UNIVERSITY OF AMSTERDAM

Faculty of Physics, Mathematics and Informatics
Graduate School of Informatics
System and Network Engineering MSc

# Project Spartan Forensics

Cybercrime and Forensics

James Gratchoff
james.gratchoff@os3.nl

Guido Kroon
guido.kroon@os3.nl

May 31, 2015

**Abstract**

*Project Spartan is the codename of the new Microsoft Edge browser and successor to its previous, Internet Explorer. This research paper gives insight into the current artefacts that the current development versions of Project Spartan leaves behind on workstations. The authors analysed what these artefacts are, where they are located and how can they be gathered. This research led to the conclusion that Project Spartan's back end does not differ much from the latest Internet Explorer versions, as Project Spartan still uses similar ways to store data on the workstation it runs on. Furthermore, an open source tool has been developed to gather some of these artefacts in an automated way. The purpose of the tool is to gather the location of the artefacts not present in the database.*

# Contents

# Chapter 1

# Introduction

Web browsing activity is a major source of information in forensics investigation [11]. Much open-source and proprietary software already exists to perform forensic investigation on the most popular leading web browsers. These forensic tools depend on the architecture of the web browsers and thus need to adapt their code to new versions or new browsers.

Microsoft is moving away from their traditional web browser, called Internet Explorer (IE), and launching their new Edge web browser, formerly codenamed Project Spartan, which will be shipped by default on Windows 10. The web browser uses the new Edge engine, which is a fork from their former Trident engine that IE is based on.[13] However, as Edge is currently still in development as Project Spartan, this research will refer to it as Project Spartan, and not as Edge.

The purpose of this project is to gather information about new artefacts that Project Spartan leaves behind on workstations. If time permits, an open source tool for analysing these artefacts will be created as a proof of concept.

## 1.1 Scope, motivation and research question

As Edge is a newly developed browser, it is interesting to research the artefacts it leaves on workstations, especially if more and more people are to start using it when Windows 10 is released next summer. Therefore, this new information may be valuable to the digital forensics community and will soon be needed for investigations. This project will only target the browser artefacts. A quick investigation of the new Cortana features has been also carried out. Information that can be found on this project is related to Project Spartan and not to the Edge browser that has not yet been released. However the browser is supposedly just to be given another product name thus the artefacts found should be the same and located in the same directory structure with a difference in the path name. The path name that will be used in Edge is not known on the day of writing.

Overall discussion of the significance and motivation resulted in the following research question:

*What and where are the artefacts Project Spartan leaves behind on workstations, and how can these artefacts be gathered for further analysis to serve as forensic evidence?*

The above research question can be divided into the following research sub-questions:

1. *As the new Project Spartan engine is forked from its predecessor's Trident engine used with IE, how much does Project Spartan differ from its predecessor and to what extent can existing forensic toolkits for browsers still gather these artefacts in the same way they gather artefacts for Internet Explorer?*

2. *Can a tool be developed, based on the assembled results, in order to gather the artefacts of the Project Spartan web browser in an automated way?*

# Chapter 2

# Related work

Due to it being a recent product, no forensic research related to Windows 10 or Project Spartan/ Edge has been published at the time of writing. However much research has been done regarding web browser forensics. This project started by analysing the structure of Project Spartan and also how the latest version of Microsoft IE stored its information. Version 10 and later of IE will be referred to as IE v10+ in the rest of this report. Then the similar features of Microsoft Project Spartan were compared to IE v10+ in terms of artefacts location and databases. Furthermore the new features of Project Spartan were analysed and traced back to find where the artefacts location were stored on disk.

## 2.1 Browser forensics

Forensics tools that investigate browser activity, rely on the location of artefacts stored on disk. These locations are specific to each browser. Thus these tools need to adapt the locations and way of gathering information when a new browser is released. Forensics investigators need to gather detailed and trustworthy information about all the artefacts left on the disk by the browser. Moreover, any kind of information that a browser leaves behind can be valuable and of extreme importance in investigations. That is why it is important not to neglect any artefacts that could lead to a stronger proof of user activity.

