MachineAccountQuota Transitive Quota: 110 Accounts and Beyond

Background

If you aren't familiar with MachineAccountQuota (MAQ), I recommend skimming my previous blog post on the subject.

TLDR

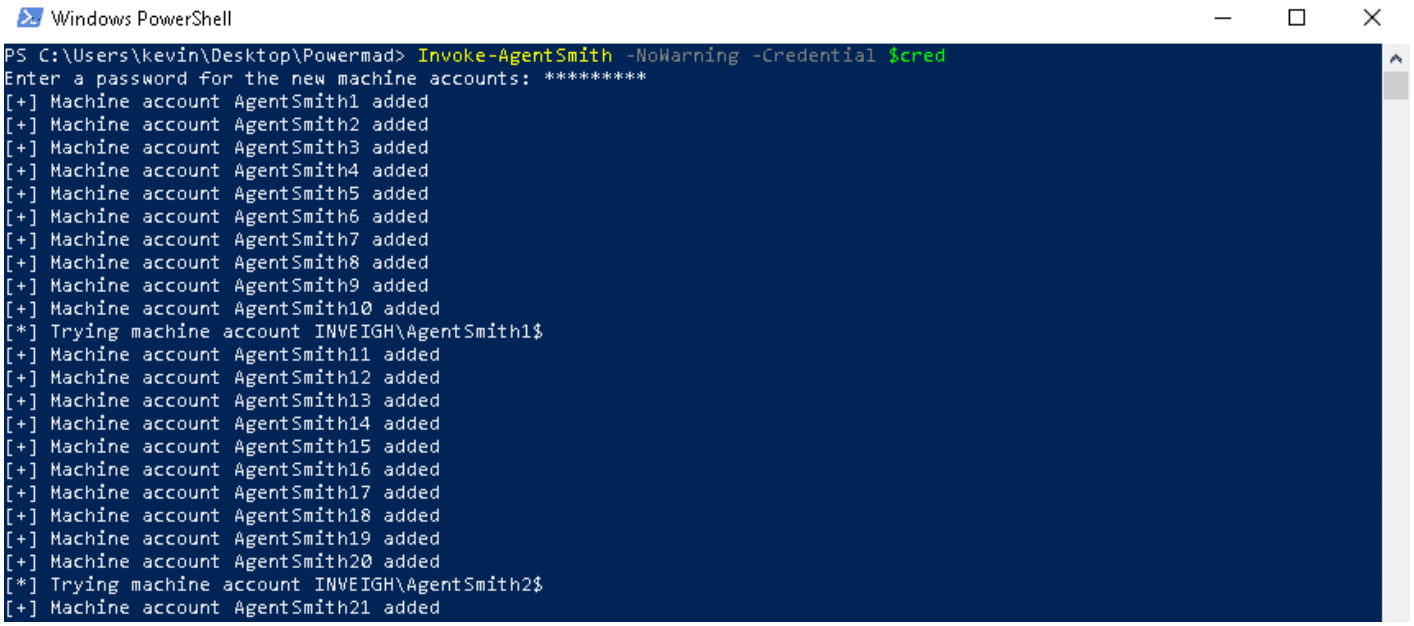
Active Directory (AD) tracks transitive accounts created through MAQ to limit the number of accounts that can be added from a single unprivileged source account. AD calculates the maximum using a formula of Q * (Q + 1), where Q is the current MAQ setting. The default MAQ setting of 10 results in a limit of 110 permitted transitive accounts. However, the transitive quota can often be exceeded by large amounts.

