One Token to Rule Them All: Post-Exploitation Fun in Windows Environments

Luke Jennings

21st May 2008

Introduction

The Increasing Importance of Post-Exploitation

- Microsoft Technology is Pervasive
- Security has improved significantly since their Trustworthy Computing Initiative
- Consequently, Post-Exploitation has become increasingly important

Introduction

The Potential for Token Abuse for Post-Exploitation

- This has been addressed to some extent. Metasploit is a good example
- From a penetration tester’s perspective, not much has been done with windows access tokens
- They are integral to Microsoft's whole concept of single sign-on within an active directory environment
- Difficult to convince clients of the importance of keeping their administrators' desktops as secure as their servers etc

Windows Access Tokens: An Overview

- Not well understood
- Processes have a primary token associated which dictates their privileges
- Threads by default use this but can impersonate other users temporarily through impersonation tokens
- Tokens have four different security levels (delegation most interesting) : -
    - Anonymous
    - Identification
    - Impersonation
    - Delegation
- Interactive logons result in delegation level tokens being created
- Non-interactive logons normally result in impersonation level tokens, but can produce delegation tokens if a service is trusted for delegation e.g. EFS file server

Token Abuse

- Tokens may be present on compromised systems that allow privilege escalation of some sort

- Domain Privilege Escalation
  - If domain user tokens are available then other systems may be accessible using these
  - e.g. compromising a DBA's workstation might allow his token to be used to gain access to sensitive database servers that are secure from direct compromise

- Local Privilege Escalation
  - Under some circumstances, presence of impersonation level tokens may allow local privilege escalation
  - E.g. compromising an SQL Server instance running as an unprivileged service account. If an administrator connects using windows authentication then their access token can be used to escalate privileges to local Administrator

Requirements From a Penetration Tester's Perspective

- Enumerate available tokens from a compromised system

- Perform common post-exploitation tasks using specific tokens
  - Execute processes
  - Force remote connections in order to capture challenge/response LANMAN/NTLM hashes
  - Add a user to a host, a user to a group etc

Introducing Incognito

An Overview

- Why the name?
  - Because it allows you to assume another’s identity

- Functionality
  - List available tokens by unique username or group
  - Create processes with a specified token e.g. cmd.exe
  - Snarf challenge/response hashes of all tokens
  - Attempt to add a user to a host with all tokens
  - Attempt to add a user to a group on a host with all tokens
  - Can be used remotely with communication over named pipes (pwdump style)

Introducing Incognito

Enumerating Tokens

- Uses NtQuerySystemInformation() API call to enumerate all handles on the system

```
NTSTATUS WINAPI NtQuerySystemInformation( SYSTEM_INFORMATION_CLASS
   SystemInformationClass, PVOID SystemInformation, ULONG
   SystemInformationLength, PULONG ReturnLength );
```

