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 L 32,000 context

Endpoint Forensics Report

Introduction
This report provides a detailed analysis of endpoint data collected during an incident response investigation into suspicious activities detected on multiple endpoints across our network.

Summary of Findings
Our analysis revealed several key events, including initial access through right-to-left override character tricking users into executing benign file types. The exact vector for initial access is unknown but the execution of malicious code occurred across all three endpoints with the adversary utilizing Command and Scripting Interpreter, software deployment tools like PsExec64.exe and Sdelete64.exe, as well as process injection to maintain persistence and execute code on systems for prolonged periods of time.

Patterns suggest a sophisticated threat actor employing various tools and techniques to maintain stealth and evade detection. The tradecraft suggests that this could be a targeted intrusion given the level of sophistication and customized approach used by the adversary.

Access Summary
The aim is to provide a clear understanding of the adversary actions. Initially, on Endpoint UTICA.dmevals.local, the adversary attempted privilege escalation through process "m.exe," executing a command with Mimikatz's sekurlsa module. This was followed by library loads and an open process event targeting lsass.exe. The goal appears to have been injecting malicious code into LSASS for credential dumping.

On Endpoint SCRANTON.dmevals.local, a PowerShell script execution occurred, employing steganography techniques to potentially conceal malicious code within a bitmap image. The manipulated bytes were then executed using IEX (Invoke-Expression). This may be an attempt to run arbitrary code in a hidden manner.

Subsequently, on Endpoint UTICA.dmevals.local, the adversary accessed LSASS process memory credentials through Mimikatz's sekurlsa module. Following this,

Investigation Summary

wininit.exe was executed with SYSTEM privileges, and lsass.exe was launched as a subprocess of wininit.exe. Cached domain credentials were then dumped from LSASS process memory.

Later, on Endpoint NEWYORK.dmevals.local, the adversary accessed the Local Security Authority Subsystem Service (LSASS) process memory to dump credentials. They injected into LSASS using the "/inject" option with the target name "krbtgt."

Throughout these events, the adversary demonstrated a clear intent to escalate privileges and access credentials, aligning with MITRE Tactics TA0004 (Privilege Escalation) and TA0006 (Credential Access). The actions suggest a sophisticated attack aimed at gaining control over the system and moving laterally within the network.

Mobility Summary
The threat actor gained initial access by leveraging Remote Services, using valid accounts for lateral movement across multiple endpoints throughout the environment.
Initial Execution (Endpoint SCRANTON.dmevals.local): The attacker executed commands remotely via Windows Remote Management (WinRM), then expanded access to additional systems through this method.

Lateral Movement and Remote Execution (Endpoints UTICA.dmevals.local and NASHUA.dmevals.local): Multiple connections were made to remote systems through WinRM and Remote Desktop Protocol, targeting hosts at IP addresses 10.0.0.4 and 172.18.39.2. The attacker may have attempted to pivot between these systems or establish persistence using WMI commands.

Further Lateral Movement (Endpoint NASHUA.dmevals.local): An actor with valid credentials used Remote Desktop Protocol (RDP) and third-party software deployment tools like PSEXESVC.exe to gain access to multiple hosts on the network. The actor executed malicious python code hosted on a compromised system in the C:\Windows\Temp directory.

The threat actor maintained persistence by utilizing various Microsoft protocols and tools, including WinRM, RDP, and third-party software deployment tools, indicating an intent to expand access and maintain persistence within the environment.

Data Movement Summary
On April 30th, at approximately 00:35:26 UTC, an adversary likely transferred tools onto a compromised endpoint using Command and Control protocols. The downloaded file was identified as a Windows Portable Executable (PE) named python.exe, suggesting the setup of infrastructure for potential data theft or

Investigation Summary

staging.

The following day, May 1st, at 07:09:23 UTC, communication was initiated with another compromised device to facilitate tool transfer via Ingress Tool Transfer. The transferred file was identified as a Windows PE executable. A day later, on May 2nd, an encoded file was decoded using certutil.exe, potentially extracting another payload.

