randori

A low interaction honeypot with a vengeance

@avuko

Tuesday 17 October 2017

Introduction

It is not a war out there.

It is a pandemic.

Introduction: About me

- IR@KPN-CERT (NL)
- Historian (MA)
- avuko
- @avuko

Conception

Conception

Randori (乱取り) is a practice
in which a designated aikidoka
defends against multiple attackers
in quick succession

Conception: Simple insight

- RDP brute force attack
- Attacker only has port 3389 (RDP) open
- !

Conception: github.com/avuko/aiki

```
// create a private key used by the SSHd to encrypt communications
func buildkeys() (priv_pem []byte) {
}
// set up non-bruteforcable account details
func unguessable() (username string, password string) {
}
// ssh client that can reuse captured usernames and passwords
func aiki(ip string, username string, password string) {
}
func main() {
// start fake SSHd server
config := &ssh.ServerConfig{
}
// connect back to anyone connecting to the fake SSHd server
go aiki(ip, username, password)
}
```

Conception: Randori vs aiki

https://github.com/avuko/randori

- Support more protocols
- Stop building “fake” services
- Scale to keep up with the bots

Techniques

Techniques

- PAM for logging
- sshd/telnetd for listening
- Golang/ØMQ for randori
- SQLite/Redis/Graphviz for analysis

Techniques: P(wn) A(ll) M(alware)

- ONsec-Lab gave us `pam_steal`
- Captures successful logins, I wanted failed ones

(ONsec-Lab: https://github.com/ONsec-Lab/scripts/tree/master/pam_steal)

Techniques: pam_randori

```c
retval=pam_get_item(pamh, PAM_SERVICE, &servicename);
retval=pam_get_item(pamh, PAM_RHOST, &rhostname);
retval=pam_get_user(pamh, &username, NULL);
retval=pam_get_item(pamh, PAM_AUTHTOK, &password);
log = fopen (LOGFILE, "a");
[...]

fprintf(log, "%s\t%s\t%s\t%s\t%s\n", (char *) timestamp,
                   (char *) servicename, (char *) rhostname,
                       (char *) username, (char *) password);
fclose( log);
}
```

ØMQ tails this log and fans it out to randori workers

(Yes, I just copy-pasted the code from the ØMQ website)

Techniques: PAM integration

- Telnetd just needed xinetd
- Still working on xrdp, vnc
- 31337 patching of OpenSSH

Techniques: OpenSSH (auth-pam.c)

diff ./auth-pam.c ../randori/deploy/auth-pam.c

```
820c820
<   const char junk[] = "\b\n\r\177INCORRECT";
---
>   /* const char junk[] = "\b\n\r\177INCORRECT"; */
829c829,830
<       ret[i] = junk[i % (sizeof(junk) - 1)];
---
>       /* ret[i] = junk[i % (sizeof(junk) - 1)]; */
>       ret[i] = wire_password[i];
```

Techniques: Increasing attempts

Bots have a hard time handling ~~anything~~:

- Authentication delays
- Connection limitations
- Max # of authentication attempts
- Strong(ish) ciphers

Techniques: Increasing attempts: telnetd

```
defaults
{
        instances = unlimited
        cps = 2000 1
        per_source = 2000
}
```

Techniques: Increasing attempts: sshd

```
Ciphers chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,
   aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com,
   arcfour256,arcfour128,aes128-cbc
HostKeyAlgorithms ssh-rsa,rsa-sha2-512,rsa-sha2-256,
   ecdsa-sha2-nistp256,ssh-ed25519
KexAlgorithms curve25519-sha256@libssh.org,ecdh-sha2-nistp256,
   ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group1-sha1,
   diffie-hellman-group14-sha1,diffie-hellman-group-exchange-sha256
MACs 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, hmac-sha1-96
MaxStartups 1000
MaxSessions 500
MaxAuthTries 100
```

Techniques: Randori principles

- Try all usernames/passwords the attacker uses
- Try only those credentials, nothing more
- Back out as early as possible
- Try not to execute code
- Resist temptation

