Demo Investigation

Endpoints

Hostname	First Timestamp	Last Timestamp	Events
SCRANTON.dmevals.local	5/1/2020 10:55:56 PM	5/2/2020 4:18:37 AM	136
NASHUA.dmevals.local	5/1/2020 11:10:23 PM	5/1/2020 11:17:52 PM	58
UTICA.dmevals.local	5/2/2020 3:55:06 AM	5/2/2020 4:21:21 AM	112
NEWYORK.dmevals.local	5/2/2020 4:02:44 AM	5/2/2020 4:17:35 AM	9

Investigation Summary

An investigation based on the https://github.com/OTRF/detection-hackathon-apt29 dataset. Investigation and endpoint summaries are generated by a large language model (LLM) from the Investigation events, comments and Micro-Technique evidence.

Generated with: llama 3.1 70B Q3 K M 32,000 context

On December 12, 2022, our security team detected suspicious activity on endpoint UTICA.dmevals.local. An incident response timeline investigation was conducted to identify the root cause of the event, assess its impact, and inform mitigation strategies.

The attacker gained initial access via certutil.exe, established persistence through Windows Management Instrumentation (WMI), and used PowerShell for command and control. Credential dumping from Local Security Authority Subsystem Service (LSASS) memory and cached domain credentials was observed. The actor also employed rundll32.exe to bypass restrictions and utilized Remote Desktop Protocol (RDP) for lateral movement.

The root cause of the incident is attributed to a misconfigured endpoint, which allowed an attacker to gain initial access via certutil.exe. Weaknesses in Windows Management Instrumentation (WMI) persistence mechanism and the use of PowerShell contributed to the actor's ability to maintain access and move laterally within the network.

The incident resulted in unauthorized access to sensitive data, including domain credentials and potentially other confidential information. The attacker's lateral movement via RDP indicates a significant risk to operational integrity and potential disruption of critical services.

Initially, the adversary gained access into the network through spearphishing. Once inside, the attacker executed malicious code using PowerShell and Command and Scripting Interpreter techniques to establish persistence. This was done by executing PowerShell scripts and system utilities such as sdelete64.exe, PsExec64.exe, and csc.exe on SCRANTON and powershell.exe with a bypass command and non-interactive window style on UTICA.

The adversary used various techniques to evade defenses, including uninstalling or disabling security software and obfuscating data and scripts. On UTICA, the execution of rundll32.exe with a VoidFunc command might be an attempt by an attacker to evade detection by running malicious code within a legitimate system process.

On NASHUA, the threat actor used Python to execute commands on the system

Investigation Summary

followed by PowerShell launching a rar.exe executable. The rar.exe was then used to compress files in multiple locations including the desktop and roaming profile.

Throughout the attack, the adversary maintained persistence through Event Triggered Execution, specifically Windows Management Instrumentation Event Subscription. On UTICA, events suggest that an attacker created a WMI filter, consumer, and action to execute powershell.exe commands when a specific event occurred.

The attacker attempted to escalate privileges through various means using Windows utilities and built-in commands, including Mimikatz to dump cached credentials from the Local Security Authority Subsystem Service (LSASS) process. This was followed by further exploitation with additional scripts conducted in order to harvest credentials from an administrative user or SYSTEM for lateral movement.

Subsequent attempts were made using PowerShell commands that loaded malicious assemblies into memory, potentially allowing access to restricted information. Additionally, there were multiple instances of file opening and closing related to LSASS credential dumping, which may have contained password material from legitimate users' accounts.

The adversary also executed tools commonly used in penetration testing such as Mimikatz or other credential dumping utilities like procdump, which is known to create local copies of lsass memory allowing offline analysis without requiring elevated privileges during runtime.

Multiple system calls requesting memory dump operations against the LSASS process were detected, facilitating further malicious activity via stolen credentials obtained through successful exploitation efforts carried out beforehand by adversary group(s).

The attacker utilized legitimate tools such as PsExec64.exe from SysinternalsSuite and PSEXESVC.exe for lateral movement, executing commands remotely against other systems on the network. They also established remote interactive sessions via RDP with remote IP's.

Throughout this campaign, the attackers primarily utilized encrypted channels, including HTTPS, to hide their activities from security controls. They demonstrated an ability to adapt and modify their tactics, techniques, and procedures (TTPs) in response to changing network conditions and defensive measures, leveraging automation scripts to execute commands across multiple systems simultaneously.

