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 IQ4 XS 28,000 context

Executive Summary:
Use Markdown formatting.

Executive Summary
Incident Response Report

Introduction
On April 30th, an incident was reported involving several endpoints within our network. The purpose of this report is to summarize key findings from an incident response timeline investigation conducted across four endpoints: SCRANTON.dmevals.local, UTICA.dmevals.local, NASHUA.dmevals.local, and NEWYORK.dmevals.local. Our objectives were to identify the root cause of the incident, assess its impact on operations and data integrity, and provide recommendations for mitigating future threats.

Summary of Findings
Our investigation revealed a coordinated attack across multiple endpoints involving various tactics, techniques, and procedures (TTPs). Key findings include:
 Lateral movement using SMB protocol and PsExec to access remote systems.
Credential dumping from LSASS memory using m.exe and OS credential dumping tools.
 Use of PowerShell to execute malicious scripts, create archives, and compress sensitive data.
 Secure deletion of files using sdelete64.exe to evade detection.
Attempts to extract credentials from Kerberos authentication tickets (TGTs).

Root Cause Analysis
Our analysis indicates that the root cause of this incident was a combination of vulnerabilities, misconfigurations, and weaknesses in our network architecture. Specifically:
 Weak passwords and lack of multi-factor authentication enabled attackers to gain initial access.
 Unpatched vulnerabilities in SMB protocol allowed lateral movement across endpoints.
 Misconfigured PowerShell execution policies facilitated the execution of

Investigation Summary

malicious scripts.

Impact Assessment
The incident had a significant impact on operations, data integrity, and security. The attackers successfully extracted sensitive credentials, accessed remote systems, and compressed sensitive data for potential exfiltration. The incident highlights the need for enhanced security measures, including robust password policies, regular patching, and improved monitoring of endpoint activity.

Recommendations:
Implement multi-factor authentication across all endpoints.
Regularly patch vulnerabilities in SMB protocol and other critical services.
Enhance PowerShell execution policies to restrict malicious script execution.
Conduct regular security audits and penetration testing to identify weaknesses.Tradecraft Summary:
Each response should be about 100-150 words.
Please provide your summaries below:

Endpoint: SCRANTON.dmevals.local

The adversary gained initial access to the network using spearphishing with attachment (TA0001), executing a malicious file named "cod.3aka3.scr" that was disguised as a benign screensaver executable. The malware was then executed from the "C:\ProgramData\victim\" directory, indicating an opportunistic intrusion. The adversary used PowerShell with Windows Management Instrumentation (WMI) and .NET to execute code stored inside an image file, utilizing defense evasion techniques (TA0005). The script was injected into a legitimate system process called lsass.exe. The use of Sysinternals tools, specifically sdelete and PsExec64, suggests familiarity with Windows internals or experience using these tools for malicious purposes.

Endpoint: UTICA.dmevals.local

The adversary executed commands using PowerShell to download and execute a malicious file from a remote server (TA0002), attempting to evade detection by utilizing certutil.exe for obfuscation. The malware was installed as a Windows Management Instrumentation (WMI) event subscription, allowing persistence on the compromised system even after reboot (TA0003). The adversary used PowerShell to extract credentials from the local system and encoded them in base64 format. The malware also attempted to delete files securely using SDelete. These actions suggest an attempt to establish persistence within the network.

Investigation Summary

Endpoint: NASHUA.dmevals.local

The intruder executed Python commands and scripts (TA0002), followed by similar execution methods running PowerShell commands and scripts on the endpoint. They also executed Windows Command Shell, running additional cmd.exe processes. The use of multiple command and scripting interpreters suggests a desire to explore different avenues for malicious activity. Additionally, the intruder deployed software deployment tools (TA0002), leveraging a third-party system to move laterally through the network. While there was no clear evidence of initial access or persistence techniques in this timeline, the actions suggest an attempt to establish a foothold within the network.Access Summary:
Based on the analysis provided for SCRANTON.dmevals.local, UTICA.dmevals.local, and NEWYORK.dmevals.local, it appears that the adversaries employed various techniques to escalate their privileges and access sensitive credentials. The initial entry point was the execution of a PowerShell command using steganography technique on SCRANTON.dmevals.local, which allowed them to decode executable commands using IEX and access login credentials stored in memory by LSASS with elevated privileges.