Techniques: Telnet randori

toritelnet.go (telnet is ugly)

```
func noyoudont() {
    // IAC: do -> don't ; will -> wont
    }
func authcheck(ip, username, password string) (response []byte) {
    connect()
    noyoudont()
    func read_and_ignore_buf()
    func write_username()
    func read_and_ignore_buf()
    func write_password()
    func read_and_ignore_buf()
    response := read_and_store_buf()
}
func ip_user_pass_response_to_zeromq()
```

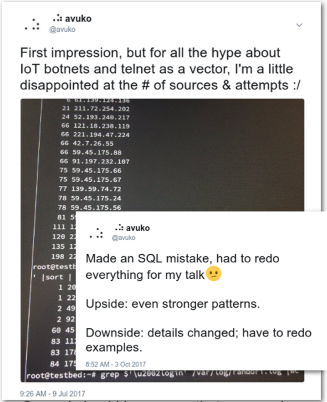
Techniques: SSH randori

torissh.go (so much cleaner than telnet)

```
func authcheck(ip, username, password string) (response []byte) {
    sshconnect()
    response := get_ssh_server_version()
}
func ip_user_pass_response_to_zeromq()
```

Techniques: So much fail!

- Spent too much time trying to increase attempts
- ØMQ should not be logging into syslog
- Constant issues with disk space
- No clean way to get SSH agents/ IAC commands
- No RDP/VNC support yet

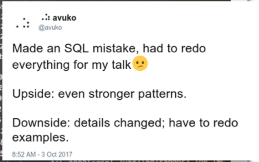


.:: avuko
@avuko

First impression, but for all the hype about IoT botnets and telnet as a vector, I'm a little disappointed at the # of sources & attempts :/

```
      6 61.139.124.136
     21 211.72.254.202
     24 52.193.240.217
     66 121.18.238.119
     66 221.194.47.224
     66 42.7.26.55
     66 59.45.175.88
     66 91.197.232.107
     75 59.45.175.66
     75 59.45.175.67
     77 139.59.74.72
     78 59.45.175.24
     78 59.45.175.56
     81 5
    111 1
    120 2
    135 1
    198 2
root@testb
' |sort |
      1 20
      1 22
      2 49
      2 92
     60 45
     83 11
     83 17
     84 17
root@testbed:~# grep $'\u2002login' /var/log/randori.log |wc
```

9:26 AM - 9 Jul 2017



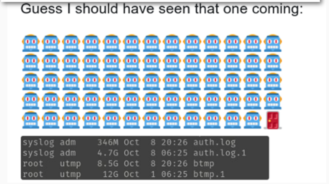
.:: avuko
@avuko

Made an SQL mistake, had to redo everything for my talk😕

Upside: even stronger patterns.

Downside: details changed; have to redo examples.

8:52 AM - 3 Oct 2017



Guess I should have seen that one coming:

🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖
🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖
🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖
🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖
🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🤖🚪

```
syslog adm    346M Oct  8 20:26 auth.log
syslog adm    4.7G Oct  8 06:25 auth.log.1
root   utmp   8.5G Oct  8 20:26 btmp
root   utmp    12G Oct  1 06:25 btmp.1
```

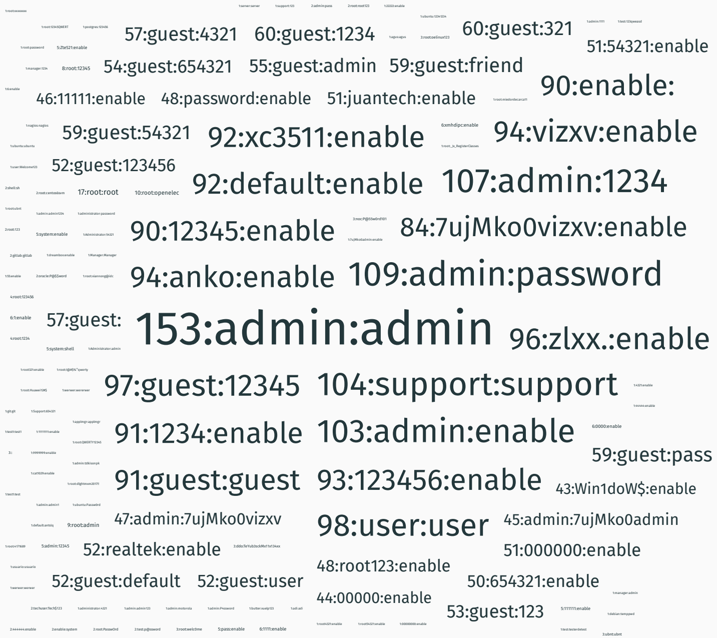
Results

Results

A ten ton catastrophe,
on a sixty pound chain

[Nick Cave, Jubilee Street]

