Survivalism: Systematic Analysis of Windows Malware Living-Off-The-Land

Frederick Barr-Smith
Oxford University

Xabier Ugarte-Pedrero
Cisco Systems

Mariano Graziano
Cisco Systems

Riccardo Spolaor
Oxford University

Ivan Martinovic
Oxford University

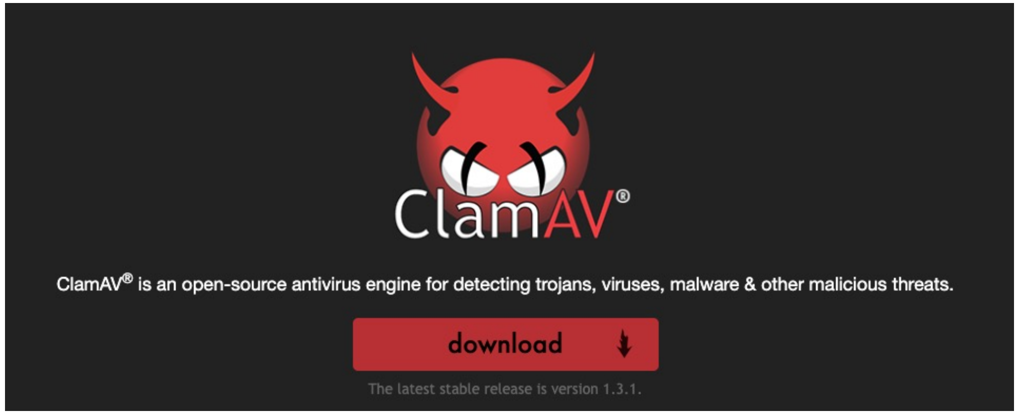
IEEE Symposium on Security and Privacy (SP).

Outline

- INTRODUCTION
- BACKGROUND & RELATED WORK
- MOTIVATION: ANTI-VIRUS PRODUCTS VS. LIVING-OFF-THE-LAND TECHNIQUES
- MEASURING LOTL PREVALENCE
- MAIN TAKEAWAYS AND DISCUSSION
- LIMITATIONS & FUTURE WORK

INTRODUCTION

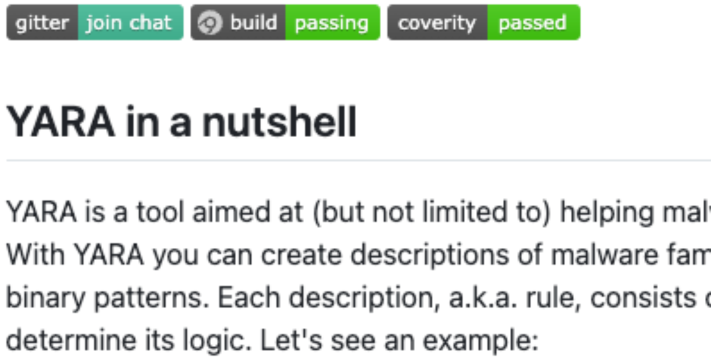
- Malware development and detection is a cat and mouse game.
- Anti-virus (AV) products implement static and heuristic analysis technologies to detect, classify and prevent malware.
  - 1 static: mainly signature-based detection
  - 2 heuristic analysis: analyze the behavioral characteristics and code patterns of the program to infer whether the file is malicious



ClamAV® is an open-source antivirus engine for detecting trojans, viruses, malware & other malicious threats.

The latest stable release is version 1.3.1.

https://github.com/VirusTotal/yara



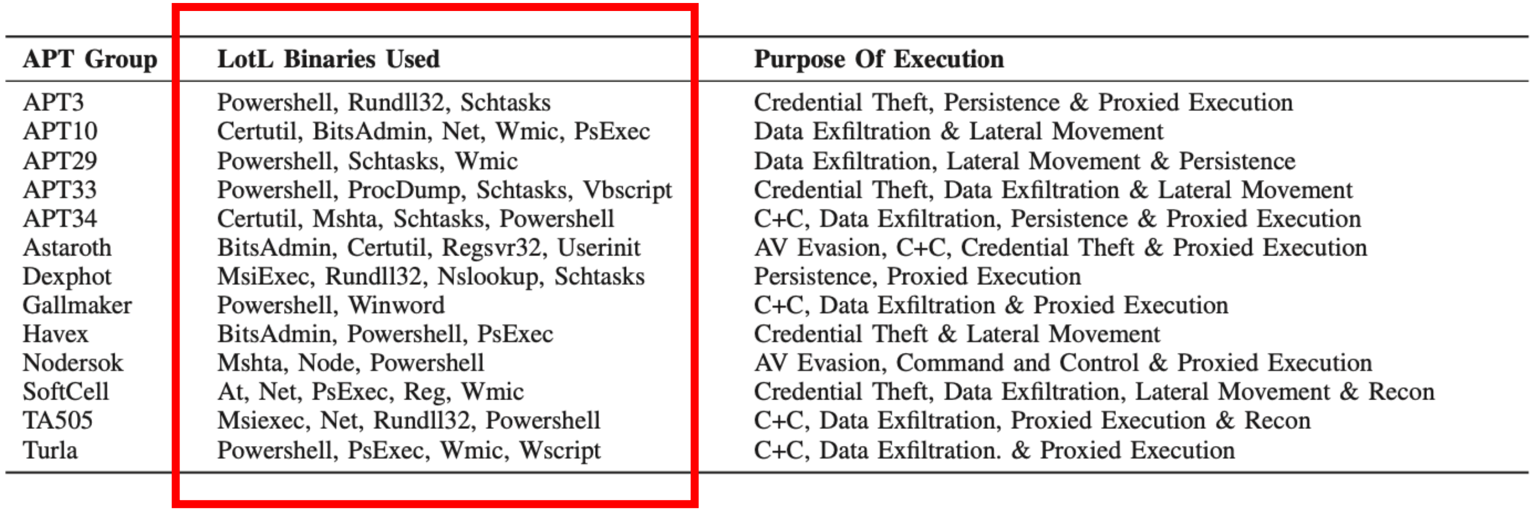
YARA in a nutshell

YARA is a tool aimed at (but not limited to) helping mal
With YARA you can create descriptions of malware fam
binary patterns. Each description, a.k.a. rule, consists
determine its logic. Let's see an example:

INTRODUCTION

- VT analyses suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community.

Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community.



