

CSE509

Computer System Security



2023-03-30

Post-exploitation

Michalis Polychronakis

Stony Brook University

**Committee on Oversight and Government Reform
U.S. House of Representatives
114th Congress**



**The OPM Data Breach: How the Government Jeopardized Our
National Security for More than a Generation**

Majority Staff Report

**Hon. Jason Chaffetz, Chairman
Committee on Oversight and Government Reform**

**Hon. Mark Meadows, Chairman
Subcommittee on Government Operations**

**Hon. Will Hurd, Chairman
Subcommittee on Information Technology**

September 7, 2016

www.oversight.house.gov

Timeline of Key Events

July 2012

- ✓ Attackers had access to OPM's network, according to US-CERT.¹ US-CERT found malware (Hikit) resided on an OPM server since 2012.²

November 2013

- ✓ First evidence of adversarial activity by the attacker associated with the breach that US-CERT informed OPM about in March 2014.³

December 2013

- ✓ First evidence of adversarial activity associated with the 2015 breaches (including harvesting of credentials from OPM contractors) by the attacker that was not identified until April 2015.⁴

March 20, 2014

- ✓ US-CERT notifies OPM of a data exfiltration from OPM's network.⁵ OPM, working with US-CERT, determines and implements a strategy to monitor the attackers' movements to gather counterintelligence. This breach involved data that included manuals and IT system architecture information, but the full extent of exfiltrated data is unknown.
- ✓ The strategy remains in place until the "Big Bang" on May 27, 2014.

March 25, 2014

- ✓ Situation report takes place with CIO Donna Seymour and US-CERT.⁶

March 27, 2014

- ✓ As OPM monitors the hackers, it develops a "Plan for full shut down [of systems] if needed."⁷

Attackers had access since at least 2012

Breach was initially discovered in 2014

[...]

April 23, 2015

- ✓ OPM determines there had been a “major incident” involving the exfiltration of personnel records, which triggers a requirement to notify Congress.⁴⁰
- ✓ OPM notifies Congress of a “major incident” on April 30, 2015.⁴¹

April 24, 2015

- ✓ OPM orders a global quarantine to address malware identified by CylanceProtect.⁴²

April 26, 2015

- ✓ Cylance engineers identify adversarial activity related to an RDP session to a background investigation database indicating this session took place in June 2014.⁴³

May 8, 2015

- ✓ US-CERT establishes with a high degree of certainty that personnel records data/PII had been stolen.⁴⁴

May 20, 2015

- ✓ OPM determines there was a major incident regarding the exfiltration of background investigation data, which triggers a requirement to notify Congress.
- ✓ OPM notifies Congress on May 27, 2015.⁴⁵

Got EIP, now what?

Typical goals

- Determine the value of the compromised target
- Maintain access (often for as long as possible)
- Infect more hosts and gather information

Typical steps

- Information gathering
- Privilege escalation
- Persistence
- Hiding
- Lateral movement