Results: Infection vectors



Results: I changed my mind

I was so focused on the war,

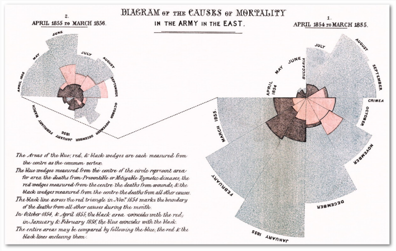
I forgot about the casualties

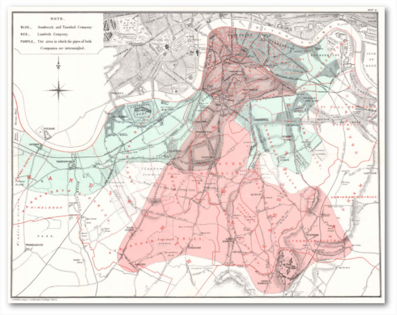
Results: hacker vs IR

Ethical hacker	Incident Responder
Botnets are a weapon of war	Botnets are a disease
randori: Offensive/ hacking back	randori: Analyse systemic weaknesses, catalogue infections
IoT as a battlefield	IoT epidemiology

Results: Medicine model

Study botnets as pathogens living in host populations




NOTE.
BLUE.. Southwark and Vauxhall Company
RED.. Lambeth Company
PURPLE.. The area in which the pipes of both Companies are intermingled.

Data visualisation by Florence Nightingale (1858), Cholera experiment by John Snow (1854)

Results: Studying botnets as pathogens

- Create a table of IPs with a string of distinct SSH clients observed

```
distinct(clients.ip),group_concat(distinct(clients.client))
```

- Create a table of IPs with a string of all user/password combos used

```
distinct(ip),group_concat(user,pass)
```

- Create an ssdeep of the ssh client strings and credentials

```
ssdeep.hash(distinct_clients + all_user_pass_combos)
```

Results: How ssdeep can help

- Context triggered piecewise hashes (CTPH)
- High tolerance for the “fuzziness” of bruteforce attacks
- Compare ssdeep hashes to see if inputs are similar

Results: Ssdeep example

```
ssdeep.hash(”libssh2_1.7.0|adminasdf123adminasdf123adminasdf123
            admin1q2w3e4radmin1q2w3e4radmin1q2w3e4r”)

   ’3:EWKv8Vz+IXLEWIXLEWIXLoi+KU9i+KU9R:EWKvEz+qwWqwWqUinU9inU9R’


ssdeep.hash(”libssh2_1.7.0|adminasdf123adminasdf123adminasdf123
            adminabc123@adminabc123@adminabc123@”)

   ’3:EWKv8Vz+IXLEWIXLEWIXLEHTuTuG:EWKvEz+qwWqwWqwy’


ssdeep.compare(”3:EWKv8Vz+IXLEWIXLEWIXLEHTuTuG:EWKvEz+qwWqwWqwy”,
   ”3:EWKv8Vz+IXLEWIXLEWIXLoi+KU9i+KU9R:EWKvEz+qwWqwWqUinU9inU9R”)

   >>> 32
```

Results: Botnet strain grouping with ssdeep and Redis

https://github.com/avuko/kathe

```
ssdeep string:
3:aabbccddee:ccdd...
  aabbccd    ccdd...
   abbccdd    cdd...
    bbccdde
```

