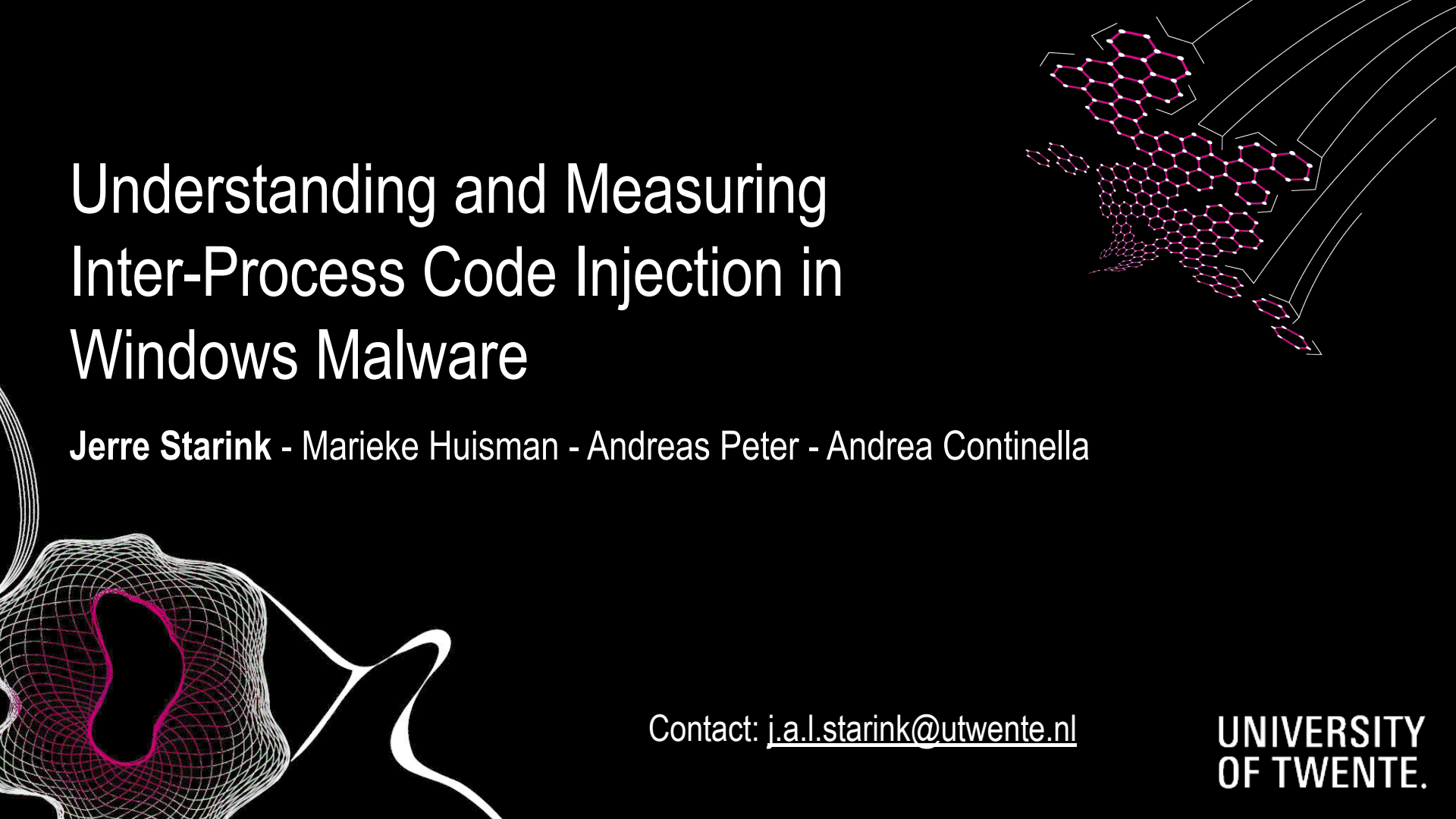
Understanding and Measuring Inter-Process Code Injection in Windows Malware

Jerre Starink - Marieke Huisman - Andreas Peter - Andrea Continella

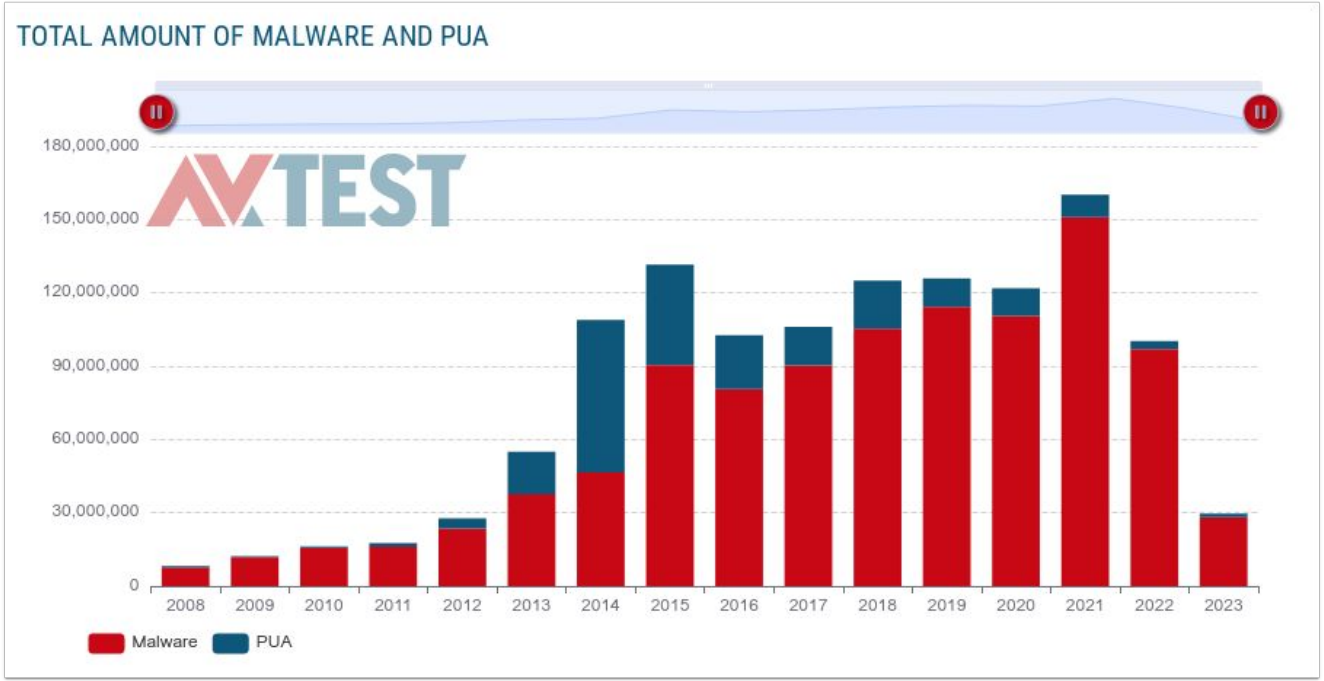


Contact: j.a.l.starink@utwente.nl

UNIVERSITY OF TWENTE.

Malware

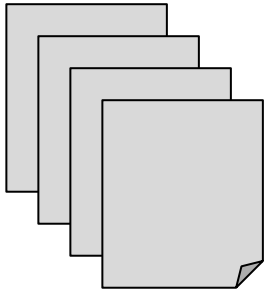
Malware



[1] https://www.av-test.org/en/statistics/malware/

What is Code Injection?

Malware.exe

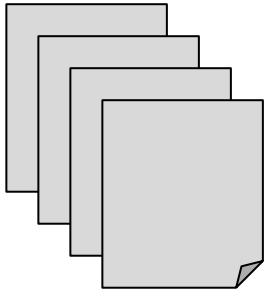


Documents

What is Code Injection?

Malware.exe

chrome.exe

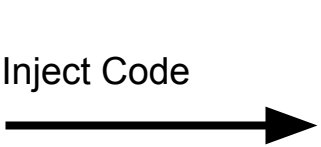


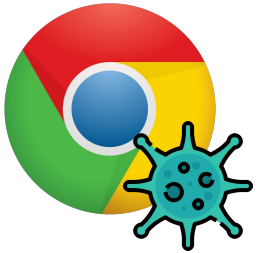
Documents

What is Code Injection?

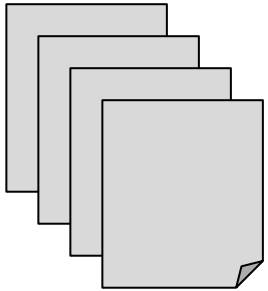
Malware.exe

Inject Code





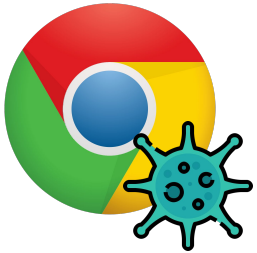
chrome.exe



Documents

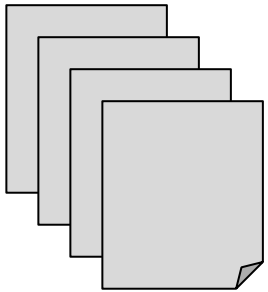
What is Code Injection?

