Simple SIEMan met a WMIman

Easy ways to add context to your logs

Craig Bowser
@shad0wtrackers
Introduction

15+ years in InfoSec

Worked mainly in DoD with some DOJ and now DOE experience

GSEC GCED CISSP, yeah!

Christian, Father, Husband, Geek, Scout Leader who also does some woodworking

To Do List > To Do Open Slots
Background

Bringing in logs from various MS Windows machines
Monitoring logons, logoffs, print activity, etc
But AD has a ton of information that I'm not using

Question: What low hanging fruit can I get that will add context and increase visibility?
Each idea will have four parts

Information: What I'm getting, how I'm getting it and where I'm using it
Each idea will have four parts

Information: What I'm getting, how I'm getting it and where I'm using it

Why is this important?
Each idea will have four parts

Information: What I'm getting, how I'm getting it and where I'm using it

Why is this important?

Suggestions/Solutions
Each idea will have four parts

Information: What I'm getting, how I'm getting it and where I'm using it

Why is this important?

Suggestions/Solutions

Level of Effort
Active Inactive Accounts

The following commands will gather all disabled accounts into an csv

Get-ADUser -Filter {Enabled -eq $false} | FT samAccountName | export-csv C:\Data\InactiveAccounts.csv -NoTypeInformation

OR

Search-ADAccount -AccountDisabled | select samAccountName | export-csv C:\Data\InactiveAccounts.csv -NoTypeInformation

Monitor for deleted accounts and add those
C:\Windows\system32>powershell "Get-ADUser -Filter {Enabled -eq $false} | select samAccountName | export-csv C:\tools\InactiveAccounts.csv -NoTypeInformation"
Active Inactive Accounts

The following commands will gather all disabled accounts into an csv

Get-ADUser -Filter {Enabled -eq $false} | Select samAccountName | Export-Csv C:\Data\InactiveAccounts.csv -NoTypeInformation

Search-ADAccount -AccountDisabled | Select samAccountName | Export-Csv C:\Data\InactiveAccounts.csv -NoTypeInformation

Why is this important?

- Attackers will use already created accounts to help hide their activity
- If the account has the access to their target, they don't need to perform privilege escalation
Active Inactive Accounts

The following commands will gather all disabled accounts into an csv

Solution

- Delete accounts as soon as they are not needed
- For occasional workers, notify the SOC regarding who has been authorized to be re-enabled
Active Inactive Accounts

The following commands will gather all disabled accounts into an csv

Get-ADUser -Filter {Enabled -eq $false} | FT samAccountName | export-csv C:\Data\InactiveAccounts.csv -NoTypeInformation

OR

Search-ADAccount -AccountDisabled | select samAccountName | export-csv C:\Data\InactiveAccounts.csv -NoTypeInformation

Level of Effort Minimum
Accounts with passwords set to never expire

Accounts that have never been accessed

Search-ADAccount –PasswordNeverExpires | export-csv C:\Data4Splunk\noexpirepassword.csv –NoTypeInformation

get-aduser -f {-not ( lastlogontimestamp -like "*" ) -and ( enabled -eq $true )} | export-csv C:\Data4Splunk\neverloggedon.csv -NoTypeInformation
C:\Windows\system32\powershell -f (Get-Aduser -f (-Not (LastLogonTimestamp -like '*')) -Properties * | Select SamAccountName, Created) -Export-Csv C:\tools\neverloggedin.csv -NoTypeInformation
Accounts with passwords set to never expire

Accounts that have never been accessed

Why is this important?

• Accounts with non-expiring passwords – ripe for brute force attacks

• Unused accounts tend to be forgotten

• Could be used as entry points into and around network
Accounts with passwords set to never expire

Accounts that have never been accessed

Solution

• Never give an account a non-expiring password
• Set up to rotate passwords periodically
• Monitor like a hawk
• If an account isn't being used, disable or delete
Accounts with passwords set to never expire

Accounts that have never been accessed

Search-ADAccount –PasswordNeverExpires | export-csv C:\Data4Splunk\noexpirepassword.csv -NoTypeInformation

get-aduser -f {-not ( lastlogontimestamp -like "*" )} -and (enabled -eq $true)} | export-csv C:\Data4Splunk\neverloggedon.csv -NoTypeInformation

Level of Effort

Minimum
Collect local accounts

wmic /failfast:on /node :@machines.txt /output:localaccounts.txt useraccount get name /format:csv

Or

Wmic /failfast:on /node:machinename USERACCOUNT WHERE "Disabled=0 AND LocalAccount=1" GET Name,Domain
Collect local accounts

Why is this important?

