PRACTICAL CYBER SECURITY FOR CYBER PRACTITIONERS (CS668)

Assignment 2
Group 12

231110008 - ARPIT NIGAM
210246 - AYUSH KUMAR
210298 - DANISH VASDEV

a)Approach to Mapping Finished Reporting to ATT&CK

The snippet from the document is:

Initial Access

When conducting phishing operations, UNC2970 engaged with targets initially over LinkedIn masquerading as recruiters.[1] Once UNC2970 contacts a target, they would attempt to shift the conversation to WhatsApp, where they would continue interacting with their target before sending a phishing payload that masqueraded as a job description.In at least one case, UNC2970 continued interacting with a victim even after the phishing payload was executed and detected, asking for screenshots of the detection.The phishing payloads primarily utilised by UNC2970 are Microsoft Word documents embedded with macros to perform remote-template injection[2] to pull down and execute a payload from a remote command and control (C2).[3] Mandiant has observed UNC2970 tailoring the fake job descriptions to specific targets.

Reference No.	ID	Tactic	Technique	Sub Technique	Explanation
1	T1566.003	Initial Access	Phishing	Spear Phishing via Service	
2	T1059.005	Execution	Command and Scripting Interpreter	Visual Basic	
3	T1105	Command and Control	Ingress Tool Transfer		

The snippet from the document is:

The C2 servers utilised by UNC2970 for remote template injection have primarily been compromised WordPress sites[4], a trend observed in other UNC2970 code families as well as those used by other DPRK groups. At the time of analysis, the remote template was no longer present on the C2, however following this phishing activity, Mandiant identified it beaconing to a C2 associated with PLANKWALK.[5]

…

The ZIP file delivered by UNC2970 contained what the victim thought was a skills assessment test for a job application. In reality, the ZIP contained an ISO file,which

> included a trojanized version of TightVNC that Mandiant tracks as LIDSHIFT. The victim was instructed to run the TightVNC application [6] which, along with the other files, are named appropriately to the company the victim had planned to take the assessment for.In addition to functioning as a legitimate TightVNC viewer, LIDSHIFT contained multiple hidden features. The first was that upon execution by the user, the malware would send a beacon back to its hardcoded C2 [7]; the only interaction this needed from the user was the launching of the program. This lack of interaction differs from what MSTIC observed in their recent blog post. The initial C2 beacon from LIDSHIFT contains the victim's initial username and hostname.

Reference No.	ID	Tactic	Technique	Sub Technique	Explanation
4	T1584.006	Resource Development	Compromise Infrastructure	Web Services	
5	T1071.001	Command and Control	Application Layer Protocol	Web Protocols	
6	T1204.002	Execution	User Execution	Malicious File	
7	T1071.001	Command and Control	Application Layer Protocol	Web Protocols	

The snippet from the document is:

> LIDSHIFT's second capability is to reflectively inject an encrypted DLL into memory.[8] The injected DLL is a trojanized Notepad++ plugin that functions as a downloader, which Mandiant tracks as LIDSHOT. LIDSHOT is injected as soon as the victim opens the drop down inside of the TightVNC Viewer application. LIDSHOT has two primary functions:system enumeration[9] and downloading[10] and executing shellcode from the C2.[11]
>
> LIDSHOT sends the following information back to its C2[12] :
> 1.Computer Name
> 2.Product name as recorded in the following registry key
> 3.SOFTWARE\\Microsoft\\WindowsNT\\CurrentVersion\\ProductName
> 4.IP address
> 5.Process List with User and Session ID associate per process

Reference No.	ID	Tactic	Technique	Sub Technique	Explanation
4	T1055.001	Defence Evasion	Process Injection	Dynamic-Link-Library Injection	
5	T1071.001	Discovery	System Information Enumeration	Web Protocols	
6	T1204.002	Execution	User Execution	Malicious File	
7	T1071.001	Command and Control	Application Layer Protocol	Web Protocols	

Defensive Recommendation

1.Compromise Infrastructure: Web Services

This technique cannot be easily mitigated with preventive controls since it is based on behaviours performed outside of the scope of enterprise defences and controls.

2.Phishing: Spear Phishing via Service

Mitigation	**Description**
Antivirus/Antimalware	Anti-virus can also automatically quarantine suspicious files.
Restrict Web-Based Content	Determine if certain social media sites, personal webmail services, or other services that can be used for spear phishing are necessary for business operations and consider blocking access if activity cannot be monitored well or if it poses a significant risk.
User Training	Users can be trained to identify social engineering techniques and spear phishing messages with malicious links.