Private browsing has also become popular as it is a way of increasing privacy while browsing. Using private browsing, the browser is not supposed to store any browsing activity during the session. Thus it is understandable that private browsing forensics has been a developing area of research. Said et al [12] researched Microsoft IE as well as Mozilla Firefox and Google Chrome regarding their privacy browsing features. They concluded that Google Chrome and Mozilla Firefox complete a better task in hiding their private browsing data, while Internet Explorer seems to leave evidence 'all over the hard drive'. Chivers [6] conducted another research project targeting the private browsing feature of IE 10, and was able to recover data from private browsing in a specific window of time. Indeed by carving log files he was able to identify some substantial records of private browsing that had taken place the last time the browser was opened. Due to the short life cycle of private browsing records in the database,

these records could not be found after opening the browser a second time. To carve the log files containing the previous records of the private browsing he developed a tool, ESECarve.

## 2.2 Structure of Internet explorer

A great deal of research has been done related to IE version 10 and later. And from our early investigation on the structure of Project Spartan we could say that it is extremely similar to IE v10+. Microsoft Project Spartan and IE v10+ rely on an Extensible Storage Engine (ESE) database, previously known as Joint Engine Technology (JET), to store their information. Metz [9], detailed in his research what the format of the database is and Chivers [6] describes how the ESE works. In IE 10, a single database named WebCacheV1.dat is dedicated to storing artefacts. This database is located at:

IE WebCache database location

```
%LocalAppData%\Microsoft\Windows\WebCache
```

Artefacts present in this database differ in their type (e.g. Cache, History, Cookies) and these types are divided into different containers tables ('Containers_XX'). These containers can be identified using another container table present in the same database, named 'Containers', that acts as an index table specifying which artefacts correspond to which containers. Each container shares the same fields that can be found in [9]. All these fields are valuable for forensic investigations. The functioning of the database follows the steps described in [6]. When a transaction is taking place the ESE first stores in memory the information regarding this transaction in a log cache, then it subsequently stores in memory the necessary database pages. As soon as the system is ready it writes to the log file (e.g. V01.log). After this, if possible, the database is updated with the new transaction and proceeds in a clean state, if not it will proceed in a dirty state. If the state of the database is dirty it will have to be recovered using the `.chk` files (that stores logged transactions from a known checkpoint) and the corresponding log files. The database can also be recovered to a clean state using the `esentutl` Windows tool.

Most of the artefacts are not only stored in the database but can also be found on the disk as files. For IE 10, these artefacts are located in the sub directories of:

IE directory location

```
%LocalAppData%\Microsoft\InternetExplorer\
```

The artefacts that can be found there are for example the cache files, the cookies, the favourites and what have you. Another location where IE stores information is in the registry key[3]. The information located there is obfuscated but can be read with IE PassView[4]. The information that can be found there is auto complete forms, auto complete password or typed URLs. The location of the registry key is:

IE registry key location

```
HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\
```

# Chapter 3

# Approach

The first part of the research was to understand the structure of the Project Spartan browser and understand what methods it was using to store information about an user. Secondly an investigation on how and where artefacts were found in most used web-browsers was carried out. This investigation was mainly focused on IE version 10 and later, due to the similarities found in the first step with the Project Spartan browser. Further to this investigation the authors compared the artefacts from IE and Project Spartan and documented what new artefacts could be found on Project Spartan. The next step was to discover where and how artefacts are stored on the disk. Thereafter tools used to investigate browser activity were tested on the new browser. The last step was to summarise what had been found using available tools and to create a tool that is able to find the new artefacts discovered on Project Spartan.

The following tools were used for this research:

- **ESEDatabaseView v1.30** [10] ESEDatabaseView is simple utility to browse through ESE structured database files, developed by Nir Sofer. We used it to browse through the ESE databases Project Spartan uses to store its data in, namely the `WebCacheV01.dat` and the `Spartan.edb` files.
- **ESECarve v1.20** ESECarve is a forensic tool written by Chivers that is used to inspect and and recover deleted data from ESE database files.
- **Notepad++ v6.7.8 (with the hex editor plugin v0.9.5)** [7] Notepad++ is an open source text editor for Windows operating systems. Together with the hex editor plugin we used this tool to open and read contents of many files.