Actual goals/steps depend on the tactical objective

ATT&CK Matrix for Enterprise

layout: side ▾

show sub-techniques

hide sub-techniques

Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery
10 techniques	7 techniques	9 techniques	13 techniques	19 techniques	13 techniques	42 techniques	17 techniques	30 techniques
Active Scanning (3)	Acquire Infrastructure (7)	Drive-by Compromise	Command and Scripting Interpreter (8)	Account Manipulation (5)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Adversary-in-the-Middle (3)	Account Discovery (4)
Gather Victim Host Information (4)	Compromise Accounts (3)	Exploit Public-Facing Application	Container Administration Command	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Brute Force (4)	Application Window Discovery
Gather Victim Identity Information (3)	Compromise Infrastructure (7)	External Remote Services	Deploy Container	Boot or Logon Autostart Execution (14)	Access Token Manipulation (5)	BITS Jobs	Credentials from Password Stores (5)	Browser Bookmark Discovery
Gather Victim Network Information (6)	Develop Capabilities (4)	Hardware Additions	Exploitation for Client Execution	Boot or Logon Initialization Scripts (5)	Boot or Logon Autostart Execution (14)	Build Image on Host	Exploitation for Credential Access	Cloud Infrastructure Discovery
Gather Victim Org Information (4)	Establish Accounts (3)	Phishing (3)	Inter-Process Communication (3)	Browser Extensions	Boot or Logon Initialization Scripts (5)	Debugger Evasion	Forced Authentication	Cloud Service Dashboard
Phishing for Information (3)	Obtain Capabilities (6)	Replication Through Removable Media	Native API	Compromise Client Software Binary	Create or Modify System Process (4)	Deobfuscate/Decode Files or Information	Forge Web Credentials (2)	Cloud Service Discovery
Search Closed Sources (2)	Stage Capabilities (6)	Supply Chain Compromise (3)	Scheduled Task/Job (5)	Create Account (3)	Domain Policy Modification (2)	Deploy Container	Input Capture (4)	Cloud Storage Object Discovery
Search Open Technical Databases (5)			Serverless Execution	Create or	Escape to Host	Direct Volume Access	Modify	Container and Resource Discovery
						Domain Policy Modification (2)		Debugger Evasion
						Execution Guardrails (1)		

Information Gathering

Look for valuable information that will facilitate subsequent steps

- Understand system configuration

- Identify security/monitoring/management mechanisms

- Enumerate users, devices, hosts, ...

- Pinpoint potential next targets

- Gather any useful intelligence information

Initial compromised host may be useful as a stepping stone/staging server for subsequent attacks

Main sources of data

- Local information

- Network reconnaissance

- Long-term passive host/network monitoring

Local Enumeration

Users and roles

- Username and passwords, access lists, policies, ...

- Access/activity logs, command history files, ...

System information

- Installed services, running processes, cron jobs, log files, configuration files, scripts, ...

- Network interfaces, routes, DNS servers, proxies, open ports, ...

- Network shares, backup destinations, ...

- Devices (adapters, camera/mic, USB, etc.), driver/firmware versions, ...

- Virtualization, containers, ...

Personal information

- Notes, stored passwords, HTTP cookies, emails, chat history, pictures, temp folders, ...

Deployed defenses

- Firewalls, AVs, performance monitoring and system management tools, ...

- Installed patches, updates, program versions, ...

<code>/etc/passwd</code>	Listing of system user accounts
<code>/etc/ftpusers</code>	Listing of users allowed to access the FTP server
<code>/etc/pam.d</code>	Pluggable Authentication Module (PAM) config files
<code>/etc/shadow</code>	Actual passwords for cracking
<code>/etc/hosts.allow</code>	Hostnames that are allowed to access the system
<code>/etc/hosts.deny</code>	Hostnames not allowed to access the system
<code>/etc/securetty</code>	Listing of TTY interfaces that will permit a root login
<code>/etc/security</code>	Security policies
<code>/etc/rc.d</code>	Service and program startup files
<code>/etc/crontab</code>	Scheduled tasks
<code>/etc/fstab</code>	Partition information
<code>/etc/ssh</code>	Read or modify the SSH configuration
<code>/etc/sysctl.conf</code>	Kernel options
<code>/etc/sysconfig</code>	System configuration files

<code>/etc/dhcp</code>	Information about DHCP connections
<code>/etc/resolv.conf</code>	DNS configuration
<code>/etc/ldap/ldap.conf</code>	LDAP configuration
<code>/etc/samba/smb.conf</code>	Samba configuration
<code>/var/log/messages</code>	System messages
<code>/var/log/wtmp</code>	Currently logged-in users
<code>/var/log/lastlog</code>	History of logged-in users
<code>/etc/apt/sources.list</code>	Package repositories (may incl. custom ones)
<code>~/.bash_history</code>	
<code>~/.ssh/known_hosts</code>	
<code>~/.ssh/id_rsa</code>	
<code>~/.git</code>	
<code>~/*</code>	
<code>/*</code>	

Remote Enumeration

Network caches (ARP, DNS, browser history, ...)

Configured remote shares/servers

Currently open sockets

Network scanning

Long-term Collection

Network sniffing, MiTM attacks, incoming/outgoing connections, resolved DNS names, ...