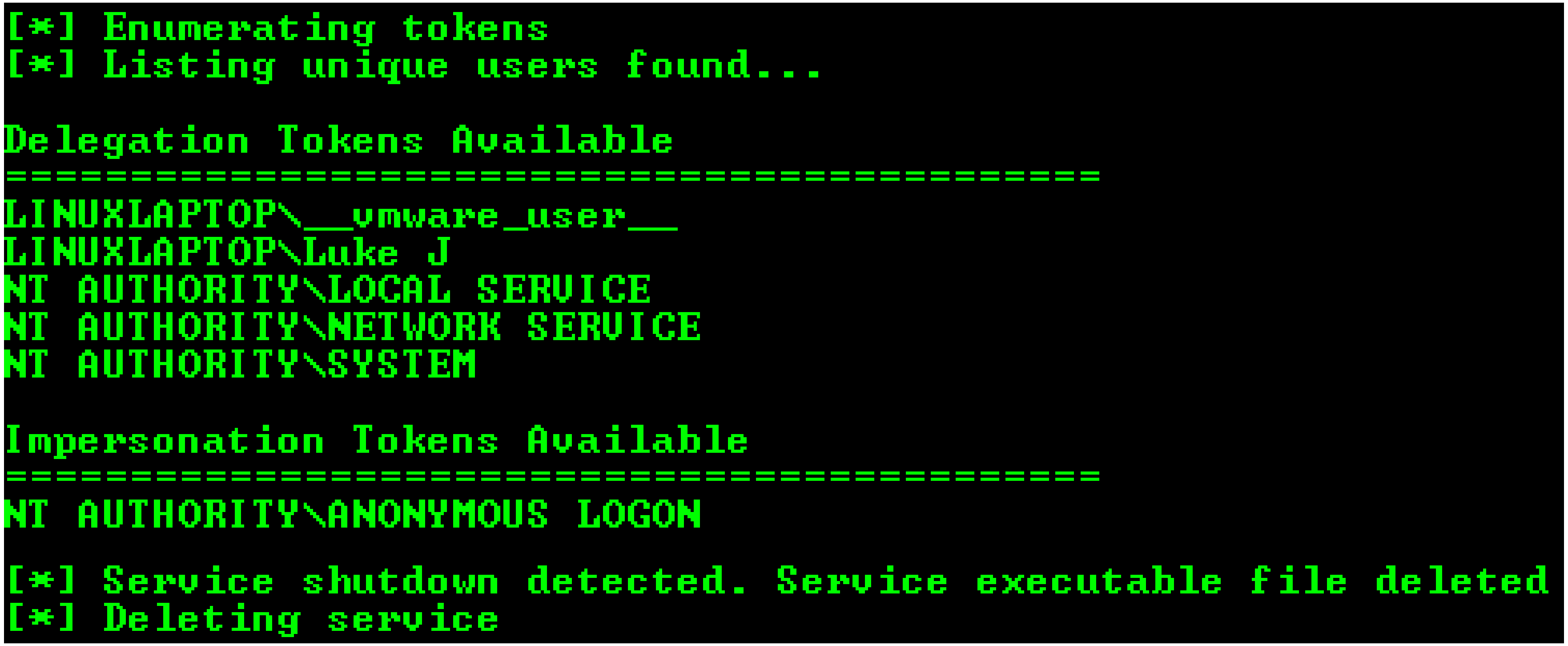
- NtQueryObject() API call is then used to determine which handles are access tokens

```
NTSTATUS NtQueryObject( HANDLE Handle, OBJECT_INFORMATION_CLASS
   ObjectInformationClass, PVOID ObjectInformation, ULONG
   ObjectInformationLength, PULONG ReturnLength );
```

- Various other API calls, such as GetTokenInformation() and LookupAccountSid() are then used to discover information about the tokens such as the username and groups associated and security level (impersonation, delegation etc)

Introducing Incognito

Enumerating Tokens



```
[*] Enumerating tokens
[*] Listing unique users found...

Delegation Tokens Available
============================================
LINUXLAPTOP\__vmware_user__
LINUXLAPTOP\Luke J
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM

Impersonation Tokens Available
============================================
NT AUTHORITY\ANONYMOUS LOGON

[*] Service shutdown detected. Service executable file deleted
[*] Deleting service
```

Introducing Incognito

Creating Processes

- Processes can be created using tokens present on the system, which is implemented via the CreateProcessAsUser() API call, which allows a token handle to be specified.

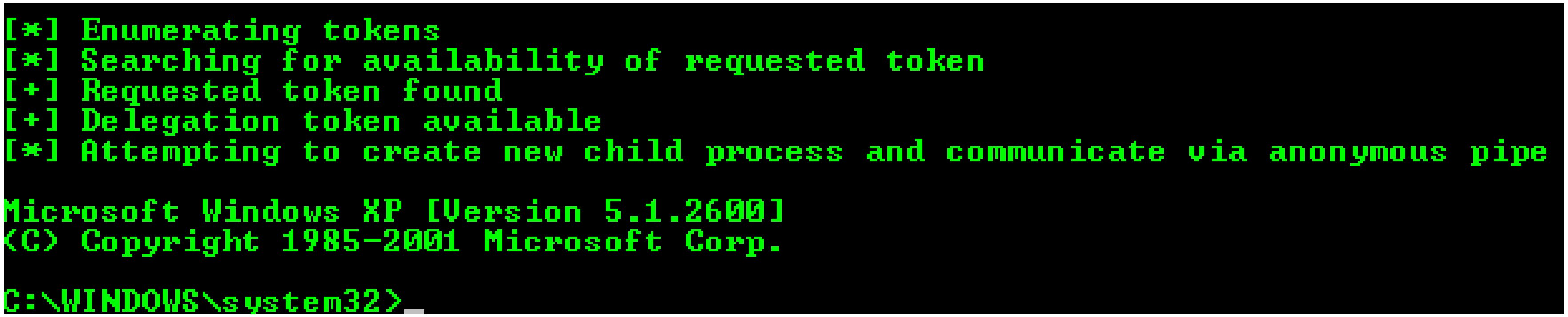
```
BOOL WINAPI CreateProcessAsUser( HANDLE hToken, LPCTSTR
    lpApplicationName, LPTSTR lpCommandLine, LPSECURITY_ATTRIBUTES
    lpProcessAttributes, LPSECURITY_ATTRIBUTES lpThreadAttributes,
    BOOL bInheritHandles, DWORD dwCreationFlags, LPVOID
    lpEnvironment, LPCTSTR lpCurrentDirectory, LPSTARTUPINFO
    lpStartupInfo, LPPROCESS_INFORMATION lpProcessInformation );
```