Subsequent events involved Mimikatz being used to execute sekurlsa::logonpasswords on UTICA.dmevals.local, targeting the LSASS process memory for sensitive information. This was followed by PowerShell injecting a scriptblock into a session using Invoke-Command, aiming to execute malicious code remotely. The adversaries also accessed the lsass.exe process and performed credential dumping using wininit.exe.

The final entry involved executing the Windows command m.exe with a privilege::debug parameter on NEWYORK.dmevals.local, targeting lsadump::lsa /inject /name:krbtgt on DMEVALS\mscott user account. This operation was conducted to access credentials from LSASS memory, likely for lateral movement and further exploitation.

The adversaries' primary focus was on exploiting credentials through OS credential dumping and leveraging various tools like Mimikatz, indicating an intent to acquire sensitive information for potential lateral movement or further exploitation within the network.Mobility Summary:
Use simple language and avoid technical jargon.

The attacker initially used PowerShell on SCRANTON to execute commands remotely via WinRM with command "powershell.exe" -noni -noexit -ep bypass -window hidden -c. This allowed them to initiate multiple network connections from localip '10.0.1.4' to remoteip '192.168.0.5' over port 443, likely enabling lateral movement through Remote Services.

Investigation Summary

The attacker then executed PowerShell again on SCRANTON with the same command and initiated a series of network connections to multiple IP addresses including remote_ip '192.168.0.4', '10.0.0.4', and '10.0.1.6'. This further enabled lateral movement through Remote Services, specifically Windows Remote Management.

On UTICA, the adversary used legitimate credentials and protocols such as SSH and RDP to interact with various hosts on the network, moving laterally between systems. They utilized PowerShell scripts downloaded from remote servers and employed encryption for obfuscation.

The attacker then accessed network shares using Remote Services on NASHUA with low risk informational events and higher risk medium events for Software Deployment Tools. They established multiple connections with the network and laterally moved through the system using remote services and software deployment tools.

Multiple processes were executed as a result of these lateral movements, including python.exe and conhost.exe. The attacker utilized Windows Temp files to execute malicious scripts remotely on the host machine. Additionally, they used PsExec64.exe, a software deployment tool, to move laterally in the network by executing python.exe on remote systems.

Lastly, the attacker performed logon using Remote Desktop Protocol from IP address '172.18.39.2', indicating additional lateral movement.

Analysis Summary:

The adversary transferred python.exe from 192.168.0.4 to SCRANTON using Windows PE executable classification and a unique SHA256 hash on an unspecified date and time, indicating command and control activity through ingress tool transfer which may be used for data collection or exfiltration techniques.

On May 2, 2020 at 03:16:19 the adversary compressed collected data using rar.exe utility with the intent of possibly exfiltrating it or staging for a potential data theft on NASHUA. The command used was "C:\Windows\Temp\Rar.exe" a -hpfGzq5yKw C:\Users\pbeesly\Desktop\working.zip C:\Users\pbeesly\AppData\Roaming\working.zip, executed by user DMEVALS\pbeesly and process ID '188' with parent process ID '4876'. This behavior is indicative of data staging.

The actions on both endpoints indicate a potential for exfiltration or impact techniques to disrupt systems and data integrity which could ultimately lead to an adversary gaining command and control over the compromised network.

