



WORKSHOP

# CATCH ME IF YOU CAN

MALWARE Hide and seek

Start →





## WORKSHOP



## AGENDA:

- Welcome to the Cyber Village
- Amenazas comunes
- Casos reales de ataques de Malware
- Ejercicio practico
- Conclusión y preguntas





# AMENAZAS COMUNES DE MALWARE

## Virus

Programas que se replican y se insertan en archivos o programas existentes, propagándose a medida que se ejecutan.

## Spyware

Malware diseñado para recopilar información personal o confidencial sin el conocimiento del usuario.

## Ransomware

Malware que encripta archivos o secuestra sistemas, exigiendo un rescate para su liberación.

## Troyanos

Se presenta como software legítimo, engañando a los usuarios para que lo instalen y permitan el acceso no autorizado a sus sistemas.





# CASOS REALES

## WANNACRY

2017 - ataque masivo en todo el mundo aprovechando una vulnerabilidad en el protocolo SMB (Server Message Block) de Windows para infectar sistemas y cifrar archivos, exigiendo un rescate en Bitcoin para su desbloqueo.

## SolarWinds

2020 - Los atacantes infiltraron el software Orion de SolarWinds, utilizado por numerosas organizaciones y agencias gubernamentales, y colocaron un malware en una actualización del software. Esto les permitió acceder a redes y sistemas críticos, lo que resultó en una grave violación de seguridad a gran escala.

## NotPetya

2017 - se propagó inicialmente a través de una actualización falsa de software en Ucrania, pero rápidamente se extendió a nivel mundial. Este malware era un ransomware que cifraba los archivos de los sistemas infectados, pero a diferencia de otros no había una opción real de recuperación de datos incluso después del pago del rescate.



# CASOS REALES

## HIDE AND SEEK BOTNET

[LINK](#) 2018 Conocida como HNS infecto dispositivos IoT como camaras IP y routers

## PEGASUS SPYWARE

Utilizado para espiar a periodistas, activistas y otras personas de alto perfil. Infecta smartphones y puede acceder a mensajes, llamadas y otros datos personales.

## TEMU

[LINK](#)  
[GrizzlyReports](#)



WORKSHOP

## TEMU

PDD Holdings Inc. - la empresa creadora de la app Temu

“Cuando un producto o servicio es gratis es porque el producto eres tú”

Que dice Wikipedia?



# CASOS REALES TEMU



Security issue	TEMU	SHEIN	Alibaba.com	Amazon	TikTok	eBay
1 Local compiling with "package compile" executed with <code>getRuntime.exec()</code>	No	No	No	No	No	No
2 Requesting information if app runs with root rights ("superuser")	Yes	Yes	Yes	Yes	No	Yes
3 Request process list with "getRunningAppProcesses()"	Yes	No	Yes	Yes	Yes	Yes
4 Requesting system logs from "/system/bin/logcat"	Yes	No	No	No	No	No
5 Accessing debugger status with "Debug.isDebuggerConnected()"	Yes	Yes	Yes	Yes	No	Yes
6 Reading and writing system files in "sys/devices/"	Yes	Yes	Yes	Yes	No	No
7 Accessing external storage with "ExternalStorage"	Yes	Yes	Yes	Yes	Yes	Yes
8 Making screenshots ("get rootView()", "peekDecorView()" in "getWindow()")	Yes	Yes	Yes	Yes	Yes	No
9 Requesting the MAC address	Yes	Yes	Yes	Yes	No	Yes
10 Putting MAC address into a JSON to send the information to server	Yes	No	No	No	No	No
11 Code obfuscation with most JAVA code: unnamed files, folders, functions	Yes	No	No	No	Yes	No
12 android.permission.CAMERA	Yes	Yes	Yes	Yes	No	Yes
13 android.permission.WRITE_EXTERNAL_STORAGE	Yes	Yes	Yes	Yes	Yes	Yes
14 android.permission.RECORD_AUDIO	Yes	No	Yes	Yes	No	No
15 android.permission.INSTALL_PACKAGES	Yes	No	No	No	No	No
16 android.permission.INTERNET	Yes	No	No	Yes	No	No
17 android.permission.WAKE_LOCK	Yes	No	Yes	No	No	No
18 Putting location information into JSON to send the information to server	Yes	No	No	Yes	No	No

- **getRuntime.exec():** Ejecuta comandos del sistema a través de Runtime.exec()
- **("superuser"):** Verifica si la aplicación tiene acceso de root.
- **getRunningAppProcesses():** Solicita una lista de procesos que se están ejecutando en el dispositivo.
- **/system/bin/logcat:** Obtiene los registros del sistema (logs).