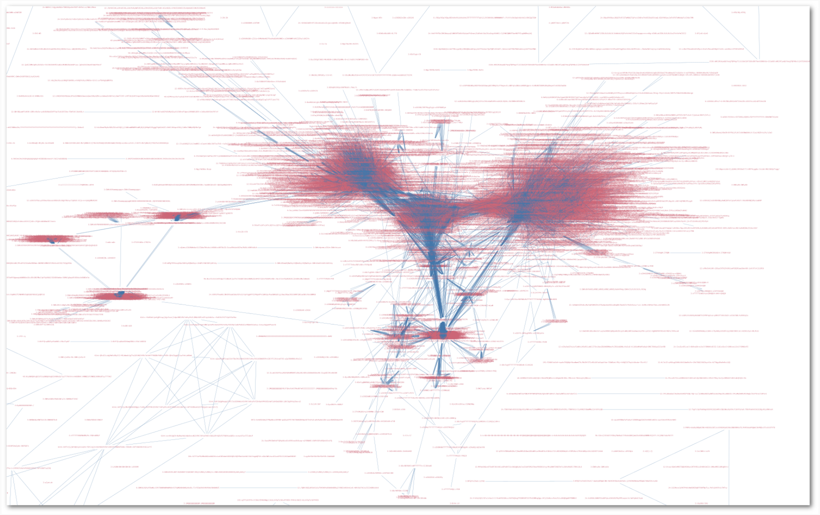
```
rolling_window:
3:aabbccd
...
3:aabbccddee:ccdd
3:aabbccdffg:ccde
```

```
ssdeep:
3:aabbccddee:ccdd
3:aabbccdffg:ccde,9
...
...
```

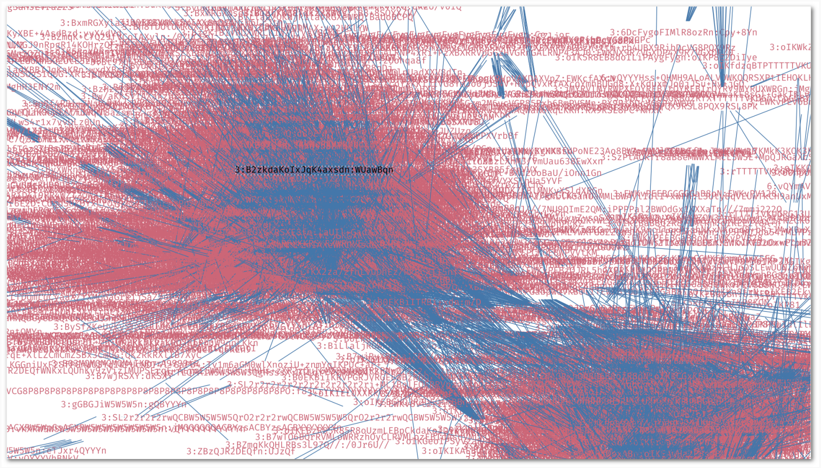
```
info:sha256:2dbcs353adef5de...
ssdeep:ip

info:filename:198.51.100.2
sha256:ssdeep

info:ssdeep:3:aabbccddee:cc...
sha256:ip
```