- Malware uses to propagate or maintain persistence
- Alert on new local accounts
- Alert when these accounts try to access domain resources
Collect local accounts

Solution

- Users shouldn’t be able to create local accounts
- List of local accounts should be short
- Investigate new accounts
- Investigate local accounts trying to access network resources
Collect local accounts

wmic /node:<remote-ip> /user:<username> useraccount list full

Or

wmic /node:machinename USERACCOUNT WHERE "Disabled=0 AND LocalAccount=1" GET Name,Domain

Level of Effort
Medium
Last Reboot Time

wmic /failfast:on /node:@machines.txt
/output:results.txt OS Get LastBootUpTime,CSName

* multiple ways to skin this cat

Also can monitor for event ID 6005
Last Reboot Time

wmic /failfast:on /node:@machines.txt /output:results.txt.txt OS Get LastBootUpTime,CSName

Why is this important?

• Help verify patches have been applied and/or configurations updated
• Also, catch unexpected reboots
Last Reboot Time

wmic /failfast:on /node:@machines.txt /output:results.txt OS Get LastBootUpTime,CSName

* multiple ways:

Solution

Use SIEM to create a report and track history
wmic /failfast:on /node:@machines.txt /output:results.txt OS Get LastBootUpTime,CSName

* multiple ways to skin this cat

Also can monitor for event ID 6005

Level of Effort

Minimum – Medium
Users who always logged on

Psloggedon -I \COMPUTER
Users who always logged on

Why this is important?
• More troubleshooting than for security
• HD can use this before escalating issues
• Can possibly show persistence if attacker is careless/stupid
Users who always logged on

Psloggedon -l \\COMPUNITER

Solution

Forced reboots minimum once per week
Users who always logged on

Psloggedon -l \COMPUTER

Level of Effort
Minimum
Get-aduser –Filter * -searchbase “OU=service, DC=you” | export-csv C:\Data\serviceaccounts.csv -NoTypeInformation

OR

-filter {name LIKE “svc*”} | export-csv C:\Data\serviceaccounts.csv -NoTypeInformation

OR

however you identify your services accounts.

| select name | export-csv C:\Data\ServiceAccounts.csv -NoTypeInformation
Split these accounts into three (3) groups:

1. Accounts that do some type of scanning.
   Few sources, many destinations

2. Accounts used for applications that phone home.
   Many sources, few destinations

3. Accounts whose use doesn't fall into 1 or 2.
   Few sources, few destinations
   Many sources, many destinations
eventcode=4624 | stats count by host EventCode
```
<table>
<thead>
<tr>
<th>host</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache</td>
<td>2</td>
</tr>
<tr>
<td>TESTPC2</td>
<td>4</td>
</tr>
<tr>
<td>TESTPC3</td>
<td>4</td>
</tr>
</tbody>
</table>
```
Service Account Activity

Get-aduser -Filter * -searchbase "OU=service, DC=you" | export-csv C:\Data\serviceaccounts.csv -NoTypeInformation

OR

Why is this important?

- Service accounts often have uber permissions
- Tend to be fire and forget efforts
- Passwords tend not to be changed

| select name | export-csv C:\Data\ServiceAccounts.csv -NoTypeInformation
Service Account Activity

Get-aduser -Filter * -searchbase “OU=service, DC=you” | export-csv C:\Data\serviceaccounts.csv -NoTypeInformation

Solution

- ID how and when and where each account is used. If possible, block everywhere else

- Purchase a tool to automatically change password everywhere on a regular basis
Service Account Activity

Get-aduser -Filter * -searchbase "OU=service, DC=you" | export-csv C:\Data\serviceaccounts.csv -NoTypeInformation

Level of Effort

Major

- Who owns each account?
- Tools are expensive and hard to install and configure
- Lots of tweaking to get monitoring with high fidelity
Track a specific windows patch

wmic /failfast:on /node:@machines.txt qfe where hotfixid="KB958644" list full
Track a specific windows patch

wmic /failfast:on /node:@machines.txt qfe
where hotfixid="KB958644" list full

Why is this important?