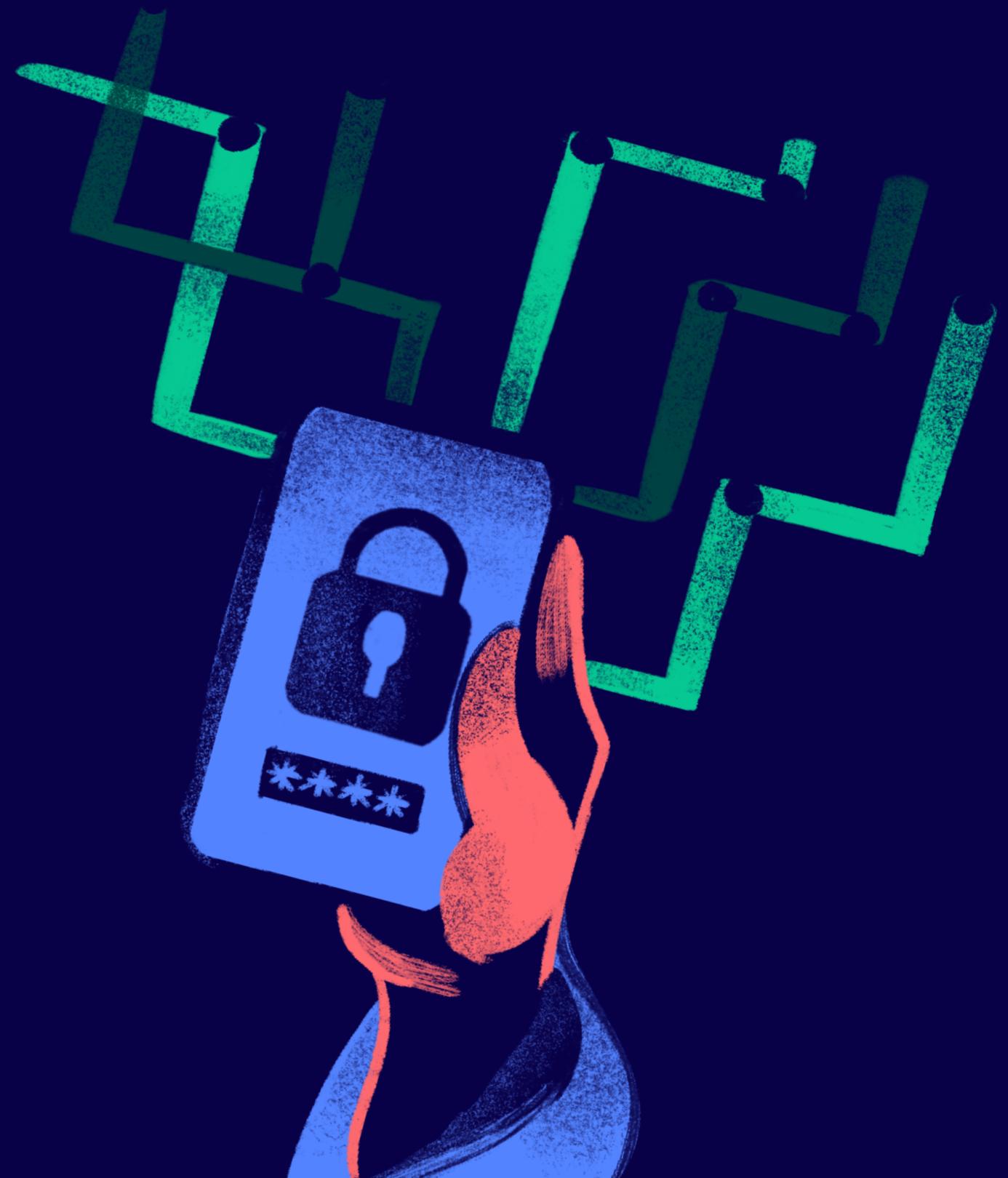
# CASOS REALES TEMU

```
16.     public static int e() {  
17.         InputStream inputStream = null;  
18.         try {  
19.             String format = String.format("cmd package compile -m speed-profile -f %s", xmg.mobilebase.apm.common.d.H().t().getPackageName());  
20.             xmg.mobilebase.apm.common.c.g("Ppm.Compile", "executeCompile cmd:" + format);  
21.             Process exec = Runtime.getRuntime().exec(format);  
22.             if (exec == null) {  
23.                 xmg.mobilebase.apm.common.c.g("Ppm.Compile", "executeCompile process is null.");  
24.                 return -1;  
25.             }  
26.             InputStream inputStream2 = exec.getInputStream();  
27.             if (inputStream2 != null) {  
28.                 String readline = new BufferedReader(new InputStreamReader(inputStream2)).readLine();  
29.                 xmg.mobilebase.apm.common.c.g("Ppm.Compile", "executeCompile res str:" + readline);  
30.                 boolean equalsIgnoreCase = "Success".equalsIgnoreCase(readline);  
31.                 if (inputStream2 != null) {  
32.                     try {  
33.                         inputStream2.close();  
34.                     } catch (IOException e12) {  
35.                         e12.printStackTrace();  
36.                     }  
37.                 }  
38.                 return equalsIgnoreCase ? 1 : 0;  
39.             }  
40.             xmg.mobilebase.apm.common.c.g("Ppm.Compile", "executeCompile InputStream is null.");  
41.             if (inputStream2 != null) {  
42.                 try {  
43.                     inputStream2.close();  
44.                 } catch (IOException e13) {  
45.                     e13.printStackTrace();  
46.                 }  
47.             }  
48.             return -1;  
49.         } catch (Throwable th2) {  
50.             try {  
51.                 xmg.mobilebase.apm.common.c.h("Ppm.Compile", "executeCompile error", th2);  
52.             return -1;  
53.         } finally {  
54.             if (S != S) {  
55.                 try {  
56.                     inputStream.close();  
57.                 } catch (IOException e14) {  
58.                     e14.printStackTrace();  
59.                 }  
60.             }  
61.         }  
62.     }  
63. }
```