prior art: Brian Wallace, ssdeep at scale (2015), https://github.com/bwall/ssdc

Results: Botnet strains attacking a honeypot



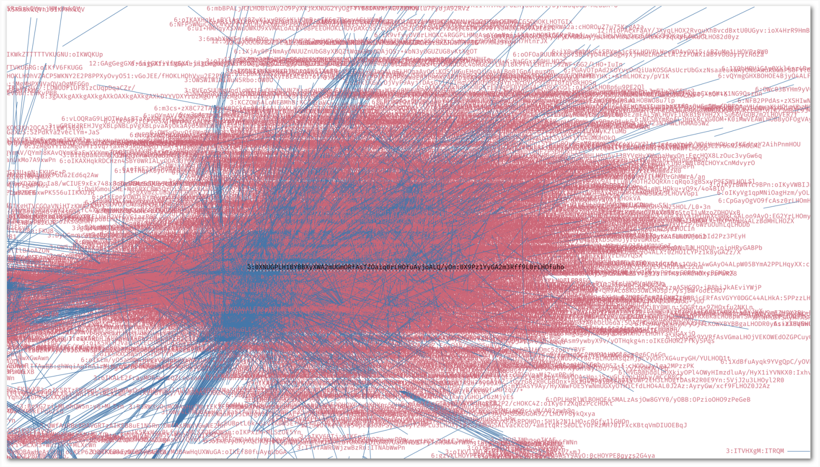
Results: Left cluster



Results: Left cluster

```
smembers info:ssdeep:3:B2zkdaKoIxJqK4axsdn:WUawBqn
1) ”sha256:f1bd01791c71e0c8e74b8f0e245a4628bb5d90b
           3a67db2d6f1a1749a1ea14d85:
  filename:198.51.100.194”

select * from attacks1 where ip = ’198.51.100.194’;
1|2017-09-07T02:51:12+00:00|sshd|198.51.100.194|root|uClinux
1|2017-09-07T02:51:14+00:00|sshd|198.51.100.194|root|admintrup
1|2017-09-07T02:51:16+00:00|sshd|198.51.100.194|root|admin
1|2017-09-07T02:51:17+00:00|sshd|198.51.100.194|root|Zte521
1|2017-09-07T02:51:19+00:00|sshd|198.51.100.194|root|anko
1|2017-09-07T02:51:22+00:00|sshd|198.51.100.194|root|dreambox
```

Results: Right cluster



Results: Right cluster

```
smembers info:ssdeep:6:0XNUGPLH1BYBBXyXWA2mUGmORfA
         sTZOaiq0rLHOfuAyjoALQ/yOn:0X9Pz1YyGA2m3Rf
         f9L0rLHOfuho
1) ”sha256:ada3c6cf0d498e93dc4752e3ab76a7aa63bcaa6
           a7137f37996b4eef69486f5d8:
  filename:198.51.100.251”
```

Results: Strains, right cluster (MIRAI?)

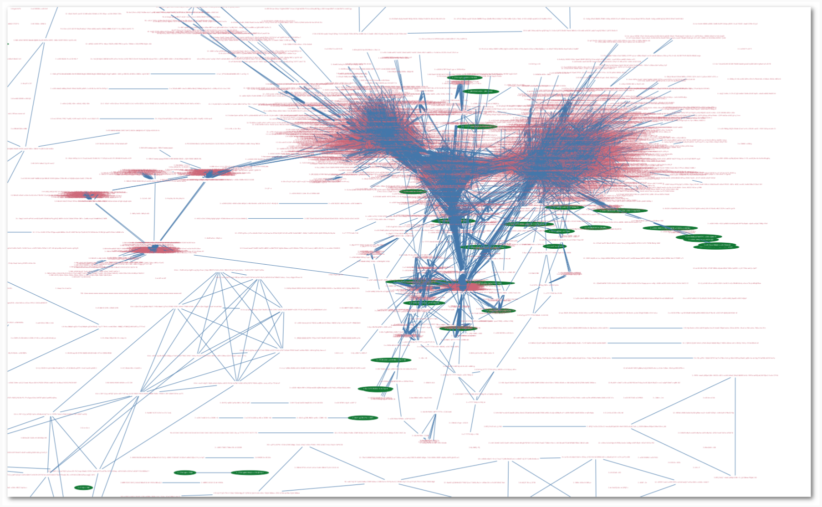
```
select * from attacks1 where ip = '198.51.100.251';
1|2017-08-11T19:25:49+00:00|login|198.51.100.251|guest|guest
1|2017-08-11T19:25:53+00:00|login|198.51.100.251|admin|1234
1|2017-08-11T19:26:17+00:00|login|198.51.100.251|1234|enable
1|2017-08-11T19:26:29+00:00|login|198.51.100.251|support|support
1|2017-08-11T19:26:54+00:00|login|198.51.100.251|default|enable
1|2017-08-11T19:27:06+00:00|login|198.51.100.251|guest|12345
1|2017-08-11T19:27:08+00:00|login|198.51.100.251|admin|password
1|2017-08-11T19:27:26+00:00|login|198.51.100.251|admin|Win1doW$
1|2017-08-11T19:27:51+00:00|login|198.51.100.251|12345|enable
1|2017-08-11T19:28:02+00:00|login|198.51.100.251|system|
1|2017-08-11T19:28:24+00:00|login|198.51.100.251||enable
1|2017-08-11T19:28:57+00:00|login|198.51.100.251|admin|enable
1|2017-08-11T19:29:09+00:00|login|198.51.100.251|user|user
1|2017-08-11T19:29:12+00:00|login|198.51.100.251|admin|7ujMko0admin
1|2017-08-11T19:29:37+00:00|login|198.51.100.251|password|enable
1|2017-08-11T19:30:10+00:00|login|198.51.100.251|zlxx.|enable
```

