Driver-Based Attacks: Past and Present

Dec 13, 2021 | 7 min read | Jake Baines

Last updated at Fri, 01 Dec 2023 19:19:33 GMT

"People that write Ring 0 code and write it badly are a danger to society." - Mickey Shkatov

There is no security boundary between an administrator and the Windows kernel, according to the Microsoft Security Servicing Criteria for Windows. In our analysis of CVE-2021-21551, a write-what-where vulnerability (see CWE-123) in a Dell driver, we found that Dell's update didn't fix the write-what-where condition but only limited access to administrative users. According to Microsoft's definition of security boundaries, Dell's fix removed the security issue. However, the partially fixed driver can still help attackers.

There's an attack technique called Bring Your Own Vulnerable Driver (BYOVD). In this attack, an adversary with administrative privileges installs a legitimately signed driver on the victim system. The legitimate driver has a vulnerability that the attacker exploits to gain ring 0 access. Access to ring 0 allows the attacker to subvert or disable security mechanisms and allows them to hide deeper in the system.

Known usage in the wild

BYOVD is a common technique used by advanced adversaries and opportunistic attackers alike. To illustrate this, the following table is a non-exhaustive list

Topics

Metasploit (654)
Vulnerability Management (359)
Research (236)
Detection and Response (205)
Vulnerability Disclosure (148)
Emergent Threat Response (141)
Cloud Security (136)
Security Operations (20)

Popular Tags

Search Tags

Metasploit
Metasploit Weekly Wrapup
Vulnerability Management
Research
Logentries

2021	Candiru	physmem.sys	Hilscher	N/A
2021	Iron Tiger	procexp152.sys	Process Explorer	N/A
2021	Iron Tiger	cpuz141.sys	CPUID CPU-Z	CVE-2017-15303
2021	GhostEmperor	dbk64.sys	CheatEngine	N/A
2021	ZINC	viraglt64.sys	Vir.IT eXplorer	CVE-2017-16238
2021	Various Cryptominers using XMRig	winring00x64.sys	OpenLibSys	N/A
2021	TunnelSnake	vboxdrv.sys	VirtualBox	CVE-2008-3431
2020	RobbinHood	gdrv.sys	Gigabyte	CVE-2018-19320
2020	Trickbot	rwdrv.sys	RWEverything	N/A
2020	InvisiMole	speedfan.sys	Alfredo Milani Comparetti Speedfan	CVE-2007-5633
2020	ZeroCleare	vboxdrv.sys	VirtualBox	Unclear
2020	Winnti Group	vboxdrv.sys	VirtualBox	CVE-2008-3431
2020	AcidBox	vboxdrv.sys	VirtualBox	Unclear
2020	Dustman	vboxdrv.sys	VirtualBox	CVE-2008-3431
2019	Doppelpaymer	kprocesshacker.sys	Process Hacker	N/A
2018	LoJax	rwdrv.sys	RWEverything	N/A

2024-47575 Exploited in Zero-Day Attacks READ MORE
Multiple Vulnerabilities in Common Unix Printing System (CUPS) READ MORE
High-Risk Vulnerabilities in Common Enterprise Technologies READ MORE
CVE-2024-40766: Critical Improper Access Control Vulnerability Affecting SonicWall Devices READ MORE

2018	Slingshot	speedfan.sys	Miami Comparetti Speedfan	2007-5633
2018	Slingshot	goad.sys	??	Unclear
2017	The Lamberts	sandra.sys	SiSoftware Sandra	CVE-2010-1592
2016	Remsec	aswsnx.sys	Avast!	Unclear
2016	Remsec	sandbox.sys	Agnitum Output	Unclear
2015	Equation Group	elbycdio.sys	CloneCD	CVE-2009-0824
2015	Derusbi	nicm.sys, nscm.sys, ncpl.sys	Novell	CVE-2013-3956
2014	Turla	vboxdrv.sys	VirtualBox	CVE-2008-3431
2012	Shamoon	elrawdsk.sys	Eldos Rawdisk	N/A

We believe that attacks or exploits that are *actually* used in the wild are, practically by definition, worthwhile for attackers. The table above illustrates that BYOVD **is** a valuable technique. Given these bad drivers' wide use in the wild, it would be beneficial for the security community to identify exploitable drivers and minimize or block their use.