On the same day, at 03:16:19 UTC, data was collected and archived using a utility like 'Rar.exe', likely to compress or encrypt the information prior to exfiltration. The execution of 'Rar.exe' with specific commands and options implies that the adversary had control over the compromised system through Command and Control.

Moments later, at 07:55:26 UTC, the adversary established a network connection to a cloud storage service using compromised credentials, possibly transferring data out of the network. The archived file was moved from the desktop to a roaming folder, suggesting that the attacker intended to make the data accessible across different systems or networks.

The adversary's intent is clear: to exfiltrate data through command and control channels, leveraging third-party tools for archive and transfer. The use of these tools indicates a high degree of sophistication and control over the compromised systems, increasing the likelihood of successful exfiltration.

Root Cause Analysis
Our investigation indicates that the root cause of this incident lies in vulnerabilities within our email security posture, allowing for successful spear phishing attacks. Additionally, weaknesses in endpoint configuration and lack of adequate monitoring contributed to the attacker's ability to remain undetected.

Impact Assessment
The incident has significant implications for operations, data integrity, and confidentiality. Potential compromise of sensitive data and credentials poses a substantial risk to business continuity. Immediate action is required to contain further damage and ensure the security posture of our network.

Recommendations
Implement enhanced email security measures, including advanced threat detection and phishing protection.
Conduct thorough endpoint vulnerability assessments and remediate identified weaknesses.
Enhance monitoring capabilities to detect similar activity in the future.

Investigation Summary

By taking these steps, we can minimize the impact of this incident and improve our overall security posture moving forward.

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 incident began with an initial connection from 10.0.1.6 to 10.0.0.4, initiating SMB communication over TCP port 445. The threat actor then launched a series of lateral movement attempts using the PSEXESVC utility, followed by the deployment of Python and Conhost executables.

Summary of Findings:
The timeline reveals a coordinated attack involving the exploitation of Windows services for lateral movement and data exfiltration. The threat actor used the PSEXESVC utility to execute commands remotely, deployed Python scripts to collect sensitive files, and utilized Conhost to establish persistence on the compromised system. Notably, the attacker employed PowerShell to compress collected files into an archive.

Root Cause Analysis:
The root cause of this incident appears to be a combination of vulnerabilities in Windows services and misconfigurations allowing for lateral movement. The threat actor exploited these weaknesses to gain access to sensitive data and deploy malicious tools.

Impact Assessment:
This incident had significant implications for operations, data integrity, and security posture. Sensitive files were exfiltrated, and the attacker established persistence on the compromised system, potentially allowing for future attacks. The use of legitimate Windows utilities and services made detection challenging, highlighting the need for enhanced monitoring and threat hunting capabilities.

The adversary gained access to the system through unknown means but was able to execute a series of commands using Python. The first command was run at 3:15:04 and used Python to launch a new process. Shortly after, at 3:15:48, another command was executed using PowerShell, which is likely being used as an intermediary to further expand the adversary's access.

The repeated use of sdelete64.exe across multiple executions suggests an attempt by the adversary to cover their tracks or potentially remove evidence

NASHUA.dmevals.local Summary

of their activities. The fact that they accepted the EULA for SDelete in each instance implies a deliberate effort to utilize this tool for malicious purposes.

Throughout these events, registry modifications were made to reflect the acceptance of the SDelete EULA, which indicates a possible attempt by the adversary to maintain persistence on the system or ensure future access.

An actor with valid credentials used Remote Desktop Protocol (RDP) and third-party software deployment tools like PSEXESVC.exe to gain access to multiple hosts on the network. The actor used RDP to remotely log in to systems from IP address 10.0.1.4, then executed malicious python code hosted on a compromised system in the C:\Windows\Temp directory.