# Chapter 4

# Artefacts Analysis

This section describes where Project Spartan store its artefacts on disk and detail whereas or not these artefacts could be found in the Extensible Storage Engine (ESE) database. A section also describes what features, that are likely to leave artefacts, are not implemented yet in Project Spartan.

## 4.1 Database

Microsoft Project Spartan uses the same database structure as the latest versions of Internet Explorer, namely the ESE database. The Internet Explorer 10 ESE database structure has been researched in-depth by [8].

The main WebCache database file is located in a dispersed fashion, which differs per user, hence the `%LocalAppData%` environment variable:

Spartan's WebCache database location

```
%LocalAppData%\Spartan\Database\WebCacheV01.dat
```

Numerous tools exist for reading these database files, which we will also list in the Tools section, but we mostly used `ESECarve`, `ESEDatabaseView` and `esentutl`. Within this database file, all sorts of information is stored, but not much actual content (some, but definitely not all). Rather, it is more like an index which keeps track of all the locations where the actual artefacts are stored.

When we try to open this file with a hex editor, we can see that this version still uses the same format that previous versions use. The hex dump of the database headers can be interpreted as follows [8] (see figure 4.1).

This can be verified when analysing the database with `esentutl`, which is installed by default on every Windows system. Note that the database is using little endian, so when compared with a hexadecimal dump, every byte range needs to be read in reverse order. While the database mostly stores Metadata as opposed to actual content, there are some interesting artefacts to be harvested from within this file, such as:

1. visited URLs (see section 4.5)
2. Cortana search queries (see section 4.8)
3. download history (see section 4.6).

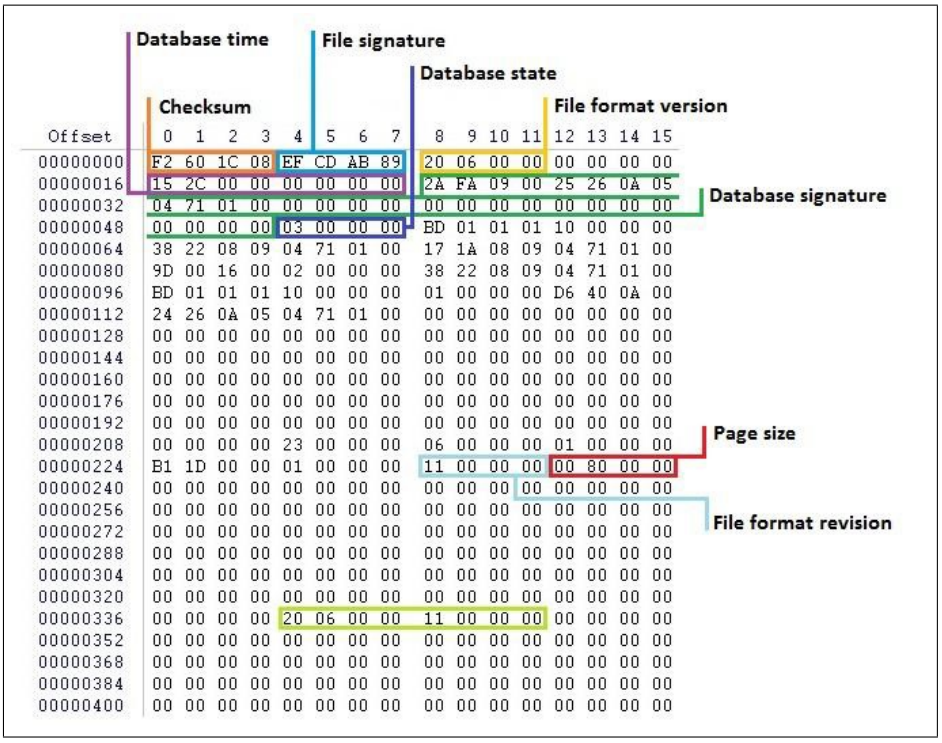


Figure 4.1: Hex dump with annotations.[8]

The ESE database also contains the location of every other artefacts that are stored locally on the system (see figure 4.2).