The Slightly Longer Version

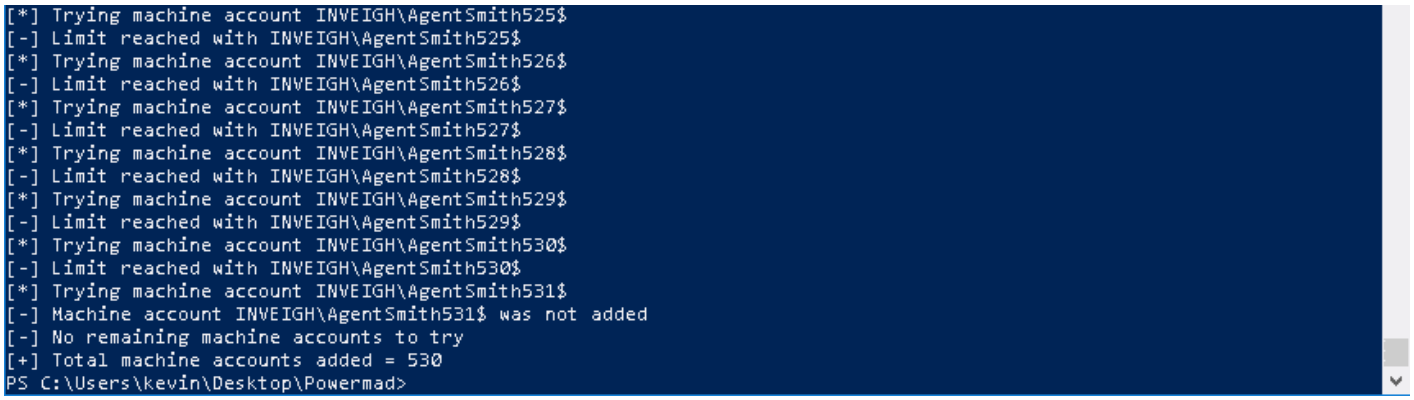
Early on when I started playing around with MAQ, I tested creating accounts recursively. Using just the New-MachineAccount function from Powermad, I went through the process of adding machine accounts and then using the created accounts to add more accounts. Since AD leverages the ms-DS-CreatorSID attribute to calculate the current MAQ count for an account, I was curious to see how AD would handle throwing multiple SIDs into the mix through recursive account creation. I found that AD did indeed track transitive accounts for MAQ and did not permit an unprivileged user to basically add an unlimited number of accounts.

Later, while I was putting together my previous MAQ blog post, I decided to revisit transitive account creation. This time, my manual efforts resulted in a total of 20 accounts created from a single unprivileged account.

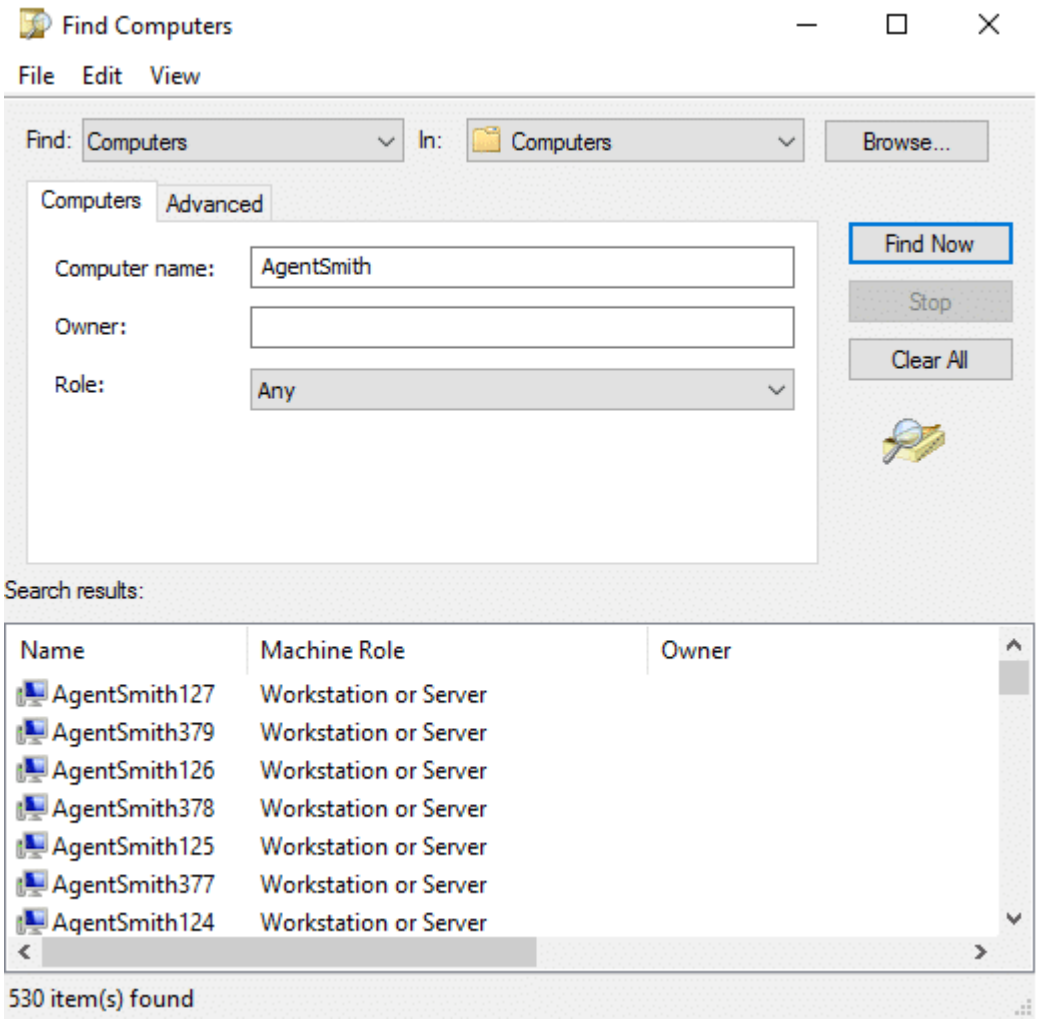
Next, I threw together a PowerShell function to automate the process and more easily test the full pool of created machine accounts. I quickly found myself adding way more than 20 accounts by creating the first 10 machine accounts and then cycling through each machine account while adding 10 machine accounts from each one.



