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
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
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
```

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

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 a 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.
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
Service Account Activity

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
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
Service Account Activity

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

| select name | 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

| select name | 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.
Last Reboot Time

\texttt{wmic /node:\COMPUTER OS Get LastBootUpTime}

* multiple ways to skin this cat.

Also can monitor for event ID 6005
Last Reboot Time

wmic /node:COMPUTER OS Get LastBootUpTime

* multiple ways to skin this cat.

Why is this important?

• Help verify patches have been applied and/or configurations updated

• Also, catch unexpected reboots
wmic /node:\COMPUTER OS Get LastBootUpTime
* multiple ways to skin this cat.


Also can monitor for event ID 6005

Solution

Use SIEM to create a report and track history.
wmic /node:COMPUTER OS Get LastBootUpTime

* multiple ways to skin this cat.


Also can monitor for event ID 6005

Level of Effort

Minimum – Medium
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
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 periodically (see tool suggestions from II.
- 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
File/Folder monitoring

Example to monitor the finance department fileserver:

```
get-aduser -filter { "all finance users" } | select samAccountName | export-csv C:\Data4Splunk\finance.csv -NoTypeInformation
```

Add any users not in this OU that have legit access to this fileserver. Monitor for anyone outside this group who tries to access the fileserver. Do the same with HR, R&D and any other departments as required.
File/Folder monitoring

Example to monitor the finance department fileserver:

```
get-aduser -filter { "all finance users" } | select samAccountName | export-csv C:\Data4Splunk\finance.csv -NoTypeInformation
```

Why is this important?

- Assists with Insider Threat.
- If an adversary compromises an account, odd behavior is more easily detected.
Example to monitor the finance department fileserver:

```
get-aduser -filter { “all finance users” } | select samAccountName | export-csv C:\Data4Splunk\finance.csv
```

**Solution**

- DLP
- Enable windows file/folder auditing
- Network segmentation
File/Folder monitoring

Example to monitor the finance department fileserver:

`get-aduser -filter { “all finance users” } | select samAccountName | export-csv C:\Data4Splunk\finance.csv -NoTypeInformation`
Users who always logged on

Psloggedon -I \\COMPUTER

Then find all users who have been logged on for more 5 days
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 \COMPUTER

Then find all users who have been logged

Solution

Forced reboots minimum once per week
Users who always logged on

Psloggedon -l \COMPUTER

Then find all users who have been logged on for more 5 days

Level of Effort
Minimum
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
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 create local accounts.
- List what local accounts are on each machine and that list should be short.
- Investigate new accounts.
- Investigate and enabled disabled account.
- 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
Track a specific Windows patch

wmic qfe where hotfixid="KB958644" list full
Track a specific windows patch

wmic qfe where hotfixid="KB958644" list full

Why is this important?
Spot check vulnerability of enterprise to current attacks
Track a specific windows patch

wmic qfe where hotfixid="KB958644" list full

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

wmic 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