STOPPING SCRIPT AND FILELESS ATTACKS USING AMSI ML MODELS IN REALTIME
WHO AM I

Ankit Garg

Security Researcher @Microsoft Defender Research
AGENDA

- Script Based and Fileless Attacks
- Introduction to AMSI, How it is helpful in stopping attacks.
- How Microsoft Defender client and cloud integration works.
- Client and Cloud Based ML Models.
- Case Study from the ML Models Blocks
SCRIPT BASED ATTACKS

• Initial Attack Vector (Macro Documents, Attachments having various scripts)
• Downloading further Payloads.
• Executing Fileless Payloads and doing Persistence.
• Advance Frameworks like Powershell Empire, kodiac and many more.
EMOTET DOWNLOADER USING VBA MACRO

Good afternoon,

In view of your payment documents for the period from September to December, please find a list of uncleared bills and payment details enclosed.

Best Regards,

Denis Rempel - Stonebridge Financial

Attachment: Ft_0_5556034.doc (645 KB)
TRICKBOT CAMPAIGN

Trickbot Downloaders using various obfuscation

Very Hard to Scale manually. So we need some automation!!
We need to protect the first customer

96% of malware are seen only once

Malicious downloads encountered more than once

Source: Windows Defender Antivirus

ML is needed to scale and for pro-active response
AMS1 is an open interface that allows antivirus solutions to inspect script behavior and content on execution.

It support following scanning:

- File Based
- Memory Based
- Stream Scanning
- content url/ip
WINDOWS 10 SCRIPT EXECUTION ENGINES HAVE AMSI INTEGRATIONS

2015

Windows 10 Launch
- PowerShell
- VBScript
- JScript
- UAC

```
private void ReallyCompile(bool optimize)
{
    var br = new Context();
    if (optimize)
    {
        br.private void PerformSecurityChecks()
    }

    $executionContext.InvokeCommand.InvokeScript(…)
    $executionContext.InvokeCommand.NewScriptBlock(…)
    $executionContext.InvokeCommand.ExpandString(…)
}
```

Scans content at PowerShell script compile times. Includes dynamically-loaded content
<table>
<thead>
<tr>
<th>Year</th>
<th>Windows 10 Script Execution Engines Have AMSI Integrations</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015</td>
<td>Windows 10 Launch</td>
</tr>
<tr>
<td></td>
<td>- PowerShell</td>
</tr>
<tr>
<td></td>
<td>- VBScript</td>
</tr>
<tr>
<td></td>
<td>- JScript</td>
</tr>
<tr>
<td></td>
<td>- UAC</td>
</tr>
<tr>
<td>2017</td>
<td>Windows 10 Fall Creators Update</td>
</tr>
<tr>
<td></td>
<td>- JScript behavior instrumentation</td>
</tr>
<tr>
<td></td>
<td>- VBScript behavior instrumentation</td>
</tr>
<tr>
<td>2018</td>
<td>Office 365 VBA macro behavior instrumentation</td>
</tr>
<tr>
<td>2019</td>
<td>WMI behavior instrument</td>
</tr>
<tr>
<td>2021</td>
<td>XLM4.0 Macro</td>
</tr>
</tbody>
</table>
AMSI BEHAVIOR COM OBJECT CALL LOGGING
AMSI SCRIPT INSTRUMENTATION

Instruments COM objects
• Logs when COM objects and its methods are invoked along with parameters
• Calls AMSI synchronously prior to all executes

• WScript.Shell.Run()
• Shell.Application.ShellExecute()
• Wscript.Shell.Exec()
• MMC20.Application.Document.ActiveView.ExecuteShellComm and
• Execute (generic "object.Execute", often used with the obj returned by "WSHController.CreateScript")

• Aborts script behavior execute if detected by AV product
Before Run() executes, will call AMSI scanner with whole buffer contents
No AMSI behavior call, no execute scan trigger.
WE NEED BETTER SOLUTION TO STOP THESE ATTACKS!
WINDOWS DEFENDER CLOUD

Customer’s Machine

Defender Client

Sends features

Scan/behavior

Decision (Malware/Clean?)

Sends features:
- Emulated behavior (eg api calls)
- Fuzzy hashes
- ML feature vectors
- Process behavior events
- Etc

Realtime ML classifiers

Decider rules and ranking logic

Windows Defender Cloud

Global file information

SmartScreen

O365

ATP

+more!
PROBLEM: IT IS TOO COSTLY TO ASK THE CLOUD FOR EVERY FILE WE SCAN

Script behavior or dynamic content
BUILDING THE ML MODELS

1. Collecting the data
2. Featuring the data
3. Labeling the data
4. Choose right ML Algo for prediction
5. Deploying the Model
STEP1 : COLLECTING DATA

We uses following data source for our AMSI Models :

• RealTime Telemetry

• Sandbox Detonation Data

• Data from third party like VT,RL
STEP 2: SELECTING FEATURES
CLIENT FEATURE VECTORS

Sends features:
- Emulated behavior (e.g., API calls)
- Fuzzy hashes
- ML feature vectors
- Process behavior events
- Etc

Decider rules and ranking logic

Windows Defender Cloud
- Global file information
- SmartScreen
- O365
- ATP
- AppRep
- +more!