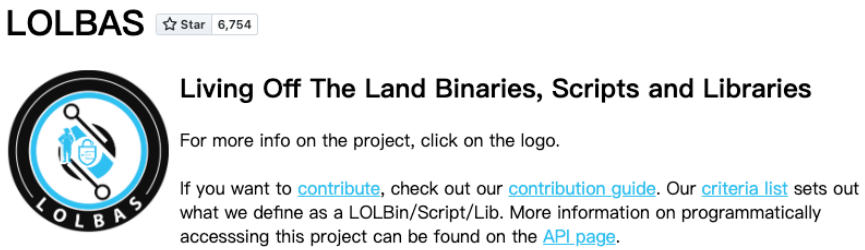
APT Group	LotL Binaries Used	Purpose Of Execution
APT3	Powershell, Rundll32, Schtasks	Credential Theft, Persistence & Proxied Execution
APT10	Certutil, BitsAdmin, Net, Wmic, PsExec	Data Exfiltration & Lateral Movement
APT29	Powershell, Schtasks, Wmic	Data Exfiltration, Lateral Movement & Persistence
APT33	Powershell, ProcDump, Schtasks, Vbscript	Credential Theft, Data Exfiltration & Lateral Movement
APT34	Certutil, Mshta, Schtasks, Powershell	C+C, Data Exfiltration, Persistence & Proxied Execution
Astaroth	BitsAdmin, Certutil, Regsvr32, Userinit	AV Evasion, C+C, Credential Theft & Proxied Execution
Dexphot	MsiExec, Rundll32, Nslookup, Schtasks	Persistence, Proxied Execution
Gallmaker	Powershell, Winword	C+C, Data Exfiltration & Proxied Execution
Havex	BitsAdmin, Powershell, PsExec	Credential Theft & Lateral Movement
Nodersok	Mshta, Node, Powershell	AV Evasion, Command and Control & Proxied Execution
SoftCell	At, Net, PsExec, Reg, Wmic	Credential Theft, Data Exfiltration, Lateral Movement & Recon
TA505	Msiexec, Net, Rundll32, Powershell	C+C, Data Exfiltration, Proxied Execution & Recon
Turla	Powershell, PsExec, Wmic, Wscript	C+C, Data Exfiltration. & Proxied Execution

- LotL techniques refer to the use of binaries that are already present on systems or are easy to install (e.g., signed, legitimate tools) to conduct post-exploitation activity.

- AddinUtil.exe(a tool used to update Microsoft Office Add-Ins) can be used to execute malicious payload

INTRODUCTION

- It is hard to find a precise definition for the *Living off the Land* technique

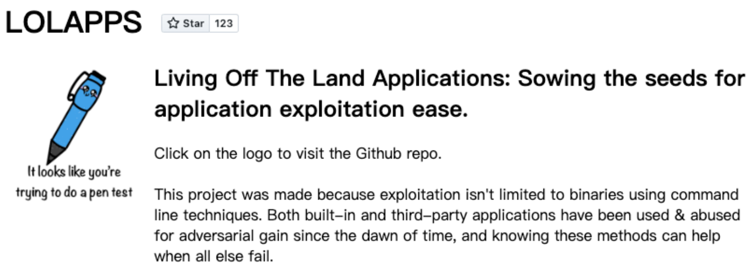


LOLBAS

Living Off The Land Binaries, Scripts and Libraries

For more info on the project, click on the logo.

If you want to contribute, check out our contribution guide. Our criteria list sets out what we define as a LOLBin/Script/Lib. More information on programmatically accesssing this project can be found on the API page.



LOLAPPS

Living Off The Land Applications: Sowing the seeds for application exploitation ease.

Click on the logo to visit the Github repo.

This project was made because exploitation isn't limited to binaries using command line techniques. Both built-in and third-party applications have been used & abused for adversarial gain since the dawn of time, and knowing these methods can help when all else fail.

- RQ1: Can LotL techniques effectively evade commercial AV?

- RQ2: How prevalent is the use of LotL binaries in malware?
  - ➢ What purposes do malware binaries use LotL techniques for?
  - ➢ Which malware families and types use LotL binaries most prolifically and how does their usage differ?

- RQ3: What are the overlaps and differences in the behavior of legitimate and malicious binaries with respect to the usage of LotL binaries? How would this affect detection by heuristic AV engines?

BACKGROUND & RELATED WORK

- Due to its novelty, there is significant confusion regarding the term Living-Off-The-Land binary(LOLbin).
- *Define a LOLbin as any binary with a recognized legitimate use, that is leveraged during an attack to directly perform a malicious action; or to assist indirectly, in a sequence of actions that have a final malicious outcome.*
- Examples: *Reg.exe*, *Sc.exe* and *Wmic.exe*
- Most binaries installed by default are signed by *Microsoft Authenticode*
- External signed binaries: *PsExec.exe* or other SysInternals binaries.

