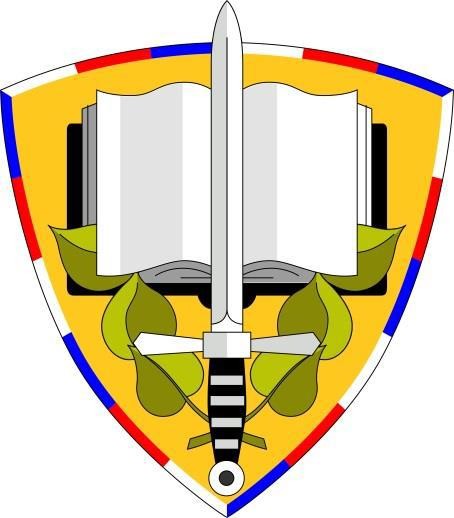
UNIVERSITY OF DEFENCE IN BRNO

**FACULTY OF MILITARY TECHNOLOGY**



**DEVELOPMENT OF MACRO MALWARE FOR PHISHING**

|  |  |
| --- | --- |
| Author: | Quy Tai DANG, 22-5KB-C |
| Supervisor: | plk. gšt. doc. Ing. Petr Františ, Ph.D. |

BRNO 2022

# ABSTRACT

Since the term "phishing" first appeared in the 1990s, it is by far the most common attack performed by cybercriminals. Phishing exploits "social engineering" to psychologically manipulate the victim into clicking on a link or opening a malicious attachment, etc., to steal the victim's sensitive information. Over time, phishing tactics have become increasingly sophisticated. Macro malware has also grown in popularity over the years. It easily conceals itself in Microsoft Office files and is spread via email attachments or ZIP files.

By combining phishing and macro malware, the project aims to develop and program a functional malware demonstrator that spreads the content of documents and e-mail attachments. Analysis of related subjects, formulation of hypotheses, and implementation of specific experiments are the research methodologies used. The project simulated a real-life phishing attack with two main factors: malware macro and the hacker's server. Macro Malware was developed on Microsoft Excel using Visual Basic for Applications (VBA) and Visual Basic Script (VBS). The functionalities of a malware macro include hiding in an attachment, running automatically on opening files, setting up a remote connection with the hacker's server, and hiding communications with the remote server on the victim's computer. The hacker's server is developed in Python and uses the Socket Programming Method to directly control the victim's computer through Malware Macro. This project focuses on the development of Macro Malware, which exploits functionalities provided by Microsoft Office Applications in a "malicious" manner. This has made anti-virus software detection and warning extremely difficult. The aim of this project is also to help readers understand the process lifecycle of a phishing attack with Malware Macro, which in turn will increase the awareness of these phishing attacks and the techniques being used.

**KEYWORDS**: Phishing. Macro malware. Attack Surface Reduction (ASR). Web Scraping with VBA Excel. Socket Server Python. Connect and Command Server (C&C). Run in the background.

TABLE OF CONTENT

[ABSTRACT 2](#_Toc102554349)

[LIST OF ABBREVIATIONS 4](#_Toc102554350)

[I. INTRODUCTION 6](#_Toc102554351)

[II. BACKGROUND KNOWLEDGE 7](#_Toc102554352)

[1. Macro VBA event 7](#_Toc102554353)

[2. Common Malware Macro Techniques 7](#_Toc102554354)

[III. BUILD MACRO 9](#_Toc102554355)

[1. Macros are automatically executed 9](#_Toc102554356)

[2. Programming the Windows Script Host Shell (WshShell object) 9](#_Toc102554357)

[3. Bypass ASR. 10](#_Toc102554358)

[a. ASR - Block all Office applications from creating child processes 10](#_Toc102554359)

[b. Demo Excel-Outlook: Bypass ASR by COM 10](#_Toc102554360)

[IV. CONNECT AND COMMAND CHANNEL 11](#_Toc102554361)

[1. Reverse Shell 11](#_Toc102554362)

[2. Web Scraping with VBA Excel – “SendtoServer()” Function 11](#_Toc102554363)

[3. Connect and Command Server 13](#_Toc102554364)

[4. Encrypt data 15](#_Toc102554365)

[V. MAINTAIN MACRO WITH LNK SHORTCUT 16](#_Toc102554366)

[VI. LIMITATION OF MALICIOUS MACRO FOR PHISHING DEVELOPMENT 19](#_Toc102554367)

[1. Social engineering 19](#_Toc102554368)

