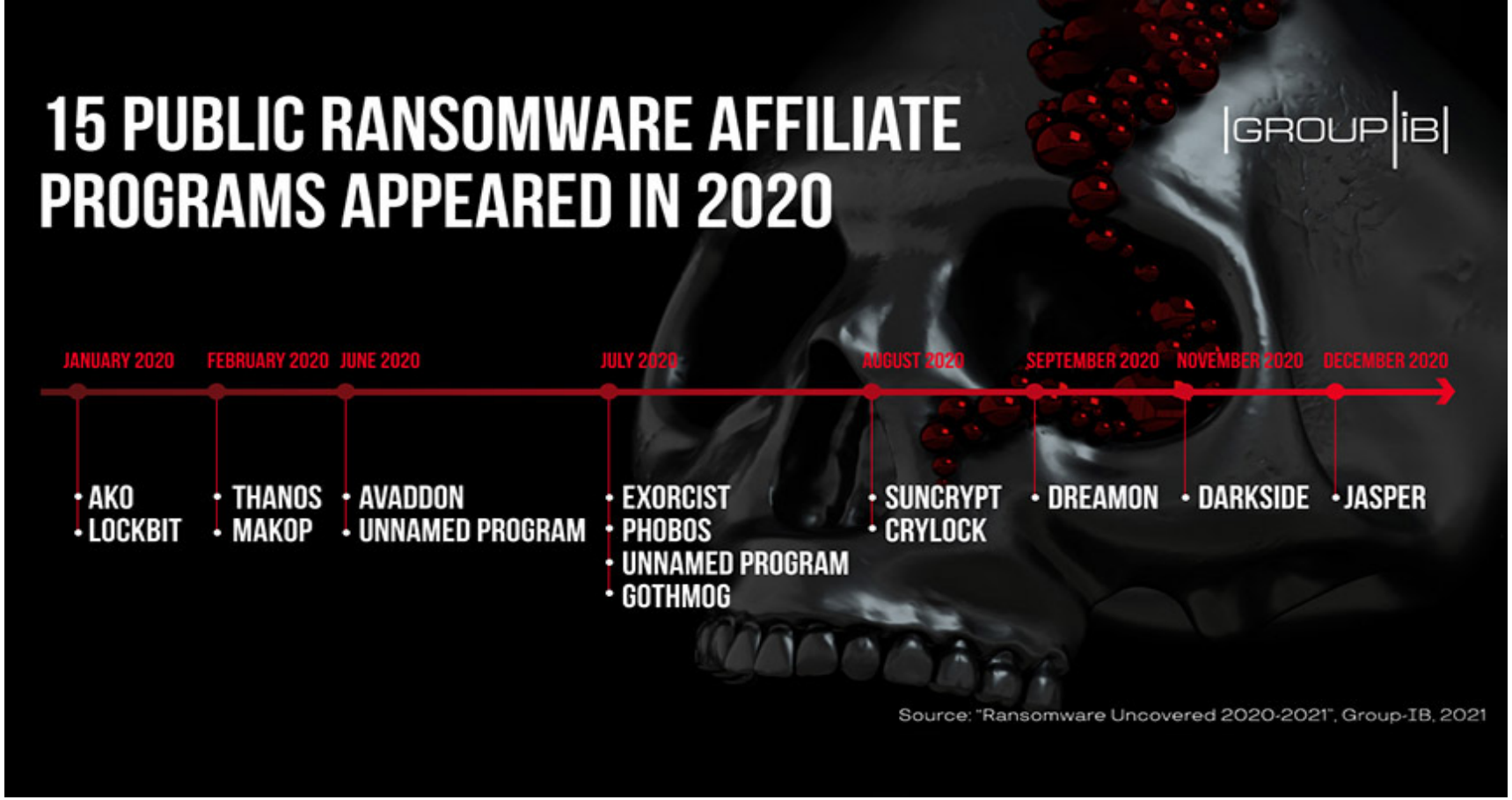
THE UNIVERSITY OF MEMPHIS
Center for Information Assurance

AI-Powered Ransomware Detection (AIRaD) Framework

Subash Poudyal, Dipankar Dasgupta
{spoudyal, dasgupta}@memphis.edu
Center for Information Assurance
Department of Computer Science
The University of Memphis

Introduction

- Advance Malware (such as ransomware) are now in forefront of national news, which are taking advantage of the pandemic and attacking various industrial and government sectors including healthcare.
- Various approaches continually being proposed to combat malware, but the dynamic nature of the malware often bypasses the security checkpoints.
- Our AI-based ransomware detection tool (AIRaD) uses hybrid analysis in a ML framework and is based on behavioral chains, linking multiple levels of intermediate codes.
- Our experiments demonstrate that AIRaD tool is able to detect ransomware with high accuracy (and low false positive rate) when tested with real malware samples from the VirusTotal website (a distribution site for research).



- Ransomware hackers profited $370 million in 2020 - paid by cryptocurrencies, which represents a 336% increase over known 2019 earnings. (By Chainalysis firm)
- 23% of incidents are ransomware compared to 2019. (IBM Threat Intelligence Index report)
- 127 new ransomware families discovered in 2020. (Trend Micro report).

