan visualisasi tersebut juga berperan sebagai *attacker* untuk melakukan serangan *sniffing* dengan menggunakan *tool* *wireshark*.

2. Arsitektur Program

Arsitektur program merupakan penggambaran umum program yang akan dibuat, dapat dilihat pada gambar 3.6.



Gambar 3. 5 Arsitektur Program

Saat seorang *admin* melakukan *remote server* dengan menerapkan metode *port knocking* hal ini masih terdapat kelemahan karena *sequence port* yang sedang di *remote* oleh *admin* masih berbentuk *plaintext* sehingga jika terdapat kemungkinan terjadi penyadapan maka *attacker* dapat dengan mudah memahami *sequence port* yang sedang di *remote* oleh *admin*. Maka dari permasalahan tersebut perlu adanya tingkat keamanan tambahan pada *port knocking* dengan melakukan *enkripsi sequence port* menggunakan algoritma XTEA.

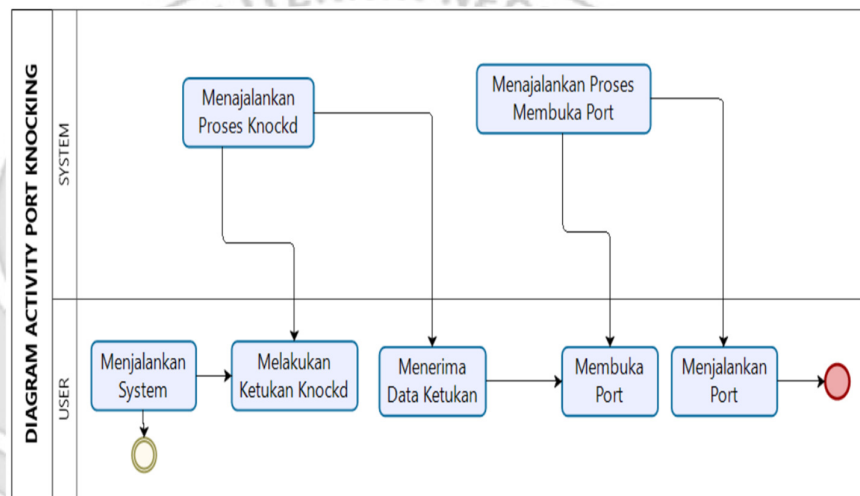
3. Activity Diagram

Activity diagram menggambar berbagai alir aktivitas dari sistem yang sedang dirancang, bagaimana masing-masing alir berawal, pengambil keputusan yang mungkin terjadi dan bagaimana mereka berakhir.

Activity diagram yang terdapat pada penerapan keamanan *port* pada *port knocking* yang dibuat ialah sebagai berikut:

a. Diagram *Port knocking*

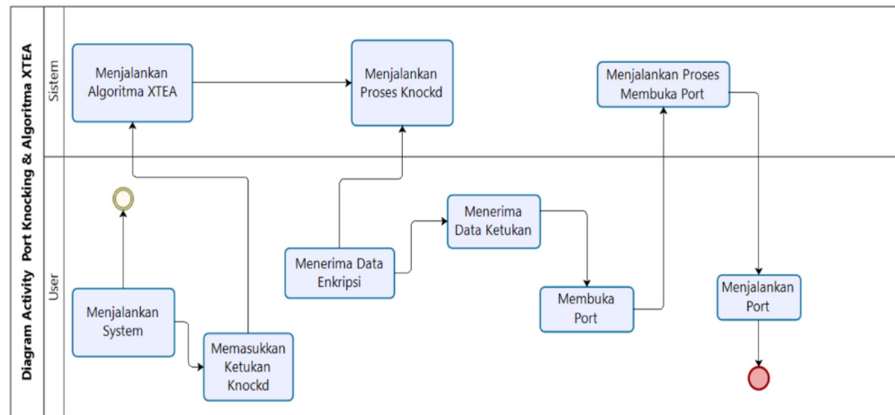
Diagram *activity port knocking* memiliki dua partis, yaitu *admin* dan sistem. Dalam sistem ini *admin* akan melakukan ketukan *port* yang akan di *remote* menggunakan *knockd*.



Gambar 3. 6 Diagram *Port Knocking*

b. Diagram *Port knocking* dan Algoritma XTEA

Diagram *activity port knocking* dan algoritma XTEA memiliki dua partis, yaitu *admin* dan sistem. Dalam sistem ini *admin* akan memasukkan *sequence* untuk membuka *port* kemudian *sequence* tersebut dilakukan enkripsi.

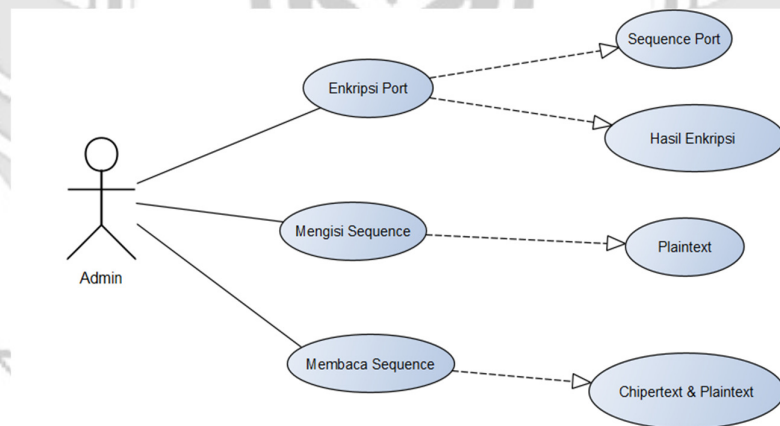


Gambar 3. 7 Diagram *Port Knocking* dan Algoritma XTEA

4. Use Case

Dalam sistem ini *user* terdapat satu *actor* yang memiliki hak dan akses pada sistem yang akan dibuat

a. Use Case *Admin*

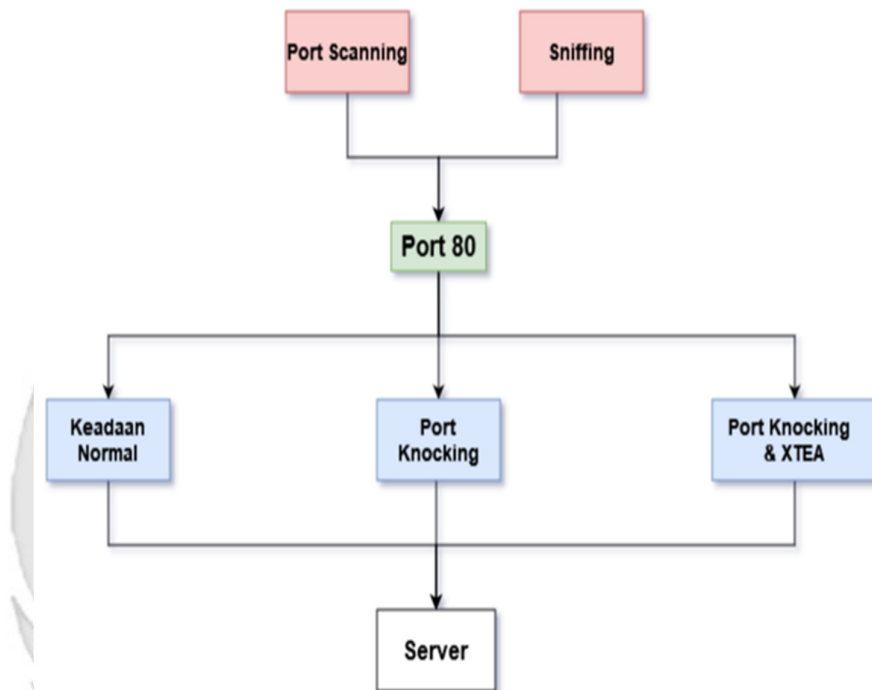


Gambar 3. 8 *Usecase Admin*

Admin dapat melakukan *enkripsi* pada *sequence port*. *Admin* dapat mengisi *port* dalam bentuk *plaintext* dan *admin* juga dapat membaca data *sequence port* dalam bentuk *plaintext* dan *chipertext*.

3.2.4 Skenario Pengujian

Skenario Pengujian merupakan penggambaran umum pengujian yang akan dibuat, dapat dilihat pada gambar 3.9



Gambar 3. 9 Skenario Pengujian

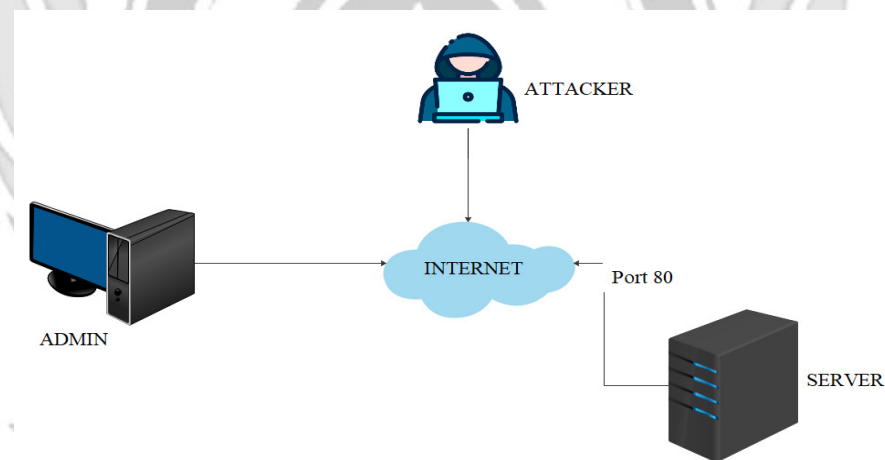
Terdapat beberapa ksebuah *server* dalam kondisi normal, *server* yang telah diterapkan metode *port knocking* dan *server* yang menerapkan metode *port knocking* serta algoritma XTEA untuk melakukan enkripsi *sequence port*. Pada masing-masing kondisi *server* tersebut telah terdapat beberapa *port* kemudian akan dilakukan pengujian pada masing-masing *server* dengan melakukan serangan *port scanning* dan serangan *sniffing*.

3.2.5 Pengujian Sistem

Implementasi pada sistem ini akan dilakukan pada *server* dengan sistem operasi Ubuntu *Server* kemudian pada *server* tersebut telah terdapat *port knocking* untuk membuka dan menutup *port* pada saat *admin* melakukan *remote server*. Kemudian akan terdapat sebuah proses yang berjalan untuk mengenkripsi *sequence port* sebelum seorang *admin* melakukan *remote* terhadap *server*.

a. Pengujian *Server* Keadaan Normal

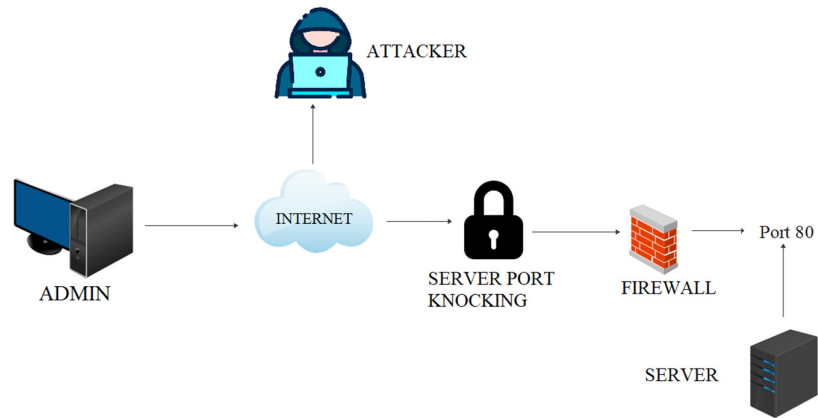
Tahap pengujian sistem yang akan dilakukan pada penelitian ini yaitu melakukan serangan *sniffing* dan *port scanning* dalam keadaan jaringan normal.



Gambar 3. 10 Pengujian Keadaan Normal

b. Pengujian Penerapan *Port knocking*

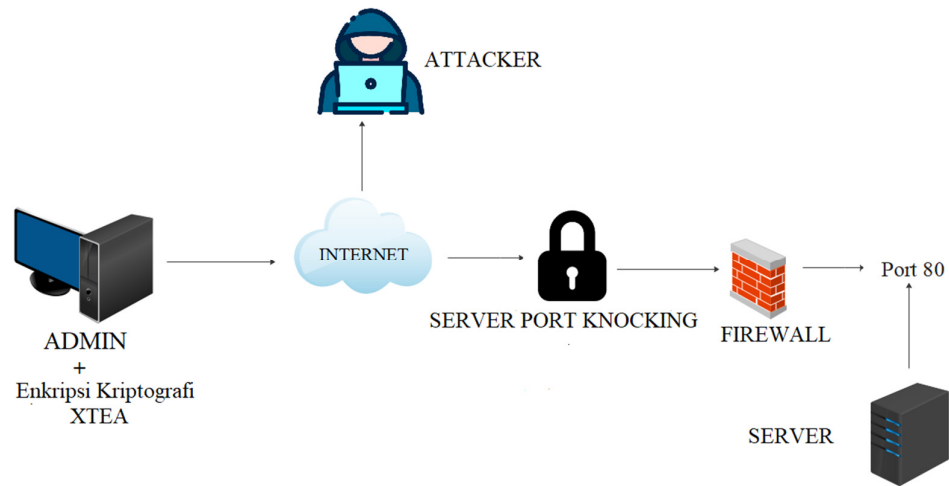
Tahap pengujian sistem yang akan dilakukan pada penelitian ini yaitu melakukan serangan *sniffing* dan *port scanning* dalam keadaan jaringan telah menerapkan *Port knocking*.



Gambar 3. 11 Pengujian Penerapan *Port knocking*

c. Pengujian Penerapan *Port knocking* dan Algoritma XTEA

Tahap pengujian selanjutnya sistem yang akan dilakukan pada penelitian ini yaitu melakukan serangan *sniffing* dan *port knocking* dalam keadaan jaringan telah menerapkan *port knocking* dan algoritma XTEA.



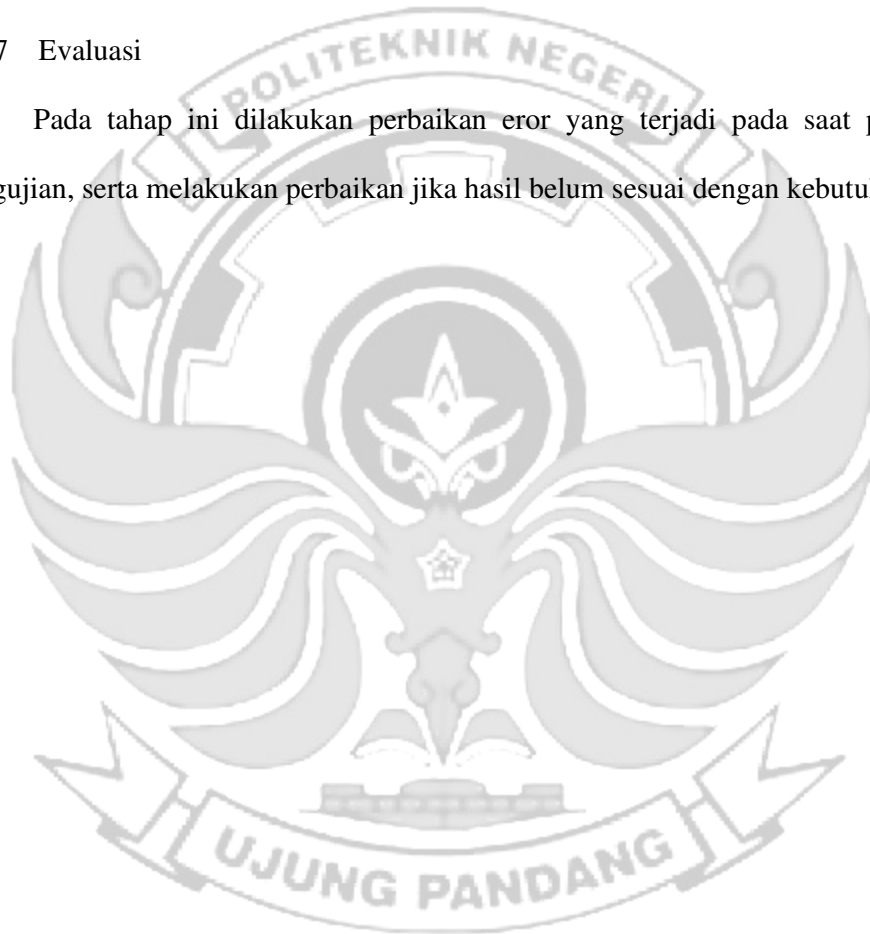
Gambar 3. 12 Pengujian Penerapan *Port knocking* dan Algoritma XTEA pada *Server*

3.2.6 Hasil Penelitian

Setelah semua tahapan perancangan sistem, implementasi dan pengujian sistem telah selesai dilakukan maka pada tahap ini dilakukan pengambilan kesimpulan berdasarkan rumusan masalah terhadap hasil yang telah dicapai dari seluruh tahapan penelitian

3.2.7 Evaluasi

Pada tahap ini dilakukan perbaikan eror yang terjadi pada saat proses pengujian, serta melakukan perbaikan jika hasil belum sesuai dengan kebutuhan.



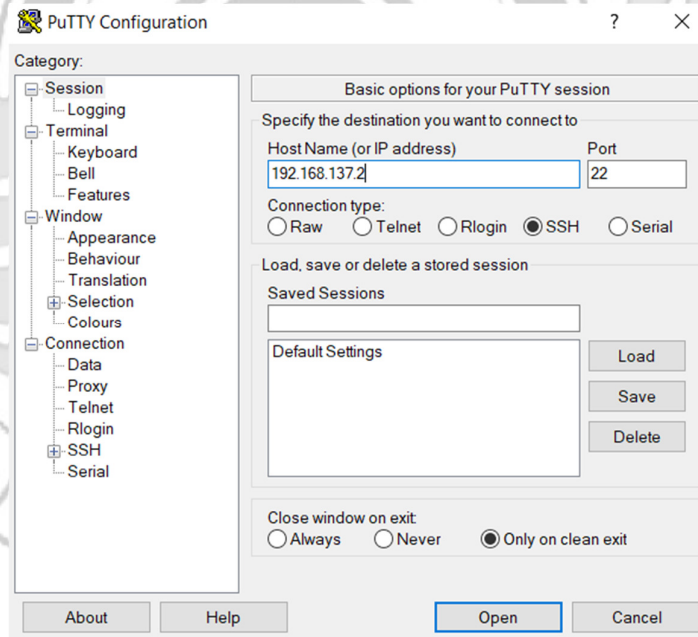
BAB IV HASIL DAN PEMBAHASAN

4.1 Pengujian Server

4.1.1 Pengujian Server Tidak Ada Sistem Keamanan

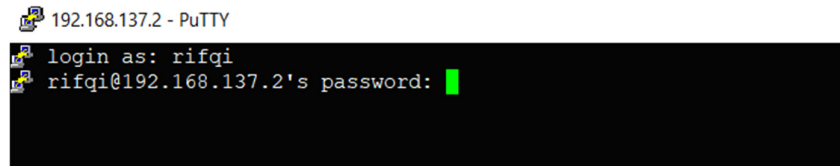
Kegiatan rutin yang dilakukan oleh seorang *admin* jaringan yaitu mengakses *server* secara langsung, pada penelitian ini terdapat beberapa *port* yang dapat diakses oleh seorang *admin* yaitu SSH, TELNET, HTTP, FTP dan SMTP.

- a. Langkah Uji Coba melakukan *remote* pada *port* 22 (SSH)
 - 1) Menjalankan aplikasi putty dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 1 Konfigurasi Putty

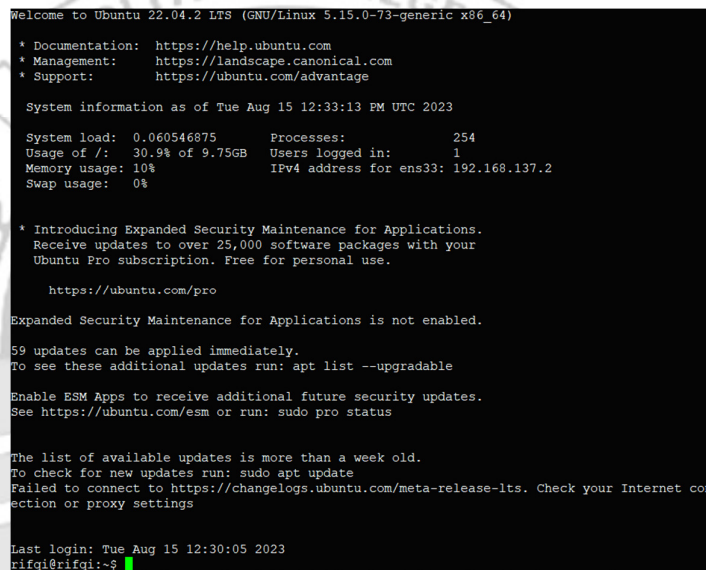
2) Tampilan ketika proses *login*, memasukkan *username* dan password.



```
192.168.137.2 - PuTTY
login as: rifqi
rifqi@192.168.137.2's password: █
```

Gambar 4. 2 Proses *Login* Putty

3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2



```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875   Processes:    254
Usage of /:   30.9% of 9.75GB Users logged in:  1
Memory usage: 10%         IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 3 Proses *Login* Telah Berhasil Dilakukan

4) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).



```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi# █
```

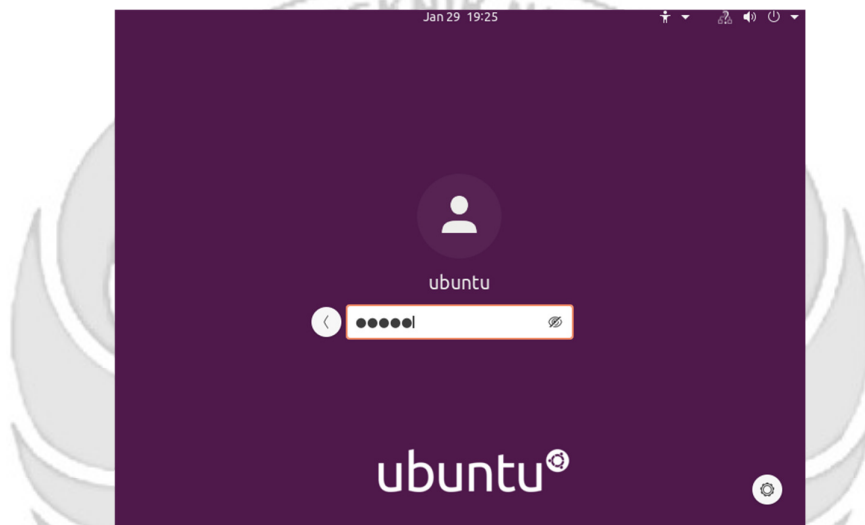
Gambar 4. 4 Masuk ke Super *User*

5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses ping ke *user* dengan alamat IP 192.168.137.3 dan proses ping berhasil

```
root@rifqi: /home/rifqi
root@rifqi:/home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data.
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

Gambar 4. 5 Proses PING

- 6) Tampilan ketika proses login pada admin, memasukkan username dan password.



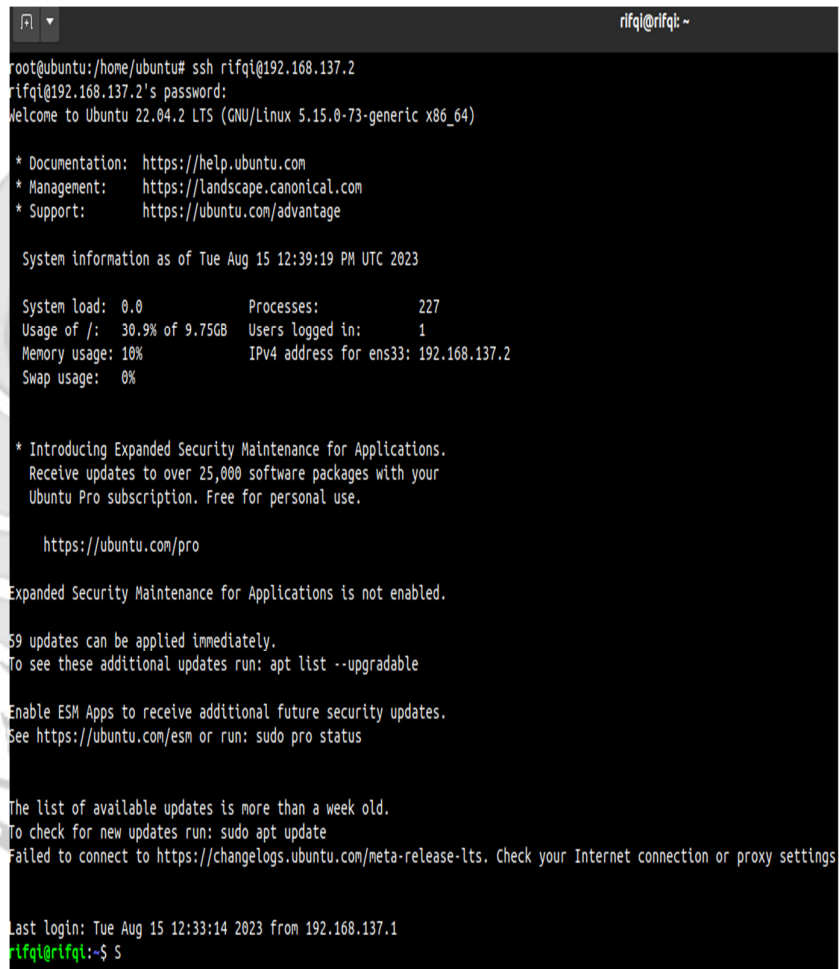
Gambar 4. 6 Proses Login Admin

- 7) Selanjutnya menjalankan admin untuk melakukan remote pada port 22 (SSH), tampilan ketika masuk ke level user yang lebih tinggi (super user).

```
root@ubuntu: /home/ubuntu
ubuntu@ubuntu:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu: /home/ubuntu# S
```

Gambar 4. 7 Mask Super User

- 8) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port 22* (SSH). Membuktikan bahwa *server* dapat dengan mudah diakses oleh orang yang tidak berhak karena pada *server* tidak di tutup total dengan cara mendrop semua akses menggunakan iptables (*firewall*).



```
rifqi@rifqi: ~
root@ubuntu:/home/ubuntu# ssh rifqi@192.168.137.2
rifqi@192.168.137.2's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:39:19 PM UTC 2023

System load:  0.0          Processes:      227
Usage of /:   30.9% of 9.75GB  Users logged in:  1
Memory usage: 10%         IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

69 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

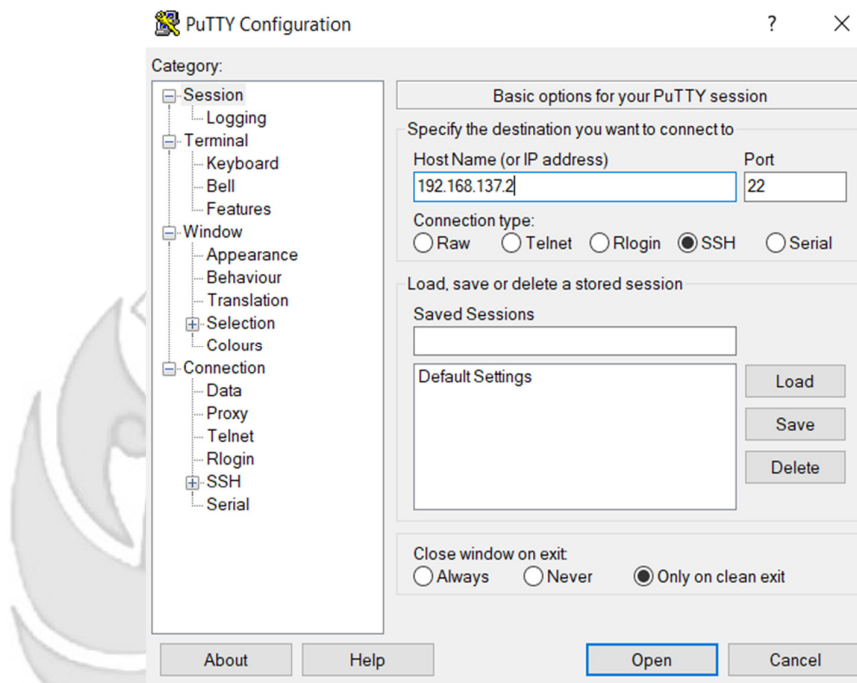
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:33:14 2023 from 192.168.137.1
rifqi@rifqi:~$
```

Gambar 4. 8 SSH Berhasil Diakses

b. Langkah Uji Coba melakukan *remote* pada *port* 23 (TELNET)

- 1) Menjalankan aplikasi putty dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 9 Konfigurasi Putty

- 2) Tampilan ketika proses *login*, memasukkan *username* dan password.



Gambar 4. 10 Proses Login Putty

- 3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875   Processes:            254
Usage of /:   30.9% of 9.75GB Users logged in:       1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 11 Proses *Login* Telah Berhasil Dilakukan

- 4) Tampilan ketika masuk ke level *user* yang lebih tinggi (super *user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

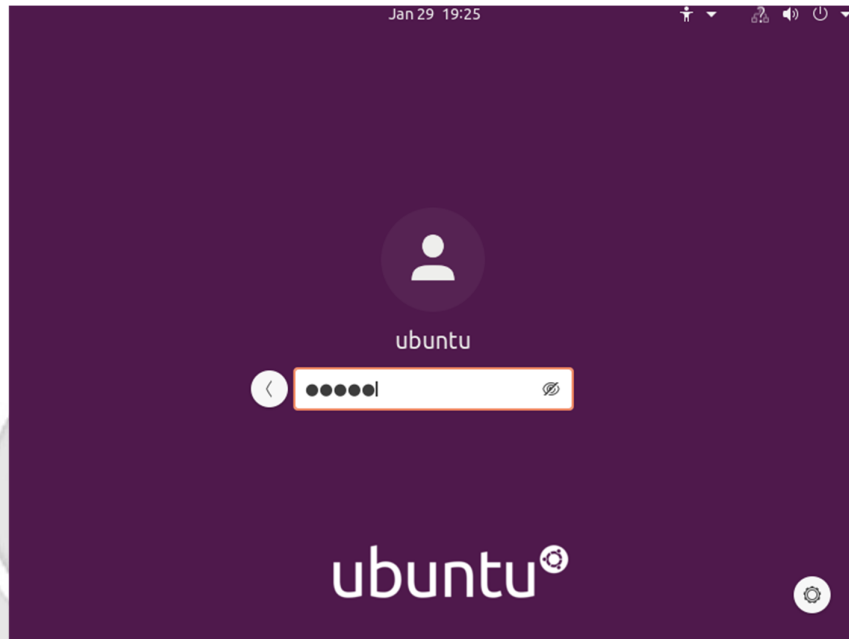
Gambar 4. 12 Masuk Super *User*

- 5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses ping ke *user* dengan alamat IP 192.168.137.3 dan proses ping berhasil.

```
root@rifqi: /home/rifqi
root@rifqi: /home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data.
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

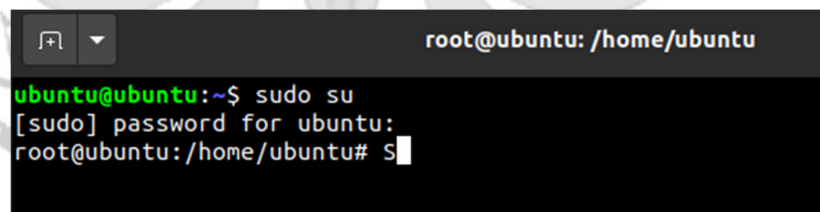
Gambar 4. 13 Proses PING

- 6) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan *password*.



Gambar 4. 14 Proses *Login Admin*

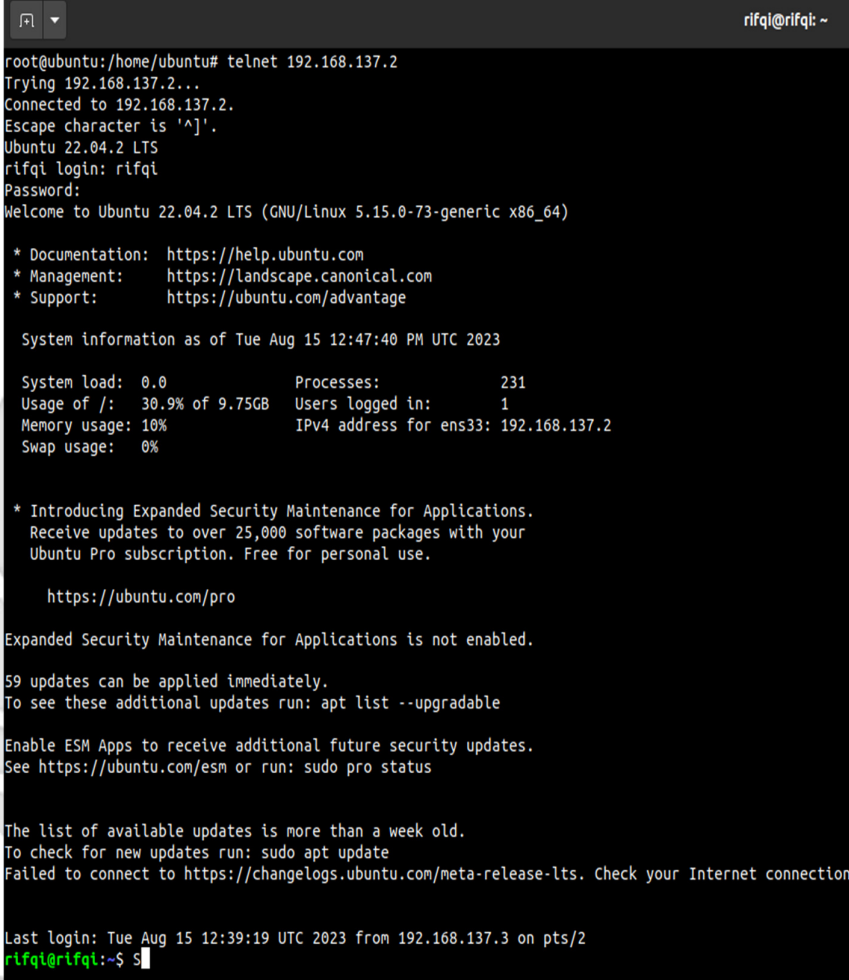
- 7) Selanjutnya menjalankan *admin* untuk melakukan *remote* pada *port 23* (TELNET), tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).



Gambar 4. 15 Masuk *Super User*

- 8) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port 23* (TELNET). Membuktikan bahwa *server* dapat dengan mudah diakses oleh orang

yang tidak berhak karena pada *server* tidak di tutup total dengan cara mendrop semua akses menggunakan iptables (*firewall*).



```
root@ubuntu:/home/ubuntu# telnet 192.168.137.2
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^]'.
Ubuntu 22.04.2 LTS
rifqi login: rifqi
Password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:47:40 PM UTC 2023

System load:  0.0          Processes:      231
Usage of /:   30.9% of 9.75GB   Users logged in:  1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

 * Introducing Expanded Security Maintenance for Applications.
 * Receive updates to over 25,000 software packages with your
 * Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

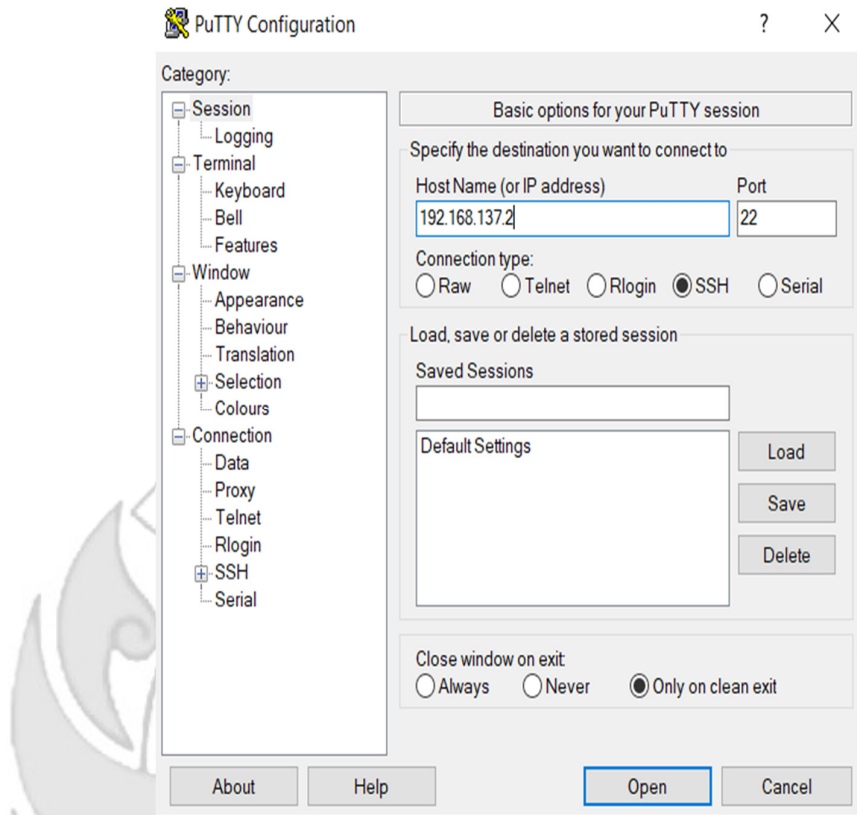
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection

Last login: Tue Aug 15 12:39:19 UTC 2023 from 192.168.137.3 on pts/2
rifqi@rifqi:~$
```

Gambar 4. 16 TELNET Berhasil DIakses

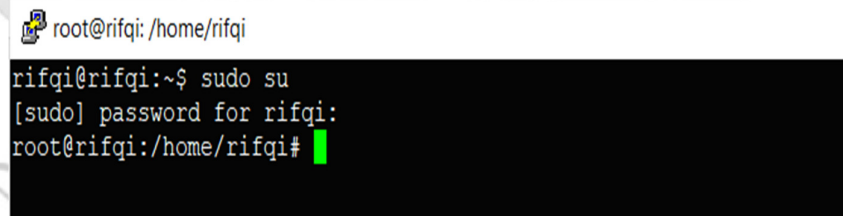
c. Langkah Uji Coba melakukan *remote* pada *port* 80 (HTTP)

- 1) Menjalankan aplikasi putty dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 17 Konfigurasi Putty

- 2) Tampilan ketika proses *login*, memasukkan *username* dan password.



Gambar 4. 18 Proses *Login* Putty

- 3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2


```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875   Processes:           254
Usage of /:   30.9% of 9.75GB Users logged in:     1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 19 Proses *Login* Telah Berhasil Dilakukan

4) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi:/home/rifqi#
```

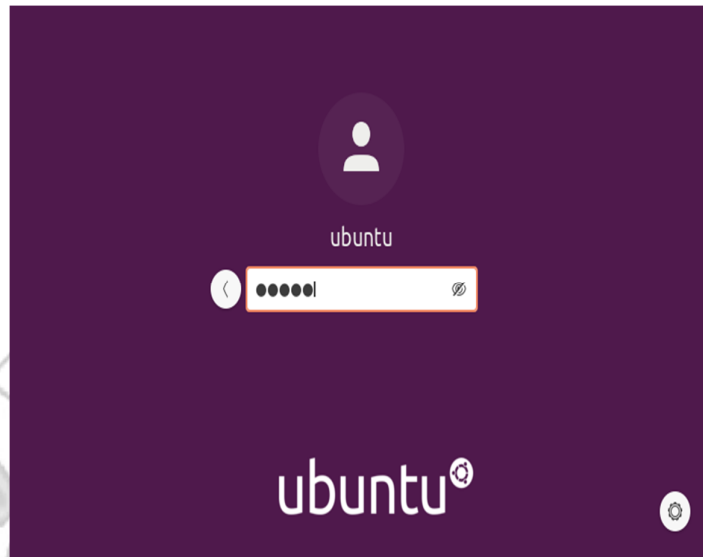
Gambar 4. 20 Masuk Super *User*

5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses ping ke *user* dengan alamat IP 192.168.137.3 dan proses ping berhasil

```
root@rifqi: /home/rifqi
root@rifqi:/home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data:
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

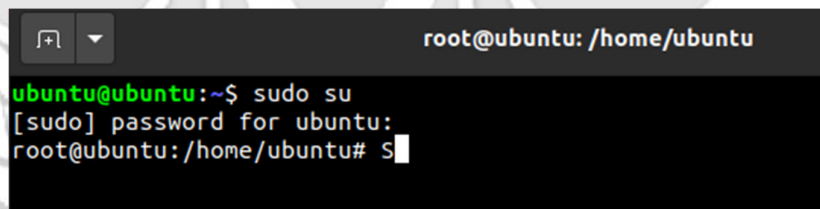
Gambar 4. 21 Proses PING

- 6) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan *password*.