[2. Living off the Land (LotL) 20](#_Toc102554369)

[3. Gmail 20](#_Toc102554370)

[4. Phishing on social network, SMS, etc 21](#_Toc102554371)

[5. Microsoft Antimalware Scan Interface (AMSI) 21](#_Toc102554372)

[VII. CONCLUSION 22](#_Toc102554373)

[VIII. APPENDIX 23](#_Toc102554374)

[IX. REFERENCES 24](#_Toc102554375)

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| Abbreviation | Meaning |
| AMSI | Antimalware Scan Interface |
| ASR | Attack surface reduction rules |
| C&C, C2 | Command and Control Server |
| COM | Component Object Model |
| DDoS | Distributed Denial of Services |
| HTTP | Hyper Text Transmission Protocol |
| LNK | Link - Shortcut File Extension (Windows) |
| LOLBins | Living Off The Land Binaries and Scripts (and now also Libraries) |
| TCP | Transmission Control Protocol |
| VBA | Visual Basic for Applications |
| VBS | Microsoft Visual Basic Scripting Edition |
| WshShell | Windows Script Host Shell |

**LIST OF FIGURES**

[Figure 1 Component Object Model 8](file:///D:\Stuff\Final-STČ_DANG-QUY-TAI.docx#_Toc102554376)

[Figure 2 The "Hello World" MessageBox can auto open when opening the Excel file. 9](file:///D:\Stuff\Final-STČ_DANG-QUY-TAI.docx#_Toc102554377)

[Figure 3 WshShell opens PowerShell and executes “ipconfig” being detected as malicious 10](file:///D:\Stuff\Final-STČ_DANG-QUY-TAI.docx#_Toc102554378)

[Figure 4 Excel creates child process PowerShell (from Task Manager on Windows) 10](#_Toc102554379)

[Figure 5 Nothing is running underneath Excel 11](#_Toc102554380)

[Figure 6 Outlook spawn the PowerShell, child processes are under Outlook 11](#_Toc102554381)

[Figure 7 SendtoServer() Function 12](#_Toc102554382)

[Figure 8 C2 Server 13](file:///D:\Stuff\Final-STČ_DANG-QUY-TAI.docx#_Toc102554383)

[Figure 9 HTTP Start - initial POST Request with Data = "START" is sent 13](file:///D:\Stuff\Final-STČ_DANG-QUY-TAI.docx#_Toc102554384)

[Figure 10 TCP and HTTP calc 14](file:///D:\Stuff\Final-STČ_DANG-QUY-TAI.docx#_Toc102554385)

[Figure 11 TCP and HTTP ipconfig 14](file:///D:\Stuff\Final-STČ_DANG-QUY-TAI.docx#_Toc102554386)

[Figure 12 HTTP ipconfig was encrypted 15](file:///D:\Stuff\Final-STČ_DANG-QUY-TAI.docx#_Toc102554387)

[Figure 13 Summary of the attack scenario 16](#_Toc102554388)

[Figure 14 Protect VBA code from being viewed 16](file:///D:\Stuff\Final-STČ_DANG-QUY-TAI.docx#_Toc102554389)

[Figure 15 Running Macro without opening Excel by VBScript 17](file:///D:\Stuff\Final-STČ_DANG-QUY-TAI.docx#_Toc102554390)

[Figure 16 Fake Excel and Excel Properties 18](file:///D:\Stuff\Final-STČ_DANG-QUY-TAI.docx#_Toc102554391)

[Figure 17 Bat file for opening Excel.exe and VBScript at the same time 18](#_Toc102554392)

[Figure 18 Gmail alerts the mail contains malicious attachment 20](file:///D:\Stuff\Final-STČ_DANG-QUY-TAI.docx#_Toc102554393)

[Figure 19 Macros from the Internet are blocked by default 21](#_Toc102554394)

# INTRODUCTION

A macro is a tool in office software such as Microsoft Office that allows you to automate operations and add functionality to forms, reports, spreadsheets, and other documents. This is accomplished by utilizing the VBA suite's utilities to exploit and control all of the document's aspects. For these features, macros unwittingly provide a way for bad actors (such as hackers) to incorporate harmful code (malware) and use it as a base for assaults on the user's machine.