The 10 industries most targeted by ransomware attackers in 2020 (Source: Trend Micro)

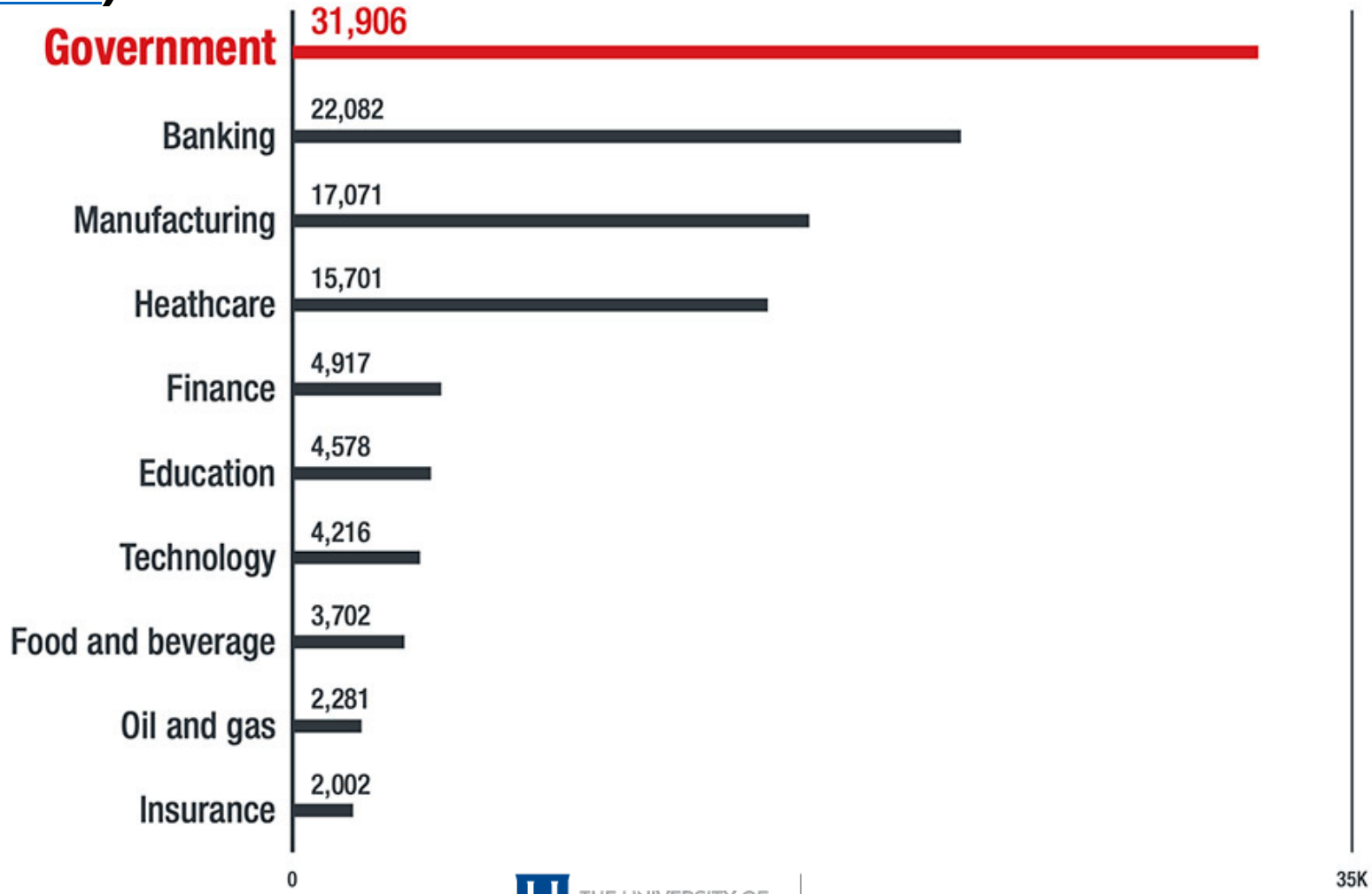


TABLE I: Recent ransomware families with some properties

Family	Propagation strategy	Date appeared	Cryptographic technique	C&C Server
Cerber	Email spam, RIG and magnitude exploit kit	2015	RC4 and 2048-RSA	IP range
TeslaCrypt	Angler browser exploit kit	2015	AES-256	Tor anonymity network
Shade	Malicious websites, exploit kits, infected email attachments	2015	256-AES	Tor anonymity network
Locky	Spam campaigns, Neutrino exploit kit, Nuclear exploit kit, RIG exploit kit	2016	RSA and AES	Using DGA algorithm
Dharma	Unprotected RDP port, Spam campaigns	2016	256-AES and 1024-RSA	Random IPs/locations
Stop	Malicious email attachments/advertisements, torrent websites	2018	256-AES and 1024-RSA	Listed address
GandCrab	Spam emails, exploit kits	2018	AES and RSA	Tor anonymity network/DGA
Ryuk	TrickBot and RDP	2018	AES and RSA	Internet-facing Mikrotik router
Anatova	Spear phishing in private p2p network	2019	RSA and Salsa20	Listed address
AgeLocker	Age encryption tool of Google	2020	X25519 (an ECDH curve), ChaChar20-Poly1305, HMAC-SHA256	Tor network

Table shows recent ransomware families with properties such as propagation strategy, cryptographic technique and command and control server used.

```
<div class="note private">
<br><br>
<div class="bold">For data recovery needs decryptor.</div>
<br>
<div>If you want to buy a decryptor, click the button<br><div>
<br><br>
<script> function tokhxvpmomub(search,replace,subject){if(!(replace instanceof Array)){replace=new Array(replace);if(search instanc
instanceof Array)}search=new Array(search);while(search.length>replace.length){replace[replace.length]=''}if(subject instanceof Arr
k=0;k<search.length;k++){var i=subject.indexOf(search[k]);while(i>-1){subject=subject.replace(search[k],replace[k]);i=subject.index
```

ocmon3.exe"
ocmon3.exe"
t8D33.tmp.bat

Welcome

Your files are Encrypted!

For data recovery needs decryptor.