Malware.exe



chrome.exe

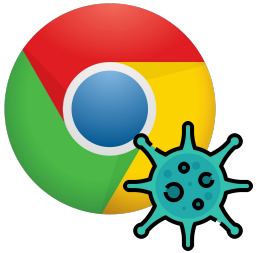
Delete files



Documents

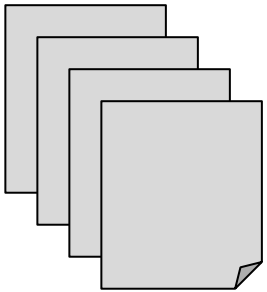
What is Code Injection?

Malware.exe



chrome.exe

Delete files



Documents

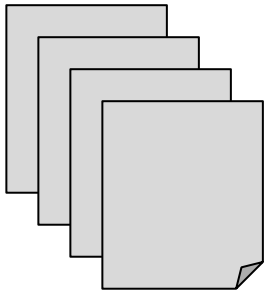
Antivirus Software

What is Code Injection?

Malware.exe

chrome.exe

Delete files



Documents

Malware remains undetected.

Antivirus Software

State-of-the-Art

- Sandboxes are able to detect traditional methods like Shell Injection, DLL Injection, or Process Hollowing (RunPE).

State-of-the-Art

- Sandboxes are able to detect traditional methods like Shell Injection, DLL Injection, or Process Hollowing (RunPE).

- There are many more code injection techniques available.

State-of-the-Art

- Sandboxes are able to detect traditional methods like Shell Injection, DLL Injection, or Process Hollowing (RunPE).
- There are many more code injection techniques available.
- Defenders and malware analysts need to stay on top of trends.

State-of-the-Art

- Sandboxes are able to detect traditional methods like Shell Injection, DLL Injection, or Process Hollowing (RunPE).
- There are many more code injection techniques available.
- Defenders and malware analysts need to stay on top of trends.
- We need to know what kind of techniques are actually being used in the wild.

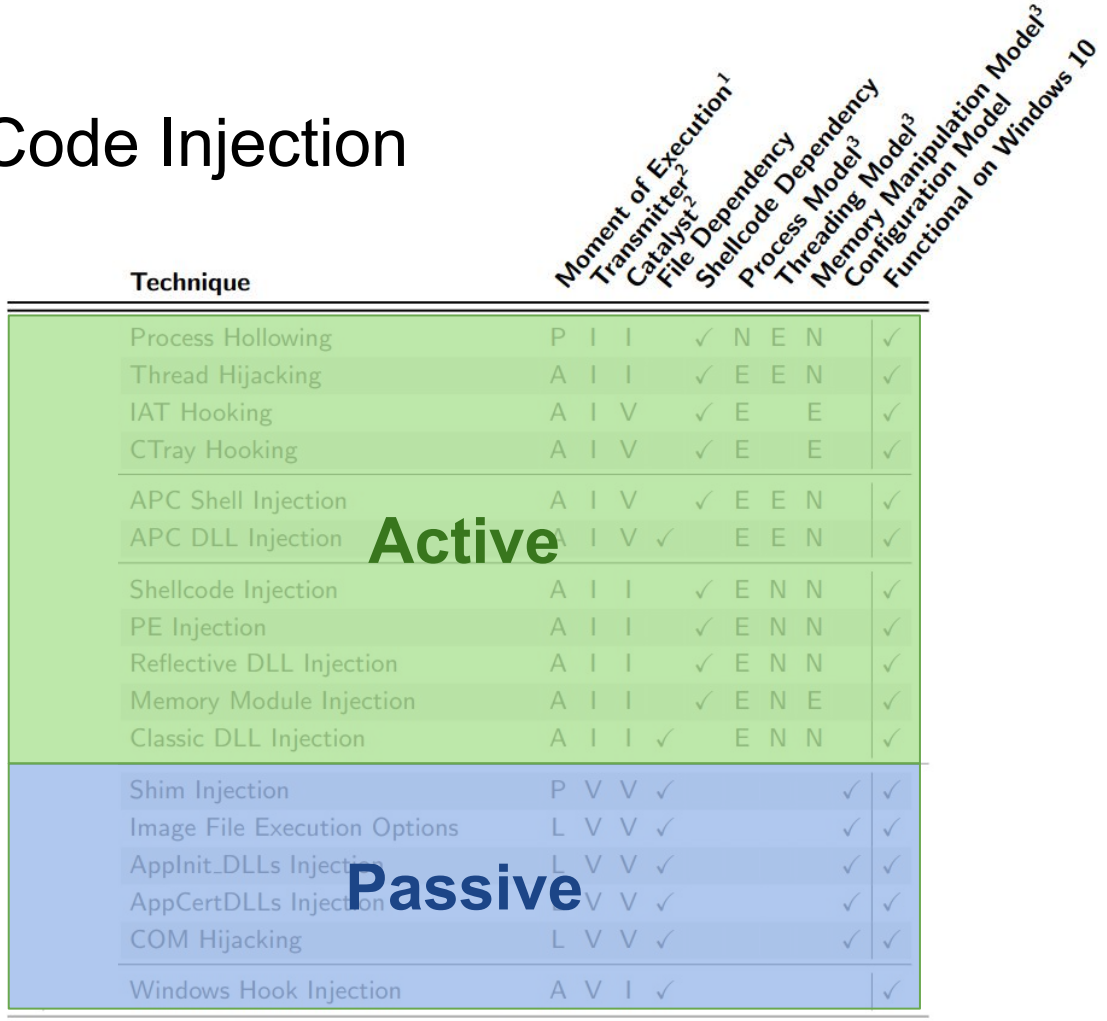
Characterizing Code Injection

Technique
Process Hollowing
Thread Hijacking
IAT Hooking
CTray Hooking
APC Shell Injection
APC DLL Injection
Shellcode Injection
PE Injection
Reflective DLL Injection
Memory Module Injection
Classic DLL Injection
Shim Injection
Image File Execution Options
AppInit_DLLs Injection
AppCertDLLs Injection
COM Hijacking
Windows Hook Injection

Characterizing Code Injection

Technique	Moment of Execution[1]	Transmitter[2]	Catalyst[2]	File Dependency	Shellcode Dependency	Process Model[3]	Threading Model[3]	Memory Manipulation Model[3]	Configuration Model	Functional on Windows 10
Process Hollowing	P	I	I	✓	N	E	N			✓
Thread Hijacking	A	I	I	✓	E	E	N			✓
IAT Hooking	A	I	V	✓	E		E			✓
CTray Hooking	A	I	V	✓	E		E			✓
APC Shell Injection	A	I	V	✓	E	E	N			✓
APC DLL Injection	A	I	V	✓		E	E	N		✓
Shellcode Injection	A	I	I	✓	E	N	N			✓
PE Injection	A	I	I	✓	E	N	N			✓
Reflective DLL Injection	A	I	I	✓	E	N	N			✓
Memory Module Injection	A	I	I	✓	E	N	E			✓
Classic DLL Injection	A	I	I	✓		E	N	N		✓
Shim Injection	P	V	V	✓					✓	✓
Image File Execution Options	L	V	V	✓					✓	✓
AppInit_DLLs Injection	L	V	V	✓					✓	✓
AppCertDLLs Injection	L	V	V	✓					✓	✓
COM Hijacking	L	V	V	✓					✓	✓
Windows Hook Injection	A	V	I	✓						✓