Gambar 4. 22 Proses Login Admin

- 7) Selanjutnya menjalankan *admin* untuk melakukan *remote* pada *port* 80 (HTTP), tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).



Gambar 4. 23 Masuk Super User

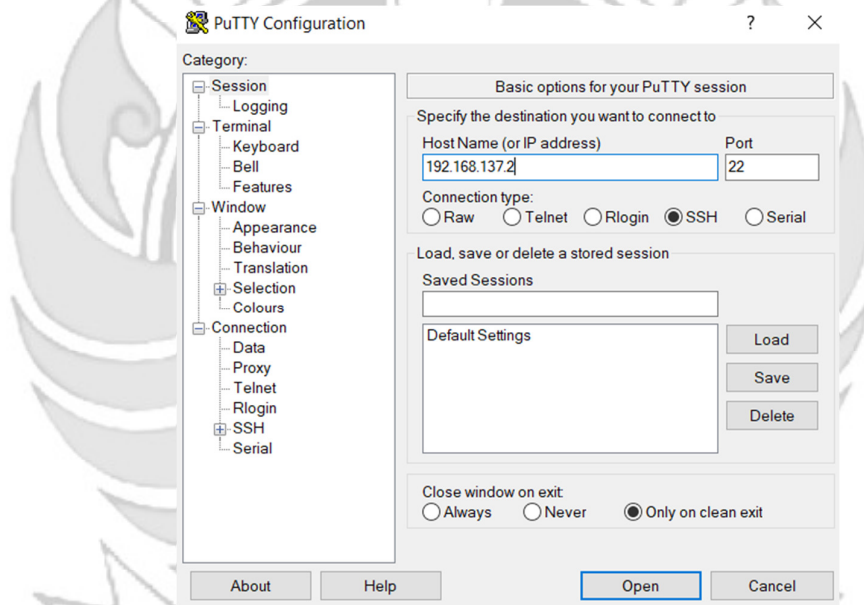
- 8) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 80 (HTTP). Membuktikan bahwa *server* dapat dengan mudah diakses oleh orang yang tidak berhak karena pada *server* tidak di tutup total dengan cara mendrop semua akses menggunakan *iptables* (*firewall*).



Gambar 4. 24 HTTP Berhasil Diakses

d. Langkah Uji Coba melakukan *remote* pada *port* 21 (FTP)

- 1) Menjalankan aplikasi putty dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 25 Konfigurasi Putty

- 2) Tampilan ketika proses *login*, memasukkan *username* dan password.



Gambar 4. 26 Proses *Login* Putty

- 3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load: 0.060546875   Processes:            254
Usage of /:   30.9% of 9.75GB   Users logged in:     1
Memory usage: 10%           IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 27 Login Telah Berhasil Dilakukan

- 4) Tampilan ketika masuk ke level *user* yang lebih tinggi (super *user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

Gambar 4. 28 Masuk Super User

- 5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses ping ke *user* dengan alamat IP 192.168.137.3 dan proses ping berhasil.

```
root@rifqi: /home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data:
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

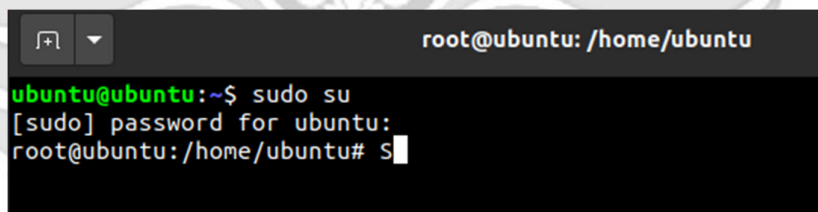
Gambar 4. 29 Proses PING

- 6) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan password.



Gambar 4. 30 Proses *Login Admin*

- 7) Selanjutnya menjalankan *admin* untuk melakukan *remote* pada *port 21* (FTP), tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).



Gambar 4. 31 Masuk Super User

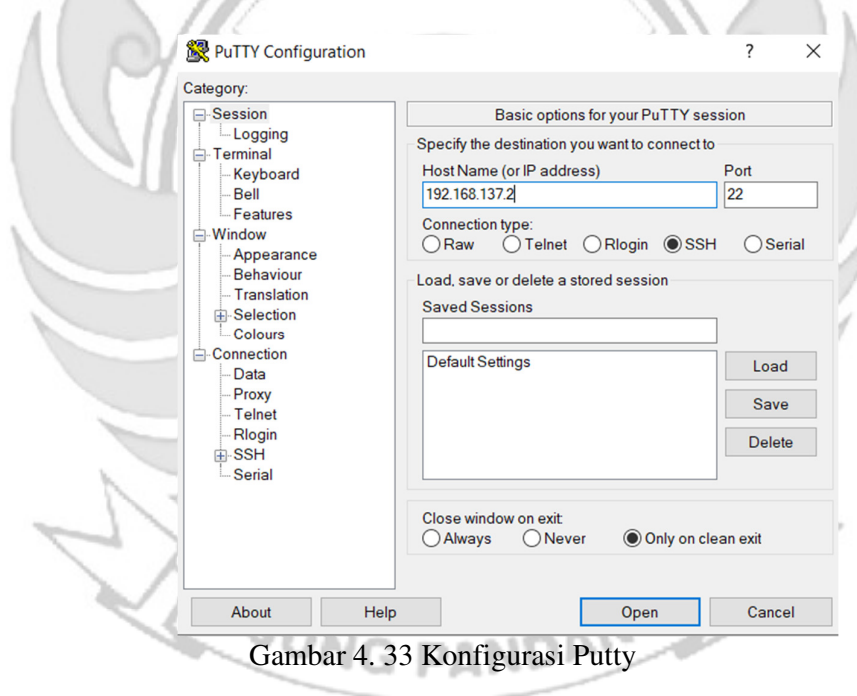
- 8) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port 21* (FTP). Membuktikan bahwa *server* dapat dengan mudah diakses oleh orang yang tidak berhak karena pada *server* tidak di tutup total dengan cara mendrop semua akses menggunakan *iptables* (*firewall*).

```
root@ubuntu:/home/ubuntu# ftp -p 192.168.137.2
Connected to 192.168.137.2.
220 (vsFTPD 3.0.5)
Name (192.168.137.2:ubuntu): politeknik
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> S
```

Gambar 4. 32 FTP Berhasil Diakses

e. Langkah Uji Coba melakukan *remote* pada *port 25* (SMTP)

- 1) Menjalankan aplikasi putty dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 33 Konfigurasi Putty

- 2) Tampilan ketika proses *login*, memasukkan *username* dan password.

```
root@rifqi:/home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi:/home/rifqi#
```

Gambar 4. 34 Proses Login Putty

- 3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875   Processes:            254
Usage of /:   30.9% of 9.75GB Users logged in:        1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 35 Proses *Login* Telah Berhasil Dilakukan

- 4) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

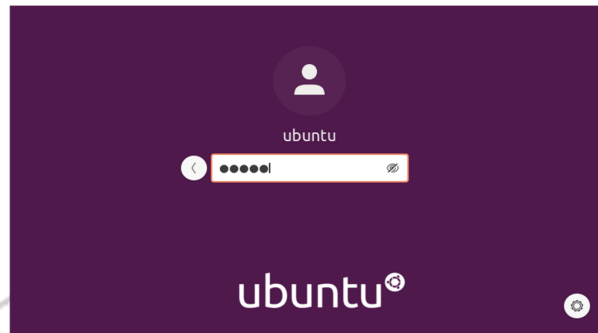
Gambar 4. 36 Masuk Super *User*

- 5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses ping ke *user* dengan alamat IP 192.168.137.3 dan proses ping berhasil.

```
root@rifqi: /home/rifqi
root@rifqi: /home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data:
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

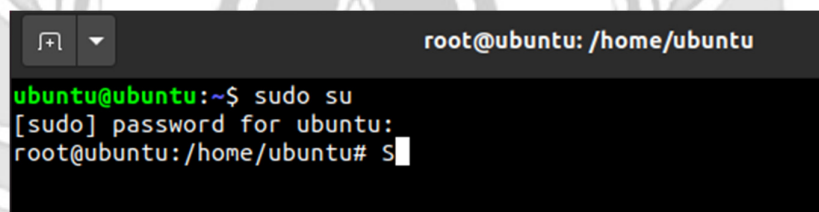
Gambar 4. 37 Proses PING

- 6) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan password.



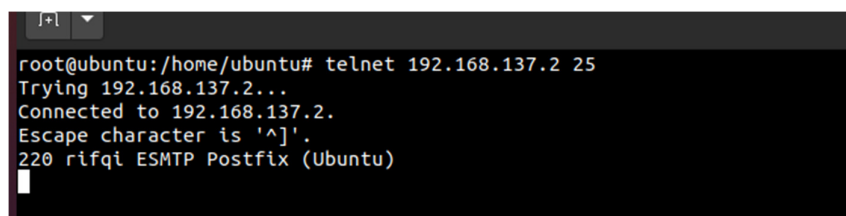
Gambar 4. 38 Proses *Login Admin*

- 7) Selanjutnya menjalankan *admin* untuk melakukan *remote* pada *port 25* (SMTP), tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).



Gambar 4. 39 Proses Masuk Super *User*

- 8) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port 25* (SMTP). Membuktikan bahwa *server* dapat dengan mudah diakses oleh orang yang tidak berhak karena pada *server* tidak di tutup total dengan cara mendrop semua akses menggunakan *iptables* (*firewall*).



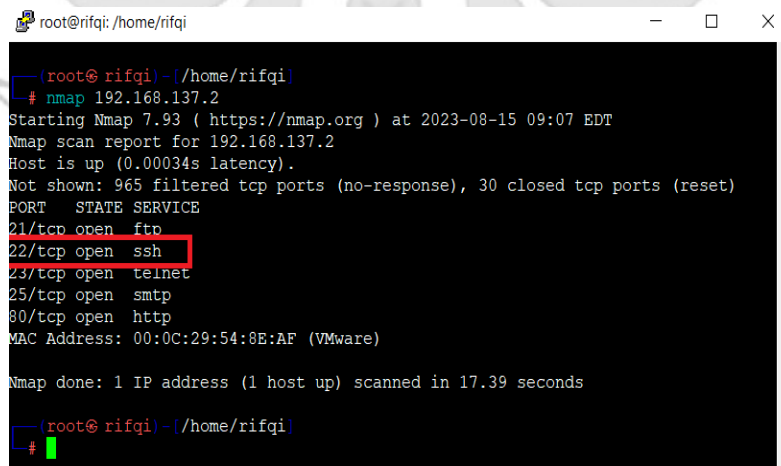
Gambar 4. 40 SMTP Berhasil Diakses

Admin tidak selamanya dapat mengakses *server* secara langsung, karena akan terdapat kondisi *admin* diberikan tugas keluar kota akan tetapi *admin* tetap diharuskan untuk mengakses *server* sehingga *admin* melakukan dengan cara *via remote*. Jika *admin* mengakses *server* secara *via remote* terdapat suatu celah keamanan yang dapat dimanfaatkan oleh *attacker* untuk melakukan penyadapan.

Pada penelitian ini *attacker* melakukan penyadapan dengan menggunakan serangan *port scanning*. Serangan *port scanning* dilakukan untuk mengetahui informasi yang terdapat pada *server* seperti celah pada *port* tujuan terbuka atau tertutup. Pada tahap pengujian *port scanning* menggunakan tool NMAP (*Network Mapper*). Berikut penjelasan menggunakan serangan *port scanning* untuk mengetahui *port* tujuan terbuka atau tertutup

a. *Port scanning port 22 (SSH)*

Pada tahap pengujian *port scanning* menggunakan tool NMAP (*Network Mapper*) dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port 22 (SSH)*. Pengujian ini dilakukan pada saat *server* dalam keadaan normal.



```
root@rifqi: /home/rifqi
--(root@ rifqi)-(/home/rifqi)
--# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 09:07 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00034s latency).
Not shown: 965 filtered tcp ports (no-response), 30 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
80/tcp    open  http
MAC Address: 00:0C:29:54:8E:AF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 17.39 seconds

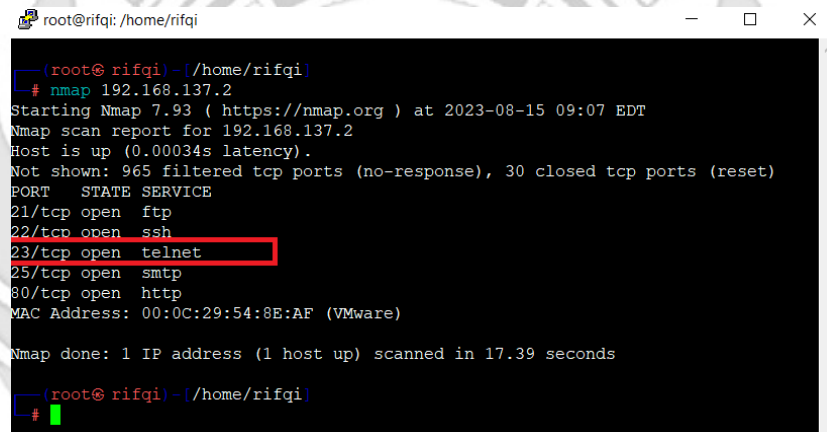
--(root@ rifqi)-(/home/rifqi)
--#
```

Gambar 4. 41 Penyerangan *Port Scanning SSH* Sebelum *Port knocking*

Terlihat pada Gambar 4. 43 bahwa *port 22* (SSH) pada *server* dalam keadaan terbuka sehingga *attacker* dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port 22* (SSH).

b. *Port Scanning port 23* (TELNET).

Pada tahap pengujian *port scanning* menggunakan *tool* NMAP (*Network Mapper*) dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port 23* (TELNET). Pengujian ini dilakukan pada saat *server* dalam keadaan normal.



```
root@rifqi: ~/home/rifqi
└─(root@ rifqi) -[~/home/rifqi]
# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 09:07 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00034s latency).
Not shown: 965 filtered tcp ports (no-response), 30 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
80/tcp    open  http
MAC Address: 00:0C:29:54:8E:AF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 17.39 seconds

└─(root@ rifqi) -[~/home/rifqi]
#
```

Gambar 4. 42 Penyerangan *Port Scanning* TELNET Sebelum *Port knocking*

Terlihat pada Gambar 4. 44 bahwa *port 23* (TELENET) pada *server* dalam keadaan terbuka sehingga *attacker* dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port 23* (TELNET).

c. *Port Scanning port 80* (HTTP)

Pada tahap pengujian *port scanning* menggunakan *tool* NMAP (*Network Mapper*) dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port 80* (HTTP). Pengujian ini dilakukan pada saat *server* dalam keadaan normal.

```
root@rifqi: /home/rifqi
└─(root@ rifqi)-[/home/rifqi]
└─# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 09:07 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00034s latency).
Not shown: 965 filtered tcp ports (no-response), 30 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
80/tcp    open  http
MAC Address: 00:0C:29:54:8E:AF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 17.39 seconds

└─(root@ rifqi)-[/home/rifqi]
└─#
```

Gambar 4. 43 Penyerangan *Port Scanning* HTTP Sebelum *Port knocking*

Terlihat pada Gambar 4. 45 bahwa *port* 80 (HTTP) pada *server* dalam keadaan terbuka sehingga *attacker* dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port* 80 (HTTP).

d. *Port Scanning port 21* (FTP)

Pada tahap pengujian *port scanning* menggunakan *tool* NMAP (*Network Mapper*) dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port* 21 (FTP). Pengujian ini dilakukan pada saat *server* dalam keadaan normal.

```
root@rifqi: /home/rifqi
└─(root@ rifqi)-[/home/rifqi]
└─# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 09:07 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00034s latency).
Not shown: 965 filtered tcp ports (no-response), 30 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
80/tcp    open  http
MAC Address: 00:0C:29:54:8E:AF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 17.39 seconds

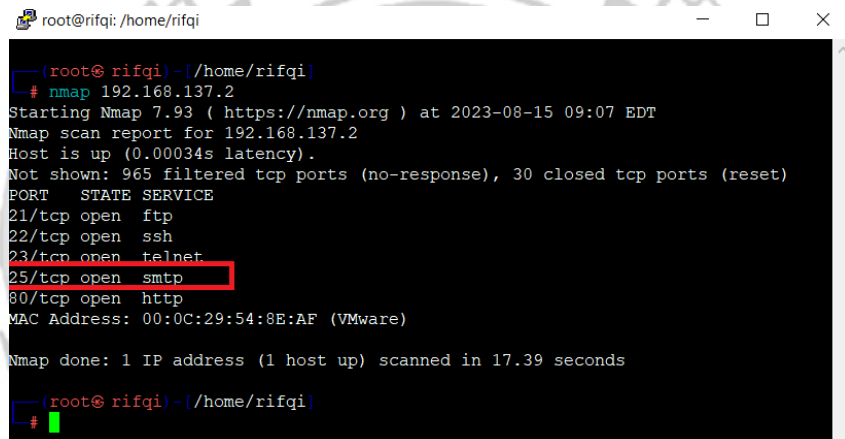
└─(root@ rifqi)-[/home/rifqi]
└─#
```

Gambar 4. 44 Penyerangan *Port Scanning* FTP Sebelum *Port knocking*

Terlihat pada Gambar 4. 46 bahwa *port 21 (FTP)* pada *server* dalam keadaan terbuka sehingga *attacker* dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port 21 (FTP)*.

e. *Scanning 25 (SMTP)*

Pada tahap pengujian *scanning* menggunakan *tool* NMAP (*Network Mapper*) dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port 25 (SMTP)*. Pengujian ini dilakukan pada saat *server* dalam keadaan normal.



```
root@rifqi: /home/rifqi
└─(root@ rifqi) - /home/rifqi
# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 09:07 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00034s latency).
Not shown: 965 filtered tcp ports (no-response), 30 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
80/tcp    open  http
MAC Address: 00:0C:29:54:8E:AF (VMware)
Nmap done: 1 IP address (1 host up) scanned in 17.39 seconds
└─(root@ rifqi) - /home/rifqi
#
```

Gambar 4. 45 Penyerangan *Port Scanning SMTP* Sebelum *Port knocking*

Terlihat pada Gambar 4. 47 bahwa *port 25 (SMTP)* pada *server* dalam keadaan terbuka sehingga *attacker* dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port 25 (SMTP)*.

Serangan yang dilakukan *attacker* menggunakan *port scanning* pada *server* dalam keadaan normal maka dapat disimpulkan *server* dalam keadaan tidak aman karena serangan *port scanning* tersebut memudahkan *attacker* mendapatkan informasi mengenai *port-port* apa saja yang sedang di *remote* oleh *admin* atau *port-port* apa saja yang dalam kondisi terbuka karena tidak ada penerapan sistem keamanan pada *server*.

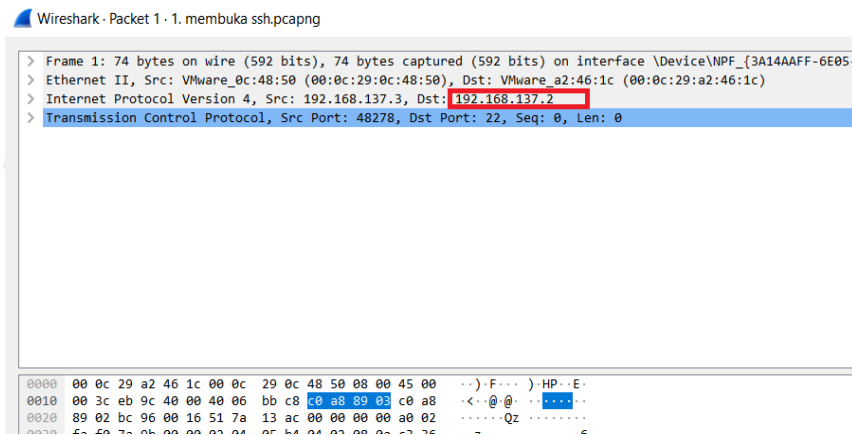
Setelah melakukan serangan dengan menggunakan metode *port scanning* selanjutnya *attacker* melakukan serangan dengan dengan tingkat yang lebih tinggi untuk mendapatkan informasi yang lebih banyak maka *attacker* melakukan serangan dengan metode *sniffing* menggunakan *wireshark*.

a. *Sniffing port 22 (SSH)*

Pada tahap pengujian penyerangan *sniffing attacker* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 22 (SSH)* untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

1	0.000000	192.168.137.3	192.168.137.2	TCP	74	48278 → 22 [SYN] Seq=0 Win=64240 Len=0
2	0.000984	192.168.137.1	192.168.137.3	ICMP	102	Redirect (Redirect for netwo
3	0.001089	192.168.137.3	192.168.137.2	TCP	74	[TCP Retransmission] [TCP Port numbers r
4	0.001200	192.168.137.2	192.168.137.3	TCP	74	22 → 48278 [SYN, ACK] Seq=0 Ack=1 Win=65
5	0.001800	192.168.137.2	192.168.137.3	TCP	74	[TCP Retransmission] 22 → 48278 [SYN, AC
6	0.001813	192.168.137.3	192.168.137.2	TCP	66	48278 → 22 [ACK] Seq=1 Ack=1 Win=64256 L

Gambar 4. 46 Port SSH



Gambar 4. 47 IP Server

Terlihat pada Gambar 4. 48 dan 4. 49 bahwa ketika seorang *administrator* jaringan melakukan *remote server* pada *port 22 (SSH)* maka seorang *attacker*

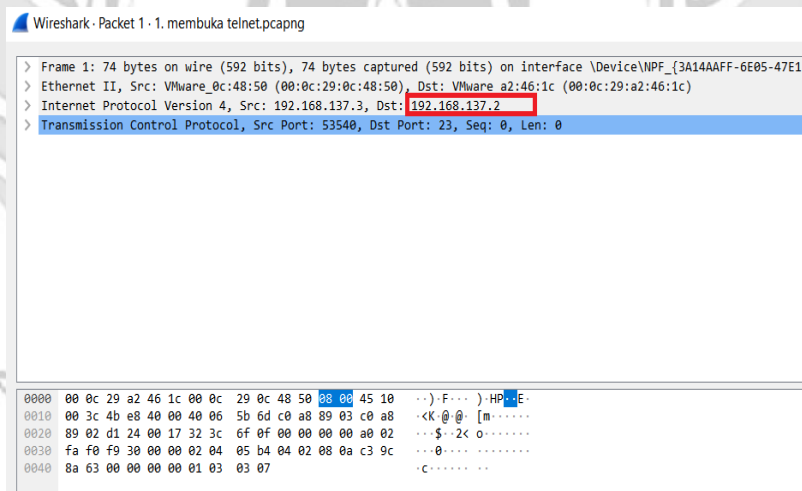
dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *port* yang sedang di *remote* serta ip dari *server*.

b. *Sniffing port 23* (TELNET)

Pada tahap pengujian penyerangan *sniffing attacker* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 23* (TELNET) untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

Time	Source	Destination	Protocol	Length	Info
1 0.000000	192.168.137.3	192.168.137.2	TCP	74	53540 → 23 [SYN] Seq=0 Win=64240 Len=0
2 0.000302	192.168.137.2	192.168.137.3	TCP	74	23 → 53540 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0

Gambar 4. 48 *Port TELNET*



Gambar 4. 49 *IP Server*

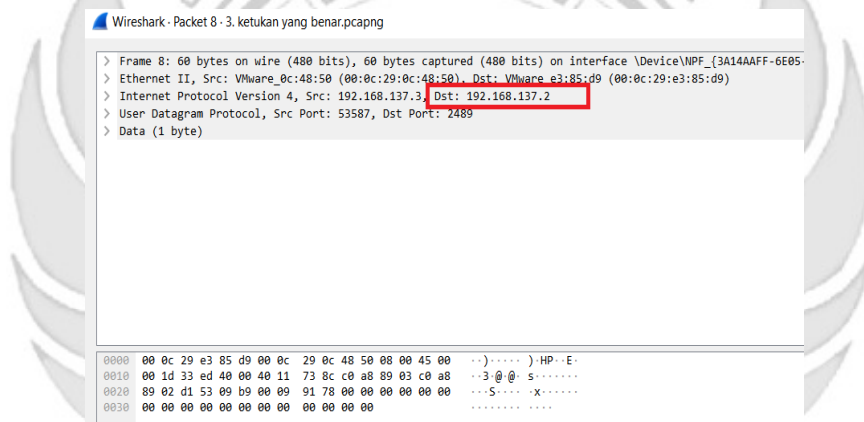
Terlihat pada Gambar 4. 50 dan 4. 51 bahwa ketika seorang *administrator* jaringan melakukan *remote server* pada *port 23* (TELNET) maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *port* yang sedang di *remote* serta ip dari *server*.

c. *Sniffing port 80 (HTTP)*

Pada tahap pengujian penyerangan *sniffing attacker* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 80 (HTTP)* untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

5	0.192324	192.168.137.3	202.67.36.152	TCP	74	58656	→ 80	[SYN]	Seq=0 Win=64240 Len=0 MSS
6	0.224387	202.67.36.152	192.168.137.3	TCP	74	80	→ 58656	[SYN, ACK]	Seq=0 Ack=1 Win=6516
7	0.238976	192.168.137.3	202.67.36.152	TCP	66	58656	→ 80	[ACK]	Seq=1 Ack=1 Win=64256 Len

Gambar 4. 50 Port HTTP



Gambar 4. 51 IP Server

Terlihat pada Gambar 4. 52 dan 4. 53 bahwa ketika seorang *administrator* jaringan melakukan *remote server* pada *port 80 (HTTP)* maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *port* yang sedang di *remote* serta ip dari *server*.

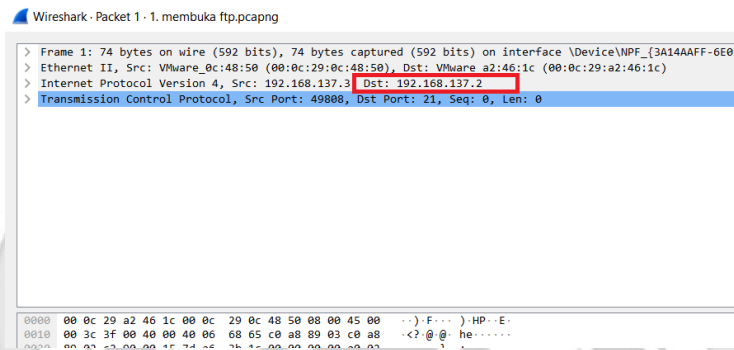
d. *Sniffing port 21 (FTTP)*

Pada tahap pengujian penyerangan *sniffing attacker* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 21 (FTP)*

untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

Time	Source	Destination	Protocol	Length	Info
1 0.000000	192.168.137.3	192.168.137.2	TCP	74	49808 → 21 [SYN] Seq=0 Win=64240 Len=
2 0.001066	192.168.137.2	192.168.137.3	TCP	74	21 → 49808 [SYN, ACK] Seq=0 Ack=1 Win=

Gambar 4. 52 Port FTP



Gambar 4. 53 IP Server

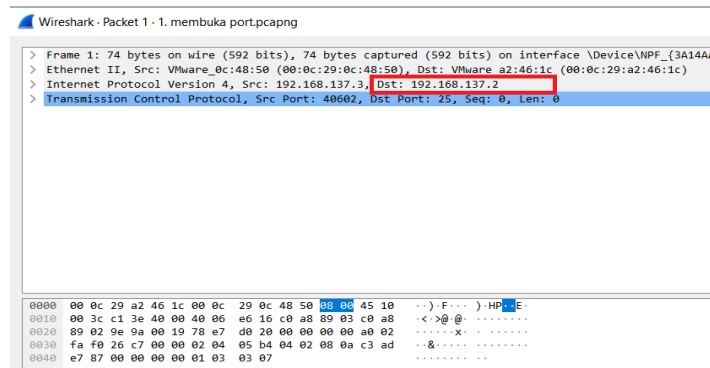
Terlihat pada Gambar 4. 54 dan 4. 55 bahwa ketika seorang *administrator* jaringan melakukan *remote server* pada *port* 21 (FTP) maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *port* yang sedang di *remote* serta ip dari *server*.

e. *Sniffing port 25 (SMTP)*

Pada tahap pengujian penyerangan *sniffing attacker* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port* 25 (SMTP) untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

No.	Time	Source	Destination	Protocol	Length	Info
1 0.000000	192.168.137.3	192.168.137.2	TCP	74	40602 → 25 [SYN] Seq=0 Win=64	
2 0.000306	192.168.137.2	192.168.137.3	TCP	74	25 → 40602 [SYN, ACK] Seq=0 A	

Gambar 4. 54 Port SMTP



Gambar 4. 55 IP SMTP

Terlihat pada Gambar 4. 56 dan 4. 57 bahwa ketika seorang *administrator* jaringan melakukan *remote server* pada *port 25* (SMTP) maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *port* yang sedang di *remote* serta *ip* dari *server*.

Serangan yang dilakukan *attacker* menggunakan *sniffing* pada *server* dalam keadaan normal maka dapat disimpulkan *server* dalam keadaan tidak aman karena serangan *sniffing* tersebut memudahkan *attacker* mendapatkan informasi mengenai *port* yang sedang di *remote* oleh *admin* serta *ip* dari *server*, maka dari celah keamanan tersebut dibutuhkan sebuah sistem keamanan yang dapat melindungi *server* agar *attacker* tidak dapat dengan mudah menemukan informasi-informasi untuk dapat mengakses *server*.

4.1.2 Pengujian Server Menggunakan Port knocking

Jika terdapat kondisi yang mengharuskan seorang *admin* melakukan *remote server*, pada penelitian ini terdapat beberapa *port* yang dapat di *remote* oleh seorang *admin* yaitu SSH, TELNET, HTTP, FTP dan SMTP maka *admin* menerapkan metode *port knocking* untuk mengatasi permasalahan yang telah dijelaskan sebelumnya.

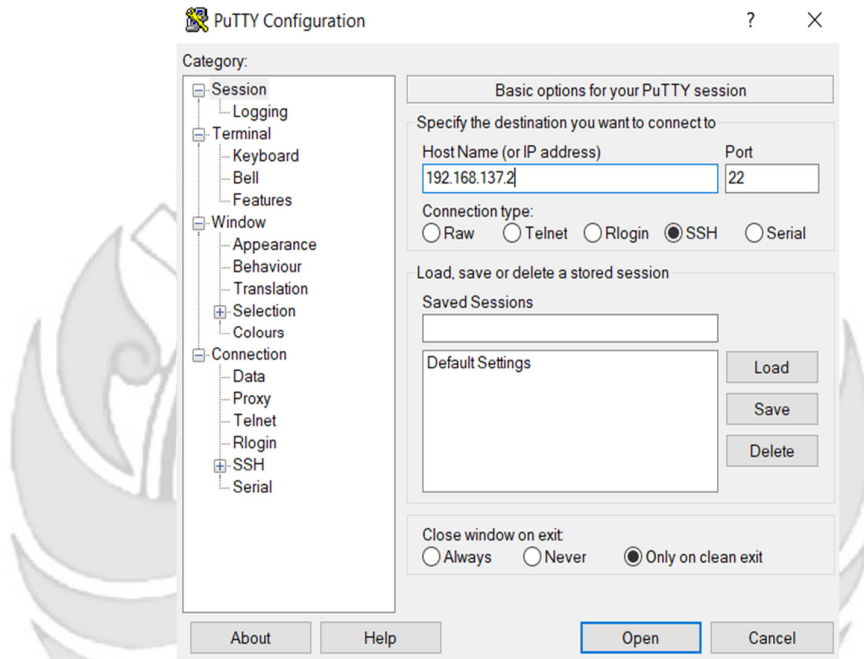
Metode *port knocking* dapat digunakan untuk menjaga semua *port* yang ditutup sampai pengguna melakukan autentikasi dengan *knock port*. Jika urutan ketukan benar maka *server* memberikan ijin kepada *client* untuk dapat mengakses *port* tersebut, tetapi apabila urutan salah maka *client* tidak dapat mengakses *port* tersebut. Jadi jika terdapat seorang *attacker* yang ingin mengakses *server* secara ilegal maka *attacker* tersebut harus mengetahui *sequence* dengan benar yang terpasang pada *server*

1. Konfigurasi *Port knocking* pada *server*: “*/etc/knockd.conf*”

```
GNU nano 6.2 /etc/knockd.conf
[options]
UseSyslog
[openSSH]
sequence = 3647,6029,4500
seq_timeout = 5
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags = syn
[closeSSH]
sequence = 4500,6029,3647
seq_timeout = 5
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags = syn
[openHTTP]
sequence = 2489,3872,1200,7381
seq_timeout = 5
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 80 -j ACCEPT
tcpflags = syn
[closeHTTP]
sequence = 7381,1200,3872,2489
seq_timeout = 5
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 80 -j ACCEPT
tcpflags = syn
[openFTP]
sequence = 3892,4820,5390,2680
seq_timeout = 5
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 21 -j ACCEPT
tcpflags = syn
[closeFTP]
sequence = 2680,5390,4820,3892
seq_timeout = 5
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 21 -j ACCEPT
tcpflags = syn
[openSMTP]
sequence = 1400,1500,1600,1700,1800
seq_timeout = 5
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 25 -j ACCEPT
tcpflags = syn
[closeSMTP]
sequence = 1800,1700,1600,1500,1400
seq_timeout = 5
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 25 -j ACCEPT
tcpflags = syn
[openTELNET]
sequence = 7324,3429,9125
seq_timeout = 5
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 23 -j ACCEPT
tcpflags = syn
[closeSSH]
sequence = 9125,3429,7324
seq_timeout = 5
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 23 -j ACCEPT
tcpflags = syn
```

Gambar 4. 56 Konfigurasi *Port knocking*

- a. Langkah Uji Coba membuka dan menutup *port 22* (SSH). Pada uji coba yang dilakukan menggunakan aplikasi *putty* untuk membuka *server*.
- 1) Menjalankan aplikasi *putty* dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 57 Konfigurasi Putty

- 1) Tampilan ketika proses *login*, memasukkan *username* dan *password*.



Gambar 4. 58 Proses Login Putty

- 2) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2 tanpa menggunakan *firewall* dan *port knocking*.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875   Processes:           254
Usage of /:   30.9% of 9.75GB Users logged in:      1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 59 Proses Login Telah Berhasil Dilakukan

3) Tampilan ketika masuk ke level *user* yang lebih tinggi (super *user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

Gambar 4. 60 Masuk ke Super *User*

4) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses *ping* ke *user* dengan alamat IP 192.168.137.3 dan proses *ping* berhasil

```
root@rifqi: /home/rifqi
root@rifqi:/home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data:
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

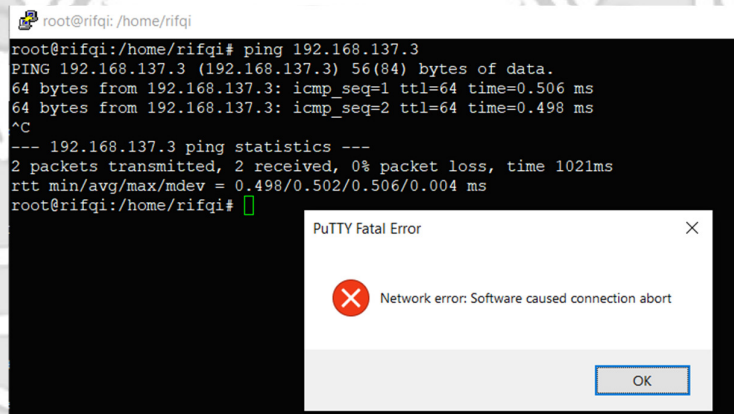
Gambar 4. 61 Proses PING

- 5) Setelah *server* berhasil terhubung dengan *melakukan proses ping* ke *user*, maka selanjutnya *server* di tutup total dengan cara mendrop semua akses sehingga tidak ada akses tcp yang dapat lewat. Menggunakan perintah iptables (*firewall*).