If you want to buy a decryptor, click the button

Yes, I want to buy

Free decryption as guarantee.
Before paying you can send us 1 file for free decryption.
To send a message or file use this link.
(If you send a file for free decryption, also send file RECOVER-FILES.HTML)
Support

And finally, if you can not contact, follow these two steps:

8. Ransom message shown

https://n224ezvhg4sgyamb.onion.link/efwdaq.php

n224ezvhg4sgyamb.onion.l...

To buy the decryptor, you must pay the cost of: 0.35 Bitcoin ($ 991.46)
You have 2 days for payment
time left :

47:59:04

after finishing offer, decryptor cost will be 0.7 Bitcoin

You can buy bitcoin on one of these sites:
blockchain.info
localbitcoins.com
btc4cc.com

send 0.35 bicoin on the Bitcoin address: 19SYVx82iShrpbeCwaQBpJcLs6TtQ2L1aU

Only after payment
Enter your REAL e-mail to get the decryptor and click Enter

```
<div class="note private">
<br><br>
<div class="bold">For data recovery needs decryptor.</div>
<br>
```

7. Encrypt using RSA-2048 public key

```
= (const CHAR *)RtlAllocateHeap(v8, 0, 256);
rintfA(v9, "%d", pe.th32ProcessID);
epyA(&String1, "taskkill /F /T /PID ");
rcatA(&String1, v9);                    // run taskkill to kill matched processes.
ateProcessA(0, &String1, 0, 0, 0, 0x8000000u, 0, 0, &StartupInfo, &ProcessInformation);
( Process32NextW(v2, &pe) );
t = (HANDLE)CloseHandle(v2);
```

Hybrid reverse engineering

- Involves both static and dynamic analysis of ransomware and benign binaries.
- Need of hybrid analysis
- Static analysis: High code coverage but misses the behavior
- Dynamic analysis: Real behavior but may have Parameter missing
- Sandboxing and dynamic binary instrumentation

AIRaD Steps and applied Methodologies.

- Hybrid multi-level behavior profiling based ransomware detection framework
- Feature extraction at DLL, function call and assembly level
- Three approaches in ML processor
- Pattern discovered useful in Yara signature creation
- ML classifier