- El código utiliza el método **Runtime.getRuntime().exec()** para ejecutar un comando que realiza una compilación de paquetes en el dispositivo del usuario.
- "cmd package compile" está diseñado para crear un programa ejecutable en el dispositivo del usuario..



WORKSHOP



## EJERCICIO PRACTICO

Keyloggers  
codigo





WORKSHOP



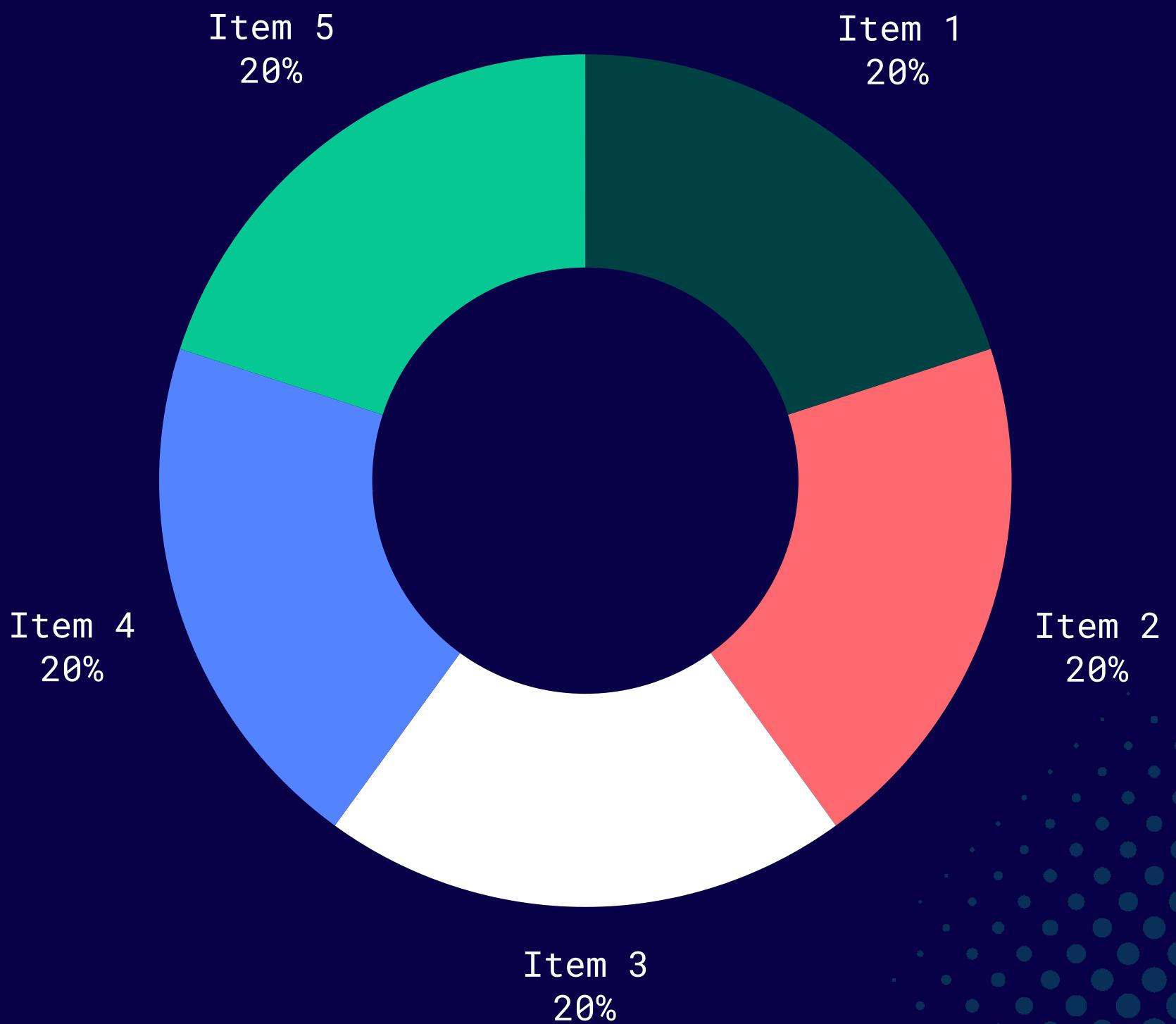
PREGUNTAS ?



WORKSHOP

# MENTIMETER

Feedback workshop



# GRACIAS



Hasta la próxima!