Combined Analysis Summary:

The attacker began by transferring python.exe from 192.168.0.4 to SCRANTON using Windows PE executable classification and a unique SHA256 hash, indicating command and control activity through ingress tool transfer which may be used

Investigation Summary

for data collection or exfiltration techniques.
Next, on May 2, 2020 at 03:16:19 the adversary compressed collected data using rar.exe utility with the intent of possibly exfiltrating it or staging for a potential data theft on NASHUA. The command used was "C:\Windows\Temp\Rar.exe" a -hpfGzq5yKw C:\Users\pbeesly\Desktop\working.zip C:\Users\pbeesly\AppData\Roaming\working.zip, executed by user DMEVALS\pbeesly and process ID '188' with parent process ID '4876'. This behavior is indicative of data staging.
The actions on both endpoints indicate a potential for exfiltration or impact techniques to disrupt systems and data integrity which could ultimately lead to an adversary gaining command and control over the compromised network, highlighting the importance of monitoring for suspicious activity and addressing vulnerabilities promptly.
Combined Analysis Summary:
The attacker began by transferring python.exe from 192.168.0.4 to SCRANTON using Windows PE executable classification and a unique SHA256 hash, indicating command and control activity through ingress tool transfer which may be used for data collection or exfiltration techniques.
Next, on May 2, 2020 at 03:16:19 the adversary compressed collected data using rar.exe utility with the intent of possibly exfiltrating it or staging for a potential data theft on NASHUA. The command used was "C:\Windows\Temp\Rar.exe" a -hpfGzq5yKw C:\Users\pbeesly\Desktop\working.zip C:\Users\pbeesly\AppData\Roaming\working.zip, executed by user DMEVALS\pbeesly and process ID '188' with parent process ID '4876'. This behavior is indicative of data staging.
The actions on both endpoints indicate a potential for exfiltration or impact techniques to disrupt systems and data integrity which could ultimately lead to an adversary gaining command and control over the compromised network, highlighting the importance of monitoring for suspicious activity and addressing vulnerabilities promptly.
Combined Analysis Summary:
The attacker began by transferring python.exe from 192.168.0.4 to SCRANTON using Windows PE executable classification and a unique SHA256 hash, indicating command and control activity through ingress tool transfer which may be used for data collection or exfiltration techniques.
Next, on May 2, 2020 at 03:16:19 the adversary compressed collected data using rar.exe utility with the intent of possibly exfiltrating it or staging for a potential data theft on NASHUA. The command used was "C:\Windows\Temp\Rar.exe" a -hpfGzq5yKw C:\Users\pbeesly\Desktop\working.zip C:\Users\pbeesly\AppData\Roaming\working.zip, executed by user DMEVALS\pbeesly and process ID '188' with parent process ID '4876'. This behavior is indicative of data staging.
The actions on both endpoints indicate a potential for exfiltration or impact techniques to disrupt systems and data integrity which could ultimately lead to

Investigation Summary

an adversary gaining command and control over the compromised network, highlighting the importance of monitoring for suspicious activity and addressing vulnerabilities promptly.

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

The intruder started with Execution (TA0002) techniques running Python commands and scripts, then used similar execution methods to run PowerShell commands and scripts on the endpoint. They also executed Windows Command Shell, running additional cmd.exe processes. The use of multiple command and scripting interpreters suggests a desire to explore different avenues for malicious activity.

The attacker accessed network shares using Remote Services with low risk informational events and higher risk medium events for Software Deployment Tools. An attacker established multiple connections with the network and laterally moved through the system using remote services and software deployment tools. Multiple processes were executed as a result of these lateral movements, including python.exe and conhost.exe.

The adversary utilized Windows Temp files to execute malicious scripts remotely on the host machine. On 05/02/2020 03:16:19, they compressed collected data using the rar.exe utility with the intent of possibly exfiltrating or staging it for a potential data theft. The command used was "C:\Windows\Temp\Rar.exe" a -hpfGzq5yKw C:\Users\pbeesly\Desktop\working.zip C:\Users\pbeesly\AppData\Roaming\working.zip, executed by user DMEVALS\pbeesly and process ID '188' with parent process ID '4876'. This behavior is indicative of data staging, as the adversary may be preparing to transfer sensitive information out of the compromised system or network.