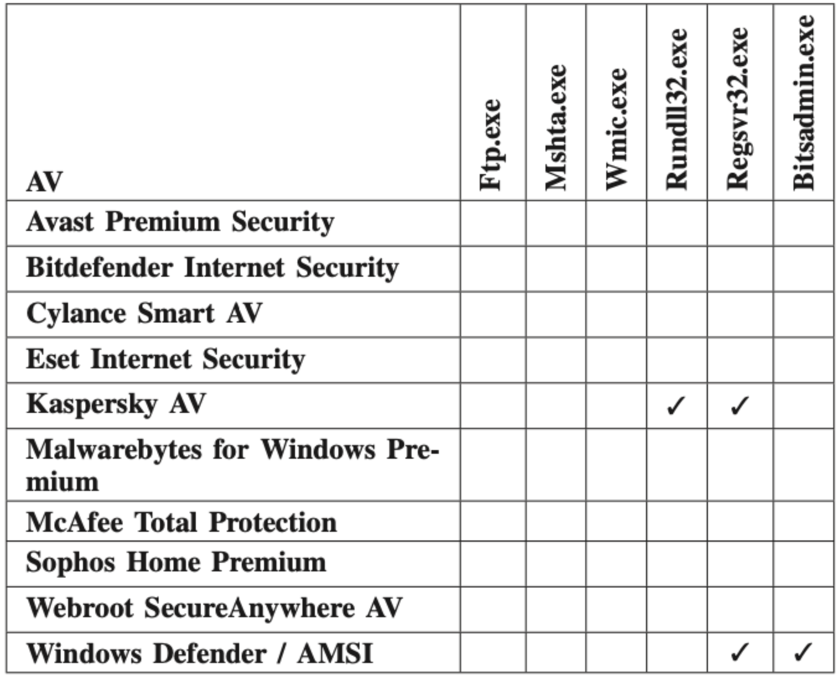
Learn / Sysinternals /

PsExec v2.43

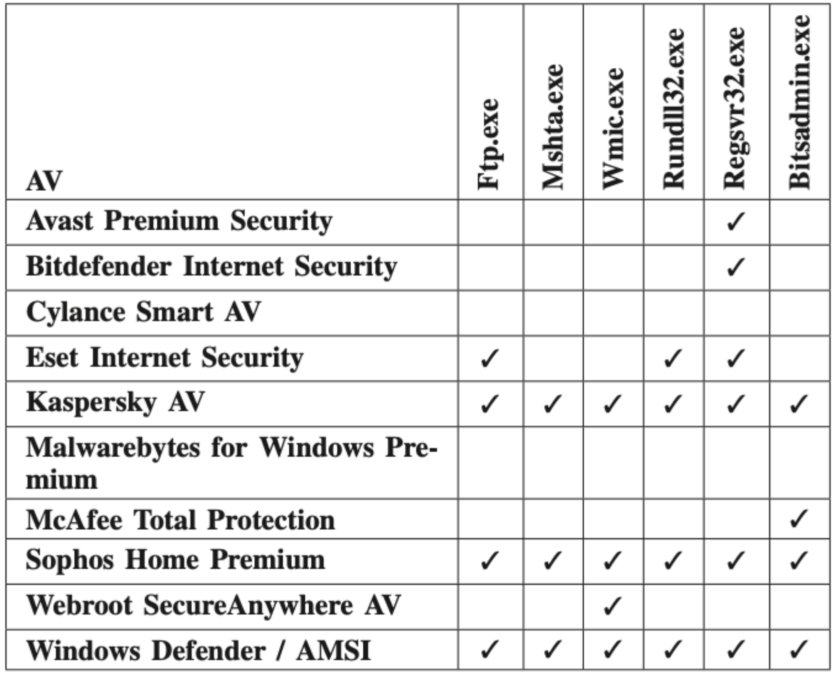
發行項 • 2023/10/12 • 7 位參與者

MOTIVATION: ANTI-VIRUS PRODUCTS VS. LIVING-OFF-THE-LAND TECHNIQUES

- RQ1: Can LotL techniques effectively evade commercial AV?
- Leveraged a reverse shell to assess how vulnerable AV systems(top 10 popular) are to evasion by malware deploying LotL techniques.



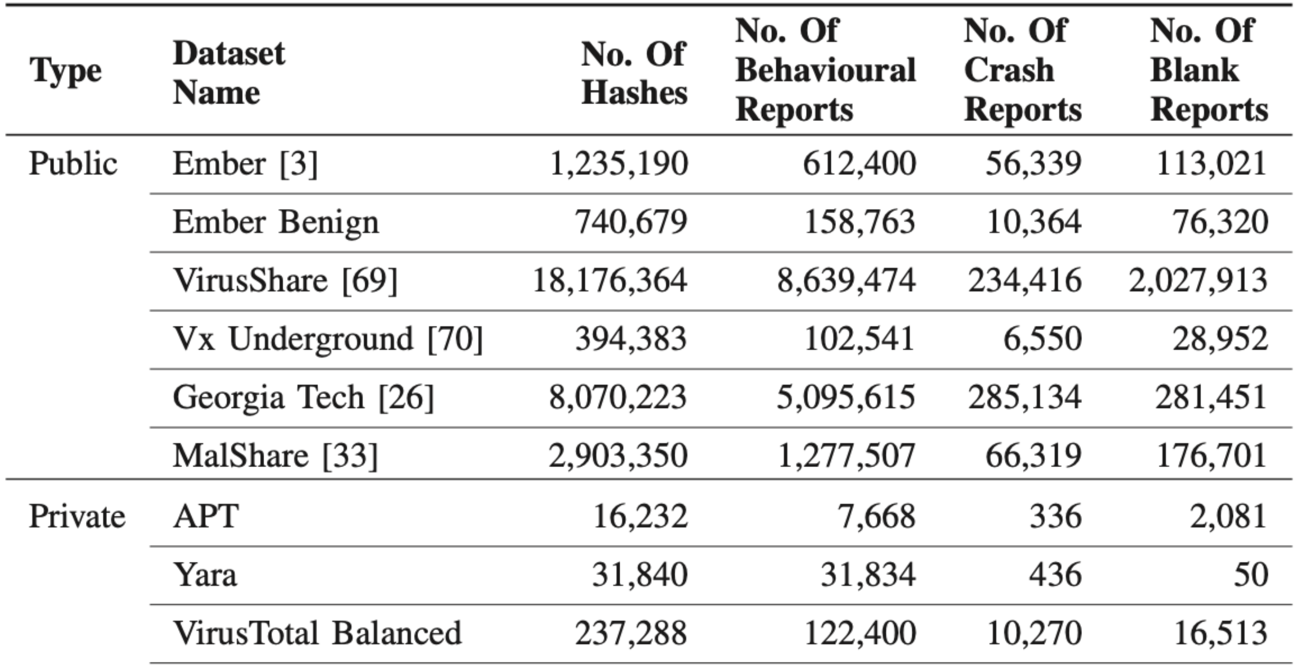
AV	Ftp.exe	Mshta.exe	Wmic.exe	Rundll32.exe	Regsvr32.exe	Bitsadmin.exe
Avast Premium Security						
Bitdefender Internet Security						
Cylance Smart AV						
Eset Internet Security						
Kaspersky AV				✓	✓	
Malwarebytes for Windows Premium						
McAfee Total Protection						
Sophos Home Premium						
Webroot SecureAnywhere AV						
Windows Defender / AMSI					✓	✓



AV	Ftp.exe	Mshta.exe	Wmic.exe	Rundll32.exe	Regsvr32.exe	Bitsadmin.exe
Avast Premium Security					✓	
Bitdefender Internet Security					✓	
Cylance Smart AV						
Eset Internet Security	✓			✓	✓	
Kaspersky AV	✓	✓	✓	✓	✓	✓
Malwarebytes for Windows Premium						
McAfee Total Protection						✓
Sophos Home Premium	✓	✓	✓	✓	✓	✓
Webroot SecureAnywhere AV			✓			
Windows Defender / AMSI	✓	✓	✓	✓	✓	✓

MEASURING LOTL PREVALENCE

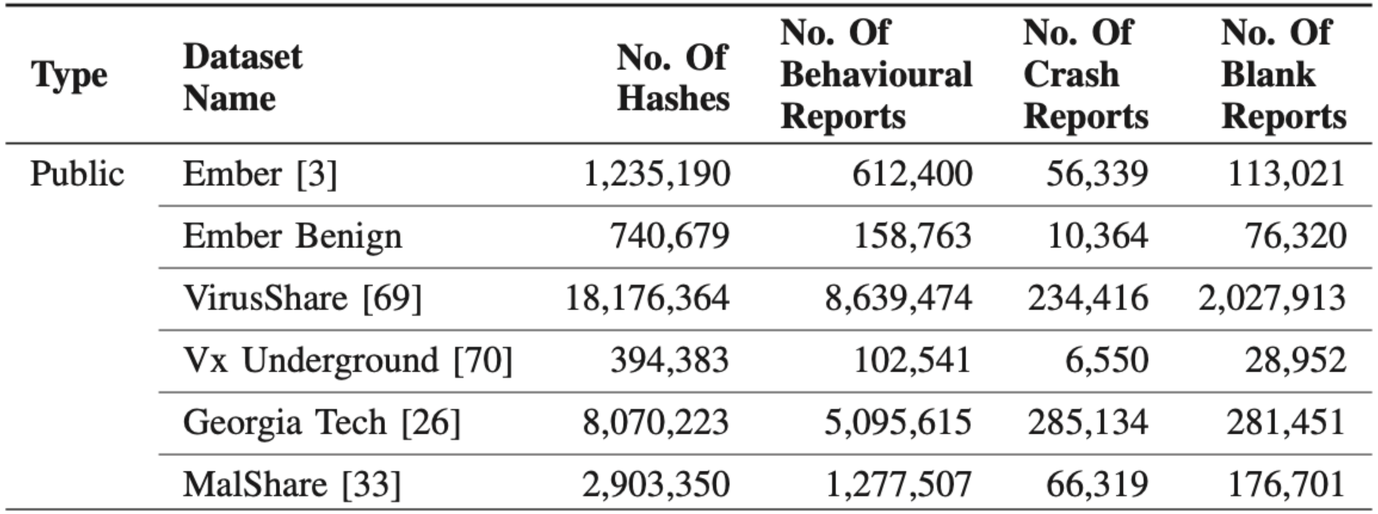
- RQ2: How prevalent is the use of LotL binaries in malware?
  - ➢ What purposes do malware binaries use LotL techniques for?
  - ➢ Which malware families and types use LotL binaries most prolifically and how does their usage differ?
- Dataset Composition - (collected 31,805,549 samples)
  - 1 Public Datasets (6)
  - 2 Private Datasets (3)