- This is a useful post-exploitation feature. A common example would be to create a new instance of cmd.exe to gain access to a command shell running under the context of the specified token

MWR InfoSecurity

Introducing Incognito

Creating Processes



```
[*] Enumerating tokens
[*] Searching for availability of requested token
[+] Requested token found
[+] Delegation token available
[*] Attempting to create new child process and communicate via anonymous pipe

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>_
```

Introducing Incognito

Snarfing Hashes

- With the aid of a sniffer, LANMAN/NTLM challenge response hashes can be snarfed by forcing connections to remote servers with delegation level tokens. This makes use of the WNetAddConnection2() API call and iterates through all available tokens

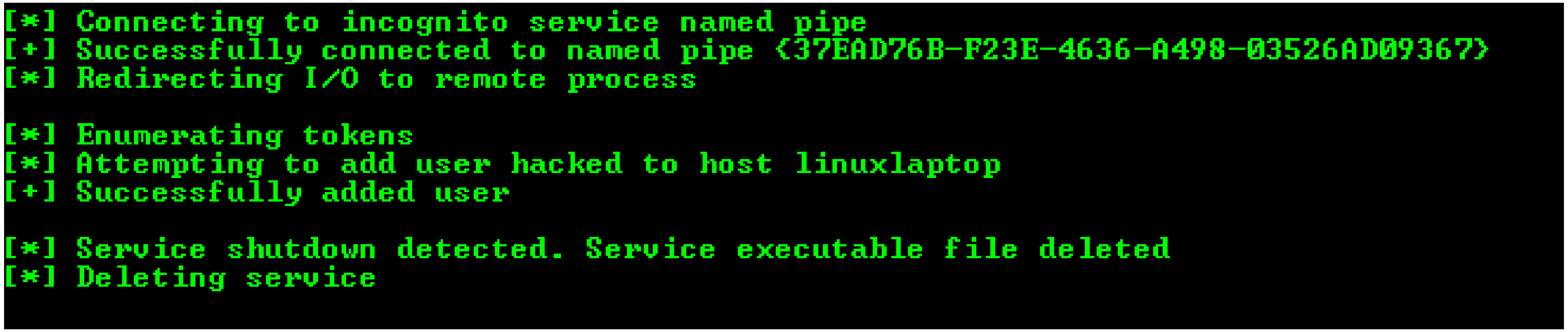
```
DWORD WNetAddConnection2( LPNETRESOURCE lpNetResource, LPCTSTR
    lpPassword, LPCTSTR lpUsername, DWORD dwFlags );
```

Introducing Incognito

Adding Users to Hosts

- Incognito can use the Network Management set of API calls to perform some common user and group management operations.

- The NetUserAdd(), NetGroupAddUser() and NetLocalGroupAddMembers() API calls allow users to be added to hosts, users to be added to global groups on hosts and users to be added to localgroups on hosts, respectively.



```
[*] Connecting to incognito service named pipe
[+] Successfully connected to named pipe {37EAD76B-F23E-4636-A498-03526AD09367}
[*] Redirecting I/O to remote process

[*] Enumerating tokens
[*] Attempting to add user hacked to host linuxlaptop
[+] Successfully added user

[*] Service shutdown detected. Service executable file deleted
[*] Deleting service
```