Hackers typically utilize macros to run malicious code or download malicious malware and execute it on the user's PC using this approach. And the most prevalent method of dissemination is via email. When a user gets an email message with an attached text containing concealed harmful code, when the user opens the attached text, the code in the macro will be run on the user's PC. These executables can encrypt the user's computer, hijack and transform the user's computer into a ghost machine participating in DDoS assaults, and so on, depending on the objective of each attack. Although the auto-run function of macro code has been removed from newer versions of Microsoft Office software, this is still one of the most prevalent assaults on email users in recent years, including fooling the user into triggering the macro code.

There are now various ways to avoid, identify, and delete macro harmful codes in text files in order to prevent malware and ensure the safety of systems in general. Hackers, on the other hand, utilize a variety of tactics and strategies to get around such software, inflicting harm to users' devices.

In this research, I have covered some popular macro malware strategies before delving into the way of combining multiple techniques to create a Phishing assault using Malware Macro. I evaluate the phishing assault from both the standpoint of a hacker and the perspective of a victim in order to have a better understanding of the problem.

The collected results can be used to construct macro malware, as well as to demonstrate how to carry out a phishing attack with Macro, as well as to identify limitations and obstacles that must be overcome in order to produce macro malware in the future. The report also provides users a better understanding of text transmission and how to recognise and avoid macro malicious code dangers.

# BACKGROUND KNOWLEDGE

## Macro VBA event

Macros are developed using VBA - an implementation of Microsoft's Event-Driven Programming language.

In VBA, an event is defined as a user activity in a Microsoft Application that can trigger the execution of a specific macro. In Microsoft Applications, there are many types of objects that may be associated with various events. For example, Excel events include events at the application, workbook, sheet, and chart levels, etc. along with specific events: open workbook, create new sheet, modify cells in sheet, change sheet color, and so on. VBA events allow users to create macros that are executed automatically when a specific event occurs. They improve the user experience and by adding interactivity between objects and between Microsoft Office Applications.

Hackers may completely exploit the events that are triggered in a specified order and at specific times to develop a real-life attack, they can analyze which event works best in their scenario and use the appropriate macro techniques.

## Common Malware Macro Techniques

Microsoft office documents are now widely utilized in the flow of information between numerous people (individuals, corporations, etc.), making this one of the most prevalent attacks today. When a target gets a phishing email with a malicious attachment, a phishing assault using macro malware begins. The victim then downloads the document to his computer, opens it, and activates Macro. Malicious macros will conduct malicious operations on the computer once they are enabled. In the sub following parts, I have demonstrated frequently applied Macro methods

* Combine Macros with Shell Objects and LOLBins

Macros are created in the VBA language and includes extensive capabilities that allow for deep system access. Based on this ability, an attacker will create scripts that simply download and execute the downloaded payload from the Office process itself. Shell Objects and different LOLBins[[1]](#footnote-1) will frequently be used in the snippets. Blocking all LoLBins is not possible in most environments since they are also required by legitimate processes.

* Evading Parent/Child Analysis

Strange processes will be launched as child processes of Microsoft Office applications. Based on this abnormality, advanced defensive systems can quickly detect and alert. Therefore, attackers will often devise strategies to execute payloads by evading detection through process generation techniques from other system processes other than Microsoft Office.

COM is a popular approach for Evading Parent/Child Analysis. In Windows, COM components as well as COM interfaces are assigned an ID. All of these IDs, together with information on the components or interfaces they represent, are stored in the Windows registry. COM empowers components to communicate with one another via the operating system.

Diagram

Description automatically generated

Figure Component Object Model

COM allows Software A to access functionality from Software B. The entity which holds the desired functionality is referred to as the COM server and the entity which is requesting the functionality is the COM client, even though these are two software components running on the same machine.

By this capability, it may be entirely abused to insert and perform malicious code as a valid program, allowing harmful malware to remain active. An attacker can utilize VBA to refer to any COM object and leverage its capabilities.

* Scheduled Task Creation and Registry Modification

Once a computer has been infected with malware, an attacker will generally try to determine a means to keep the Macro running in the background. There are several approaches, the most frequent of which being Setting a Scheduled Task and tampering with the Registry. The attacker will use Schedule Task to construct tasks that will execute on the attacker's preferred schedule. An attacker can use VBA code to get access to the Registry, where they can store payloads, change system settings, and create persistence points (to ensure that the malicious code can be relaunched even if the user restarts the system).

* Embed payload/binary directly within the macro/document itself

One of the most typical methods is to execute the payload included in the document on the system. This approach has both advantages and disadvantages. When the payload is written to the disk, it will be evaluated by AVs and tracked on the system. However, this strategy is still useful for the attacker in swiftly dominating the victim.

