# Detecting Pass-the-Ticket

## Pass-the-Ticket

Pass-the-Ticket (PtT) is a lateral movement technique used by attackers to move laterally within a network by abusing Kerberos TGT (Ticket Granting Ticket) and TGS (Ticket Granting Service) tickets. Instead of using NTLM hashes, PtT leverages Kerberos tickets to authenticate to other systems and access network resources without needing to know the users' passwords. This technique allows attackers to move laterally and gain unauthorized access across multiple systems.

### Attack Steps:

- The attacker gains administrative access to a system, either through an initial compromise or privilege escalation.
- The attacker uses tools such as Mimikatz or Rubeus to extract valid TGT or TGS tickets from the compromised system's memory.

```
PS C:\Users\JENNY_HICKMAN\Downloads> .\Rubeus.exe monitor /interval:30

  ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.6.1

[*] Action: TGT Monitoring
[*] Monitoring every 30 seconds for new TGTs


[*] 3/14/2021 5:43:22 PM UTC - Found new TGT:

  User                  :  Administrator@LAB.INTERNAL.LOCAL
  StartTime             :  3/14/2021 7:41:45 PM
  EndTime               :  3/15/2021 5:41:45 AM
  RenewTill             :  3/21/2021 7:41:45 PM
  Flags                 :  name_canonicalize, pre_authent, initial, renewable, forwardable
  Base64EncodedTicket   :

    doIF4DCCBdygAwIBBaEDAgEWooIEzjCCBMphggTGMIIEwqADAgEFoRQbEkxBQi5JTlRFUk5BTC5MT0NBTKInMCWgAwIBAqEeMBwb
    BmtyYnRndBsSTEFCLkJOVEVSTkFMLkxPQ0FMo4IEejCCBHagAwIBEqEDAgECooIEaASCBGTcSpzAfratrgeuxc4tniCBFJdJ98B0
    mbsdOyiXUsFonpwpQO5goWuMI+O/lJbT49Lv0OmKHLVPbT6jb4fgURXkF4F97zqxwY6AgxDwpRgmuBX+7P363rEtLAOFXs7TYilT
    IrWdIIL/CamXDZWKsK6zV1DGvx2ZcQBE51PCldQNksKmFO6z06JuMGC71e+u8yg9paYRNXpORNHDQ/luI8LMKRaJN/Q8FrWqkzYj
    gfj8dNcmKHvxFif6MVRzl/dlyi8pJxSKA3Ib4FLTd/wXNqlmb7LBaoNReR+mSZz30WBTMrzvtmIHYI3HqZAe4W9tRaRs/15sLnpI
    E1ntHl9E9vdEF1fIQanROI8j4g2tUUt4XFoqJw6EZu7zWYqUZJds4NMcyAIsHX2ITXNfS3f7c592IzJq8a8L6y66wMd5tH6DjtNC
    pqSHo+R5PoAzEQO1VNz6HH9ju563O41tIE/OQJWlL6HJvlsKQ26v1rIROvT7H1AVb8ZG/7IznD1OL/m4CgN3eWyzw2KvLWaPQ9yA
    8R9zv7RW2mnHII7JqKEAv7OmL/Kg2kw8O9qfOQiH/TGEngWickKc7ZWhoaVvMJ/YLsqmDV1T1p8Zq38MgcuZH8uKQaqh48cLNQA1
    I5whEtcvEdEy8j4TDXTUsn19KNrphlLOil/KFsIoQ2I+J3k4A57x9ZIV4JRnLvs/BmjxAYeUQInGGCf+MpwRlCrhbXDkRZKBOlUN
    qBcHwpOh4cga7kpcDVFRtExXyrGSSnv+VdClshHALH+VJUcolaeATmZC2IneAbXNQJhZJscUQ8umQqzQKLNWOEzt4x9O4YLXR/gz
    gkbAOdtwv8diJL7kDhQXRXoWqdAmDW/Zkh/rahAGl4MJoOgmLkvU4LJXArXfn61AiLu9FDDJfDDuw3id6YyxQgbUHjpLOqW5lLvH
    0UweR9br31y4pJQwyZgHbbsaQE1j5v+C2E/C/aCV/e6/gwugTOSWfMcKzIpC3PiLqVd15ShX/5Z/YDkBG5LctiQ9DqgeILxjrinV
    2JqX55+hkjA2V+PsasYD+7E2cL90EPBQaeEsS/7nF2TARGkJLhP5St4BENTwAPJod45Vpb3aSuIkZ2VCjmVFH+Nypkt5ki8lcWay
```