```
iptables -I INPUT -p tcp -j DROP
```

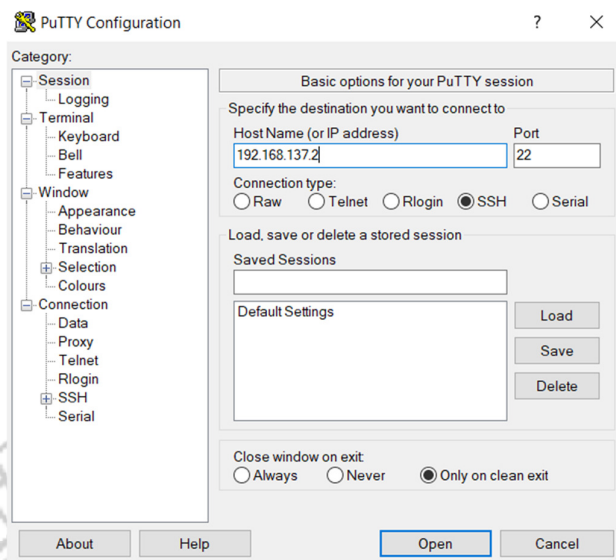
Gambar 4. 62 Mendrop Akses *Server*

Setelah mendrop semua akses pada *server* dengan perintah iptables maka *server* tidak dapat diakses lagi.



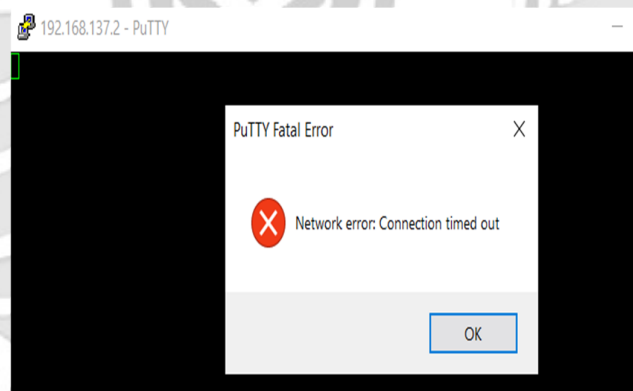
Gambar 4. 63 Akses *Server* di *Drop*

- 6) Setelah *server* berhasil ditutup dengan perintah iptables maka akan dilakukan uji coba kembali masuk ke *server* yang dituju yaitu IP 192.168.137.2 dengan menggunakan aplikasi putty dan membuktikan bahwa *server* tersebut sudah ditutup sehingga tidak dapat diakses secara bebas.



Gambar 4. 64 Konfigurasi Putty

- 7) Hasil ketika *server* terbukti berhasil ditutup sehingga *server* tidak dapat diakses secara bebas.



Gambar 4. 65 *Server* Tidak Dapat di Akses

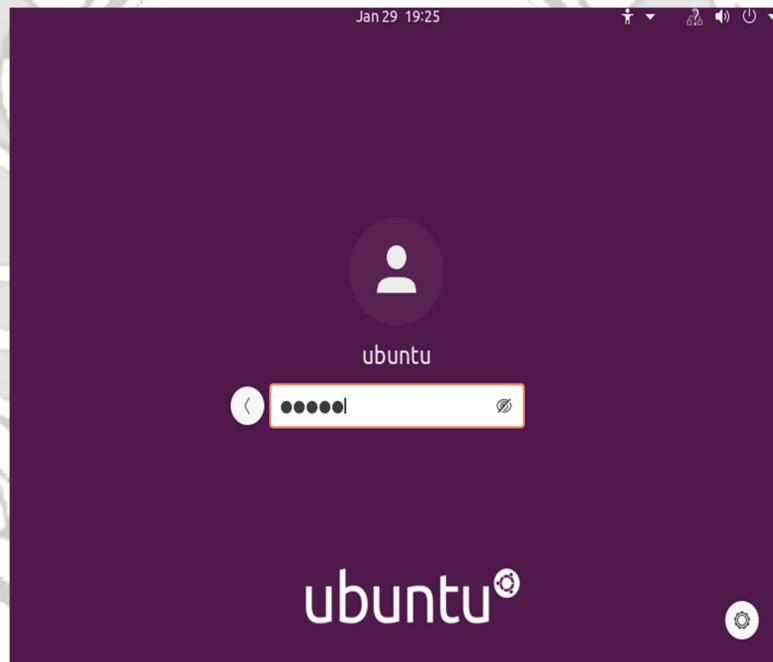
Setelah akses dari *server* ditutup, hanya terdapat satu metode yang dipakai untuk masuk ke sistem *server* dengan cara menggunakan metode *port knocking*. Untuk dapat mengakses *server* maka harus melalui *admin* yang sudah dilakukan penginstalan packet *knockd* agar dapat menggunakan metode *port knocking*.

```
root@ubuntu:/home/ubuntu# sudo systemctl status knockd
● knockd.service - Port-Knock Daemon
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor prese
   Active: active (running) since Tue 2023-08-15 06:32:15 PDT; 2s ago
     Docs: man:knockd(1)
    Main PID: 2592 (knockd)
      Tasks: 1 (limit: 4572)
     Memory: 656.0K
    CGroup: /system.slice/knockd.service
            └─2592 /usr/sbin/knockd -i ens33

Aug 15 06:32:15 ubuntu systemd[1]: Started Port-Knock Daemon.
Aug 15 06:32:15 ubuntu knockd[2592]: starting up, listening on ens33
lines 1-12/12 (END)
```

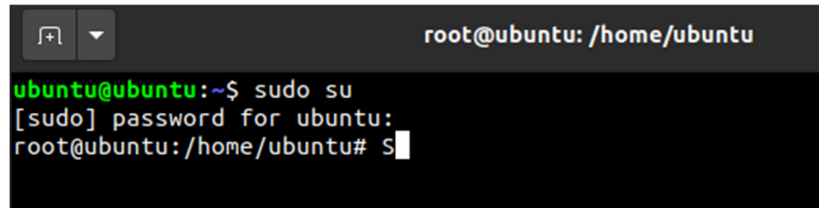
Gambar 4. 66 Pengecekan Status *Knockd*

- 8) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan *password*.



Gambar 4. 67 Proses *Login Admin*

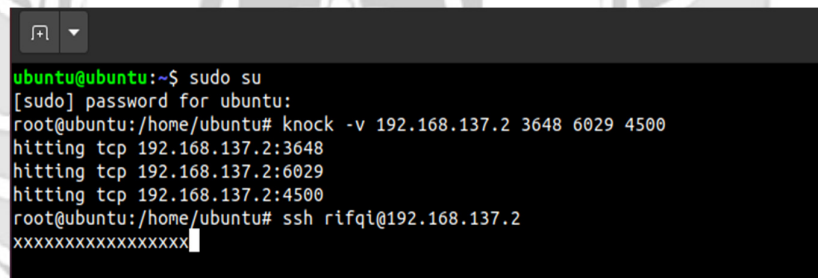
9) Tampilan ketika masuk ke level *user* yang lebih tinggi (super *user*).



```
root@ubuntu: /home/ubuntu
ubuntu@ubuntu:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu: /home/ubuntu# s
```

Gambar 4. 68 Masuk ker Super User

10) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 22 (SSH) menggunakan ketukan yang salah. Membuktikan bahwa *server* tidak dapat diakses menggunakan *port* 22 (SSH) secara bebas jika ketukan tidak sesuai dengan konfigurasi yang dilakukan di *server*.



```
ubuntu@ubuntu:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu: /home/ubuntu# knock -v 192.168.137.2 3648 6029 4500
hitting tcp 192.168.137.2:3648
hitting tcp 192.168.137.2:6029
hitting tcp 192.168.137.2:4500
root@ubuntu: /home/ubuntu# ssh rifqi@192.168.137.2
xxxxxxxxxxxxxxxxxxxx
```

Gambar 4. 69 Proses Membuka SSH Ketukan Salah

11) Selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 22 (SSH) menggunakan ketukan yang benar. Membuktikan bahwa *server* dapat diakses menggunakan *port* 22 (SSH) jika ketukan yang sesuai dengan konfigurasi yang dilakukan di *server*.


```
rifqi@rifqi: ~  
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 3647 6029 4500  
hitting tcp 192.168.137.2:3647  
hitting tcp 192.168.137.2:6029  
hitting tcp 192.168.137.2:4500  
root@ubuntu:/home/ubuntu# ssh rifqi@192.168.137.2  
rifqi@192.168.137.2's password:  
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Tue Aug 15 01:33:52 PM UTC 2023  
  
System load: 0.01318359375   Processes:            230  
Usage of /:  30.9% of 9.75GB   Users logged in:     1  
Memory usage: 10%           IPv4 address for ens33: 192.168.137.2  
Swap usage:  0%  
  
* Introducing Expanded Security Maintenance for Applications.  
  Receive updates to over 25,000 software packages with your  
  Ubuntu Pro subscription. Free for personal use.  
  
  https://ubuntu.com/pro  
  
Expanded Security Maintenance for Applications is not enabled.  
  
59 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection  
  
Last login: Tue Aug 15 12:47:41 2023  
rifqi@rifqi:~$
```

Gambar 4. 70 Proses Membuka SSH Ketukan Benar

12) Setelah dilakukan uji coba masuk ke *server* mengakses *port 22* (SSH) menggunakan teknik *port knocking* dengan ketukan yang benar selanjutnya dilakukan uji coba menutup *server* yang terbuka dengan ketukan yang salah maka *server* tersebut masih dapat diakses.

```
rifqi@rifqi: ~
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 4600 6928 3547
hitting tcp 192.168.137.2:4600
hitting tcp 192.168.137.2:6928
hitting tcp 192.168.137.2:3547
root@ubuntu:/home/ubuntu# ssh rifqi@192.168.137.2
rifqi@192.168.137.2's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 01:37:08 PM UTC 2023

System load:  0.0          Processes:      234
Usage of /:   30.9% of 9.75GB  Users logged in:  1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

 * Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection

Last login: Tue Aug 15 13:33:53 2023 from 192.168.137.3
rifqi@rifqi:~$
```

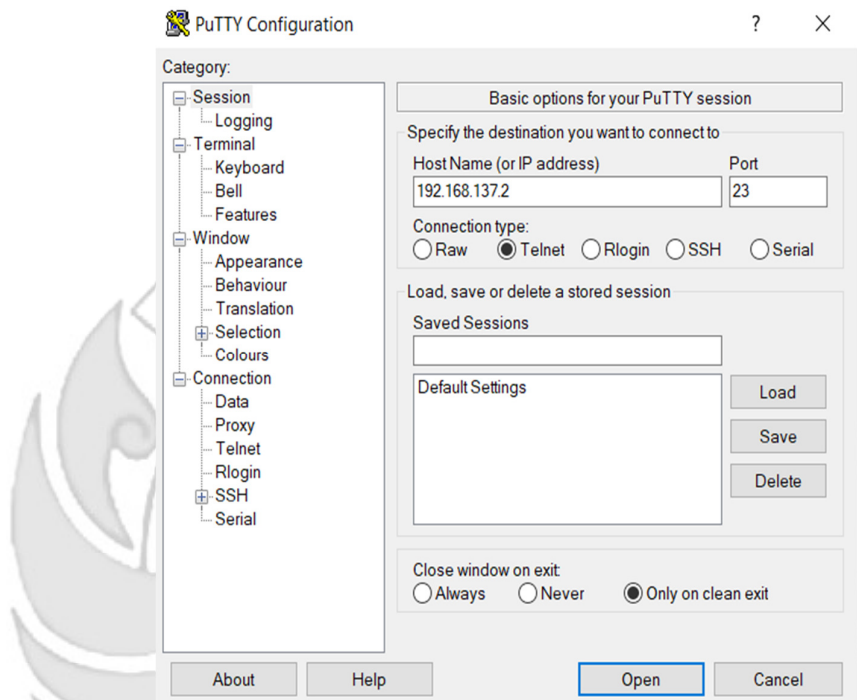
Gambar 4. 71 Proses Menutup SSH Ketukan Salah

- 13) Setelah dilakukan uji coba menutup *server* dengan ketukan yang salah, selanjutnya menutup *server* agar tidak dapat diakses oleh orang yang tidak berhak maka digunakan teknik *port knocking* dengan ketukan yang benar.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 4500 6029 3647
hitting tcp 192.168.137.2:4500
hitting tcp 192.168.137.2:6029
hitting tcp 192.168.137.2:3647
root@ubuntu:/home/ubuntu# ssh rifqi@192.168.137.2
XXXXXXXXXXXX
```

Gambar 4. 72 Proses Menutup SSH Ketukan Benar

- b. Langkah uji coba untuk membuka dan menutup *port 23* (TELNET)
- 1) Menjalankan aplikasi putty dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 73 Konfigurasi Putty

- 2) Tampilan ketika proses *login*, memasukkan *username* dan *password*.



Gambar 4. 74 Proses Login Putty

- 3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2 tanpa menggunakan *firewall* dan *port knocking*.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875   Processes:            254
Usage of /:   30.9% of 9.75GB Users logged in:        1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

69 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet con
nection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 75 Proses *Login* Telah Berhasil Dilakukan

- 4) Tampilan ketika masuk ke level *user* yang lebih tinggi (super *user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

Gambar 4. 76 Masuk ke Super *User*

- 5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses *ping* ke *user* dengan alamat IP 192.168.137.3 dan proses *ping* berhasil.

```
root@rifqi: /home/rifqi
root@rifqi: /home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data:
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

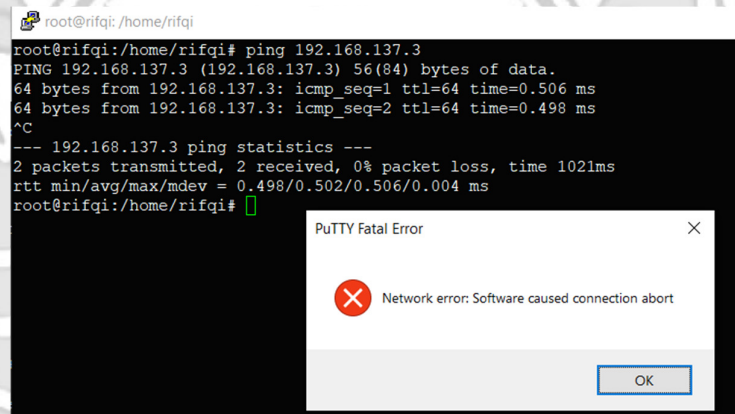
Gambar 4. 77 Proses PING

- 6) Setelah *server* berhasil terhubung dengan *melakukan* proses *ping* ke *user*, maka selanjutnya *server* di tutup total dengan cara mendrop semua akses sehingga tidak ada akses tcp yang dapat lewat. Menggunakan perintah iptables (*firewall*).

```
iptables -I INPUT -p tcp -j DROP
```

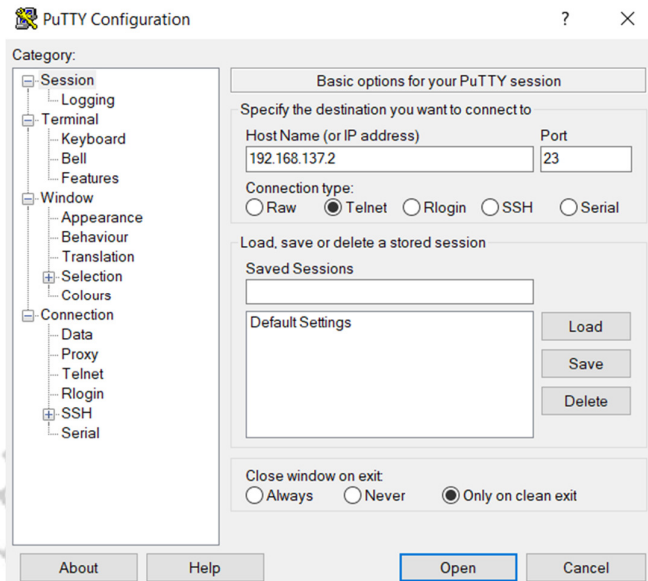
Gambar 4. 78 Mendrop Akses Server

Setelah mendrop semua akses pada *server* dengan perintah iptables maka *server* tidak dapat diakses lagi.



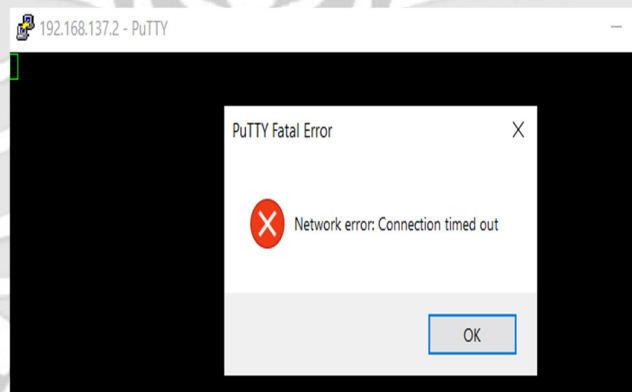
Gambar 4. 79 Akses Server di Drop

- 7) Setelah *server* berhasil ditutup dengan perintah iptables maka akan dilakukan uji coba kembali masuk ke *server* yang dituju yaitu IP 192.168.137.2 dengan menggunakan aplikasi putty dan membuktikan bahwa *server* tersebut sudah ditutup sehingga tidak dapat diakses secara bebas.



Gambar 4. 80 Konfigurasi Putty

- 8) Hasil ketika *server* terbukti berhasil ditutup sehingga *server* tidak dapat diakses secara bebas.



Gambar 4. 81 Server Tidak Dapat di Akses

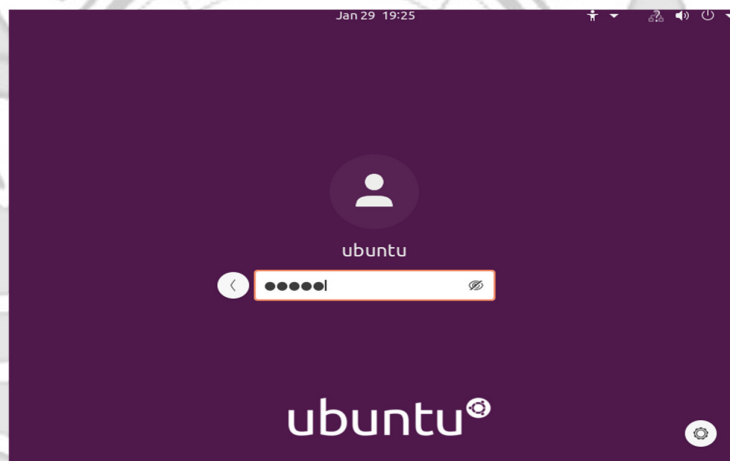
- 9) Setelah akses dari *server* ditutup, hanya terdapat satu metode yang dipakai untuk masuk ke sistem *server* dengan cara menggunakan metode *port knocking*. Untuk dapat mengakses *server* maka harus melalui *admin* yang sudah dilakukan penginstalan packet *knockd* agar dapat menggunakan metode *port knocking*.

```
root@ubuntu:/home/ubuntu# sudo systemctl status knockd
● knockd.service - Port-Knock Daemon
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor prese
   Active: active (running) since Tue 2023-08-15 06:32:15 PDT; 2s ago
     Docs: man:knockd(1)
   Main PID: 2592 (knockd)
     Tasks: 1 (limit: 4572)
    Memory: 656.0K
     CGroup: /system.slice/knockd.service
            └─2592 /usr/sbin/knockd -i ens33

Aug 15 06:32:15 ubuntu systemd[1]: Started Port-Knock Daemon.
Aug 15 06:32:15 ubuntu knockd[2592]: starting up, listening on ens33
lines 1-12/12 (END)
```

Gambar 4. 82 Pengecekan Status *Knockd*

10) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan *password*.



Gambar 4. 83 Proses *Login Admin*

11) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).

```
root@ubuntu: /home/ubuntu
ubuntu@ubuntu:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu: /home/ubuntu# S
```

Gambar 4. 84 Masuk ke *Super User*

12) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 23 (TELNET) menggunakan ketukan yang salah. Membuktikan bahwa *server* tidak dapat diakses menggunakan *port* 23 (TELNET) secara bebas jika ketukan tidak sesuai dengan konfigurasi yang dilakukan di *server*.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 7345 3439 9126
hitting tcp 192.168.137.2:7345
hitting tcp 192.168.137.2:3439
hitting tcp 192.168.137.2:9126
root@ubuntu:/home/ubuntu# telnet 192.168.137.2
Trying 192.168.137.2...
xxxxxxx
```

Gambar 4. 85 Proses Membuka TELNET Ketukan Salah

13) Selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 23 (TELNET) menggunakan ketukan yang benar. Membuktikan bahwa *server* dapat diakses menggunakan *port* 23 (TELNET) jika ketukan yang sesuai dengan konfigurasi yang dilakukan di *server*.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 7324 3429 9125
hitting tcp 192.168.137.2:7324
hitting tcp 192.168.137.2:3429
hitting tcp 192.168.137.2:9125
root@ubuntu:/home/ubuntu# telnet 192.168.137.2
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^]'.
Ubuntu 22.04.2 LTS
rifqi login: rifqi
Password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 01:47:11 PM UTC 2023

System load:  0.0          Processes:    235
Usage of /:   30.9% of 9.75GB  Users logged in:  1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:  0%
```

Gambar 4. 86 Proses Membuka TELNET Ketukan Benar

- 14) Setelah dilakukan uji coba masuk ke *server* mengakses *port* 23 (TELNET) menggunakan teknik *port knocking* dengan ketukan yang benar selanjutnya dilakukan uji coba menutup *server* yang terbuka dengan ketukan yang salah maka *server* tersebut masih dapat diakses.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 9126 3429 7324
hitting tcp 192.168.137.2:9126
hitting tcp 192.168.137.2:3429
hitting tcp 192.168.137.2:7324
root@ubuntu:/home/ubuntu# telnet 192.168.137.2
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^]'.
Ubuntu 22.04.2 LTS
rifqi login: rifqi
Password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 01:50:00 PM UTC 2023

System load: 0.0          Processes:                236
Usage of /:  30.9% of 9.75GB Users logged in:             1
Memory usage: 10%        IPv4 address for ens33: 192.168.137.2
Swap usage:  0%
```

Gambar 4. 87 Proses Menutup TELNET Ketukan Salah

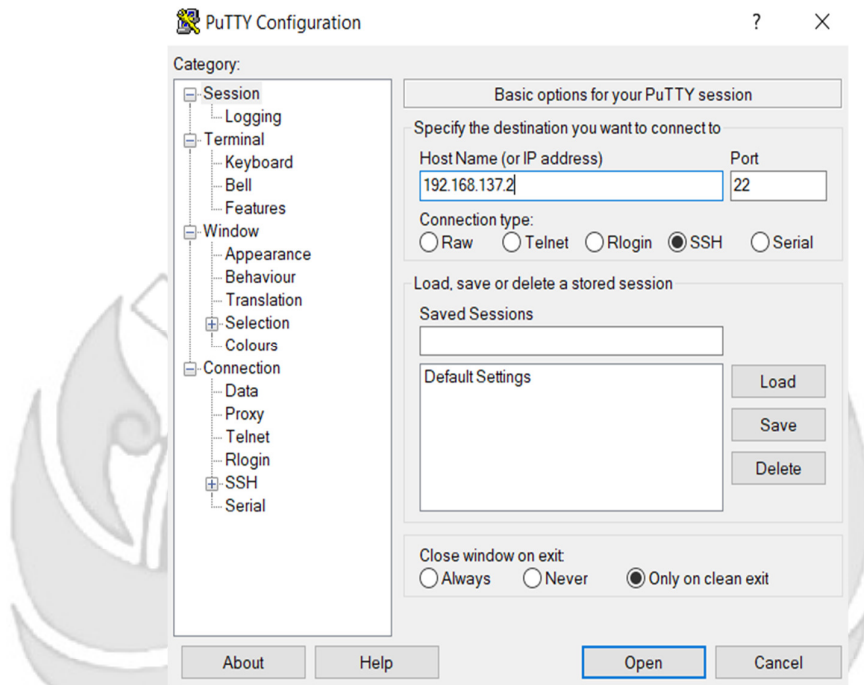
- 15) Setelah dilakukan uji coba menutup *server* dengan ketukan yang salah, selanjutnya menutup *server* agar tidak dapat diakses oleh orang yang tidak berhak maka digunakan teknik *port knocking* dengan ketukan yang benar.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 9125 3429 7324
hitting tcp 192.168.137.2:9125
hitting tcp 192.168.137.2:3429
hitting tcp 192.168.137.2:7324
root@ubuntu:/home/ubuntu# telnet 192.168.137.2
Trying 192.168.137.2...
```

Gambar 4. 88 Proses Menutup TELNET Ketukan Benar

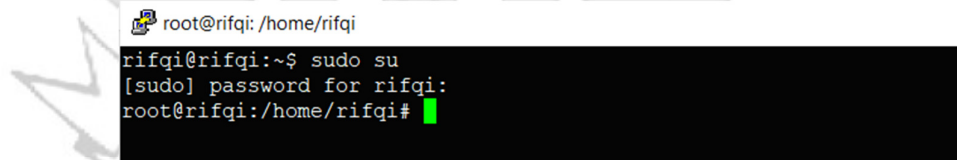
c. Langkah uji coba untuk membuka dan menutup *port 80* (HTTP).

- 1) Menjalankan aplikasi putty dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 89 Konfigurasi Putty

- 1) Tampilan ketika proses *login*, memasukkan *username* dan *password*.



Gambar 4. 90 Proses Login Putty

- 2) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2 tanpa menggunakan *firewall* dan *port knocking*.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875   Processes:            254
Usage of /:   30.9% of 9.75GB Users logged in:       1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

69 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet con
nection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 91 Proses *Login* Telah Berhasil Dilakukan

3) Tampilan ketika masuk ke level *user* yang lebih tinggi (super *user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

Gambar 4. 92 Masuk ke Super *User*

4) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses *ping* ke *user* dengan alamat IP 192.168.137.3 dan proses *ping* berhasil

```
root@rifqi: /home/rifqi
root@rifqi: /home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data.
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

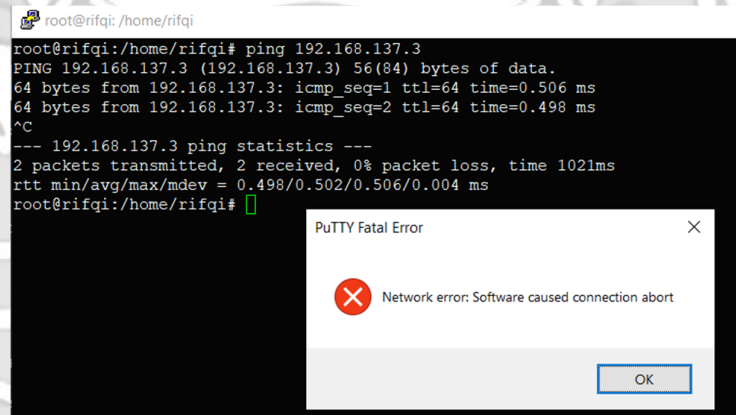
Gambar 4. 93 Proses PING

- 5) Setelah *server* berhasil terhubung dengan *melakukan* proses *ping* ke *user*, maka selanjutnya *server* di tutup total dengan cara mendrop semua akses sehingga tidak ada akses tcp yang dapat lewat. Menggunakan perintah iptables (*firewall*).

```
iptables -I INPUT -p tcp -j DROP
```

Gambar 4. 94 Mendrop Akses *Server*

Setelah mendrop semua akses pada *server* dengan perintah iptables maka *server* tidak dapat diakses lagi.



Gambar 4. 95 *Server* Tidak Dapat Diakses

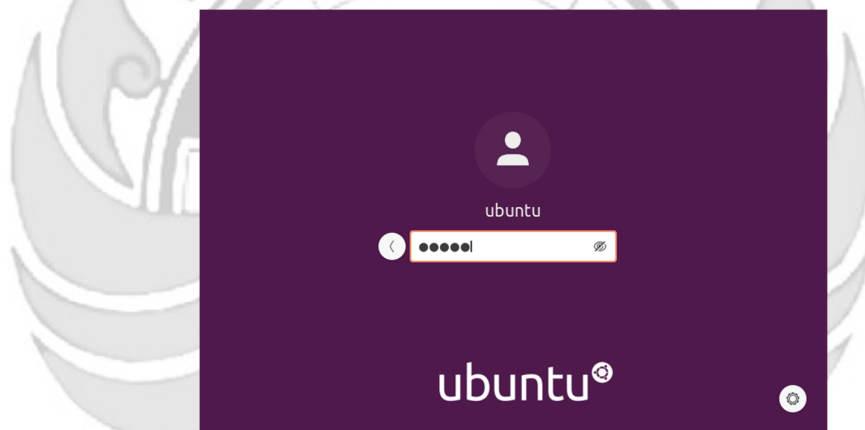
- 6) Setelah akses dari *server* ditutup, hanya terdapat satu metode yang dipakai untuk masuk ke sistem *server* dengan cara menggunakan metode *port knocking*. Untuk dapat mengakses *server* maka harus melalui *admin* yang sudah dilakukan penginstalan packet *knockd* agar dapat menggunakan metode *port knocking*.

```
root@ubuntu:/home/ubuntu# sudo systemctl status knockd
● knockd.service - Port-Knock Daemon
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor prese
   Active: active (running) since Tue 2023-08-15 06:32:15 PDT; 2s ago
     Docs: man:knockd(1)
    Main PID: 2592 (knockd)
      Tasks: 1 (limit: 4572)
     Memory: 656.0K
    CGroup: /system.slice/knockd.service
           └─2592 /usr/sbin/knockd -i ens33

Aug 15 06:32:15 ubuntu systemd[1]: Started Port-Knock Daemon.
Aug 15 06:32:15 ubuntu knockd[2592]: starting up, listening on ens33
lines 1-12/12 (END)
```

Gambar 4. 96 Pengecekan Status *Knockd*

- 7) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan *password*.



Gambar 4. 97 Proses *Login Admin*

- 8) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).

```
root@ubuntu: /home/ubuntu
ubuntu@ubuntu:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu:/home/ubuntu# S
```

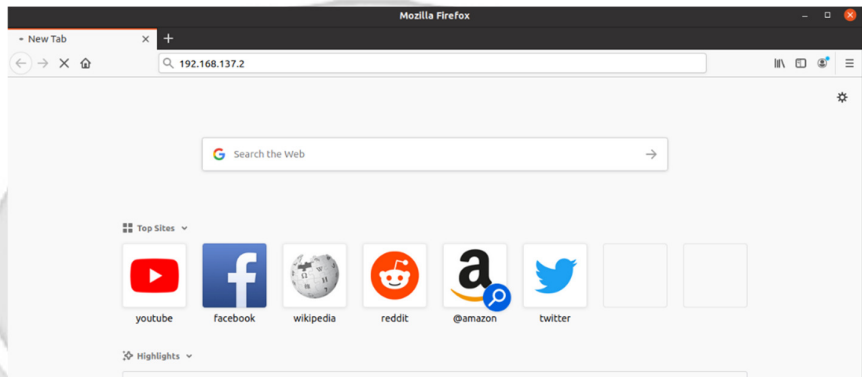
Gambar 4. 98 Masuk ke *Super User*

- 9) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port 80* (HTTP) menggunakan ketukan yang salah.

```
root@ubuntu:/home/ubuntu# knock -v 2889 4873 1200 7281
hitting tcp 0.0.11.73:4873
hitting tcp 0.0.11.73:1200
hitting tcp 0.0.11.73:7281
root@ubuntu:/home/ubuntu#
```

Gambar 4. 99 Proses Membuka HTTP Ketukan Salah

10) Maka ketika ketukan yang dimasukkan tidak sesuai untuk mengakses *port* 80 (HTTP) membuktikan bahwa hasil web tidak dapat diakses



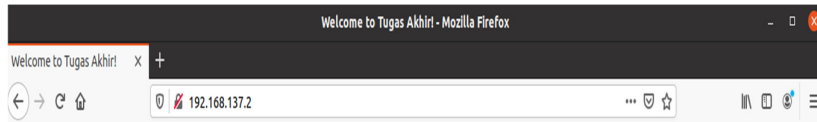
Gambar 4. 100 HTTP Tidak Dapat Diakses

11) Selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 80 (HTTP) menggunakan ketukan yang benar.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 2489 3872 1200 7381
hitting tcp 192.168.137.2:2489
hitting tcp 192.168.137.2:3872
hitting tcp 192.168.137.2:1200
hitting tcp 192.168.137.2:7381
root@ubuntu:/home/ubuntu#
```

Gambar 4. 101 Gambar Proses Membuka HTTP Ketukan Benar

12) Maka ketika ketukan yang dimasukkan sudah sesuai untuk mengakses *port* 80 (HTTP) membuktikan bahwa hasil web dapat diakses



Sukses Tugas Akhir, Lulus 2023!

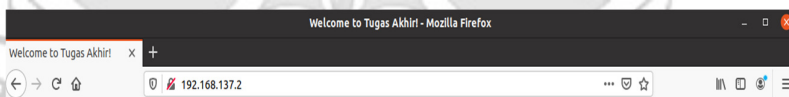
Gambar 4. 102 HTTP Dapat Diakses

- 13) Setelah dilakukan uji coba masuk ke *server* mengakses *port* 80 (HTTP) menggunakan teknik *port knocking* dengan ketukan yang benar selanjutnya dilakukan uji coba menutup *port* 80 (HTTP) yang terbuka dengan ketukan yang salah.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 7391 1500 3873
hitting tcp 192.168.137.2:7391
hitting tcp 192.168.137.2:1500
hitting tcp 192.168.137.2:3873
root@ubuntu:/home/ubuntu#
```

Gambar 4. 103 Proses Menutup HTTP Ketukan Salah

- 14) Maka ketika ketukan yang dimasukkan untuk menutup *port* 80 (HTTP) tidak sesuai, maka hasil web dari *port* 80 (HTTP) masih dapat diakses.



Sukses Tugas Akhir, Lulus 2023!

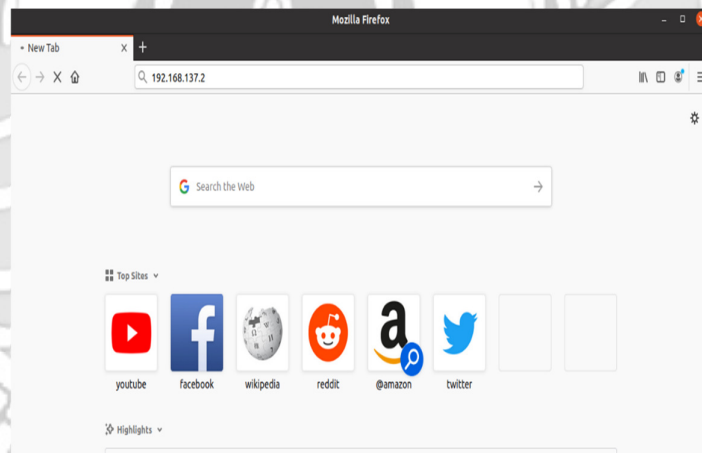
Gambar 4. 104 HTTP Masih Dapat Diakses

15) Setelah *port* 80 (HTTP) berhasil dibuka dan diakses oleh *admin* maka untuk mencegah agar *port* tidak dapat diakses oleh orang yang tidak berhak, maka *admin* menutup akses *port* 80 (HTTP) dengan ketukan yang benar menggunakan teknik *port* knocking.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 7381 1200 3872 2489
hitting tcp 192.168.137.2:7381
hitting tcp 192.168.137.2:1200
hitting tcp 192.168.137.2:3872
hitting tcp 192.168.137.2:2489
root@ubuntu:/home/ubuntu#
```

Gambar 4. 105 Proses Menutup HTTP Ketukan Benar

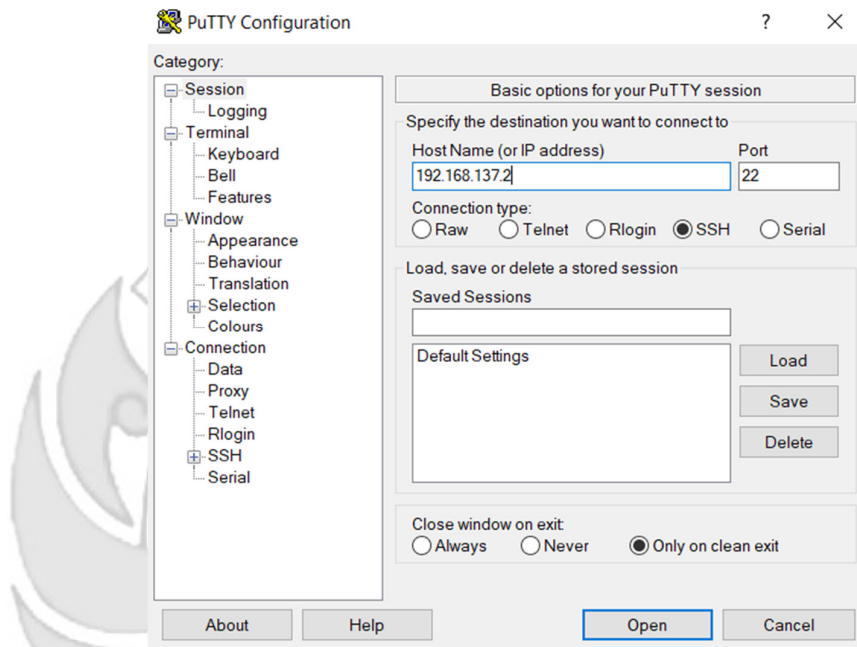
16) Maka ketika ketukan yang dimasukkan untuk menutup *port* 80 (HTTP) telah sesuai, maka hasil web dari *port* 80 (HTTP) tidak dapat diakses.



Gambar 4. 106 HTTP Tidak Dapat Diakses

d. Langkah uji coba untuk membuka dan menutup *port* 21 (FTP).

- 1) Menjalankan aplikasi putty dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 107 Konfigurasi Putty

- 2) Tampilan ketika proses *login*, memasukkan *username* dan *password*.



Gambar 4. 108 Proses Login Putty

- 3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2 tanpa menggunakan *firewall* dan *port knocking*.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875   Processes:            254
Usage of /:   30.9% of 9.75GB Users logged in:        1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 109 Proses *Login* Telah Berhasil Dilakukan

- 4) Tampilan ketika masuk ke level *user* yang lebih tinggi (super *user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi:/home/rifqi#
```

Gambar 4. 110 Masuk ke Super *User*

- 5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses *ping* ke *user* dengan alamat IP 192.168.137.3 dan proses *ping* berhasil.

```
root@rifqi: /home/rifqi
root@rifqi:/home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data:
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

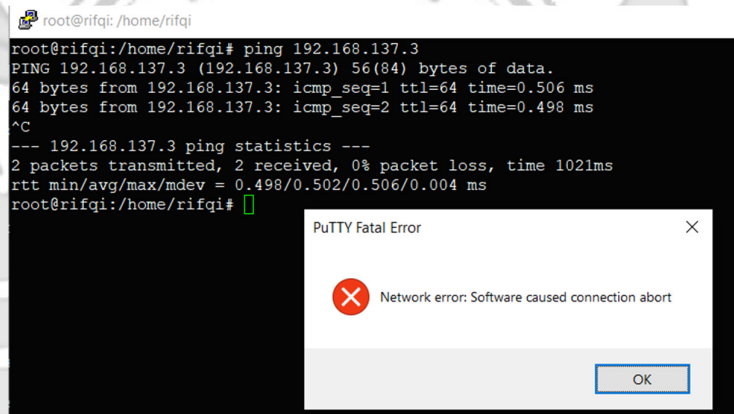
Gambar 4. 111 Proses PING

- 6) Setelah *server* berhasil terhubung dengan *melakukan* proses *ping* ke *user*, maka selanjutnya *server* di tutup total dengan cara mendrop semua akses sehingga tidak ada akses *tcp* yang dapat lewat. Menggunakan perintah *iptables* (*firewall*).

