

Blue Team Level 1 Certification
(Standard)

Introduction to BTL1

Welcome to Blue Team Level 1

4 Topics

Lab and Forum Access

SECURITY FUNDAMENTALS DOMAIN

Introduction to Security Fundamentals

1 Topic

Soft Skills

7 Topics

Security Controls

5 Topics 1 Quiz

Networking 101

6 Topics 1 Quiz

Management Principles

4 Topics 1 Quiz

PHISHING ANALYSIS DOMAIN

PA1) Introduction to Emails and Phishing

7 Topics 1 Quiz

PA2) Types of Phishing Emails

10 Topics 2 Quizzes

PA3) Tactics and Techniques Used

12 Topics 2 Quizzes

PA4) Investigating a Phishing Email

8 Topics 2 Quizzes

PA5) Analysing URLs, Attachments, and Artifacts

8 Topics 1 Quiz

PA6) Taking Defensive Actions

12 Topics 1 Quiz

PA7) Report Writing

7 Topics 1 Quiz

PA8) Phishing Response Challenge

3 Topics 1 Quiz

THREAT INTELLIGENCE DOMAIN

TI1) Introduction to Threat Intelligence

7 Topics

TI2) Threat Actors & APTs

6 Topics 2 Quizzes

TI3) Operational Threat Intelligence

7 Topics 1 Quiz

TI4) Tactical Threat Intelligence

7 Topics 1 Quiz

TI5) Strategic Threat Intelligence

5 Topics 1 Quiz

TI6) Malware and Global Campaigns

6 Topics 1 Quiz

DIGITAL FORENSICS DOMAIN

DF1) Introduction to Digital Forensics

5 Topics

DF2) Forensics Fundamentals

10 Topics 5 Quizzes

Hashing and Integrity

Blue Team Level 1 Certification (Standard) > DF2) Forensics Fundamentals > Hashing and Integr...

IN PROGRESS

Digital Forensics Domain HASHING AND INTEGRITY



Creating hashes is an important part of digital forensics, as it allows any tampering or modification of evidence to be immediately visible. This lesson will cover what hashes are, how they can be retrieved, how they're used to ensure integrity of digital evidence, how they can be cracked (even though they're a one-way function!), and finish up with some practical exercises in the next lesson.

WHAT ARE HASHES?

Hash values, which come in the form of text strings, are the unique fingerprint of a file or string. If I had a text file with the letter "ABC" in it, I could generate a hash value. Now if I went back into that file and added the letter "D" to it, and retrieved the file's hash value, it will be different than the initial one. We have modified the contents, so now the fingerprint is different.

The most common hash to work with is Message Digest 5, commonly referred to as MD5. Two other common hashes include SHA1, and SHA256. Due to collisions, an event where two different data values can have the same hash value, MD5 is no longer used as a secure standard, and SHA256 is taking over as the most common algorithm to use. We have already covered how to generate MD5, SHA1, and SHA256 hashes in the Phishing Analysis domain, but we'll provide a quick overview here.

Gathering Hashes in Windows

In the below screenshot we are using PowerShell on a Windows system to generate different file hashes for an executable file named "wallpaperHD.exe". By default, the command `get-filehash <file>` will generate a SHA256 hash. If we want to retrieve the MD5 or SHA1 values, we need to add the `-algorithm` flag to specify what hashes we want. Using `get-filehash -algorithm md5 <file>` we are able to retrieve the md5 hash, and the same method can be applied for SHA1.

```
PS C:\Users\JBeam\Desktop\Malware> get-filehash .\wallpaperHD.exe
Algorithm Hash Path
-----
SHA256 2403873290EE4F03F98A89A2FEFF9BF30DCBA61FCF614CDAC24129DA54442762 C:\Users\JBeam\Desktop\Malware...

PS C:\Users\JBeam\Desktop\Malware> get-filehash -algorithm md5 .\wallpaperHD.exe
Algorithm Hash Path
-----
MD5 0C4374D72E166F15ACDFE44E9398D026 C:\Users\JBeam\Desktop\Malware...

PS C:\Users\JBeam\Desktop\Malware> get-filehash -algorithm sha1 .\wallpaperHD.exe
Algorithm Hash Path
-----
SHA1 FBAC123E604137654759F2FBC4C5957D588103D1 C:\Users\JBeam\Desktop\Malware...
```