The container that is being viewed in figure 4.2 shows all the container IDs of the other containers that can be viewed. It shows what content is being stored in which container and where it can be found on the system (folder paths).

## 4.2 Cache

Project Spartan stores its caches in a dispersed fashion as well, which differs per user, hence the `%LocalAppData%` environment variable:

Project Spartan's cache location

```
%LocalAppData%\packages\microsoft.windows.spartan_{PackageID}\
AC\#!001\Spartan\Cache
```

Just like IE there are four cache folders in this directory, which each contain a portion of the cache. They contain all sorts of content which is saved locally when browsing with Project Spartan, like HTML pages, pictures and even downloads which are stored here temporarily before they are moved to the actual download folder. This is an example of such a cache folder:

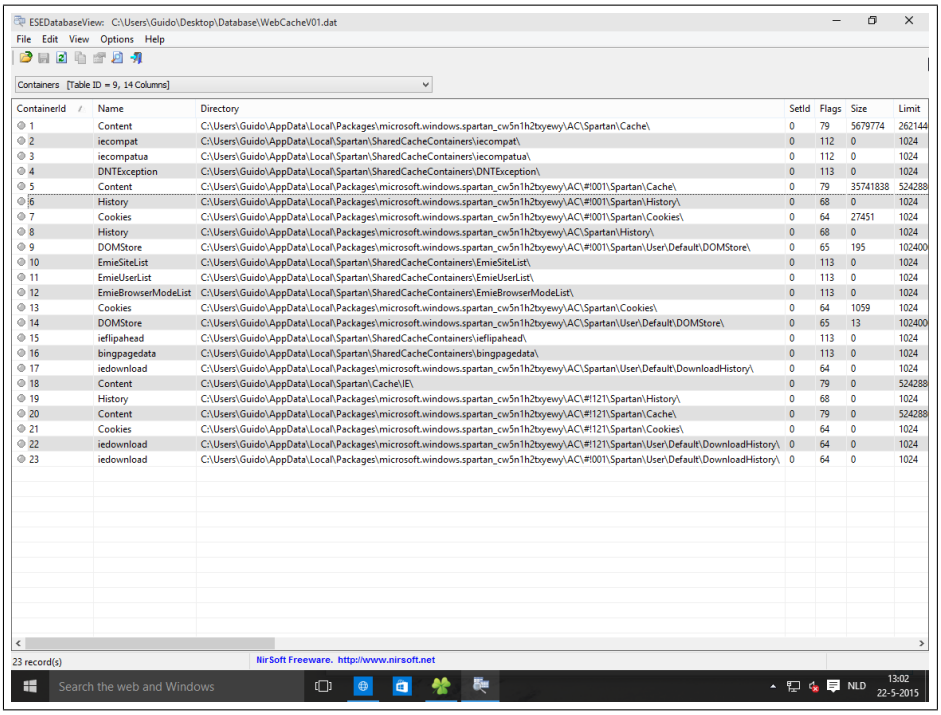


Figure 4.2: Location of folders in disk.

## 4.3 Cookies

Project Spartan stores its cookies in a dispersed fashion as well, which differs per user, hence the `%LocalAppData%` environment variable:

Project Spartan's cookie location

```
%LocalAppData%\packages\microsoft.windows.spartan_{PackageID}\
AC\#!001\Spartan\Cookies\
```

The cookies are stored in `txt` files with a randomly chosen name. This is an example of a cookie '1YTEYKVD.txt':

Project Spartan's cookie example

```
gglckVUogwAAAAfJxZP1Heveresttech.net/
2147484672224664780830443727142055259030443526*
```

Project Spartan knows which cookie file belongs to which domain as this is being tracked in the WebCachev01.dat database.

