GitHub - shellster/DCSYNCMonitor: Monitors for DCSYNC and DCSHADOW attacks and create custom Windows Events for these events.

By shellster

Archived: 2026-04-05 21:42:15 UTC

Description

This tool is an application/service that can be deployed on Domain controllers to alert on Domain Controller Syncronization attempts. When an attempt is detected, the tool will write an event to the Windows Event Log. These events can be correlated in a SIEM. In addition, this tool can take a list of valid DC IP's and, in this configuration, only alert when a DC SYNC attempt comes from a non-DC ip. This tool is meant to provide Blue Teams with a way to combat DC SYNC and DC SHADOW attacks without commercial tools like Microsoft ATA or fancy IDS/IPS.

Brief Youtube Video: https://www.youtube.com/watch?v=oLND9QZfaJc

Installation Instructions

To install this tool, you can use either the pre-built binaries or build the tool yourself. Link for prebuilt binaries is here:

32bit Service:
https://github.com/shellster/DCSYNCMonitor/raw/master/Release/DCSYNCMONITORSERVICE.exe

64bit Service:
https://github.com/shellster/DCSYNCMonitor/raw/master/x64/Release/DCSYNCMONITORSERVICE.exe

You will need either Winpcap or Npcap installed on your domain controller. Winpcap should work, but is not recommended as the packet capture methods are not as efficient or thorough as NPcap. This tool has only been briefly tested with Winpcap.

To install Npcap, downloaded the installer it from here: https://nmap.org/npcap/

You should make sure that the following options are checked:

- Automatically start the Npcap driver at boot time
- Restrict Npcap driver's access to Administrators only

After installing, you will need to reboot the domain controller.

Npcap does not install the supporting library DLLs into the System's DLL search path, so you will need to perform the following tasks after installing:

```
 copy "%WINDIR%\System32\Npcap\*.dll" "%WINDIR%\System32\"

#If Applicable (32bit Service on 64bit System):
copy "%WINDIR%\SYSWOW64\Npcap\*.dll" "%WINDIR%\SYSWOW64\"
```

note: If the previous step is not completed, you will recieve errors about a missing wpcap.dll or Packet.dll when attempting to run the tool.

Now copy the DCSYNCMONITOR.EXE from this project into an appropriate location. We recommend %WINDIR%\SYSTEM32 for either 32bit systems or 64bit systems with a 64bit service, or %WINDIR%\SYSWOW64 if you are using the 32bit service on a 64bit system.

The tool can now be run. However, you can either run it one of two ways:

Without a configuration file

In this mode, the tool will write a DCSYNCALERT Warning event to the Windows Application Event Log everytime a new IP (not seen in the previous five minutes) attempts to perform a DC SYNC against the domain controller. This will include legitimate syncronization activities between domain controllers.

With a configuration file

A configuration file called, "dc_ip_list.conf" can be placed in the same directory as the tool. If this file exists, it should contain one IPv4 (or long form IPv6) address per line. The tool will ingest this list on start-up. In this mode, no events will be written for DC Sync attempts from matching IP addresses. However, if a DC Sync attempt occurs from any other IP address, a DCSYNCALERT Error event will be written to the Windows Application Event Log.

note Changes to the dc_ip_list.conf file will **not** take affect until the service is stopped and restarted.

The usual way to use this tool is to install it as a service. Once the tool is placed in the correct folder, this can easily be accomplished by running:

```
DCMONITORSERVICE.exe -install
```

Once you have installed the service, you will need to start it manually from the Services.msc menu or by using appropriate net or sc commands. It will auto-start on future reboots.

Should you need to uninstall the service, run the following command:

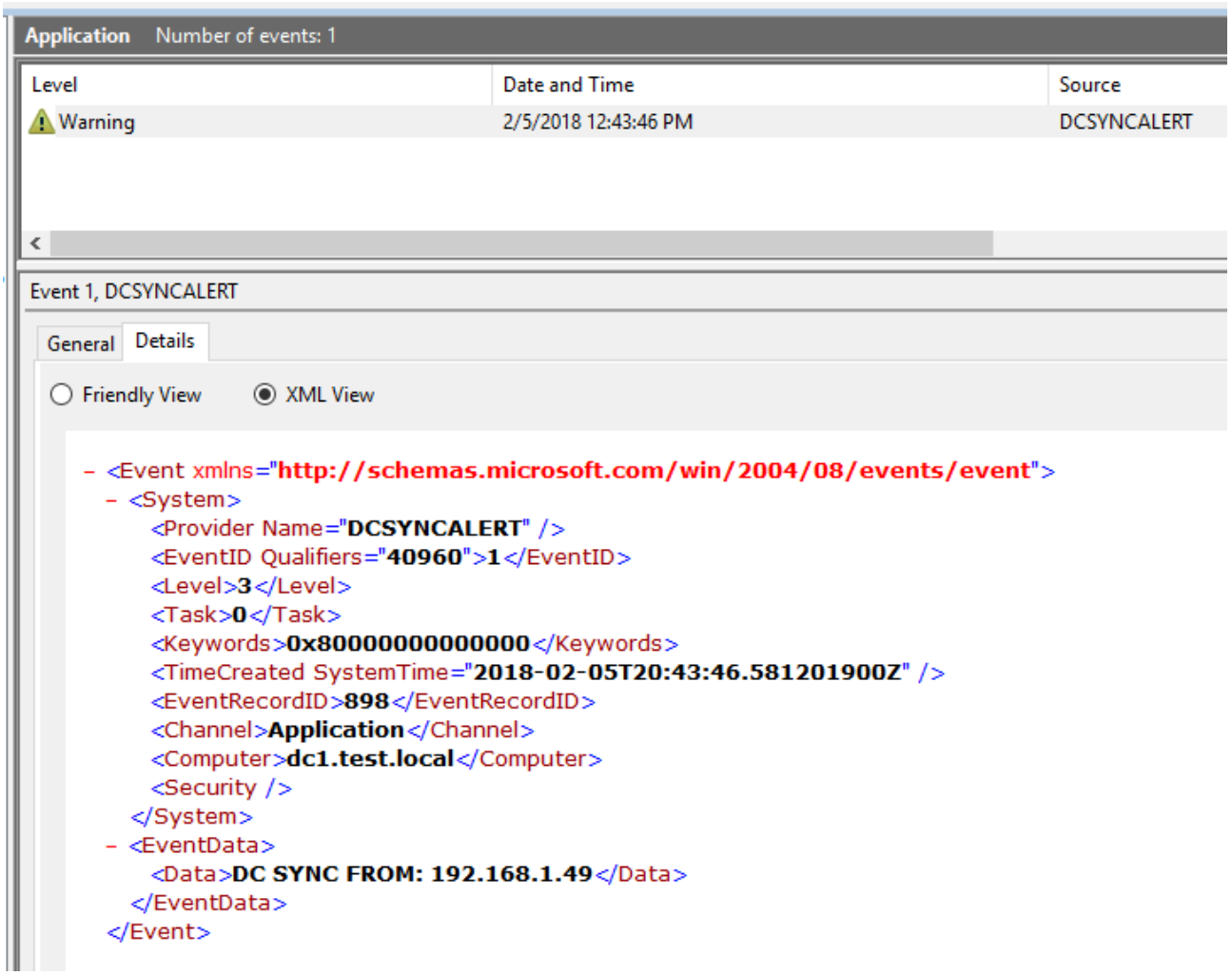
```
DCMONITORSERVICE.exe -remove
```

Finally, to run the tool in stand-alone mode, without installing a service (especially useful for debugging):