Results: “Hajime” botnet strains

- Hard to detect: evolving/ adapting
- “Hajime” botnet:

```
pass='5up' and service='login'
```

Results: “Hajime”: hiding in plain sight



Results: “Hajime” sample

```
select * from attacks1 where ip = ’198.51.100.42’;
1|2017-08-05T05:44:26+00:00|login|198.51.100.42|admin|ERRU$
1|2017-08-05T05:44:33+00:00|login|198.51.100.42|osteam|5up
1|2017-08-05T05:44:39+00:00|login|198.51.100.42|admin|adslroot
1|2017-08-05T05:44:47+00:00|login|198.51.100.42|admin|free
1|2017-08-05T05:44:58+00:00|login|198.51.100.42|attack|enable
1|2017-08-05T05:45:03+00:00|login|198.51.100.42|admin|online
1|2017-08-05T05:45:08+00:00|login|198.51.100.42|admin|21232
1|2017-08-05T05:45:13+00:00|login|198.51.100.42|admin|263297
1|2017-08-05T05:45:18+00:00|login|198.51.100.42|user|
1|2017-08-05T05:45:23+00:00|login|198.51.100.42|admin|amvqnekk
```

Ethics

Ethics

I will abstain from all intentional wrong-doing and harm.
Whatsoever I shall see or hear I will never divulge.

Hippocratic Oath, 500-300 BC, paraphrased

Ethics: Could but won’t

- Kill your TV/internet/baby camera
    - Think Brickerbot, perhaps Hajime
- Create my own botnet
    - This trick might already be or might become part of botnet behaviour
- Disclose credentials
    - See the work @GDI_FDN is doing trying to solve recent pastebin dumps
- Steal pictures of/ look at you or your family
    - Your ~~QNAP~~ NAS is probably part of a botnet. So is your baby monitor.

Ethics: Want but can’t

In our current ”Cyberwar” model, these might be crimes:

- Investigate infected devices
    - Study infection vectors, mutations, case fatality rates, basic reproductive ratio
- Help individuals with infected devices
    - Similar to what @GDI_FDN does
- Make vendors responsible for weak security
    - Gather evidence to attribute weaknesses to vendors

Next steps

Next steps

Everybody wants to be a warrior.

Nobody wants to be a nurse.

Next steps: “Hacking back” law?

(ACDC)

- Establish attribution of an attack
- Disrupt cyberattacks without damaging others’ computers
- Retrieve and destroy stolen files
- Monitor the behavior of an attacker
- Utilize beaconing technology

Next steps: Hacking back law (ACDC)

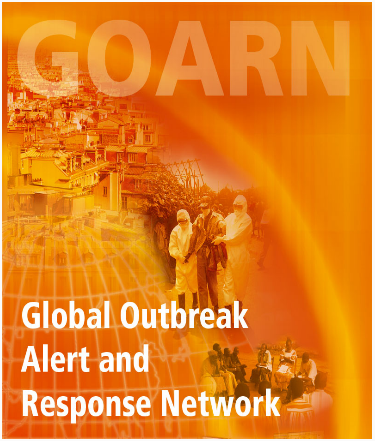
- “Active Cyber Defense Certainty Act”. Really?
- Again completely ignores the victims and causes

Please note the aggressive language:

”*I also hope it spurs a new generation of tools and methods to level the lopsided cyber battlefield*”

Rep. Tom Graves (R)

Next steps: International Cooperation



Technical expertise

On the ground where needed

Collaboration of existing institutions and networks

Constantly alert and ready to respond

World Health Organization

Next steps: Shortlist

- Care for those infected
- Study global cyber issues as systemic diseases

(And also)

- Fix bugs
- Add support for more protocols
- Match device types to infections

It is not a Cyberwar out there.

It is a Cyberpandemic.

.:: avuko
@avuko

at moment just before you sit down and try t hideous hack (Tech Model Railroad Club aning) you've been chewing over.

