Incident Detection and Response Report

Client: Microsoft Windows

Project: Incident Detection and Response on Windows 8.1

Report Date: September 29, 2025

Prepared By: Patience Okafor

Responder: Patience Okafor

Contact information:

https://www.linkedin.com/in/patienceokafor/

TABLE OF CONTENTS

INTRODUCTION

Timeline Of Events

On **November 1, 2024**, the following events occurred

9:00 AM: An application named “Putty” was downloaded on to the Windows based system.

9:35 AM: The application was last modified.

EXECUTIVE SUMMARY

This report provides a detailed account of the detection, investigation, and remediation of a suspected security incident involving unauthorized remote access on a Windows-based system. Using a suite of cybersecurity tools, including TCPView, Process Explorer, and Autoruns from Microsoft Sysinternals, the investigation identified an active, unauthorized connection facilitated by a potentially malicious application, *putty.exe*. This application had established a remote connection, potentially compromising the system and posing a threat to the larger network.

The investigation commenced with the detection of an unusual process, *putty.exe*, which was connected to a remote IP address via TCPView. Further analysis with Process Explorer traced this process back to *explorer.exe*, indicating that the unauthorized application was likely downloaded through Internet Explorer and was configured to persist on the system via an autorun entry under the Windows registry. Autoruns confirmed the presence of a suspicious autorun task labelled “secure” in the registry, set to autostart upon system login.

Upon confirming the malicious nature of the application, containment efforts were initiated, including immediate network disconnection to prevent lateral movement and additional potential compromise of connected systems. Remediation involved terminating the active connection, eradicating all traces of the malicious application, and removing registry entries to prevent re-execution on startup. Finally, proactive security measures, including firewall activation, system updates, and antivirus software installation, were implemented to safeguard the system from future threats.

The report concludes with security recommendations focused on preventing similar incidents in the future. These include enhancing network monitoring and alert, restricting administrative privileges, enabling windows defender and firewall policies, and enhancing email and download filtering. These preventive measures aim to enhance security monitoring, reduce the attack surface, and strengthen the overall resilience of the system.

1. TECHNICAL SUMMARY

1.1 Tools Used for Investigation

a. TCPView: Used to list all active TCP and UDP network connections, including remote addresses, port numbers, connection states, and process IDs associated with each connection.

b. Process Explorer: Provided detailed information on running processes, memory usage, threads, and open file handles.

c. Autoruns: Displayed all programs configured to run automatically upon Windows startup, including scheduled tasks, services, drivers, and codecs.

1.2 Evidence Collected

a. Using TCPView revealed that the connection to the remote address was in an "Established" state and identified the specific port number used. It also indicated that the connection was initiated by a downloaded application named *putty.exe*.

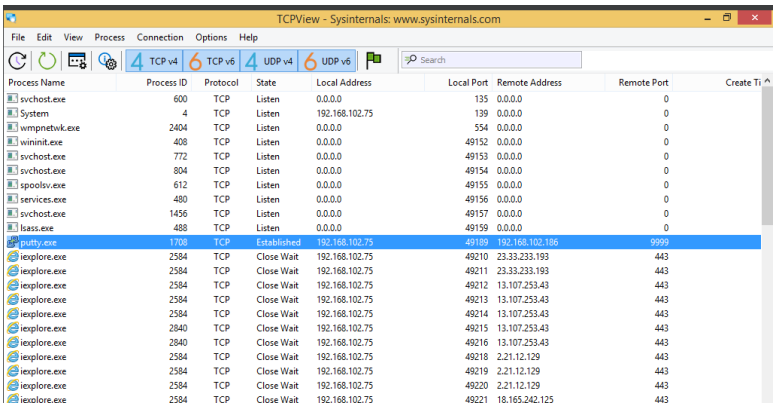
b. Using Process Explorer displayed the process tree, with *explorer.exe* as the parent process and *putty.exe* as a child process.

c. Using Autoruns showed the task "secure" entered in the Windows registry, enabling automatic connection to the remote address upon user login.