The use of valid credentials and third-party tools allowed the actor to potentially move undetected across the network, using the deployment software as a conduit for their malicious activity. This technique enabled the actor to bypass traditional security controls and maintain persistence on the compromised systems.

The adversary collected data on May 2nd, 2020 at 03:16:19 UTC and archived it using a utility, likely to compress or encrypt the information. This was done prior to exfiltration, suggesting that the attacker aimed to minimize detection while removing the data from the network. The use of a utility like 'Rar.exe' indicates that the adversary employed a third-party tool to package the collected files into an archive.

The execution of 'Rar.exe' with specific commands and options implies that the adversary had control over the compromised system, possibly through command and control (TA0011). This level of control would allow them to manipulate the data, making it easier to exfiltrate. The fact that the archived file was moved from the desktop to a roaming folder suggests that the attacker intended to make the data accessible across different systems or networks, increasing the likelihood of successful exfiltration.

The adversary then utilized Windows Command Shell (cmd.exe) to execute a series of commands, starting with the execution of Rar.exe, a file archiver, at 3:16:19. The specific command and arguments passed to Rar.exe suggest that it was used to extract files from a zip archive located on the desktop and in the roaming user profile.

Further analysis reveals that the adversary then launched cmd.exe again, this time using it to execute sdelete64.exe, a file deletion tool. This occurred at 3:16:52, and the command line arguments suggest that it was used to delete

NASHUA.dmevals.local Summary

```
files from specific locations, including the Windows Temp directory.
```

NEWYORK.dmevals.local Summary

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 incident started with an unknown process execution on May 2, 2020. The attackers utilized Windows Management Service (WMS) to establish remote connectivity from 10.0.1.5 to 10.0.0.4 on port 5985. This action was followed by a series of events including the creation and modification of files, specifically the m.exe file created in C:\Windows\System32\.

The attackers then used this newly created process to inject code into LSASS (Local Security Authority Subsystem Service) memory using the debug privilege. Furthermore, they accessed LSASS's memory to extract credentials using a technique known as "OS Credential Dumping". This action was associated with an elevated risk and represented a significant threat to endpoint security.

The attackers also used Kerberos tickets for lateral movement. Specifically, event 4769 indicated the request of a service ticket which is often indicative of lateral movement. Moreover, the attackers used a Golden Ticket to obtain access to the system.

On May 2, 2020, at 08:04:25, the adversary accessed the LSASS process memory utilizing 'm.exe' with elevated privileges in an attempt to dump credentials. They injected into LSASS using the "/inject" option with the target name "krbtgt," which is likely indicative of attempting to gain access to the Kerberos Ticket Granting Ticket account.

The loading of additional libraries, including cryptdll.dll, samlib.dll, hid.dll, and WinSCard.dll, may suggest an attempt to further interact with the system or exploit specific vulnerabilities. The use of 'm.exe' with elevated privileges suggests that the adversary may have already gained administrative access, which they are leveraging to inject into LSASS and potentially dump credentials.

The intent behind this action appears to be the escalation of privileges, allowing the adversary to move laterally within the network and access restricted information. Overall, this behavior aligns with MITRE Tactic TA0006 (Credential Access) and Technique T1003 (OS Credential Dumping), as well as TA0004 (Privilege Escalation).

The incident had significant implications for endpoint security as it allowed

NEWYORK.dmevals.local Summary

attackers to extract credentials, move laterally within the network, and obtain elevated privileges. The impact on data integrity is moderate due to the potential extraction of sensitive information from the compromised endpoints. Operations were also moderately impacted, with some disruption possible during the investigation and remediation process.

Analysis suggests that the root cause of this incident was likely related to an initial misconfiguration or exploitation of vulnerabilities within the WMS and its associated components. This highlights the need for enhanced security measures, including robust monitoring, vulnerability management, and secure configuration practices.

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

Executive Summary:

The response should not exceed 400 words.

