Linux Incident Response - disk copying

Taz Wake | Halkyn Consulting Ltd | Originally published 2023-11-30

Background and Overview

Disk imaging is a critical component in incident response and forensic analysis. Three tools are commonly used for this purpose: **dd**, **dcfldd**, and **dc3dd**. This article gives you a quick introduction to the tools and common arguments.

Originating from the Unix environment, dd is a versatile command-line utility for converting and copying files, frequently used to create raw disk images. One big advantage dd has is that it is almost certainly present in any Linux-like OS (YMMV with containers though).

The "newer" tools dcfldd and dc3dd, meanwhile, are enhanced versions of dd, specifically designed with forensic functionality in mind. They extend dd's capabilities, offering features tailored to the needs of digital forensic practitioners. They do, generally, need to be installed though.

dd: The Classic Tool

dd is a fundamental tool in Unix and Linux systems, renowned for its simplicity and effectiveness in creating a bit-by-bit copy of a data source. It operates with a straightforward syntax, where the primary arguments are if= for input file (source) and of= for output file (destination). The bs= (block size) argument is critical for determining the size of each read/write operation.

As an example:

```
```

creates an image of the sda drive saved ast /mnt/storage/disk_image.img. However, dd lacks advanced forensic features like hashing or progress indicators.

dcfldd: The Forensic Enhancement

dcfldd, developed by the Defence Computer Forensics Laboratory, enhances dd with features like hashing on-the-fly, status output, and verification. Its syntax is similar to dd, but with additional options.

For example

```
```

This command not only creates an image but also calculates the SHA-256 hash, writing it to hash.txt. This functionality is invaluable in ensuring the integrity of the disk image for forensic purposes.

```
root@asgard:~# dcfldd if=/dev/sdf of=/cases/disk_image.img bs=4M hash=sha256 hashlog=~/sdf_hashlog.txt
256 blocks (1024Mb) written.
256+0 records in
256+0 records out
root@asgard:~# cat ~/sdf_hashlog.txt

Total (sha256): fdb768fd8635e9c5461c4732980dc277d5ad61380a73173cb90ea219a76ec0b2
```

Example of dcfldd in use

dc3dd: A Further Step in Forensic Imaging

Finally dc3dd. This was a project under the Department of Defence Cyber Crime Center (DC3) looking to take dcfldd's capabilities further. It introduces more hashing algorithms, error logging, and data wiping. dc3dd also supports segmented output, which is useful for managing large disk images

As shown, this will be largely identical to the dcfldd output, but allows the use of SHA512 hashes. The output is also more verbose and the log= option gives the ability to record statistics and diagnostics.

```
root@asgard:~# dc3dd if=/dev/sdf of=/cases/disk_image.img hash=sha512 hlog=~/sdf_hash.txt log=~/sdf_log.txt

dc3dd 7.2.646 started at 2023-11-27 16:22:46 +0000
compiled options:
command line: dc3dd if=/dev/sdf of=/cases/disk_image.img hash=sha512 hlog=/root/sdf_hash.txt log=/root/sdf_log.txt
device size: 2097152 sectors (probed),    1,073,741,824 bytes
sector size: 512 bytes (probed)
  1073741824 bytes ( 1 G ) copied ( 100% ),    9 s, 110 M/s

input results for device `/dev/sdf':
   2097152 sectors in
   0 bad sectors replaced by zeros
   0d3a49df19da227efe5bae31ce9390693ac23677da76e14d526dbccad844148c6c0eb8413886bb3503775f03cef6a362173748bf6699492f41feb642ad65bd85 (sha512)

output results for file `/cases/disk_image.img':
   2097152 sectors out

dc3dd completed at 2023-11-27 16:22:55 +0000
```

Example of dc3dd in use

Note: The documentation says hlog and log should record different information, however in testing it looks like they often store the same output, which is a record of what dc3dd outputs to screen. The difference is that the log file also records error messages and any failed copies.

Comparing Block Sizes (bs=)

The block size is pivotal in these tools (although dc3dd does not support the bs= argument). It determines how much data is read and written in each operation. Larger block sizes can increase the speed of the imaging process but may use more system resources and might not be as effective in handling bad sectors. Conversely, a smaller block size may be slower but is often more reliable, particularly in cases of damaged or compromised drives. It is ultimately down to the incident responder to determine the best size for their particular use.

Comparisons

dd in Detail: Versatility and Limitations

dd's simplicity is both a strength and a weakness. Its versatility allows it to function in a wide range of scenarios, from disk imaging to data recovery. However, it lacks forensic features like hashing, which are crucial for ensuring the integrity and admissibility of digital evidence. The absence of a progress indicator can also be a drawback in incident response situations where time is critical.

dcfldd: Enhanced Functionality for Forensics

dcfldd addresses some of dd's limitations by introducing hashing and progress status. These features provide immediate feedback on the imaging process and ensure the integrity of the data being captured. This makes dcfldd particularly suitable for scenarios where verifying the authenticity of the disk image is as important as the imaging process itself.

dc3dd: Advanced Forensic Features

dc3dd extends these capabilities even further. Its segmented output is particularly useful for managing large datasets or storing images on media with size limitations. The error logging feature of dc3dd is essential in forensic investigations, as it provides a record of any issues encountered during the imaging process, which might be critical in understanding the state of the evidence.

Practical Examples and Usage Scenarios

Each tool has its specific scenarios where it excels. For instance, dd might be preferred in situations where a quick, straightforward disk image is needed without additional forensic features. dcfldd would be more suitable in environments where integrity checking is crucial, but the dataset is not overly large or complex. dc3dd would be the tool of choice in comprehensive forensic investigations, dealing with large or problematic datasets where detailed logging and advanced hashing are necessary.

Technical Considerations and Best Practices

When using these tools, it's important to consider the system's capabilities, the size and state of the data source, and the specific requirements of the incident response or forensic investigation. Practitioners should be aware of the implications of choosing different block sizes, the importance of integrity checking, and the need for detailed logging in certain scenarios.

Summary

dd, dcfldd, and dc3dd are powerful tools in the arsenal of any incident response or forensic professional. Understanding their differences, strengths, and limitations is key to

effectively using them in various scenarios. By selecting the appropriate tool and options, practitioners can ensure the efficient and reliable acquisition of digital evidence, crucial in the ever-evolving field of digital forensics.

Further learning

If you want to look into this further, and get practical experience in copying evidence during incident response scenarios, as well as investigate a realistic, nation-state intrusion into a Linux network then have a look at FOR577 Linux Incident Response from SANS Institute. You can find out more from https://sans.org/for577 and even book a seat today. Training is available live and in-person, live and online or fully on-demand.

About Halkyn Consulting

Halkyn Consulting Ltd is a UK-based security and risk management consultancy providing 24/7 incident response, digital forensics, threat hunting, virtual security leadership (vCISO) and security compliance support to organisations worldwide. We deliver clear, practical, vendor-independent advice based on real-world threats — no hype, no upsell.

More free security resources are available at www.halkynconsulting.co.uk/security-resources.

Web: www.halkynconsulting.co.uk | **Email:** info@halkynconsulting.co.uk | **Tel:** +44 1244 940858