Investigation Summary

The attacker transferred a Windows executable from an external system into the compromised environment, downloading it with python.exe from 192.168.0.4. This file transfer may be part of a larger data collection effort, gathering information and tools for exfiltration or impact tactics.

On UTICA, the actor decoded a file using certutil.exe, creating a new file named "kxwn.lock" in the user's AppData directory. Subsequently, the actor used net.exe to establish a network connection with 13.107.42.12 over port 443, providing login credentials for an Outlook account.

This connection likely facilitated data exfiltration to cloud storage services such as Dropbox or Google Docs. The actor then transferred a Windows PE file, classified as a Windows executable with a SHA-256 hash of "581449ef7c37587169ef45b1efbe1875236f3f2b83a0e52c8788e5fa4eb7d6e5", into the compromised environment.

On NASHUA, a user account initiated an archival process using Rar.exe on a desktop file and moved it to the roaming directory. This action may be a preliminary step in data staging for theft or exfiltration, as compressing and encrypting the data can make it easier to transport and more difficult to detect.

The actor initially transferred tools into the compromised environment, gathering information and tools for potential exfiltration or impact tactics. The actor then used various techniques to establish command and control, including decoding files and establishing network connections with remote IP addresses. Throughout these events, the actor demonstrated a clear intent to stage data for exfiltration, using third-party utilities like Rar.exe to compress and encrypt the data.

NASHUA.dmevals.local Summary

Endpoint Name: NASHUA.dmevals.local
Operating System: Windows

First Event Timestamp: 05/02/2020 02:55:24
Last Event Timestamp: 05/02/2020 08:29:22
Observed IP Addresses: 10.0.1.6
Observed Users: DMEVALS\pbeesly, pbeesly

Executive Summary

The threat actor began by using Python to execute commands on the system, followed by PowerShell launching a rar.exe executable. The rar.exe was then used to compress files in multiple locations including the desktop and roaming profile. A subsequent command and control channel was established using cmd.exe to execute an sdelete64.exe utility which was then used to delete files.

The adversary started by establishing a network connection using Remote Services via Windows Logon events, utilizing valid credentials for user 'pbeesly' from domain 'DMEVALS.LOCAL' with remote IP '10.0.1.4'. They leveraged Software Deployment Tools, executing PSEXESVC.exe and python.exe processes, allowing the adversary to move laterally through the network. The use of conhost.exe with the -ForceV1 flag suggests an attempt to escalate privileges or bypass security controls.

The threat actor's actions indicate a level of persistence and execution, using built-in utilities like PowerShell, cmd, and other system tools in an attempt at defense evasion through living off the land tactics. This behavior continued throughout the timeline, with repeated executions of PSEXESVC.exe and python.exe, indicating a sustained effort by the adversary to maintain access and move laterally through the network.

A user account initiated an archival process using Rar.exe on a desktop file and moved it to the roaming directory. This could be a preliminary step in data staging for theft or exfiltration, as compressing and encrypting the data can make it easier to transport and more difficult to detect.

NEWYORK.dmevals.local Summary

NEWYORK.dmevals.local

Windows
05/02/2020 02:55:25
05/02/2020 08:29:21
10.0.0.4
pbeesly, DMEVALS\mscott, dschrute, mscott, kmalone

Summary of Findings

This incident involved attempted unauthorized credential access and privilege escalation on a host system. A user account attempted to run Mimikatz to extract Windows credentials suggesting attackers had already compromised an administrative user's login session with domain privileges but wanted to expand their permissions through lateral movement or evade detection using alternative authentication mechanisms.

Multiple instances of file opening and closing related to LSASS credential dumping were detected on this system, indicating attackers employed tools commonly used in penetration testing such as mimikatz or other credential dumping utilities like procdump. These files may have contained password material from legitimate users' accounts for unauthorized access by malicious actors.

Attackers also used rundll32.exe to interact with comsvcs.dll, resulting in system calls requesting memory dump operations against the LSASS process thereby facilitating further malicious activity via stolen credentials obtained through successful exploitation efforts carried out beforehand by adversary group(s).

Root Cause Analysis

The root cause of this incident is attributed to the attackers' ability to exploit vulnerabilities and leverage tools commonly used in penetration testing. The use of mimikatz, procdump, and other credential dumping utilities suggests a high degree of sophistication and familiarity with Windows system internals.