* Download Content from outside

Rather of embedding the payload in the document, as described above, the attacker can use the system's functions to download the payload from the external environment for execution. Some code examples demonstrate how to utilize the XMLHTTP library in conjunction with ADODB to write files; access the system's API directly, such as URLDownloadToFileA, or use the built-in tools to conduct payload download.

An assault is actually a mix of several approaches targeted at hijacking, obfuscation for malware, and malware maintenance. I was able to exploit the fantastic capabilities of Excel in a malicious way by combining the macro malware tactics indicated above, as well as focusing on abusing the Living off the Land strategy.

# BUILD MACRO

## Macros are automatically executed

**Graphical user interface, application, table, Excel

Description automatically generatedset the to "hen the Work B.nk is 
Sub Korkbookgen( ) 
YsgBox ("Hello "orld")**Workbook Open() is an Excel's event that is part of the workbook's events level. It causes the Macro to be executed automatically when the Workbook is accessed by the User.

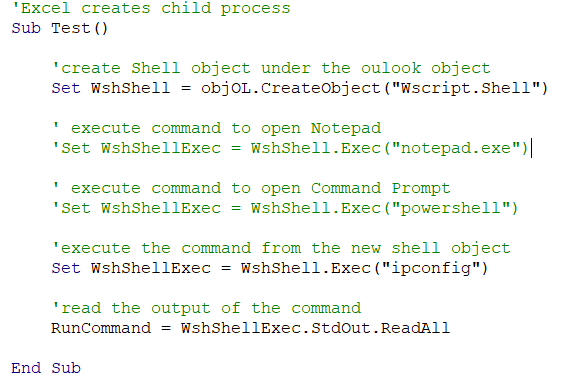
Figure The "Hello World" MessageBox can auto open when opening the Excel file.

## [Programming the Windows Script Host](https://www.informit.com/articles/article.aspx?p=1187429) Shell (WshShell object)

WshShell is a generic name for a powerful object that enables you to query and interact with various aspects of the Windows Shell. It can display information to the user, run applications, create shortcuts, work with the Registry, control Windows' environment variables and access the object's properties and methods. Performing this task does not require to have administrative credentials. Therefore, as a security best practice, consider performing this task as a user without administrative credentials. WshShell refers to the Shell object exposed via the Automation interface of WScript. Therefore, we must use CreateObject to return this object:

*Set objWshShell = WScript.CreateObject("WScript.Shell")*

I have utilised the Shell Object to execute some commands such as open the Notepad application, PowerShell, run the Exec("ipconfig") command.  Everything worked properly in the Notepad, but when I changed to PowerShell and ipconfig, the Window Defender displayed a security alert and blocked the Excel file. The results are shown in the figure below.

Graphical user interface, text, application

Description automatically generated



Figure WshShell opens PowerShell and executes “ipconfig” being detected as malicious

## Bypass ASR.

### ASR - Block all Office applications from creating child processes

Microsoft Defender Antivirus Exploit Guard is a set of intrusion prevention capabilities that includes ASR rules, Exploit protection, Network protection, and Controlled folder access. The ASR rules are rules to lock down various attack vectors commonly used in malware and target certain software behaviors and “Block all Office applications from creating child processes” is a rule in the ASR

Return to example in III.2, Excel can open and close applications, particularly other Microsoft Office applications, using Macros. However, the act of accessing PowerShell through Excel is regarded suspicious because it violates a rule of ASR, which states that all Office programs should be prevented from spawning child processes, and Window Defender blocked this action.

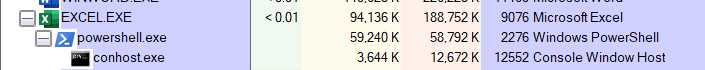


Figure Excel creates child process PowerShell (from Task Manager on Windows)

### Demo Excel-Outlook: Bypass ASR by COM

Evading Parent/Child Analysis is a popular Malware Macro approach that I mentioned above. To bypass ASR, I choose to employ the COM Excel and Outlook in my attack. Since the ASR rules stated that it would prevent office apps, which are Word, Excel, PowerPoint, OneNote, and Access, from creating child processes (Figure 5), it is still missing a very common application - Outlook. "Block all Office apps from generating child processes" is extremely difficult to do in Outlook because using other apps' features is a very common occurrence with an Email app like Outlook. For example, if we open an email in Outlook and it contains an excel file - a spreadsheet, Outlook can use Excel's convenient feature to display it directly in Outlook. It makes sense for Outlook to spawn another process. (Figure 6)