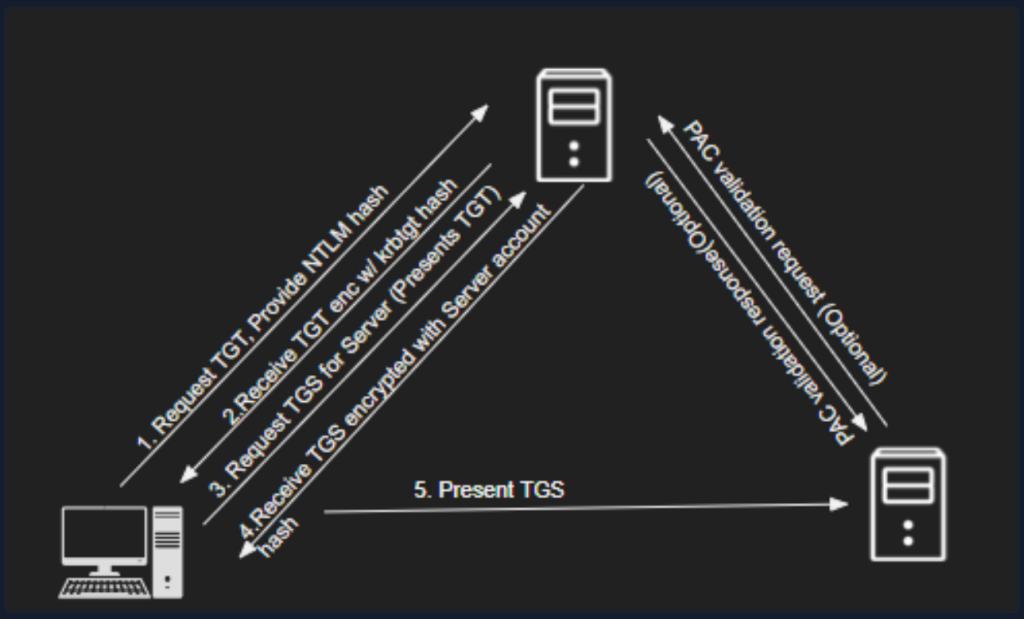
- The attacker submits the extracted ticket for the current logon session. The attacker can now authenticate to other systems and network resources without needing plaintext passwords.

```
C:\Users\JENNY_HICKMAN\tools>Rubeus.exe ptt /ticket:doIF4DCCBdygAwIBBaEDAgEWooIEzjCCBMphggTGMIIEwqADAgE
FoRQbEkxBQi5JTlRFUk5BTC5MT0NBTKInMCWgAwIBAqEeMBwbBmtyYnRndBsSTEFCLkJOVEVSTkFMLkxPQ0FMo4IEejCCBHagAwIBEq
EDAgECooIEaASCBGTcSpzAfratrgeuxc4tniCBFJdJ98B0mbsdOyiXUsFonpwpQO5goWuMI+O/lJbT49Lv0OmKHLVPbT6jb4fgURXkF
4F97zqxwY6AgxDwpRgmuBX+7P363rEtLAOFXs7TYilTIrWdIIL/CamXDZWKsK6zV1DGvx2ZcQBE51PCldQNksKmFO6z06JuMGC71e+u
8yg9paYRNXpORNHDQ/luI8LMKRaJN/O8FrWqkzYjgfj8dNcmKHvxFif6MVRzl/dlyi8pJxSKA3Ib4FLTd/wXNqlmb7LBaoNReR+mSZz

C:\Users\JENNY_HICKMAN\tools>klist

Current LogonId is 0:0x5d8438

Cached Tickets: (1)

#0>     Client: Administrator @ LAB.INTERNAL.LOCAL
        Server: krbtgt/LAB.INTERNAL.LOCAL @ LAB.INTERNAL.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
        Start Time: 3/14/2021 19:41:45 (local)
        End Time:   3/15/2021 5:41:45 (local)
        Renew Time: 3/21/2021 19:41:45 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:
```

### Kerberos Authentication Process

Kerberos is a network authentication protocol used to securely authenticate users and services within a Windows Active Directory (AD) environment. The following steps occur in the Kerberos authentication process:

- The user (client) initiates the authentication process by requesting a Ticket Granting Ticket (TGT) from the Key Distribution Center (KDC), typically part of the Active Directory domain controller.
- The KDC verifies the user's identity (usually through a password) and issues a TGT encrypted with the user's secret key. The TGT is valid for a specific period and allows the user to request service tickets without needing to re-authenticate.
- The client sends a service ticket request (TGS-REQ) to the KDC for the service using the TGT obtained in the previous step.
- The KDC validates the client's TGT and, if successful, issues a service ticket (TGS) encrypted with the service account's secret key and containing the client's identity and a session key. The client then receives the service ticket (TGS) from the KDC.
- The client connects to the server and sends the TGS to the server as part of the authentication process.