```
iptables -I INPUT -p tcp -j DROP
```

Gambar 4. 112 Mendrop Akses *Server*

Setelah mendrop semua akses pada *server* dengan perintah *iptables* maka *server* tidak dapat diakses lagi.



```
root@rifqi:/home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data:
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=0.506 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.498 ms
^C
--- 192.168.137.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1021ms
rtt min/avg/max/mdev = 0.498/0.502/0.506/0.004 ms
root@rifqi:/home/rifqi#
```

PuTTY Fatal Error
Network error: Software caused connection abort
OK

Gambar 4. 113 *Server* Tidak Dapat Diakses

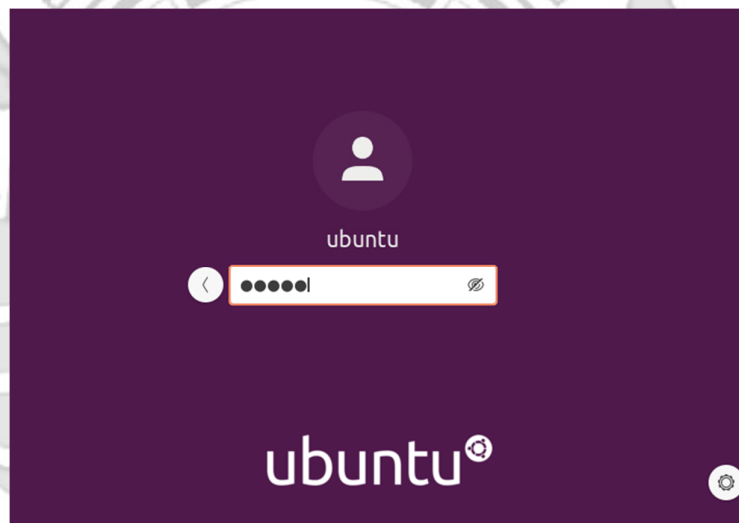
- 7) Setelah akses dari *server* ditutup, hanya terdapat satu metode yang dipakai untuk masuk ke sistem *server* dengan cara menggunakan metode *port knocking*. Untuk dapat mengakses *server* maka harus melalui *admin* yang sudah dilakukan penginstalan packet *knockd* agar dapat menggunakan metode *port knocking*.

```
root@ubuntu:/home/ubuntu# sudo systemctl status knockd
● knockd.service - Port-Knock Daemon
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor prese
   Active: active (running) since Tue 2023-08-15 06:32:15 PDT; 2s ago
     Docs: man:knockd(1)
    Main PID: 2592 (knockd)
      Tasks: 1 (limit: 4572)
     Memory: 656.0K
    CGroup: /system.slice/knockd.service
           └─2592 /usr/sbin/knockd -i ens33

Aug 15 06:32:15 ubuntu systemd[1]: Started Port-Knock Daemon.
Aug 15 06:32:15 ubuntu knockd[2592]: starting up, listening on ens33
lines 1-12/12 (END)
```

Gambar 4. 114 Pengecekan Satus *Knockd*

- 8) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan *password*.



Gambar 4. 115 Proses *Login Admin*

- 9) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).

```
root@ubuntu: /home/ubuntu
ubuntu@ubuntu:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu: /home/ubuntu# S
```

Gambar 4. 116 Masuk ke Super *User*

10) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 21 (FTP) menggunakan ketukan yang salah. Membuktikan bahwa *port* 21 (FTP) tidak dapat diakses secara bebas jika ketukan yang tidak sesuai dengan konfigurasi yang dilakukan di *server*.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 3893 4830 5290 2680
hitting tcp 192.168.137.2:3893
hitting tcp 192.168.137.2:4830
hitting tcp 192.168.137.2:5290
hitting tcp 192.168.137.2:2680
root@ubuntu:/home/ubuntu# ftp -p 192.168.137.2
ftp: connect: Connection timed out
ftp>
```

Gambar 4. 117 Proses Membuka FTP Ketukan Salah

11) Selanjutnya *admin* melakukan percobaan untuk mengakses *port* 21 (FTP) menggunakan ketukan yang benar. Membuktikan *port* 21 (FTP) dapat diakses jika ketukan sesuai dengan konfigurasi yang dilakukan di *server*.

```
root@ubuntu:/home/ubuntu# ftp -p 192.168.137.2
Connected to 192.168.137.2.
220 (vsFTPd 3.0.5)
Name (192.168.137.2:ubuntu): politeknik
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Gambar 4. 118 Proses Membuka FTP Ketukan Benar

12) Setelah dilakukan uji coba masuk ke *server* mengakses *port* 21 (FTP) menggunakan teknik *port knocking* dengan ketukan yang benar selanjutnya dilakukan uji coba menutup *port* 21 (FTP) yang terbuka dengan ketukan yang salah maka *server* tersebut masih dapat diakses

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 2690 5990 4820 1892
hitting tcp 192.168.137.2:2690
hitting tcp 192.168.137.2:5990
hitting tcp 192.168.137.2:4820
hitting tcp 192.168.137.2:1892
root@ubuntu:/home/ubuntu# ftp -p 192.168.137.2
Connected to 192.168.137.2.
220 (vsFTPD 3.0.5)
Name (192.168.137.2:ubuntu): politeknik
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Gambar 4. 119 Proses Menutup FTP Ketukan Salah

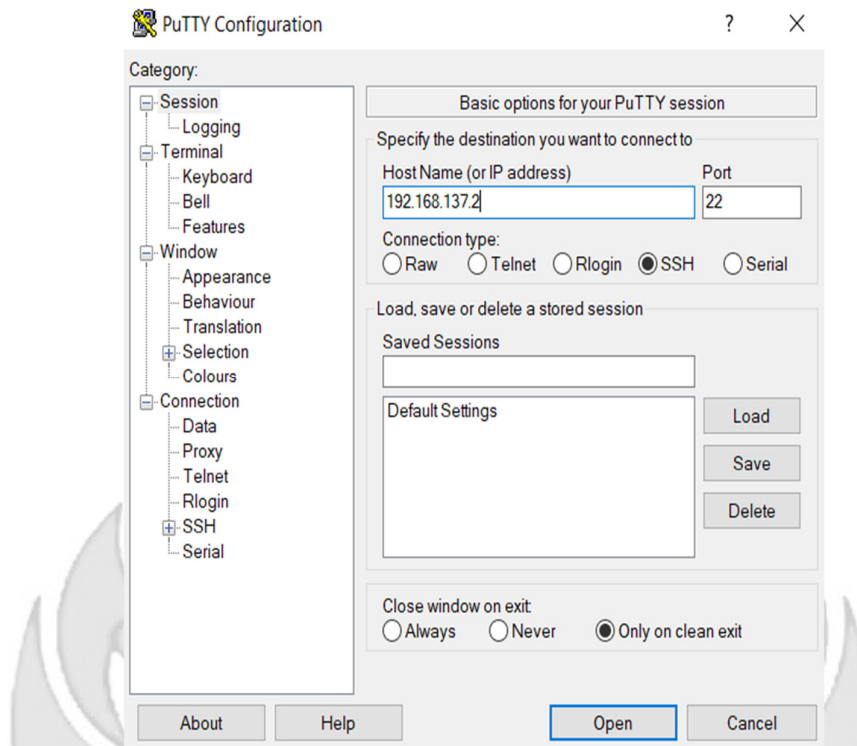
- 13) Setelah dilakukan uji coba menutup *port* 21 (FTP) dengan ketukan yang salah, selanjutnya menutup *port* 21 (FTP) agar tidak dapat diakses oleh orang yang tidak berhak maka digunakan teknik *port knocking* dengan ketukan yang benar.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 2680 5390 4820 3892
hitting tcp 192.168.137.2:2680
hitting tcp 192.168.137.2:5390
hitting tcp 192.168.137.2:4820
hitting tcp 192.168.137.2:3892
root@ubuntu:/home/ubuntu# ftp -p 192.168.137.2
ftp: connect: Connection timed out
ftp>
```

Gambar 4. 120 Proses Menutup FTP Ketukan Benar

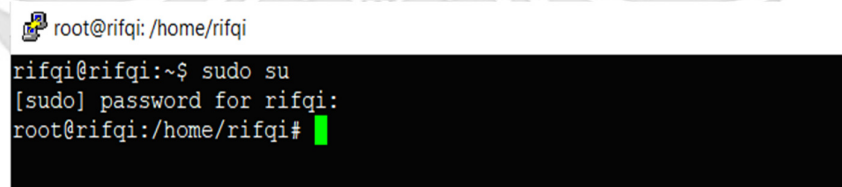
e. Langkah uji coba membuka dan menutup *port* 25 (SMTP).

- 1) Menjalankan aplikasi putty dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 121 Konfigurasi Putty

2) Tampilan ketika proses *login*, memasukkan *username* dan *password*.



Gambar 4. 122 Proses Login Putty

3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2 tanpa menggunakan *firewall* dan *port knocking*.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875   Processes:            254
Usage of /:   30.9% of 9.75GB Users logged in:        1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 123 Proses *Login* Telah Berhasil

- 4) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

Gambar 4. 124 Masuk ke Super *User*

- 5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses *ping* ke *user* dengan alamat IP 192.168.137.3 dan proses *ping* berhasil

```
root@rifqi: /home/rifqi
root@rifqi: /home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data:
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

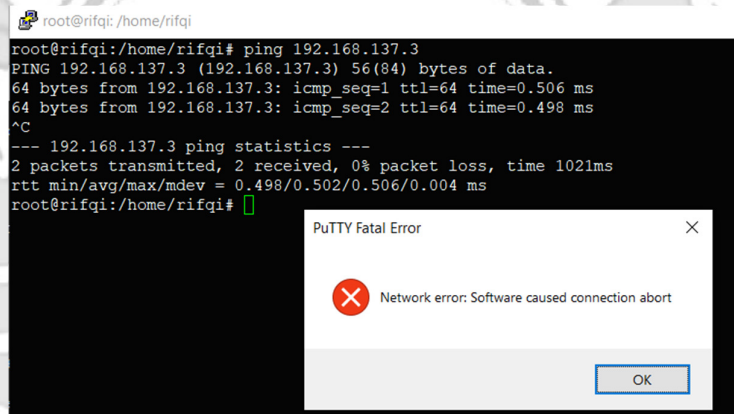
Gambar 4. 125 Proses PING

- 6) Setelah *server* berhasil terhubung dengan *melakukan* proses *ping* ke *user*, maka selanjutnya *server* di tutup total dengan cara mendrop semua akses sehingga tidak ada akses *tcp* yang dapat lewat. Menggunakan perintah *iptables* (*firewall*).

```
iptables -I INPUT -p tcp -j DROP
```

Gambar 4. 126 Mendrop Akses Server

Setelah mendrop semua akses pada *server* dengan perintah *iptables* maka *server* tidak dapat diakses lagi.



Gambar 4. 127 Akses Server di Drop

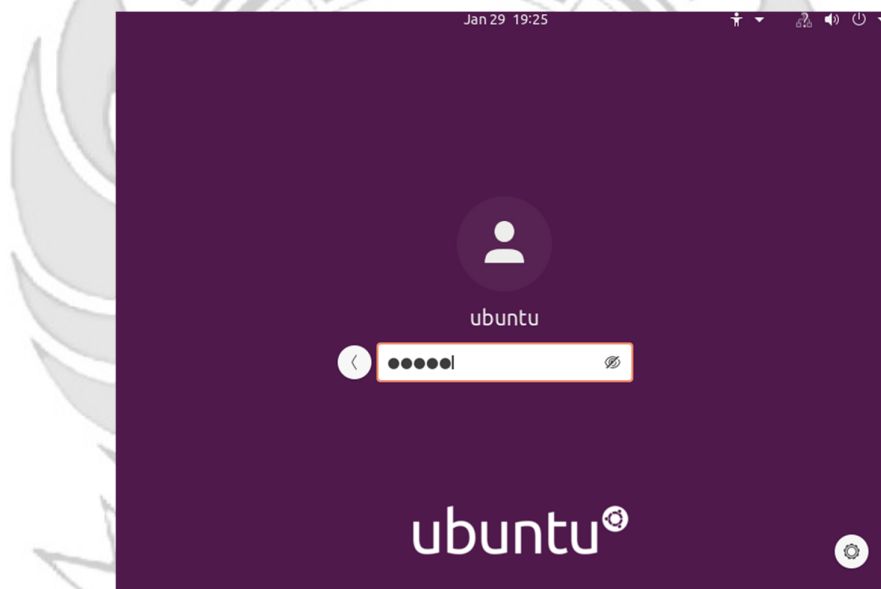
- 7) Setelah akses dari *server* ditutup, hanya terdapat satu metode yang dipakai untuk masuk ke sistem *server* dengan cara menggunakan metode *port knocking*. Untuk dapat mengakses *server* maka harus melalui *admin* yang sudah dilakukan penginstalan packet *knockd* agar dapat menggunakan metode *port knocking*.

```
root@ubuntu:/home/ubuntu# sudo systemctl status knockd
● knockd.service - Port-Knock Daemon
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor prese
   Active: active (running) since Tue 2023-08-15 06:32:15 PDT; 2s ago
     Docs: man:knockd(1)
    Main PID: 2592 (knockd)
      Tasks: 1 (limit: 4572)
     Memory: 656.0K
    CGroup: /system.slice/knockd.service
            └─2592 /usr/sbin/knockd -i ens33

Aug 15 06:32:15 ubuntu systemd[1]: Started Port-Knock Daemon.
Aug 15 06:32:15 ubuntu knockd[2592]: starting up, listening on ens33
lines 1-12/12 (END)
```

Gambar 4. 128 Pengecekan Status *Knockd*

- 8) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan *password*.



Gambar 4. 129 Proses *Login Admin*

- 9) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).

```
root@ubuntu: /home/ubuntu
ubuntu@ubuntu:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu: /home/ubuntu# S
```

Gambar 4. 130 Masuk *Super User*

10) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 25 (SMTP) menggunakan ketukan yang salah. Membuktikan bahwa *port* 25 (SMTP) tidak dapat diakses secara bebas jika ketukan yang tidak sesuai dengan konfigurasi yang dilakukan di *server*.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 1500 1600 1700 1800 1900
hitting tcp 192.168.137.2:1500
hitting tcp 192.168.137.2:1600
hitting tcp 192.168.137.2:1700
hitting tcp 192.168.137.2:1800
hitting tcp 192.168.137.2:1900
root@ubuntu:/home/ubuntu# telnet 192.168.137.2 25
Trying 192.168.137.2...
xxxxxxx
```

Gambar 4. 131 Proses Membuka SMTP Ketukan Salah

11) Selanjutnya *admin* melakukan percobaan untuk mengakses *port* 25 (SMTP) menggunakan ketukan yang benar. Membuktikan *port* 25 (SMTP) dapat diakses jika ketukan sesuai dengan konfigurasi yang dilakukan di *server*.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 1400 1500 1600 1700 1800
hitting tcp 192.168.137.2:1400
hitting tcp 192.168.137.2:1500
hitting tcp 192.168.137.2:1600
hitting tcp 192.168.137.2:1700
hitting tcp 192.168.137.2:1800
root@ubuntu:/home/ubuntu# telnet 192.168.137.2 25
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^]'.
220 rifqi ESMTP Postfix (Ubuntu)
```

Gambar 4. 132 Proses Membuka SMTP Ketukan Benar

12) Setelah dilakukan uji coba masuk ke *server* mengakses *port* 25 (SMTP) menggunakan teknik *port knocking* dengan ketukan yang benar selanjutnya dilakukan uji coba menutup *port* 21 (SMTP) yang terbuka dengan ketukan yang salah maka *server* tersebut masih dapat diakses.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 1900 1800 1700 1600 1500
hitting tcp 192.168.137.2:1900
hitting tcp 192.168.137.2:1800
hitting tcp 192.168.137.2:1700
hitting tcp 192.168.137.2:1600
hitting tcp 192.168.137.2:1500
root@ubuntu:/home/ubuntu# telnet 192.168.137.2 25
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^'.
```

Gambar 4. 133 Proses Menutup SMTP Ketukan Salah

- 13) Setelah dilakukan uji coba menutup *port 25* (SMTP) dengan ketukan yang salah, selanjutnya menutup *port 25* (SMTP) agar tidak dapat diakses oleh orang yang tidak berhak maka digunakan teknik *port knocking* dengan ketukan yang benar.

```
root@ubuntu:/home/ubuntu# knock -v 192.168.137.2 1800 1700 1600 1500 1400
hitting tcp 192.168.137.2:1800
hitting tcp 192.168.137.2:1700
hitting tcp 192.168.137.2:1600
hitting tcp 192.168.137.2:1500
hitting tcp 192.168.137.2:1400
root@ubuntu:/home/ubuntu# telnet 192.168.137.2 25
Trying 192.168.137.2...
xxxxxxxxxxx
```

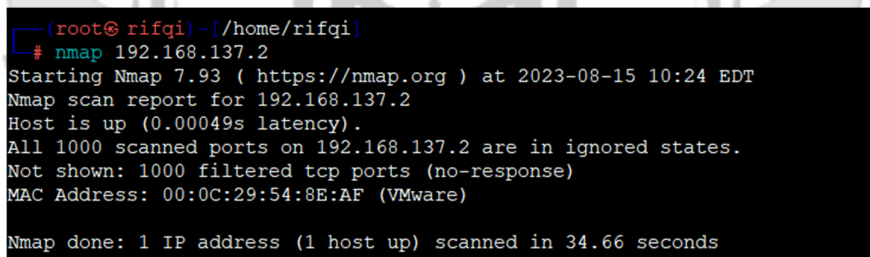
Gambar 4. 134 Proses Membuka SMTP Ketukan Benar

Admin tidak selamanya dapat mengakses *server* secara langsung, karena akan terdapat kondisi *admin* diberikan tugas keluar kota akan tetapi *admin* tetap diharuskan untuk mengakses *server* sehingga *admin* melakukan dengan cara *via remote*. Jika *admin* mengakses *server* secara *via remote* terdapat suatu celah keamanan yang dapat dimanfaatkan oleh *attacker* untuk melakukan penyadapan, maka untuk mengamankan *server* dari penyadapan, *admin* melakukan *remote server* dengan menerapkan metode *port knocking* pada *server* untuk mempersulit *attacker* mendapatkan informasi *port-port* apa saja dalam kondisi terbuka atau *port* apa saja yang sedang di *remote* oleh *admin*

Pada penelitian ini *attacker* melakukan peyadapan ketika *admin* melakukan *remote server* dengan menggunakan serangan *port scanning*. Serangan *port scanning* dilakukan untuk mengetahui informasi yang terdapat pada *server* seperti celah pada *port* tujuan terbuka atau tertutup. Pada tahap pengujian *port scanning* menggunakan tool NMAP (*Network Mapper*). Berikut penjelasan menggunakan serangan *port scanning* untuk mengetahui *port* tujuan terbuka atau tertutup.

a. *Port scanning port 22 (SSH)*

Pada tahap pengujian *port scanning* menggunakan tool *Network Mapper (NMAP)* dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port 22 (SSH)*. Pengujian ini dilakukan pada saat *port knocking* sesudah implementasi pada *server*.



```
(root@rifqi)~/home/rifqi
# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 10:24 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00049s latency).
All 1000 scanned ports on 192.168.137.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:0C:29:54:8E:AF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 34.66 seconds
```

Gambar 4. 135 Penyerangan *Port Scanning SSH* Setelah *Port knocking*

Terlihat pada Gambar 4. 139 bahwa setelah *server* menggunakan teknik *port knocking* menjadikan *port 22 (SSH)* dalam keadaan tertutup sehingga *attacker* tidak dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port 22 (SSH)*.

b. *Port Scanning port 23 (TELNET)*.

Pada tahap pengujian *port scanning* menggunakan tool *Network Mapper (NMAP)* dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port 23*

(TELNET). Pengujian ini dilakukan pada saat *port knocking* sesudah implementasi pada *server*.

```
(root@rifqi)~/home/rifqi
└─# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 10:24 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00049s latency).
All 1000 scanned ports on 192.168.137.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:0C:29:54:8E:AF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 34.66 seconds
```

Gambar 4. 136 Penyerangan *Port Scanning* TELNET Setelah *Port knocking*

Terlihat pada Gambar 4. 140 bahwa setelah *server* menggunakan teknik *port knocking* menjadikan *port* 23 (TELNET) dalam keadaan tertutup sehingga *attacker* tidak dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port* 23 (TELNET).

c. *Port Scanning port* 80 (HTTP)

Pada tahap pengujian *port scanning* menggunakan *tool Network Mapper* (NMAP) dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port* 80 (HTTP). Pengujian ini dilakukan pada saat *port knocking* sesudah implementasi pada *server*.

```
(root@rifqi)~/home/rifqi
└─# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 10:24 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00049s latency).
All 1000 scanned ports on 192.168.137.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:0C:29:54:8E:AF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 34.66 seconds
```

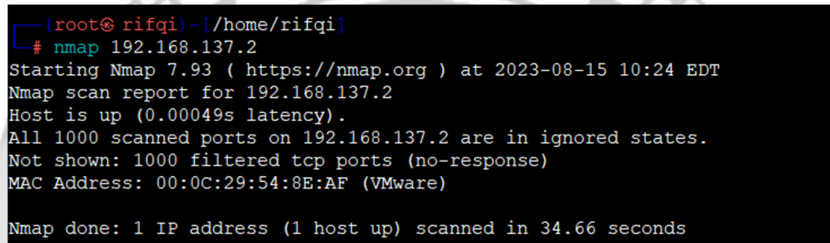
Gambar 4. 137 Penyerangan *Port Scanning* HTTP Setelah *Port knocking*

Terlihat pada Gambar 4. 141 bahwa setelah *server* menggunakan teknik *port knocking* menjadikan *port* 80 (HTTP) dalam keadaan tertutup sehingga *attacker*

tidak dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port* 80 (HTTP).

d. *Port Scanning port 21* (FTP)

Pada tahap pengujian *port scanning* menggunakan *tool Network Mapper* (NMAP) dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port* 21 (FTP). Pengujian ini dilakukan pada saat *port knocking* sesudah implementasi pada *server*.



```
(root@rifqi)~/home/rifqi
# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 10:24 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00049s latency).
All 1000 scanned ports on 192.168.137.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:0C:29:54:8E:AF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 34.66 seconds
```

Gambar 4. 138 Penyerangan *Port Scanning* FTP Setelah *Port knocking*

Terlihat pada Gambar 4. 142 bahwa setelah *server* menggunakan teknik *port knocking* menjadikan *port* 21 (FTP) dalam keadaan tertutup sehingga *attacker* tidak dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port* 21 (FTP).

e. *Scanning 25* (SMTP)

Pada tahap pengujian *scanning* menggunakan *tool Network Mapper* (NMAP) dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port* 25 (SMTP). Pengujian ini dilakukan pada saat *port knocking* sesudah implementasi pada *server*.

```
(root@rifqi)~/home/rifqi
# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 10:24 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00049s latency).
All 1000 scanned ports on 192.168.137.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:0C:29:54:8E:AF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 34.66 seconds
```

Gambar 4. 139 Penyerangan *Port Scanning* SMTP Setelah *Port knocking*

Terlihat pada Gambar 4. 143 bahwa setelah *server* menggunakan teknik *port knocking* menjadikan *port 25* (SMTP) dalam keadaan tertutup sehingga *attacker* tidak dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port 25* (SMTP).

Penerapan dengan metode *port knocking* pada *server* dapat mengatasi masalah sebelumnya karena dengan menggunakan *port knocking port-port* yang sedang *dirremote* oleh *admin* atau *port-port* yang berada dalam kondisi terbuka tidak dapat diketahui oleh *attacker* walaupun menggunakan serangan *port scanning*, dengan ini maka tidak ada informasi yang didapatkan *attacker* untuk mengakses *server* secara bebas.

Setelah melakukan serangan dengan menggunakan metode *port scanning* selanjutnya *attacker* melakukan serangan dengan dengan tingkat yang lebih tinggi untuk mendapatkan informasi yang lebih banyak maka *attacker* melakukan serangan dengan metode *sniffing* menggunakan *wireshark*.

a. *Sniffing port 22* (SSH)

Pada tahap pengujian penyerangan *sniffing* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 22* (SSH) untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

3. ketukan yang benar.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.137.3	192.168.137.2	UDP	60	46969 → 3647 Len=1
2	0.000164	192.168.137.1	192.168.137.3	ICMP	71	Redirect (Redirect for network)
3	0.000197	192.168.137.3	192.168.137.2	UDP	43	46969 → 3647 Len=1
4	0.000747	192.168.137.3	192.168.137.2	TCP	74	33286 → 6029 [SYN] Seq=0 Win=6
5	0.000783	192.168.137.3	192.168.137.2	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 33286 → 6029 [SYN] Seq=0 Win=6
6	0.001733	192.168.137.3	192.168.137.2	UDPCAP	60	[Malformed Packet]
7	0.001792	192.168.137.3	192.168.137.2	UDPCAP	43	[Malformed Packet]
8	2.309999	192.168.137.3	192.168.137.2	TCP	74	44478 → 22 [SYN] Seq=0 Win=642
9	2.310051	192.168.137.1	192.168.137.3	ICMP	102	Redirect (Redirect for network)
10	2.310097	192.168.137.3	192.168.137.2	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 44478 → 22 [SYN] Seq=0 Win=642
11	2.310373	192.168.137.2	192.168.137.3	TCP	74	22 → 44478 [SYN, ACK] Seq=0 Ack=44478 Win=642 Len=0
12	2.310517	192.168.137.1	192.168.137.2	ICMP	102	Redirect (Redirect for network)
13	2.310545	192.168.137.2	192.168.137.3	TCP	74	[TCP Retransmission] 22 → 44478 [SYN, ACK] Seq=0 Ack=44478 Win=642 Len=0
14	2.310631	192.168.137.3	192.168.137.2	TCP	66	44478 → 22 [ACK] Seq=1 Ack=1 Win=0 Len=0

Gambar 4. 140 Sequence SSH

Wireshark - Packet 1 - 1. membuka ssh.pcapng

```

> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{3A14AAFF-6E05-47E1-8175-E031668AA183}, id 0
> Ethernet II, Src: VMWare_0c:48:50 (00:0c:29:0c:48:50), Dst: VMWare_a2:46:1c (00:0c:29:a2:46:1c)
> Internet Protocol Version 4, Src: 192.168.137.3, Dst: 192.168.137.2
> Transmission Control Protocol, Src Port: 48278, Dst Port: 22, Seq: 0, Len: 0

```

```

0000  00 0c 29 a2 46 1c 00 0c 29 0c 48 50 08 00 45 00  ..F...  )HP..E.
0010  00 3c eb 9c 40 00 06 bb c8 e0 a8 89 03 c0 a8  <..@ @  ....
0020  89 02 bc 96 00 16 51 7a 13 ac 00 00 00 a0 02  .....Qz .....
0030  50 40 70 0b 00 00 00 00 0f 54 00 00 00 00 00  .....

```

Gambar 4. 141 IP Server

8	2.309999	192.168.137.3	192.168.137.2	TCP	74	44478 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=782050
9	2.310051	192.168.137.1	192.168.137.3	ICMP	102	Redirect (Redirect for network)
10	2.310097	192.168.137.3	192.168.137.2	TCP	74	[TCP Retransmission] [TCP Port numbers reused] 44478 → 22 [SYN] Seq=0
11	2.310373	192.168.137.2	192.168.137.3	TCP	74	22 → 44478 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM T

Gambar 4. 142 Port SSH

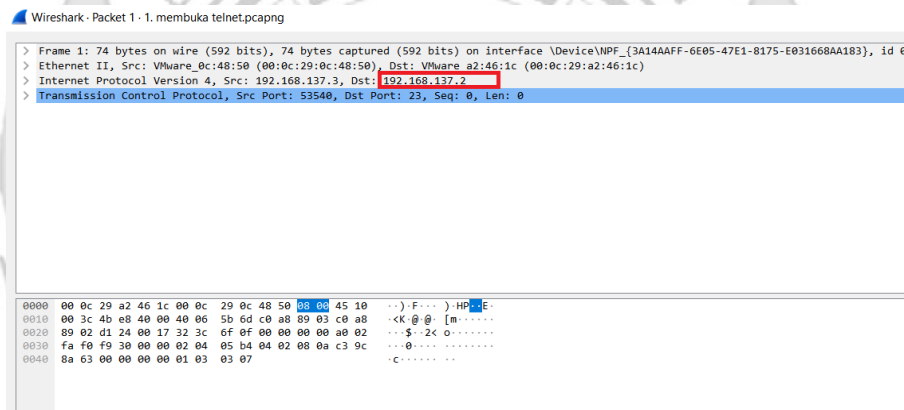
Terlihat pada Gambar 4. 144, 4. 4. 145 dan 4. 146 bahwa ketika seorang *admin* jaringan melakukan *remote server* pada *port 22* (SSH) dengan teknik *port knocking* maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *sequence* dan *port* yang diketuk oleh *admin* jaringan, serta ip dari *server*, sehingga dengan informasi-informasi yang didapatkan maka *attacker* dapat mengakses *server*.

b. *Sniffing port 23 (TELNET)*

Pada tahap pengujian penyerangan *sniffing* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 23 (TELNET)* untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

1	0.000000	192.168.137.3	192.168.137.2	UDP	60 60773 → 7324	Len=1
2	0.000141	192.168.137.1	192.168.137.3	ICMP	71 Redirect	(Redirect for network)
3	0.000173	192.168.137.3	192.168.137.2	UDP	43 60773 → 7324	Len=1
4	0.001452	192.168.137.3	192.168.137.2	TCP	74 41104 → 3429 [SYN]	Seq=0 Win=64240 Len=0 MSS=1

Gambar 4. 143 Sequence TELNET



Gambar 4. 144 IP Server

25	10.310975	192.168.137.3	192.168.137.2	TCP	74 47638 → 23 [SYN]	Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=783918347 TSecr=0
26	10.310955	192.168.137.1	192.168.137.3	ICMP	102 Redirect	(Redirect for network)
27	10.310906	192.168.137.3	192.168.137.2	TCP	74 [TCP Retransmission]	[TCP Port numbers reused] 47638 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=783918347 TSecr=0

Gambar 4. 145 Port TELNET

Terlihat pada Gambar 4. 147, 4. 148 dan 4. 149 bahwa ketika seorang *admin* jaringan melakukan *remote server* pada *port 23 (TELNET)* dengan teknik *port knocking* maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *sequence* dan *port* yang diketuk oleh *admin* jaringan, serta ip dari *server*, sehingga dengan informasi-informasi yang didapatkan maka *attacker* dapat mengakses *server*.

c. *Sniffing port 80 (HTTP)*

Pada tahap pengujian penyerangan *sniffing* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 80 (HTTP)* untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

1	0.000000	192.168.137.3	192.168.137.2	UDP	60 37805 → 7381	Len=1
2	0.000050	192.168.137.1	192.168.137.3	ICMP	71 Redirect	(Redirect
3	0.000081	192.168.137.3	192.168.137.2	UDP	43 37805 → 7381	Len=1
4	0.001129	192.168.137.3	192.168.137.2	UDP	60 48948 → 1200	Len=1
5	0.001162	192.168.137.3	192.168.137.2	UDP	43 48948 → 1200	Len=1
6	0.002138	192.168.137.3	192.168.137.2	UDP	60 38873 → 3872	Len=1
7	0.002163	192.168.137.3	192.168.137.2	UDP	43 38873 → 3872	Len=1
8	0.003141	192.168.137.3	192.168.137.2	UDP	60 53587 → 2489	Len=1
9	0.003172	192.168.137.3	192.168.137.2	UDP	43 53587 → 2489	Len=1
10	3.897992	192.168.137.3	216.239.38.120	QUIC	1399 Initial, DCID=5495dbd9c05a70ef,	
11	3.900106	192.168.137.3	216.239.38.120	TLSv1.2	155 Application Data	
12	4.002010	216.239.38.120	192.168.137.3	QUIC	1399 Handshake, DCID=417e68, SCID=d45	

Gambar 4. 146 Sequence HTTP

Wireshark - Packet 8 - 3. ketukan yang benar.pcapng

```

> Frame 8: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{3A14AAFF-6E05-47E1-8175-E031668AA183},
> Ethernet II, Src: VMware_0c:48:50 (00:0c:29:0c:48:50), Dst: VMware_e3:85:d9 (00:0c:29:e3:85:d9)
> Internet Protocol Version 4, Src: 192.168.137.3, Dst: 192.168.137.2
> User Datagram Protocol, Src Port: 53587, Dst Port: 2489
> Data (1 byte)
  
```

Gambar 4. 147 IP Server

5	0.192324	192.168.137.3	202.67.36.152	TCP	74 58656 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1
6	0.224387	202.67.36.152	192.168.137.3	TCP	74 80 → 58656 [SYN, ACK] Seq=0 Ack=1 Win=65160
7	0.238976	192.168.137.3	202.67.36.152	TCP	66 58656 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0

Gambar 4. 148 Port HTTP

Terlihat pada Gambar 4. 150, 4. 151 dan 4. 152 bahwa ketika seorang *admin* jaringan melakukan *remote server* pada *port 80 (HTTP)* dengan teknik *port knocking* maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *sequence* dan *port* yang diketuk oleh *admin* jaringan,

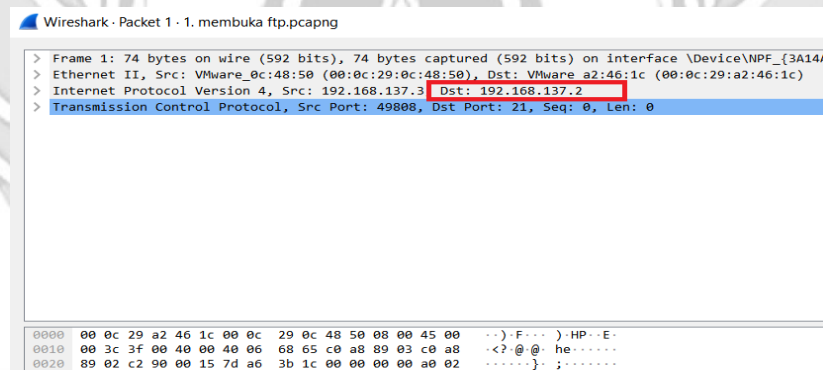
serta ip dari *server*, sehingga dengan informasi-informasi yang didapatkan maka *attacker* dapat mengakses *server*.

d. *Sniffing port 21* (FTP)

Pada tahap pengujian penyerangan *sniffing* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 21* (FTP) untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

1	0.000000	192.168.137.3	192.168.137.2	TCP	74	52382 → 3890 [SYN] Seq=0
2	0.000310	192.168.137.3	192.168.137.2	TCP	74	45374 → 4890 [SYN] Seq=0
3	0.000537	192.168.137.3	192.168.137.2	TCP	74	50336 → 5890 [SYN] Seq=0
4	0.001535	192.168.137.3	192.168.137.2	TCP	74	45146 → 6890 [SYN] Seq=0
5	3.065514	192.168.137.3	192.168.137.2	TCP	74	49832 → 21 [SYN] Seq=0 W
6	3.065811	192.168.137.2	192.168.137.3	TCP	74	21 → 49832 [SYN, ACK] Se
7	3.066085	192.168.137.3	192.168.137.2	TCP	66	49832 → 21 [ACK] Seq=1 A
8	3.174950	192.168.137.2	192.168.137.3	FTP	86	Response: 220 (vsFTPd 3.
9	3.175730	192.168.137.3	192.168.137.2	TCP	66	49832 → 21 [ACK] Seq=1 A

Gambar 4. 149 Sequence FTP



Gambar 4. 150 IP Server

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.137.3	192.168.137.2	TCP	74	49808 → 21 [SYN] Seq=0 Win=64240
2	0.001066	192.168.137.2	192.168.137.3	TCP	74	21 → 49808 [SYN, ACK] Seq=0 Ack=1

Gambar 4. 151 Port FTP

Terlihat pada Gambar 4. 153, 4. 154 dan 4. 155 bahwa ketika seorang *admin* jaringan melakukan *remote server* pada *port 21* (FTTP) dengan teknik *port knocking* maka seorang *attacker* dengan menggunakan metode *sniffing* dapat

dengan mudah mengetahui *sequence* dan *port* yang diketuk oleh *admin* jaringan, serta ip dari *server*, sehingga dengan informasi-informasi yang didapatkan maka *attacker* dapat mengakses *server*.