Type	Dataset Name	No. Of Hashes	No. Of Behavioural Reports	No. Of Crash Reports	No. Of Blank Reports
Public	Ember [3]	1,235,190	612,400	56,339	113,021
	Ember Benign	740,679	158,763	10,364	76,320
	VirusShare [69]	18,176,364	8,639,474	234,416	2,027,913
	Vx Underground [70]	394,383	102,541	6,550	28,952
	Georgia Tech [26]	8,070,223	5,095,615	285,134	281,451
	MalShare [33]	2,903,350	1,277,507	66,319	176,701
Private	APT	16,232	7,668	336	2,081
	Yara	31,840	31,834	436	50
	VirusTotal Balanced	237,288	122,400	10,270	16,513

MEASURING LOTL PREVALENCE

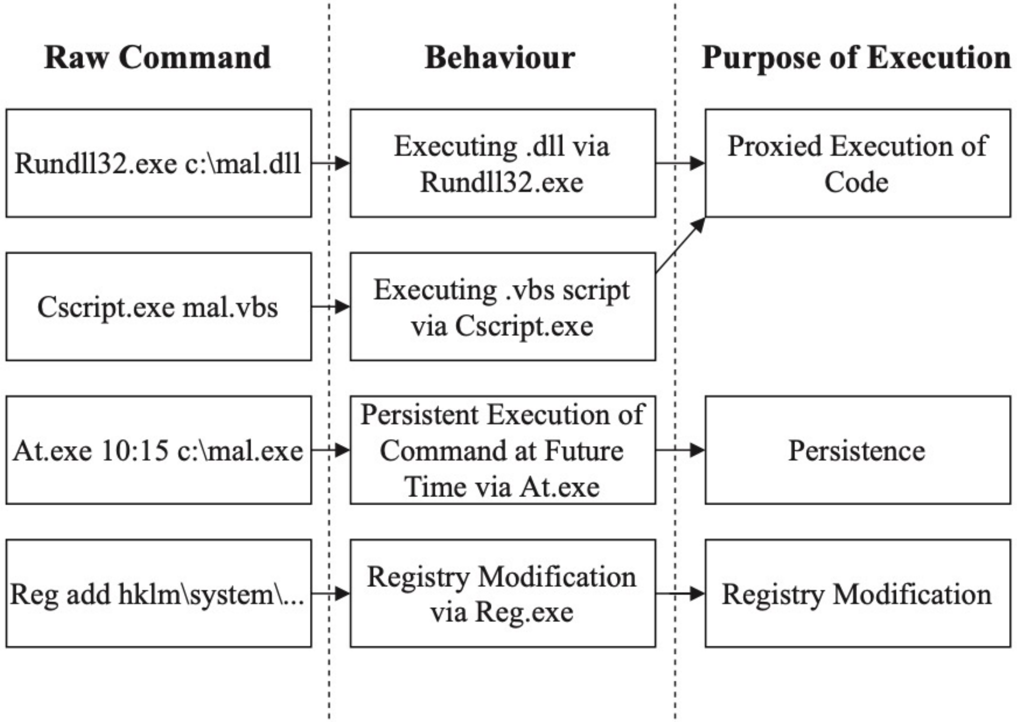
- RQ2: How prevalent is the use of LotL binaries in malware?
  - ➢ What purposes do malware binaries use LotL techniques for?
  - ➢ Which malware families and types use LotL binaries most prolifically and how does their usage differ?
- Analysis Pipeline: VT analysis + data augmentation
- Pattern matching: processed all the collected behavior reports (shell-cmd & process)



Type	Dataset Name	No. Of Hashes	No. Of Behavioural Reports	No. Of Crash Reports	No. Of Blank Reports
Public	Ember [3]	1,235,190	612,400	56,339	113,021
	Ember Benign	740,679	158,763	10,364	76,320
	VirusShare [69]	18,176,364	8,639,474	234,416	2,027,913
	Vx Underground [70]	394,383	102,541	6,550	28,952
	Georgia Tech [26]	8,070,223	5,095,615	285,134	281,451
	MalShare [33]	2,903,350	1,277,507	66,319	176,701

MEASURING LOTL PREVALENCE

- RQ2: How prevalent is the use of LotL binaries in malware?
  - ➢ What purposes do malware binaries use LotL techniques for?
  - ➢ Which malware families and types use LotL binaries most prolifically and how does their usage differ?



Raw Command | Behaviour | Purpose of Execution

Rundll32.exe c:\mal.dll → Executing .dll via Rundll32.exe → Proxied Execution of Code

Cscript.exe mal.vbs → Executing .vbs script via Cscript.exe → Proxied Execution of Code

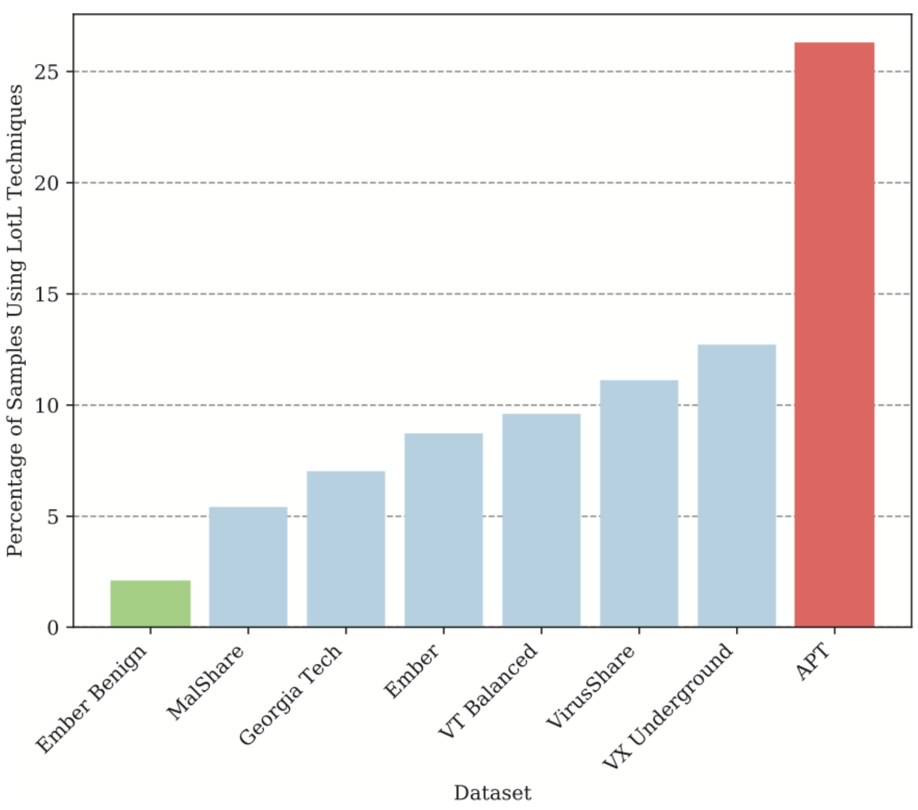
At.exe 10:15 c:\mal.exe → Persistent Execution of Command at Future Time via At.exe → Persistence