Defensive Recommendation: Restrict the usage of social media sites that can be used for spear phishing and Scan any downloaded file or executable using Anti-Viruses.Train Users regarding Spear-Phishing Campaigns and their techniques.

3.Command and Scripting Interpreter: PowerShell

Mitigation	**Description**
Antivirus/Antimalware	Anti-virus can be used to automatically quarantine suspicious files.

Code Signing	Set PowerShell execution policy to execute only signed scripts.
Disable or Remove Feature or Program	It may be possible to remove PowerShell from systems when not needed, but a review should be performed to assess the impact to an environment, since it could be in use for many legitimate purposes and administrative functions. Disable/restrict the WinRM Service to help prevent uses of PowerShell for remote execution.
Execution Prevention	Use application control where appropriate. PowerShell Constrained Language mode can be used to restrict access to sensitive or otherwise dangerous language elements such as those used to execute arbitrary Windows APIs or files.
Privileged Account Management	When PowerShell is necessary, consider restricting PowerShell execution policy to administrators. Be aware that there are methods of bypassing the PowerShell execution policy, depending on environment configuration. PowerShell JEA (Just Enough Administration) may also be used to sandbox administration and limit what commands admins/users can execute through remote PowerShell sessions.

Defensive Recommendations:Restrict the execution of PowerShell scripts to known and approved applications.Implement application whitelisting policies to allow only authorised scripts to run.HIDS can be utilised as an advanced endpoint protection solutions that can detect and block malicious PowerShell activities.

4. User Execution: Malicious File

Mitigation	**Description**

Behavior Prevention on Endpoint	On Windows 10, various Attack Surface Reduction (ASR) rules can be enabled to prevent the execution of potentially malicious executable files (such as those that have been downloaded and executed by Office applications/scripting interpreters/email clients or that do not meet specific prevalence, age, or trusted list criteria). Note: cloud-delivered protection must be enabled for certain rules.
Execution Prevention	Application control may be able to prevent the running of executables masquerading as other files.
User Training	Use user training as a way to bring awareness to common phishing and spear phishing techniques and how to raise suspicion for potentially malicious events.

Defensive Recommendations:On Windows 10/Windows 11, enable Attack Surface Reduction (ASR) rules to prevent Visual Basic scripts from executing potentially malicious downloaded content.If the user is not completely aware of the files that they are dealing with, then the user should use antivirus to automatically detect and quarantine the files.

5.Deobfuscate/Decode Files or Information

This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features.

6.Hijack Execution Flow: DLL Search Order Hijacking

Mitigation	**Description**
Audit	Use auditing tools capable of detecting DLL search order hijacking opportunities on systems within an enterprise and correct them. Toolkits like the PowerSploit framework contain PowerUp modules that can be used to explore systems for DLL hijacking weaknesses. Use the program sxstrace.exe that is included with Windows along with manual inspection to check manifest files for side-by-side problems in software.

Execution Preventio n	Adversaries may use new DLLs to execute this technique. Identify and block potentially malicious software executed through search order hijacking by using application control solutions capable of blocking DLLs loaded by legitimate software.
Restrict Library Loading	Disallow loading of remote DLLs. This is included by default in Windows Server 2012+ and is available by patch for XP+ and Server 2003+. Enable Safe DLL Search Mode to force search for system DLLs in directories with greater restrictions (e.g. `%SYSTEMROOT%`)to be used before local directory DLLs (e.g. a user's home directory) The Safe DLL Search Mode can be enabled via Group Policy at Computer Configuration > [Policies] > Administrative Templates > MSS (Legacy): MSS: (SafeDllSearchMode) Enable Safe DLL search mode. The associated Windows Registry key for this is located at `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\SafeDLLSearchMode`

Defensive Recommendations: Ensure that applications and services have the minimum necessary permissions to access and load DLLs.Avoid running applications with elevated privileges unless necessary.As mentioned earlier,Implement application whitelisting to control which applications and DLLs are allowed to execute.

7. Hijack Execution Flow: DLL Side-Loading

Mitigation	Description
Application Developer Guidance	When possible, include hash values in manifest files to help prevent side-loading of malicious libraries.
Update Software	Update software regularly to include patches that fix DLL side-loading vulnerabilities.

Defensive Recommendations: Digitally sign DLLs to ensure the integrity and authenticity of the files.Configure systems to validate the digital signatures of DLLs before loading them.As mentioned earlier,Implement application whitelisting to control which applications and DLLs are allowed to execute.