M - 25 Jun 2017

.:: avuko
@avuko

Dear lazyweb, is there a universal way telnet client to diff. between a success unsuccessful login, besides command execution?

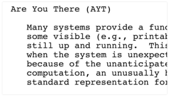
2:26 PM - 30 Jun 2017

.:: avuko
@avuko

By the gods, telnet is <insert ugly words describing something completely insane>.

AYT?

Id rather not be...

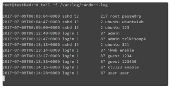


```
Are You There (AYT)

    Many systems provide a func
    some visible (e.g., printab
    still up and running.  This
    when the system is unexpect
    because of the unanticipate
    computation, an unusually h
    standard representation for
```

9:47 PM - 4 Jul 2017

.:: avuko
@avuko

Costing me too much sleep, but the first step is done.
Many more steps to before randori is ready, but all bots are welcome until then.

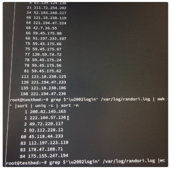


```
root@testbed:~# tail -f /var/log/randori.log

2017-07-09T00:03:04+0000 sshd 52         217 root passw0rp
2017-07-09T00:04:47+0000 sshd 13         '2 ubuntu ubuntu1ub
2017-07-09T00:09:00+0000 sshd 13         '2 ubuntu 123
2017-07-09T00:12:40+0000 login 1         87 admin
2017-07-09T00:12:47+0000 login 1         87 admin tzlkisonpk
2017-07-09T00:13:13+0000 sshd 13         '2 ubuntu 321
2017-07-09T00:13:14+0000 login 1         87 1kwb enable
2017-07-09T00:13:27+0000 login 1         87 guest 1234
2017-07-09T00:13:33+0000 login 1         87 guest 123456
2017-07-09T00:14:15+0000 login 1         87 klv123 enable
2017-07-09T00:14:28+0000 login 1         87 user user
```

2:21 AM - 9 Jul 2017

.:: avuko
@avuko

First impression, but for all the hype about IoT botnets and telnet as a vector, I'm a little disappointed at the # of sources & attempts :/

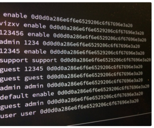


```
     21 211.72.254.202
     24 52.193.240.217
     66 121.18.238.119
     66 221.194.47.224
     66 42.7.26.55
     66 59.45.175.88
     66 91.197.232.107
     75 59.45.175.66
     75 59.45.175.67
     77 139.59.74.72
     78 59.45.175.24
     78 59.45.175.56
     81 59.45.175.62
    111 121.18.238.125
    120 221.194.47.233
    135 121.18.238.106
    198 221.194.47.236
root@testbed:~# grep $'\u2002login' /var/log/randori.log | awk
' |sort | uniq -c | sort -n
      1 200.82.145.163
      1 222.104.57.120
      2 49.72.220.117
      2 92.112.228.12
     60 45.118.44.233
     83 112.197.123.110
     83 178.47.108.71
     84 175.155.247.194
root@testbed:~# grep $'\u2002login' /var/log/randori.log |wc
```

9:26 AM - 9 Jul 2017

.:: avuko
@avuko

sting, testing, one-two. Is this thing on?"



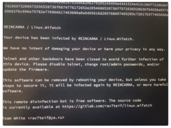
```
 enable 0d0d0a286e6f6e6529206c6f67696e3a20
vizxv enable 0d0d0a286e6f6e6529206c6f67696e3a20
123456 enable 0d0d0a286e6f6e6529206c6f67696e3a20
admin 1234 0d0d0a286e6f6e6529206c6f67696e3a20
12345 enable 0d0d0a286e6f6e6529206c6f67696e3a20
support support 0d0d0a286e6f6e6529206c6f67696e3a20
guest 12345 0d0d0a286e6f6e6529206c6f67696e3a20
guest guest 0d0d0a286e6f6e6529206c6f67696e3a20
admin admin 0d0d0a286e6f6e6529206c6f67696e3a20
default enable 0d0d0a286e6f6e6529206c6f67696e3a20
guest admin 0d0d0a286e6f6e6529206c6f67696e3a20
user user 0d0d0a286e6f6e6529206c6f67696e3a20
```