Gathering Hashes in Linux

On a Linux system generating hashes is a lot quicker. We can use the following three commands to generate SHA256, MD5, and SHA1 hashes respectively:

- `sha256sum <file>`
- `md5sum <file>`
- `sha1sum <file>`



<input type="radio"/> Section Introduction, Forensics Fundamentals
<input type="radio"/> Introduction to Data Representation
<input checked="" type="radio"/> Activity) Data Representation
<input type="radio"/> Hard Disk Drive Basics
<input type="radio"/> SSD Drive Basics
<input type="radio"/> File Systems
<input checked="" type="radio"/> Lab) File Systems
<input type="radio"/> Digital Evidence and Handling
<input type="radio"/> Order of Volatility
<input type="radio"/> Metadata and File Carving
<input checked="" type="radio"/> Lab) Metadata and File Carving
<input type="radio"/> Memory, Pagefile and Hibernation File
<input checked="" type="radio"/> Hashing and Integrity
<input checked="" type="radio"/> Lab) Hashing and Integrity
<input checked="" type="radio"/> Activity) End of Section Review, Forensics Fundamentals
<input type="radio"/> DF3) Digital Evidence Collection
<input checked="" type="radio"/> 8 Topics 1 Quiz
<input type="radio"/> DF4) Windows Investigations
<input checked="" type="radio"/> 3 Topics 3 Quizzes
<input type="radio"/> DF5) Linux Investigations
<input checked="" type="radio"/> 4 Topics 2 Quizzes
<input type="radio"/> DF6) Volatility
<input checked="" type="radio"/> 3 Topics 1 Quiz
<input type="radio"/> DF7) Autopsy
<input checked="" type="radio"/> 4 Topics 1 Quiz
SECURITY INFORMATION AND EVENT MANAGEMENT DOMAIN
<input type="radio"/> SI1) Introduction to SIEM
<input checked="" type="radio"/> 7 Topics 1 Quiz
<input type="radio"/> SI2) Logging
<input checked="" type="radio"/> 6 Topics 2 Quizzes
<input type="radio"/> SI3) Aggregation
<input checked="" type="radio"/> 2 Topics 1 Quiz
<input type="radio"/> SI4) Correlation
<input checked="" type="radio"/> 6 Topics 1 Quiz
<input type="radio"/> SI5) Using Splunk
<input checked="" type="radio"/> 5 Topics 2 Quizzes
INCIDENT RESPONSE DOMAIN
<input type="radio"/> IR1) Introduction to Incident Response
<input checked="" type="radio"/> 8 Topics 1 Quiz
<input type="radio"/> IR2) Preparation Phase
<input checked="" type="radio"/> 10 Topics 2 Quizzes
<input type="radio"/> IR3) Detection and Analysis Phase
<input checked="" type="radio"/> 7 Topics 4 Quizzes
<input type="radio"/> IR4) Containment, Eradication, and Recovery Phase
<input checked="" type="radio"/> 5 Topics 1 Quiz
<input type="radio"/> IR5) Lessons Learned and Reporting
<input checked="" type="radio"/> 7 Topics
<input type="radio"/> IR6) MITRE ATT&CK
<input checked="" type="radio"/> 13 Topics 2 Quizzes
BTL1 EXAM
<input type="radio"/> Exam Preparation

```
root@SBTLab2: ~/Desktop
root@SBTLab2:~/Desktop# sha256sum wallpaperHD.exe
240387329dee4f03f98a89a2feff9bf38dcbaf61cf614cdac24129da54442762 wallpaperHD.exe
root@SBTLab2:~/Desktop#
root@SBTLab2:~/Desktop#
root@SBTLab2:~/Desktop# md5sum wallpaperHD.exe
8c4374d72e166f15acdfe44e9398d026 wallpaperHD.exe
root@SBTLab2:~/Desktop#
root@SBTLab2:~/Desktop# sha1sum wallpaperHD.exe
f8ac123e04137654759f2fbc4c5957d5881d3d1 wallpaperHD.exe
root@SBTLab2:~/Desktop#
root@SBTLab2:~/Desktop#
```