## Related Windows Security Events

During user access to network resources, several Windows Event Logs are generated to record the logon process and related activities.

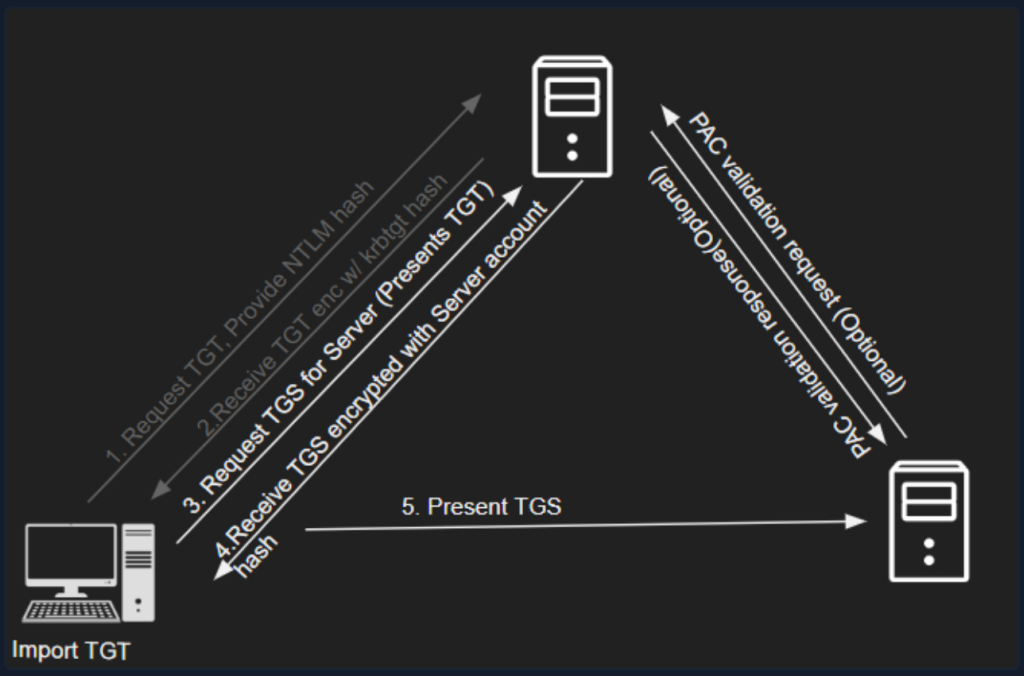
- Event ID 4648 (Explicit Credential Logon Attempt): This event is logged when explicit credentials (e.g., username and password) are provided during logon.
- Event ID 4624 (Logon): This event indicates that a user has successfully logged on to the system.
- Event ID 4672 (Special Logon): This event is logged when a user's logon includes special privileges, such as running applications as an administrator.
- Event ID 4768 (Kerberos TGT Request): This event is logged when a client requests a Ticket Granting Ticket (TGT) during the Kerberos authentication process.
- Event ID 4769 (Kerberos Service Ticket Request): When a client requests a Service Ticket (TGS Ticket) to access a remote service during the Kerberos authentication process, Event ID 4769 is generated.

| _time | RecordNumber | EventCode | name | ComputerName | user | src_ip | Logon_Type |
|---|---|---|---|---|---|---|---|
| 2023-07-28 10:42:57 | 1569 | 4648 | A logon was attempted using explicit credentials | BLUE.corp.local | RAUL_LYNN | | |
| 2023-07-28 10:42:57 | 1570 | 4624 | An account was successfully logged on | BLUE.corp.local | RAUL_LYNN | 127.0.0.1 | 2 |
| 2023-07-28 10:42:57 | 1571 | 4624 | An account was successfully logged on | BLUE.corp.local | RAUL_LYNN | 127.0.0.1 | 2 |
| 2023-07-28 10:42:57 | 1572 | 4672 | Special privileges assigned to new | BLUE.corp.local | RAUL_LYNN | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | logon | | | | |
| 2023-07-28 10:42:57 | 67290 | 4768 | A Kerberos authentication ticket (TGT) was requested | DC01.corp.local | RAUL_LYNN | ::ffff:10.10.0.101 | |
| 2023-07-28 10:42:57 | 67291 | 4769 | A Kerberos service ticket was requested | DC01.corp.local | RAUL_LYNN@CORP.LOCAL | ::ffff:10.10.0.101 | |
| 2023-07-28 10:42:57 | 67292 | 4768 | A Kerberos authentication ticket (TGT) was requested | DC01.corp.local | RAUL_LYNN | ::ffff:10.10.0.101 | |
| 2023-07-28 10:42:57 | 67293 | 4769 | A Kerberos service ticket was requested | DC01.corp.local | RAUL_LYNN@CORP.LOCAL | ::ffff:10.10.0.101 | |
| 2023-07-28 10:42:57 | 67294 | 4672 | Special privileges assigned to new logon | DC01.corp.local | RAUL_LYNN | | |
| 2023-07-28 10:42:57 | 67295 | 4624 | An account was successfully logged on | DC01.corp.local | RAUL_LYNN | 10.10.0.101 | 3 |