Impact Assessment

This incident has significant implications for the security posture of the affected organization. The attempted unauthorized credential access and privilege escalation demonstrate the attackers' intent to expand their permissions and move laterally within the network. The use of stolen

NEWYORK.dmevals.local Summary

credentials obtained through successful exploitation efforts carried out beforehand by adversary group(s) further exacerbates the risk of unauthorized access to sensitive data and systems.

The fact that multiple users, including pbeesly, DMEVALS\mscott, dschrute, mscott, and kmalone, were observed on the affected system suggests a potential insider threat or compromised account. The incident highlights the need for enhanced security measures, including robust access controls, monitoring, and incident response protocols to prevent and detect similar incidents in the future.

UTICA.dmevals.local Summary

Endpoint Name: UTICA.dmevals.local
Operating System: Windows

First Event Timestamp: 08/25/2011 18:27:29
Last Event Timestamp: 05/02/2020 08:29:23
Observed IP Addresses: 10.0.1.5
Observed Users: DMEVALS\mscott, DMEVALS\dschrute, dschrute

Summary of Findings:
The incident involved an adversary establishing Command and Control (C2) with a compromised system through Ingress Tool Transfer. This was followed by multiple instances of PowerShell execution, credential dumping via LSASS memory, WMI event subscription creation, and file modifications. The adversary demonstrated persistence in maintaining access to the compromised system.

Root Cause Analysis:
The root cause of the incident appears to be the initial exploitation of a vulnerability or misconfiguration that allowed the adversary to establish C2 with the compromised system. This was likely facilitated by weaknesses in the system's security posture, such as inadequate patch management, insufficient monitoring, or poor configuration.

Impact Assessment:
The impact of the incident is significant, with potential risks including unauthorized access to sensitive data, disruption of operations, and lateral movement within the network. The adversary's ability to maintain persistence on the compromised system increases the risk of further exploitation and potential damage.

TA0001 Initial Access
The adversary likely used spearphishing to gain initial access into the network.

The event timeline starts at 0755 hours with conhost.exe executing a command and immediately followed by certutil.exe decoding a blob from C:\Users\dschrute\AppData\Roaming\Microsoft\kxwn.lock. These events suggest that an adversary might have used the downloaded malware to gain initial access into the system.

TA0002 Execution
The adversary executed malicious code on the system using PowerShell and Command and Scripting Interpreter techniques.
Multiple instances of powershell.exe executing with a bypass command and non-interactive window style indicate that an attacker was attempting to

UTICA.dmevals.local Summary

execute commands without being detected at 0758 hours. The presence of encoded commands further supports this theory.

TA0003 Persistence
The adversary used Event Triggered Execution, specifically Windows Management Instrumentation Event Subscription, to maintain persistence on the system.
At 0800 and 0812 hours, events suggest that an attacker created a WMI filter, consumer, and action to execute powershell.exe commands when a specific event occurred.

TA0005 Defense Evasion
The adversary used various techniques to evade defenses, including uninstalling or disabling security software and obfuscating data and scripts.
At 0809 hours, the execution of rundll32.exe with a VoidFunc command might be an attempt by an attacker to evade detection by running malicious code within a legitimate system process.

The analysis indicates that on May 2nd, a malicious actor used a combination of Windows utilities and built-in commands to elevate their privileges on an endpoint within the DMEVALS domain. This included the execution of Mimikatz to dump cached credentials from the Local Security Authority Subsystem Service (LSASS) process, followed by further exploitation with the aid of additional scripts.

The malicious actor's primary objective is believed to be privilege escalation for lateral movement and potential access to sensitive information within the target network.

Remote Execution and Lateral Movement throughout the network appeared to be achieved through a combination of PowerShell scripts executed over various protocols including HTTP/S, RDP, WinRM, and WMI. Initial infection occurred via exploitation of the "Get-Content" command in PowerShell, which enabled the adversary to download malicious code.

The attackers leveraged legitimate tools such as WinRM for lateral movement to evade detection, as well as RDP protocol connections. They also used PowerShell scripts to interact with remote systems and execute commands on targeted machines.

Throughout this campaign, the attackers primarily utilized encrypted channels, including HTTPS, to hide their activities from security controls. The attackers took advantage of the fact that many organizations allow outbound HTTPS traffic, making it a common blind spot for many defenders.

UTICA.dmevals.local Summary

The attackers demonstrated an ability to adapt and modify their tactics, techniques, and procedures (TTPs) in response to changing network conditions and defensive measures. They leveraged automation scripts to execute commands across multiple systems simultaneously, enabling them to move quickly and stay one step ahead of defenders.