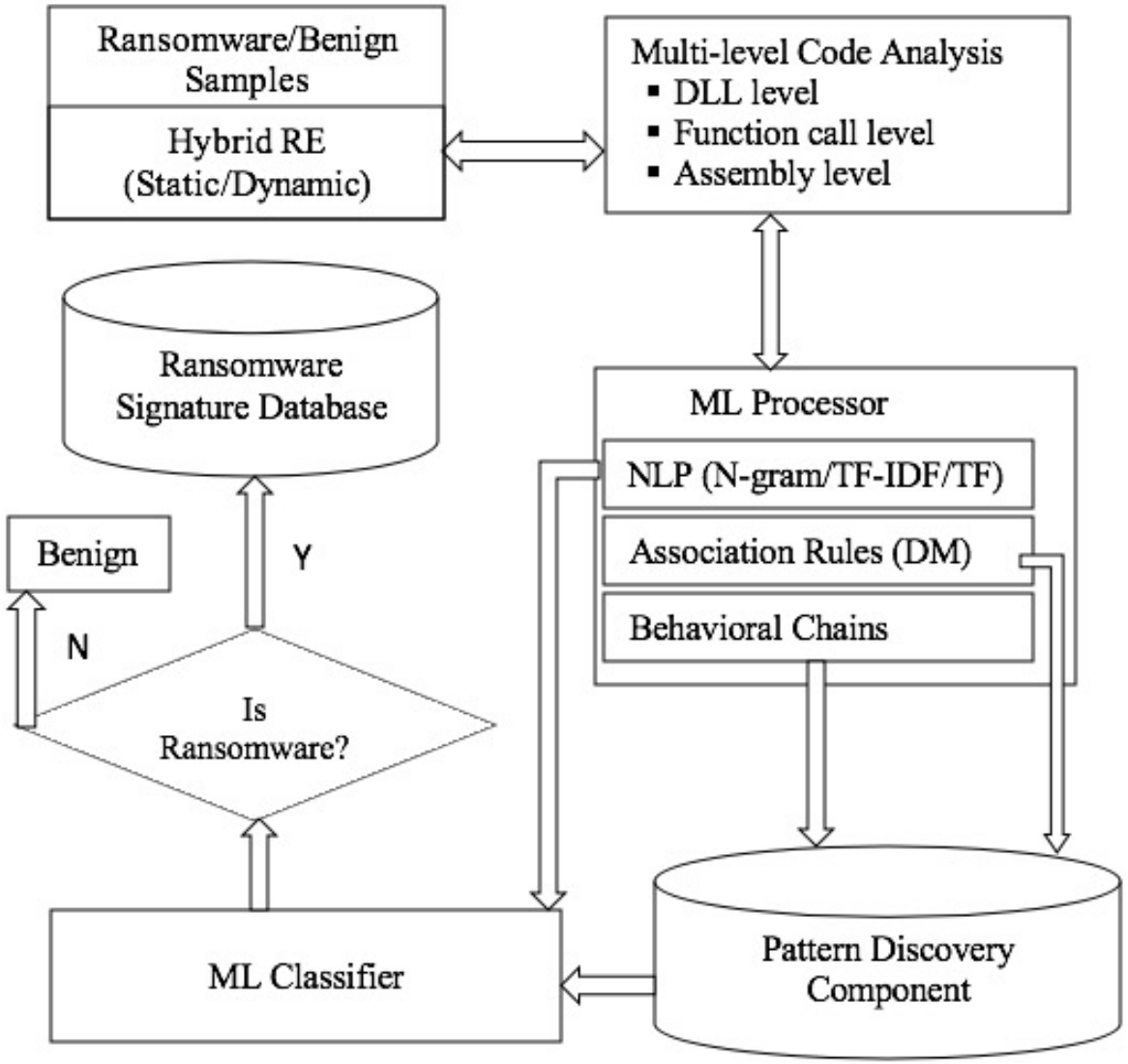


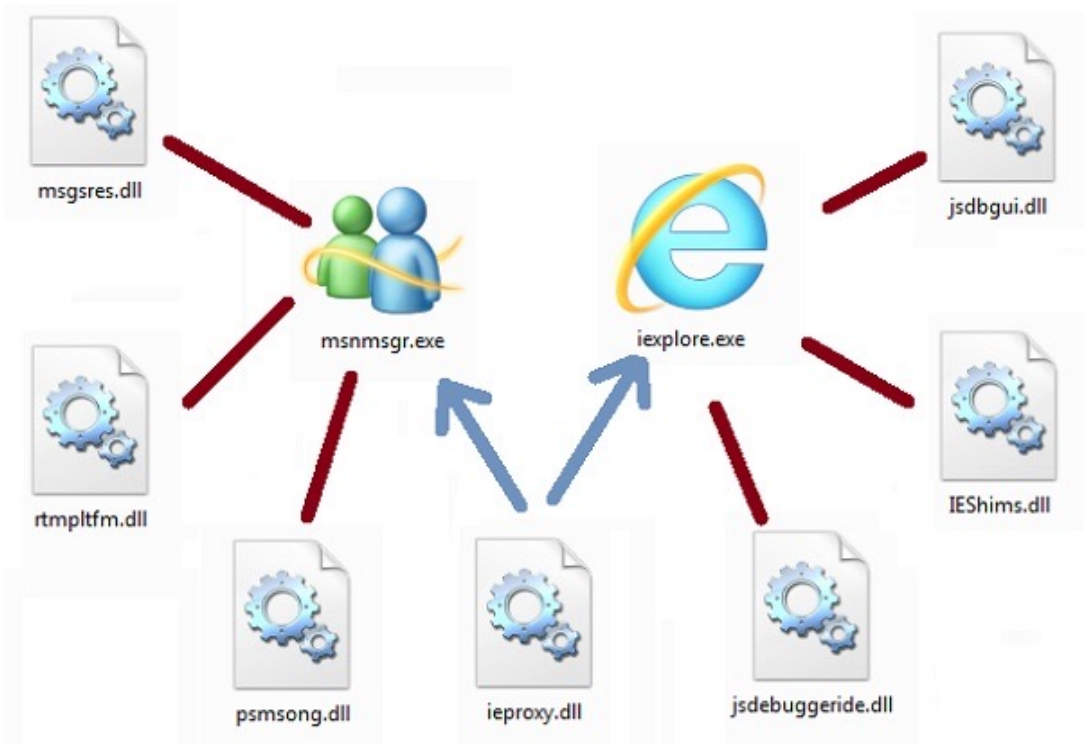
Fig: The overall backend architecture for AI-powered ransomware detection framework

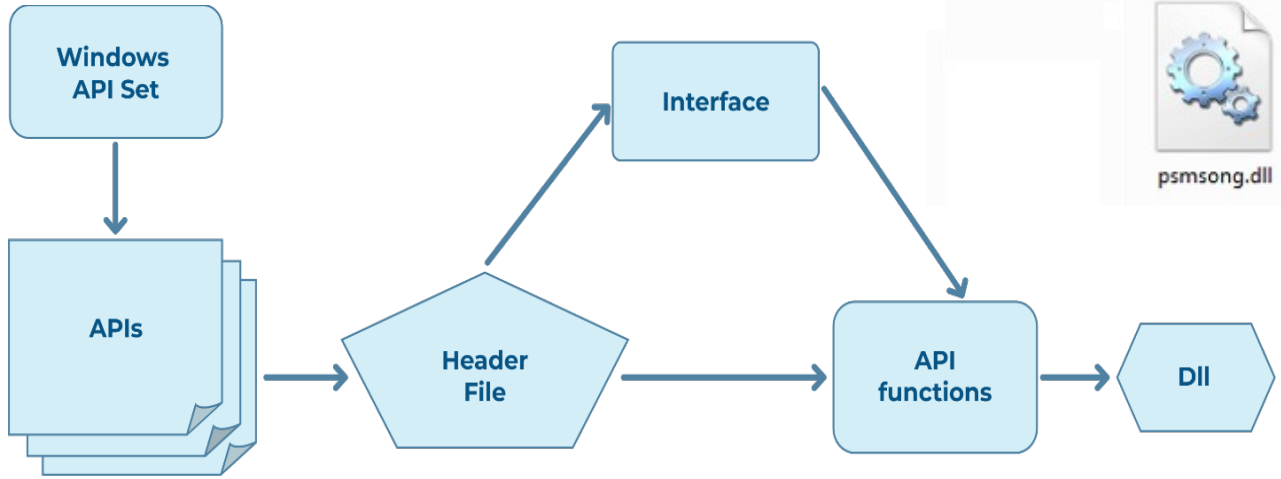
Dynamic binary instrumentation

- Instrumentation techniques have been used by programmers to diagnose program crashes, analyze errors and to write trace information.
- Comes in two types: code instrumentation and binary instrumentation.
- PIN makes tracking of every instruction executed possible by taking complete control over the run-time execution of the binary.