1	0.000000	192.168.137.3	192.168.137.2	UDP	60 34910 → 1400	en=1
2	0.000084	192.168.137.1	192.168.137.3	ICMP	71 Redirec	(Re
3	0.000119	192.168.137.3	192.168.137.2	UDP	43 34910 → 1400	en=1
4	0.000713	192.168.137.3	192.168.137.2	UDP	60 60330 → 1500	en=1
5	0.000737	192.168.137.3	192.168.137.2	UDP	43 60330 → 1500	en=1
6	0.000928	192.168.137.3	192.168.137.2	UDP	60 47965 → 1600	en=1
7	0.000942	192.168.137.3	192.168.137.2	UDP	43 47965 → 1600	en=1
8	0.002101	192.168.137.3	192.168.137.2	UDP	60 52986 → 1700	en=1
9	0.002176	192.168.137.3	192.168.137.2	UDP	43 52986 → 1700	en=1
10	0.002754	192.168.137.3	192.168.137.2	UDP	60 35305 → 1800	en=1
11	0.002812	192.168.137.3	192.168.137.2	UDP	43 35305 → 1800	en=1
12	2.856960	192.168.137.1	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1.1	
13	3.858963	192.168.137.1	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1.1	

Gambar 4. 152 Sequence SMTP

e. *Sniffing port 25 (SMTP)*

Pada tahap pengujian penyerangan *sniffing* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 25 (SMTP)* untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

Wireshark · Packet 1 · 1. membuka port.pcapng

```

> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{3A14AAFF-6E
> Ethernet II, Src: VMware_0c:48:50 (00:0c:29:0c:48:50), Dst: VMware_a2:46:1c (00:0c:29:a2:46:1c)
> Internet Protocol Version 4, Src: 192.168.137.3, Dst: 192.168.137.2
> Transmission Control Protocol, Src Port: 40602, Dst Port: 25, Seq: 0, Len: 0
  
```

```

0000  00 0c 29 a2 46 1c 00 0c 29 0c 48 50 08 0c 45 10  ..).F...)-HP..E.
0010  00 3c c1 3e 40 00 40 06 e6 16 c0 a8 89 03 c0 a8  -<.>@. ....
0020  89 02 9e 9a 00 19 78 e7 d0 20 00 00 00 00 a0 02  ....X. ....
0030  fa f0 26 c7 00 00 02 04 05 b4 04 02 08 0a c3 ad  ..&.....
0040  e7 87 00 00 00 01 03 03 07  ..
  
```

Gambar 4. 153 IP Server

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.137.3	192.168.137.2	TCP	74	40602 → 25 [SYN] Seq=0 Win=
2	0.000286	192.168.137.2	192.168.137.3	TCP	74	25 → 40602 [SYN, ACK] Seq=

Gambar 4. 154 Port SMTP

Terlihat pada Gambar 4. 156, 4. 157 dan 4. 158 bahwa ketika seorang *admin* jaringan melakukan *remote server* pada *port 25* (SMTP) dengan teknik *port knocking* maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *sequence* dan *port* yang diketuk oleh *admin* jaringan, serta *ip* dari *server*, sehingga dengan informasi-informasi yang didapatkan maka *attacker* dapat mengakses *server*.

Penggunaan metode *port knocking* pada *server* untuk mempersulit *attacker* melakukan penyadapan masih terdapat kelemahan jika *attacker* menggunakan serangan dengan tingkat yang lebih tinggi yaitu menggunakan metode *sniffing*. Serangan dengan menggunakan metode *sniffing* *attacker* mendapatkan informasi-informasi penting yaitu *ip server*, *port server* serta *sequence plaintext*. Informasi *sequence port* yang berbentuk *plain text* tersebut maka *attacker* dapat dengan mudah memahami *sequence* yang digunakan untuk membuka *port* yang telah ditutup dengan metode *port knocking*, maka dari celah keamanan tersebut dibutuhkan sebuah sistem keamanan yang dapat melindungi *server* agar *attacker* tidak dapat dengan mudah memahami *sequence port* yang digunakan untuk membuka *port*.

4.1.3 Pengujian Server Menggunakan Port knocking dan Algoritma XTEA

Jika seorang *admin* melakukan *remote server* pada *port-port* yang terdapat pada *server* yaitu SSH, TELNET, HTTP, FTP dan SMTP dengan menerapkan metode *port knocking* pada *server* masih terdapat kelemahan *sequence port* yang di

remote oleh *admin* masih berbentuk *plaintext* sehingga jika terdapat penyadapan *attacker* dapat dengan mudah memahami *sequence port* yang sedang *remote* oleh *server*.

Pada permasalahan yang telah dijelaskan maka penelitian ini memberikan tingkat keamanan pada *port knocking* dengan menerapkan algoritma XTEA pada *port knocking* untuk mengamankan *sequence port* sehingga tidak mudah ntuk dipahami *attacker*.

Sebelum menerapkan algoritma XTEA pada *port knocking* ketukan yang dilakukan oleh *client* langsung dikirim ke *server* untuk autentikasi dengan *knock port*. Jika urutan ketukan benar maka *server* memberikan ijin kepada *client* untuk dapat mengakses *port* tersebut, tetapi apabila urutan salah maka *client* tidak dapat mengakses *port* tersebut. Kemudian setelah memberikan tingkat keamanan pada *port knocking* dengan menggunakan algoritma xtea ketukan yang dilakukan oleh *client* di enkripsi terlebih dahulu kemudian dikirim ke *server* untuk autentikasi dengan *knock port*. Jika urutan ketukan benar maka *server* memberikan ijin kepada *client* untuk dapat mengakses *port* tersebut, tetapi apabila urutan salah maka *client* tidak dapat mengakses *port* ters

Pemrograman algoritma XTEA dalam mengamankan sebuah *sequence port* menggunakan bahasa *python*. Adapun proses utama pada sistem ini adalah melakukan enkripsi pada *sequence port* agar berbentuk *chiphertext* sehingga *attacker* sulit untuk memahami *sequence* yang digunakan oleh *admin* ketika melakukan *remote* terhadap *server*.

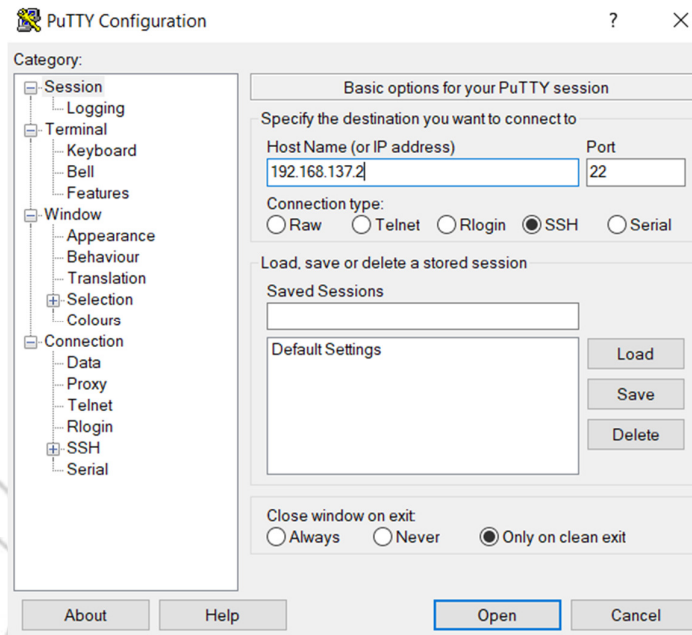
1. Konfigurasi *Port knocking* pada server: “/etc/knockd.conf”

```
GNU nano 6.2 /etc/knockd.conf
[options]
UseSyslog
[openSSH]
sequence = 3647,6029,4500
seq_timeout = 5
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags = syn
[closeSSH]
sequence = 4500,6029,3647
seq_timeout = 5
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags = syn
[openHTTP]
sequence = 2489,3872,1200,7381
seq_timeout = 5
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 80 -j ACCEPT
tcpflags = syn
[closeHTTP]
sequence = 7381,1200,3872,2489
seq_timeout = 5
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 80 -j ACCEPT
tcpflags = syn
[openFTP]
sequence = 3892,4820,5390,2680
seq_timeout = 5
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 21 -j ACCEPT
tcpflags = syn
[closeFTP]
sequence = 2680,5390,4820,3892
seq_timeout = 5
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 21 -j ACCEPT
tcpflags = syn
[openSMTP]
sequence = 1400,1500,1600,1700,1800
seq_timeout = 5
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 25 -j ACCEPT
tcpflags = syn
[closeSMTP]
sequence = 1800,1700,1600,1500,1400
seq_timeout = 5
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 25 -j ACCEPT
tcpflags = syn
[openTELNET]
sequence = 7324,3429,9125
seq_timeout = 5
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 23 -j ACCEPT
tcpflags = syn
[closeSSH]
sequence = 9125,3429,7324
seq_timeout = 5
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 23 -j ACCEPT
tcpflags = syn
```

Gambar 4. 155 Konfigurasi *Port knocking*

a. Langkah Uji Coba membuka dan menutup *port 22* (SSH). Pada uji coba yang dilakukan menggunakan aplikasi *putty* untuk membuka *server*.

- 1) Menjalankan aplikasi *putty* dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 156 Konfigurasi Putty

- 2) Tampilan ketika proses *login*, memasukkan *username* dan *password*.



Gambar 4. 157 Proses Login Putty

- 3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2 tanpa menggunakan *firewall* dan *port knocking*.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875      Processes:           254
Usage of /:   30.9% of 9.75GB   Users logged in:    1
Memory usage: 10%             IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 158 Proses *Login* Telah Berhasil Dilakukan

- 4) Tampilan ketika masuk ke level *user* yang lebih tinggi (super *user*).

```
root@ubuntu: /home/ubuntu

ubuntu@ubuntu:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu: /home/ubuntu# S
```

Gambar 4. 159 Masuk Super *User*

- 5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses *ping* ke *user* dengan alamat IP 192.168.137.3 dan proses *ping* berhasil.

```
root@rifqi: /home/rifqi

root@rifqi: /home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data.
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

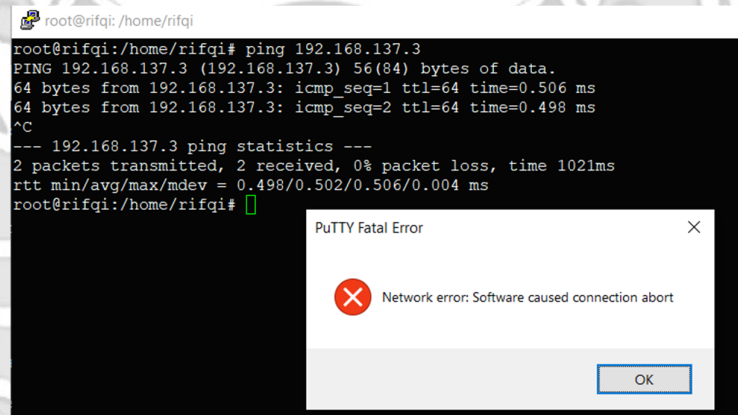
Gambar 4. 160 Proses PING

- 6) Setelah *server* berhasil terhubung dengan *melakukan* proses *ping* ke *user*, maka selanjutnya *server* di tutup total dengan cara mendrop semua akses sehingga tidak ada akses tcp yang dapat lewat. Menggunakan perintah iptables (*firewall*).

```
iptables -I INPUT -p tcp -j DROP
```

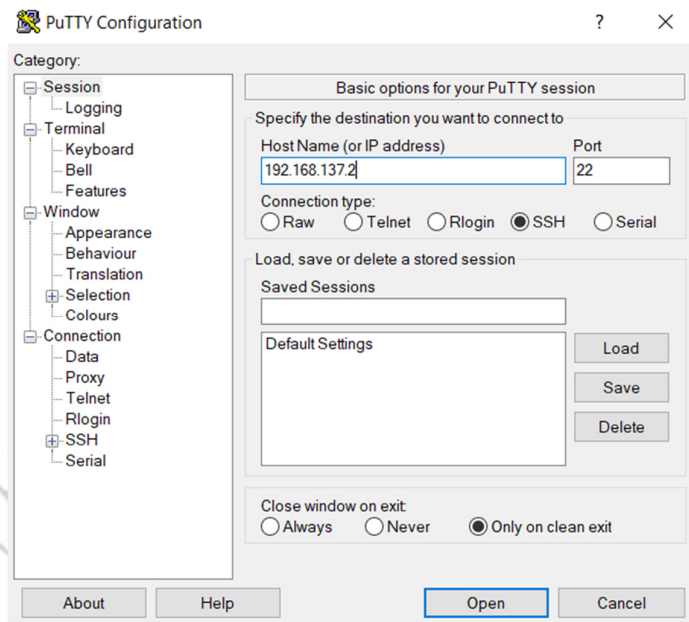
Gambar 4. 161 Mendrop Akses *Server*

Setelah mendrop semua akses pada *server* dengan perintah iptables maka *server* tidak dapat diakses lagi.



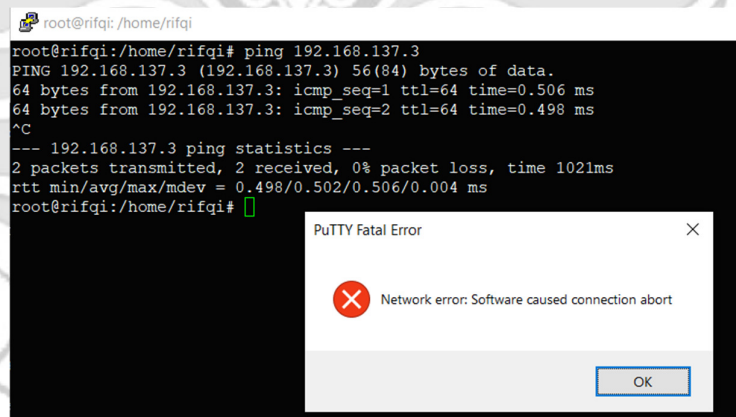
Gambar 4. 162 Akses *Server* di Drop

- 7) Setelah *server* berhasil ditutup dengan perintah iptables maka akan dilakukan uji coba kembali masuk ke *server* yang dituju yaitu IP 192.168.137.2 dengan menggunakan aplikasi putty dan membuktikan bahwa *server* tersebut sudah ditutup sehingga tidak dapat diakses secara bebas.



Gambar 4. 163 Konfigurasi Putty

- 8) Hasil ketika *server* terbukti berhasil ditutup sehingga *server* tidak dapat diakses secara bebas.



Gambar 4. 164 Server Tidak Dapat Diakses

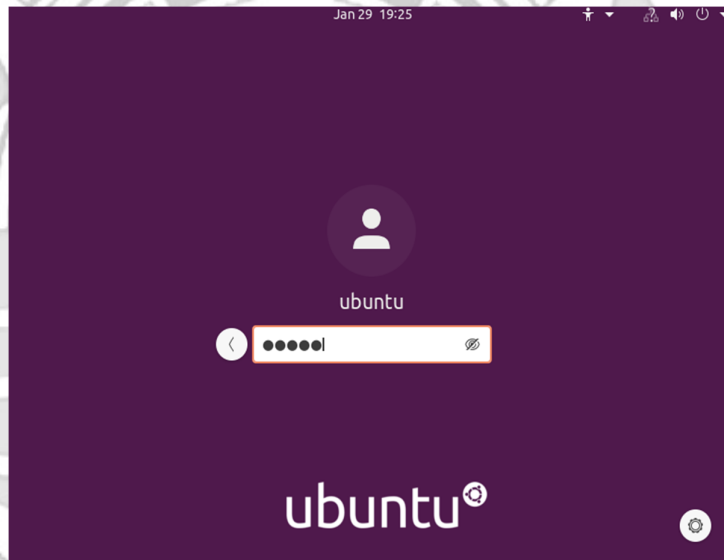
Setelah akses dari *server* ditutup, hanya terdapat satu metode yang dipakai untuk masuk ke sistem *server* dengan cara menggunakan metode *port knocking*. Untuk dapat mengakses *server* maka harus melalui *admin* yang sudah dilakukan penginstalan packet *knockd* agar dapat menggunakan metode *port knocking*.

```
root@ubuntu:/home/ubuntu# sudo systemctl status knockd
● knockd.service - Port-Knock Daemon
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor prese
   Active: active (running) since Tue 2023-08-15 06:32:15 PDT; 2s ago
     Docs: man:knockd(1)
   Main PID: 2592 (knockd)
    Tasks: 1 (limit: 4572)
   Memory: 656.0K
    CGroup: /system.slice/knockd.service
           └─2592 /usr/sbin/knockd -i ens33

Aug 15 06:32:15 ubuntu systemd[1]: Started Port-Knock Daemon.
Aug 15 06:32:15 ubuntu knockd[2592]: starting up, listening on ens33
lines 1-12/12 (END)
```

Gambar 4. 165 Pengecekan Status *Knockd*

- 9) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan *password*.



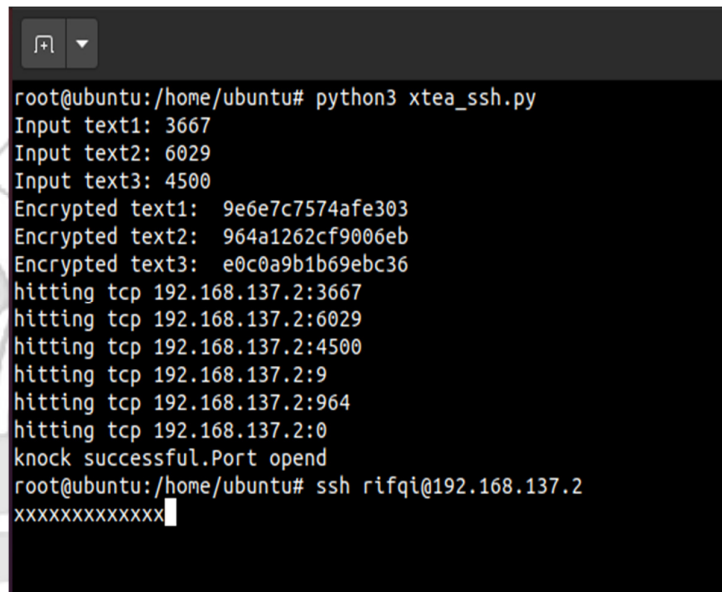
Gambar 4. 166 Proses *Login Admin*

- 10) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).

```
root@rifqi:/home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi:/home/rifqi#
```

Gambar 4. 167 Masuk ke Super *User*

11) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 22 (SSH) menggunakan ketukan yang salah. Membuktikan bahwa *server* tidak dapat diakses menggunakan *port* 22 (SSH) secara bebas jika ketukan tidak sesuai dengan konfigurasi yang dilakukan di *server*.

A terminal window with a dark background and white text. The prompt is root@ubuntu:/home/ubuntu#. The user runs python3 xtea_ssh.py. The script prompts for three text inputs: 3667, 6029, and 4500. It then outputs three encrypted strings: 9e6e7c7574afe303, 964a1262cf9006eb, and e0c0a9b1b69ebc36. Next, it sends a series of TCP connection attempts to 192.168.137.2 on ports 3667, 6029, 4500, 9, 964, and 0. The output shows 'knock successful.Port open'. Finally, the user runs ssh rifqi@192.168.137.2, and the terminal shows a series of 'x' characters.

```
root@ubuntu:/home/ubuntu# python3 xtea_ssh.py
Input text1: 3667
Input text2: 6029
Input text3: 4500
Encrypted text1: 9e6e7c7574afe303
Encrypted text2: 964a1262cf9006eb
Encrypted text3: e0c0a9b1b69ebc36
hitting tcp 192.168.137.2:3667
hitting tcp 192.168.137.2:6029
hitting tcp 192.168.137.2:4500
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:964
hitting tcp 192.168.137.2:0
knock successful.Port open
root@ubuntu:/home/ubuntu# ssh rifqi@192.168.137.2
xxxxxxxxxxxxx
```

Gambar 4. 168 Proses Membuka SSH Ketukan Salah

12) Selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 22 (SSH) menggunakan ketukan yang benar. Membuktikan bahwa *server* dapat diakses menggunakan *port* 22 (SSH) jika ketukan yang sesuai dengan konfigurasi yang dilakukan di *server*.

```
root@ubuntu:/home/ubuntu# python3 xtea_ssh.py
Input text1: 3647
Input text2: 6029
Input text3: 4500
Encrypted text1: 9e6e7c7574afe111
Encrypted text2: 2c84fba363a076d1
Encrypted text3: 62b91c9f1adb328
hitting tcp 192.168.137.2:3647
hitting tcp 192.168.137.2:6029
hitting tcp 192.168.137.2:4500
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:2
hitting tcp 192.168.137.2:62
knock successful.Port open
root@ubuntu:/home/ubuntu# ssh rifqi@192.168.137.2
rifqi@192.168.137.2's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 02:37:56 PM UTC 2023

System load:  0.0          Processes:      233
Usage of /:   30.9% of 9.75GB Users logged in: 1
Memory usage: 10%        IPv4 address for ens33: 192.168.137.2
Swap usage:   0%
```

Gambar 4. 169 Membuka SSH Ketukan Benar

13) Setelah dilakukan uji coba masuk ke server mengakses port 22 (SSH) menggunakan teknik *port knocking* dengan ketukan yang benar selanjutnya dilakukan uji coba menutup server yang terbuka dengan ketukan yang salah maka server tersebut masih dapat diakses

```
root@ubuntu:/home/ubuntu# python3 xtea_ssh.py
Input text1: 4500
Input text2: 6028
Input text3: 3667
Encrypted text1: 9e6e7c7573545f9a
Encrypted text2: 98a54bd30ec13e39
Encrypted text3: a8ba623e61e10c12
hitting tcp 192.168.137.2:4500
hitting tcp 192.168.137.2:6028
hitting tcp 192.168.137.2:3667
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:98
hitting tcp 192.168.137.2:0
knock successful.Port open
root@ubuntu:/home/ubuntu# ssh rifqi@192.168.137.2
rifqi@192.168.137.2's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 02:39:15 PM UTC 2023

System load:  0.0          Processes:      236
Usage of /:   30.9% of 9.75GB Users logged in: 1
Memory usage: 10%        IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

 * Introducing Expanded Security Maintenance for Applications.
 * Receive updates to over 25,000 software packages with your
 * Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro
```

Gambar 4. 170 Menutup SSH Ketukan Salah

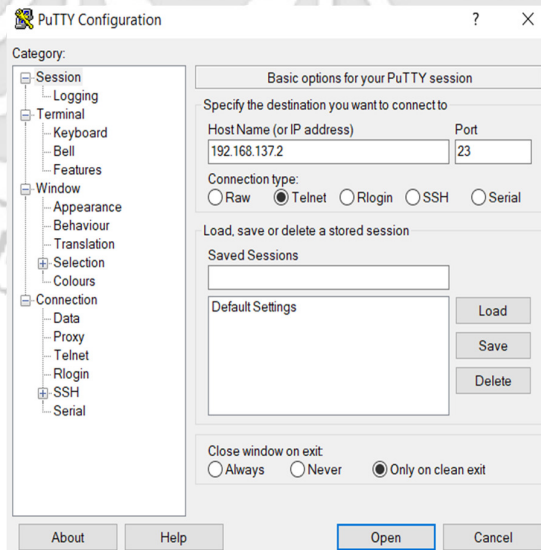
- 14) Setelah dilakukan uji coba menutup *server* dengan ketukan yang salah, selanjutnya menutup *server* agar tidak dapat diakses oleh orang yang tidak berhak maka digunakan teknik *port knocking* dengan ketukan yang benar.

```
root@ubuntu: /home/ubuntu# python3 xtea_ssh.py
Input text1: 4500
Input text2: 6029
Input text3: 3647
Encrypted text1: 9e6e7c7573545f9a
Encrypted text2: 98a54bd38ec13e38
Encrypted text3: 34ece1fab71dc98
hitting tcp 192.168.137.2:4500
hitting tcp 192.168.137.2:6029
hitting tcp 192.168.137.2:3647
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:98
hitting tcp 192.168.137.2:34
knock successful.Port open
root@ubuntu: /home/ubuntu# ssh rifqi@192.168.137.2
xxxxxxxxx
```

Gambar 4. 171 Menutup SSH Ketukan Benar

- b. Langkah uji coba untuk membuka dan menutup *port* 23 (TELNET)

- 1) Menjalankan aplikasi putty dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 172 Konfigurasi Putty

- 2) Tampilan ketika proses *login*, memasukkan *username* dan *password*.

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

Gambar 4. 173 Proses *Login* Putty

- 3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2 tanpa menggunakan *firewall* dan *port knocking*.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875      Processes:           254
Usage of /:   30.9% of 9.75GB   Users logged in:    1
Memory usage: 10%             IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 174 Proses *Login* Telah Berhasil Dilakukan

- 4) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

Gambar 4. 175 Masuk ke Super *User*

- 5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses *ping* ke *user* dengan alamat IP 192.168.137.3 dan proses *ping* berhasil.

```
root@rifqi: /home/rifqi
root@rifqi:/home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data.
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

Gambar 4. 176 Proses PING

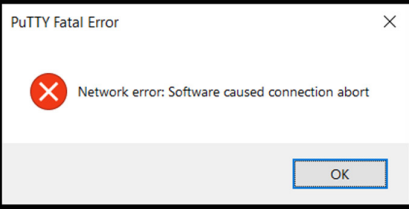
- 6) Setelah *server* berhasil terhubung dengan melakukan proses *ping* ke *user*, maka selanjutnya *server* di tutup total dengan cara mendrop semua akses sehingga tidak ada akses tcp yang dapat lewat. Menggunakan perintah iptables (*firewall*).

```
iptables -I INPUT -p tcp -j DROP
```

Gambar 4. 177 Mendrop Akses Server

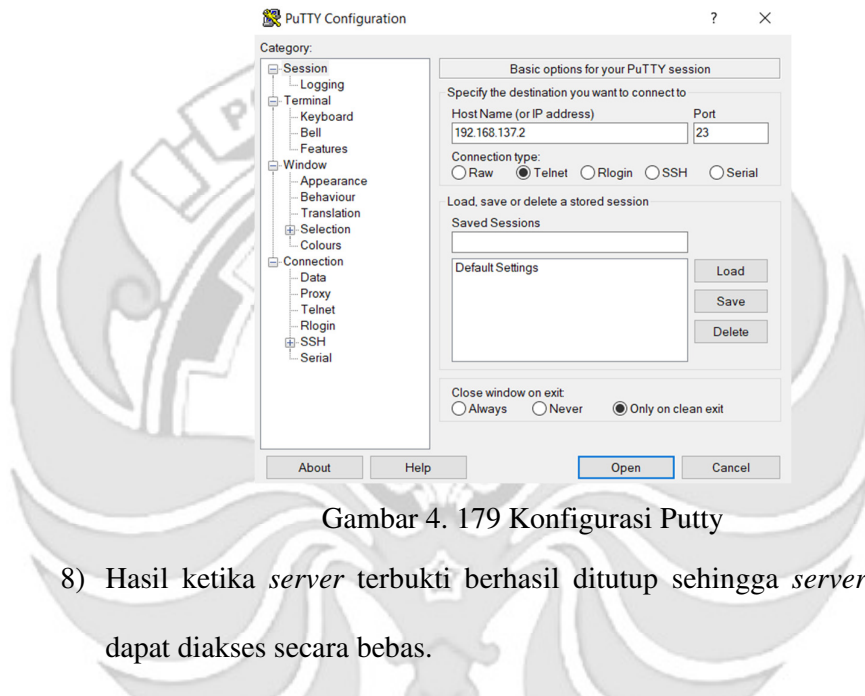
Setelah mendrop semua akses pada *server* dengan perintah iptables maka *server* tidak dapat diakses lagi.

```
root@rifqi: /home/rifqi
root@rifqi:/home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data.
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=0.506 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.498 ms
^C
--- 192.168.137.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1021ms
rtt min/avg/max/mdev = 0.498/0.502/0.506/0.004 ms
root@rifqi:/home/rifqi#
```



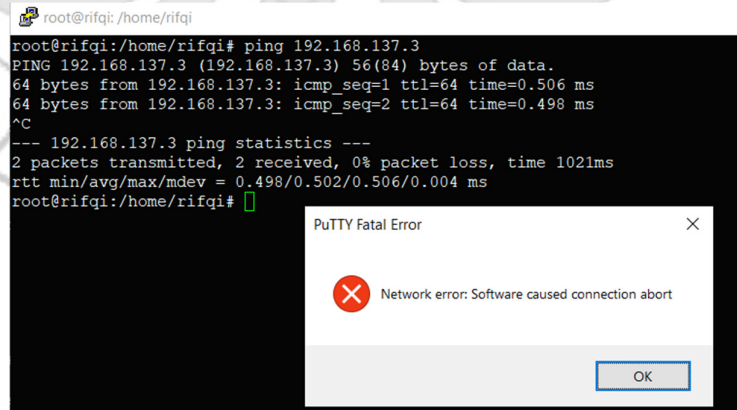
Gambar 4. 178 Akses Server di Drop

- 7) Setelah *server* berhasil ditutup dengan perintah iptables maka akan dilakukan uji coba kembali masuk ke *server* yang dituju yaitu IP 192.168.137.2 dengan menggunakan aplikasi putty dan membuktikan bahwa *server* tersebut sudah ditutup sehingga tidak dapat diakses secara bebas.



Gambar 4. 179 Konfigurasi Putty

- 8) Hasil ketika *server* terbukti berhasil ditutup sehingga *server* tidak dapat diakses secara bebas.



Gambar 4. 180 Server Tidak Dapat Diakses

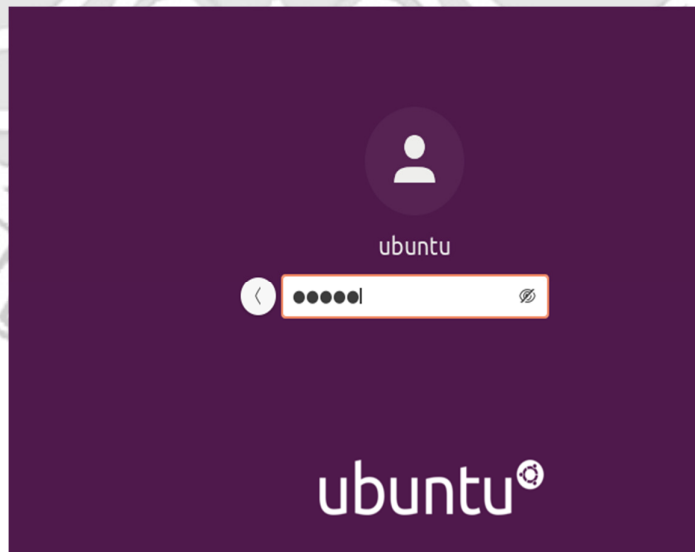
- 9) Setelah akses dari *server* ditutup, hanya terdapat satu metode yang dipakai untuk masuk ke sistem *server* dengan cara menggunakan metode *port knocking*. Untuk dapat mengakses *server* maka harus melalui *admin* yang sudah dilakukan penginstalan packet *knockd* agar dapat menggunakan metode *port knocking*.

```
root@ubuntu:/home/ubuntu# sudo systemctl status knockd
● knockd.service - Port-Knock Daemon
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor prese
   Active: active (running) since Tue 2023-08-15 06:32:15 PDT; 2s ago
     Docs: man:knockd(1)
    Main PID: 2592 (knockd)
      Tasks: 1 (limit: 4572)
     Memory: 656.0K
    CGroup: /system.slice/knockd.service
            └─2592 /usr/sbin/knockd -i ens33

Aug 15 06:32:15 ubuntu systemd[1]: Started Port-Knock Daemon.
Aug 15 06:32:15 ubuntu knockd[2592]: starting up, listening on ens33
lines 1-12/12 (END)
```

Gambar 4. 181 Pengecekan Status *Knockd*

- 10) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan *password*.



Gambar 4. 182 Proses *Login Admin*

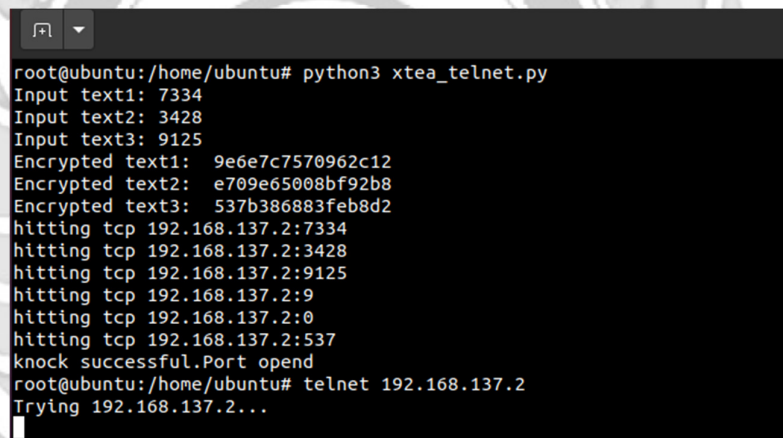
11) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).



```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

Gambar 4. 183 Masuk ke Super *User*

12) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 23 (TELNET) menggunakan ketukan yang salah. Membuktikan bahwa *server* tidak dapat diakses menggunakan *port* 23 (TELNET) secara bebas jika ketukan tidak sesuai dengan konfigurasi yang dilakukan di *server*.



```
root@ubuntu: /home/ubuntu# python3 xtea_telnet.py
Input text1: 7334
Input text2: 3428
Input text3: 9125
Encrypted text1: 9e6e7c7570962c12
Encrypted text2: e709e65008bf92b8
Encrypted text3: 537b386883feb8d2
hitting tcp 192.168.137.2:7334
hitting tcp 192.168.137.2:3428
hitting tcp 192.168.137.2:9125
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:0
hitting tcp 192.168.137.2:537
knock successful.Port open
root@ubuntu: /home/ubuntu# telnet 192.168.137.2
Trying 192.168.137.2...
```

Gambar 4. 184 Membuka TELNET Ketukan Salah

13) Selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 23 (TELNET) menggunakan ketukan yang benar. Membuktikan bahwa *server* dapat diakses menggunakan *port* 23 (TELNET) jika ketukan yang sesuai dengan konfigurasi yang dilakukan di *server*.

```
root@ubuntu:/home/ubuntu# python3 xtea_telnet.py
Input text1: 7324
Input text2: 3429
Input text3: 9125
Encrypted text1: 9e6e7c7570962d7b
Encrypted text2: 7597af5fd2b6df18
Encrypted text3: 3580d871805cbd21
hitting tcp 192.168.137.2:7324
hitting tcp 192.168.137.2:3429
hitting tcp 192.168.137.2:9125
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:7597
hitting tcp 192.168.137.2:3580
knock successful.Port open
root@ubuntu:/home/ubuntu# telnet 192.168.137.2
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^]'.
Ubuntu 22.04.2 LTS
rifqi login: rifqi
Password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 02:49:03 PM UTC 2023

System load:  0.1640625      Processes:           234
Usage of /:   30.9% of 9.75GB Users logged in:    1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:  0%

 * Introducing Expanded Security Maintenance for Applications.
 * Receive updates to over 25,000 software packages with your
 * Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.
```

Gambar 4. 185 Membuka TELNET Ketukan Benar

- 14) Setelah dilakukan uji coba masuk ke *server* mengakses *port* 23 (TELNET) menggunakan teknik *port knocking* dengan ketukan yang benar selanjutnya dilakukan uji coba menutup *server* yang terbuka dengan ketukan yang salah maka *server* tersebut masih dapat diakses

```
root@ubuntu:/home/ubuntu# python3 xtea_telnet.py
Input text1: 9126
Input text2: 3429
Input text3: 7324
Encrypted text1: 9e6e7c757e78fa3f
Encrypted text2: f9287784c20b8920
Encrypted text3: 7871097aca45f668
hitting tcp 192.168.137.2:9126
hitting tcp 192.168.137.2:3429
hitting tcp 192.168.137.2:7324
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:0
hitting tcp 192.168.137.2:6777
knock successful.Port open
root@ubuntu:/home/ubuntu# telnet 192.168.137.2
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^]'.
Ubuntu 22.04.2 LTS
rifqi login: rifqi
Password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 02:51:50 PM UTC 2023

System load:  0.0068359375   Processes:            234
Usage of /:   30.9% of 9.75GB   Users logged in:     1
Memory usage: 10%           IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

 * Introducing Expanded Security Maintenance for Applications.
 * Receive updates to over 25,000 software packages with your
 * Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.
```

Gambar 4. 186 Menutup TELNET Ketukan Salah

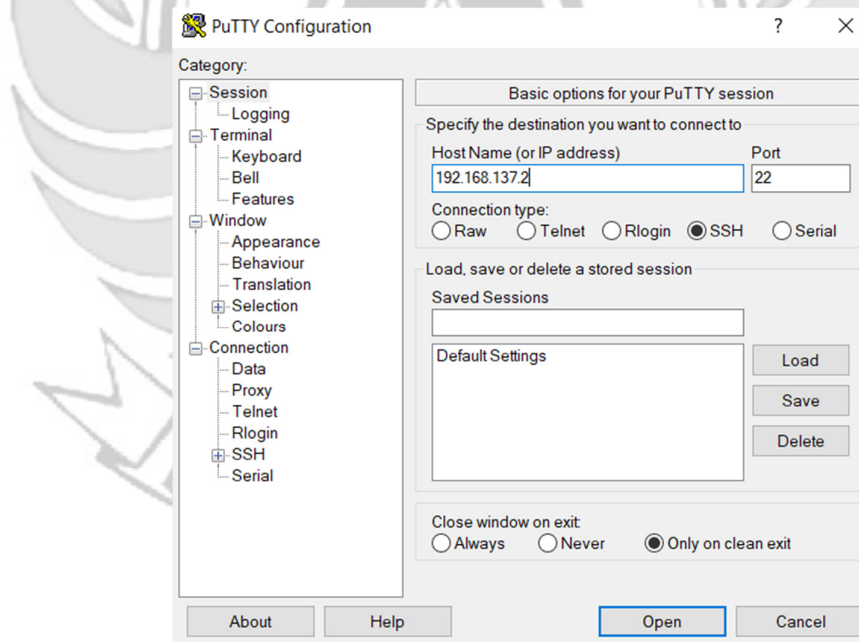
15) Setelah dilakukan uji coba menutup *server* dengan ketukan yang salah, selanjutnya menutup *server* agar tidak dapat diakses oleh orang yang tidak berhak maka digunakan teknik *port knocking* dengan ketukan yang benar.