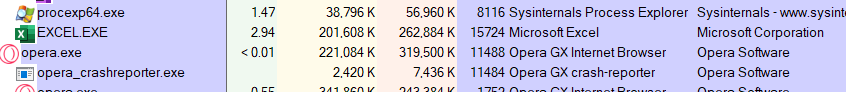
****

Figure Nothing is running underneath Excel

**IZCOUTLOOKEXE 
powersheILexe 
conhostexe 
@ScreenC 
O WUOFHostexe 
<001 
86512K 
50368 K 
3636 K 
1.700K 
7.816K 
139.832K 
58.468 K 
12848K 
12788K 
15664 Microsonoubook 
12216 Windows Powershell 
3964 Console Window Host 
1100 
Microsof Corporation 
Microsoft Corporation 
Microsoft Corporation**

Figure Outlook spawn the PowerShell, child processes are under Outlook

Programming with Window Script Host or interaction between Applications is "fertile ground" for hackers to exploit. Disabling all interactions to avoid being exploited seems impossible, it will disrupt the nifty features of Microsoft products and make them much more difficult to use.

In this section, Evading Parent/Child Analysis technique using COM to Bypass ASR was successfully employed. In addition, the "Living off the Land (LotL)" attack technique is demonstrated very clearly when attackers have exploited legitimate Microsoft Office Applications and system functions to develop malicious Macro.

# CONNECT AND COMMAND CHANNEL

## Reverse Shell

A reverse shell (also known as a connect-back shell) is a type of session shell that can connect from one server serving as a target to another server acting as a host. The target will then establish an outbound connection, while the host will listen. In this project, Malicious Macro will serve as Target and the hacker's Server will serve as Host.

My aim is to establish a C2 channel to remotely access the target and control the victim's computer. In other words, my target is to be able to login to the command line which is a shell and then execute commands.

But the firewall prevents unauthorized connections from the hacker's server to the victim's computer. However, it has no control over connections from within the system to the outside. A Request from the Macro to the outside (to a certain website) is something the firewall does not interfere with. Applying Reverse Shell is a nice approach to get around firewalls.

## Web Scraping with VBA Excel – “SendtoServer()” Function

"Web Scraping with VBA Excel" is a technique for extracting desired information from an HTML web page and saving it to a local file on your local PC. It aids in the transmission of important information from a web page. An example of a day trader collecting market data from a financial website into an excel spreadsheet using macros. Internet Explorer and basic HTTP XML requests are two typical methods for accomplishing this. I employed HTTP XML requests with a ServerXMLHTTP object in my project, which allows developers to use HTTP, GET, and POST methods.

Graphical user interface, text, application, email

Description automatically generated

Figure SendtoServer() Function

Function SendtoServer() has a parameter which is the Url of the Hacker's Host Server. This URL is of the form: http://<SERVER IP>:<SERVER PORT>. The initial POST Request with Data = "START" is sent. The Macro then got a response from the Host Server, which was saved in the variable ReplyTXT. This response is a command that the Hacker sends to the Macro then the Macro receives the command and executes it by using the RunCommand() function. The output of the RunCommand() function, as executed by the WshShell object, is overwritten in the Data variable and sent to Server in the second POST Request. The process of sending Request and receiving Response is in a While loop and will be executed continuously until the "STOP" command is received from the Server. As a result, the "Connect and Command" Channel has been established.

## Connect and Command Server

Text

Description automatically generated with low confidenceText

Description automatically generatedText

Description automatically generatedCommand and Control Server (C&C or C2) is the server attackers use to distribute and control malware on target systems. I built the C2 server in Python language using Socket Programming.

Figure C2 Server

Then I used WireShark[[2]](#footnote-2) to analyze the connection between the Server and the Macro. With the HTTP protocol is used to access and send the results to the Server, and the TCP protocol is used to establish a session between the Macro and the Server and send the command from Server to macro.

**Graphical user interface, text, application

Description automatically generated**

Figure HTTP Start - initial POST Request with Data = "START" is sent

“calc”: The command causes the app Calculator to appear on the victim's computer screen

**Text

Description automatically generated with medium confidenceA picture containing calendar

Description automatically generated**

Figure TCP and HTTP calc

A picture containing table

Description automatically generatedA picture containing table

Description automatically generated"ipconfig": Displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings.

Figure TCP and HTTP ipconfig

## Encrypt data

In Protocol HTTP we can see the body of ip configuration that we take from victim.