## 4.4 Bookmarks

Project Spartan stores its bookmarks in a dispersed fashion as well, which differs per user, hence the `%LocalAppData%` environment variable:

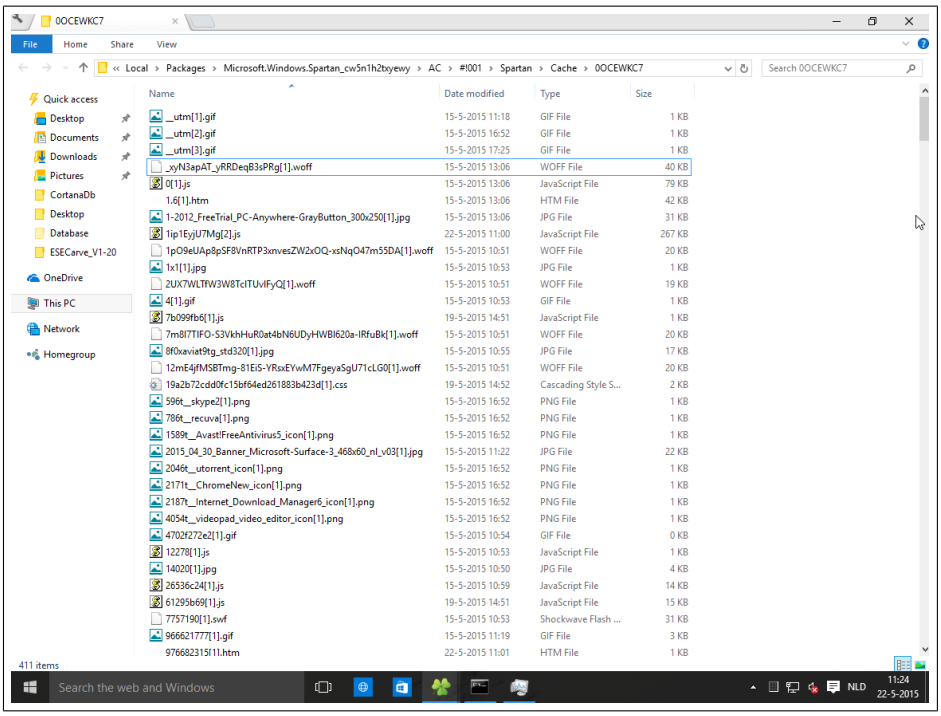


Figure 4.3: Cache folder screenshot.

**Project Spartan's bookmarks location**

```
%LocalAppData%\packages\microsoft.windows.spartan_{PackageID}\
AC\Spartan\User\Default\Favorites
```

Figure 4.4 shows a screenshot of the bookmarks folder.

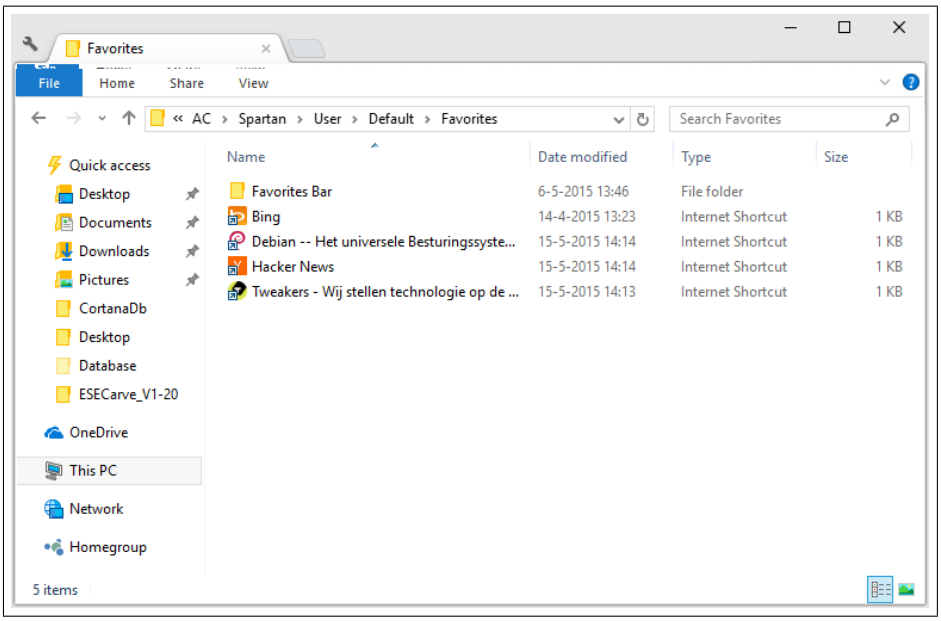


Figure 4.4: Bookmarks folder screenshot.