```
root@ubuntu:/home/ubuntu# python3 xtea_telnet.py
Input text1: 9125
Input text2: 3429
Input text3: 7324
Encrypted text1: 9e6e7c757e78fa3c
Encrypted text2: 14287e755be29834
Encrypted text3: c6ed5971bc4f1fd7
hitting tcp 192.168.137.2:9125
hitting tcp 192.168.137.2:3429
hitting tcp 192.168.137.2:7324
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:14287
hitting tcp 192.168.137.2:0
knock successful.Port opened
root@ubuntu:/home/ubuntu# telnet 192.168.137.2
Trying 192.168.137.2...
xxxxxxx
```

Gambar 4. 187 Menutup TELNET Ketukan Benar

c. Langkah uji coba untuk membuka dan menutup *port* 80 (HTTP).

- 1) Menjalankan aplikasi putty dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 188 Konfigurasi Putty

- 2) Tampilan ketika proses *login*, memasukkan *username* dan *password*.

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

Gambar 4. 189 Proses *Login* Putty

- 3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2 tanpa menggunakan *firewall* dan *port knocking*.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875   Processes:            254
Usage of /:   30.9% of 9.75GB Users logged in:       1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 190 Proses *Login* Telah Berhasil Dilakukan

- 4) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

Gambar 4. 191 Masuk ke Super *User*

- 5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses *ping* ke *user* dengan alamat IP 192.168.137.3 dan proses *ping* berhasil.

```
root@rifqi: /home/rifqi
root@rifqi:/home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data.
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

Gambar 4. 192 Proses PING

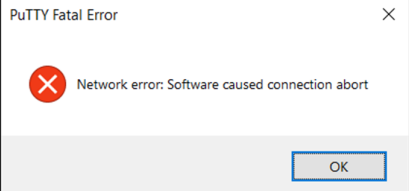
- 6) Setelah *server* berhasil terhubung dengan *melakukan* proses *ping* ke *user*, maka selanjutnya *server* di tutup total dengan cara mendrop semua akses sehingga tidak ada akses tcp yang dapat lewat. Menggunakan perintah iptables (*firewall*).

```
iptables -I INPUT -p tcp -j DROP
```

Gambar 4. 193 Mendrop Akses Server

Setelah mendrop semua akses pada *server* dengan perintah iptables maka *server* tidak dapat diakses lagi.

```
root@rifqi: /home/rifqi
root@rifqi:/home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data.
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=0.506 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.498 ms
^C
--- 192.168.137.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1021ms
rtt min/avg/max/mdev = 0.498/0.502/0.506/0.004 ms
root@rifqi:/home/rifqi#
```



Gambar 4. 194 Akses Server di Drop

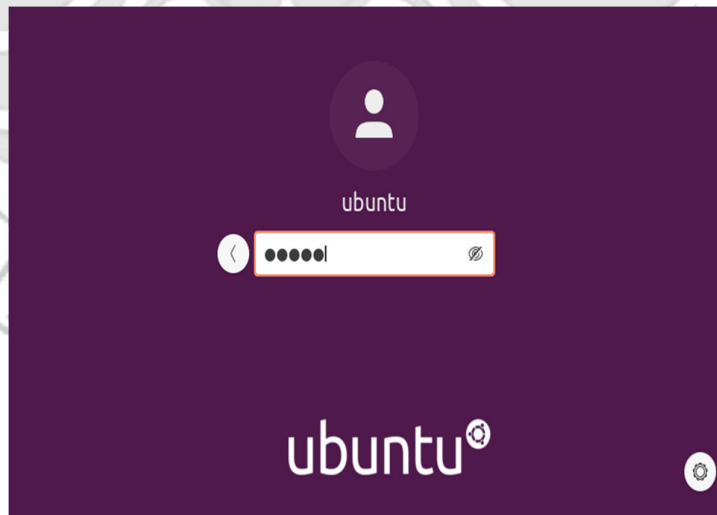
- 7) Setelah akses dari *server* ditutup, hanya terdapat satu metode yang dipakai untuk masuk ke sistem *server* dengan cara menggunakan metode *port knocking*. Untuk dapat mengakses *server* maka harus melalui *admin* yang sudah dilakukan penginstalan packet *knockd* agar dapat menggunakan metode *port knocking*.

```
root@ubuntu:/home/ubuntu# sudo systemctl status knockd
● knockd.service - Port-Knock Daemon
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor prese
   Active: active (running) since Tue 2023-08-15 06:32:15 PDT; 2s ago
     Docs: man:knockd(1)
    Main PID: 2592 (knockd)
      Tasks: 1 (limit: 4572)
     Memory: 656.0K
    CGroup: /system.slice/knockd.service
           └─2592 /usr/sbin/knockd -i ens33

Aug 15 06:32:15 ubuntu systemd[1]: Started Port-Knock Daemon.
Aug 15 06:32:15 ubuntu knockd[2592]: starting up, listening on ens33
lines 1-12/12 (END)
```

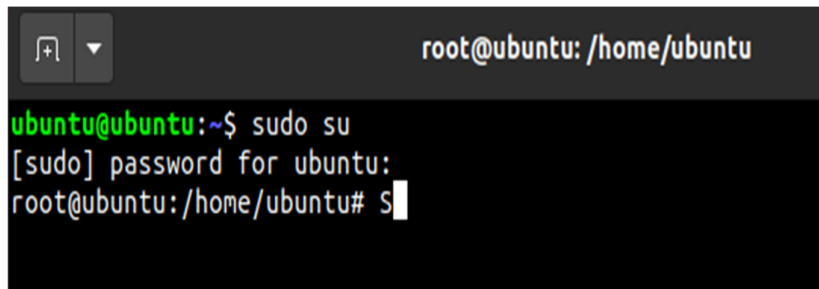
Gambar 4. 195 Pengecekan Status *Knockd*

- 8) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan *password*.



Gambar 4. 196 Proses *Login Admin*

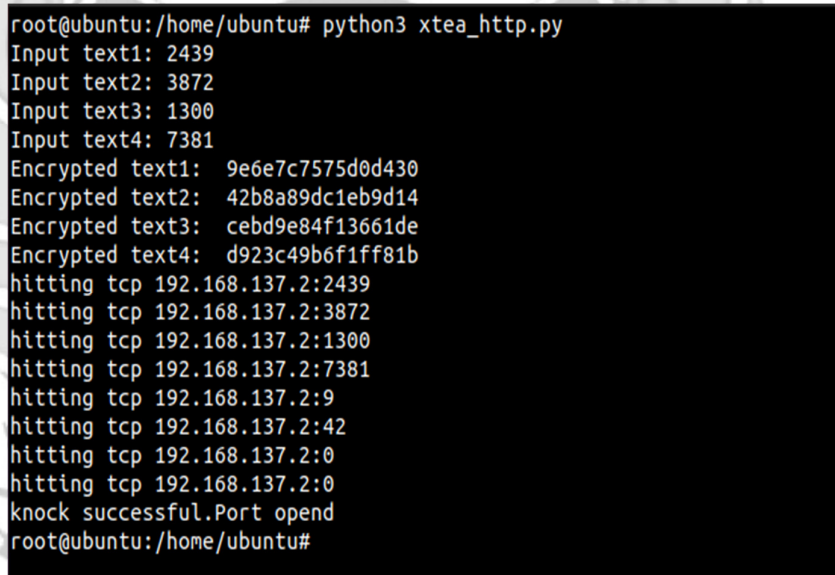
9) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).



```
root@ubuntu: /home/ubuntu  
ubuntu@ubuntu:~$ sudo su  
[sudo] password for ubuntu:  
root@ubuntu: /home/ubuntu# S
```

Gambar 4. 197 Masuk Super User

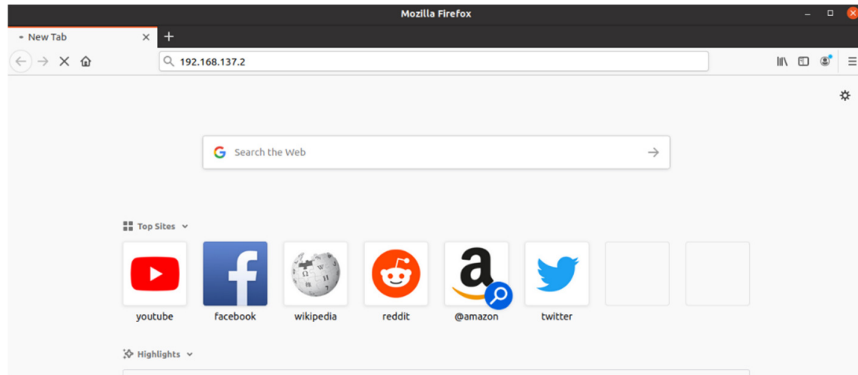
10) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 80 (HTTP) menggunakan ketukan yang salah.



```
root@ubuntu: /home/ubuntu# python3 xtea_http.py  
Input text1: 2439  
Input text2: 3872  
Input text3: 1300  
Input text4: 7381  
Encrypted text1: 9e6e7c7575d0d430  
Encrypted text2: 42b8a89dc1eb9d14  
Encrypted text3: cebd9e84f13661de  
Encrypted text4: d923c49b6f1ff81b  
hitting tcp 192.168.137.2:2439  
hitting tcp 192.168.137.2:3872  
hitting tcp 192.168.137.2:1300  
hitting tcp 192.168.137.2:7381  
hitting tcp 192.168.137.2:9  
hitting tcp 192.168.137.2:42  
hitting tcp 192.168.137.2:0  
hitting tcp 192.168.137.2:0  
knock successful.Port open  
root@ubuntu: /home/ubuntu#
```

Gambar 4. 198 Proses Membuka HTTP Ketukan Salah

11) Maka ketika ketukan yang dimasukkan tidak sesuai untuk mengakses *port* 80 (HTTP) membuktikan bahwa hasil web tidak dapat diakses



Gambar 4. 199 HTTP Tidak Dapat Diakses

12) Selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 80 (HTTP) menggunakan ketukan yang benar.

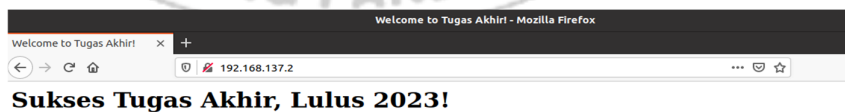
```

root@ubuntu:/home/ubuntu# python3 xtea_http.py
Input text1: 2489
Input text2: 3872
Input text3: 1200
Input text4: 7381
Encrypted text1: 9e6e7c7575d0df4d
Encrypted text2: e20cd5ff36af6344
Encrypted text3: 6d02cfafee4d94e9
Encrypted text4: 52952fbe73487ee4
hitting tcp 192.168.137.2:2489
hitting tcp 192.168.137.2:3872
hitting tcp 192.168.137.2:1200
hitting tcp 192.168.137.2:7381
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:0
hitting tcp 192.168.137.2:6
hitting tcp 192.168.137.2:52952
knock successful.Port open
root@ubuntu:/home/ubuntu#

```

Gambar 4. 200 Membuka HTTP Ketukan Benar

13) Maka ketika ketukan yang dimasukkan sudah sesuai untuk mengakses *port* 80 (HTTP) membuktikan bahwa hasil web dapat diakses



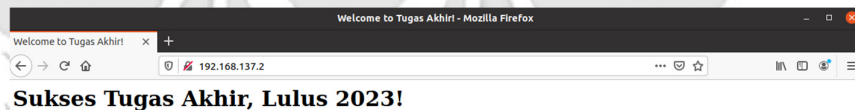
Gambar 4. 201 HTTP Dapat Diakses

- 14) Setelah dilakukan uji coba masuk ke *server* mengakses *port* 80 (HTTP) menggunakan teknik *port knocking* dengan ketukan yang benar selanjutnya dilakukan uji coba menutup *port* 80 (HTTP) yang terbuka dengan ketukan yang salah.

```
root@ubuntu:/home/ubuntu# python3 xtea_http.py
Input text1: 7391
Input text2: 1300
Input text3: 3872
Input text4: 2481
Encrypted text1: 9e6e7c75709626fa
Encrypted text2: 3589b37837576842
Encrypted text3: 5ccc5d6692e2ffde
Encrypted text4: 1eab8841262029d5
hitting tcp 192.168.137.2:7391
hitting tcp 192.168.137.2:1300
hitting tcp 192.168.137.2:3872
hitting tcp 192.168.137.2:2481
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:3589
hitting tcp 192.168.137.2:5
hitting tcp 192.168.137.2:1
knock successful.Port opened
root@ubuntu:/home/ubuntu#
```

Gambar 4. 202 Menutup HTTP Ketukan Salah

- 15) Maka ketika ketukan yang dimasukkan untuk menutup *port* 80 (HTTP) tidak sesuai, maka hasil web dari *port* 80 (HTTP) masih dapat diakses.



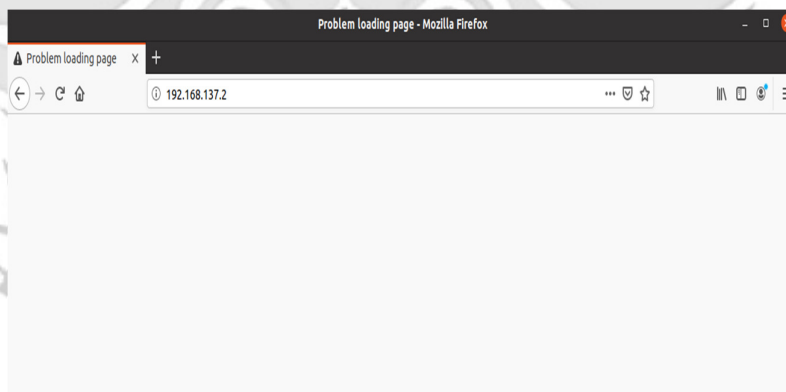
Gambar 4. 203 HTTP Dapat Diakses

- 16) Setelah *port* 80 (HTTP) berhasil dibuka dan diakses oleh *admin* maka untuk mencegah agar *port* tidak dapat diakses oleh orang yang tidak berhak, maka *admin* menutup akses *port* 80 (HTTP) dengan ketukan yang benar menggunakan teknik *port knocking*.

```
root@ubuntu:/home/ubuntu# python3 xtea_http.py
Input text1: 7381
Input text2: 1200
Input text3: 3872
Input text4: 2489
Encrypted text1: 9e6e7c757096270b
Encrypted text2: 7ccb2233942f6c6c
Encrypted text3: 5d3534c03d6cb489
Encrypted text4: 8b46e0abb29f8ed8
hitting tcp 192.168.137.2:7381
hitting tcp 192.168.137.2:1200
hitting tcp 192.168.137.2:3872
hitting tcp 192.168.137.2:2489
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:7
hitting tcp 192.168.137.2:5
hitting tcp 192.168.137.2:8
knock successful.Port open
root@ubuntu:/home/ubuntu#
```

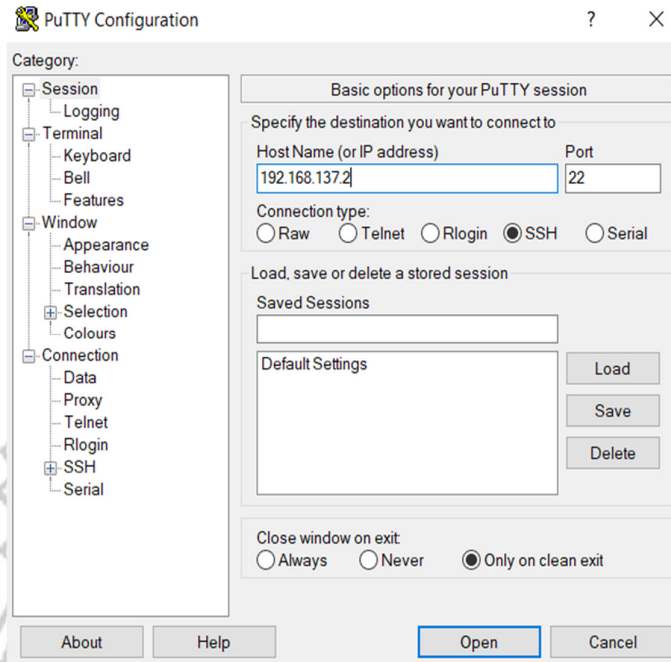
Gambar 4. 204 Menutup HTTP Ketukan Benar

17) Maka ketika ketukan yang dimasukkan untuk menutup *port* 80 (HTTP) telah sesuai, maka hasil web dari *port* 80 (HTTP) tidak dapat diakses.



Gambar 4. 205 HTTP Tidak Dapat Diakses

- d. Langkah uji coba untuk membuka dan menutup *port* 21 (FTP).
 - 1) Menjalankan aplikasi putty dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 206 Konfigurasi Putty

2) Tampilan ketika proses *login*, memasukkan *username* dan *password*.



Gambar 4. 207 Proses Login Putty

3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2 tanpa menggunakan *firewall* dan *port knocking*.


```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875   Processes:    254
Usage of /:   30.9% of 9.75GB Users logged in: 1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:  0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 208 Proses *Login* Telah Berhasil Dilakukan

- 4) Tampilan ketika masuk ke level *user* yang lebih tinggi (super *user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

Gambar 4. 209 Masuk ke Super *User*

- 5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses *ping* ke *user* dengan alamat IP 192.168.137.3 dan proses *ping* berhasil.

```
root@rifqi: /home/rifqi
root@rifqi: /home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data.
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

Gambar 4. 210 Proses PING

- 6) Setelah *server* berhasil terhubung dengan *melakukan* proses *ping* ke *user*, maka selanjutnya *server* di tutup total dengan cara mendrop semua akses sehingga tidak ada akses tcp yang dapat lewat. Menggunakan perintah iptables (*firewall*).

```
iptables -I INPUT -p tcp -j DROP
```

Gambar 4. 211 Mendrop Akses *Server*

Setelah mendrop semua akses pada *server* dengan perintah iptables maka *server* tidak dapat diakses lagi.

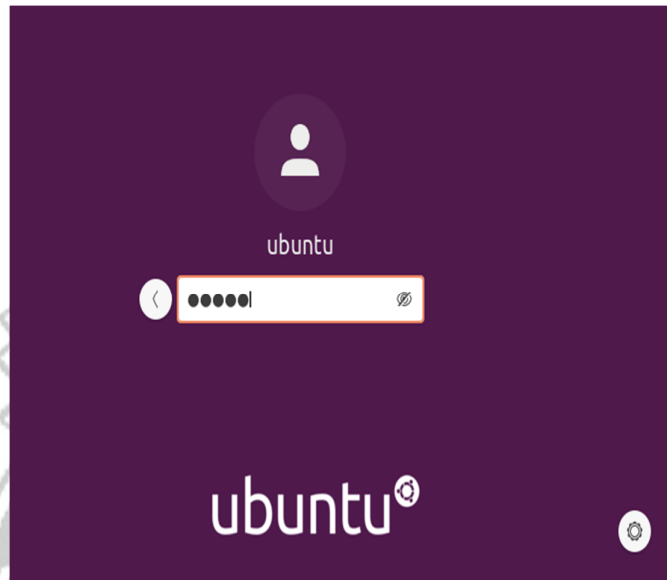
- 7) Setelah akses dari *server* ditutup, hanya terdapat satu metode yang dipakai untuk masuk ke sistem *server* dengan cara menggunakan metode *port knocking*. Untuk dapat mengakses *server* maka harus melalui *admin* yang sudah dilakukan penginstalan packet *knockd* agar dapat menggunakan metode *port knocking*.

```
root@ubuntu:/home/ubuntu# sudo systemctl status knockd
● knockd.service - Port-Knock Daemon
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor prese
   Active: active (running) since Tue 2023-08-15 06:32:15 PDT; 2s ago
     Docs: man:knockd(1)
   Main PID: 2592 (knockd)
     Tasks: 1 (limit: 4572)
    Memory: 656.0K
    CGroup: /system.slice/knockd.service
           └─2592 /usr/sbin/knockd -i ens33

Aug 15 06:32:15 ubuntu systemd[1]: Started Port-Knock Daemon.
Aug 15 06:32:15 ubuntu knockd[2592]: starting up, listening on ens33
lines 1-12/12 (END)
```

Gambar 4. 212 Pengecekan Status *Knockd*

- 8) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan *password*.



Gambar 4. 213 Proses *Login Admin*

- 9) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).

```
root@rifqi:/home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi:/home/rifqi#
```

Gambar 4. 214 Masuk ke Super *User*

- 10) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 21 (FTP) menggunakan ketukan yang salah. Membuktikan bahwa *port* 21 (FTP) tidak dapat diakses secara bebas jika ketukan yang tidak sesuai dengan konfigurasi yang dilakukan di *server*.

```
root@ubuntu:/home/ubuntu# python3 xtea_ftp.py
Input text1: 38992
Input text2: 4820
Input text3: 5390
Input text4: 1690
Encrypted text1: 9e6e7c66ae308e7a
Encrypted text2: f20ead32771bebe9
Encrypted text3: 858844804f034ebb
Encrypted text4: db1afd0f3648cf03
hitting tcp 192.168.137.2:38992
hitting tcp 192.168.137.2:4820
hitting tcp 192.168.137.2:5390
hitting tcp 192.168.137.2:1690
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:0
hitting tcp 192.168.137.2:61060
hitting tcp 192.168.137.2:0
knock successful.Port open
root@ubuntu:/home/ubuntu# ftp -p 192.168.137.2
ftp: connect: Connection timed out
ftp> █
```

Gambar 4. 215 Proses Membuka FTP Ketukan Salah

- 11) Selanjutnya *admin* melakukan percobaan untuk mengakses *port* 21 (FTP) menggunakan ketukan yang benar. Membuktikan *port* 21 (FTP) dapat diakses jika ketukan sesuai dengan konfigurasi yang dilakukan di *server*.

```
root@ubuntu:/home/ubuntu# python3 xtea_ftp.py
Input text1: 3892
Input text2: 4820
Input text3: 5390
Input text4: 2680
Encrypted text1: 9e6e7c7574a1dae4
Encrypted text2: 7fd609c1679bd7ee
Encrypted text3: ac516755200ac9da
Encrypted text4: 5ddbd52f4c2f45bb
hitting tcp 192.168.137.2:3892
hitting tcp 192.168.137.2:4820
hitting tcp 192.168.137.2:5390
hitting tcp 192.168.137.2:2680
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:7
hitting tcp 192.168.137.2:0
hitting tcp 192.168.137.2:5
knock successful.Port open
root@ubuntu:/home/ubuntu# ftp -p 192.168.137.2
Connected to 192.168.137.2.
220 (vsFTPd 3.0.5)
Name (192.168.137.2:ubuntu): politeknik
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> █
```

Gambar 4. 216 Proses Membuka FTP Ketukan Benar

- 12) Setelah dilakukan uji coba masuk ke *server* mengakses *port* 21 (FTP) menggunakan teknik *port knocking* dengan ketukan yang benar selanjutnya dilakukan uji coba menutup *port* 21 (FTP) yang terbuka dengan ketukan yang salah maka *server* tersebut masih dapat diakses

```

root@ubuntu:/home/ubuntu# python3 xtea_ftp.py
Input text1: 2680
Input text2: 4390
Input text3: 4820
Input text4: 3893
Encrypted text1: 9e6e7c7575d2d847
Encrypted text2: db5ea57030751355
Encrypted text3: 77079dc9f80cc4f7
Encrypted text4: fba0b1dd8d1e3012
hitting tcp 192.168.137.2:2680
hitting tcp 192.168.137.2:4390
hitting tcp 192.168.137.2:4820
hitting tcp 192.168.137.2:3893
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:0
hitting tcp 192.168.137.2:11543
hitting tcp 192.168.137.2:0
knock successful.Port open
root@ubuntu:/home/ubuntu# ftp -p 192.168.137.2
Connected to 192.168.137.2.
220 (vsFTPD 3.0.5)
Name (192.168.137.2:ubuntu): politeknik
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>

```

Gambar 4. 217 Proses Menutup FTP Ketukan Salah

- 13) Setelah dilakukan uji coba menutup *port* 21 (FTP) dengan ketukan yang salah, selanjutnya menutup *port* 21 (FTP) agar tidak dapat diakses oleh orang yang tidak berhak maka digunakan teknik *port knocking* dengan ketukan yang benar.

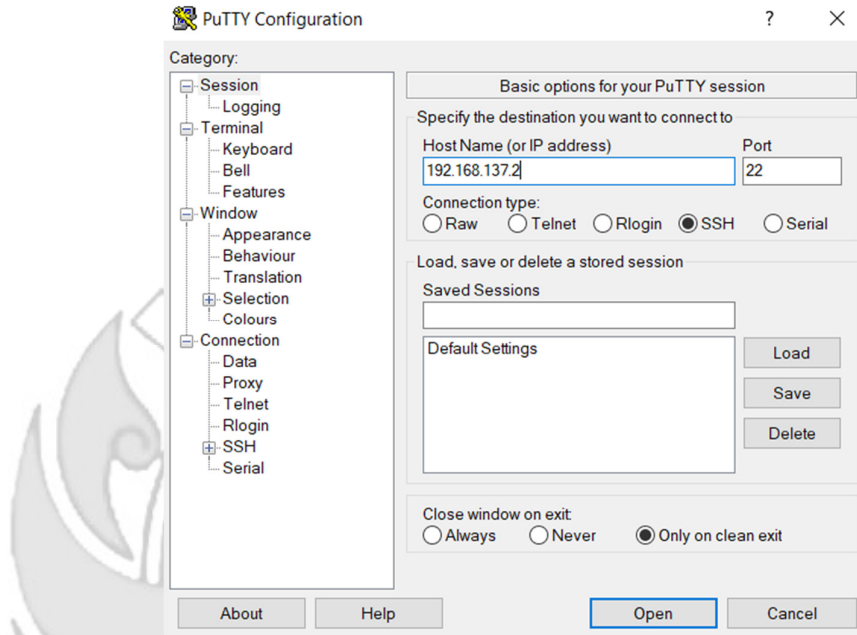
```

root@ubuntu:/home/ubuntu# python3 xtea_ftp.py
Input text1: 2680
Input text2: 5390
Input text3: 4820
Input text4: 3892
Encrypted text1: 9e6e7c7575d2d847
Encrypted text2: db5ea57031d916b9
Encrypted text3: 879085677e281c0b
Encrypted text4: 243b29521d92afe7
hitting tcp 192.168.137.2:2680
hitting tcp 192.168.137.2:5390
hitting tcp 192.168.137.2:4820
hitting tcp 192.168.137.2:3892
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:0
hitting tcp 192.168.137.2:51309
hitting tcp 192.168.137.2:243
knock successful.Port open
root@ubuntu:/home/ubuntu# ftp -p 192.168.137.2
ftp: connect: Connection timed out
ftp>

```

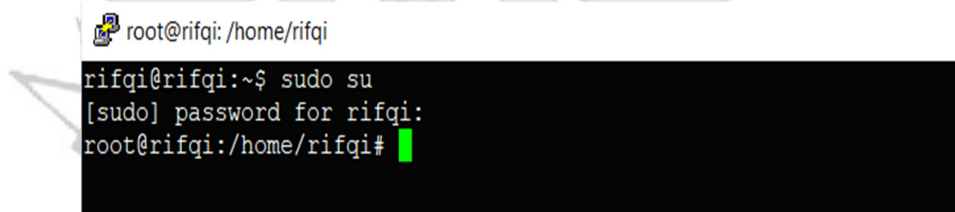
Gambar 4. 218 Proses Menutup FTP Ketukan Benar

- e. Langkah uji coba membuka dan menutup *port 25* (SMTP).
- 1) Menjalankan aplikasi *putty* dan memasukkan alamat IP *server* yang di tuju yaitu 192.168.137.2



Gambar 4. 219 Konfigurasi Putty

- 2) Tampilan ketika proses *login*, memasukkan *username* dan *password*.



Gambar 4. 220 Proses Login

- 3) Tampilan ketika proses *login* sudah dilakukan, maka sudah masuk ke alamat *server* yang dituju yaitu IP 192.168.137.2 tanpa menggunakan *firewall* dan *port knocking*.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug 15 12:33:13 PM UTC 2023

System load:  0.060546875   Processes:            254
Usage of /:   30.9% of 9.75GB Users logged in:     1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 15 12:30:05 2023
rifqi@rifqi:~$
```

Gambar 4. 221 Proses *Login* Telah Berhasil Dilakukan

- 4) Tampilan ketika masuk ke level *user* yang lebih tinggi (super *user*).

```
root@ubuntu: /home/ubuntu
ubuntu@ubuntu:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu: /home/ubuntu# S
```

Gambar 4. 222 Masuk Super *User*

- 5) Ketika sudah masuk ke level *user* yang lebih tinggi, maka dilakukan proses *ping* ke *user* dengan alamat IP 192.168.137.3 dan proses *ping* berhasil.

```
root@rifqi: /home/rifqi
root@rifqi: /home/rifqi# ping 192.168.137.3
PING 192.168.137.3 (192.168.137.3) 56(84) bytes of data.
64 bytes from 192.168.137.3: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 192.168.137.3: icmp_seq=2 ttl=64 time=0.434 ms
64 bytes from 192.168.137.3: icmp_seq=3 ttl=64 time=0.457 ms
64 bytes from 192.168.137.3: icmp_seq=4 ttl=64 time=0.463 ms
```

Gambar 4. 223 Proses PING

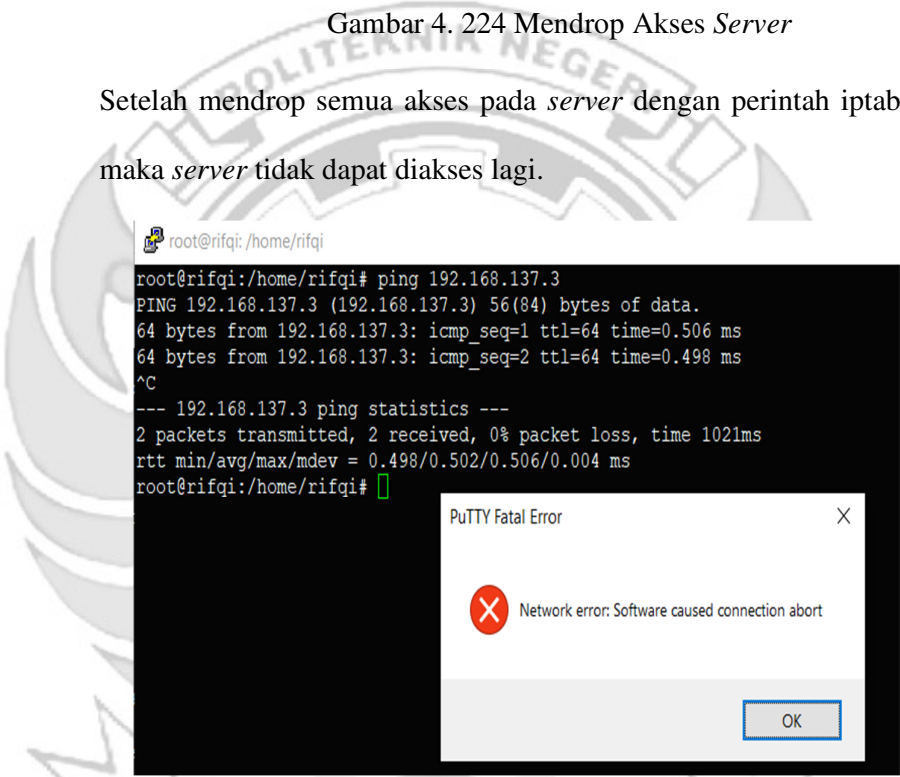
- 6) Setelah *server* berhasil terhubung dengan *melakukan* proses *ping* ke *user*, maka selanjutnya *server* di tutup total dengan cara mendrop

semua akses sehingga tidak ada akses tcp yang dapat lewat.
Menggunakan perintah iptables (*firewall*).

```
iptables -I INPUT -p tcp -j DROP
```

Gambar 4. 224 Mendrop Akses Server

Setelah mendrop semua akses pada *server* dengan perintah iptables maka *server* tidak dapat diakses lagi.



Gambar 4. 225 Server Tidak Dapat Diakses

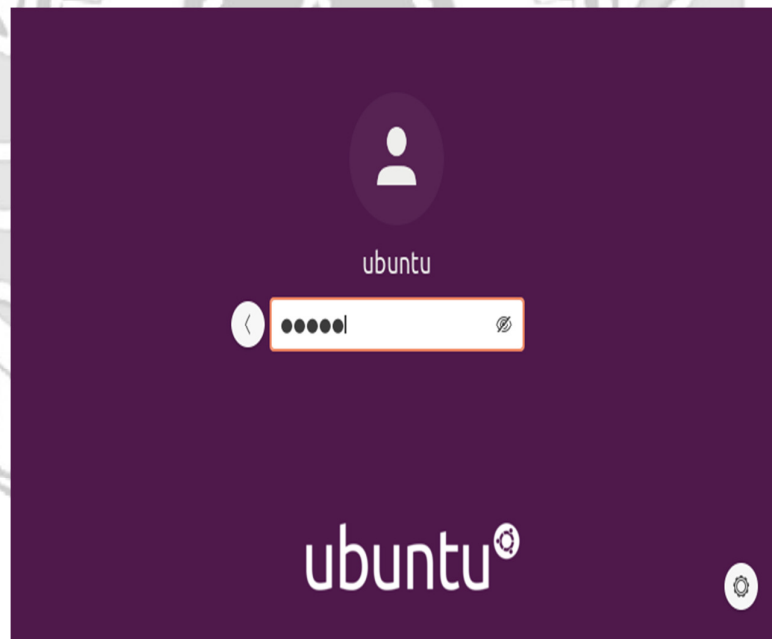
- 7) Setelah akses dari *server* ditutup, hanya terdapat satu metode yang dipakai untuk masuk ke sistem *server* dengan cara menggunakan metode *port knocking*. Untuk dapat mengakses *server* maka harus melalui *admin* yang sudah dilakukan penginstalan packet *knockd* agar dapat menggunakan metode *port knocking*.

```
root@ubuntu:/home/ubuntu# sudo systemctl status knockd
● knockd.service - Port-Knock Daemon
   Loaded: loaded (/lib/systemd/system/knockd.service; disabled; vendor prese>
   Active: active (running) since Tue 2023-08-15 06:32:15 PDT; 2s ago
     Docs: man:knockd(1)
   Main PID: 2592 (knockd)
    Tasks: 1 (limit: 4572)
   Memory: 656.0K
   CGroup: /system.slice/knockd.service
           └─2592 /usr/sbin/knockd -i ens33

Aug 15 06:32:15 ubuntu systemd[1]: Started Port-Knock Daemon.
Aug 15 06:32:15 ubuntu knockd[2592]: starting up, listening on ens33
lines 1-12/12 (END)
```

Gambar 4. 226 Pengecekan Status *Knockd*

- 8) Tampilan ketika proses *login* pada *admin*, memasukkan *username* dan *password*.



Gambar 4. 227 Proses *Login Admin*

- 9) Tampilan ketika masuk ke level *user* yang lebih tinggi (*super user*).

```
root@rifqi: /home/rifqi
rifqi@rifqi:~$ sudo su
[sudo] password for rifqi:
root@rifqi: /home/rifqi#
```

Gambar 4. 228 Masuk ke Super *User*

- 10) Setelah *admin* berhasil *login* selanjutnya *admin* melakukan percobaan untuk masuk ke *server* dengan mengakses *port* 25 (SMTP) menggunakan ketukan yang salah. Membuktikan bahwa *port* 25 (SMTP) tidak dapat diakses secara bebas jika ketukan yang tidak sesuai dengan konfigurasi yang dilakukan di *server*.

```
root@ubuntu: /home/ubuntu# python3 xtea_smtp.py
Input text1: 1500
Input text2: 1600
Input text3: 1700
Input text4: 1800
Input text5: 1900
Encrypted text1: 9e6e7c7576a1b681
Encrypted text2: 580f1bc1008572ca
Encrypted text3: 55130af09c0f49e7
Encrypted text4: 6e87f07c1a64a14d
Encrypted text5: f2c89a287601e54d
hitting tcp 192.168.137.2:1500
hitting tcp 192.168.137.2:1600
hitting tcp 192.168.137.2:1700
hitting tcp 192.168.137.2:1800
hitting tcp 192.168.137.2:1900
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:580
hitting tcp 192.168.137.2:55130
hitting tcp 192.168.137.2:6
hitting tcp 192.168.137.2:0
knock successful.Port opened
root@ubuntu: /home/ubuntu# telnet 192.168.137.2 25
Trying 192.168.137.2...
s
```

Gambar 4. 229 Proses Membuka SMTP Ketukan Salah

- 11) Selanjutnya *admin* melakukan percobaan untuk mengakses *port* 25 (SMTP) menggunakan ketukan yang benar. Membuktikan *port* 25

(SMTP) dapat diakses jika ketukan sesuai dengan konfigurasi yang dilakukan di *server*.