Hierarchy of windows DLL

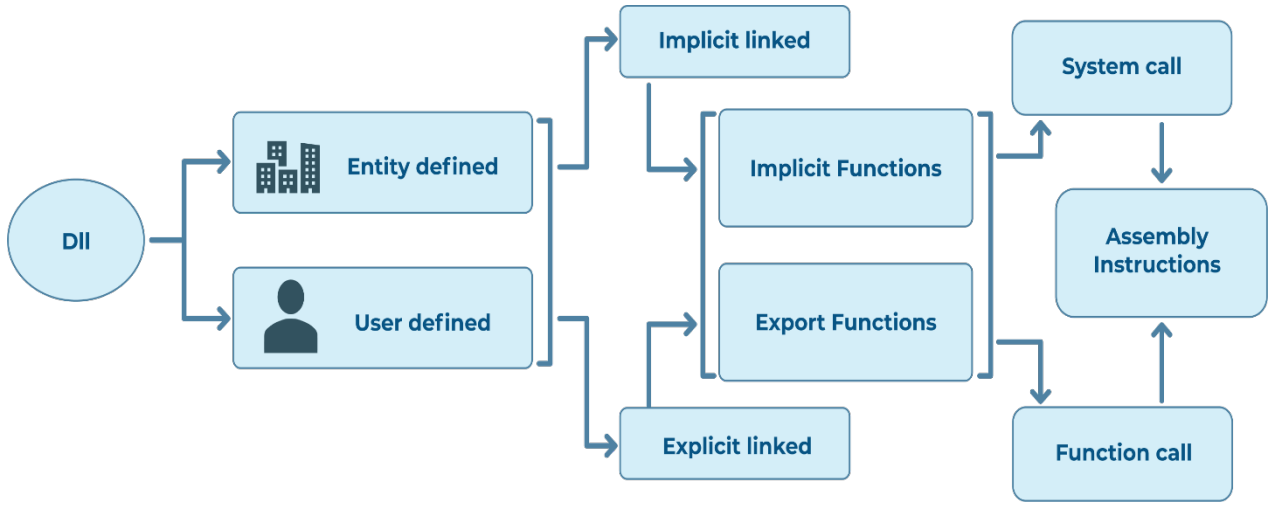
- **A dynamic link library (DLL) contains code and data that can be used by more than one program at the same time.**
- **The main benefit of DLL is code re-usability and efficient memory usage.**





Hierarchy of function calls and assembly instructions in a DLL

- A function call is a piece of code that actually has lines of instructions that makes an impact to the system or user.
- Each DLL which is implicitly or explicitly linked consists of both import and export functions as shown in figure.



Level 1: Association rules at DLL level

- The first-row rule shows the use of ADVAPI32, OLE32, SHELL32, and WS2_32 DLL implies KERNEL32 DLL.
- This rule contains DLLs associated with ransomware specific behavior including encryption

Association rules at DLL level
[ADVAPI32, OLE32, SHELL32, WS2_32] → [KERNEL32]
[ADVAPI32, OLE32, SHELL32, WS2_32] → [ADVAPI32]
[ADVAPI32, OLE32, SHELL32, WININET] → [KERNEL32]
[ADVAPI32, OLE32, SHELL32, WININET, WS2_32] → [KERNEL32]
[KERNEL32, OLE32, SHELL32, WININET, WS2_32] → [ADVAPI32]
[ADVAPI32, KERNEL32, OLE32, SHELL32, WS2_32] → [WININET]
[ADVAPI32, KERNEL32, OLE32, WININET, WS2_32] → [SHELL32]
[ADVAPI32, KERNEL32, OLE32, SHLWAPI, WININET, WS2_32] → [SHELL32]
[ADVAPI32, KERNEL32, OLE32, SHELL32, SHLWAPI, WININET] → [WS2_32]
[ADVAPI32, KERNEL32, OLE32, SHELL32, WININET, WS2_32] → [SHLWAPI]

Level 2: Association rules at Function level

- The first-row rule shows the use of GetCurrentThreadId, GetTickCount, RtlUnwind, WideCharToMultiByte, lstrlenW implies ExitProcess.
- This rule contains function calls that are specific to the anti-analysis behavior of ransomware.

Association rules at function level
[GetCurrentThreadId, GetTickCount, RtlUnwind, WideCharToMultiByte, lstrlenW] → [ExitProcess]
[CloseHandle, GetModuleHandleA, GetTickCount, ReadFile, RtlUnwind, WideCharToMultiByte] → [GetProcAddress]
[CloseHandle, GetCurrentProcessId, GetModuleHandleA, ReadFile, RtlUnwind, WideCharToMultiByte] → [ExitProcess]
[ExitProcess, GetCurrentThreadId, GetTickCount, SetFilePointer, Sleep, WideCharToMultiByte] → [GetModuleHandleA]
[ExitProcess, GetCurrentThreadId, GetModuleHandleA, RtlUnwind, SetFilePointer, WideCharToMultiByte] → [WriteFile] 1.0)
[CloseHandle, ExitProcess, GetCurrentProcessId, GetModuleHandleA, RtlUnwind, WideCharToMultiByte] → [ReadFile]
[GetCurrentProcess, InterlockedIncrement, IsDebuggerPresent, RaiseException, RtlUnwind, SetUnhandledExceptionFilter, Sleep, UnhandledExceptionFilter, VirtualProtect, WideCharToMultiByte] → [HeapCreate]
[EnterCriticalSection, ExitProcess, GetLastError, GetProcAddress, LeaveCriticalSection, WriteFile] → [HeapReAlloc]

Level 3: Association rules at Assembly level

- These rules are specific to different function calls defined for various chains.