Characterizing Code Injection



Two Main Classes of Code Injection

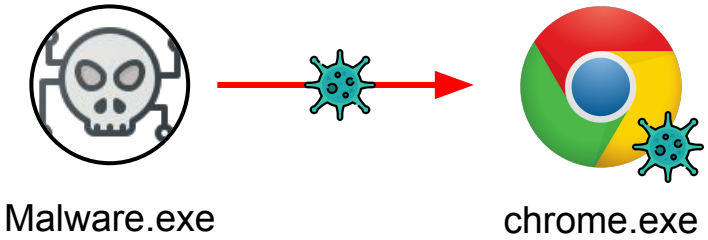
Active Injection

Malware.exe

chrome.exe

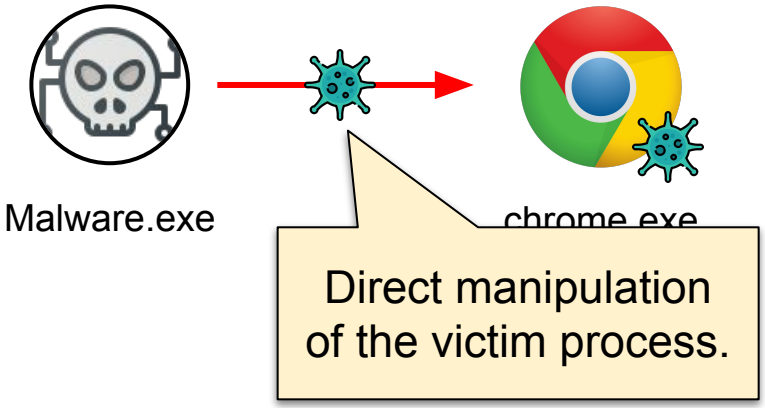
Two Main Classes of Code Injection

Active Injection



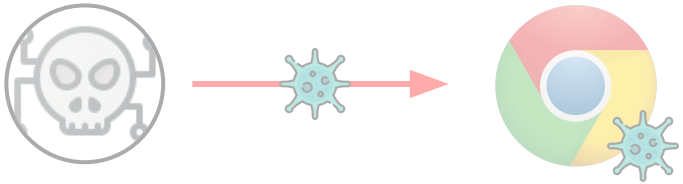
Two Main Classes of Code Injection

Active Injection



Two Main Classes of Code Injection

Active Injection



Malware.exe

chrome.exe

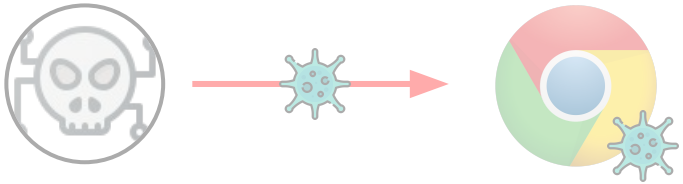
Passive Injection

Malware.exe

chrome.exe

Two Main Classes of Code Injection

Active Injection



Malware.exe

chrome.exe

Passive Injection

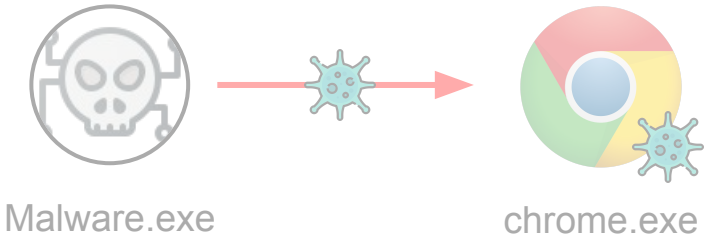
Malware.exe

chrome.exe

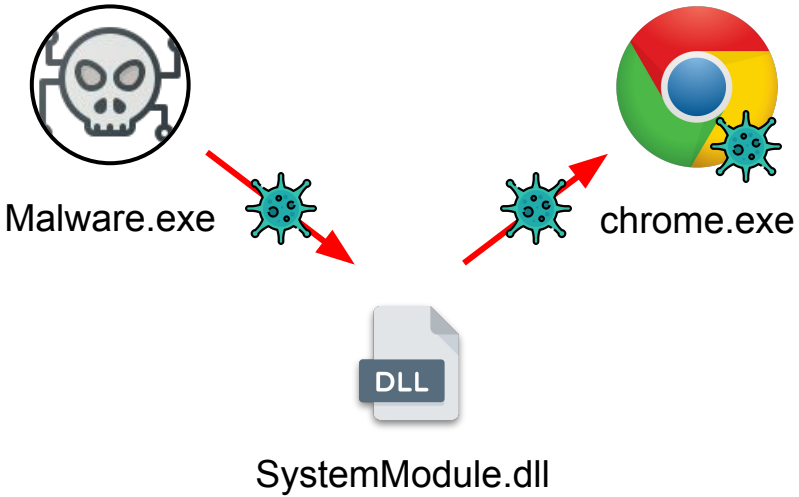
SystemModule.dll

Two Main Classes of Code Injection

Active Injection

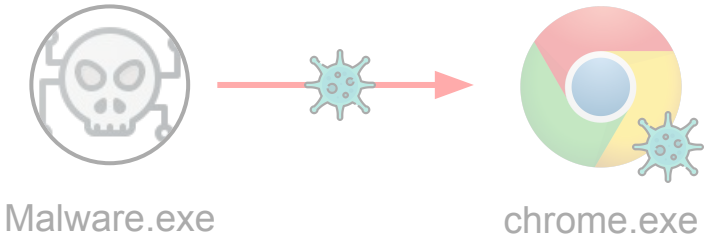


Passive Injection

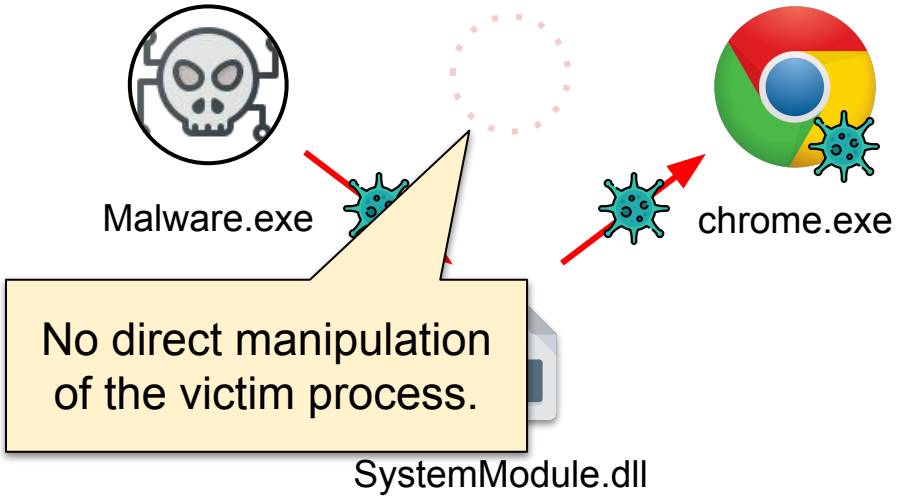


Two Main Classes of Code Injection

Active Injection

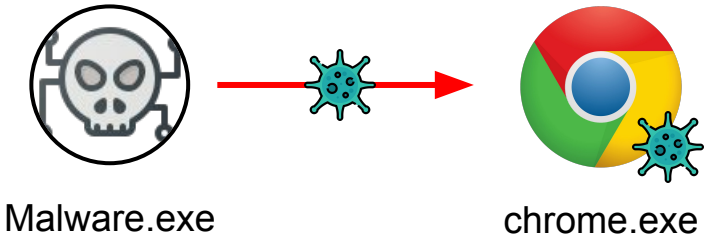


Passive Injection