```
root@ubuntu:/home/ubuntu# python3 xtea_smtp.py
Input text1: 1400
Input text2: 1500
Input text3: 1600
Input text4: 1700
Input text5: 1800
Encrypted text1: 9e6e7c7576a0c9e2
Encrypted text2: 709a11a84a89710f
Encrypted text3: 8e1876c2231ce119
Encrypted text4: 25e01cd4a7b1b0e5
Encrypted text5: 96882e214a9d172c
hitting tcp 192.168.137.2:1400
hitting tcp 192.168.137.2:1500
hitting tcp 192.168.137.2:1600
hitting tcp 192.168.137.2:1700
hitting tcp 192.168.137.2:1800
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:709
hitting tcp 192.168.137.2:8
hitting tcp 192.168.137.2:25
hitting tcp 192.168.137.2:31346
knock successful.Port open
root@ubuntu:/home/ubuntu# telnet 192.168.137.2 25
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^]'.
220 rifqi ESMTTP Postfix (Ubuntu)
```

Gambar 4. 230 Proses Membuka SMTP Ketukan Benar

- 12) Setelah dilakukan uji coba masuk ke *server* mengakses *port* 25 (SMTP) menggunakan teknik *port knocking* dengan ketukan yang benar selanjutnya dilakukan uji coba menutup *port* 21 (SMTP) yang terbuka dengan ketukan yang salah maka *server* tersebut masih dapat diakses

```
root@ubuntu:/home/ubuntu# python3 xtea_smtp.py
Input text1: 1900
Input text2: 1800
Input text3: 1700
Input text4: 1600
Input text5: 1500
Encrypted text1: 9e6e7c7576ad1ffd
Encrypted text2: b664d4b36f3570cd
Encrypted text3: 93bbdd0b78d6d74d
Encrypted text4: 483c2d20b3f0667a
Encrypted text5: 3dbef5e76a3a6023
hitting tcp 192.168.137.2:1900
hitting tcp 192.168.137.2:1800
hitting tcp 192.168.137.2:1700
hitting tcp 192.168.137.2:1600
hitting tcp 192.168.137.2:1500
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:0
hitting tcp 192.168.137.2:93
hitting tcp 192.168.137.2:483
hitting tcp 192.168.137.2:3
knock successful.Port open
root@ubuntu:/home/ubuntu# telnet 192.168.137.2 25
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^]'.
220 rifqi ESMTTP Postfix (Ubuntu)
```

Gambar 4. 231 Menutup *Server* Ketukan Salah

- 13) Setelah dilakukan uji coba menutup *port* 25 (SMTP) dengan ketukan yang salah, selanjutnya menutup *port* 25 (SMTP) agar tidak dapat diakses oleh orang yang tidak berhak maka digunakan teknik *port knocking* dengan ketukan yang benar.

```
root@ubuntu:/home/ubuntu# python3 xtea_smtp.py
Input text1: 1800
Input text2: 1700
Input text3: 1600
Input text4: 1500
Input text5: 1400
Encrypted text1: 9e6e7c7576acf854
Encrypted text2: 79b919b4458b0631
Encrypted text3: 1eb6d70441907158
Encrypted text4: 780b21ee963d0ac4
Encrypted text5: 9b7f1ec14af343ab
hitting tcp 192.168.137.2:1800
hitting tcp 192.168.137.2:1700
hitting tcp 192.168.137.2:1600
hitting tcp 192.168.137.2:1500
hitting tcp 192.168.137.2:1400
hitting tcp 192.168.137.2:9
hitting tcp 192.168.137.2:79
hitting tcp 192.168.137.2:1
hitting tcp 192.168.137.2:780
hitting tcp 192.168.137.2:9
knock successful.Port opened
root@ubuntu:/home/ubuntu# telnet 192.168.137.2 25
Trying 192.168.137.2...
xxxxx
```

Gambar 4. 232 Menutup Server Ketukan Benar

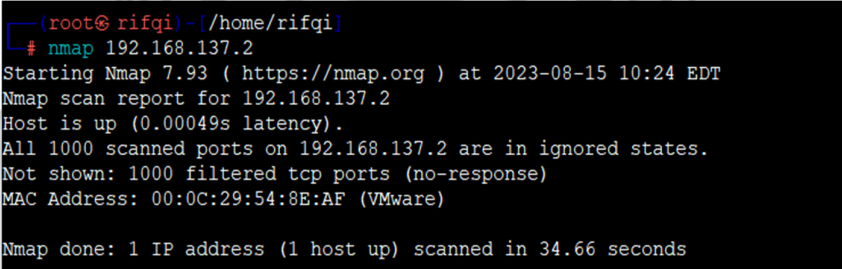
Admin tidak selamanya dapat mengakses *server* secara langsung, karena akan terdapat kondisi *admin* diberikan tugas keluar kota akan tetapi *admin* tetap diharuskan untuk mengakses *server* sehingga *admin* melakukan dengan cara *via remote*. Jika *admin* mengakses *server* secara *via remote* terdapat suatu celah keamanan yang dapat dimanfaatkan oleh *attacker* untuk melakukan penyadapan, maka untuk mengamankan *server* dari penyadapan, *admin* melakukan *remote server* dengan menerapkan metode *port knocking* pada *server* untuk mempersulit *attacker* mendapatkan informasi *port-port* apa saja dalam kondisi terbuka atau *port* apa saja yang sedang di *remote* oleh *admin*

Pada penelitian ini *attacker* melakukan penyadapan ketika *admin* melakukan *remote server* dengan menggunakan serangan *port scanning*. Serangan *port scanning* dilakukan untuk mengetahui informasi yang terdapat pada *server* seperti celah pada *port* tujuan terbuka atau tertutup. Pada tahap pengujian *port scanning*

menggunakan tool NMAP (*Networkk Mapper*). Berikut penjelasan menggunakan serangan *port scanning* untuk mengetahui *port* tujuan terbuka atau tertutup.

a. *Port scanning port 22 (SSH)*

Pada tahap pengujian *port scanning* menggunakan tool *Network Mapper* (*NMAP*) dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port 22* (*SSH*). Pengujian ini dilakukan pada saat *port knocking* sesudah implementasi pada *server*.



```
(root@rifqi)-[~/home/rifqi]
# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 10:24 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00049s latency).
All 1000 scanned ports on 192.168.137.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:0C:29:54:8E:AF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 34.66 seconds
```

Gambar 4. 233 Penyerangan *Port Scanning SSH* Setelah *Port knocking*

Terlihat pada Gambar 4. 240 bahwa setelah *server* menggunakan teknik *port knocking* menjadikan *port 22* (*SSH*) dalam keadaan tertutup sehingga *attacker* tidak dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port 22* (*SSH*).

b. *Port Scanning port 23 (TELNET)*.

Pada tahap pengujian *port scanning* menggunakan tool *Network Mapper* (*NMAP*) dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port 23* (*TELNET*). Pengujian ini dilakukan pada saat *port knocking* sesudah implementasi pada *server*.

```
(root@rifqi)-[/home/rifqi]
# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 10:24 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00049s latency).
All 1000 scanned ports on 192.168.137.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:0C:29:54:8E:AF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 34.66 seconds
```

Gambar 4. 234 Penyerangan *Port Scanning* TELNET Setelah *Port knocking*

Terlihat pada Gambar 4. 241 bahwa setelah *server* menggunakan teknik *port knocking* menjadikan *port* 23 (TELNET) dalam keadaan tertutup sehingga *attacker* tidak dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port* 23 (TELNET).

c. *Port Scanning port* 80 (HTTP)

Pada tahap pengujian *port scanning* menggunakan *tool Network Mapper* (NMAP) dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port* 80 (HTTP). Pengujian ini dilakukan pada saat *port knocking* sesudah implementasi pada *server*.

```
(root@rifqi)-[/home/rifqi]
# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 10:24 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00049s latency).
All 1000 scanned ports on 192.168.137.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:0C:29:54:8E:AF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 34.66 seconds
```

Gambar 4. 235 Penyerangan *Port Scanning* HTTP Setelah *Port knocking*

Terlihat pada Gambar 4. 242 bahwa setelah *server* menggunakan teknik *port knocking* menjadikan *port* 80 (HTTP) dalam keadaan tertutup sehingga *attacker* tidak dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port* 80 (HTTP).

d. *Port Scanning port 21 (FTP)*

Pada tahap pengujian *port scanning* menggunakan *tool Network Mapper (NMAP)* dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port* 21 (FTP). Pengujian ini dilakukan pada saat *port knocking* sesudah implementasi pada *server*.

```
(root@rifqi)~/home/rifqi]
# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 10:24 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00049s latency).
All 1000 scanned ports on 192.168.137.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:0C:29:54:8E:AF (VMware)
Nmap done: 1 IP address (1 host up) scanned in 34.66 seconds
```

Gambar 4. 236 Penyerangan *Port Scanning* FTP Setelah *Port knocking*

Terlihat pada Gambar 4. 243 bahwa setelah *server* menggunakan teknik *port knocking* menjadikan *port* 21 (FTP) dalam keadaan tertutup sehingga *attacker* tidak dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port* 21 (FTP).

e. *Scanning 25 (SMTP)*

Pada tahap pengujian *scanning* menggunakan *tool Network Mapper (NMAP)* dengan men-*scan* IP *server* 192.168.137.2 untuk melihat status *port* 25 (SMTP). Pengujian ini dilakukan pada saat *port knocking* sebelum implementasi pada *server*.

```
(root@rifqi)~/home/rifqi]
# nmap 192.168.137.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-15 10:24 EDT
Nmap scan report for 192.168.137.2
Host is up (0.00049s latency).
All 1000 scanned ports on 192.168.137.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:0C:29:54:8E:AF (VMware)
Nmap done: 1 IP address (1 host up) scanned in 34.66 seconds
```

Gambar 4. 237 Penyerangan *Port Scanning* SMTP Setelah *Port knocking*

Terlihat pada Gambar 4. 244 bahwa setelah *server* menggunakan teknik *port knocking* menjadikan *port 25* (SMTP) dalam keadaan tertutup sehingga *attacker* tidak dapat mengetahui jika *admin* jaringan melakukan *remote server* pada *port 25* (SMTP).

Penerapan dengan metode *port knocking* pada *server* dapat mengatasi masalah sebelumnya karena dengan menggunakan *port knocking port-port* yang sedang *remote* oleh *admin* atau *port-port* yang berada dalam kondisi terbuka tidak dapat diketahui oleh *attacker* walaupun menggunakan serangan *port scanning*, dengan ini maka tidak ada informasi yang didapatkan *attacker* untuk mengakses *server* secara bebas.

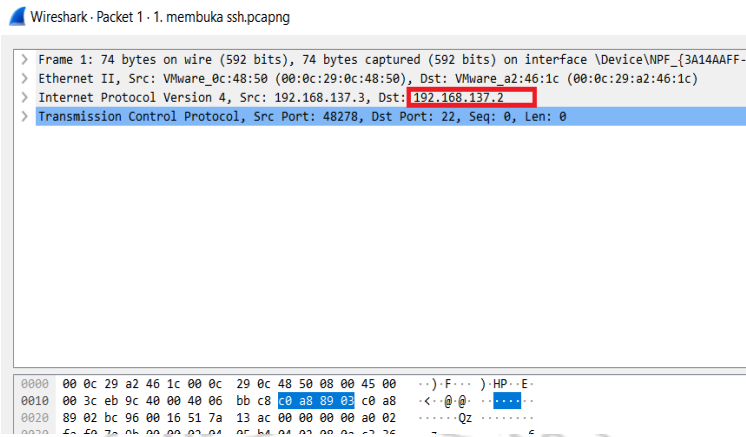
Setelah melakukan serangan dengan menggunakan metode *port scanning* selanjutnya *attacker* melakukan serangan dengan dengan tingkat yang lebih tinggi untuk mendapatkan informasi yang lebih banyak maka *attacker* melakukan serangan dengan metode *sniffing* menggunakan *wireshark*.

a. *Sniffing port 22* (SSH)

Pada tahap pengujian penyerangan *sniffing* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 22* (SSH) untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

9	9.656761	192.168.137.3	192.168.137.2	UDP	60	49547 → 3647	Len=1
10	9.656880	192.168.137.1	192.168.137.3	ICMP	71	Redirect	(
11	9.656908	192.168.137.3	192.168.137.2	UDP	43	49547 → 3647	Len=1
12	9.661944	192.168.137.3	192.168.137.2	TCP	74	45190 → 6020	[SYN] Seq=
13	9.661984	192.168.137.3	192.168.137.2	TCP	74	[TCP R	ransmission] [
14	9.668109	192.168.137.3	192.168.137.2	UDPENCAP	60	[Malformed	Packet]
15	9.668146	192.168.137.3	192.168.137.2	UDPENCAP	43	[Malformed	Packet]
16	9.678550	192.168.137.3	192.168.137.2	UDP	60	56416 → 100	Len=1
17	9.678615	192.168.137.3	192.168.137.2	UDP	43	56416 → 100	Len=1
18	9.685947	192.168.137.3	192.168.137.2	TCP	74	60826 → 0	[SYN] Seq=0
19	9.686014	192.168.137.3	192.168.137.2	TCP	74	[TCP R	ransmission] [
20	9.691954	192.168.137.3	192.168.137.2	UDP	60	37209 → 882	Len=1
21	9.692000	192.168.137.3	192.168.137.2	UDP	43	37209 → 882	Len=1

Gambar 4. 238 *Sequence* Enkripsi SSH



Gambar 4. 239 IP Server

22	14.267293	VMware_c0:00:01	VMware_0c:48:50	ARP	42 Who has 192.168.137.3? Tell 192
23	14.268127	VMware_0c:48:50	VMware_c0:00:01	ARP	60 192.168.137.3 is at 00:0c:29:0c
24	17.767416	192.168.137.3	192.168.137.2	TCP	74 41074 → 22 [SYN] Seq=0 Win=6424

Gambar 4. 240 Port SSH

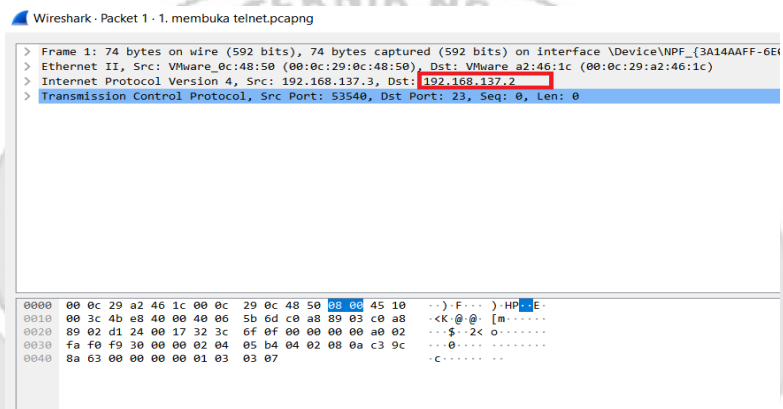
Terlihat pada Gambar 4. 245, 4. 246 dan 4. 247 bahwa ketika seorang *admin* jaringan melakukan *remote server* pada *port 22* (SSH) dengan teknik *port knocking* maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *sequence* dan *port* yang diketuk oleh *admin* jaringan, serta ip dari *server*. Walaupun *attacker* dapat membaca *sequence* yang dilakukan, tetapi ada banyak *sequence* yang muncul sehingga *attacker* harus menentukan *sequence* yang sebenarnya dari sekian banyak *sequence* yang muncul.

b. Sniffing port 23 (TELNET)

Pada tahap pengujian penyerangan *sniffing* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 23* (TELNET) untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

1	0.000000	192.168.137.1	192.168.137.255	BROWSER	243	Host Announcement	DESKTOP-05B
2	18.288272	192.168.137.3	192.168.137.2	UDP	60	38658 → 7324	Len=1
3	18.288445	192.168.137.1	192.168.137.3	ICMP	71	Redirect	(Redirect f
4	18.288486	192.168.137.3	192.168.137.2	UDP	43	38658 → 7324	Len=1
5	18.294867	192.168.137.3	192.168.137.2	TCP	74	56410 → 3429	[SYN] Seq=0 Win=
6	18.294925	192.168.137.3	192.168.137.2	TCP	74	[TCP Retransmission]	[TCP Por
7	18.301305	192.168.137.3	192.168.137.2	UDP	60	33333 → 9125	Len=1
8	18.301373	192.168.137.3	192.168.137.2	UDP	43	33333 → 9125	Len=1
9	18.322316	192.168.137.3	192.168.137.2	UDP	60	55297 → 100	Len=1
10	18.322391	192.168.137.3	192.168.137.2	UDP	43	55297 → 100	Len=1
11	18.328173	192.168.137.3	192.168.137.2	TCP	74	48056 → 0	[SYN] Seq=0 Win=642
12	18.328239	192.168.137.3	192.168.137.2	TCP	74	[TCP Retransmission]	[TCP Por
13	18.345181	192.168.137.3	192.168.137.2	UDP	60	33639 → 5	Len=1
14	18.345347	192.168.137.3	192.168.137.2	UDP	43	33639 → 5	Len=1
15	20.673623	192.168.137.3	192.168.137.2	TCP	74	39536 → 23	[SYN] Seq=0 Win=64

Gambar 4. 241 Sequence Enkripsi TELNET



Gambar 4. 242 IP Server

25	10.310875	192.168.137.3	192.168.137.2	TCP	74	47638 - 23	[SYN] Seq=0 Win=64240
26	10.310955	192.168.137.1	192.168.137.3	ICMP	102	Redirect	(Redirect fi
27	10.310986	192.168.137.3	192.168.137.2	TCP	74	[TCP Retransmission]	[TCP Port n

Gambar 4. 243 Port Server

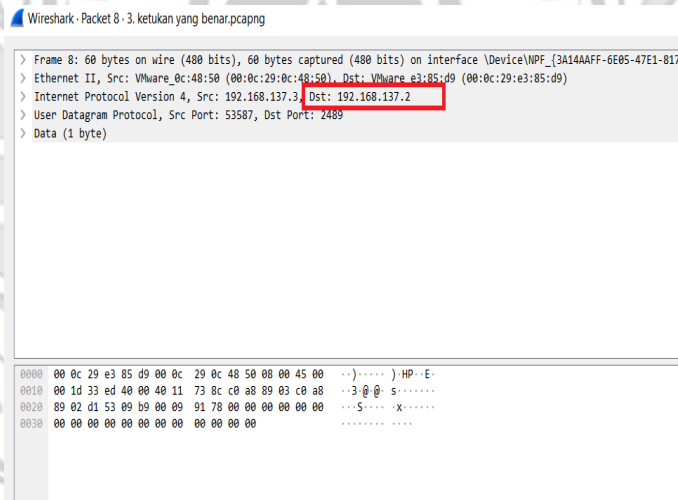
Terlihat pada Gambar 4. 248, 4. 249 dan 4. 250 bahwa ketika seorang *admin* jaringan melakukan *remote server* pada *port* 23 (TELNET) dengan teknik *port knocking* maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *sequence* dan *port* yang diketuk oleh *admin* jaringan, serta ip dari *server*. Walaupun *attacker* dapat membaca *sequence* yang dilakukan, tetapi ada banyak *sequence* yang muncul sehingga *attacker* harus menentukan *sequence* yang sebenarnya dari sekian banyak *sequence* yang muncul.

c. *Sniffing port 80 (HTTP)*

Pada tahap pengujian penyerangan *sniffing* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 80 (HTTP)* untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

49	19.215778	192.168.137.3	192.168.137.2	UDP	60 37011	2489	Len=1
50	19.215867	192.168.137.1	192.168.137.3	ICMP	71	Redirect	(Redirect f
51	19.215900	192.168.137.3	192.168.137.2	UDP	43 37011	2489	Len=1
52	19.222098	192.168.137.3	192.168.137.2	UDP	60 59316	3872	Len=1
53	19.222171	192.168.137.3	192.168.137.2	UDP	43 59316	3872	Len=1
54	19.228017	192.168.137.3	192.168.137.2	UDP	60 47756	1200	Len=1
55	19.228073	192.168.137.3	192.168.137.2	UDP	43 47756	1200	Len=1
56	19.234555	192.168.137.3	192.168.137.2	UDP	60 36556	7381	Len=1
57	19.234609	192.168.137.3	192.168.137.2	UDP	43 36556	7381	Len=1
58	19.248012	192.168.137.3	192.168.137.2	UDP	60 48379	100	en=1
59	19.248074	192.168.137.3	192.168.137.2	UDP	43 48379	100	en=1
60	19.253933	192.168.137.3	192.168.137.2	UDP	60 48874	549	en=1
61	19.253985	192.168.137.3	192.168.137.2	UDP	43 48874	549	en=1
62	19.259816	192.168.137.3	192.168.137.2	UDP	60 53191	3728	Len=1
63	19.259858	192.168.137.3	192.168.137.2	UDP	43 53191	3728	Len=1
64	19.267405	192.168.137.3	192.168.137.2	UDP	60 53837	34	Len=1
65	19.267495	192.168.137.3	192.168.137.2	UDP	43 53837	34	Len=1

Gambar 4. 244 Sequence Enkripsi HTTP



Gambar 4. 245 IP Server

5	0.192324	192.168.137.3	202.67.36.152	TCP	74 58656	80	[SYN] Seq=0
6	0.224387	202.67.36.152	192.168.137.3	TCP	74 80	58656	[SYN, ACK] S
7	0.238976	192.168.137.3	202.67.36.152	TCP	66 58656	80	[ACK] Seq=1

Gambar 4. 246 Port HTTP

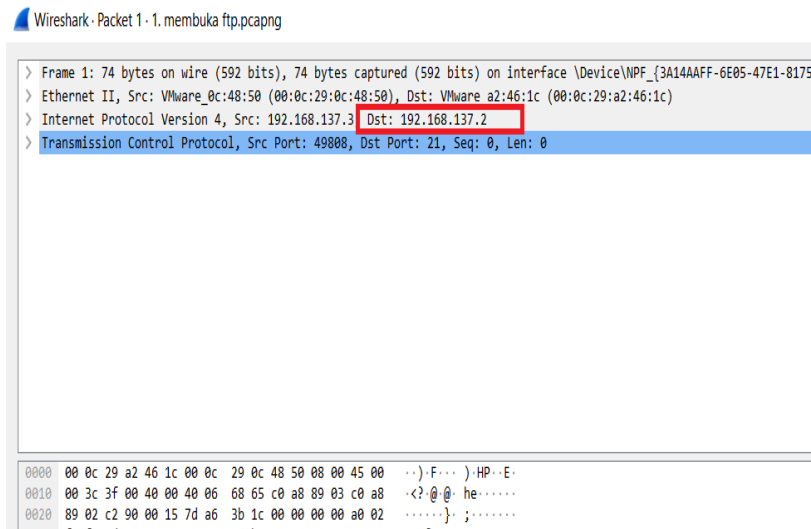
Terlihat pada Gambar 4. 251, 4. 252 dan 4. 253 bahwa ketika seorang *admin* jaringan melakukan *remote server* pada *port* 80 (HTTP) dengan teknik *port knocking* maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *sequence* dan *port* yang diketuk oleh *admin* jaringan, serta ip dari *server*. Walaupun *attacker* dapat membaca *sequence* yang dilakukan, tetapi ada banyak *sequence* yang muncul sehingga *attacker* harus menentukan *sequence* yang sebenarnya dari sekian banyak *sequence* yang muncul.

d. *Sniffing port 21 (FTP)*

Pada tahap pengujian penyerangan *sniffing* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port* 21 (FTP) untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

25	19.093183	192.168.137.3	192.168.137.2	UDP	60 43001 → 3892	Len=1
26	19.093337	192.168.137.1	192.168.137.3	ICMP	71 Redirec	
27	19.093380	192.168.137.3	192.168.137.2	UDP	43 43001 → 3892	Len=1
28	19.099246	192.168.137.3	192.168.137.2	UDP	60 35383 → 4820	Len=1
29	19.099306	192.168.137.3	192.168.137.2	UDP	43 35383 → 4820	Len=1
30	19.106903	192.168.137.3	192.168.137.2	UDP	60 37075 → 5390	Len=1
31	19.106953	192.168.137.3	192.168.137.2	UDP	43 37075 → 5390	Len=1
32	19.117923	192.168.137.3	192.168.137.2	UDP	60 36944 → 2680	Len=1
33	19.118022	192.168.137.3	192.168.137.2	UDP	43 36944 → 2680	Len=1
34	19.131441	192.168.137.3	192.168.137.2	UDP	60 41611 → 100	Len=1
35	19.131502	192.168.137.3	192.168.137.2	UDP	43 41611 → 100	Len=1
36	19.149870	192.168.137.3	192.168.137.2	UDP	60 38796 → 6	Len=1
37	19.149924	192.168.137.3	192.168.137.2	UDP	43 38796 → 6	Len=1

Gambar 4. 247 Enkripsi *Sequence* FTP



Gambar 4. 248 IP Server

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.137.3	192.168.137.2	TCP	74	49808 → 21 [SYN] Seq
2	0.001066	192.168.137.2	192.168.137.3	TCP	74	21 → 49808 [SYN, ACK]

Gambar 4. 249 Port FTP

Terlihat pada Gambar 4. 254, 4. 255 dan 4. 256 bahwa ketika seorang *admin* jaringan melakukan *remote server* pada *port 21* (FTP) dengan teknik *port knocking* maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *sequence* dan *port* yang diketuk oleh *admin* jaringan, serta ip dari *server*. Walaupun *attacker* dapat membaca *sequence* yang dilakukan, tetapi ada banyak *sequence* yang muncul sehingga *attacker* harus menentukan *sequence* yang sebenarnya dari sekian banyak *sequence* yang muncul.

e. *Sniffing port 25 (SMTP)*

Pada tahap pengujian penyerangan *sniffing* menggunakan *tool wireshark* ketika *admin* jaringan melakukan *remote server* pada *port 25 (SMTP)* untuk mendapatkan informasi – informasi yang dapat digunakan untuk mengakses *port* yang sedang di *remote* oleh *admin*.

1	0.000000	192.168.137.3	192.168.137.2	UDP	60 45150 → 1400	Len=1
2	0.000150	192.168.137.1	192.168.137.3	ICMP	71 Redirec	(Redirec
3	0.000177	192.168.137.3	192.168.137.2	UDP	43 45150 → 1400	Len=1
4	0.005720	192.168.137.3	192.168.137.2	UDP	60 38991 → 1500	Len=1
5	0.005759	192.168.137.3	192.168.137.2	UDP	43 38991 → 1500	Len=1
6	0.011717	192.168.137.3	192.168.137.2	UDP	60 60873 → 1600	Len=1
7	0.011763	192.168.137.3	192.168.137.2	UDP	43 60873 → 1600	Len=1
8	0.018094	192.168.137.3	192.168.137.2	UDP	60 41004 → 1700	Len=1
9	0.018148	192.168.137.3	192.168.137.2	UDP	43 41004 → 1700	Len=1
10	0.024007	192.168.137.3	192.168.137.2	UDP	60 34767 → 1800	Len=1
11	0.024076	192.168.137.3	192.168.137.2	UDP	43 34767 → 1800	Len=1
12	0.037767	192.168.137.3	192.168.137.2	UDP	60 42237 → 100	Len=1
13	0.037817	192.168.137.3	192.168.137.2	UDP	43 42237 → 100	Len=1
14	0.052742	192.168.137.3	192.168.137.2	UDP	60 52157 → 66	Len=1
15	0.052817	192.168.137.3	192.168.137.2	UDP	43 52157 → 66	Len=1
16	0.058312	192.168.137.3	192.168.137.2	UDP	60 55401 → 38	Len=1
17	0.058359	192.168.137.3	192.168.137.2	UDP	43 55401 → 38	Len=1
18	0.063996	192.168.137.3	192.168.137.2	UDP	60 33883 → 180	Len=1
19	0.064044	192.168.137.3	192.168.137.2	UDP	43 33883 → 180	Len=1

Gambar 4. 250 Enkripsi *Sequence SMTP*

Wireshark · Packet 1 · 1. membuka port.pcapng