M - 14 Jul 2017

.:: avuko
@avuko

Your system attacking my system has this instead of telnetd.🙈

🎼It's like a jungle sometimes
it makes we wonder how I keep from going under.



```
742069732066726565207366747761726520546865207363757263652063646500a6973
69662f6c696e75782e776966617463680a0a5465616d205768697465203c72617637746569
```

```
REINCARNA / Linux.Wifatch

Your device has been infected by REINCARNA / Linux.Wifatch.

We have no intent of damaging your device or harm your privacy in any way.

Telnet and other backdoors have been closed to avoid further infection of
this device. Please disable telnet, change root/admin passwords, and/or
update the firmware.

This software can be removed by rebooting your device, but unless you take
steps to secure it, it will be infected again by REINCARNA, or more harmful
software.

This remote disinfection bot is free software. The source code
is currently available at https://gitlab.com/rav7teif/linux.wifatch

Team White <rav7teif@ya.ru>
```

5:09 PM - 17 Jul 2017

.:: avuko
@avuko

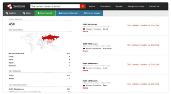
top working passwords:
43:root:oelinux123
51:root:1234
60:root:admin
162:admin:admin
387:root:root

oelinux123 = 4G WiFi. Also: IoT scanbot.

12:19 AM - 18 Jul 2017

.:: avuko
@avuko

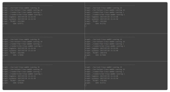
The first telnet attacks I'm monitoring come from Eastern Europe (.RS). Wondering which device says: "The connect number is limited"?



8:14 PM - 23 Jul 2017

.:: avuko
@avuko

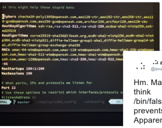
Open for business... }:)



11:28 PM - 25 Jul 2017

.:: avuko
@avuko

really have to cater to those spoiled bots... lso, how weak are IoT SSH ervices/cipherssuites etc. to be so easily wnd by lame malware?



```
## this might help those stupid bots

Ciphers chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes12
8-gcm@openssh.com,aes256-gcm@openssh.com,arcfour256,arcfour128,aes128-cbc
HostKeyAlgorithms ssh-rsa,rsa-sha2-512,rsa-sha2-256,ecdsa-sha2-nistp256,ssh-
ed25519
KexAlgorithms curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nist
p384,ecdh-sha2-nistp521,diffie-hellman-group1-sha1,diffie-hellman-group14-sh
a1,diffie-hellman-group-exchange-sha256
MACs umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@open
ssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.c
ssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hma
-96
MaxStartups 100:1:200
MaxSessions 100

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols s
#ListenAddress ::
RMAL  master  </deploy/sshd_config   sshdconfig   6%
```

3 AM - 30 Jul 2017

.:: avuko
@avuko

Hm. Many Dropbear SSH daemons out there think
/bin/false
prevents unwanted access and abuse.
Apparently forgot about port forwarding :/

10:18 PM - 30 Jul 2017

.:: avuko
@avuko

Talk accepted by @hack_lu: \o/

Nerves kicking in in 3..2..1.. 0_o

1:51 PM - 30 Aug 2017

4 Retweets 16 Likes

.:: avuko
@avuko

Hey everybody, does anyone know a free Linux VNC server which supports PAM_RHOST? TigerVNC's pam support only passes username & password.🙏

12:34 PM - 19 Sep 2017

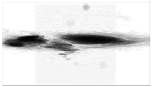
.:: avuko
@avuko

Considering a dirty, dirty, filthy hack: Likely evil user input into a shellscript as root 'ca no PAM support. This can't be 2017, right?

5:20 PM - 19 Sep 2017

.:: avuko
@avuko

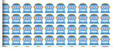
Not sure yet what to make of this.
These are bots attacking a honeypot.
Groups/links=similarity (sddeep) of ssh client & bruteforce pattern.



6:00 PM - 24 Sep 2017

.:: avuko
@avuko

Oops... My bad.
ould have seen that one coming:



.:: avuko
@avuko

Made an SQL mistake, had to redo everything for my talk😕

Upside: even stronger patterns.

Downside: details changed; have to redo examples.

8:52 AM - 3 Oct 2017