We can also retrieve the hash values of text strings using the command `echo -n <text> | string`, as demonstrated below.

```
File Actions Edit View Help
root@SBTLab2: ~/Desktop
root@SBTLab2:~/Desktop#
```

EVIDENCE INTEGRITY

Now that you understand how hashes work, and how they're generated, you should be able to see how this applies to digital forensics, and ensuring the integrity of files or evidence. In most investigations involving a hard drive, a hash will be generated from the hard drive, and then a complete copy of the storage media will be taken at a bit-by-bit level, meaning that **everything** possible from the disk is copied to a fresh hard drive. This new hard drive then has its hash generated, to ensure that this is the exact same value as the original, proving that an exact copy was successfully generated. This allows forensic analysts or investigators to work on a copy of the evidence, instead of analyzing the actual disk which could result in loss of evidence if anything went wrong, or the court could argue that the evidence may have been tampered with, and is therefore not viable for use in court during legal proceedings.

REVERSING HASHES

Hashing functions are one-way, meaning that it is not possible to reverse the initial value directly. The only way is by employing brute force or dictionary attacks to crack the hash. You need to calculate the hash of every potential combination, or against a huge list of strings, and compare it to the hash you want to crack. You can also use free online services that have already calculated millions of hashes.

We're going to use a command-line tool called Hashcat to perform a dictionary attack against a target MD5 hash. In this attack, Hashcat will generate the hash value of the text strings in the word list, and see if the hash matches the one we're trying to crack. If they match, we've found the text string that has been hashed, and in a way, reversed the hashing process. This type of activity is typically done when penetration testers or red team members gain access to account credentials that are encrypted, and try to find the plain text versions of them.

So for this example, we have the target MD5 hash in a text file named "md5hash.txt" and our random wordlist "BTL1_Word_List.txt".

```
root@Desktop:~/BTL1_Word_List - Microsoft
File Edit Search View Document Help
Warning, you are using the root account, you may harm your system.
cd136adafe6fb774fa58b9e28f906

root@Desktop:~/BTL1_Word_List - Microsoft
File Edit Search View Document Help
Warning, you are using the root account, you may harm your system.
xanphobia
spanning
seimical
accured
Pefortify
parcidize
mable
caryl
provable
pasteurised
hugton
superfarm
unnsurely
egg
chlorobenzene
rainbow
prococession
downwing
consolidation
arte
neogothic
torribly
decrecendo
```

To set Hashcat to work, we need to open a terminal in the same location as our two files, in this case, the Desktop. We'll right-click the desktop and select "Open Command Prompt" and use the following command:

- ☐ Using RDP and SSH
- ☐ How to Start Your Exam

```
hashcat -m 0 md5hash.txt BTL1_Word_List.txt
```

- "hashcat" selects the tool we want to use
- "-m 0" selects the hash mode, 0 is md5
- "md5hash.txt" the file name containing the hashes to crack
- "BTL1_Word_List.txt" the file name of our wordlist

```
The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.
cd13b6a6af66fb774faa589a9d18f906:rainbow

Session.....: hashcat
Status.....: Cracked
Hash.Type.....: MD5
Hash.Target.....: cd13b6a6af66fb774faa589a9d18f906
Time.Started.....: Thu May 28 12:21:36 2020 (0 secs)
Time.Estimated...: Thu May 28 12:21:36 2020 (0 secs)
Guess.Base.....: File (BTL1_Word_List.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 3910 H/s (0.04ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.....: 203/203 (100.00%)
Rejected.....: 0/203 (0.00%)
Restore.Point...: 0/203 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1...: xenophobia -> belting

Started: Thu May 28 12:21:34 2020
Stopped: Thu May 28 12:21:37 2020
root@S8TLab2:~/Desktop#
```

Once Hashcat has identified a matching hash, it will present the plain text value of the hash we wanted to crack! It's very important to remember that a dictionary attack will only ever succeed if the right text string is in the word list. In this case, the answer was Rainbow - if this word wasn't in the list, Hashcat wouldn't have identified that the hash we're cracking had the same value, and we wouldn't have got a result.

Quizzes

☒ Lab) Hashing and Integrity

[< Previous Topic](#)

[Back to Lesson](#)

[Privacy & Cookies Policy](#)