Passive collection of user input

- Key logging, file logging, screen capture, ...

- Credentials, addresses, input form data, messages, ...

Privilege Escalation

Exploiting a process may not always provide super-user access

- Process running under standard (non-privileged) user account

- Sandboxing (e.g., Chrome, Adobe Reader)

- Containers (e.g., jail, LXC, Docker)

- Virtualized guest OS

Different techniques

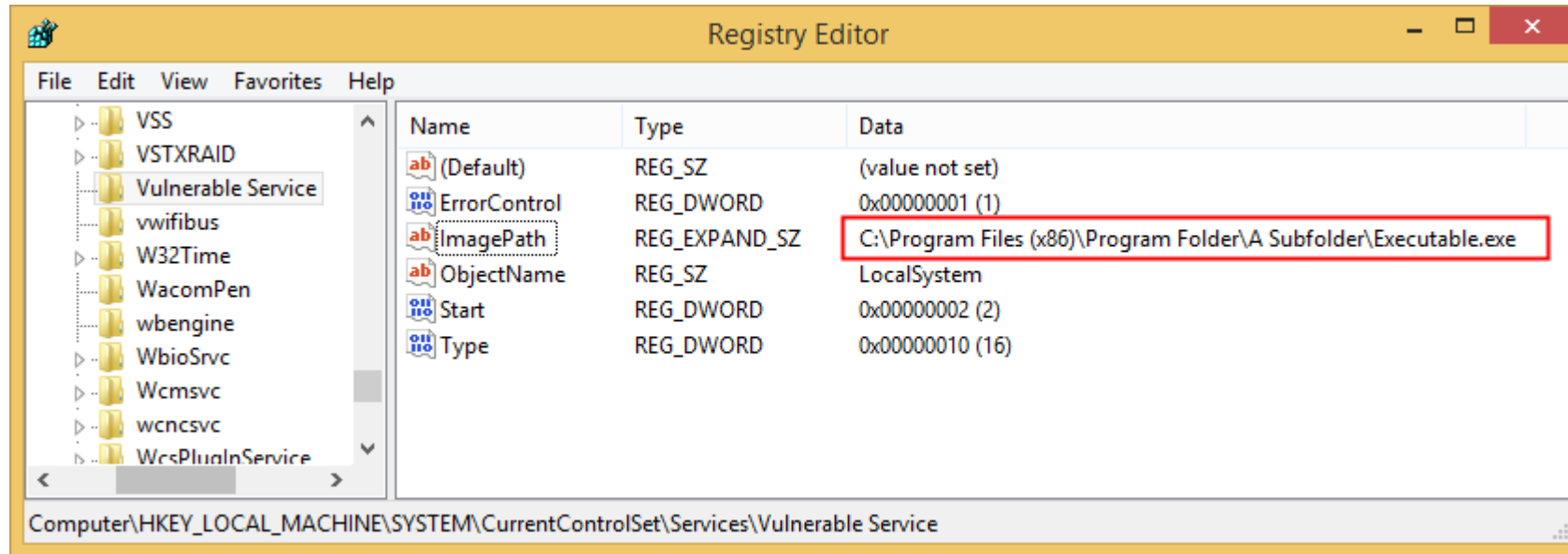
- Various OS-level tricks: plant binary into special system directory, abuse users with '.' in \$PATH, abuse \$LD_PRELOAD, loadable modules, symbolic links, ...

- Exploit vulnerabilities in user-space system services

- Exploit sandbox/kernel/hypervisor-level vulnerabilities

Example: Unquoted Service Paths

Occurs when a service executable path is not enclosed with quotation marks and contains space



Wrong: C:\Program Files (x86)\Program Folder\A Subfolder\Executable.exe

Correct: "C:\Program Files (x86)\Program Folder\A Subfolder\Executable.exe"

Example: Unquoted Service Paths

When the OS attempts to run this service, it will look at the following paths in order and will run the first EXE found:

`C:\Program.exe`

`C:\Program Files.exe`

`C:\Program Files (x86)\Program.exe`

`C:\Program Files (x86)\Program Folder\A.exe`

`C:\Program Files (x86)\Program Folder\A Subfolder\Executable.exe`

Due to the way `CreateProcess()` works

If an attacker can drop a malicious executable in one of these paths, Windows will run it as SYSTEM upon service restart

The attacker should have the right privileges on one of these folders

Windows Privileges

Unix root == Windows SYSTEM or Administrator

Standard user: needs Administrator permission for system-level actions

Administrator: highest-privilege user account

SYSTEM: same privilege level as Administrator, but not a regular account

Internal account used by the OS to run system services

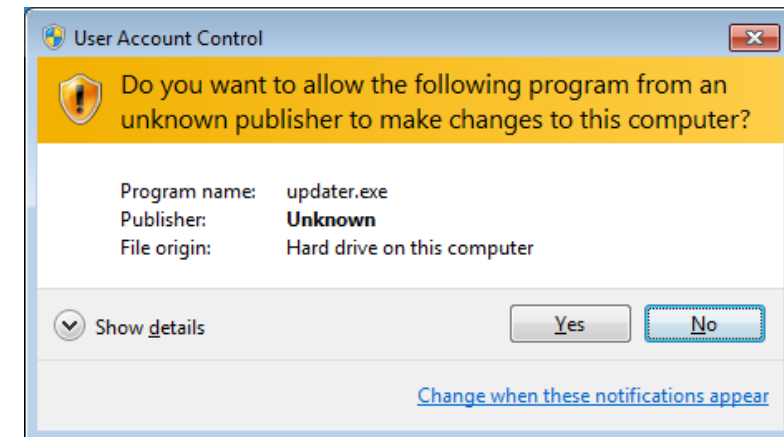
User Access Control (UAC)

“super-user when necessary”

Request user approval for system tasks

Aims to prevent unauthorized changes (e.g., by malware)

Many ways to bypass



Persistence

Not always needed (!)

- May be easy to just re-exploit or (remotely) re-authenticate

- Attacker may be interested in just a short-term goal

...but usually desirable

- Single-shot exploits (e.g., client-side exploits through phishing): the attacker typically has only one chance of fooling the victim into clicking on the malicious link/file

- Systems may be patched soon: previous vulnerabilities may disappear

- An initially uninteresting target may become useful in the future

Persistence

Type of access

Continuous access (persistent connection)

Push: connect to the target through backdoor whenever needed (has become challenging due to NAT/firewalls)

Pull: target connects periodically to C&C server (typical in botnets)

Access lifetime

Process (e.g., memory-resident malware): lost after process termination/reboot

System (e.g., typical malware): persistence across reboots

Machine (e.g., MBR or firmware-level rootkits): persistence across reformat

Achieving Persistence

Extra malicious code into the system

- Executables/scripts that run at system startup

- More stealthy rootkits/backdoors/trojans

- Shellcode/ROP, DLL injection, ...

Use existing accounts/add new ones

- Take advantage of (or enable) existing remote access mechanisms (ssh, VNC, RDP, configuration management systems, ...)

Introduce new vulnerabilities (and then re-exploit)

- Not common, but may be a useful option (especially for web applications)

- Crypto backdoors, enable insecure options, downgrade protocols, ...

Achieving Persistence

Stealthiness is important!

Minimize the artifacts and “noise” of the attack, and stay under the radar

Avoid extra accounts

Better replace service (e.g., VNC server) with a modified version that permits login from a special user/password

Avoid extra open ports

Use port knocking, use pull instead of push, ...

Avoid unusual remote endpoints

Host C&C server on Google/Amazon/other non-suspicious address

Hiding

Evade detection and cover any tracks

IDS, AV, syslog, firewalls, process monitors, ...

Important step for testing an organization's security posture

(Besides being vulnerable in the first place)

Nobody will take action for a problem they don't know it exists

Main goals

Do not affect normal operation: avoid crashes, slowdowns, and any other disruption

Do not raise suspicion: avoid noisy or clearly unanticipated behavior

Achieving Stealth

Evasion

Transform attack vectors/malicious executables to avoid detection by scanners and analysis systems

Polymorphism, metamorphism, obfuscation, packing, anti-VM, anti-debugging, ...