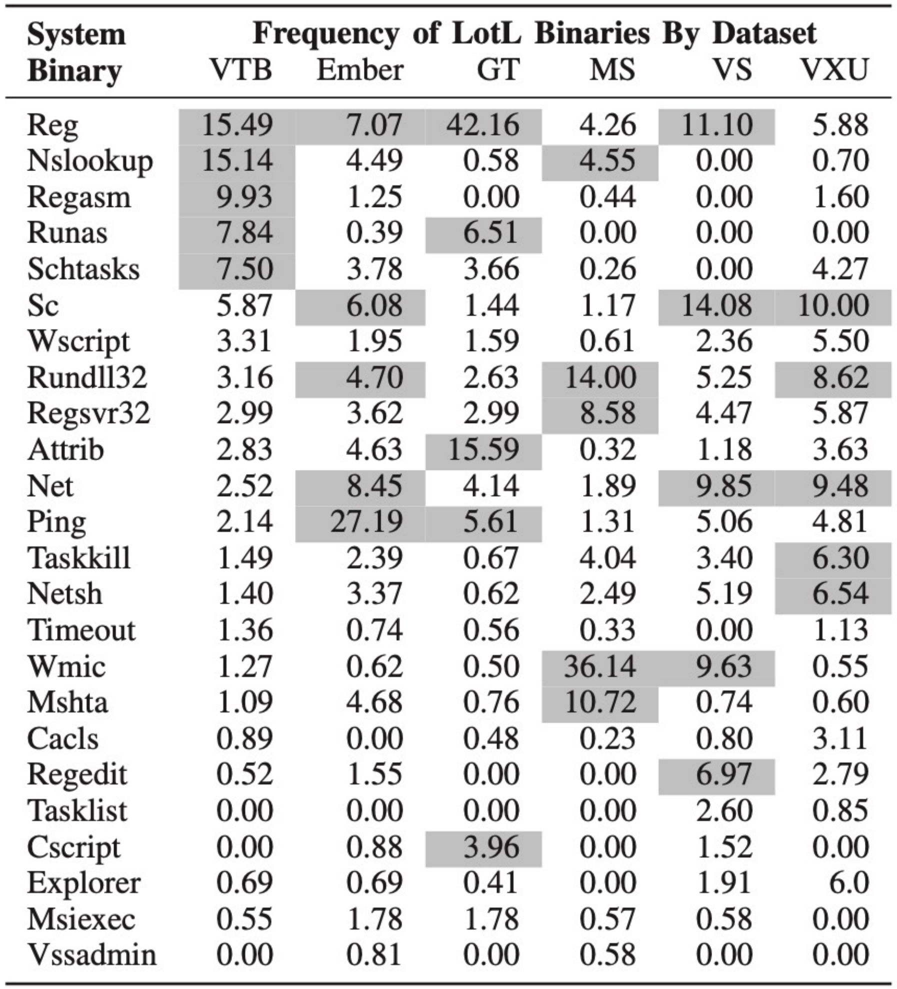
Reg add hklm\system\... → Registry Modification via Reg.exe → Registry Modification

- Execution
  - ➢ Proxied Execution
  - ➢ Persistence
  - ➢ Delayed Execution
- Modification of system components.
  - ➢ Firewall Modification
  - ➢ Registry Modification
  - ➢ Permissions Modification
- Else
  - ➢ File Opening
  - ➢ Reconnaissance
  - ➢ Task stopping

MEASUREMENT RESULTS

- RQ2: How prevalent is the use of LotL binaries in malware?