Use cases

Those unfamiliar with BYOVD are probably wondering *why* these attackers are doing this. By far, the number one reason adversaries are using BYOVD is to bypass Windows Driver Signature Enforcement (DSE). DSE ensures that only signed kernel drivers can be loaded. By installing and exploiting a vulnerable driver, attackers can

- DSEFix (using CVE-2008-3841)
- TDL (using CVE-2008-3841)
- KDU (using multiple vulnerabilities including CVE-2015-2291, CVE-2018-19320, CVE-2019-18845, CVE-2019-16098, and CVE-2019-8372)

Each of these tools is authored by the same individual, hfiref0x. Stryker, DSEFix, and TDL are all deprecated or in read-only mode. Notably Stryker and DSEFix run afoul of PatchGuard and are no longer suitable for most situations. KDU, a tool that supports more than 14 different vulnerable drivers as the "provider," is the unsigned driver loader of choice.

Once the attacker has loaded their unsigned driver into the kernel, they can accomplish a wide variety of tasks they wouldn't be able to otherwise. Some obvious examples include unhooking EDR callbacks or hiding exploitation/rootkit artifacts. The attacker can write themselves a UEFI rootkit. Or just overwrite all data (resulting in BSoD). Or inject code into other processes.

The Dell drivers discussed below should be able to facilitate these types of attacks. Connor McGarr demonstrated Dell's dbutil_2_3.sys (which is vulnerable to CVE-2021-21551) can be used to execute attacker code in kernel mode. Because the write-what-where condition persists in the follow-on drivers, dbutildrv2.sys 2.5 and 2.7, Dell has delivered three unique signed drivers that can execute attacker code in kernel mode.

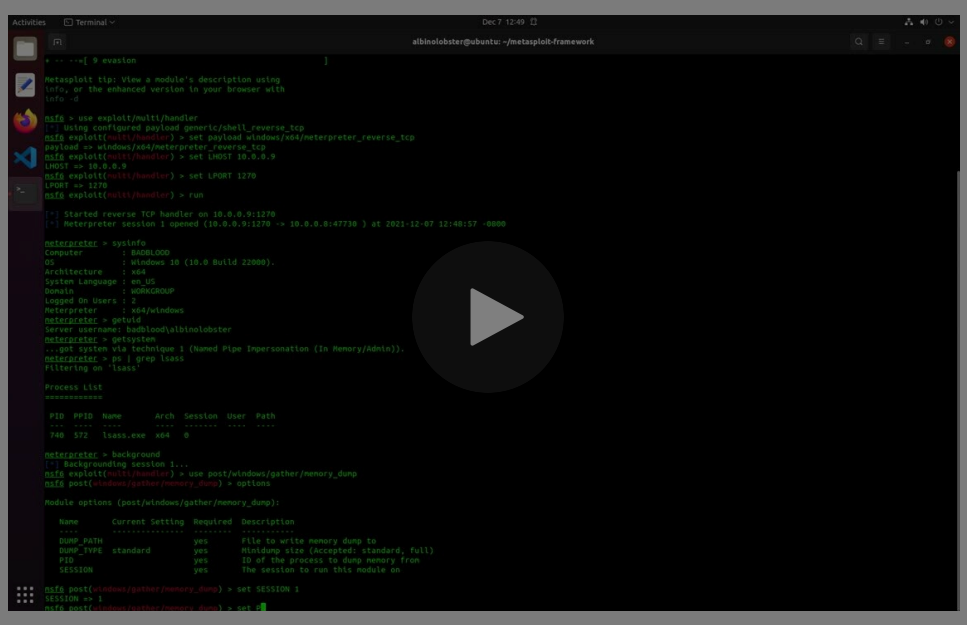
The previously mentioned attacks largely focused on executing code in kernel mode. However, BYOVD also enables a simpler data-oriented attack that allows the attacker to subvert LSA protection.