But as a attacker we want to avoid detection as much as possible. We don't want if someone can capture the packets, we don't want them to see this strange information going across the WireShark so we have to encrypt our data.

Base64 file encoding is a technique commonly employed in malware. The Base64-encrypted data is easy to detect and decode, and the output data is larger than the input data. Other algorithms which use crypto principles (such as XOR, multiple XOR, or combinations Base64 plus XOR, etc). In the context of this project, I only used Base64 as an example of one of the steps a hacker must take to create an attack: hiding the connection.

A picture containing table

Description automatically generated

Text

Description automatically generated

Figure HTTP ipconfig was encrypted

# MAINTAIN MACRO WITH LNK SHORTCUT

By exploiting Excel's legitimate ability "Web Scraping with VBA," a Connect and Command channel between the macro and the Server Socket is established. However, in order to maintain control of the victim's machine, we must keep the macro running in the background. To do this, I utilized the LNK shortcut to build a Fake Excel on the target Computer. When the victim launches Fake Excel, the Excel.exe interface appears and the Malicious Macro is launched. Along with that, the Living off the Land technique is still the primary source of inspiration in this section.

**FAKE 
BAT 
Shell Object 
POST 
Reply 
Maliciuos Macro 
C2 channel 
S— fig 
Hacker's Server 
guyrai dang@mob.cz **

Figure Summary of the attack scenario

Just like you can set password to protect workbooks and worksheets, you can password protect a macro in Excel from being viewed

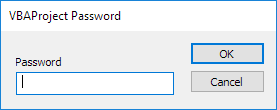
****

Figure Protect VBA code from being viewed

The VBScript file allows us to execute a Microsoft Office file at a defined location. The VBScript code: *objExcel.Visible = False*, which aids in hiding the Excel window. As a result, when this VBScript file is executed, our Macro will run in the background on the victim's system.

Text

Description automatically generatedGraphical user interface, application, qr code

Description automatically generated

Figure Running Macro without opening Excel by VBScript

There are two ways to create VBScript and bat files on the victim machine. The first method is to employ the Dropping file technique, which is a common technique of the Malware macro that I described above. VBA allows Macros to send a Get Request to the Server, receive a Response, download those two files, and save them in a certain location on the victim's machine. However, Anti-virus software will identify this method since downloading a .bat file, .exe file, etc. from the outside is a very suspicious behavior of Macro.

The second method is to create those files directly on the victim's computer. Continuing to make use of Macros' legal features, VBA allows us to create a file on the victim's computer using FileSystemObject.

In Windows, LNK is a filename extension for shortcuts to local files. LNK file shortcuts allow users to quickly access executable files without having to navigate the program's entire path. Because LNK files provide a convenient alternative to opening a file, so threat actors can use them to create script-based threats.

The VBA Macro allows users to create a Desktop LNK ShortCut using the Windows Script Host (WshShell object). Hacker will construct a fake Excel with the criterion "as similar to the original as possible" in order to "phish" the victim.

Graphical user interface, text, application, email

Description automatically generated

**Graphical user interface, text, application, email

Description automatically generated**

Figure Fake Excel and Excel Properties

As we can observe in figure 16, the target of Fake Excel is a bat file and this bat file will open Excel.exe as well as run the VBScript file.

A picture containing graphical user interface

Description automatically generated

Figure Bat file for opening Excel.exe and VBScript at the same time

# LIMITATION OF MALICIOUS MACRO FOR PHISHING DEVELOPMENT

## Social engineering

Social engineering is a type of phishing. Social engineering is a sort of non-technical assault that focuses on changing human behavior rather than exploiting security flaws in technologies and systems.

The assault described in this study included a variety of Social engineering approaches, including a phishing email, phishing postings, phishing messages from social media, and so on. The intention is to deceive victims into downloading a Malicious Attachment, enabling Macro, and then creating a Fake Excel on their PC.

Email phishing may be a life-threatening condition. It forges government agencies, online payments, orders, and so on, and instills anxiety in users, or it may be a winning announcement, or it may pretend to be an acquaintance in order to deceive the user into downloading the malicious document.

The strategies used to dupe users into enabling macro are becoming more sophisticated. The first is misrepresent the Meaning of the Security Warning, in which the hacker attempts to persuade victims that the security warning means that the document has been protected in some way to protect its contents. "Private text has been buried to warn against the unauthorized person," for example. "Please enable macros to access protected document," or "This is a locked document, please select the option to enable content," click Enable Content. This is a brilliant technique of persuading people under the guise of security, which most users don't comprehend. When the substance of the paper has been disguised, the second tactic is to pique the victim's interest. The hacker would inform the victim that in order to read the full text of the document, the victim must agree to the conditions and click on the allow content icon for some security or legal reason.