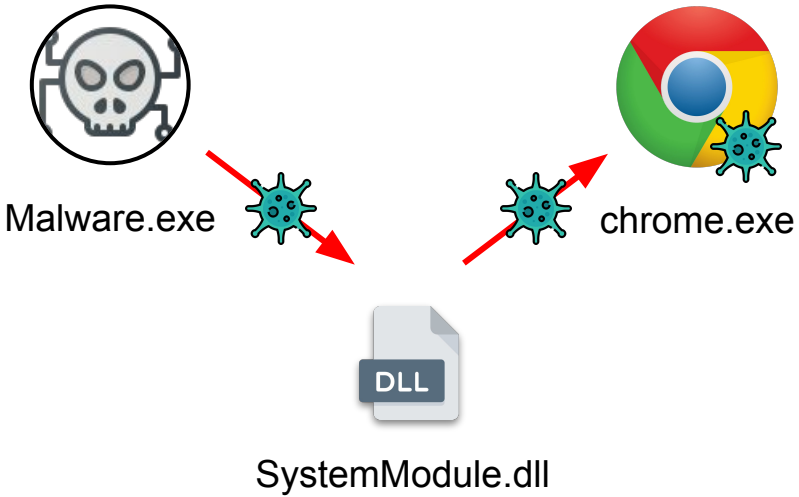


Two Main Classes of Code Injection

Active Injection

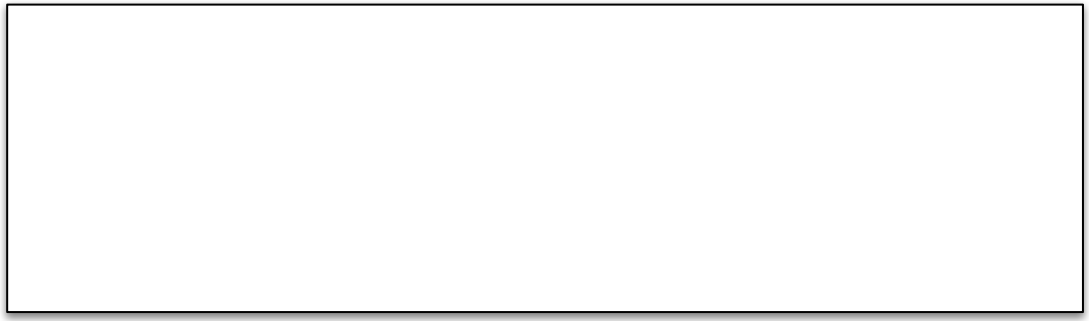


Passive Injection

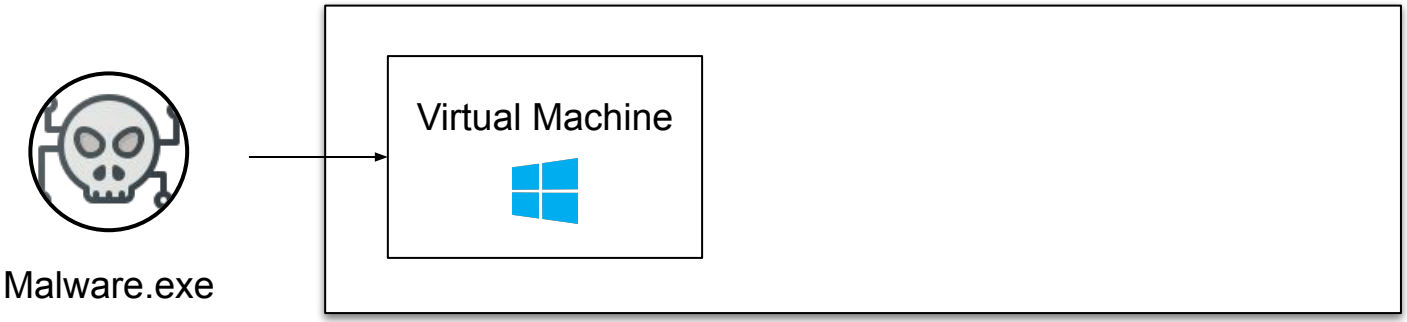


Detecting Code Injection

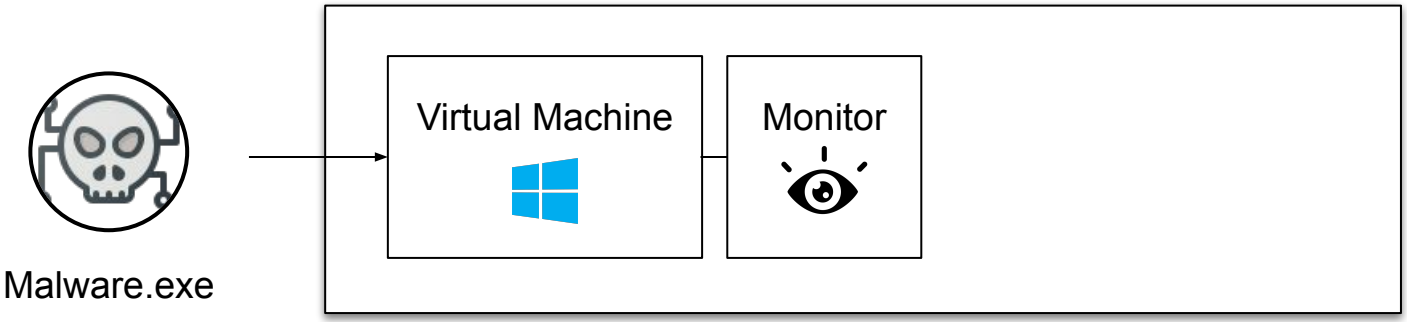
Malware.exe



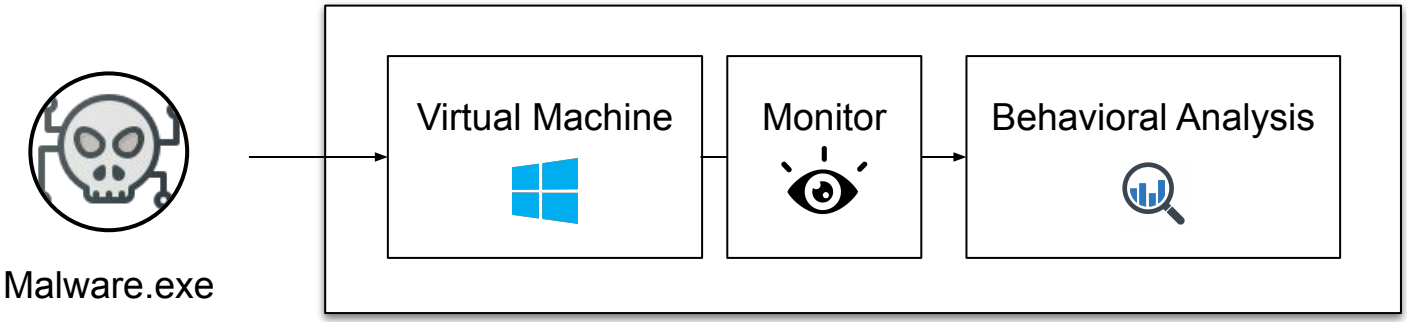
Detecting Code Injection



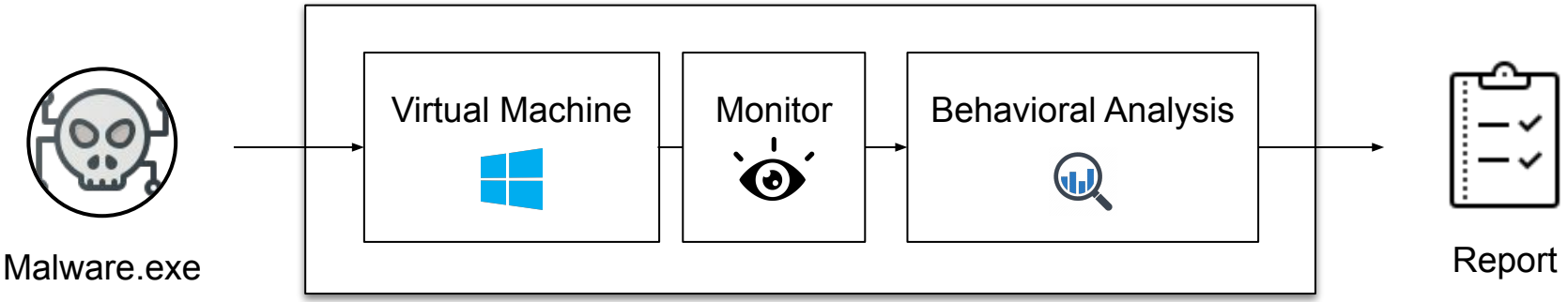
Detecting Code Injection



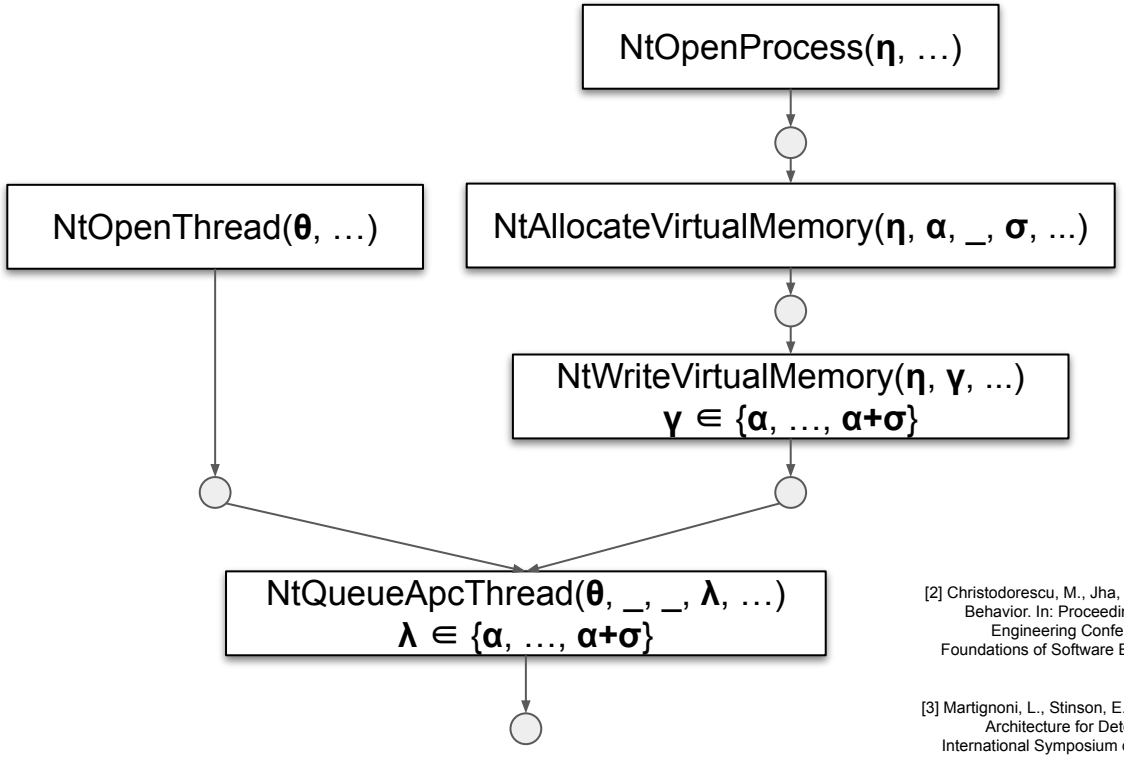
Detecting Code Injection



Detecting Code Injection



Detecting Code Injection - Behavior Graphs



[2] Christodorescu, M., Jha, S., Kruegel, C.: Mining Specifications of Malicious Behavior. In: Proceedings of the Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering. Association for Computing Machinery (2007)

[3] Martignoni, L., Stinson, E., Fredrikson, M., Jha, S., Mitchell, J.C.: A Layered Architecture for Detecting Malicious Behaviors. In: Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID) (2008)

Evaluation

- Model all 17 Code Injection techniques as behavior graphs.