### Pass-the-Ticket Detection Opportunities

Detecting Pass-the-Ticket attacks can be challenging, as attackers are leveraging valid Kerberos tickets instead of traditional credential hashes. The key distinction is that when the Pass-the-Ticket attack is executed, the Kerberos Authentication process will be partial. For example, an attacker imports a TGT ticket into a logon session and requests a TGS ticket for a remote service. From the Domain Controller perspective, the imported TGT was never requested before from the attacker's system, so there won't be an associated Event ID 4768.




This approach can be converted into the following Splunk detection: Look for `Event ID 4769 (Kerberos Service Ticket Request)` or `Event ID 4770 (Kerberos Service Ticket was renewed)` without a prior `Event ID 4768 (Kerberos TGT Request)` from the same system within a specific time window.

Another approach is looking for mismatches between Service and Host IDs (in `Event ID 4769`) and the actual Source and Destination IPs (in `Event ID 3`). Note that there will be several legitimate mismatches, but unusual hostnames or services should be investigated further.

Also, in cases where an attacker imports a TGS ticket into the logon session, it is important to review `Event ID 4771 (Kerberos Pre-Authentication Failed)` for mismatches between Pre-Authentication type and Failure Code. For example, `Pre-Authentication type 2 (Encrypted Timestamp)` with `Failure Code 0x18 (Pre-authentication information was invalid)` would indicate that the client sent a Kerberos AS-REQ with a pre-authentication encrypted timestamp, but the KDC couldn't decrypt it.

It is essential to understand that these detection opportunities should be enhanced with behavior-based detection. In other words, context is vital. Looking for Event IDs `4769`, `4770`, or `4771` alone will likely generate many false positives. Correlate the event logs with user and system behavior patterns, and consider whether there are any suspicious activities associated with the user or system involved in the logs.

Let's now navigate to the bottom of this section and click on "Click here to spawn the target system!". Then, access the Splunk interface at http://[Target IP]:8000 and launch the Search & Reporting Splunk application. The vast majority of searches covered from this point up to end of this section can be replicated inside the target, offering a

majority of searches covered from this point up to end of this section can be replicated inside the target, offering a more comprehensive grasp of the topics presented.

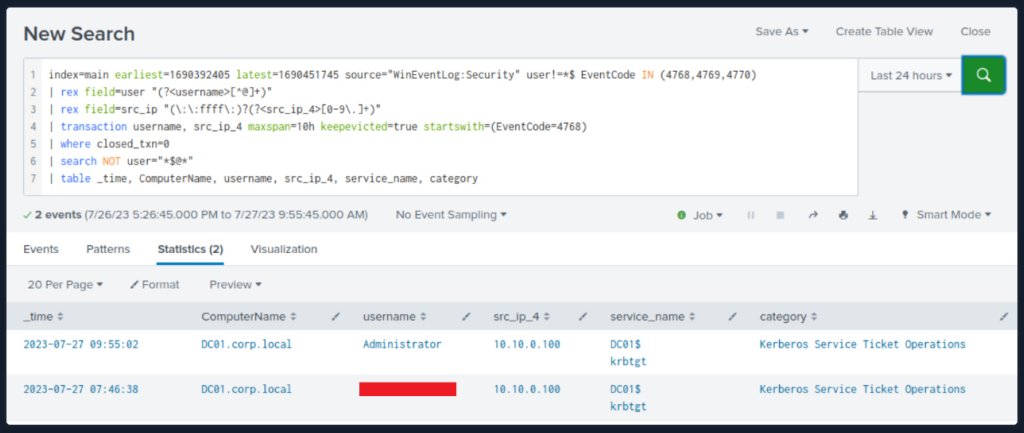
## Detecting Pass-the-Ticket With Splunk

Now let's explore how we can identify Pass-the-Ticket, using Splunk.