LSA protection prevents non-protected processes from

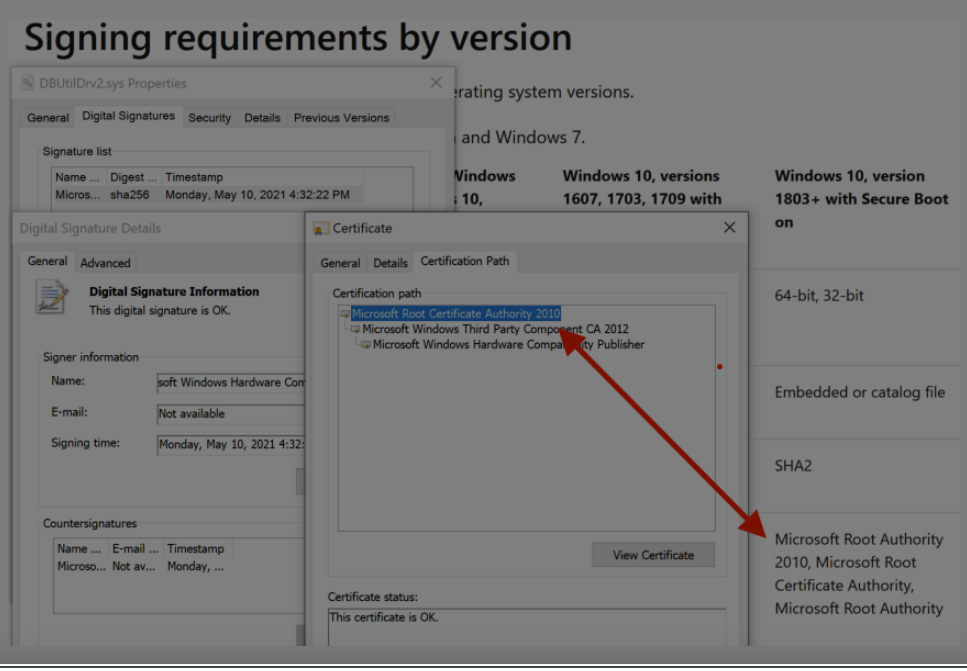
simply mask out the LSA protection. Once masked out, the attacker is free to dump lsass.exe's memory. There are a couple of good open-source implementations of this: mimidrv (a signed driver that is part of mimikatz) and PPLKiller (uses RTCore64.sys).

Exploitation using the Dell drivers

We've developed a Metasploit module that implements the LSA protection attack using the new Dell drivers (dbutildrv2.sys 2.5 and 2.7). An attacker with escalated privileges can use the module to enable or disable process protection on arbitrary PID. The following proof-of-concept video demonstrates unprotecting *lsass.exe* and dumping memory from metasploit.



The Dell drivers are especially valuable because they are compatible with the newest signing requirements issued by Microsoft.



Preventing users from updating their computers' firmware via driver blacklist is a non-starter.

While conducting this research, Rapid7 did reach out to Dell about this issue. They stated the following:

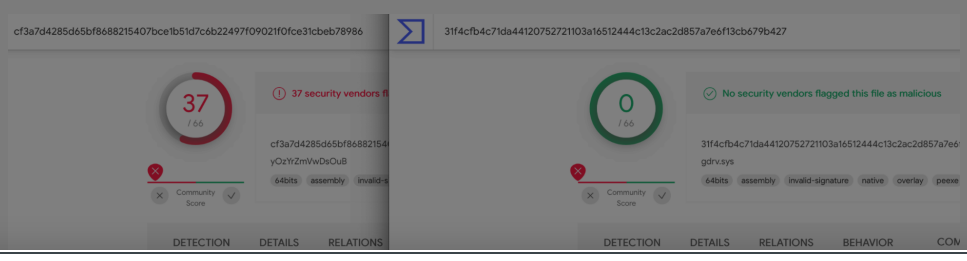
> After careful consideration with the product team, we have categorized this issue as a weakness and not a vulnerability due to the privilege level required to carry out an attack. This is in alignment with the guidance provided in the Windows Driver Model. We are not planning on releasing a security advisory or issuing a CVE on this.