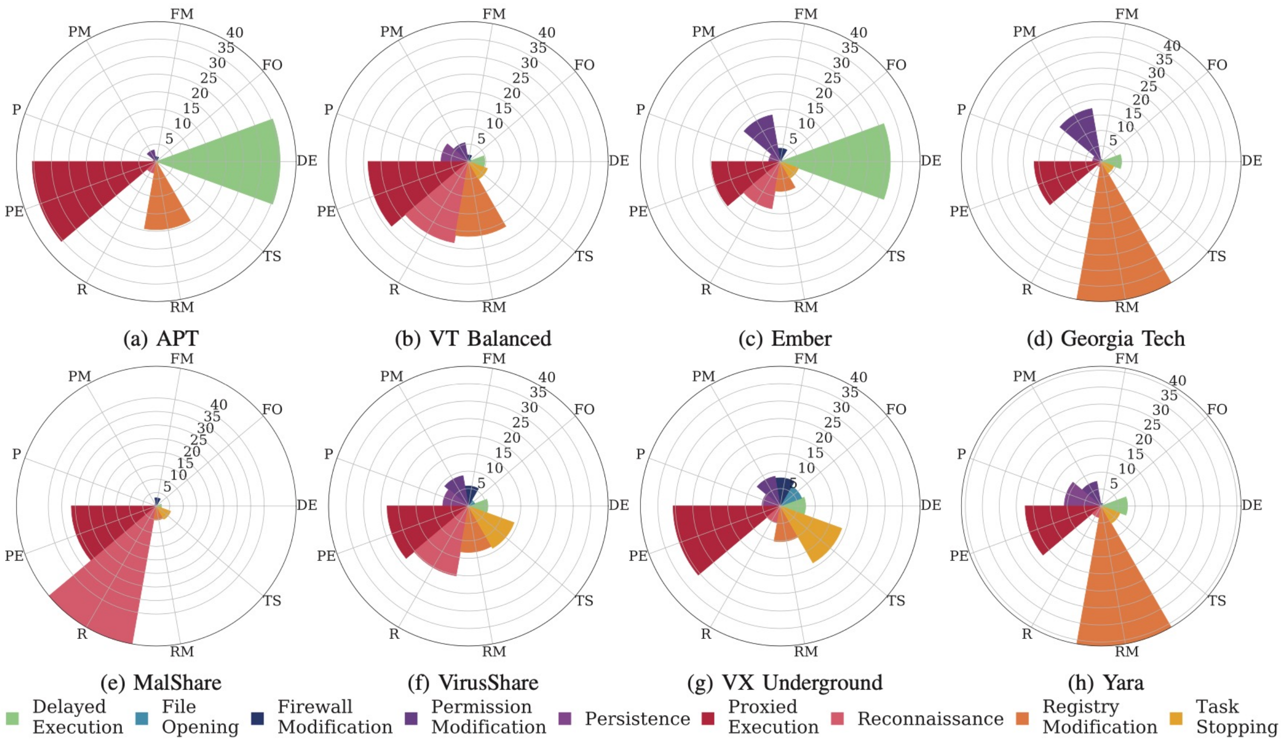




System Binary	Frequency of LotL Binaries By Dataset VTB	Ember	GT	MS	VS	VXU
Reg	15.49	7.07	42.16	4.26	11.10	5.88
Nslookup	15.14	4.49	0.58	4.55	0.00	0.70
Regasm	9.93	1.25	0.00	0.44	0.00	1.60
Runas	7.84	0.39	6.51	0.00	0.00	0.00
Schtasks	7.50	3.78	3.66	0.26	0.00	4.27
Sc	5.87	6.08	1.44	1.17	14.08	10.00
Wscript	3.31	1.95	1.59	0.61	2.36	5.50
Rundll32	3.16	4.70	2.63	14.00	5.25	8.62
Regsvr32	2.99	3.62	2.99	8.58	4.47	5.87
Attrib	2.83	4.63	15.59	0.32	1.18	3.63
Net	2.52	8.45	4.14	1.89	9.85	9.48
Ping	2.14	27.19	5.61	1.31	5.06	4.81
Taskkill	1.49	2.39	0.67	4.04	3.40	6.30
Netsh	1.40	3.37	0.62	2.49	5.19	6.54
Timeout	1.36	0.74	0.56	0.33	0.00	1.13
Wmic	1.27	0.62	0.50	36.14	9.63	0.55
Mshta	1.09	4.68	0.76	10.72	0.74	0.60
Cacls	0.89	0.00	0.48	0.23	0.80	3.11
Regedit	0.52	1.55	0.00	0.00	6.97	2.79
Tasklist	0.00	0.00	0.00	0.00	2.60	0.85
Cscript	0.00	0.88	3.96	0.00	1.52	0.00
Explorer	0.69	0.69	0.41	0.00	1.91	6.0
Msiexec	0.55	1.78	1.78	0.57	0.58	0.00
Vssadmin	0.00	0.81	0.00	0.58	0.00	0.00

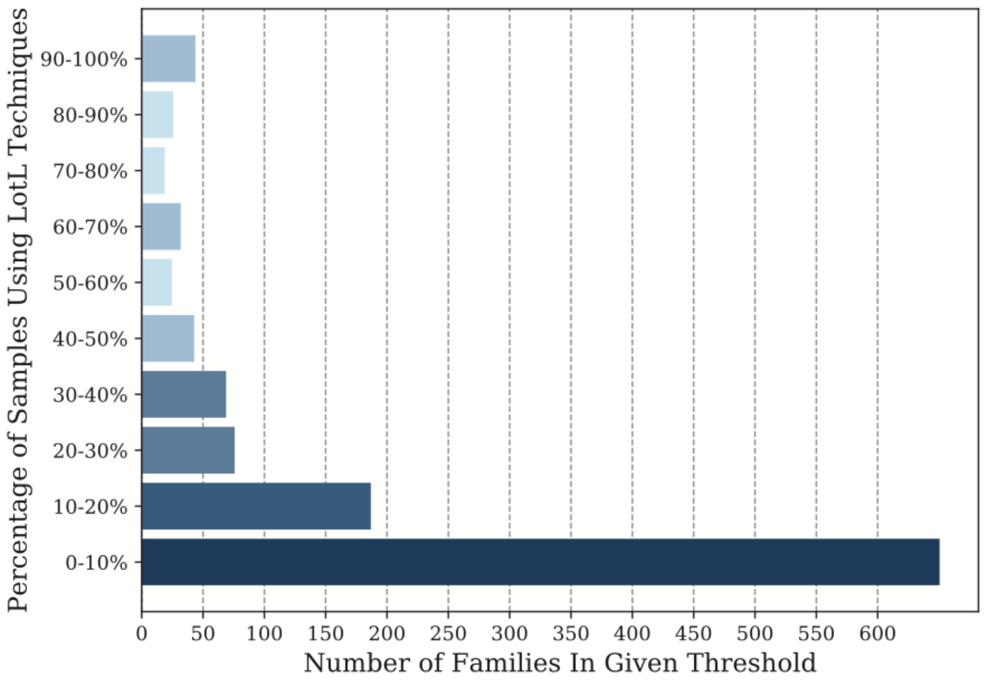
MEASUREMENT RESULTS

- RQ2: How prevalent is the use of LotL binaries in malware?



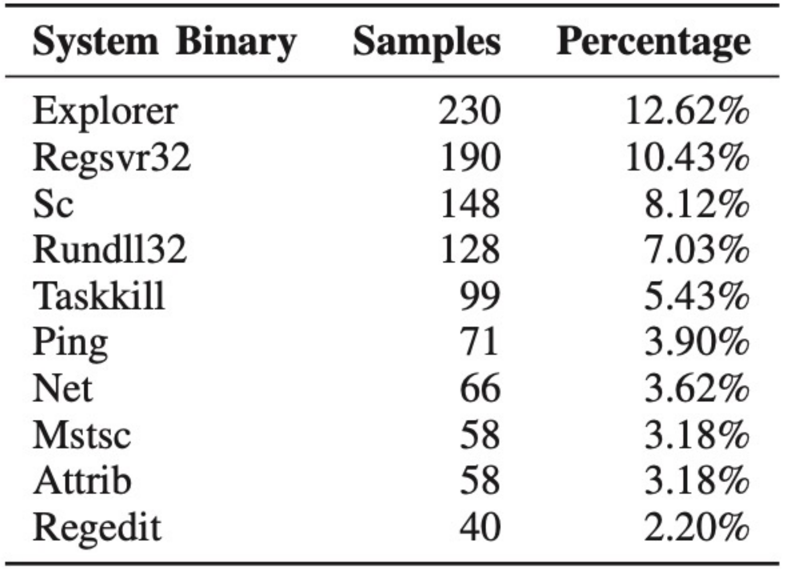
(a) APT (b) VT Balanced (c) Ember (d) Georgia Tech