8. Masquerading: Match Legitimate Name or Location

Mitigation	Description
Code Signing	Require signed binaries and images.
Execution Prevention	Use tools that restrict program execution via application control by attributes other than file name for common operating system utilities that are needed.
Restrict File and Directory Permissions	Use file system access controls to protect folders such as C:\Windows\System32.

Defensive Recommendations: Check the signature and hash values to ensure the integrity and authenticity of the file and DLL before loading.Maintain logs and check for DLLS with identical/similar names, to legitimate Files/DLLs, using Log analysis.As mentioned earlier,Implement application whitelisting to control which applications and DLLs are allowed to execute.

9.Command and scripting Interpreter: Visual Basic

Mitigation	Description
Antivirus/Antimalware	Anti-virus can be used to automatically quarantine suspicious files.
Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to prevent Visual Basic scripts from executing potentially malicious downloaded content.
Disable or Remove	Turn off or restrict access to unneeded VB components.

Feature or Program	
Execution Prevention	Use application control where appropriate. VBA macros obtained from the Internet, based on the file's Mark of the Web (MOTW) attribute, may be blocked from executing in Office applications (ex: Access, Excel, PowerPoint, Visio, and Word) by default starting in Windows Version 2203.
Restrict Web-Based Content	Script blocking extensions can help prevent the execution of scripts and HTA files that may commonly be used during the exploitation process. For malicious code served up through ads, ad blockers can help prevent that code from executing in the first place.

Defensive Recommendations: On Windows 10/Windows 11, enable Attack Surface Reduction (ASR) rules to prevent Visual Basic scripts from executing potentially malicious downloaded content.Turn off or restrict access to unneeded VB components.The user can use Script blocking extensions can help prevent the execution of scripts and HTA files.

10.Process Injection: Dynamic-link Library Injection

Mitigation	Description
Behavior Prevention on Endpoint	Some endpoint security solutions can be configured to block some types of process injection based on common sequences of behaviour that occur during the injection process.

Defensive Recommendations: Enable and configure User Account Control (UAC) settings to prompt users for approval before executing potentially malicious DLL injection actions.The user must make sure that their access privileges must not be as same as administrators.The user must password protect the administrator account.

11.Process Discovery

This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features.

12. System Information Discovery

This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features.

13. System Network Configuration Discovery: Internet Connection Discovery

This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features.

14. Archive Collected Data: Archive via Custom Method

This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features.

15. Archive Collected Data: Archive via Utility

Mitigation	Description
Audit	System scans can be performed to identify unauthorised archival utilities.

Defensive Recommendation: Logs can be maintained to identify suspicious files which are archived and stored on the local storage.Restrict the archive utility to function in the background without user’s authorisation.

16. Automated Collection

Mitigation	Description

Encrypt Sensitive Information	Encryption and off-system storage of sensitive information may be one way to mitigate collection of files, but may not stop an adversary from acquiring the information if an intrusion persists over a long period of time and the adversary is able to discover and access the data through other means. Strong passwords should be used on certain encrypted documents that use them to prevent offline cracking through Brute Force techniques.
Remote Data Storage	Encryption and off-system storage of sensitive information may be one way to mitigate collection of files, but may not stop an adversary from acquiring the information if an intrusion persists over a long period of time and the adversary is able to discover and access the data through other means.

Defensive Recommendation: Implement network segmentation to isolate sensitive data and critical systems from less secure or publicly accessible parts of the network.Restrict lateral movement within the network to limit the impact of automated collection tools.Implement strong encryption algorithms to safeguard data against automated collection attempts.

17. Clipboard Data

This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features.

18. Data Staged: Local Data Staging

This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features.

19. Input Capture: Keylogging

This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features.

20. Screen Capture

This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features.

21. Application Layer Protocol: Web Protocols

Mitigation	Description
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level.

Defensive Recommendations: Employ NIDS and other such Network monitoring systems to detect and prevent any abnormal usage of Web Protocol.

22. Ingress Tool Transfer

Mitigation	Description
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware or unusual data transfer over known protocols like FTP can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools.

Defensive Recommendations: Deploy network traffic monitoring solutions to detect anomalous or suspicious network activity.UseNetwork intrusion detection systems (NIDS) to identify patterns associated with known tool transfer methods.

23. Exfiltration Over C2 Channel

Mitigation	Description
Data Loss Prevention	Data loss prevention can detect and block sensitive data being sent over unencrypted protocols.
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool command and control signatures over time or construct protocols in such a way to avoid detection by common defensive tools.

Defensive Recommendations: The user should monitor network data for uncommon data flows. Processes utilising the network that do not normally have network communication or have never been seen before are suspicious. NIDS and NIPS can be employed to detect or prevent such requests.