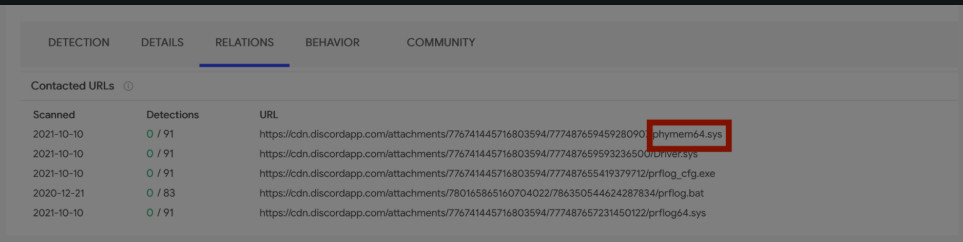
Other exploitation in the wild

Of course, we are not the first to use the Dell drivers in a malicious manner. As we noted in our AttackerKB analysis, dbutil_2_3.sys can be found associated with malware on VirusTotal. The newer versions of the driver, dbutildrv2.sys version 2.5 and 2.7, haven't appeared to be used maliciously yet. However, we do note a fair amount of other activity associated with BYOVD-related drivers that haven't yet been mentioned in this write up:

- asrdrv101.sys (CVE-2018-1071[0-2]?)
- asrdrv102.sys (CVE-2018-1071[0-2]?)
- ucorew64.sys
- piddrv64.sys
- atillk64.sys (CVE-2019-7246)

The point is that this is a fairly active and perhaps under-reported technique. It seems only the most well-known vulnerable drivers are flagged by AV. Even a well-known driver like the gdrv.sys isn't flagged.





Detection and mitigation guidance

Perhaps the best way to protect your systems is to utilize Microsoft's driver block rules. The list is full of known bad drivers and, if used correctly, will allow you to block the driver from being loaded. Of course, this only protects you from known vulnerable drivers that Microsoft adds to this list, but it's better than nothing. The Dell drivers are not currently in the list, but Dell has indicated they are working with Microsoft to add dbutil_2_3.sys. However, as discussed earlier, the newer versions are unlikely to ever get added. Detecting the Dell drivers through your preferred EDR solution might be an alternative solution. The SHA-1 hashes are:

dbutil_2_3.sys	c948ae14761095e4d76b55d9de86412258be7afd
dbutildrv2.sys (2.5)	90a76945fd2fa45fab2b7bcfdaf6563595f94891
dbutildrv2.sys (2.7)	b03b1996a40bfea72e4584b82f6b845c503a9748

If you are able to enable Hypervisor-Protected Code Integrity (HVCI) then you should absolutely do so. And, of course, you should have secure boot enabled at the very least.

We can all try to improve the Windows driver ecosystem by following Microsoft guidance on potentially dangerous drivers. Specifically, we can help by submitting drivers with vulnerabilities to the Microsoft Security Intelligence Driver Submission page for security

SUBSCRIBE

POST TAGS

Risk Management

Emergent Threat Response

AUTHOR

Jake Baines

VIEW JAKE'S POSTS

SHARING IS CARING

Related Posts

EMERGENT THREA...

Fortinet FortiManager CVE-2024-47575 Exploited in Zero-Day

READ FULL POST

EMERGENT THREA...

Multiple Vulnerabilities in Common Unix Printing System

READ FULL POST

EMERGENT THREA...

High-Risk Vulnerabilities in Common Enterprise

READ FULL POST

EMERGENT THREA...

CVE-2024-40766: Critical Improper Access Control

READ FULL POST

VIEW ALL POSTS