I ran the function repeatedly and found that the most common result was 110 accounts created. However, the function often randomly exceeded 110 by large amounts.



To be sure of the results, I verified that the accounts were actually added to AD.



The results appear to be random when exceeding the transitive quota. As the function rotates through the created accounts, it will often go from success, to failing, and then back to successfully adding again.

```
[+] Machine account AgentSmith228 added
[+] Machine account AgentSmith229 added
[+] Machine account AgentSmith230 added
[*] Trying machine account INVEIGH\AgentSmith51$
[-] Limit reached with INVEIGH\AgentSmith51$
[*] Trying machine account INVEIGH\AgentSmith52$
[-] Limit reached with INVEIGH\AgentSmith52$
[*] Trying machine account INVEIGH\AgentSmith53$
[-] Limit reached with INVEIGH\AgentSmith53$
[*] Trying machine account INVEIGH\AgentSmith54$
[-] Limit reached with INVEIGH\AgentSmith54$
[*] Trying machine account INVEIGH\AgentSmith55$
[-] Limit reached with INVEIGH\AgentSmith55$
[*] Trying machine account INVEIGH\AgentSmith56$
[-] Limit reached with INVEIGH\AgentSmith56$
[*] Trying machine account INVEIGH\AgentSmith57$
[-] Limit reached with INVEIGH\AgentSmith57$
[*] Trying machine account INVEIGH\AgentSmith58$
[-] Limit reached with INVEIGH\AgentSmith58$
[*] Trying machine account INVEIGH\AgentSmith59$
[-] Limit reached with INVEIGH\AgentSmith59$
[*] Trying machine account INVEIGH\AgentSmith60$
[-] Limit reached with INVEIGH\AgentSmith60$
[*] Trying machine account INVEIGH\AgentSmith61$
[+] Machine account AgentSmith231 added
[+] Machine account AgentSmith232 added
[+] Machine account AgentSmith233 added
```

Note, the function achieved the same results on domains made up of both single and multiple domain controller configurations.

Microsoft's Response

I sent my PowerShell function and notes over to MSRC. They informed me of the transitive quota with the formula of Q * (Q + 1). Therefore, the 110 default maximum is by design.

Microsoft recently stated that exceeding the transitive quota may be a bug. However, it will not be addressed at this time.

Usages?

From a standard testing perspective, I'm not sure this one has much practical value. It might be fun to bring out in offense versus defense type competitions.

Invoke-AgentSmith

I've added the Invoke-AgentSmith function shown above to Powermad in case anyone wants to play around with the technique in a test lab.

Special thanks to Karl Fosaaen for the Agent Smith photoshop.

Note: Researchers have recently dubbed some Android malware as Agent Smith. I've had this stuff sitting around while the case was still open with MSRC. I've elected to not go through the effort of changing the Agent Smith references here to something else.