The adversary initially established command and control by transferring tools into the compromised environment, specifically downloading a Windows PE file classified as a Windows executable with a SHA-256 hash of "581449ef7c37587169ef45b1efbe1875236f3f2b83a0e52c8788e5fa4eb7d6e5". This was followed by the use of certutil.exe to decode a file, creating a new file named "kxwn.lock" in the user's AppData directory.

The adversary then utilized net.exe to establish a network connection with a remote IP address (13.107.42.12) over port 443, using HTTPS protocol and providing login credentials for an Outlook account. This connection was likely used to exfiltrate data to cloud storage services such as Dropbox or Google Docs. The use of existing web services like these provides significant cover for the adversary's activities, making it harder to detect malicious behavior.

Throughout these events, the adversary demonstrated a clear intent to establish command and control, transfer tools, and ultimately exfiltrate data from the compromised environment using various techniques and tactics.

SCRANTON.dmevals.local Summary

Endpoint Name: SCRANTON.dmevals.local
Operating System: Windows
First Event Timestamp: 08/25/2011 18:27:29
Last Event Timestamp: 05/02/2020 08:29:16
Observed IP Addresses: 10.0.1.4
Observed Users: DMEVALS\pbeesly, pbeesly, DMEVALS\mscott, DMEVALS\kmalone, kmalone

The Summary:
The adversary used a python.exe download from an external IP address 192.168.0.4 to establish command and control with the endpoint on April 30th, 2020.
The malware was masqueraded as Right-to-Left Override on May 2nd, 2020, which initiated its first appearance.
The malware executed persistence techniques through registry run keys, startup folders, and boot or logon autostart execution to maintain access to the endpoint.
Furthermore, the adversary utilized powershell.exe for OS credential dumping, process injection, and portable executable injection to evade defenses.
The impact of this incident is assessed as medium risk due to the potential for data theft and system compromise.

The adversary initiated their attack with a Right-to-Left Override (RTLO) technique on May 2nd, 2020. This was followed by several executions of PowerShell scripts and system utilities such as sdelete64.exe, PsExec64.exe, and csc.exe. The use of these tools suggests that the adversary aimed to maintain persistence on the compromised system.

The timeline indicates that the adversary attempted to evade detection through process injection into lsass.exe and conhost.exe processes, which are legitimate system components. This may have been done to conceal their malicious activities from security monitoring tools. Furthermore, the execution of hostui.exe and powershell.exe with obfuscated commands suggests an attempt to establish persistence on the compromised system.

The repeated executions of PsExec64.exe also indicate that the adversary attempted to expand their control over the network by executing commands remotely. The use of these techniques in combination indicates a high degree of intent to evade detection and maintain a persistent foothold within the targeted environment.

Throughout this activity, the adversary attempted to gain elevated permissions through various means, including accessing credential material stored in the Local Security Authority Subsystem Service (LSASS) process memory. This was

SCRANTON.dmevals.local Summary

done using the LSASS Memory technique, which allowed them to harvest credentials from an administrative user or SYSTEM and conduct lateral movement. The adversary also executed a PowerShell command that loaded a malicious assembly into memory, potentially allowing them to access restricted information.

Additionally, the adversary attempted to dump credentials from the operating system and software, likely to obtain account login and credential material for further unauthorized access. This was done using the OS Credential Dumping technique, which can provide clear text passwords or hashes that can be used for lateral movement and accessing restricted information.

The adversary's actions suggest a strong intent to escalate privileges and gain unauthorized access to sensitive information. They utilized Windows Remote Management and Remote Desktop Protocol across several hosts on the network to conduct lateral movement. The actor leveraged multiple tools and protocols including PowerShell, WinRM, RDP, and PsExec64.exe from SysinternalsSuite for lateral movement throughout the environment.

Software deployment tools like PsExec64 appear to have been used for executing commands remotely against other systems on the network. The actor also established remote interactive sessions via RDP with remote IP's as 172.18.39.2. These techniques were likely used by the adversary for gaining access and escalating privileges on various endpoints throughout the environment.

The adversary transferred a Windows executable from an external system into a compromised environment using an Ingress Tool Transfer technique. The file was downloaded by python.exe from the IP address 192.168.0.4, indicating potential lateral movement or command and control activity. This action may be part of a larger data collection effort, with the adversary gathering information and tools in preparation for exfiltration or impact tactics.