Incognito Demo

Unsuspected 0wnage

- I'm currently logged off. My account is safe...right?
  - It is supposed to be as all tokens should be cleaned up when a session ends (for example, when you log off in the case of an interactive login)
  - Whilst researching this topic I stumbled upon a case where this isn't true
  - On unpatched systems the tokens for interactive logins appear to persist after logoff.
  - Tokens are only reported as impersonation level tokens through the W32 API but if you try to use them it turns out they can access network resources!
  - Tokens do not disappear until reboot

- **Therefore, if you logged into "that ropey old development system" briefly a month ago as a Domain Admin then your entire Windows Domain just got 0wned! :)**

Unsuspected 0wnage Demo

Unsuspected 0wnage – Terminal Services

- How do you use Terminal Services?
    - When using Terminal Services it is possible to close the window without logging off
    - Doing this leaves your session logged on
    - It allows you to leave programs running and reconnect at a later date
        - Some people use this functionality
        - Some people intend to use this functionality and then forget and do not reconnect for a long time
        - Some people don’t intend to use this functionality and just click close because that it is easier than logging out!

    - **However you use it, this can lead to sensitive tokens persisting for much longer periods of time**

Metasploit Integration

- I am a big fan of the Metasploit project

- Whilst writing Incognito it occurred to me that Incognito's functionality would be very useful as a Meterpreter module
  - It would allow Incognito's features to be utilised directly through an exploit launched from Metasploit
  - The Meterpreter runs as a thread and so implementing the functionality to cause that thread to impersonate another token would automatically allow all of the existing Meterpreter functionality to be used under the context of other tokens
  - The automated exploitation functionality offered by Metasploit could be used to perform certain actions automatically on every compromised host across a large network
  - e.g. Tell Metasploit to try to compromise every windows system on a particular address range, snarf the hashes of all the available tokens and try to add a new domain administrator.

MWR InfoSecurity

Incognito Meterpreter Module Demo

Precision Strikes

Locating targets that might house interesting tokens

- When using Incognito’s functionality for domain privilege escalation it is a useful general tool for “squeezing all the juice” out of a compromise
- However, a penetration test may often be highly targeted e.g. attacking an organisation’s financial SQL Server databases
- These systems may be very well protected and so safe from direct compromise. However, other systems that house tokens which have access to these systems may be less well protected e.g. the DBA’s desktop
- It is likely that it would not be feasible to make the effort to compromise as much of a large network as possible in a brute force approach, hoping compromised systems might house these sensitive tokens.
- It would be nice if it were simple to locate systems that house these tokens before making the effort to compromise them…..a precision strike

Locating Tokens

- Incognito comes with a supplementary tool which offers the ability to enumerate what tokens might be present on a system before compromising it
- The tool is called find_token and it can sweep a network remotely in order to search for particular tokens
- This is achieved via the API call NetWkstaUserEnum() which can be used to enumerate the currently logged on users
- To sweep an entire domain using this API call, it is only necessary to have a standard domain user account.
- It is not perfect since it lists users that have logged on previously but have since logged off and so can produce false positives. However, this can be very useful for unpatched hosts that do not clean up their tokens on logoff!

Find_token Demo

Targeted Penetration Testing With Incognito

A Basic Methodology

1) Determine targets (e.g. critical database servers)
2) Conduct conventional penetration testing
3) If penetration attempts fail then attempt to enumerate who has access to the system with standard techniques (e.g. members of local administrators group)
4) Locate other systems that might currently house the relevant users' tokens (with find_token’s functionality)
5) Attempt to penetrate these systems
6) If present, use tokens from compromised hosts to compromise main targets

Defence

- Limiting the use of privileged accounts helps reduce exposure (e.g. run as)
- Use "Account is sensitive and cannot be delegated" option in active directory for highly privileged accounts (doesn't work with interactive sessions)
- Don't have ropey old boxes lying around! In particular, secure your administrators' desktops
- Resources are only as secure as the weakest system that any account that can be used to access them is currently logged into. Therefore, it is important to have strong policies governing security requirements for systems in order for sensitive accounts to be used to access them.
- Use separate administrative accounts for accessing development/test systems that are likely to be more prone to compromise than their production system counterparts

References

Whitepaper available from: -

“Security Implications of Windows Access Tokens”

http://www.mwrinfosecurity.com/publications/

Incognito on Sourceforge: -

http://sourceforge.net/projects/incognito

Metasploit Framework Development Trunk: -

http://metasploit.com/svn/framework3/trunk/