Disabling NTLM Authentication Guide – part 2 – Logs

willssysadmintechblog.wordpress.com/2023/08/23/disabling-ntlm-authentication-guide-part-2

August 23, 2023

Part 1: Disabling NTLM Authentication Guide – part 1 – Prerequisites

Part 3: Disabling NTLM Authentication Guide – part 3 – Migrating to Kerberos

Logs

In this section I'm going to go over the logs you'll want to have quick access to. These logs are invaluable for:

- Providing admins with reports of what computers are using NLTM authenticaiton
- Real-time log analysis for admins and users that are testing services
- Verifying Kerberos is working properly

For log analysis I'll post some OpenSearch configuration snippets, but this configuration could be adapted to another log aggregation solution or a script that searches logs. When I post OpenSearch snippets, I'm assuming you're parsing fields out of your Windows logs with Logstash. If you're not doing that then OpenSearch/ELK is a bit less useful.

NTLM Log Generation

First, you want to see what systems are using NTLM authentication, specifically, NTLM passthrough authentication. NTLM passthrough really going to be used exclusively, except for NTLM authentication done internally on a domain controller (DC), which we don't care about. The following security policy settings generate NTLM logs. I recommend setting these on all systems, or at least the "domain" setting.

Security Settings / Local Policies / Security Options

- Network security: Restrict NTLM: Audit NTLM authentication in this domain
  - DC focused logs, but will also generate logs on member computers
  - DC has received NTLM authentication request. All domain account NTLM auth requests will end up at the DC at some point to validate credentials. These logs are the most informative of the three.
  - EventID 8004
- Network security: Restrict NTLM: Audit incoming NTLM traffic
  - Computer has received an NTLM passthrough authentication request
  - Contains information about the process that received the request, but this isn't very helpful honestly.
- Network security: Restrict NTLM: Outgoing NTLM traffic to remote servers
  Can set to "Audit all" to generate audit events only

These logs will go in the log channel *Microsoft-Windows-NTLM/Operational*. Each of these settings has a counterpart "enforcement" setting sitting right next to it. If you enable those settings you will be blocking NTLM authentication. That's what we're moving towards, not what we're starting with. The "enforcement" settings create comparable logs to the "audit" logs we want to generate. These audit logs have the following formats:

Network security: Restrict NTLM: Audit NTLM authentication in this domain

EventID 8004 from DCs, 8003 from member servers (Block logs are 4004 & 4003)

```
Domain Controller Blocked Audit: Audit NTLM authentication to this domain
controller.
Secure Channel name: <SERVERNAME>
User name: <USERNAME>
Domain name: <DOMAIN>
Workstation name: <ORIGINATING WORKSTATION>
Secure Channel type: 2
```

Network security: Restrict NTLM: Audit incoming NTLM traffic

EventID 8002 (Block log is 4002)

```
NTLM server blocked audit: Audit Incoming NTLM Traffic that would be blocked
Calling process PID: 1344
Calling process name: C:\Windows\System32\svchost.exe
Calling process LUID: 0x3E4
Calling process user identity: <COMPUTER ACCOUNT OF SERVER I think>
Calling process domain identity: <DOMAIN>
Mechanism OID: (NULL)
```

This log is useless.

Network security: Restrict NTLM: Outgoing NTLM traffic to remote servers

EventID 8001 (Block log is 4001)

```
NTLM client blocked audit: Audit outgoing NTLM authentication traffic that would
be blocked.
Target server: HOST/<SERVER FQDN>
Supplied user: (NULL)
Supplied domain: (NULL)
PID of client process: 2788
Name of client process: <PROCESS MAKING NTLM REQUEST>
LUID of client process: 0x4ECCC0FEF
User identity of client process: SYSTEM
Domain name of user identity of client process: NT AUTHORITY
Mechanism OID: (NULL)
```

This log has some use and can identify the process making an outgoing NTLM request. To really use it requires getting logs from member computers though, and that can be tricky and require more infrastructure devoted to logging.