Spot check vulnerability of enterprise to current attacks
Track a specific windows patch

wmic /failfast:on /node:@machines.txt qfe where hotfixid="KB958644" list full

Solution
Ideal for tracking specific patch effort
Track a specific windows patch

wmic /failfast:on /node:@machines.txt qfe where hotfixid="KB958644" list full

Level of Effort

Medium
Conclusion and Way Forward

Don't settle for simply ingesting logs, use AD, Powershell, WMI, etc to enhance and contextualize what you see and monitor.

Continue to determine how else information obtainable with WMI and AD can be compared to account and network activity.
Slides and Scripts will be posted at

www.shadowtrackers.net/presentations.html

Questions/Suggests/Comments

shadowtrackersnet@gmail.com

@shad0wtrackers
Working script

from Admin prompt!!! If using task scheduler, run with admin privs

powershell "Get-ADUser -Filter {Enabled -eq $false} | select samAccountName | export-csv C:\tools\InactiveAccounts.csv -NoTypeInformation"

c:\tools\pscp.exe -pw Path1Finder! c:\tools\InactiveAccounts.csv splunk@10.0.2.25:/opt/splunk/etc/apps/search/lookups/InactiveAccounts.csv

Splunk search: sourcetype="wineventlog:security" user!="*$ user!="ANONYMOUS LOGON" user!="LOCAL SERVICE" user!="NETWORK SERVICE" user!=SYSTEM [inputlookup InactiveAccounts.csv | fields samAccountName | rename samAccountName as user] | stats count by user
powershell "search-adaccount -passwordneverexpires | select samaccountname | export-csv C:\tools\passneverexpire.csv -NoTypeInformation"

c:\tools\pscp.exe -pw Path1Finder! c:\tools\passneverexpire.csv splunk@10.0.2.25:/opt/splunk/etc/apps/search/lookups/passneverexpire.csv

powershell "get-aduser -f { -not (lastlogontimestamp -like "*\") -and (enabled -eq $true)} | select samaccountname | export-csv C:\tools\neverloggedin.csv -NoTypeInformation"

c:\tools\pscp.exe -pw Path1Finder! c:\tools\neverloggedin.csv splunk@10.0.2.25:/opt/splunk/etc/apps/search/lookups/neverloggedin.csv

|inputlookup neverloggedin.csv | eval createdate = strftime(created,"%m/%d/%Y %I:%M:%S %p") | eval older = now() - 604800 | where older > createdate | table created samaccountname
Batch file getlocal.bat

sourcetype="wineventlog:security"
EventCode=4624 OR
EventCode=4625 OR
EventCode=4634 OR
EventCode=4768  [ | inputlookup
localaccounts.csv | fields Name |
rename Name as user ] | table _time
EventCode user host
Batch file:
@echo off
setlocal enabledelayedexpansion
(For /F "delims=" %%A in ("wmic /failfast:on /node:@machines.txt OS Get LastBootUpTime,CSName /format:csv |findstr "[0-9N]"") do ( set "line=%%A"
echo !line:~0,-1!
)) > lastboot.csv

c:\tools\pscp.exe -pw Path1Finder! c:\tools\lastboot.csv
splunk@10.0.2.25:/opt/splunk/etc/apps/search/lookups/lastboot.csv

|inputlookup lastboot.csv | eval boottime = strptime(LastBootUpTime,"%Y%m%d%H%M%S.%6Q") | eval older = now() - 604800 | where older < boottime | convert timeformat="%m/%d/%Y %H:%M:%S" ctime(boottime) | table CSName boottime
|inputlookup lastboot.csv | eval date=strptime(LastBootUpTime,"%Y%m%d%H%M%S") | convert timeformat="%m %d %Y %H:%M:%S" ctime(date) | table CSName date
Use a script to go through all machines on domain. Look for users logged on for more than five days.

Run `loggedon-users.py | inputlookup loggedonusers.csv | search "Days Logged On" > 5`
Scanning accounts
Start checkservices-test.ps1 on Choctaw
Let run.
Search in splunk "srv-jonesc"
host!="choctaw" | dedup host | table host with window of 1 minute
Add choctaw to machines.txt
Wait for new server to show up
Check in accounts
On pc1, 2 and 3, start checkin.ps1
Splunk search host=Apache "srv-blancm" | dedup host | table host with window of 1 minute