Fake Excel is another technique used to exploit consumers' technical illiteracy. The attributes of the fake Excel are designed to be similar to those of the actual Excel. The file names will be arranged in a manner similar to that of a system file, for example, ShortCut Excel.bat, system32.bat, and so on. These files will then be saved in difficult-to-find locations with extremely lengthy addresses.

Because of the wide range of methods, items, and victims used to manipulate human behavior, the method of protection from Social engineering is limited to users' attentiveness and awareness of Social engineering. To avoid becoming a victim of social engineering, we should regularly update our equipment with network security information. Keep an eye out for obnoxious emails. It's worth looking out regardless of how tempting the call-to-action is. Check the attachments you download as well. If you see any unusual indicators, do not enable macros. Last but not least, make sure that your system's security features are up to date, and that you have credible anti-virus alert software installed.

## Living off the Land (LotL)

For security controls that do not monitor process behavior, protecting against abuse of LoLBins combined with fileless code is difficult. The parent-child connection of the launched processes, as well as abnormalities in network activity of processes that are not normally linked with network communication, can be used to detect misuse.

Bypass ASR with COM Object, Programming the Windows Script Host, Web Scraping with VBA Excel or Create file and Create Shortcut, etc are all inspired by the Living off the Land strategy when people abuse the legal functions of Microsoft software in a harmful way.

The best security is to prevent LoLBins from being executed by employing technologies like as Windows Defender Application Control. Microsoft developed a policy block file that prevents the execution of LoLBins that are not necessary on protected systems.

Unfortunately, as LoLBins are required by legal programs, blocking all of them is not practicable in most situations.

## Gmail

Gmail provides users of Google's Advanced Protection Program and Advanced phishing and malware protection with powerful tools to prevent Phishing attacks. According to a Google announcement at the RSA conference 2020 on the topic "Malicious Documents Trends: a Gmail Perspective," approximately 50 percent of emails sent to Gmail accounts are spam and phishing emails, but Gmail has achieved 99 percent accuracy in detecting them with the new detection algorithm utilising machine learning technology. The objective for Gmail is to improve existing detection capabilities using Al interpolation and sophisticated document analyzers in order to improve detection coverage and robustness to adversarial assaults.

**email 
Compose 
edule 
fategories 
%Ne+.'neetinO. 
Join a meeting 
Hangouts 
•S:Start a 
hi 
D Spam x 
dangquytai776617185@gmail.com 
This message seems dangerous 
It might contain attachments with a virus or malicious link. Avoid downloading or opening the attachment. 
Anti-virus warning — 1 attachment contains a virus or blocked file. Downloading this attachment is disabled. 
Macro_malware.xls... 
Blocked file 
sat, Apr 23, 12109 AM (1 day ago) **With the Malware Macro that I created above, Gmail alerted the receiver that the message seems dangerous



Figure Gmail alerts the mail contains malicious attachment

## Phishing on social network, SMS, etc

Social media sites are used by more than two-thirds of all internet users. People use Facebook, Instagram, Twitter, and other platforms to stay in touch with friends and family, keep up with current events, and connect with the rest of the world. Businesses use social media to keep customers up to speed on the latest product offerings and events, as well as to promote to and attract new customers. As a result, threat actors find social media to be an appealing venue for carrying out phishing assaults.

In contrast to Gmail, which has incredibly advanced features, social networking platforms such as Facebook, Instagram, etc. have essentially no safeguards in place to protect or notify users about suspicious postings or messages they receive. The best thing social networking platforms can do is encourage victims to report Phishing Posts and Phishing Messages.

## Microsoft Antimalware Scan Interface (AMSI)

Microsoft Defender for Office 365 is a security service that is meant to keep you safe while using Office 365. For years, Microsoft has used a connection between its AMSI and Office 365 to eliminate macro malware, including successful attempts to eliminate macro scripts written in VBA. When suspicious functions are called, AMSI logs the macro behaviour, which activates a scan by the security product and terminates macro execution when malicious activity is discovered by the security product.