Blend-in

Mutate attacks to look as anticipated activity

Communicate with already accessed networks

Follow work hours, existing behavioral patterns, ...

Avoid outliers!

Achieving Stealth

Scrub evidence

- Anti-forensics: erase traces from logs

- Modify—not disable (!)—security monitors to lie about suspicious activity

Remain hidden

- Use rootkits and other stealthy techniques

- “Living off the Land” attacks: rely on already installed software and functionality (powershell, sysinternals, WMI, system utilities, ...)

- Minimize extra network activity (better: piggy-back on existing activity)

Lateral Movement

Repeat step 1, compromise more hosts

Many more possibilities, as the attacker is now in the internal network and has gained precious information

Pivoting: gain access to other network segments

Dual/multi-homed hosts (multiple real/virtual NICs)

Hosts with point-to-point VPN connections

Even air-gapped systems (e.g., by infecting USB sticks)


Precious targets

Domain controllers and other core servers (DNS, LDAP, ...)

Networked devices: routers, firewalls, printers, ...

Researchers Solve Juniper

← → ↺ <https://www.wired.com/2015/12/researchers-solve-the-juniper-mystery-and-they-say-its-partially-the-nsas-fault/> ☆ ⋮

 **WIRED**

Researchers Solve Juniper Backdoor Mystery; Signs Point to NSA

SUBSCRIBE 🔍

BUSINESS

CULTURE

DESIGN


GEAR


SCIENCE


SECURITY


TRANSPORTATION


SHARE

 SHARE 2

 TWEET

 PIN 1

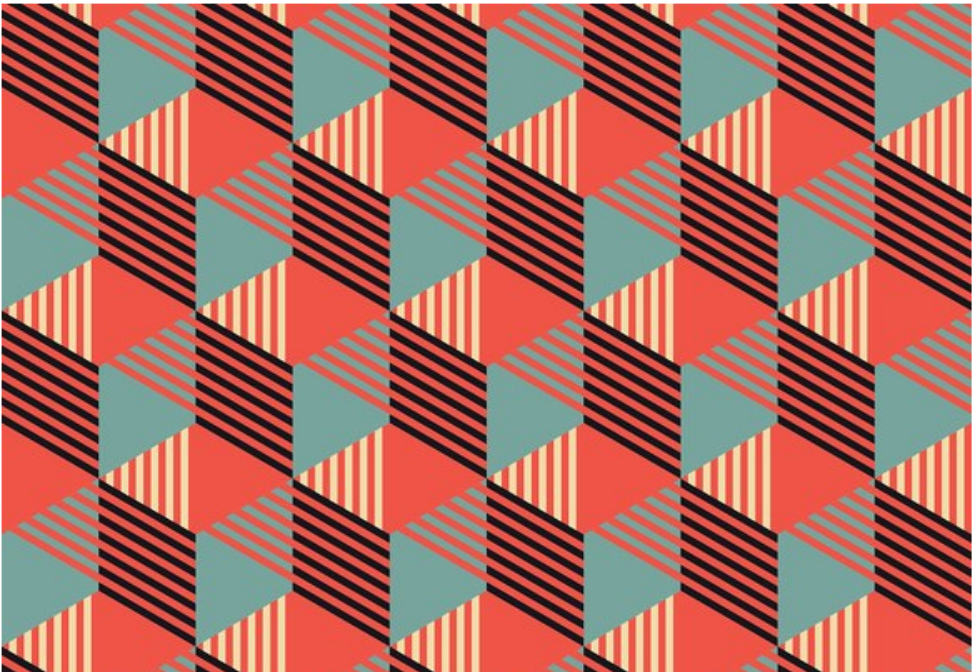
 COMMENT 17


 EMAIL

KIM ZETTER

 SECURITY 12.22.15 1:29 AM


RESEARCHERS SOLVE JUNIPER BACKDOOR MYSTERY; SIGNS POINT TO NSA






GET WIRED
Don't Let The Future Leave You Behind. Get 6 Issues For Just \$5.
SUBSCRIBE NOW