## 4.5 Visited URLs

The visited URLs is a form of Metadata that reveals information of what URLs the user browsed. It does not show the actual content of the web pages, but it is still valuable information for forensic investigators. The URLs are stored within the database file we covered in section 4.1. The following screenshot (see figure 4.5) gives an impression of these artefacts. Some columns are not shown, which also reveal information about dates and time, and which can be useful when creating time lines:

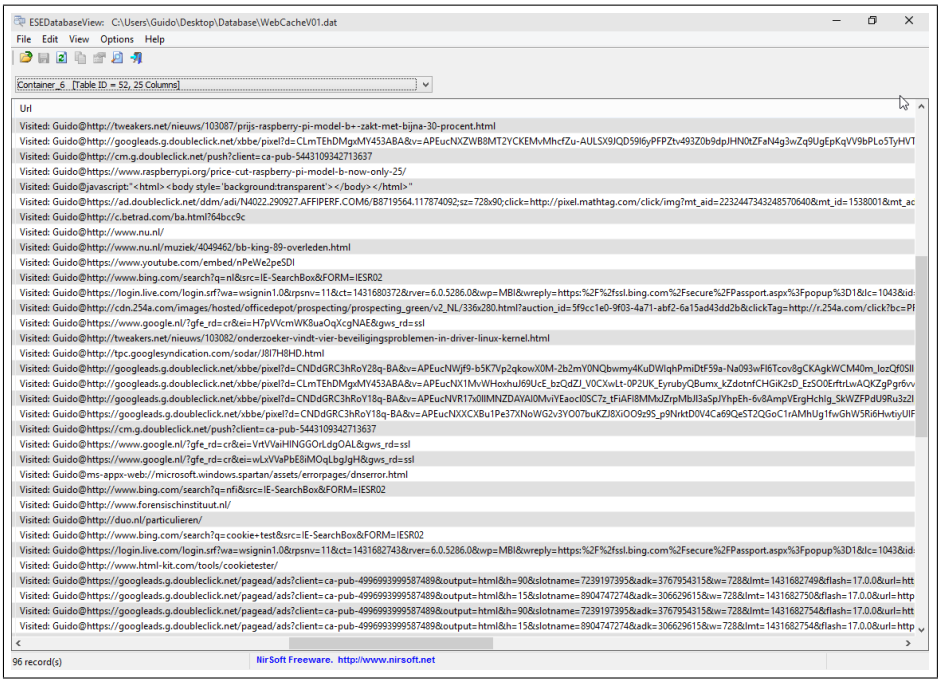


Figure 4.5: Visited URLs container screenshot.

## 4.6 Download history

Downloads history is also found in the database file we covered in section 4.1. The container name is 'iedownload'. There are multiple containers under that name, but it is container ID 17 on our system (see figure 4.6). The values are hex encoded and thus need to be converted to ASCII readable text in order to read it (see appendix B for a more detailed overview). The downloads are temporarily stored in the cache location we covered in section 4.2. When the download is completed, it is no longer to be found there, as it is then moved to the download folder.

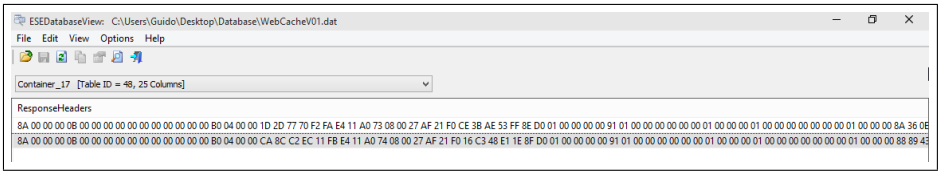


Figure 4.6: Download history artefacts in the database, which are hex encoded.

