NCAE CYBER GAMES

PLAY | LEARN | PROTECT

Introduction

Keeping bad guys out is as important as getting your infrastructure up. Servers don't help when they get knocked offline. To help you track down those pesky Red Teamers, use the information below as reference for some areas you may find anomolies.
(HINT: Often, IT and SecOPs teams work together to keep services functioning and secure. If your infrastructure is in good shape, a dedicated 'analyst' might help you in the long run)

Top Tips and Tricks

Protect

```
#Check services
service --status-all
#Service information
ps -aux
#Check for start jobs
ls /etc/init/*.conf
#Backup existing rules
iptables-save > iptables_rules.out
#Modify export if needed
vi iptables_rules.out
#Restore export
iptables-restore < firewall.out
#List rules
iptables -L
#Change password
passwd
```

Detect

```
#*NETWORK*
#Look at live network traffic
tcpdump
#Save PCAP to remote host (Kali?)
tcpdump -w - | ssh <remote ip address here> -p <port> "cat - > /tmp/<filename>.pcap"
#Monitor for new tcp connections (netstat has to be installed, yum install net-tools)
netstat -ac 5 | grep tcp
#Monitor traffic remotely (from kali?)
ssh <user>@<remote ip of host to monitor> tcpdump -i any -U -s -) -w - 'not host <kali ip>'
#*LOGS* (look below for common linux logs)
#Look at log
tail /path/to/log
#Look at log in real time
tail -f /path/to/log
#Look for keyword in log
grep -i "<keyword>" /path/to/log
#Look for sudo activity
grep -i sudo /var/log/auth.log
```

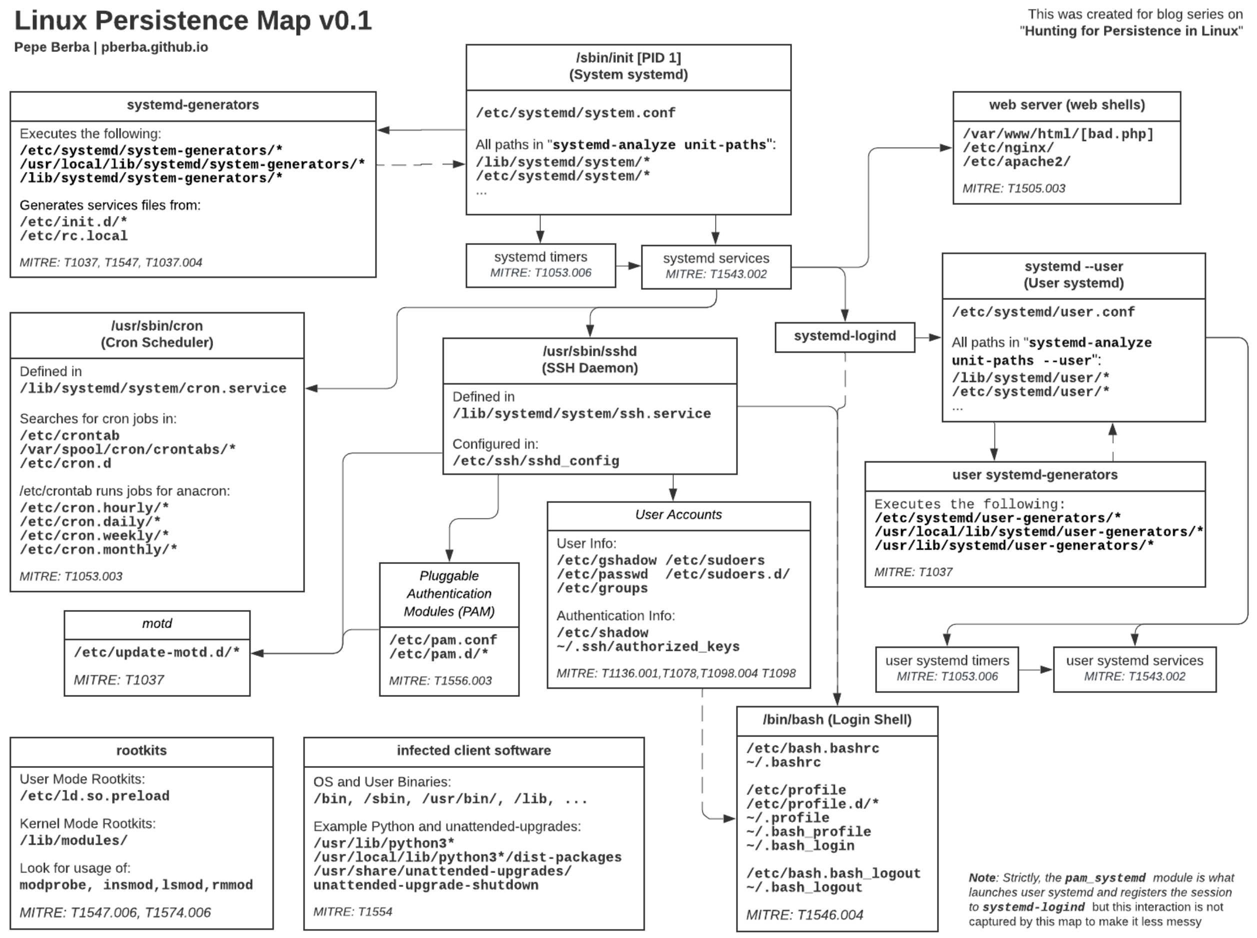
Triage

```
#View logged in users
w
#Check remote login activity
lastlog
#Check failed logins
faillog -a
#View local accounts and groups
cat /etc/passwd
cat /etc/shadow
cat /etc/group
cat /etc/sudoers
#Show root accounts
awk -F: '($3 == "0") {print}' /etc/passwd
#Active network connections
netstat -antup
#View routes
route
#List processes listening on ports
lsof -i
#Check cron
```

```
crontab -l
cat /etc/crontab
ls /etc/cron.*
#Stop a process (hint, use a command above for checking process info)
kill <process pid>
kill -9 -I <process name>
#Remove execution from a process (or just delete it)
chmod -x /path/to/malicious/file
#Move the malicious file to analyze it for potential attacker information
mv /path/to/malicious/file ~/quarantine
strings ~/quarantine/<malicious_file>
```