MOST POPULAR



PRODUCT REVIEW
Review: Google Pixel
15 HOURS



TRAILERS
The New *Assassin's Creed* Movie Trailer Will Take You Higher
13 HOURS



BUSINESS
Hey Silicon Valley: President Obama Has a To-Do List for You

24

Simple things that are now possible...

psexec: light-weight utility to execute processes on other systems

- Access through RPC/named SMB pipe

- In essence, same as remote ssh command execution on Linux

Pass the Hash: NTLM user authentication hashes are hard to crack, so just use them as-is

- Can be used in conjunction with psexec

SMB capture: elicit credential or hashes by impersonating an SMB server and triggering a connection

- Same approach also works for other services

Brute-force password guessing and hash cracking

Token impersonation: reuse access control tokens

- Useful if passwords or hashes are not available

Hash Harvesting

Cached hashes of users who have previously logged in

Read directly from Security Account Manager (SAM) - requires Administrator access

This default caching behavior can be disabled by administrators

Mimikatz, Isadump, ...

Dumping the local user's account database (SAM)

Contains only user accounts local to the particular machine

Sniffing or eliciting LM and NTLM dialogues

SMBRelay, Responder, Inveigh, ...

Dumping lsass.exe process memory

May include credentials of domain users/administrators (e.g., those logged in via RDP)

Windows Hashes

LM

Oldest password storage used by Windows – trivial to crack (!)

Turned off by default starting in Windows Vista/Server 2008

Might still linger in a network if older systems are still in use

NTHash (aka NTLM aka NT)

Used for password storage on modern Windows systems

Stored in the Security Account Manager (SAM) database and in the Domain Controller's NTDS.dit database

NTLMv1 (aka Net-NTLMv1)

Challenge/response protocol used for authentication

Uses both the LM and NTHash

NTLMv2 (aka Net-NTLMv2)

Default in Windows since Windows 2000

Pass the Hash and Relaying

Pass the Hash: authenticate to a remote server or service using the NTLM or LM hash of a password

- Instead of requiring the plaintext password

- No need for brute-force guessing

Authentication protocol weakness: password hash remains static across sessions

- LM or NTLM authentication – fixed in NTLMv2

Relaying: intercept authentication attempts and relay captured hashes to *other* machines

- No need to crack NTLMv1 or NTLMv2 hashes

NTLMv1/v2 Challenge/Response



SMB Relay Attack

Opportunistic attack: wait for someone (e.g., automated inventory scanner) to connect to the attacker's machine