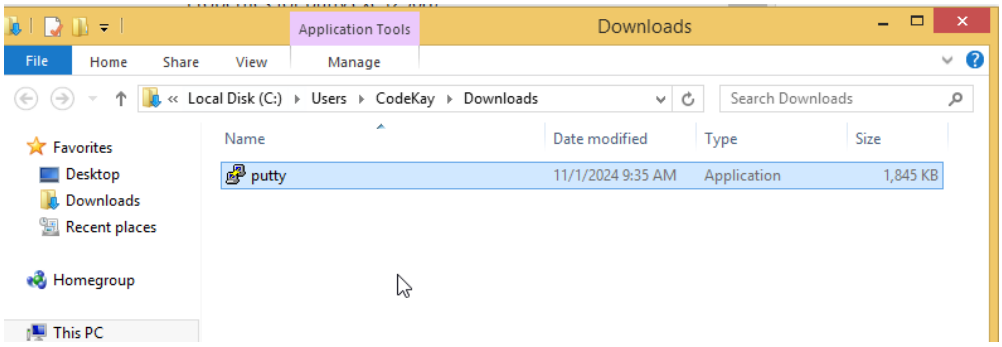
2. TECHNICAL DETAILS

2.1 Incident Detection

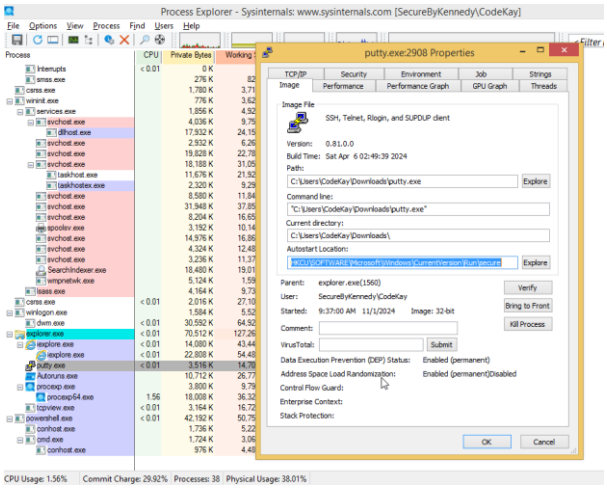
While monitoring and auditing a system with the IP address "192.168.102.186" using TCPView, I observed a suspicious process named *putty.exe* with a process ID of 1708 and a local port of 49189. This process had established a connection to the remote address 192.168.102.186 on port 9999.



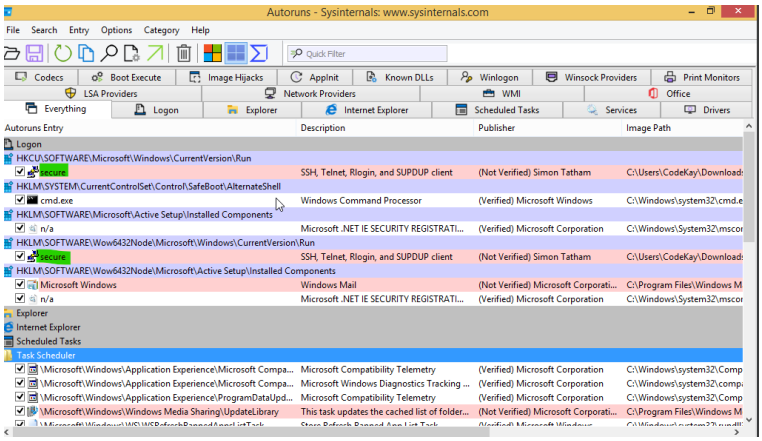
Upon identifying this process, I investigated further and determined it originated from an application named *putty*, which was downloaded and last modified on 11/1/2024 at 9:35 AM.



To understand the function of this process, I conducted additional analysis using Microsoft Sysinternals' Process Explorer. This revealed that the parent process was *explorer.exe*, indicating that the *putty* application was likely downloaded via Internet Explorer. Furthermore, the application appeared to be running a task named "secure," which was set to autostart on login under the registry key: **HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run**.