(e) MalShare (f) VirusShare (g) VX Underground (h) Yara

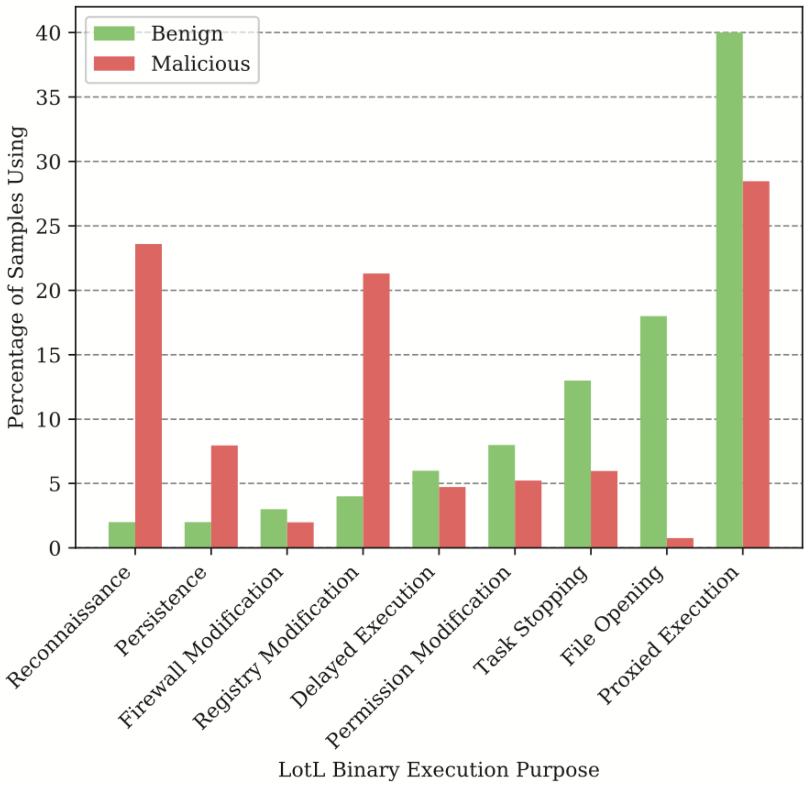


MEASUREMENT RESULTS

- RQ3: What are the overlaps and differences in the behavior of legitimate and malicious binaries with respect to the usage of LotL binaries? How would this affect detection by heuristic AV engines?

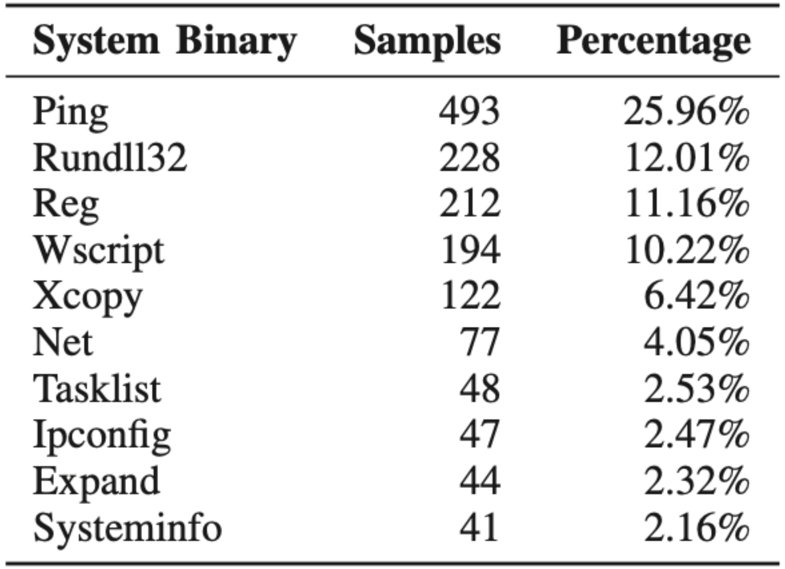


System Binary	Samples	Percentage
Explorer	230	12.62%
Regsvr32	190	10.43%
Sc	148	8.12%
Rundll32	128	7.03%
Taskkill	99	5.43%
Ping	71	3.90%
Net	66	3.62%
Mstsc	58	3.18%
Attrib	58	3.18%
Regedit	40	2.20%

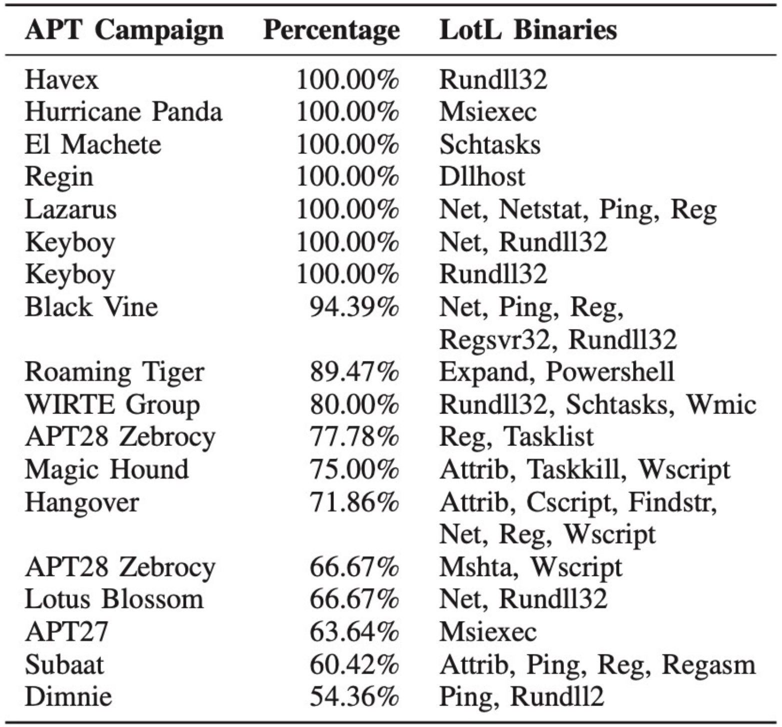


MEASUREMENT RESULTS

- RQ3: What are the overlaps and differences in the behavior of legitimate and malicious binaries with respect to the usage of LotL binaries? How would this affect detection by heuristic AV engines?



System Binary	Samples	Percentage
Ping	493	25.96%
Rundll32	228	12.01%
Reg	212	11.16%
Wscript	194	10.22%
Xcopy	122	6.42%
Net	77	4.05%
Tasklist	48	2.53%
Ipconfig	47	2.47%
Expand	44	2.32%
Systeminfo	41	2.16%



APT Campaign	Percentage	LotL Binaries
Havex	100.00%	Rundll32
Hurricane Panda	100.00%	Msiexec
El Machete	100.00%	Schtasks
Regin	100.00%	Dllhost
Lazarus	100.00%	Net, Netstat, Ping, Reg
Keyboy	100.00%	Net, Rundll32
Keyboy	100.00%	Rundll32
Black Vine	94.39%	Net, Ping, Reg, Regsvr32, Rundll32
Roaming Tiger	89.47%	Expand, Powershell
WIRTE Group	80.00%	Rundll32, Schtasks, Wmic
APT28 Zebrocy	77.78%	Reg, Tasklist
Magic Hound	75.00%	Attrib, Taskkill, Wscript
Hangover	71.86%	Attrib, Cscript, Findstr, Net, Reg, Wscript
APT28 Zebrocy	66.67%	Mshta, Wscript
Lotus Blossom	66.67%	Net, Rundll32
APT27	63.64%	Msiexec
Subaat	60.42%	Attrib, Ping, Reg, Regasm
Dimnie	54.36%	Ping, Rundll2