Here is your executive summary:

Summary of Findings:

The investigation revealed a series of coordinated events indicating a targeted attack on the endpoint. On May 1st, 2020 at 07:09:23 UTC, an Ingress Tool Transfer was detected, where an executable file with a SHA256 hash of '581449ef7c37587169ef45b1efbe1875236f3f2b83a0e52c8788e5fa4eb7d6e' was transferred to the endpoint. This file was later identified as Mimikatz, a tool commonly used for credential dumping.

On May 2nd, 2020 at 07:58:21 UTC and 07:59:10 UTC, multiple PowerShell executions were detected, with commands indicating attempts to dump credentials from LSASS memory using the 'mimikatz' command. These events suggest that the attacker successfully executed their tool on the endpoint.

Root Cause Analysis:

The root cause of this incident is attributed to a vulnerability in the endpoint's configuration, allowing an adversary to transfer and execute malicious tools. Specifically, the use of PowerShell commands with elevated privileges enabled the attacker to dump credentials from LSASS memory.

Impact Assessment:

This incident has significant implications for the organization's security posture. The successful execution of Mimikatz on the endpoint indicates that the attacker was able to access sensitive credential information, potentially allowing lateral movement within the network. Furthermore, this incident highlights a vulnerability in the endpoint's configuration, which must be addressed to prevent similar attacks in the future.

It is essential for the organization to take immediate action to remediate this vulnerability and conduct further investigation to determine the extent of the attack and potential data exfiltration.

The Tradecraft analysis should include the following elements:

UTICA.dmevals.local Summary

Initial Access

Execution

Persistence

Defense Evasion

This is a Windows system with PowerShell available and it appears the threat actor has executed various commands, scripts and interpreted languages using these tools.

Initial access was gained through an unknown vector but once inside, the adversary used Command and Scripting Interpreter to execute commands and scripts in multiple ways including using PowerShell, certutil.exe and csc.exe. They also utilized the Windows Management Instrumentation (WMI) for event triggered execution. Execution of malicious code continued throughout the timeline with the actor downloading files from a remote server using powershell.exe and running them on the system.

Persistence was achieved through WMI subscriptions to maintain access to the system across restarts and interruptions. The adversary set up a subscription to execute a command line every time an instance creation event occurred, allowing them to persist their access.

Defense evasion techniques were used throughout the timeline including obfuscating files or information by using certutil.exe to decode blobs of data. Additionally, the actor used rundll32.exe to run malicious code from an AppData directory, evading detection through this technique. The threat actor also made use of PowerShell's ability to bypass security controls and execute commands in a non-interactive manner.

The tradecraft suggests that this could be a targeted intrusion given the level of sophistication and customized approach used by the adversary.Access Summary:
500 words maximum.

The initial access point appears to have been through a process named "m.exe" which executed a command using the sekurlsa module of Mimikatz, attempting to dump LSASS memory credentials. This was followed by several library loads and an open process event targeting lsass.exe, indicating that the adversary had gained access to sensitive system information.

The timeline then reveals a PowerShell script execution, invoking a command that suggests the adversary attempted to inject malicious code into the LSASS process using Mimikatz's sekurlsa module. This was done in an attempt to escalate privileges and gain further control over the system.

UTICA.dmevals.local Summary

Subsequent events show wininit.exe executing with SYSTEM privileges, followed by lsass.exe being executed as a subprocess of wininit.exe. The adversary then attempted to dump cached domain credentials from the LSASS process memory using Mimikatz's sekurlsa module.

The final entries in the timeline reveal the use of Invoke-Mimikatz-Evals, a command that executes a PowerShell script block on a remote system. This was done twice, with the second attempt using the same command but with different parameters. The investigator comments suggest that this was an attempt to create a Kerberos golden ticket for lateral movement.