Association rules at Assembly level
[add, and, cmp, data16, jmp, lea, sbb, shl] → [xor]
[add, cmp, data16, jm, lea, sbb, shl, sub] → [xor]
[add, cmp, data16, jmp, or, rcr, sbb] → ror]
[add, cmp, data16, jmp, or, rcr, shl] → sbb]
[add, cmp, data16, jmp, rcr, sbb, shl] → [xor]
[add, cmp, fdivr, jmp, les, or, repz, sbb, sub] → [bound, ror]
[add, cmp, fdivr, jmp, les, or, repz, shl, sub] → [sbb]
[add, cmp, fdivr, jmp, les, repz, sbb, shl, sub] → [xor]
[add, call, ficom, les, lock, or, rcr, sbb, xchg] → [cmp]
[call, cmp, ficom, les, lock, or, rcr, sbb, xchg] → [add]

Ransomware behavior chains

- Ransomware specific behavioral chains are basically multi-level chains which are constructed by studying the behavior of different ransomware families.
- Both static and dynamic analysis of ransomware binaries reveals the different chains which are seen in a wide range of ransomware families.

- Chain A uses GetStartupInfoW which gets information related to the window station, desktop, and appearance of the main window.
- In Chain C ,GetSystemInfo and GetNativeSystemInfo use dwNumberOfProcessors method to check the number of processors running in a system.
- In Chain D, OpenProcessToken opens the token associated with a given process while GetTokenInformation is used to obtain the token id, session id, or security identifier of the process's owner.

CHAIN A	System services with initial setup
CHAIN B	Module enumeration
CHAIN C	Anti-analysis
CHAIN D	Access elevation
CHAIN E	Snapshot
CHAIN F	Parameter setup
CHAIN G	System profiling
CHAIN H	Encryption setup
CHAIN I	File encryption
CHAIN J	Ransom note
CHAIN K	Network enumeration
CHAIN L	Deletion
CHAIN M	Error handling
CHAIN N	CC communication

Table: Ransomware specific behavioral chains

Self deletion chain

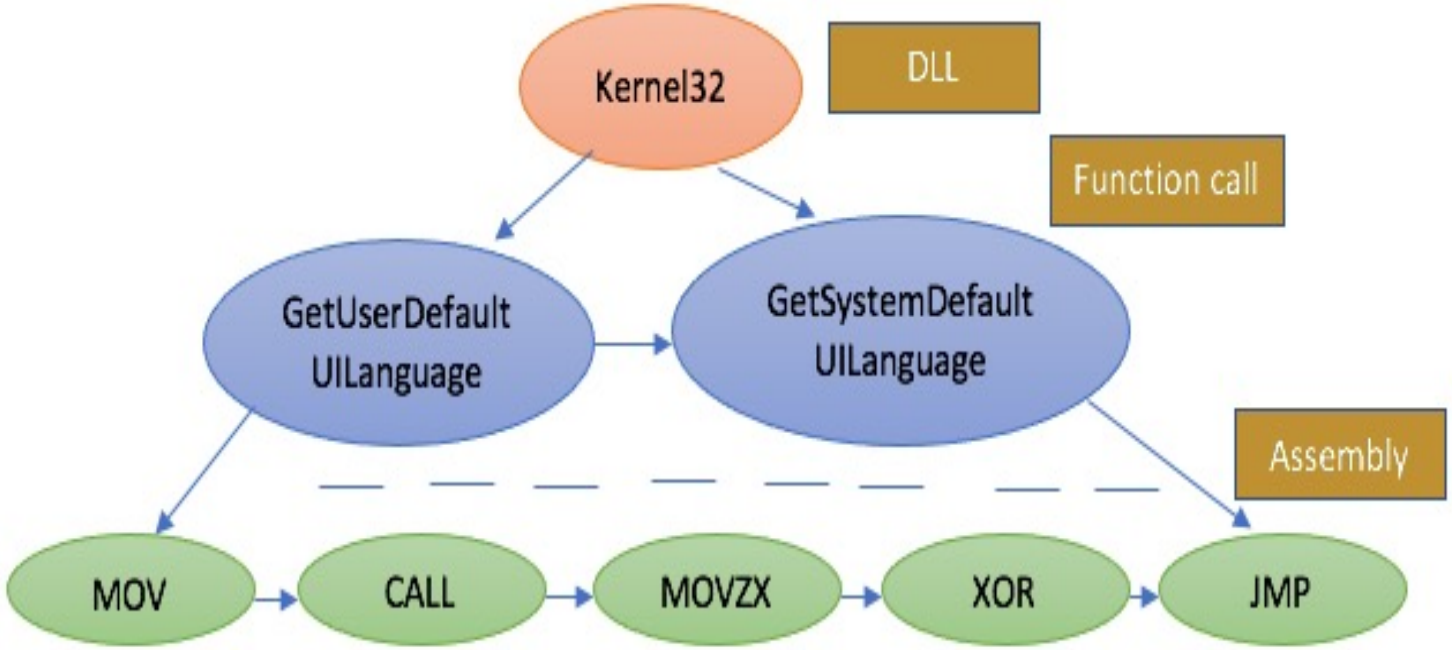
- GetModuleFileNameW gets the malware executable location while the function wsprintfW writes the previously obtained command line parameter to buffer.
- The ShellExecuteW function via lpFile parameter value as cmd.exe executes the given command.

DLL (Level 1)	Function (Level 2)	Assembly (Level 3)
kernel32, user32, shell32	GetModuleFileNameW, wsprintfW, ShellExecuteW	push, mov, push-5, call, push-4, mov, call, mov, test, jz, push-3, call, test, jz, push-3, call, add, push-6, call, push, call, int

Table: Chain L: Self deletion chain showing components at three levels

Behavioral chain for system profiling

- This 3-layered hierarchy is concerned with profiling system identifiers.
- Some of the often profiled system identifiers are keyboard layout, windows version used, the domain used, CPU identifier and so on.