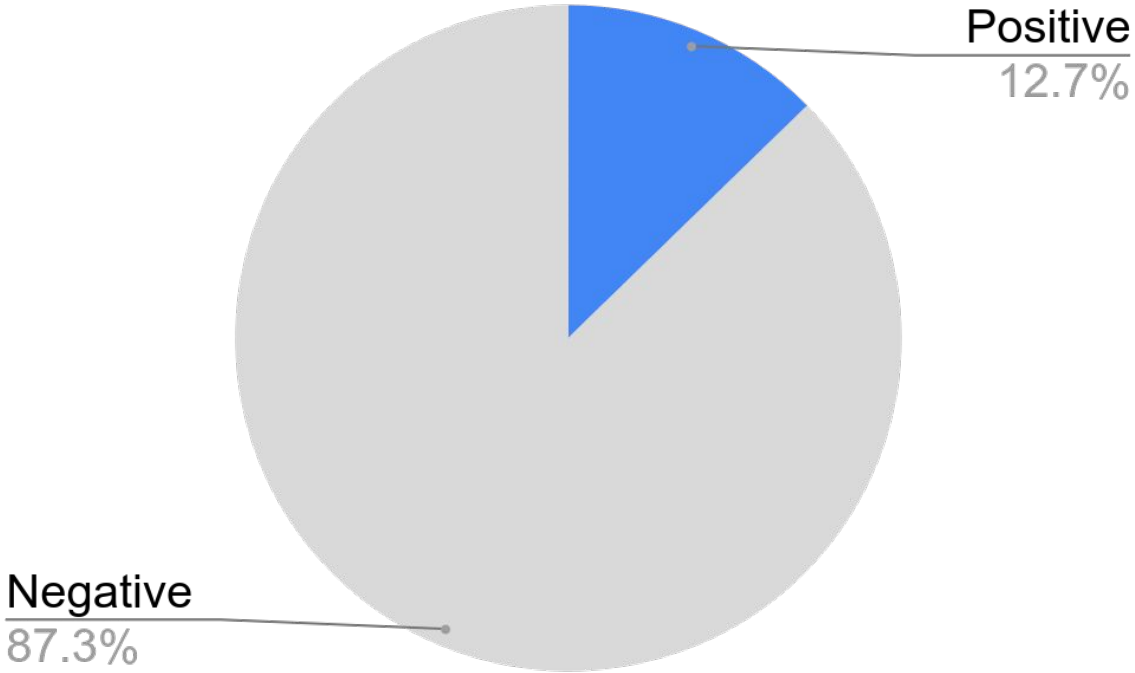
Evaluation

- Model all 17 Code Injection techniques as behavior graphs.
- Test on a ground truth dataset of 63 malware samples implementing code injection, 20 that did not, and 1,147 samples benign applications (F1: 93.0%).

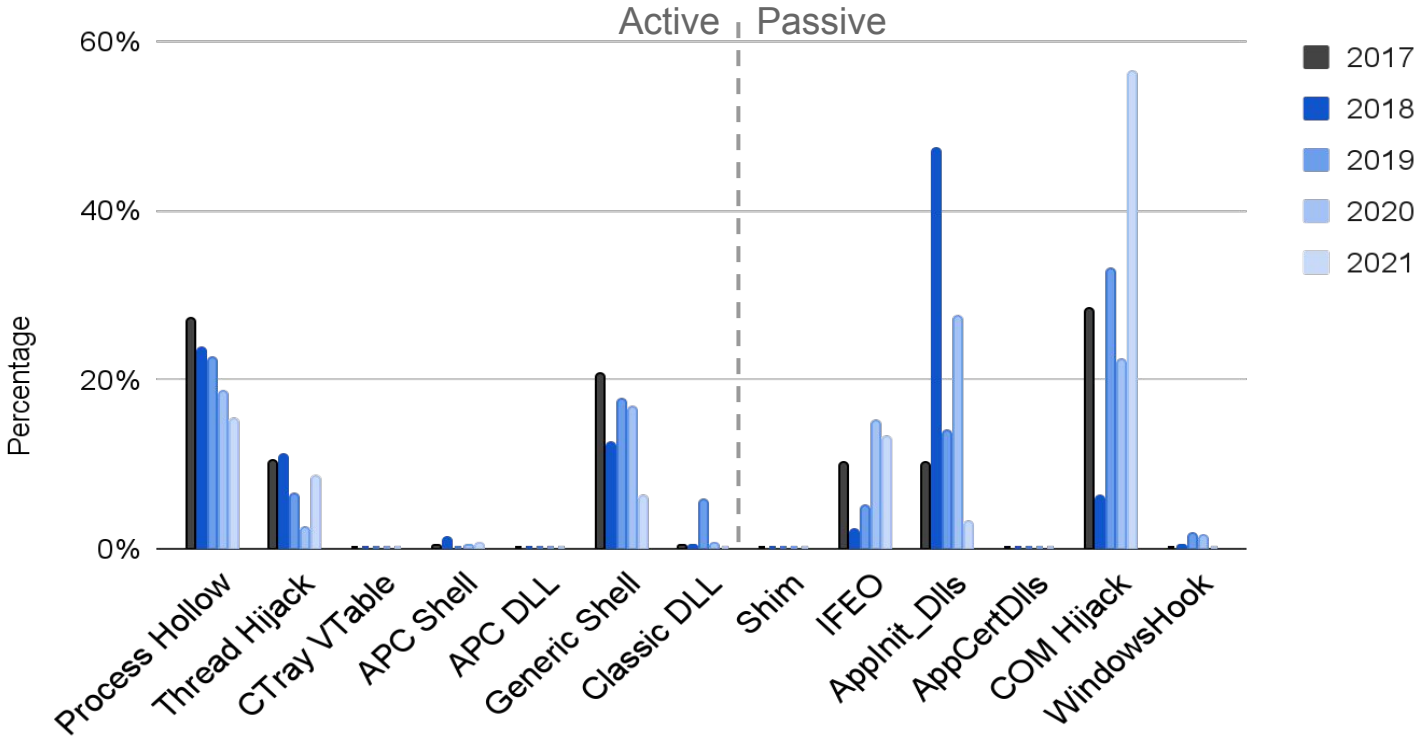
Evaluation

- Model all 17 Code Injection techniques as behavior graphs.
- Test on a ground truth dataset of 63 malware samples implementing code injection, 20 that did not, and 1,147 samples benign applications (F1: 93.0%).
- Examine 47,128 (4,278 families) random malware samples over the years 2017 - 2021 from the VirusTotal Academic Dataset.