After discovering this autorun task, I used another Sysinternals tool, Autoruns, to further investigate. This analysis confirmed that the "secure" task had instances under both **HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run and HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run.**

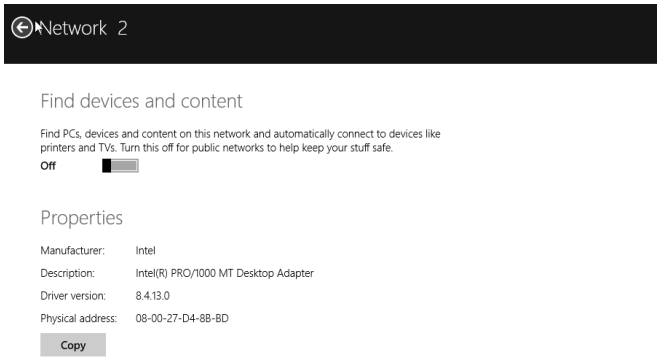


Based on these findings, I concluded that *putty.exe* was a malicious application designed to enable unauthorized remote access to the system.

2.2 Incident Response

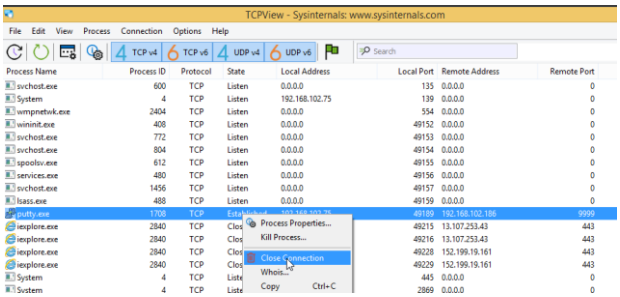
2.2.1 Containment

Given the system's connection to a wireless network, I promptly disconnected it to contain the incident and prevent potential lateral movement to other connected systems.

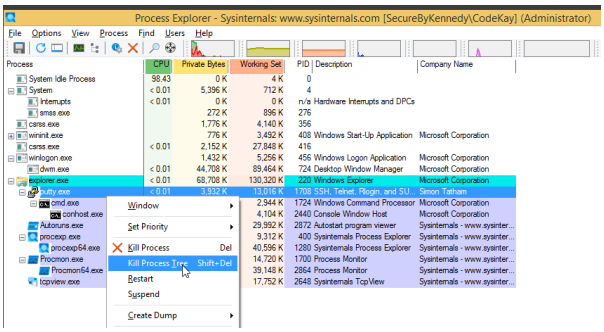


2.2.2 Eradication

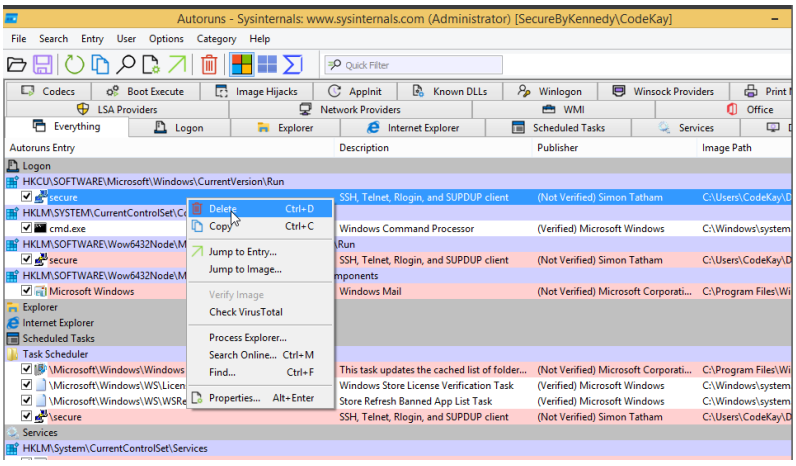
With the system isolated, I proceeded to remove all traces of the application and its associated processes and autorun tasks. First, using TCPView, I terminated the connection to the remote address, effectively halting communication with it.



Next, I used Process Explorer to kill the entire process tree initiated by *explorer.exe*, terminating any additional malicious processes potentially created by the *putty* application.



Afterwards, I deleted the "secure" autorun task from the Windows registry to prevent it from automatically executing each time a user logged in.



2.2.3 Recovery

Upon successfully eradicating the malicious processes and tasks, I reconnected the system to the network, implementing additional security measures, including enabling the Windows firewall, installing third-party antivirus software, and updating the system.