C&C communication

- Differs among different ransomware families as some use hard-coded URL, some use domain generation algorithm, and the way to get the victim's IP address also differs.
- InternetOpenWfunction opens the browser application, InternetConnectW opens a File Transfer Protocol(FTP) or HTTP session for a given site.
- May use ipv4bot.whatismyipaddress.com to find the victim's IP address Or find via command prompt.
- Connects to CC server via HttpOpenRequestW using the handle of InternetConnectW function.
- HttpAddRequestHeadersW specifies the CC server.
- InternetReadFile reads the data from a handle opened by the InternetOpenUrl, FtpOpenFile, or HttpOpenRequestfunction.
- Finally, InternetCloseHandle closes the internet handle

Encryption setup

- The CryptAcquireContextWfunction is used to acquire a handle to a key container implemented by either cryptographic service provider (CSP) or Next-generation CSP.
- The szProvider parameter specifies this information. Example: szProvider=”Microsoft Enhanced Cryptographic Providerv1.0”
- The CryptGenKeyg enerates a public/private key pair.
- The handle to the key is returned in parameter phKey.
- It has the Algid parameter which specifies the type of encryption algorithm being used.

Encryption setup contd…

- For example, Algid=0xa400 represents CALG_RSA_KEYX as the “RSA public key exchange algorithm”.
- The CryptExportKeyfunction exports a cryptographic key pair from a CSP in a secure manner.
- At the receiver end CryptImportKeyfunction should be used to receive the key pair into a recipient’s CSP.
- CryptDestroyKey destroys the encryption handle but not the keys.
- CryptReleaseContext releases the handle of a cryptographic service provider and a key container

Experimental Setup

- The dataset consisted of the binaries in a portable executable (PE) file format.
- 550 samples of ransomware were collected from VirusTotal[12] and 540 normal samples from the Windows 10 OS and open-source software.
- We experimented with coding in python, bash script, and the system with configuration Intel(R) Core(TM) i7-5500U CPU @ 2.40 GHz 2.39 GHz, 8.00 GB RAM, and 1 TB disk space.
- For running malware samples 6 virtual environments was set up with five i7 processor machines, one with 32 GB RAM and four with 8 GB RAM.

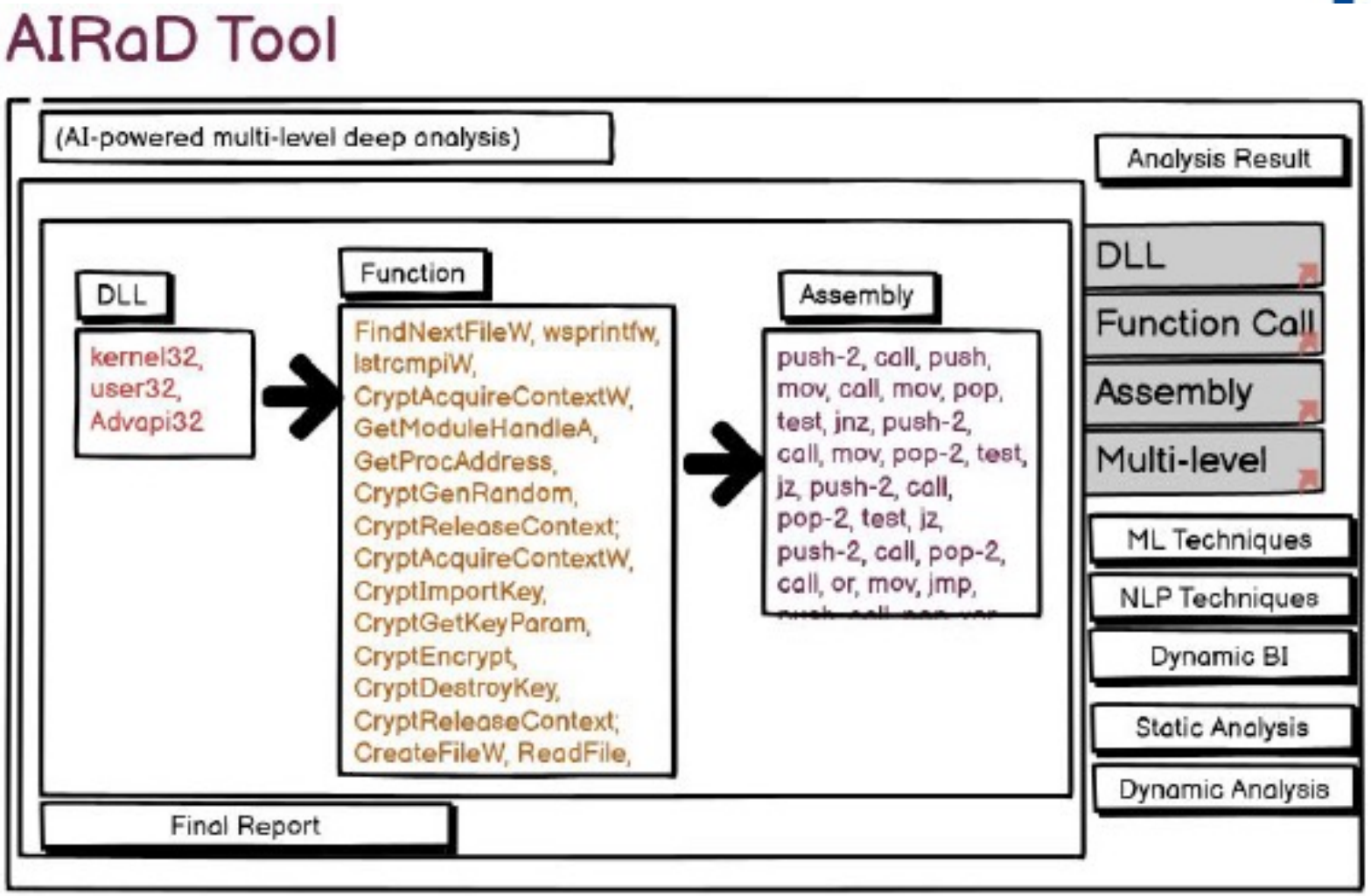
Evaluation of Multi-Level Chains with ML Classifiers

Machine learning Classifier	TPR	FPR	Precision	Recall	F-measure	Accuracy (%)
Logistic Regression	0.992	0.008	0.992	0.992	0.992	99.17
Support Vector Machine	0.995	0.005	0.995	0.995	0.995	99.54
Random Forest(RF)	0.990	0.010	0.990	0.990	0.990	98.99
J48	0.993	0.007	0.993	0.993	0.993	99.26
Adaboost with RF	0.987	0.013	0.987	0.987	0.987	98.71
Adaboost with J48	0.995	0.005	0.995	0.995	0.995	99.54

Table: Performance of Machine learning algorithms for multi-level behavioral analysis with term frequencies.