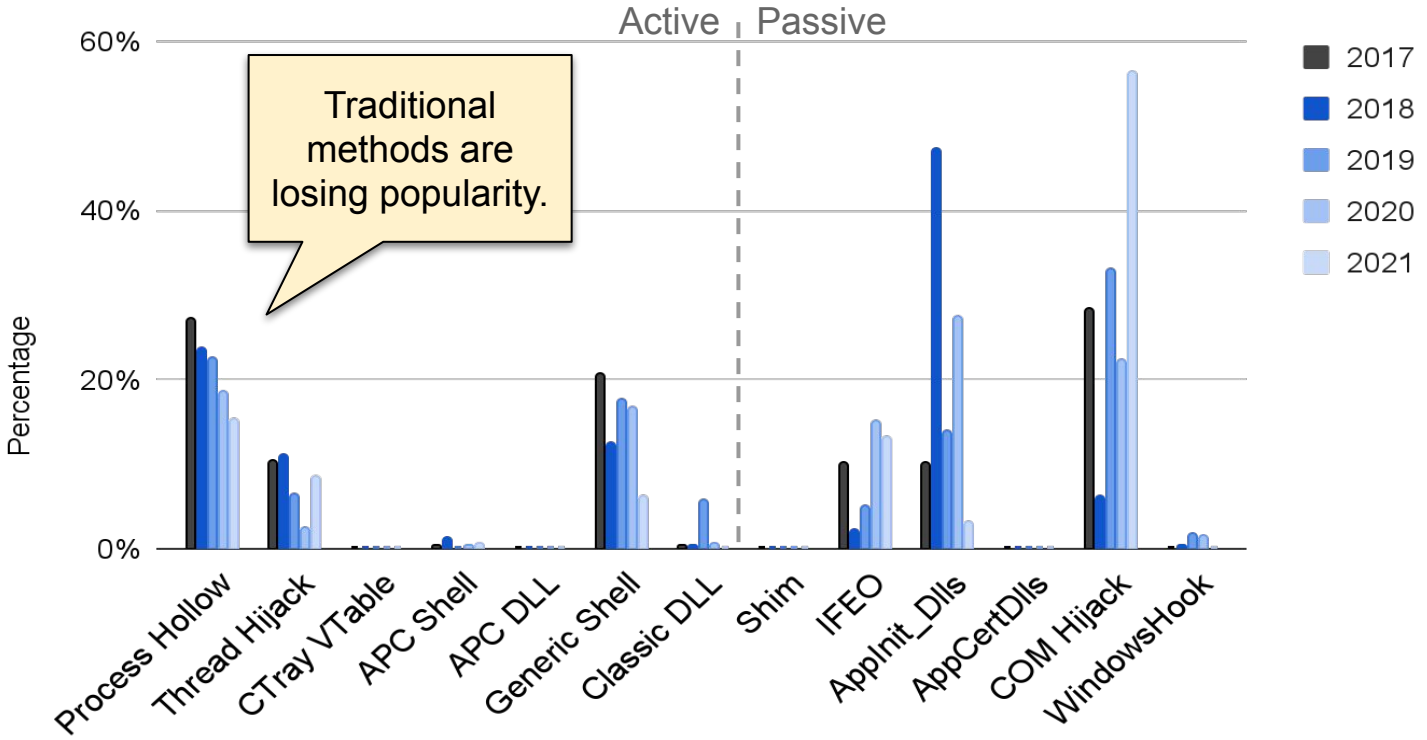
Results - General Prevalence



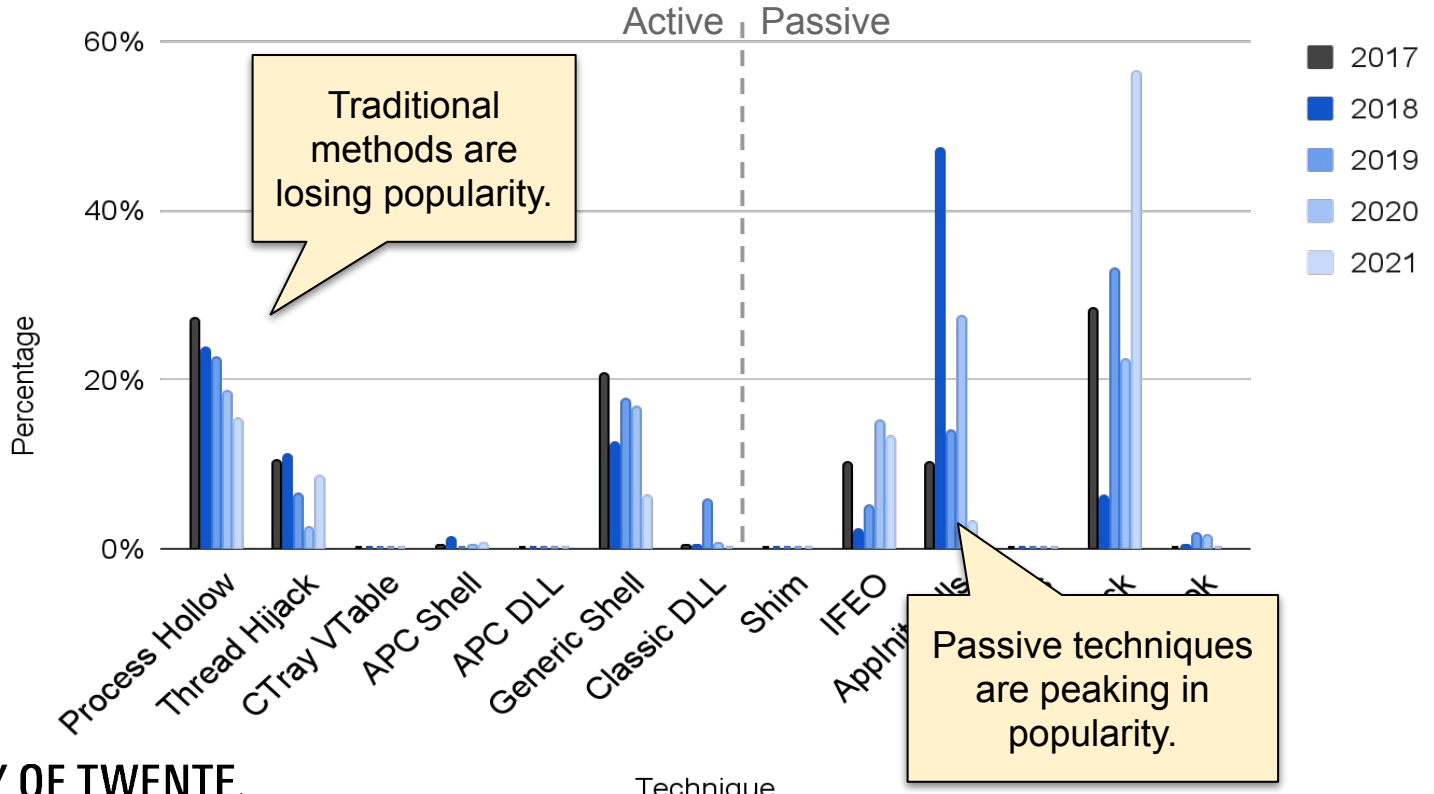
Results - Distribution of Used Techniques



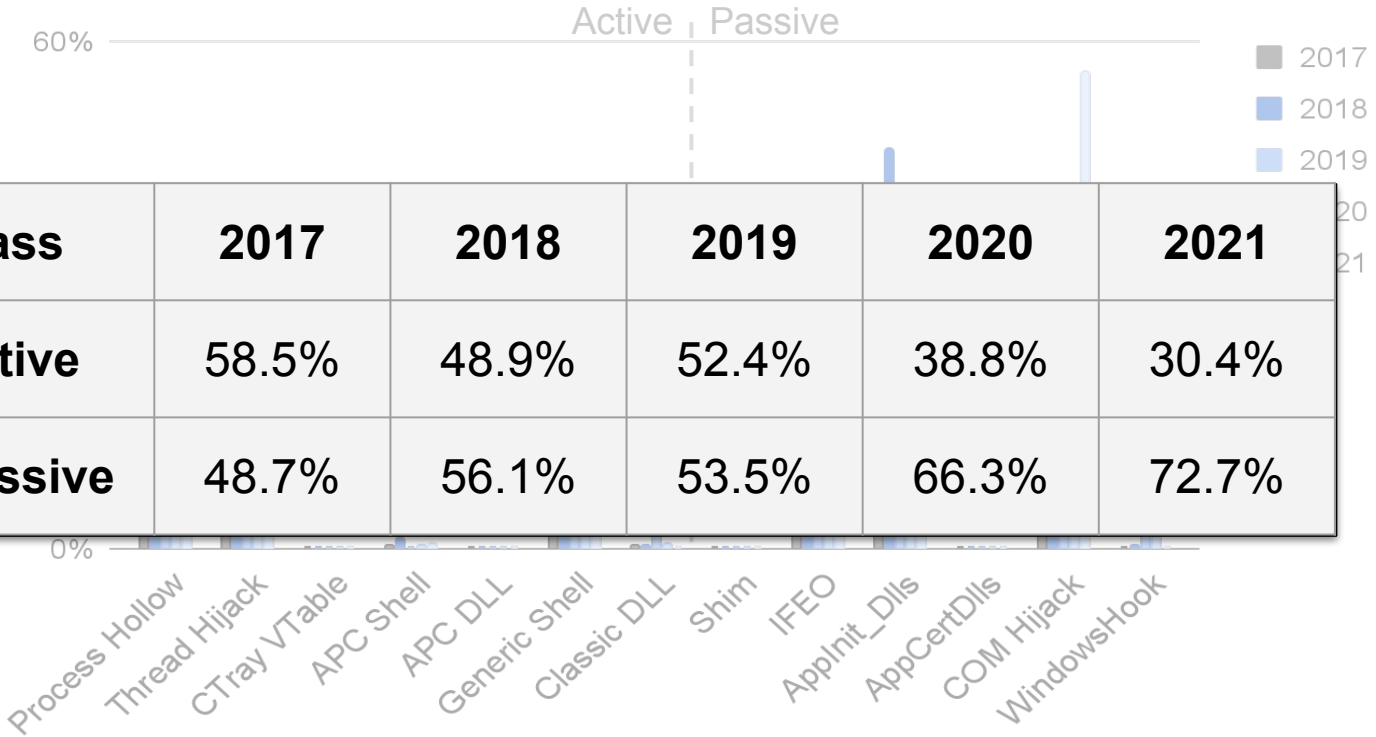
Results - Distribution of Used Techniques