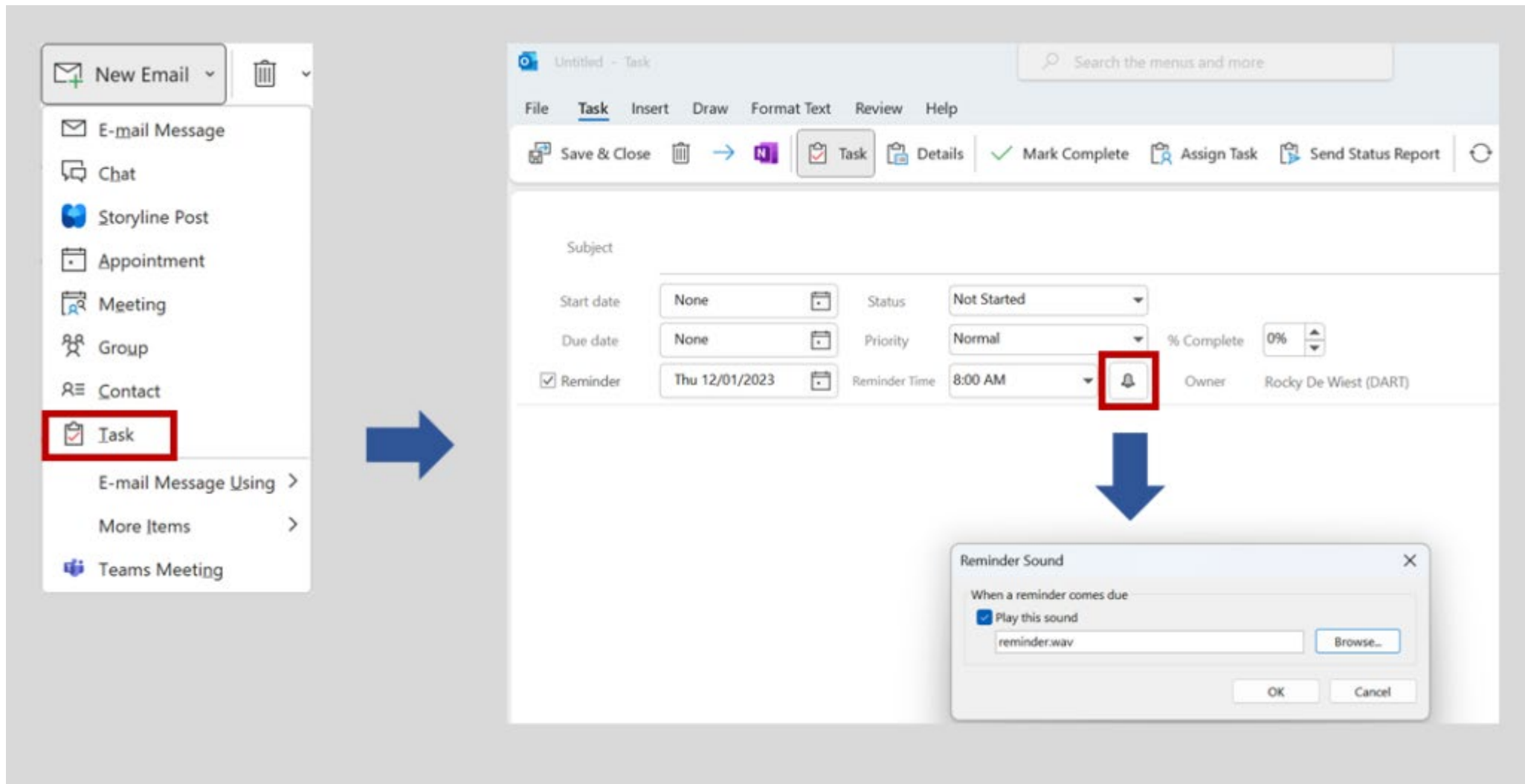


Understanding the vulnerability (CVE-2023-23397)

CVE-2023-23397 is a critical elevation of privilege vulnerability in Microsoft Outlook on Windows. It is exploited when a threat actor delivers a specially crafted message to a user. This message includes the *PidLidReminderFileParameter* extended Messaging Application Programming Interface (MAPI) property, which must be set to a Universal Naming Convention (UNC) path share on a threat actor-controlled server (via Server message block (SMB)/transmission control protocol (TCP) port 445).

In exploitation of CVE-2023-23397, threat actors can specify the value for the *PidLidReminderFileParameter* in specially crafted messages to trigger a Net-NTLMv2 hash leak to threat actor-controlled servers.

The user does not need to interact with the message: if Outlook on Windows is open when the reminder is triggered, it allows exploitation. The connection to the remote SMB server sends the user's Net-NTLMv2 hash in a negotiation message, which the threat actor can either a) relay for authentication against other systems that support NTLMv2 authentication or b) perform offline cracking to extract the password. As these are NTLMv2 hashes, they cannot be leveraged as part of a Pass-the-Hash technique. All versions of Microsoft Outlook on Windows are impacted. Outlook for Android, iOS, Mac, and users



A Few Useful Tools

Meterpreter (Metasploit)

Armitage: GUI front-end for Metasploit

mimikatz: scans memory for plaintexts passwords, hash, PIN code and Kerberos tickets

Can also perform pass-the-hash, pass-the-ticket, ...

Cain & Abel: password recovery for Windows systems using

Sniffing, cracking, capturing, recovery from caches

Powershell, sysinternals, local utilities, ...