Persistence Hunting

There is A LOT of information here. Instead of trying to go through the whole thing, take a look and identify some files that may be useful for looking for bad guys. This is good reference material.



Linux Persistence Map v0.1
Pepe Berba | pberba.github.io

This was created for blog series on
"**Hunting for Persistence in Linux**"

/sbin/init [PID 1]
(System systemd)
/etc/systemd/system.conf
All paths in "**systemd-analyze unit-paths**":
/lib/systemd/system/*
/etc/systemd/system/*
...

systemd-generators
Executes the following:
/etc/systemd/system-generators/*
/usr/local/lib/systemd/system-generators/*
/lib/systemd/system-generators/*
Generates services files from:
/etc/init.d/*
/etc/rc.local
MITRE: T1037, T1547, T1037.004

web server (web shells)
/var/www/html/[bad.php]
/etc/nginx/
/etc/apache2/
MITRE: T1505.003

systemd timers
MITRE: T1053.006

systemd services
MITRE: T1543.002

systemd-logind

systemd --user
(User systemd)
/etc/systemd/user.conf
All paths in "**systemd-analyze unit-paths --user**":
/lib/systemd/user/*
/etc/systemd/user/*
...

/usr/sbin/cron
(Cron Scheduler)
Defined in
/lib/systemd/system/cron.service
Searches for cron jobs in:
/etc/crontab
/var/spool/cron/crontabs/*
/etc/cron.d
/etc/crontab runs jobs for anacron:
/etc/cron.hourly/*
/etc/cron.daily/*
/etc/cron.weekly/*
/etc/cron.monthly/*
MITRE: T1053.003

/usr/sbin/sshd
(SSH Daemon)
Defined in
/lib/systemd/system/ssh.service
Configured in:
/etc/ssh/sshd_config

user systemd-generators
Executes the following:
/etc/systemd/user-generators/*
/usr/local/lib/systemd/user-generators/*
/usr/lib/systemd/user-generators/*
MITRE: T1037

User Accounts
User Info:
/etc/gshadow /etc/sudoers
/etc/passwd /etc/sudoers.d/
/etc/groups
Authentication Info:
/etc/shadow
~/.ssh/authorized_keys
MITRE: T1136.001,T1078,T1098.004 T1098

Pluggable Authentication Modules (PAM)
/etc/pam.conf
/etc/pam.d/*
MITRE: T1556.003

motd
/etc/update-motd.d/*
MITRE: T1037

user systemd timers
MITRE: T1053.006

user systemd services
MITRE: T1543.002

rootkits
User Mode Rootkits:
/etc/ld.so.preload
Kernel Mode Rootkits:
/lib/modules/
Look for usage of:
modprobe, insmod,lsmod,rmmod
MITRE: T1547.006, T1574.006

infected client software
OS and User Binaries:
/bin, /sbin, /usr/bin/, /lib, ...
Example Python and unattended-upgrades:
/usr/lib/python3*
/usr/local/lib/python3*/dist-packages
/usr/share/unattended-upgrades/
unattended-upgrade-shutdown
MITRE: T1554

/bin/bash (Login Shell)
/etc/bash.bashrc
~/.bashrc
/etc/profile
/etc/profile.d/*
~/.profile
~/.bash_profile
~/.bash_login
/etc/bash.bash_logout
~/.bash_logout
MITRE: T1546.004

***Note**: Strictly, the **pam_systemd** module is what launches user systemd and registers the session to **systemd-logind** but this interaction is not captured by this map to make it less messy*