We achieve highest accuracy of 99.54% with low false positive rate of 0.005 for SVM (shown in row 3) and Adaboost with J48 (shown in row 7) among others.

- This tool leverages the techniques defined in our proposed framework and can also be considered an explanatory AI tool as it identifies the distinguishing behavioral chains which help to create a unique dataset for machine learning models.
- The current snapshot shows the multi-level mapping for file encryption activity.



Comparison with related works & references

Methodology used	Malware studied	Similarity and differences with our AIRaD	Reference/Year
Survey based on static, dynamic, file entropy and network features using different machine learning techniques.	User Different ransomware and malware families.	Multi-level analysis at 2 levels (Dll and assembly), which referenced our previous work. Another similar work used trap layer, backup layer, static analysis and so on. Correlation among three levels were not done.	**[5, 6]/2018 /2019**
Used the concept of cyber vaccine with autoencoders features using static and dynamic analysis	Malware families	The features are based on strings and API calls. The malware detection performance is very low (7.3% FPR). The multi-level correlation were not used in this work.	**[7]/ 2018**
Performed static analysis and behavioral analysis including network traffic analysis.	Mobile malware	They used multi-level classification methods to design their model. Their focus is on mobile malware and also lacks multi-level correlation.	**[8]/2017**

Comparison with related works & references (cont..)

Methodology used	Malware studied	Similarity and differences with our AIRaD	Reference/Year
Used both static and dynamic methods and used K-means clustering to find opcode frequency information.	Ransomware	Used API features with frequency analysis. The multi-level correlation were not used in this work.	**[9]/2017**
Used both static and dynamic methods for malware analysis.	Malware families	Mentions about DLL, function call and assembly but in an informative context only. The dynamic analysis claims that any results (e.g., events), activities, and/or decision makings of all of the components may be recorded. The multi-level correlation were not used in this work.	**[10]/2018**
Used both static and dynamic method for malware analysis.	Malware families	Though the work used static and dynamic analysis techniques but lacks multi-level analysis and behavioral chain analysis.	**[12]/2020**
Uses both static and dynamic method for malware analysis.	Malware families	Analyzed and classified malicious code segments using hybrid analysis along with n-gram analysis of Dlls. They did not mention the multi-level code analysis at function call and assembly level and the use of AI.	**[11]/2019**

AIRaD potential for Commercialization

- AIRaD tool can assist malware detection researchers and analysts at security and anti-virus companies.
- Code analysis process (Identification, feature analysis, standard signature (Yara) creation and intelligent interpretation) can be automated by this tool.
- AI component helps to automate the task of identifying new zero-day exploits and malware.
- This can be used as a stand-alone malware monitoring, analysis and detection tool or in combination with others.
- The AIRaD tool will be the first of its kind demonstrating the analysis and detection with explainable AI, leveraging advances in reverse engineering and application of AI techniques.

References

1. Poudyal, Subash, Dipankar Dasgupta. "A multi-level ransomware detection framework using natural language processing and machine learning." *14th International Conference on Malicious and Unwanted Software" MALCON*. November 2019.
2. Poudyal, Subash, Kul Prasad Subedi, and Dipankar Dasgupta. "A framework for analyzing ransomware using machine learning." 2018 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, December 2018.
3. Poudyal, Subash, and Dipankar Dasgupta. "AI-Powered Ransomware Detection Framework." *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, December 2020.
4. Poster presentation in "CAE in Cybersecurity (CAE-C) Symposium on November 19-20, 2020"
5. Damien Warren Fernando, Nikos Komninos, Thomas Chen, "A Study On The Evolution Of Ransomware Detection Using Machine Learning And Deep Learning Techniques" 2018, NPL1
6. Daniel Gibert, Carles Mateu, Jordi Planes, "The Rise Of Machine Learning For Detection And Classification Of Malware: Research "Developments, Trends And Challenges" 2019 NPL2
7. Michael Howard, Avi Pfeffer, Mukesh Dalal, Michael Reposa, "Cyber Vaccine And Predictive-malware-defense Methods And Systems" 2018
8. Theodora H. Titonis, Nelson R. Manohar-alers, Christopher J. Wysopal, "Automated Behavioral And Static Analysis Using An Instrumented Sandbox And Machine Learning Classification For Mobile Security"2017
9. Ransomware Dectecting Method And Apparatus Based On Machine Learning Through Hybrid Analysis, 2017
10. Michael Vincentali Mesdaqemmanuel Thiouxabhishek Singhsai Vashisht, "Dynamically Adaptive Framework And Method For Classifying Malware Using Intelligent Static, Emulation, And Dynamic Analyses", 2018
11. An Apparatus And Method For Detecting Malicious Codes Using AI Based Machine Running Cross Validation Techniques,
12. Dhankha, S. R., Saldanha, A. W., & Mohanta, A. (2020). U.S. Patent Application No. 16/130,816., 2020

Thank you!

Q & A?