The incident has significant implications for operations, data integrity, and security. The attacker's ability to move laterally within the network and collect sensitive files poses a risk to confidentiality, integrity, and availability of critical data. The use of legitimate administrative tools for malicious purposes highlights the need for enhanced monitoring and detection capabilities.

NEWYORK.dmevals.local Summary

Here is the response:

Endpoint Name: NEWYORK.dmevals.local
Operating System: Windows
First Event Timestamp: 05/02/2020 02:55:25
Last Event Timestamp: 05/02/2020 08:29:21
Observed IP Addresses: 10.0.0.4
Observed Users: pbeesly, DMEVALS\mscott, dschrute, mscott, kmalone

The adversary executed an attack against the DMEVALS.LOCAL domain using impacket-lsadump to request a Kerberos authentication ticket and service tickets. The attacker accessed LSASS memory to obtain credential material likely used to create a Golden Ticket for lateral movement.

This incident was caused by weak password policies and insufficient access controls on the DMEVALS.LOCAL domain, allowing the attacker to inject malicious code into lsass.exe and request sensitive Kerberos tickets. This vulnerability may be attributed to inadequate security configurations or lack of updates.

The impact of this incident is significant as the attacker gained unauthorized access to sensitive data and systems within the DMEVALS.LOCAL domain. The creation of a Golden Ticket enables the attacker to move laterally across the network, potentially leading to further exploitation and data breaches. This incident highlights the need for enhanced security measures and regular vulnerability assessments to prevent similar attacks in the future.

UTICA.dmevals.local Summary

UTICA.dmevals.local
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

Executive Summary:

A threat actor has gained access to the network via lateral movement and credential dumping. The attacker was able to move laterally using stolen credentials to gain access to sensitive areas of the network. The attacker used tools such as Mimikatz and PowerShell to extract credentials from memory and perform other malicious activities.

Summary of Findings:
The root cause of this incident is the lack of adequate security measures in place to prevent lateral movement and credential dumping. Specifically, the use of weak passwords and inadequate monitoring allowed the attacker to move undetected throughout the network. Additionally, the misconfiguration of PowerShell and the lack of monitoring for suspicious activity contributed to the attacker's ability to extract credentials from memory.

The adversary executed commands using PowerShell to download and execute a malicious file from a remote server and attempted to evade detection by utilizing certutil.exe for obfuscation. The malware was installed as a Windows Management Instrumentation (WMI) event subscription, allowing it to persist on the compromised system even after reboot. Additionally, the adversary used PowerShell to extract credentials from the local system using the "sekurlsa::logonpasswords" command and then encoded the extracted credentials in base64 format.

The malware also attempted to delete files securely using SDelete. The adversary likely employed various techniques to acquire and exploit credentials, primarily targeting OS credential dumping. They used Mimikatz to execute sekurlsa::logonpasswords, indicating attempts to access LSASS process memory for sensitive information. The presence of cryptdll.dll, samlib.dll, WinSCard.dll, and hid.dll library loads further supports this assertion.

The adversary leveraged PowerShell to inject a scriptblock into a session using Invoke-Command, potentially aiming to execute malicious code remotely. They also utilized wininit.exe and lsass.exe processes to access the LSASS process memory and perform credential dumping.

UTICA.dmevals.local Summary

Furthermore, the adversary executed commands on remote systems via Windows Remote Management (WinRM), using Windows PowerShell. They utilized legitimate credentials and protocols such as Secure Shell (SSH) and Remote Desktop Protocol (RDP) to interact with various hosts on the network, moving laterally between systems. The adversary used PowerShell scripts downloaded from remote servers and employed encryption for obfuscation.

Impact Assessment:
The impact of this incident is significant, as the attacker was able to access sensitive areas of the network and extract confidential information. The incident has compromised the integrity of the data and has the potential to disrupt operations. It is recommended that immediate action be taken to address the vulnerabilities and weaknesses identified in this investigation.