In conclusion, the timeline reveals a pattern of attempts by the adversary to escalate privileges and gain further control over the system through various means, including LSASS memory credential dumping, PowerShell script execution, and Kerberos golden ticket creation. These actions demonstrate a clear intent to misuse or escalate privileges on the compromised system.Mobility Summary:
300 word limit.
The attacker executed malicious code on a system with multiple network connections and remote access attempts over various protocols.
The initial execution used a PowerShell process that downloaded files from an IP address at 192.168.0.4. This was followed by additional executions, some using base64 encoded commands, which interacted with the Windows Management Instrumentation service.
Multiple connections were made to remote systems through WinRM and Remote Desktop Protocol, targeting hosts at IP addresses 10.0.0.4 and 172.18.39.2. The attacker may have attempted to pivot between these systems or establish persistence using WMI commands.
Some PowerShell executions were run under the context of "NT AUTHORITY\SYSTEM", indicating potentially elevated privileges.
The use of base64 encoded strings, multiple protocol connections, and varying levels of system access all contribute to an environment with significant lateral movement risk. The observed tactics are consistent with an adversary attempting to move laterally through a network using Windows remote services and RDP.

These findings indicate a high likelihood of lateral movement within the network, enabled by the exploitation of various Microsoft protocols. It is crucial for defenders to monitor these network interactions closely and verify system integrity following such events.Data Movement Summary:
05/01/2020 07:09:23, the adversary initiated communication with a compromised device within the network to facilitate tool transfer via Ingress Tool Transfer (TA0011.T1105.000). The transferred file was identified as a Windows PE executable.

UTICA.dmevals.local Summary

A day later, on 05/02/2020 07:55:26, an encoded file, possibly malicious, was decoded using certutil.exe, potentially extracting another payload. This file may have been used to prepare the environment for exfiltration.

Moments later on 05/02/2020 08:09:58, the adversary utilized net.exe to establish a network connection to a cloud storage service (Exfiltration Over Web Service), using compromised credentials and possibly transferring data out of the network.

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

An adversary gained initial access to the endpoint via a right-to-left override (RTLO) character tricking users into executing benign file types. This technique has been seen in many targeted intrusion attempts and criminal activity. The malware then persisted with registry run keys creating autostart entries by abusing legitimate Windows processes.

The adversary utilized various techniques, including process injection as a defense evasion technique, to hide malware execution by injecting malicious code into legitimate processes like lsass.exe. They also used command and scripting interpreter (powershell), software deployment tools like PsExec64.exe and Sdelete64.exe, demonstrating efforts to maintain persistence and execute code on systems for prolonged periods of time.

The adversary may have utilized an RTLO character to create a backdoor on the targeted system for lateral movement or command and control communication in multiple phases throughout initial access through execution and defense evasion techniques. In one instance, privilege escalation was attempted through a suspicious PowerShell script execution that used the "-ep bypass" flag, indicating evasion of PowerShell's Execution Policy.

The script loaded a bitmap image from "C:\Users\pbeesly\Downloads\monkey.png" and manipulated its pixels to potentially conceal malicious code using steganography. After processing the image, the script executed the modified bytes using IEX (Invoke-Expression), which could be an attempt to run arbitrary code in a hidden manner.

A minute later, an OS Credential Dumping event was detected involving LSASS Memory access. The target process was "C:\windows\system32\lsass.exe," which is responsible for storing credential materials. Accessing this memory can provide account login and credential material, enabling lateral movement and privilege escalation.

The threat actor gained initial access by leveraging Remote Services, using a valid account for lateral movement across multiple endpoints throughout the environment. They executed commands remotely via Windows Remote Management (WinRM), then expanded access to additional systems through this method.

SCRANTON.dmevals.local Summary

The attackers also leveraged Software Deployment Tools, likely PsExec64.exe, for lateral movement and remote execution of code on various hosts. They further utilized Remote Desktop Protocol (RDP) for accessing endpoints, indicating they were able to move laterally across the network and potentially expand access from system to system by using valid accounts.

The utilization of multiple protocols and tools suggests an intent to maintain persistence within the environment. The threat actor's movement through various systems in the network indicates a sophisticated attack aimed at escalating privileges and compromising credentials.