MAIN TAKEAWAYS AND DISCUSSION

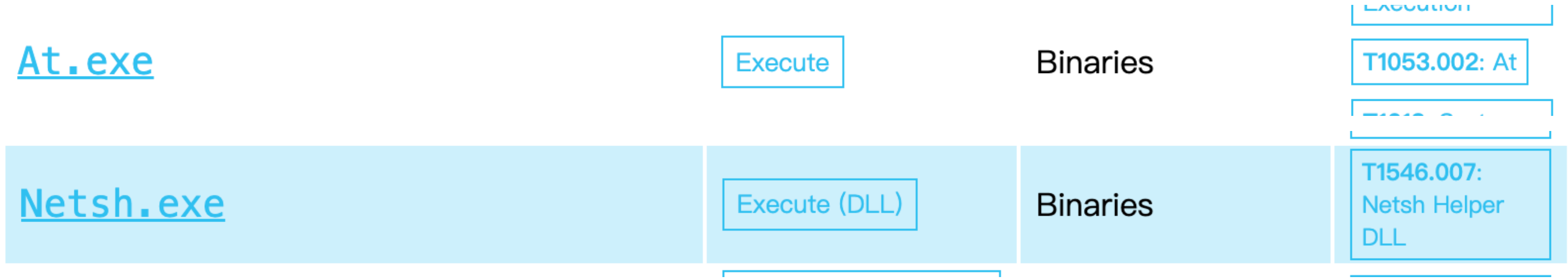
1. Almost every popular AV product had difficulties detecting malicious usage of LotL binaries.

2. There are differences in execution purposes between benign and malicious samples, providing a vector for development of detection algorithms.

3. With regards to execution purpose, we observe that LotL binaries were not only leveraged for proxied execution or evasion, but also to implement common malicious routines

4. There was a large variability of prevalence of LotL techniques among different families.

5. Legimate software used LotL binaries less than malware.



At.exe	Execute	Binaries	T1053.002: At
Netsh.exe	Execute (DLL)	Binaries	T1546.007: Netsh Helper DLL

LIMITATIONS & FUTURE WORK

- Intended or Unexpected Functionality.
  - not include a comparison of intent in our measurement results.

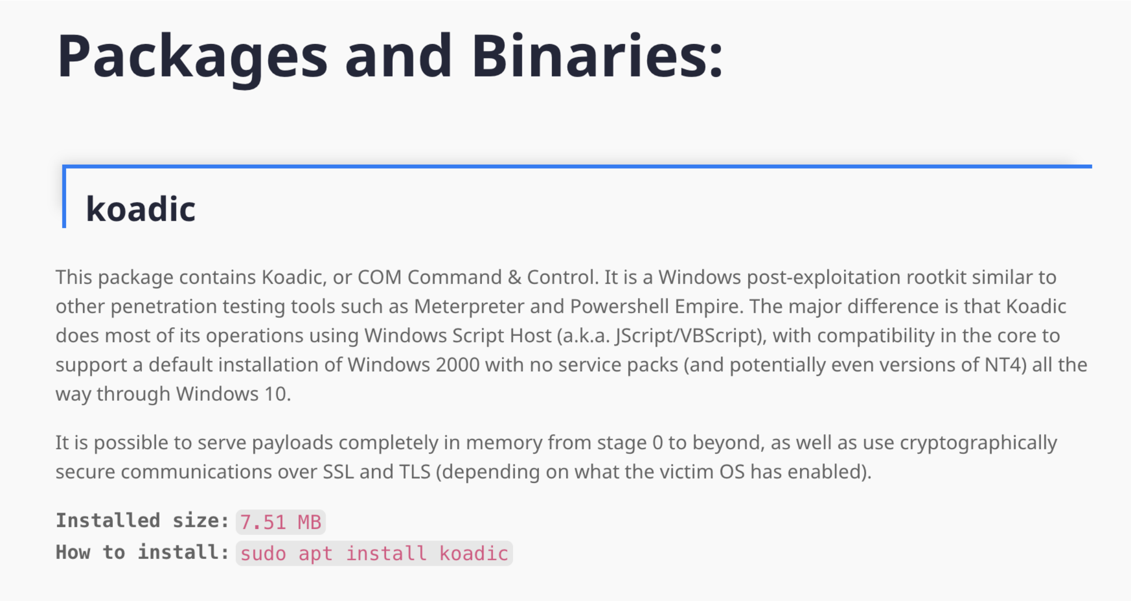
- Anti-VM Malware

- Human Operators

- Linux LotL

LotL binary based Bypass Techniques

- Ftp.exe
- Mshta.exe
- Wmic.exe
- Rundll32.exe
- Regsvr32.exe
- Bitsadmin.exe



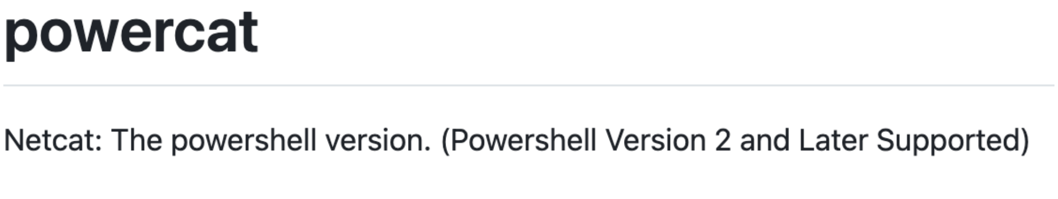
Packages and Binaries:

koadic

This package contains Koadic, or COM Command & Control. It is a Windows post-exploitation rootkit similar to other penetration testing tools such as Meterpreter and Powershell Empire. The major difference is that Koadic does most of its operations using Windows Script Host (a.k.a. JScript/VBScript), with compatibility in the core to support a default installation of Windows 2000 with no service packs (and potentially even versions of NT4) all the way through Windows 10.

It is possible to serve payloads completely in memory from stage 0 to beyond, as well as use cryptographically secure communications over SSL and TLS (depending on what the victim OS has enabled).

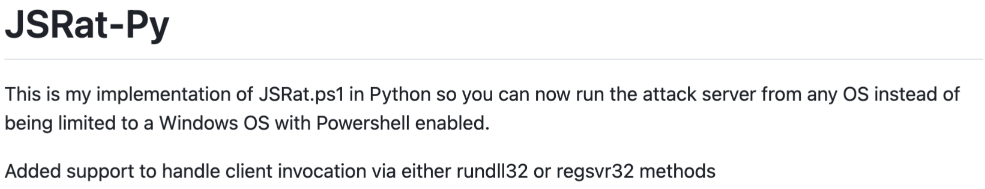
Installed size: `7.51 MB`
How to install: `sudo apt install koadic`



powercat

Netcat: The powershell version. (Powershell Version 2 and Later Supported)

Installation



JSRat-Py

This is my implementation of JSRat.ps1 in Python so you can now run the attack server from any OS instead of being limited to a Windows OS with Powershell enabled.

Added support to handle client invocation via either rundll32 or regsvr32 methods