Results - Distribution of Used Techniques

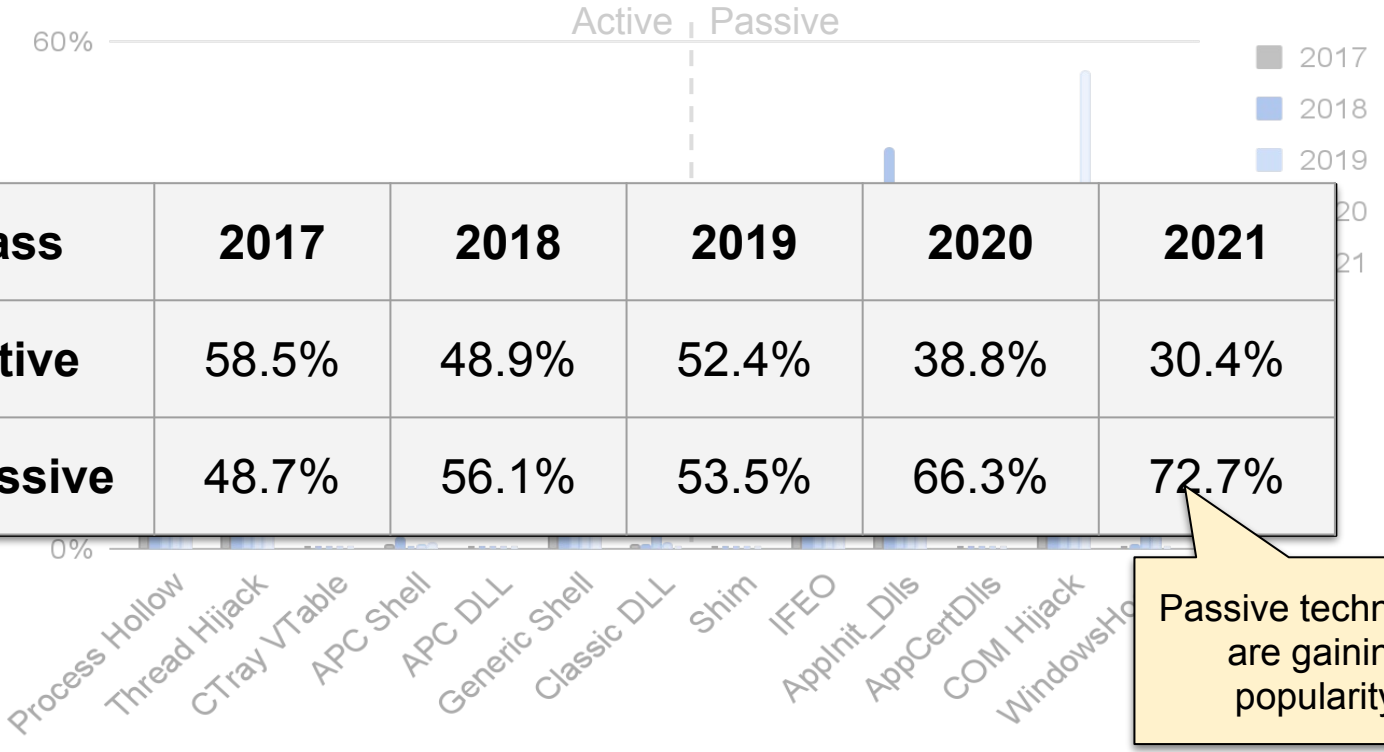


Results - Distribution of Used Techniques



Class	2017	2018	2019	2020	2021
Active	58.5%	48.9%	52.4%	38.8%	30.4%
Passive	48.7%	56.1%	53.5%	66.3%	72.7%

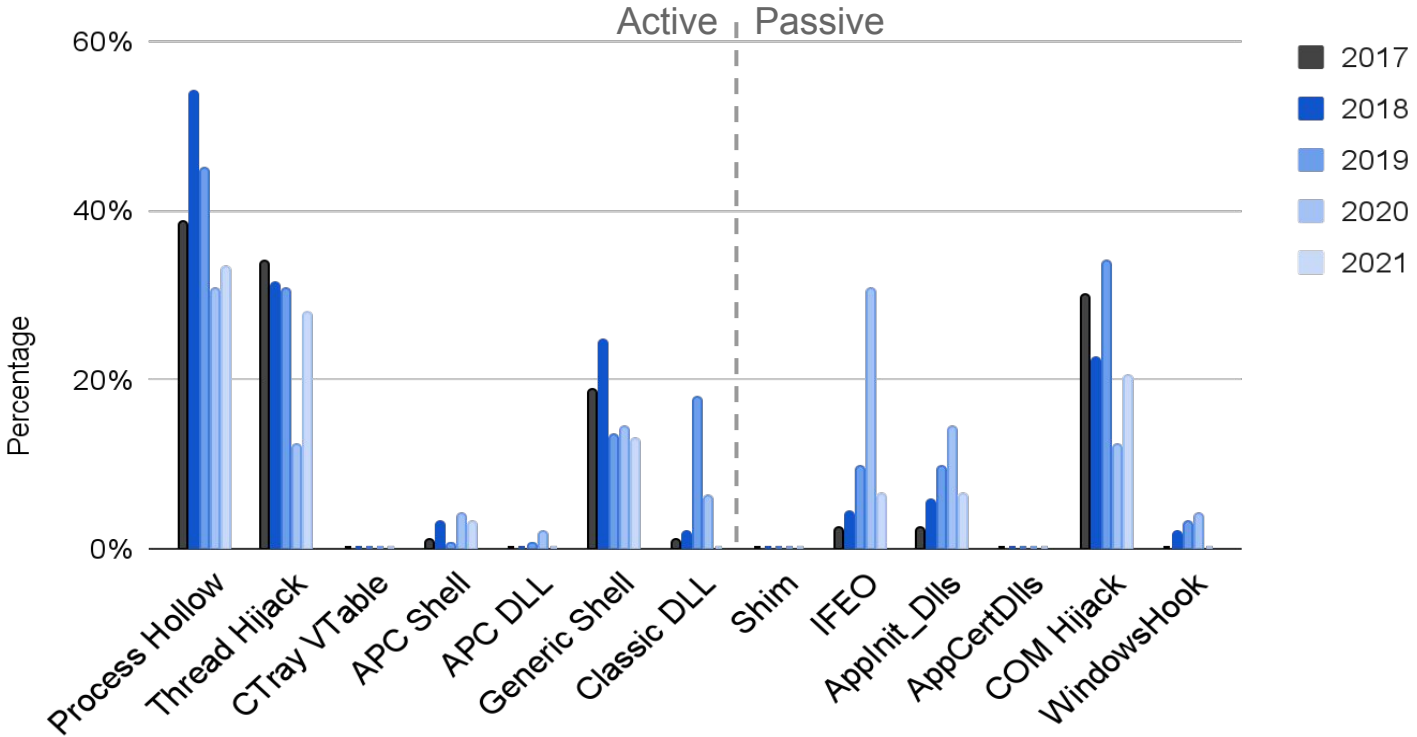
Results - Distribution of Used Techniques