**Timeframe**: `earliest=1690451665 latest=1690451745`

Detecting Pass-the-Ticket

```
index=main earliest=1690392405 latest=1690451745 source="WinEventLog:Security" user!=*$ EventCode I
| rex field=user "(?<username>[^@]+)"
| rex field=src_ip "(\:\:ffff\:)?(?<src_ip_4>[0-9\.]+)"
| transaction username, src_ip_4 maxspan=10h keepevicted=true startswith=(EventCode=4768)
| where closed_txn=0
| search NOT user="*$@*"
| table _time, ComputerName, username, src_ip_4, service_name, category
```




**Search Breakdown**:

- `index=main earliest=1690392405 latest=1690451745 source="WinEventLog:Security" user!=*$ EventCode IN (4768,4769,4770)`: This command filters events from the `main` index that fall within the specified time range. It selects events from the `WinEventLog:Security` source, where the `user` field does not end with a dollar (`$`) and the `EventCode` is one of `4768`, `4769`, or `4770`.
- `| rex field=user "(?<username>[^@]+)"`: This command extracts the `username` from the `user` field using a regular expression. It assigns the extracted value to a new field called `username`.
- `| rex field=src_ip "(\:\:ffff\:)?(?<src_ip_4>[0-9\.]+)"`: This command extracts the IPv4 address from the `src_ip` field, even if it's originally recorded as an IPv6 address. It assigns the extracted value to a new field called `src_ip_4`.
- `| transaction username, src_ip_4 maxspan=10h keepevicted=true startswith=(EventCode=4768)`: This command groups events into transactions based on the `username` and `src_ip_4` fields. A transaction begins with an event that has an `EventCode` of `4768`. The `maxspan=10h` parameter sets a maximum duration of `10` hours for a transaction. The `keepevicted=true` parameter ensures that open transactions without an ending event are included in the results.
- `| where closed_txn=0`: This command filters the results to include only open transactions, which do not have an ending event.
- `| search NOT user="*$@*"`: This command filters out results where the `user` field ends with an asterisk (`*`) and contains an at sign (`@`).
- `| table _time, ComputerName, username, src_ip_4, service_name, category`: This command displays the specified fields in a table format.

VPN Servers

⚠ **Warning:** Each time you "Switch", your connection keys are regenerated and you must re-download your VPN connection file.

All VM instances associated with the old VPN Server will be terminated when switching to a new VPN server.

Existing PwnBox instances will automatically switch to the new VPN server.

US Academy 3 Medium Load

PROTOCOL

UDP 1337 TCP 443

DOWNLOAD VPN CONNECTION FILE

**Connect to Pwnbox**

Your own web-based Parrot Linux instance to play our labs.

Pwnbox Location

UK 143ms

Terminate Pwnbox to switch location

Start Instance

∞ / 1 spawns left

Waiting to start...

Enable step-by-step solutions for all questions

## Questions

Answer the question(s) below to complete this Section and earn cubes!

Download VPN Connection File

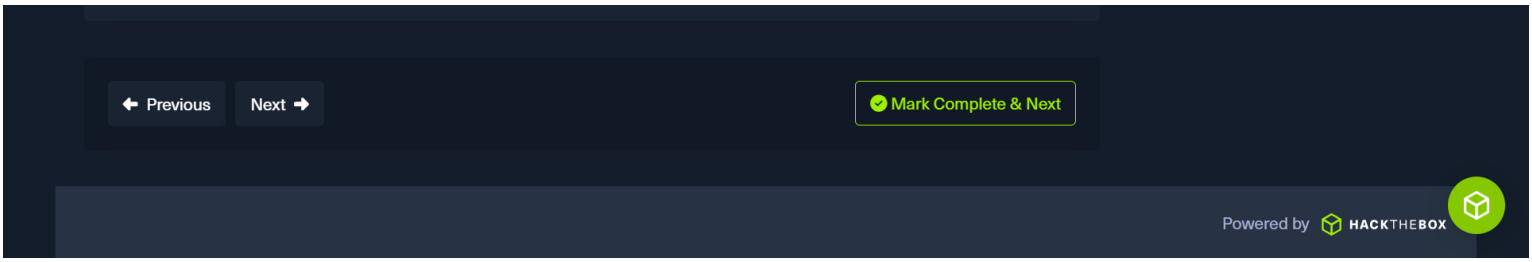
Target(s): Click here to spawn the target system!

+ 1 Execute the Splunk search provided at the end of this section to find all usernames that may be have executed a Pass-the-Ticket attack. Enter the missing username from the following list as your answer.

Administrator, _

YOUNG_WILKINSON

Submit


Previous
Next
Mark Complete & Next
Powered by
HACKTHEBOX