AMSI has proven to be a significant barrier for hackers. Despite Microsoft's attempts to expand and strengthen ASMI, there are still several ways for hackers to circumvent it. By default, AMSI is deactivated for documents accessed from a trustworthy site. When the macro security settings are set to 'Enable All Macros,' AMSI is likewise deactivated. Finally, because AMSI connects with the Office application via a user-level DLL, it may be feasible to directly access/modify AMSI.

Malicious elements commonly get access to VBA macros in order to deliver malware and ransomware. The sophistication and risk of Macro-based phishing assaults is rising by the day. That is why, beginning in April 2022, Microsoft will begin blocking Visual Basic for application (VBA) macros obtained from the public internet by default across five of its most popular Office apps, in an effort to prevent malicious actors from tricking users into executing malware on their systems.

Social engineering, the Living off the Land approach, and the exploitation of social networking platforms may be the future of Phishing assaults with Malware macros. According to history, the ingenuity of hackers appears to have no bounds.

**Security risk banner about blocked macros with a Learn More button**

Figure Macros from the Internet are blocked by default

# CONCLUSION

In this essay, I outlined and advocated a variety of common macro malware tactics written in the VBA programming language and used in text documents created with Microsoft Office. Using a variety of tactics such as phishing, deceiving parent and child processes, Web Scraping VBA Excel, LNK file as Fake Excel, and so on. I created a realistic assault in the spirit of the Living off the land approach, which exploits lawful functions in a malicious manner.

The assault proved that these tactics can entirely outperform traditional interception and detection systems. Once the user's PC has been hijacked, sophisticated macro malware will stealthily capture user information and data, and, more dangerously, it will utilize the human machine. utilized as a "launching pad" for attacks against internal computers and associated devices.

Simultaneously, the report provides an overview of Phishing attack with Malware Macro: how hackers perform fraud using Social Engineering techniques, how text transmission media such as Gmail, Social network, and Microsoft deal with Phishing attack with Malware Macro, thereby demonstrating the limits and future development of Phishing attack with Malware Macro. Not only that, but the report may enhance user awareness and give critical information to identify and avoid threats.

# Text, letter Description automatically generatedText, letter Description automatically generatedAPPENDIX

# REFERENCES

* 1. GIPSON, Susan, 2022. *Web Scraping with VBA* [online]. Guru99. April 23, 2022. [Accessed April 5, 2022]. Available at: https://www.guru99.com/data-scraping-vba.html
  2. IBM, 2021. How sockets work [online] August 31, 2021. [Accessed April 5, 2022]. Available at: https://www.ibm.com/docs/en/i/7.1?topic=programming-how-sockets-work
  3. InformIT,2008. *Programming the Windows Script Host* [online] April 29, 2008. [Accessed April 8 2022]. Available at: https://www.informit.com/articles-/article.aspx?p=1187429&seqNum=5
  4. KUBECKA, Martin, 2021. *Weaponizing Malicious Macros* [online] github.io. May 12, 2021. [Accessed April 15 2022]. Available at: https://martinkubecka.-github.io/posts/general/malicious-macros/
  5. Microsoft, 2022. *Attack surface reduction rules reference* [online]. MICROSOFT. April 12, 2022. [Accessed April 5, 2022]. Available at: https://docs.microsoft.com/en-us/microsoft-365/security/defender-endpoint/attack-surface-reduction-rules-reference?view=o365-worldwide
  6. Microsoft, 2022. *Getting started with VBA in Office* [online]. MICROSOFT. January 22, 2022. [Accessed April 2, 2022]. Available at: https://docs.microsoft.com/en-us/office/vba/library-reference/concepts/getting-started-with-vba-in-office
  7. Rahmat Nurfauzi, 2022. *ASR Rules Bypass.vba* [online] [Accessed April 5, 2022]. Available at: https://gist.github.com/infosecn1nja/24a733c5b3f0e5-a8b6f0ca2cf75967e3
  8. REDTEAM, 2021. *Abusing LNK "Features" for Initial Access and Persistence.* [online]. github.io. May 28, 2021. [Accessed April 25, 2022]. Available at: https://v3ded.github.io/redteam/abusing-lnk-features-for-initial-access-and-persistence
  9. TODD, Adam, 2020. *Malicious macros for script kiddies* [online] TRUSTEDSEC August 4, 2020. [Accessed April 19, 2022]. Available at: https://www.trustedsec.com/blog/malicious-macros-for-script-kiddies/

1. Living-off-the-land binary refers to binary files that are provided by the operating system itself for legitimate purposes but are misused by malicious actors [↑](#footnote-ref-1)
2. Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. [↑](#footnote-ref-2)