Privacy Settings

NTLM Log Searching and Reporting

The 8004 log events from DCs are what we really want. We need a way to easily search them and generate reports based on them. I have a saved query in OpenSearch Dashboards (in Kibana if you're on ELK) that lets me see the audit events quickly. In OpenSearch I use the following query to get all 8004 events that we want. To aid in reporting for admins later I throw in an aggregation:

```
"query": {
  "bool": {
    "filter": [
      {
        "match_phrase": {
          "Channel": "Microsoft-Windows-NTLM/Operational"
        }
      },
      {
        "match_phrase": {
          "EventID": "8004"
        }
      },
      {
        "range": {
          "@timestamp": {
            "gte": "starttimehere",
            "lte": "endtimehere",
            "format": "strict_date_optional_time"
          }
        }
      }
    ],
  }
},
"aggregations": {
  "server": {
    "terms": {
      "field": "SChannelName.keyword",
      "size": 100000
    },
    "aggregations": {
      "client": {
        "terms": {
          "field": "WorkstationName.keyword",
          "size": 100000
        }
      },
      "user": {
        "terms": {
          "field": "UserName.keyword",
          "size": 100000
        }
      }
    }
  }
}
```

I have a PowerShell script that generates a report to send to other admins which will give them an easy view of what servers are processing NTLM authentication requests and from where. Copy the “aggregations” json output from that query and paste it into a file. Then run the following script against it:

```
1  param(
2  [Parameter(mandatory=$true)]
```

```
[String] $filepath,
[Parameter(mandatory=$true)]
[String] $reportname
)
If (-not (Test-Path $filepath)) {
Write-Host "Error file not found: $filepath"
exit
}
$file = Get-Content $filepath -Raw
$file = $file.replace("`"`"", "")
$json = ConvertFrom-Json -InputObject $file
$servers = @{}
# blank dir for report
$reportdir = Join-Path -Path $PSScriptRoot -ChildPath "$reportname"
If (Test-Path $reportdir) {
Remove-Item -Recurse $reportdir
}
# extract data from report json & write to report
ForEach ($server in $json.server.buckets) {
Write-Host " ==== Server ===="
Write-Host "$($server.key) | total auth count: $($server.doc_count)"
$servers.Add($server.key, $server.doc_count)
$serverdir = (Join-Path -Path $reportdir -ChildPath
"servers/$($server.key)")
mkdir $serverdir
Write-Host " == Clients =="
$string = ""
ForEach ($client in $server.client.buckets) {
$string += "$($client.key) | auth count: $($client.doc_count)`r`n"
}
Write-Host $string
$string > (Join-Path -Path $serverdir -ChildPath "clients.txt")
Write-Host " == Users =="
$string = ""
ForEach ($user in $server.user.buckets) {
$string += "$($user.key) | auth count: $($user.doc_count)`r`n"
}
Write-Host $string
$string >> (Join-Path -Path $serverdir -ChildPath "users.txt")
Write-Host
}
# save server data to file
$servers | Out-File -FilePath (Join-Path -Path $reportdir -ChildPath
"overview.txt")
```

59
60
61
62
63
64
65
66
67
68
69
70
71

The script will create a folder with a CSV report. Format that CSV however you want and sort it based on NTLM auth count received during the queried time period. In our case, many workstation computers were in the list in addition to servers. Most of these can be ignored if they have a low NTLM-received count. Your workstation admins will want to view the list and identify any workstations that are acting as servers in some way.

Privacy Settings
The script also makes a folder with a sub-folder for each “server”. There are 2 files in each folder. One file contains the users authenticating with NTLM, the other contains the client computers sending those NTLM auth requests.

I found these report incredibly useful to send to IT admins. It let them know which servers needed testing and what users were using NTLM to connect to their servers. I sent an NTLM authentication report to our admins every 3 weeks for nearly one year during this project. In the final weeks of the project I sent a report weekly. The logs being used here come from the domain controllers, so it will include non-Windows servers.

Part 1: Disabling NTLM Authentication Guide – part 1 – Prerequisites

Part 2: Disabling NTLM Authentication Guide – part 2 – Logs

Part 3: Disabling NTLM Authentication Guide – part 3 – Migrating to Kerberos

Part 4: Disabling NTLM Authentication Guide – part 4 – NTLM Restrictions and Testing

Part 5: Disabling NTLM Authentication Guide – part 5 – Printers and Scanners

Part 6: Disabling NTLM Authentication Guide – part 6 – RDP

Part 7: Disabling NTLM Authentication Guide – part 7 – Kerberos Logs