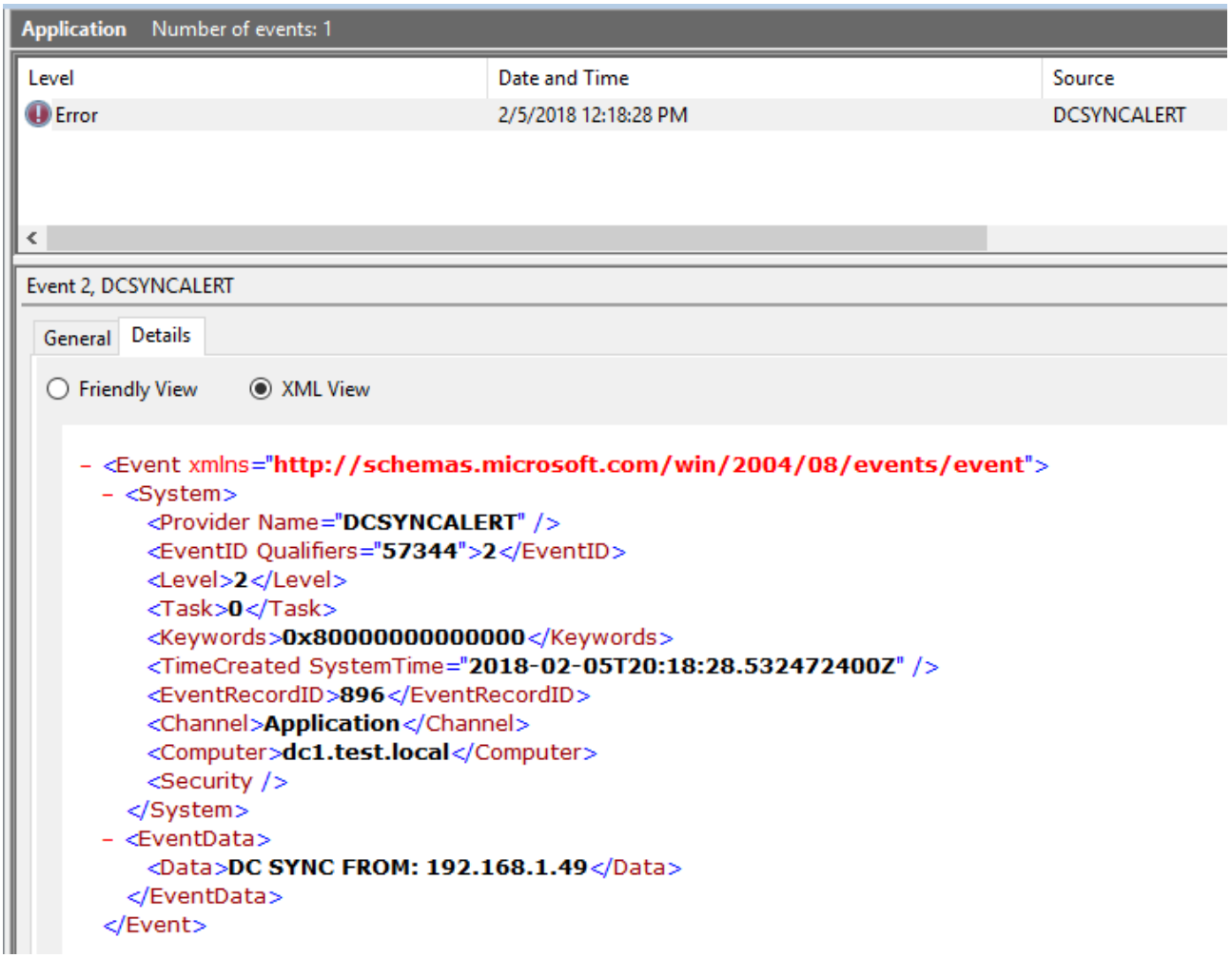
```
DCMONITORSERVICE.exe -standalone
```

Screenshots

DC SYNC Warning events occur when there is no list of valid DC IPs provided, or when a DC SYNC occurs from a valid DC IP:



DC SYNC Error events occur when a list of valid DC IPs are provided and a DC SYNC occurs from any other IP address:



Compilation Instructions

You will need Visual Studio 2015 or later. The Community (free) edition is perfectly acceptable. Once you open the project, you should be able to immediately build Dev and Release versions in both 32bit and 64bit varieties. The Debug editions should **not** be deployed in a production environment. They spit extensive error and debugging information, including tcp packet dumps (if you uncomment the following) line in the monitor.cpp file:

```
 #ifdef _DEBUG
                 //debug_print("TCP SRC IP: %s\nData:\n", tcppacket.source_ip.address.c_str());
                 //print_payload((const u_char *)tcppacket.data, tcppacket.data_length);
#endif
```

Release versions are much smaller and automatically strip debug statements.

Important Limitations

This tool has the following known limitations:

- The tool does a byte comparision for the DSNcChange Packet. This pattern should be fairly robust, but can likely be defeatable by an advanced attacker.

- The tool does not handle IPv4 fragmentation. An attacker could concievably specially craft a DC SYNC request with IPv4 fragmentation to bypass the packet sniffing.
- The tool does not handle IPv6 packet extensions. An attacker, on an IPv6 network could conceivably craft a DC SYNC request that contains extra header extensions or use a Jumbogram to bypass the signatures.
- The tool does not handle malformed packets which may or may not be correctly dropped by the kernel.
- It is highly unlikely, but a false postive could occur if a random tcp packet manages to match the 11 byte signature this tool checks for.
- This tool will only work on Server 2008 or later.

License

References

- Significant packet parsing insight and coding help was gathered from: https://www.tcpdump.org/sniffex.c
- The C++ Windows Service boilerplate was taken from here: https://code.msdn.microsoft.com/windowsapps/CppWindowsService-cacf4948
- The following page provided significant guidance on how to write to event logs: https://stackoverflow.com/questions/8559222/write-an-event-to-the-event-viewer

Credit

Tool was written by Shelby Spencer:

- Twitter: shellsterdude
- Keybase.io: shellster
- Github: shellster

FusionX generously provided me with time to update and refine this tool as well as a platform to announce it. However, this project is solely owned and developed by me.

Feedback

Suggestions, feedback, and PR's are all welcome and encouraged.