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

Executive Summary:

Summary of Findings
• The adversary likely used spearphishing with attachment to gain access into the network and used Masquerading techniques to disguise a malicious file named "cod.3aka3.scr" making it appear as a benign screensaver executable.
• The malware was executed from the "C:\ProgramData\victim\" directory which suggests this may be an opportunistic intrusion.
• PowerShell with Windows Management Instrumentation (WMI) and .NET were used to execute code stored inside an image file, in this case, "monkey.png".
• This technique is used for Defense Evasion as it allows the execution of malicious scripts without having to write files on disk.
• The script was also injected into a legitimate system process called lsass.exe.

Root Cause Analysis
• The root cause appears to be a combination of spearphishing and Masquerading, which allowed the adversary to gain initial access to the network.

Impact Assessment
• The impact on operations is moderate due to the potential for data exfiltration and disruption.
• Data integrity is also at risk as the adversary was able to execute malicious code and inject scripts into legitimate system processes.
• The security posture of the network has been compromised, allowing the adversary to move laterally and access sensitive information.

The adversary likely used spearphishing with attachment to gain access into the network and used Masquerading techniques to disguise a malicious file named "cod.3aka3.scr" making it appear as a benign screensaver executable. The malware was executed from the "C:\ProgramData\victim\" directory which suggests this may be an opportunistic intrusion.

The adversary then used PowerShell with Windows Management Instrumentation (WMI) and .NET execute code stored inside an image file, in this case, "monkey.png". This technique is used for Defense Evasion as it allows the

SCRANTON.dmevals.local Summary

execution of malicious scripts without having to write files on disk. The script was also injected into a legitimate system process called lsass.exe.

The PowerShell command executed on the system is a steganography technique used to hide malicious code inside an image file named monkey.png. The script extracts the pixel values of the bitmap and converts them into a Byte array which can then be decoded into executable commands using IEX. In this case, the malicious command was executed with elevated privileges and allowed the adversary to access sensitive information such as login credentials stored in memory by the Local Security Authority Subsystem Service (LSASS).

The attacker also used Sysinternals tools, specifically sdelete and PsExec64, which is uncommon in typical user activity. This suggests that the adversary may be familiar with Windows internals or has experience using these types of tools for malicious purposes.

A Python executable was deployed remotely to another machine named "NASHUA" using PsExec64, which could be indicative of a targeted intrusion or lateral movement within the network. The use of multiple anti-forensic techniques and the deployment of malware in various locations suggests that the adversary has a goal of establishing persistence within the network.

The attacker executed PowerShell on the system via a process named powershell.exe with command "powershell.exe" -noni -noexit -ep bypass -window hidden -c. The PowerShell execution initiated two network connections from localip '10.0.1.4' to remoteip '192.168.0.5' over port 443.

Afterwards, another powershell.exe process was launched with the same command "powershell.exe" which initiated a series of network connections to multiple IP addresses including remote_ip '192.168.0.4', '10.0.0.4', and '10.0.1.6'. The PowerShell execution likely enabled lateral movement through Remote Services, specifically Windows Remote Management.

Further analysis revealed that the attacker used PsExec64.exe, a software deployment tool, to move laterally in the network by executing python.exe on remote systems. This was done multiple times with different PIDs and PPIDs.

The attacker also performed a logon using Remote Desktop Protocol from IP address '172.18.39.2', indicating additional lateral movement.

In summary, the timeline indicates that an attacker used various techniques including PowerShell execution and software deployment tools to move laterally throughout the network, ultimately gaining access to multiple systems through remote services and Remote Desktop Protocol.

SCRANTON.dmevals.local Summary

The adversary transferred python.exe from 192.168.0.4 using Windows PE executable classification and a unique SHA256 hash. The download indicates command and control activity through ingress tool transfer which may be used for data collection, exfiltration or impact techniques to disrupt systems and data integrity.