2.3 Recommendations

Based on the findings of the incident detection and response, the following recommendations can be made to improve the security of the Windows system:

a. **Enhance Network Monitoring and Alerts**
   Set up network monitoring tools to detect unusual outbound connections or connections to known malicious IP addresses. Configure alerts for any unexpected port activity or unusual communication patterns, particularly from unfamiliar applications.
b. **Restrict Administrative Privileges**
   Limit the number of accounts with administrative privileges. By reducing access to administrative accounts, you can prevent unauthorized software installations and modifications to registry settings, such as those found with the *secure* autorun task.
c. **Enable Windows Defender and Firewall Policies**
   Ensure that Windows Defender and firewall policies are active and appropriately configured. Custom rules should be added to block unauthorized access and detect unusual connections to unknown ports or IP addresses.
d. **Enhance Email and Download Filtering**
   Block or monitor the downloading of executable files like *putty.exe* through email and web browsing. Deploy email and web filtering solutions that scan for potentially dangerous attachments or links, reducing the chance of malicious downloads.

3. CONCLUSION

This report highlights the critical steps involved in detecting, analysing, and responding to a security incident involving unauthorized access on a Windows-based system. The use of advanced monitoring tools, including TCPView, Process Explorer, and Autoruns, was instrumental in identifying a malicious process (*putty.exe*) that had established unauthorized remote access. By systematically analysing the suspicious process and associated registry entries, it was possible to determine that the application was configured to persist through autorun tasks, posing an ongoing security risk to the networked environment.

Following the identification of the incident, immediate containment measures were employed to isolate the system, preventing potential lateral movement within the network. Eradication efforts focused on removing all traces of the malicious application and its associated autorun tasks, while a series of preventive security measures were enacted to enhance the system's resilience against future attacks. The incident not only underscored the value of proactive monitoring but also highlighted the importance of comprehensive incident response protocols that facilitate swift containment and remediation.

To mitigate the risk of similar incidents in the future, this report outlines several best practices, including the enforcement of application whitelisting, regular audits of autorun entries, and stringent user access controls. These recommendations are designed to reinforce security defences and improve incident readiness across the network.

In conclusion, this report demonstrates the necessity of an organized, well-equipped approach to incident detection and response. By leveraging effective tools and adhering to best practices, organizations can significantly reduce the impact of security incidents and safeguard sensitive systems and data. Continuous improvement of incident response capabilities, coupled with proactive security measures, will be essential in maintaining a robust defence against evolving cybersecurity threats.

4. APPENDICES

4.1 Glossary of Terms

a. **Autoruns**
A tool from the Microsoft Sysinternals Suite that displays programs and services configured to run automatically upon Windows startup, including registry entries and scheduled tasks.

b. **HKCU (HKEY_CURRENT_USER)**
A registry hive in Windows containing configuration information specific to the currently logged-in user. This hive includes information about user-specific settings, such as applications set to run on startup.

c. **HKLM (HKEY_LOCAL_MACHINE)**
A registry hive in Windows that stores configuration information specific to the system and hardware, affecting all users on the device. It includes settings for applications that may run for all users upon startup.

d. **Lateral Movement**
The technique used by attackers to move within a network after gaining initial access, often to reach critical assets or escalate privileges.

e. **Local Port**
A port on the host machine through which applications communicate with network services. In this context, a local port is used by a process to establish connections with remote devices.

f. **Process Explorer**
A tool from the Microsoft Sysinternals Suite that provides detailed information about active processes, including process IDs, memory usage, parent-child process relationships, and open file handles.

g. **Process ID (PID)**
A unique identifier assigned by the operating system to each running process, allowing it to be tracked and managed independently.

h. **Process Tree**
A hierarchical structure showing the parent and child relationships between processes on a system, where child processes are initiated by parent processes.

i. **Registry Key**
A specific location within the Windows registry that stores configuration settings for the operating system and applications. Registry keys can be used to set programs to run automatically on startup.

j. **Remote Address**
The IP address of a device outside the local system with which the system is

communicating over a network. Remote addresses are often used to connect to servers or other external devices.

k. **Sysinternals Suite**
A collection of utilities by Microsoft designed to troubleshoot and manage Windows operating systems, providing tools for system monitoring, process management, and startup configuration.

l. **TCPView**
A tool from the Microsoft Sysinternals Suite that displays all active TCP and UDP connections, along with details such as remote addresses, local ports, and the associated processes.

m. **Windows Registry**
A centralized database in Windows that stores low-level settings for the operating system and installed applications, used to manage startup items, user settings, and system configurations.