Sandboxing and offline ML

Realtime ML classifiers

Scan/behaviour

Customer’s Machine

Defender Client
FEATURE SELECTION

Clean + Malicious scripts

AMSI behavior logs

COM objects referenced

COM functions referenced

COM function arguments

Normalization for COM versions

~64 COM object name features

~193 COM function name features

Character ngram features
Example set of learned features used to help in classification of malicious AMSI content

ML-selected ngram features

<table>
<thead>
<tr>
<th>COM Objects</th>
<th>COM Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>IHost</td>
<td>CreateObject</td>
</tr>
<tr>
<td>IWshShell</td>
<td>ExpandEnvironmentStrings</td>
</tr>
<tr>
<td>IFileSystem</td>
<td>FileExists</td>
</tr>
<tr>
<td>IXMLHTTPRequest</td>
<td>open</td>
</tr>
<tr>
<td></td>
<td>send</td>
</tr>
</tbody>
</table>
The dynamic script content loaded with `eval()` will also be evaluated and classified.
MAP-REDUCE FOR INITIAL DOWNSELECTION

COM function arguments
- X MB
- X AMSI Logs
- X documents

Character Ngram: Regular and lowercase
[4, 5, 6, 7, 8, 10, 14]

Minimum count

Distinctness of span

Initial feature candidate set

X feature candidates

X GB
XM distinct feature candidates

X feature candidates

~100K features is too many for performance reasons
Fast Linear SA-SDCA used to downselect features:

• Semi-Asynchronous Stochastic Dual Coordinate Ascent
• Downselection through L1 regularization feature trimming
• X final function-argument features


WHY THE CLOUD? WHY NOT THE CLIENT ML?

- Global file information (file age, prevalence)
- More costly features (features based on fuzzy hashes)
- More costly models (using more memory, large disk space, high CPU usage)
- Quickly updating ML models to respond to adversaries
- ML models are not in the hands of the adversaries
- Clean reputation models
- Quickly fixes the FP/FN
STEP3: LABELLING THE DATA

We uses 2 approaches to generate Labels:

1. Labelling Buffers based on Caller Files

2. MetaLables
LABELLING : METHOD 1

Label the AmsiBuffers as Malicious if Mal.js is Malicious.
LABELLING : METHOD 2 - METALABELS

Assumptions:

• Attacks start with a malicious file or involve a malicious file at some point.
• We have malicious file labels in retrospect sometimes
  - 0 Known clean file **first seen on device**
  - 1 Known malware file **first seen on device**
  - ? Hash (Unknown behavior) **first seen on device**

Machine Timeline (In Retrospect)

Encountered malware

Can we train this behavior as malware?
• Clearly associated with malware first arriving on this device?
• What about the reputation of this similar behavior on other devices?
INFERRING REPUTATION FROM MULTIPLE MACHINES

Aggregate reputation per behavior across machines as:
- % of time machine first-encountered malware within 1 minute of first-seeing this behaviour
- % ... within 5 minutes
- % ... within 15 minutes
- % ... within 2 days

Problem: Behavior hashes are often unique per machine if it has any user-data. So you can't make inferences from the reputation of multiple machines easily.
In addition to reputation of exact match, we build reputation of similar matches.

<table>
<thead>
<tr>
<th>Key</th>
<th>% malware within 1 minute</th>
<th>% malware within 5 minutes</th>
<th>% malware within 15 minutes</th>
<th>% malware within 60 minutes</th>
<th>% malware within 2 days</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact match</td>
<td>100%</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File name</td>
<td>70%</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy hash0</td>
<td>99%</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy hash1</td>
<td>1%</td>
<td>20%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy hash2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy hash7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Writing a rule using these noisy generic features isn’t the best. ML!
## COMBINING GENERIC FEATURE REPUTATIONS WITH ML

In a subset of AMSI scenarios we directly tie BehaviorHash to FileHash of known malware.

Really healthy machines unlikely to have encountered malware.

**nx floats describing reputation of BehaviorHash and its generic features**

<table>
<thead>
<tr>
<th>Key</th>
<th>% malware within 1 minute</th>
<th>% malware within 5 minutes</th>
<th>% malware within 15 minutes</th>
<th>% malware within 60 minutes</th>
<th>% malware within 2 days</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact match</td>
<td>100%</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File name</td>
<td>70%</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy hash0</td>
<td>99%</td>
<td>100%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy hash1</td>
<td>1%</td>
<td>20%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fuzzy hash7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LightGBM**

**MetaLabel** (probability malware based on the reputation of machines that encountered this and similar buffers)
STEP4 : MODEL SELECTION

CLIENT ML Models LR(Logistic Regression) perform the best one and for CLOUD Averaged Perceptron!!!!
CURRENT ML MODELS RUNNING

1. PowerShell
2. JavaScript
3. VBSCRIPT
4. WMI
5. VBA
CASE STUDY FROM SOME OF OUR BLOCKS
Case Study 1: TrickBot Banking Trojan Campaign

It starts with an email having a .docm file as an attachment. Further downloading a JavaScript payload.

Subscription XX is fully covered.
Good luck.

Member Subscription Info F871371.docm

Member Subscription Info F871371.dat
CLOUD MODEL RESULTS

Client

Member_Subscription_1.docm

Static and dynamic feature extraction

101010
010101
101010

Cloud

101010
010101
101010

Client ML Verdict: Suspicious

Cloud ML Verdict: Malware

Real time ML Models
Key Takeaways

- In last few years there is big shift from PE based to Script Based and Fileless Attacks.
- Integration of AMSI with various scripting engines help in getting behavior instrumentation of obfuscated Scripts.
- ML is really helpful tool for identifying patterns in the large dataset.
- Combination of Client + Cloud Models works really great.
Thanks to our contributors

- Geoff MacDonald (Microsoft Defender ATP)
- Hamish O Dea (Microsoft Defender ATP)
- Andrea Lelli (Microsoft Defender ATP)
THANK YOU!!

• Twitter: Ankit (@a00rs)
• Linkedin: ankit-garg-73a32077/