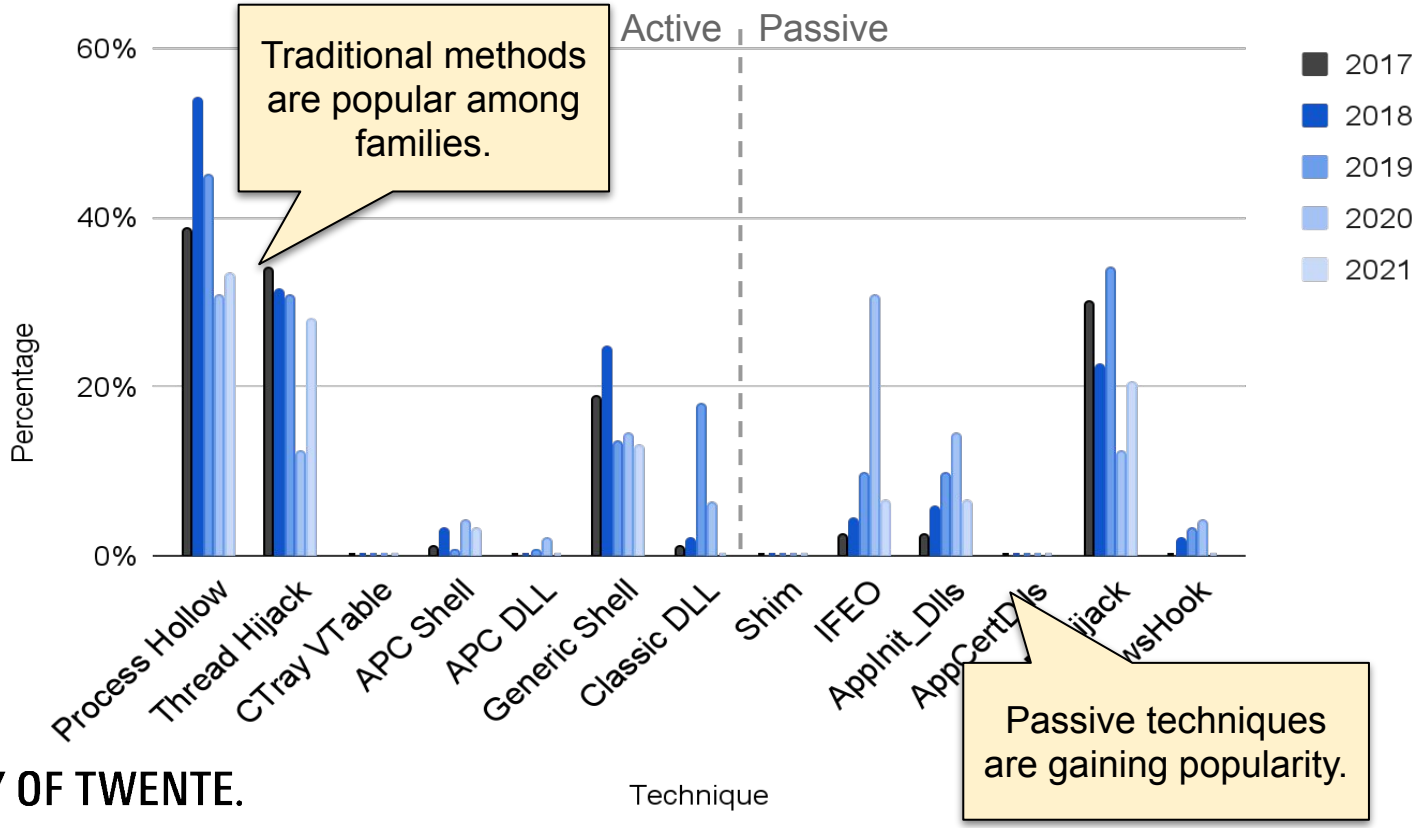
Class	2017	2018	2019	2020	2021
Active	58.5%	48.9%	52.4%	38.8%	30.4%
Passive	48.7%	56.1%	53.5%	66.3%	72.7%

Passive techniques are gaining popularity.

Results - Distribution of Used Techniques (Per Family)



Results - Distribution of Used Techniques (Per Family)



Takeaways

- Traditional methods like Process Hollowing are still the most popular.

Takeaways

- Traditional methods like Process Hollowing are still the most popular.
- However, a trend shift towards passive techniques is happening.

Takeaways

- Traditional methods like Process Hollowing are still the most popular.
- However, a trend shift towards passive techniques is happening.
- Common heuristics will be insufficient in the future.

Takeaways

- Traditional methods like Process Hollowing are still the most popular.
- However, a trend shift towards passive techniques is happening.
- Common heuristics will be insufficient in the future.
- Need for Combination of Behavioral Models.

Conclusion

- We created a taxonomy of code injection techniques.

Conclusion

- We created a taxonomy of code injection techniques.
- We created a system to detect code injection techniques in malware samples.

Conclusion

- We created a taxonomy of code injection techniques.
- We created a system to detect code injection techniques in malware samples.
- We measured a trendshift from traditional methods to passive techniques.

Conclusion

- We created a taxonomy of code injection techniques.
- We created a system to detect code injection techniques in malware samples.
- We measured a trendshift from traditional methods to passive techniques.
- Future researchers and malware analysts will have to take this into account.

Questions?

- We created a taxonomy of code injection techniques.
- We created a system to detect code injection techniques in malwa
- We measured a trendshift from traditional methods to passive techniques.
- Future researchers and malware analysts will have to take this into account.

https://github.com/utwente-scs/code-injection-malware

j.a.l.starink@utwente.nl

UNIVERSITY OF TWENTE.

Questions?

- We created a taxonomy of code injection techniques.
- We created a system to detect code injection techniques in malwa
- We measured a trendshift from traditional methods to passive techniques.
- Future researchers and malware analysts will have to take this into account.

https://github.com/utwente-scs/code-injection-malware

j.a.l.starink@utwente.nl

UNIVERSITY OF TWENTE.

Extra Slides

UNIVERSITY OF TWENTE.

Detecting Code Injection - API/System Call Logs

	A	B	C	D	E	F	G	H	I	J			
1	TimeStamp	PID	TID	Function	Return	Arg0	Arg1	Arg2	Arg3	Arg4	Arg5	Arg6	Arg7
1391	1627484992	4028	4360	NtCreateFile	0x0	0x1394	0x100081	0x73dd558	0x73dd520	0x0	0x0	0x7	0x7ffd(
1392	1627484992	4028	4360	NtCreateFile	0x0	0x1394	0x100081	0x73dce08	0x73dcdd0	0x0	0x0	0x7	0x7ffd(
1393	1627484992	4028	4360	NtWriteVirtualMemory	0x0	0xffffffffffffffff	0x83a281	0x73dc3b0	0x1	0x73dc2b8			
1394	1627484992	4028	4360	NtCreateFile	0xc0000034	0x1	0x80100080	0x73dc2a8	0x73dc270	0x0	0x0	0x7	0x1
1395	1627484992	4028	4360	NtWriteVirtualMemory	0x0	0xffffffffffffffff	0x83a281	0x73dc3b0	0x1	0x73dc2b8			
1396	1627484992	4028	4360	NtCreateFile	0x0	0x1394	0x100081	0x73dc6b8	0x73dc680	0x0	0x0	0x7	0x7ffd(
1397	1627484992	4028	4360	NtCreateFile	0x0	0x1394							ffd(
1398	1627484992	2132	2128	NtUnmapViewOfSection	0x0	0xffffffffffffffff							
1399	1627484992	4028	4360	NtWriteVirtualMemory	0x0	0xffffffffffffffff							
1400	1627484992	4028	4360	NtCreateFile	0x0	0x1394							
1401	1627484992	2132	2128	NtUnmapViewOfSection	0x0	0xffffffffffffffff							
1402	1627484992	2132	2128	LdrLoadDll	0x0	0x801							
1403	1627484992	2132	2128	LdrLoadDll	0x0	0x801							
1404	1627484992	4028	4360	NtWriteVirtualMemory	0x0	0xffffffffffffffff	0x83a281	0x73db510	0x1	0x73db418			
1405	1627484992	4028	4360	NtCreateFile	0x0	0x1394	0x100081	0x73db818	0x73db7e0	0x0	0x0	0x7	0x7ffd(
1406	1627484992	4028	4360	NtWriteVirtualMemory	0x0	0xffffffffffffffff	0x83a281	0x73dadc0	0x1	0x73dacc8			
1407	1627484992	4028	4360	NtCreateFile	0x0	0x1394	0x80100080	0x73dacb8	0x73dac80	0x0	0x0	0x7	0x1
1408	1627484992	4028	4360	NtWriteVirtualMemory	0x0	0xffffffffffffffff	0x83a281	0x73dadc0	0x1	0x73dacc8			
1409	1627484992	4028	4360	NtCreateFile	0x0	0x1394	0x100081	0x73db0c8	0x73db090	0x0	0x0	0x7	0x7ffd(
1410	1627484992	4028	4360	NtWriteVirtualMemory	0x0	0xffffffffffffffff	0x83a281	0x73da670	0x1	0x73da578			

eVirtualMemory 8 of 150

Results - Preferred Techniques Per Family

Family	2017	2018	2019	2020	2021	Total
virlock	COM	Shell	Shell			Shell
dinwod	COM	COM				COM
berbew	COM	COM	COM	COM	COM	COM
upatre	COM					COM
virut	IFEO		Shell	WinHook		IFEO
delf	Thread	Thread	Hollow	Hollow		Hollow
vobfus	Hollow	Hollow	Hollow		Hollow	Hollow
wapomi	COM					COM
allaple	COM					COM
vtflooder	COM					COM
shipup	AppInit	AppInit	AppInit	AppInit	AppInit	AppInit
gepys	AppInit	AppInit	AppInit	AppInit	AppInit	AppInit
Other	Hollow	Hollow	Hollow	Shell	Hollow	Hollow

Some families switch technique over time.