```
> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{3A14AAFF-...}
> Ethernet II, Src: VMware_0c:48:50 (00:0c:29:0c:48:50), Dst: VMware_a2:46:1c (00:0c:29:a2:46:1c)
> Internet Protocol Version 4, Src: 192.168.137.3, Dst: 192.168.137.2
> Transmission Control Protocol, Src Port: 40602, Dst Port: 25, Seq: 0, Len: 0
```

0000 00 0c 29 a2 46 1c 00 0c 29 0c 48 50 08 0c 45 10 ..).F...)HP..E-
0010 00 3c c1 3e 40 00 40 06 e6 16 c0 a8 89 03 c0 a8 <->@ @
0020 89 02 9e 9a 00 19 78 e7 d0 20 00 00 00 a0 02X.....
0030 fa f0 26 c7 00 00 02 04 05 b4 04 02 08 0a c3 ad ..&.....
0040 e7 87 00 00 00 00 01 03 03 07

Gambar 4. 251 IP *Server*

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.137.3	192.168.137.2	TCP	74	40602 → 25 [SYN] Seq=0 Win=6424
2	0.000286	192.168.137.2	192.168.137.3	TCP	74	25 → 40602 [SYN, ACK] Seq=0 Ack

Gambar 4. 252 Port SMTP

Terlihat pada Gambar 4. 257, 4. 258 dan 4. 259 bahwa ketika seorang *admin* jaringan melakukan *remote server* pada *port 25* (SMTP) dengan teknik *port knocking* maka seorang *attacker* dengan menggunakan metode *sniffing* dapat dengan mudah mengetahui *sequence* dan *port* yang diketuk oleh *admin* jaringan, serta ip dari *server*. Walaupun *attacker* dapat membaca *sequence* yang dilakukan, tetapi ada banyak *sequence* yang muncul sehingga *attacker* harus menentukan *sequence* yang sebenarnya dari sekian banyak *sequence* yang muncul.

Memberikan tingkat keamanan *tambahan* pada *port knocking* yaitu dengan menerapkan algoritma *xtea* maka dapat mempersulit *attacker* dalam melakukan proses penyadapan walaupun menggunakan tingkat serangan yang lebih tinggi yaitu serangan *sniffing*, karena informasi *sequence* yang didapatkan *attacker* melalui serangan *sniffing* terdapat banyak *sequence* yang muncul yaitu *sequence* yang asli dan *sequence chipertext* yang telah di enkripsi sehingga *attacker* harus menentukan *sequence* yang sebenarnya dari banyaknya jumlah *sequence* yang terbaca oleh serangan *sniffing* yang dilakukan *attacker*

4.2 Tabel Pengujian

4.2.1 Tabel Pengujian Server

Tabel 4. 1 Pengujian Port SSH (22)

Serangan Sistem Keamanan	Tidak Ada Serangan	<i>Port Scanning Attack</i>	<i>Sniffing Attack</i>
<i>Server Tidak Ada Sistem Keamanan</i>	<i>Port terbuka</i>	<i>Attacker mendapatkan informasi <i>port</i> kondisi terbuka</i>	<i>Attacker mendapatkan informasi <i>port</i> dan ip server plaintext</i>
<i>Server Port Knocking</i>	<i>Port Tertutup</i>	<i>Attacker tidak mendapatkan informasi <i>port</i> kondisi terbuka</i>	<i>Attacker mendapatkan Port, IP server dan Sequence plaintext</i>
<i>Server Port Knocking dan XTEA</i>	<i>Port Tertutup</i>	<i>Attacker tidak mendapatkan informasi <i>port</i> kondisi terbuka</i>	<i>Attacker mendapatkan Port, IP server dan Sequence chipertext</i>

Tabel 4. 2 Pengujian Port Telnet (23)

<div style="text-align: right;">Serangan</div> <div style="text-align: left;">Sistem Keamanan</div>	Tidak Ada Serangan	<i>Port Scanning Attack</i>	<i>Sniffing Attack</i>
<i>Server Tidak Ada Sistem Keamanan</i>	<i>Port terbuka</i>	<i>Attacker mendapatkan informasi port kondisi terbuka</i>	<i>Attacker mendapatkan informasi port dan ip server plaintext</i>
<i>Server Port Knocking</i>	<i>Port Tertutup</i>	<i>Attacker tidak mendapatkan informasi port kondisi terbuka</i>	<i>Attacker mendapatkan Port, IP server dan Sequence plaintext</i>
<i>Server Port Knocking dan XTEA</i>	<i>Port Tertutup</i>	<i>Attacker tidak mendapatkan informasi port kondisi terbuka</i>	<i>Attacker mendapatkan Port, IP server dan Sequence chipertext</i>

Tabel 4. 3 Pengujian *Port* HTTP (80)

<div style="text-align: right;">Serangan</div> <div style="text-align: left;">Sistem Keamanan</div>	Tidak Ada Serangan	<i>Port Scanning Attack</i>	<i>Sniffing Attack</i>
<i>Server Tidak Ada Sistem Keamnan</i>	<i>Port terbuka</i>	<i>Attacker mendapatkan informasi <i>port</i> kondisi terbuka</i>	<i>Attacker mendapatkan informasi <i>port</i> dan ip server plaintext</i>
<i>Server Port Knocking</i>	<i>Port Tertutup</i>	<i>Attacker tidak mendapatkan informasi <i>port</i> kondisi terbuka</i>	<i>Attacker mendapatkan <i>Port</i>, IP server dan sequence plaintext</i>
<i>Server Port Knocking dan XTEA</i>	<i>Port Tertutup</i>	<i>Attacker tidak mendapatkan informasi <i>port</i> kondisi terbuka</i>	<i>Attacker mendapatkan <i>Port</i>, IP server dan sequence chipertext</i>

Tabel 4. 4 Pengujian *Port* FTP (21)

<div style="text-align: right;">Serangan</div> <div style="text-align: left;">Sistem Keamanan</div>	Tidak Ada Serangan	<i>Port Scanning Attack</i>	<i>Sniffing Attack</i>
<i>Server Tidak Ada Sistem Keamanan</i>	<i>Port terbuka</i>	<i>Attacker mendapatkan informasi <i>port</i> kondisi terbuka</i>	<i>Attacker mendapatkan informasi <i>port</i> dan ip server plaintext</i>
<i>Server Port Knocking</i>	<i>Port Tertutup</i>	<i>Attacker tidak mendapatkan informasi <i>port</i> kondisi terbuka</i>	<i>Attacker mendapatkan <i>Port</i>, IP server dan sequence plaintext</i>
<i>Server Port Knocking dan XTEA</i>	<i>Port Tertutup</i>	<i>Attacker tidak mendapatkan informasi <i>port</i> kondisi terbuka</i>	<i>Attacker mendapatkan <i>Port</i>, IP server dan sequence chipertext</i>

Tabel 4. 5 Pengujian *Port* SMTP (25)

Serangan Sistem Keamanan	Tidak Ada Serangan	<i>Port Scanning Attack</i>	<i>Sniffing Attack</i>
<i>Server Tidak Ada Sistem Keamanan</i>	<i>Port terbuka</i>	<i>Attacker mendapatkan informasi port kondisi terbuka</i>	<i>Attacker</i> mendapatkan informasi <i>port</i> dan ip server <i>plaintext</i>
<i>Server Port Knocking</i>	<i>Port Tertutup</i>	<i>Attacker tidak mendapatkan informasi port kondisi terbuka</i>	<i>Attacker</i> mendapatkan <i>Port</i> , IP server dan sequence <i>plaintext</i>
<i>Server Port Knocking dan XTEA</i>	<i>Port Tertutup</i>	<i>Attacker tidak mendapatkan informasi port kondisi terbuka</i>	<i>Attacker</i> mendapatkan <i>Port</i> , IP server dan sequence <i>chipertext</i>

Keterangan:

1. Tidak ada serangan, pada *server* tidak ada sistem keamanan maka *port* terbuka sehingga setiap *port-port* yang terdapat pada *server* dapat diakses oleh siapa saja termasuk oleh orang-orang yang tidak memiliki hak akses.
2. Tidak ada serangan, pada *server port knocking* maka *port* tertutup sehingga hanya admin yang dapat membuka *port-port* yang terdapat pada *server*.
3. Tidak ada serangan, pada *server port knocking* dan *xtea* maka *port* tertutup sehingga hanya admin yang dapat membuka *port-port* yang terdapat pada *server*.
4. *Port scanning attack* pada *server* tidak ada sistem keamanan maka *attacker* berhasil mendapatkan informasi *port-port* kondisi terbuka yang terdapat pada *server*.
5. *Port scanning attack* pada *server port knocking* maka *attacker* tidak berhasil mendapatkan informasi *port-port* kondisi terbuka yang terdapat pada *server*.
6. *Port scanning attack* pada *server port knocking* dan *xtea* maka *attacker* tidak berhasil mendapatkan informasi *port-port* kondisi terbuka yang terdapat pada *server*.
7. *Sniffing attack* pada *server* tidak ada sistem keamanan maka *attacker* akan mendapatkan informasi yang sangat baik yaitu *port* yang sedang di *remote* oleh seorang *admin* serta alamat ip dari *server* dalam bentuk *plaintext*, informasi tersebut dapat digunakan untuk mengakses *server*.
8. *Sniffing attack* pada *server port knocking* maka *attacker* akan mendapatkan informasi yang sangat baik yaitu *port* yang sedang di *remote* oleh seorang *admin*, alamat ip dari *server* serta *sequence* yang digunakan untuk membuka

port dalam bentuk *plaintext*, informasi tersebut dapat digunakan untuk mengakses *server*.

9. *Sniffing attack* pada *server port knocking* dan *xtea* maka *attacker* akan mendapatkan informasi yaitu *port* yang sedang di *remote* oleh seorang *admin*, alamat ip dari *server* serta *sequence ciphertext*. Dengan bentuk *sequence ciphertext* yang telah di enkripsi maka akan meningkatkan kerumitan *attacker* dalam menemukan *sequence* yang sebenarnya.



BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan implementasi dan pengujian yang dilakukan untuk penerapan *port knocking* pada *server* ketika *admin* melakukan *remote server* dan penerapan algoritma XTEA pada *port knocking* disimpulkan sebagai berikut:

- a. Penerapan *port knocking* dapat memberikan keamanan pada *server* karena *server* tidak dapat diakses secara bebas setelah akses *server* di drop menggunakan *firewall* (iptables). Untuk dapat melakukan *remote server* maka harus mengetahui *sequence* pada setiap *port* yang terpasang pada *server*. Jika terdapat seorang *attacker* melakukan serangan *port scanning* maka *attacker* tidak dapat mengetahui *port* terbuka yang sedang di *remote* oleh *admin*, tetapi jika *attacker* melakukan serangan *sniffing* menggunakan *wireshark* maka *attacker* mendapatkan informasi – informasi penting untuk dapat mengakses *server* seperti *ip* dari *server*, *port server* dan *sequence port*. Jadi penerapan *port knocking* pada *server* masih memiliki celah keamanan yang dapat dimanfaatkan oleh *attacker* oleh karena itu untuk menutup celah keamanan tersebut maka pada penitian ini diusulkan penggunaan algoritma XTEA pada *port knocking*.
- b. Untuk menutup celah keamanan yang telah dijelaskn pada *point* diatas maka penerapan algoritma XTEA pada *port knocking* menambah tingkat keamanan pada *server* terutama jika *attacker* melakukan serangan *sniffing* yaitu memberikan kerumitan pada *attacker* untuk menemukan urutan *sequence* yang di baca *wireshark* karena telah terdapat *sequence* enkripsi berbasis XTEA dengan seperti itu maka penerapan algoritma XTEA pada *port knocking* dapat

menutupi kelemahan pada *portknoc king* itu sendiri jika terdapat serangan *sniffing* dari *attacker*.

5.2 Saran

Untuk selanjutnya penerapan algoritma XTEA pada *port knocking* dapat dilakukan *enkripsi port* dan *ip server* agar lebih mempersulit *attacker* melakukan penyadapan pada *server*. Dengan ini maka akan memberikan tingkat keamanan yang lebih tinggi.



DAFTAR PUSTAKA

- Aamir Bokhari, Yuta Inoue, Seiya Kato, Katsunari and Tsutomu. 2021. "Emperical Analysis Of Security And Power-Saving Features Of *Port Knocking* Technique Applied To An Iot Device." *Journal of Information Processing* 29: 572–80.
- Achmad R., Manullang, E. V., and Sanmas, E. R. (2020). Rancang Bangun Aplikasi Deteksi Dan Penanganan Serangan Ddos Dan *Port Scanning* Memanfaatkan Snort Pada Jaringan Komputer. *Jurnal Teknologi Informasi*, 8(1), 2–11.
- Admin Kominfo. 2020. "Dasar-dasar Jaringan Komputer." *Kominfo*. <https://kominfo.bengkulukota.go.id/dasar-dasar-jaringan-komputer/> (August 3, 2022).
- Albar, R., & Putra, R. O. (2022). *Sniffing* Dan Implementasi Keamanan Jaringan *Networkk Security Analysis Using the Method Sniffing* and Implementation of *Networkk Security* on Microtik Router Os V6 . 48 . 3 Using *Port Knocking* Method. *Journal of Informatics and Komputer Science*, 8(1), 1–11.
- Ali, Muhammad Rasyid, Muhamad Anda Falahuddin, Susilawati St, and M Eng. 2021. "Pembuatan *Remote* Accessabble Plc Logo Siemens Dengan Web *Server* Programming Pada Training Unit Sistem Refrigerasi." *Prosiding The 12th Industrial Research Workshop and National Seminar*: 4–5.
- Andreatos, Antonios S. 2017. "Hiding The Ssh *Port* Via Smart *Port knocking*." 11:28–31 <https://www.researchgate.net/publication/315896859>.
- Anif, M., Siswanto, and Fachri and Gunawan Prasetyo, Basuki Hari. 2020. "Aplikasi Pengamanan Data Email Menggunakan Algoritma Kriptografi Xtea Berbasis Web." *Jurnal Bit* 17(2): 46–52.
- Anusha, R., and V. Veena Devi Shastrimath. 2021. "Rfid-Ma Xtea: Cost-Effective Rfid-Mutual Authentication Design Using Xtea Security On Fpga Platform." *International Journal of Electronics and Telecommunications* 67(4): 623–29.
- Azhari, M., Mulyana, D. I., Perwitosari, F. J., & Ali, F. (2022). "Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES)." *Jurnal Pendidikan Sains dan Komputer*. 2(1), 163–171.
- Basten, Marco V a N. 2009. "Jaringan Komputer Optimalisasi Firewall Pada Jaringan Skala Luas."
- Brades, T., & Irwansyah. (2022). " Pemanfaatan Metode *Port Knocking* Dan Blocking." *Seminar Hasil Penelitian Vokasi (SEMHAVOK)*, 3(No.2), 1–9.

- Cahyani, Ika Dwi. 2011. "Sistem Keamanan Enkripsi Secure Shell (ssh) Untuk Keamanan Data." *Jurnal Teknik Elektronika Fak Teknik Universitas Pandanaran*: 1–8.
- Devie Ryana Suchendra¹, Alfian Fitra Rahman², Setia Juli Irzal Ismail³. 2017. "Penerapan Sistem Pengamanan *Port* Pada Layanan Jaringan Menggunakan *Port Knocking*." *Jurnal Lpkia* 10(2): 45–50.
- Ernawati, Rosalia, Ikhwan Ruslianto, and Syamsul Bahri. 2022. "Implementasi Metode *Port Knocking* Pada Sistem Keamanan *Server* Ubuntu Virtual Berbasis Web Monitoring." *Coding : Jurnal Komputer dan Aplikasi* 10(01): 158–69. <https://jurnal.untan.ac.id/index.php/jcskommipa/article/view/54226>.
- Fatoni, Windu Farhan and Mustika. 2022. "Dengan Metode *Port Knocking*." *Jurnal Mahasiswa Ilmu Komputer (JMik)*." 03(01).
- Febrianti, Dinda. 2017. "Pengertian, Manfaat dan Macam-Macam Jaringan Komputer." *Info Publik*. <https://infopublik.sijunjung.go.id/pengertian-manfaat-dan-macam-macam-jaringan-komputer-bagian-1/> (August 4, 2022).
- Gatra, Hikmah. 2015. "Implementasi Honeypot Pada Web *Server* Air Traffic Control (Atc) Menggunakan Kfsensor." *e Proceeding of Applied Science* 1(3): 2356. www.ask.wireshark.org.
- Halik, Idham, and Yudi Prayudi. 2005. "Studi dan Analisis Algoritma Rivest Code 6 (RC6) Dalam Enkripsi/Dekripsi Data" *Snati* 6(D). <http://journal.uui.ac.id/index.php/Snati/article/view/1402>.
- Haynes, Duncan H. 2018. "Aplikasi Web *Server* Berbasis Bahasa C Sharp." *Jurnal Teknik Komputer*: 406–10.
- Iqbal, Muhammad, Arini, and Hendra Bayu Suseno. 2020. "Analysis And Simulation Of Ubuntu *Server* Network Security Using *Port Knocking* , *Honeypot* , *Iptables* , *Icmp*." *Cyber Security dan Forensik Digital* 3(1): 27–32.
- Khadafi, Shah, S Nurmuslimah, and Florian Kelvianto Anggakusuma. 2019. "Implementasi Firewall Dan *Port Knocking* Sebagai Keamanan Data Transfer Pada *Ftp Server* Berbasis Linux Ubuntu *Server*." *Nero* 4(3): 181–88. <https://nero.trunojoyo.ac.id/index.php/nero/article/view/137>.
- Kurniawan, Budi, and Dodi Herryanto. 2017. "Perancangan Dan Implementasi Data Center Menggunakan *File Transfer Protocol (Ftp)*." *Jurnal Sistem Komputer Musirawas* 2(2): 91–97.

- Major, Will, William J. Buchanan, and Jawad Ahmad. 2020. "An Authentication Protocol Based On Chaos And Zero Knowledge Proof." *Nonlinear Dynamics* 99(4): 3065–87.
- Massandy, Danang Tri. 2009. "Algoritma Elgamal Dalam Pengamanan Pesan Rahasia." *Institut Teknologi Bandung*: 1–5. www.informatika.stei.itb.ac.id.
- Nursalim. 2013. "Keamanan Jaringan Dengan Teknik *Port Knocking*."
- Pandiangan, H P H. 2020. "Implementasi Algoritma Xtea (Extended Tyni Encryption Algoritma) Dalam Pengamanan Data *File* Dokumen Teks." *bulletin of information technology (bit)* 1(3): 122–33. <https://journal.fkpt.org/index.php/BIT/article/view/44>.
- Perbandingan, Analisis et al. 2019. "Analisis Perbandingan Sistem Autentikasi *Port Knocking* Dan Single Packet Authorization Pada *Server Raspbian*." 2(1): 28–37.
- Popeea, T., Olteanu, V., Gheorghe, L., & Rughiniş, R. (2011). "Extension of a *port knocking client-server* architecture with NTP synchronization. *Proceedings - RoEduNet IEEE International Conference*", 1–5. <https://doi.org/10.1109/RoEduNet.2011.5993704>
- Prismana, S. and I. G. L. P. E. (2016). "Implementasi Load Balancing Pada Web *Server* Dengan Menggunakan Apache." *Jurnal Manajemen Informatika.*, 5(2), 117–125.
- Putra Perdana, Wawan and Noptin Harpawi. 2013. "Pengontrolan Jarak Jauh Menggunakan Email Application." *Jurnal Teknik Elektro dan Komputer* I, No. I (May 2013): 91–98. www.atmel.com.
- Putri, Fitria Nova Hulu and Maharani. 2019. "Metode Analitis Enkripsi Dan Dekripsi Dengan Penerapan Algoritma Kriptografi Klasik Ke dalam cipher." *Jurnal Elektro dan Telekomunikasi*: 26–34.
- Rodney R Rohrmann, V. J. E. and D. M. W. P. (2017). Large scale *port* scanning through tor using parallel Nmap scans to scan large *portions* of the IPv4 range. *2017 IEEE International Conference on Intelligence and Security Informatics: Security and Big Data, ISI 2017*, 185–187. <https://doi.org/10.1109/ISI.2017.8004906>
- Qamal, Mukti. 2014. "Kriptografi *File* Citra Menggunakan Algoritma Tea (Tiny Encryption Algorithm)." *Journal Unimal (e-Jurnal Universitas Malikussaleh)* 5: 11–33.
- Rochimah, Siti, and Kusbandono Ari Bowo. 2006. "Perangkat Lunak Digital Signage Manager." *JUTI: Jurnal Ilmiah Teknologi Informasi* 5(2): 66.

- Sakti, Batara, Abdul Aziz, and Afrizal Doewes. 2016. "Uji Kelayakan Implementasi Ssh Sebagai Pengaman Ftp Server Sengan Penetration Testing." *Jurnal Teknologi & Informasi ITSmart* 2(1): 44.
- Santoso, Darryl, Agustinus Noertjahyana, and Justinus Andjarwirawan. "Implementasi Dan Analisa Snort Dan Suricata Sebagai Ids Dan Ips Untuk Mencegah Serangan Dos dan Ddos."
- Saputro, Andik, Nanang Saputro, Hendro Wijayanto, and Program Studi Informatika. 2020. "Metode Demilitarized Zone Dan *Port Knocking* Untuk Demilitarized Zone And *Port Knocking* Methods For Komputer." 3(2): 22–27.
- Sinaga, B O, S Sinurat, and T Zebua. 2021. "Modifikasi Algoritma Xtea Dengan Pembangkitan Kunci Menggunakan Metode Linear Congruential Untuk Pengamanan *File* Dokumen." *Journal of Informatics ...* 1(4): 144–52. <http://hostjournals.com/jimat/article/view/130>.
- Sitinjak, Suriski, and Yuli Fauziah. 2010. "Aplikasi Kriptografi *File* Menggunakan Algoritma Blowfish." *semnasIF* 2010(1979–2328): 78–86.
- Suhendar, Anggi Sri Septiani, Haruno Sajati, and Yenni Astuti. 2013. "Perancangan Agoritma Anggi (AA) Dengan Memanfaatkan Diffie-Hellman Dan Ronald Rivest (Rc4) Untuk Membangun Sistem Keamanan Berbasis *Port Knocking*." *Compiler* 2(2): 59–66.
- Syahrir, Yosua Y.Y., Xaverius B.N. Najoan, and Alicia A.E. Sinsuw. 2018. "Rancang Bangun Aplikasi Cross Protocol Email dan Sms." *Jurnal Teknik Informatika* 13(1).
- Yee Hunn, Stephanie Ang, Siti Zarina Siti, and Norina Binti Idris. 2012. "The Development Of Tiny Encryption Algorithm (Tea) Crypto-Core For Mobile Systems." *International Conference on Electronic Devices, Systems, and Applications*: 45–49.
- Yewale, Ms Pratiksha R. 2014. "A Modified Hybrid *Port Knocking* Technique For Host Authentication .": 673–77.
- Yusfrizal, Yusfrizal. 2019. "Rancang Bangun Aplikasi Kriptografi Pada Teks Menggunakan Metode Reverse Chiper Dan Rsa Berbasis Android." *Jurnal Teknik Informatika Kaputama (JTIK)* 3(2): 29–37.
- Yuta Inoue, Seiya Kato, Aamir, Katsunari and Tsutomu. 2020. "Empowering Resource-Constraint Iot Gateways With *Port Knocking* Security." : 362–67.

LAMPIRAN

Lampiran 1: *Attacker* Melakukan Penyadapan Pada *Server*

Server Keadaan Normal

1. *Port* SSH (22)

Dari informasi yang didapatkan ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* dapat mengakses *port* SSH.

```
(root@rifqi)~/home/rifqi
# ssh rifqi@192.168.137.2
rifqi@192.168.137.2's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Aug 17 11:09:45 AM UTC 2023

System load:  0.53515625   Processes:            258
Usage of /:   31.0% of 9.75GB   Users logged in:     1
Memory usage: 9%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

 * Introducing Expanded Security Maintenance for Applications.
 * Receive updates to over 25,000 software packages with your
 * Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

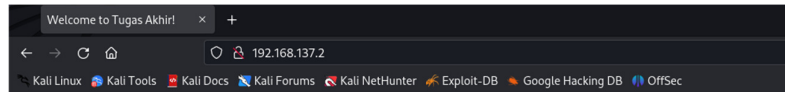
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Thu Aug 17 11:08:04 2023
rifqi@rifqi:~$
```

2. *Port* HTTP (80)

Dari informasi yang didapatkan yaitu ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* dapat mengakses *port* HTTP.



Sukses Tugas Akhir, Lulus 2023!

3. Port FTP (21)

Dari informasi yang didapatkan yaitu ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* dapat mengakses *port* FTP.

```
(root@rifqi)-[/home/rifqi]
# ftp -p 192.168.137.2
Connected to 192.168.137.2.
220 (vsFTPD 3.0.5)
Name (192.168.137.2:rifqi): politeknik
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

4. Port SMTP (25)

Dari informasi yang didapatkan yaitu ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* dapat mengakses *port* SMTP.

```
(root@rifqi)-[/home/rifqi]
# telnet 192.168.137.2 25
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^]'.
220 rifqi ESMTP Postfix (Ubuntu)
█
```


5. Port TELNET (23)

Dari informasi yang didapatkan ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* dapat mengakses *port* TELNET.

```
[root@rifqi]~/home/rifqi
└─# telnet 192.168.137.2
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^]'.
Ubuntu 22.04.2 LTS
rifqi login: rifqi
Password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu Aug 17 11:15:11 AM UTC 2023

System load:  0.16552734375   Processes:            238
Usage of /:   31.0% of 9.75GB   Users logged in:     1
Memory usage: 10%           IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Thu Aug 17 11:09:47 UTC 2023 from 192.168.137.4 on pts/1
rifqi@rifqi:~$ █
```

Server Menerapkan Port Knocking

1. Port SSH (22)

Dari informasi yang didapatkan ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* dapat mengakses *port* SSH.

```
(rifqi@rifqi)-[~]
└─$ knock -v 192.168.137.2 3647 6029 4500
hitting tcp 192.168.137.2:3647
hitting tcp 192.168.137.2:6029
hitting tcp 192.168.137.2:4500

(rifqi@rifqi)-[~]
└─$ ssh rifqi@192.168.137.2
rifqi@192.168.137.2's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Aug 16 02:24:17 PM UTC 2023

System load:  0.0244140625   Processes:           223
Usage of /:   30.9% of 9.75GB Users logged in:        1
Memory usage: 10%          IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

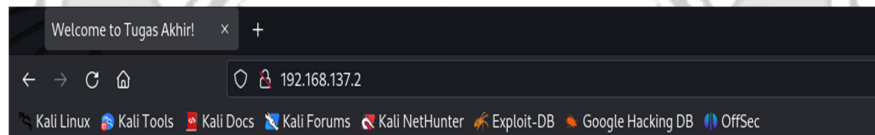
Last login: Wed Aug 16 14:24:17 2023 from 192.168.137.4
rifqi@rifqi:~$ █
```

2. Port HTTP (80)

Dari informasi yang didapatkan yaitu ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* dapat mengakses *port* HTTP.

```
(rifqi@rifqi)-[~]
└─$ knock -v 192.168.137.2 2489 3872 1200 7381
hitting tcp 192.168.137.2:2489
hitting tcp 192.168.137.2:3872
hitting tcp 192.168.137.2:1200
hitting tcp 192.168.137.2:7381

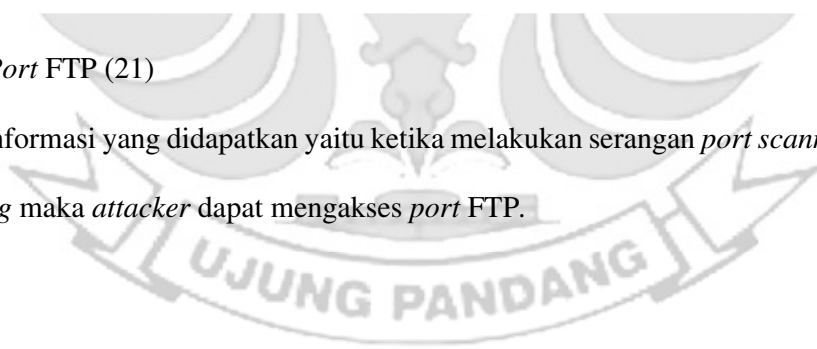
(rifqi@rifqi)-[~]
└─$
```



Sukses Tugas Akhir, Lulus 2023!

3. Port FTP (21)

Dari informasi yang didapatkan yaitu ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* dapat mengakses *port* FTP.



```
(rifqi@rifqi)~]
└─$ knock -v 192.168.137.2 3892 4820 5390 2680
hitting tcp 192.168.137.2:3892
hitting tcp 192.168.137.2:4820
hitting tcp 192.168.137.2:5390
hitting tcp 192.168.137.2:2680

(rifqi@rifqi)~]
└─$ ftp -p 192.168.137.2
Connected to 192.168.137.2.
220 (vsFTPD 3.0.5)
Name (192.168.137.2:rifqi): politeknik
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

4. Port SMTP (25)

Dari informasi yang didapatkan yaitu ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* dapat mengakses *port* SMTP.

```
(rifqi@rifqi)~]
└─$ knock -v 192.168.137.2 1400 1500 1600 1700 1800
hitting tcp 192.168.137.2:1400
hitting tcp 192.168.137.2:1500
hitting tcp 192.168.137.2:1600
hitting tcp 192.168.137.2:1700
hitting tcp 192.168.137.2:1800

(rifqi@rifqi)~]
└─$ telnet 192.168.137.2 25
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^]'.
220 rifqi ESMTD Postfix (Ubuntu)
```

5. Port TELNET (23)

Dari informasi yang didapatkan ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* dapat mengakses *port* TELNET.

```
(root@rifqi)~/home/rifqi
# knock -v 192.168.137.2 7324 3429 9125
hitting tcp 192.168.137.2:7324
hitting tcp 192.168.137.2:3429
hitting tcp 192.168.137.2:9125

(root@rifqi)~/home/rifqi
# telnet 192.168.137.2
Trying 192.168.137.2...
Connected to 192.168.137.2.
Escape character is '^]'.
Ubuntu 22.04.2 LTS
rifqi login: rifqi
Password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Wed Aug 16 02:33:28 PM UTC 2023

System load:  0.0224609375      Processes:            227
Usage of /:   30.9% of 9.75GB   Users logged in:     1
Memory usage: 10%              IPv4 address for ens33: 192.168.137.2
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

Server Menerapkan *Port Knocking* dan Algoritma XTEA

1. *Port* SSH (22)

Dari informasi yang didapatkan ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* tidak dapat mengakses *port* SSH.

```
(rifqi@ rifqi)-[~]
└─$ knock -v 192.168.137.2 3667 6039 5000
hitting tcp 192.168.137.2:3667
hitting tcp 192.168.137.2:6039
hitting tcp 192.168.137.2:5000

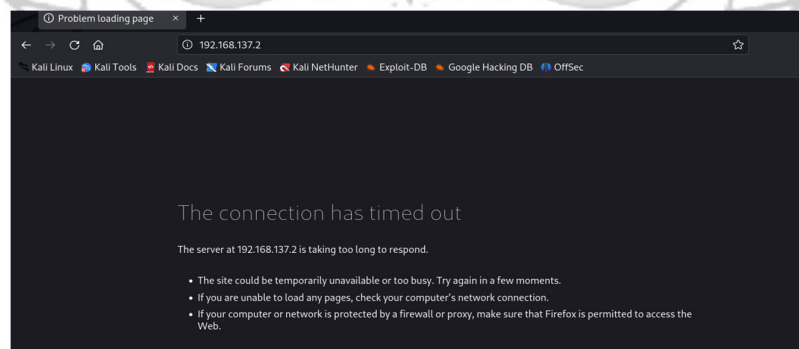
(rifqi@ rifqi)-[~]
└─$ ssh rifqi@192.168.137.2
```

2. *Port* HTTP (80)

Dari informasi yang didapatkan yaitu ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* tidak dapat mengakses *port* HTTP.

```
(rifqi@ rifqi)-[~]
└─$ knock -v 192.168.137.2 2499 3873 1500
hitting tcp 192.168.137.2:2499
hitting tcp 192.168.137.2:3873
hitting tcp 192.168.137.2:1500

(rifqi@ rifqi)-[~]
└─$
```



3. Port FTP (21)

Dari informasi yang didapatkan yaitu ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* tidak dapat mengakses *port* FTP.

```
(rifqi@rifqi)-[~]
└─$ knock -v 192.168.137.2 3894 4830 5380 2680
hitting tcp 192.168.137.2:3894
hitting tcp 192.168.137.2:4830
hitting tcp 192.168.137.2:5380
hitting tcp 192.168.137.2:2680

(rifqi@rifqi)-[~]
└─$ ftp -p 192.168.137.2
```

4. Port SMTP (25)

Dari informasi yang didapatkan yaitu ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* tidak dapat mengakses *port* SMTP.

```
(rifqi@rifqi)-[~]
└─$ knock -v 192.168.137.2 1500 1600 1700 1800
hitting tcp 192.168.137.2:1500
hitting tcp 192.168.137.2:1600
hitting tcp 192.168.137.2:1700
hitting tcp 192.168.137.2:1800

(rifqi@rifqi)-[~]
└─$ telnet 192.168.137.2 25
Trying 192.168.137.2...
```

5. Port TELNET (23)

Dari informasi yang didapatkan ketika melakukan serangan *port scanning* dan *sniffing* maka *attacker* tidak dapat mengakses *port* TELNET.

```
(rifqi@rifqi)-[~]
└─$ knock -v 192.168.137.2 7334 3429 9125
hitting tcp 192.168.137.2:7334
hitting tcp 192.168.137.2:3429
hitting tcp 192.168.137.2:9125

(rifqi@rifqi)-[~]
└─$ telnet 192.168.137.2
Trying 192.168.137.2...
```

Lampiran 2: Penjelasan Script

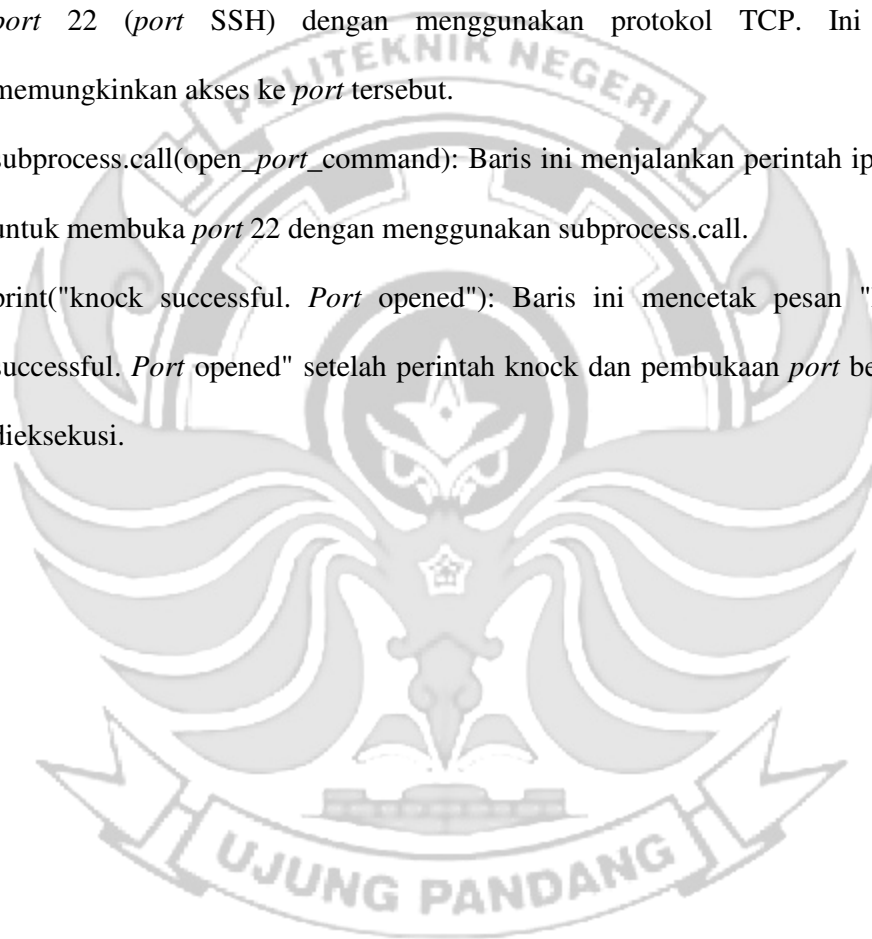
SCRIPT PORT KNOCKING

```
from binascii import hexlify
import subprocess
text1 = input("Input text1: ")
text2 = input("Input text2: ")
text3 = input("Input text3: ")
open_port_command = ["iptables", "-I", "INPUT", "-p", "tcp", "--dport", "22", "-j", "ACCEPT"]
subprocess.call(open_port_command)
print("knock successful.Port opened")
```

Keterangan:

1. `from binascii import hexlify`: Baris ini mengimpor fungsi `hexlify` dari modul `binascii`. Fungsi ini digunakan untuk mengonversi data biner menjadi representasi heksadesimal.
2. `import subprocess`: Baris ini mengimpor modul `subprocess` yang digunakan untuk menjalankan perintah shell dari dalam skrip Python.
3. `text1 = input("Input text1: ")`, `text2 = input("Input text2: ")`, `text3 = input("Input text3: ")`: Tiga baris ini meminta input dari pengguna untuk tiga teks yang akan digunakan dalam perintah "knock".

4. `subprocess.call(knock_command_plaintext)`: Baris ini menjalankan perintah "knock" dengan menggunakan `subprocess.call`. Perintah "knock" akan dieksekusi dengan argumen yang telah ditentukan sebelumnya.
5. `open_port_command = ["iptables", "-I", "INPUT", "-p", "tcp", "--dport", "22", "-j", "ACCEPT"]`: Baris ini membuat daftar perintah iptables untuk membuka *port* 22 (*port* SSH) dengan menggunakan protokol TCP. Ini akan memungkinkan akses ke *port* tersebut.
6. `subprocess.call(open_port_command)`: Baris ini menjalankan perintah iptables untuk membuka *port* 22 dengan menggunakan `subprocess.call`.
7. `print("knock successful. Port opened")`: Baris ini mencetak pesan "knock successful. *Port* opened" setelah perintah knock dan pembukaan *port* berhasil dieksekusi.



SCRIPT ALGORITMA XTEA

```
def xtea_encrypt(plain_text, key):
    # Konversi teks biasa menjadi blok 64-bit
    block = bytearray(plain_text.encode('utf-8'))
    while len(block) % 8 != 0:
        block.append(0)
    # Split key menjadi empat bagian
    k = [0] * 4
    for i in range(4):
        k[i] = int.from_bytes(key[i*4:(i*4)+4], byteorder='big')
    # Inisialisasi variabel
    delta = 0x9e3779b9
    sum_ = 0
    rounds = 32
    # Enkripsi
    for _ in range(rounds):
        v0, v1 = int.from_bytes(block[:4], byteorder='big'), int.from_bytes(block[4:],
byteorder='big')
        delta_sum = (sum_ & 3) * delta
        v0 += (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum_ + k[(sum_ & 3) ^ delta_sum])
        sum_ += delta
        v1 += (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum_ + k[(sum_ >> 11) & 3] ^
delta_sum)
        block[:4], block[4:] = v0.to_bytes(4, byteorder='big'), v1.to_bytes(4,
byteorder='big')
```

Keterangan:

1. `block = bytearray(plain_text.encode('utf-8'))`: Baris ini mengonversi teks biasa (`plain_text`) menjadi blok byte dengan menggunakan encoding UTF-8. Blok byte ini akan menjadi blok masukan untuk enkripsi.
2. `while len(block) % 8 != 0: block.append(0)`: Baris ini menambahkan padding byte dengan nilai 0 ke blok byte jika panjangnya tidak memenuhi persyaratan panjang blok XTEA yang harus kelipatan 8. Hal ini dilakukan untuk memastikan bahwa blok masukan memiliki panjang yang benar.
3. `k = [0] * 4` dan `for i in range(4): k[i] = int.from_bytes(key[i*4:(i*4)+4], byteorder='big')`: Baris ini membagi kunci (`key`) menjadi empat bagian 32-bit dan menyimpannya dalam list `k` sebagai bilangan bulat. Setiap bagian kunci diambil dari `key` menggunakan `int.from_bytes` dengan urutan byte big-endian.
4. `delta = 0x9e3779b9`, `sum_ = 0`, dan `rounds = 32`: Variabel `delta` menyimpan nilai delta yang digunakan dalam algoritma XTEA. Variabel `sum_` digunakan untuk menghitung penjumlahan delta dalam setiap putaran enkripsi. Variabel `rounds` menyimpan jumlah putaran enkripsi yang akan dilakukan.
5. Loop `for _ in range(rounds)`: Ini adalah loop yang melakukan enkripsi dalam jumlah putaran yang ditentukan.
6. `v0, v1 = int.from_bytes(block[:4], byteorder='big'), int.from_bytes(block[4:], byteorder='big')`: Baris ini mengambil dua bagian blok 32-bit dari blok byte yang akan dienkripsi. Nilai `v0` dan `v1` menyimpan bagian-bagian ini sebagai bilangan bulat.
7. `delta_sum = (sum_ & 3) * delta`: Variabel `delta_sum` menyimpan hasil perkalian antara nilai delta dan hasil operasi bitwise dari `sum_ & 3`. Operasi `&` dengan bilangan 3 digunakan untuk mendapatkan nilai modulo 4 dari `sum_`.
8. `v0 += (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum_ + k[sum_ & 3] ^ delta_sum)`: Baris ini melakukan operasi enkripsi pada `v0`. Operasi tersebut termasuk pergeseran bit (`<<` dan `>>`), operasi XOR (`^`), dan penjumlahan (`+`).
9. `sum_ += delta`: Nilai delta ditambahkan pada `sum_` untuk meng-update variabel tersebut.

10. $v1 += (((v0 \ll 4) \wedge (v0 \gg 5)) + v0) \wedge (\text{sum_} + k[(\text{sum_} \gg 11) \& 3] \wedge \text{delta_sum})$: Baris ini melakukan operasi enkripsi pada v1 dengan pola serupa seperti pada langkah 8.

11. `block[:4], block[4:] = v0.to_bytes(4, byteorder='big'), v1.to_bytes(4, byteorder='big')`: Hasil enkripsi v0 dan v1 dikonversi kembali menjadi blok byte dengan menggunakan `to_bytes` dan kemudian disimpan kembali ke dalam blok byte awal.



SCRIPT *PORT KNOCKING* DAN ALGORITMA XTEA

```
from binascii import hexlify
import subprocess
def xtea_encrypt(plain_text, key):
    # Konversi teks biasa menjadi blok 64-bit
    block = bytearray(plain_text.encode('utf-8'))
    while len(block) % 8 != 0:
        block.append(0)
    # Split key menjadi empat bagian
    k = [0] * 4
    for i in range(4):
        k[i] = int.from_bytes(key[i*4:(i*4)+4], byteorder='big')
    # Inisialisasi variabel
    delta = 0x9e3779b9
    sum_ = 0
    rounds = 32

    # Enkripsi
    for _ in range(rounds):
        v0, v1 = int.from_bytes(block[:4], byteorder='big'),
        int.from_bytes(block[4:], byteorder='big')
        delta_sum = (sum_ & 3) * delta
        v0 += (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum_ + k[(sum_ & 3) ^
        delta_sum)
        sum_ += delta
        v1 += (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum_ + k[(sum_ >> 11) & 3] ^
        delta_sum)
        block[:4], block[4:] = v0.to_bytes(4, byteorder='big'), v1.to_bytes(4,
        byteorder='big')
```

```

text1 = input("Input text1: ")
text2 = input("Input text2: ")
text3 = input("Input text3: ")

open_port_command = ["iptables", "-I", "INPUT", "-p", "tcp", "--dport", "22",
                    "-j", "ACCEPT"]
subprocess.call(open_port_command)
print("knock successful. Port opened")

```

Keterangan:

Pada penelitian ini proses algoritma xtea berjalan pada sisi *admin* knockd, jadi algoritma xtea tersebut disisipkan untuk melakukan enkripsi pada sequence *port* yang terpada pada *server* di file */etc/knockd.conf*.

1. `from binascii import hexlify`: Baris ini mengimpor fungsi `hexlify` dari modul `binascii`. Fungsi ini digunakan untuk mengonversi data biner menjadi representasi heksadesimal.
2. `import subprocess`: Baris ini mengimpor modul `subprocess` yang digunakan untuk menjalankan perintah shell dari dalam skrip Python.
3. `text1 = input("Input text1: ")`, `text2 = input("Input text2: ")`, `text3 = input("Input text3: ")`: Tiga baris ini meminta input dari pengguna untuk tiga teks yang akan digunakan dalam perintah "knock".
4. `subprocess.call(knock_command_plaintext)`: Baris ini menjalankan perintah "knock" dengan menggunakan `subprocess.call`. Perintah "knock" akan dieksekusi dengan argumen yang telah ditentukan sebelumnya.

5. `open_port_command = ["iptables", "-I", "INPUT", "-p", "tcp", "--dport", "22", "-j", "ACCEPT"]`: Baris ini membuat daftar perintah iptables untuk membuka *port 22* (*port* SSH) dengan menggunakan protokol TCP. Ini akan memungkinkan akses ke *port* tersebut.
6. `subprocess.call(open_port_command)`: Baris ini menjalankan perintah iptables untuk membuka *port 22* dengan menggunakan `subprocess.call`.
7. `print("knock successful. Port opened")`: Baris ini mencetak pesan "knock successful. *Port* opened" setelah perintah knock dan pembukaan *port* berhasil dieksekusi.
8. `block = bytearray(plain_text.encode('utf-8'))`: Baris ini mengonversi teks biasa (*plain_text*) menjadi blok byte dengan menggunakan encoding UTF-8. Blok byte ini akan menjadi blok masukan untuk enkripsi.
9. `while len(block) % 8 != 0: block.append(0)`: Baris ini menambahkan padding byte dengan nilai 0 ke blok byte jika panjangnya tidak memenuhi persyaratan panjang blok XTEA yang harus kelipatan 8. Hal ini dilakukan untuk memastikan bahwa blok masukan memiliki panjang yang benar.
10. `k = [0] * 4` dan `for i in range(4): k[i] = int.from_bytes(key[i*4:(i*4)+4], byteorder='big')`: Baris ini membagi kunci (*key*) menjadi empat bagian 32-bit dan menyimpannya dalam list *k* sebagai bilangan bulat. Setiap bagian kunci diambil dari *key* menggunakan `int.from_bytes` dengan urutan byte big-endian.
11. `delta = 0x9e3779b9`, `sum_ = 0`, dan `rounds = 32`: Variabel *delta* menyimpan nilai *delta* yang digunakan dalam algoritma XTEA. Variabel *sum_* digunakan untuk

menghitung penjumlahan delta dalam setiap putaran enkripsi. Variabel rounds menyimpan jumlah putaran enkripsi yang akan dilakukan.

12. `Loop for _ in range(rounds)`: Ini adalah loop yang melakukan enkripsi dalam jumlah putaran yang ditentukan.
13. `v0, v1 = int.from_bytes(block[:4], byteorder='big'), int.from_bytes(block[4:], byteorder='big')`: Baris ini mengambil dua bagian blok 32-bit dari blok byte yang akan dienkripsi. Nilai v0 dan v1 menyimpan bagian-bagian ini sebagai bilangan bulat.
14. `delta_sum = (sum_ & 3) * delta`: Variabel delta_sum menyimpan hasil perkalian antara nilai delta dan hasil operasi bitwise dari `sum_ & 3`. Operasi `&` dengan bilangan 3 digunakan untuk mendapatkan nilai modulo 4 dari `sum_`.
15. `v0 += (((v1 << 4) ^ (v1 >> 5)) + v1) ^ (sum_ + k[sum_ & 3] ^ delta_sum)`: Baris ini melakukan operasi enkripsi pada v0. Operasi tersebut termasuk pergeseran bit (`<<` dan `>>`), operasi XOR (`^`), dan penjumlahan (`+`).
16. `sum_ += delta`: Nilai delta ditambahkan pada `sum_` untuk meng-update variabel tersebut.
17. `v1 += (((v0 << 4) ^ (v0 >> 5)) + v0) ^ (sum_ + k[(sum_ >> 11) & 3] ^ delta_sum)`: Baris ini melakukan operasi enkripsi pada v1 dengan pola serupa seperti pada langkah 8.
18. `block[:4], block[4:] = v0.to_bytes(4, byteorder='big'), v1.to_bytes(4, byteorder='big')`: Hasil enkripsi v0 dan v1 dikonversi kembali menjadi blok byte dengan menggunakan `to_bytes` dan kemudian disimpan kembali ke dalam blok byte awal.