## 4.7 Web Notes

Interestingly enough, Microsoft Spartan stores its Web Notes in the bookmarks folder as well (see figure 4.7):

Project Spartan's Web Notes location

```
%LocalAppData%\packages\microsoft.windows.spartan_{PackageID}\
AC\Spartan\User\Default\Favorites
```

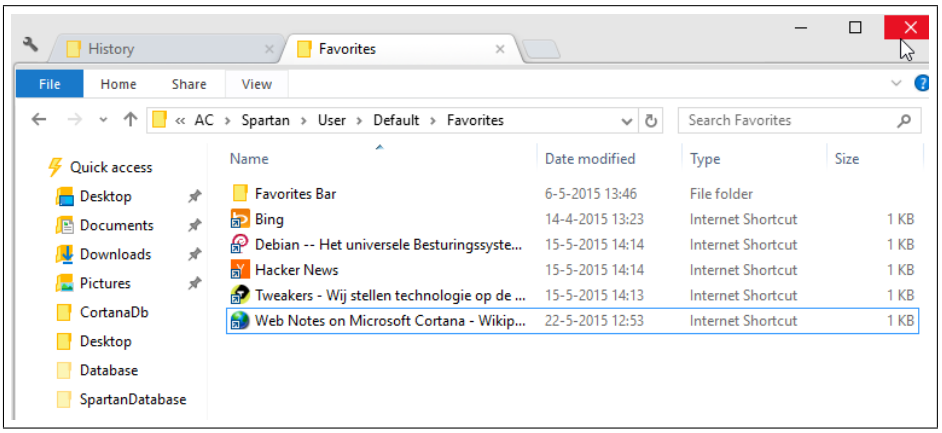


Figure 4.7: Web Notes location screenshot, which are stored as part of the favourites.

However, temporary files are stored the History folder (see figure 4.8):

Project Spartan's temporary Web Notes location

```
%LocalAppData%\packages\microsoft.windows.spartan_{PackageID}\
AC\#!001\Spartan\History\
```

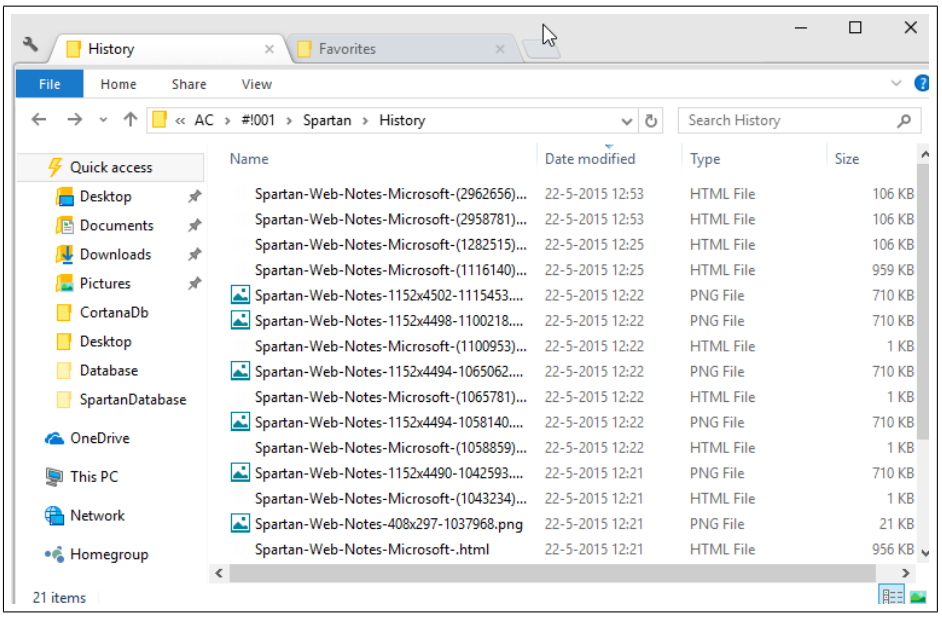


Figure 4.8: Web Notes artefacts that are temporarily stored in the History folder, which contains drafts that are not saved to the favourites.

## 4.8 Cortana

Project Spartan uses Bing as its search engine for Cortana search queries. This is an experimental feature that was not available to our country yet (The Netherlands), so we used an OpenVPN[5] connection to the US to test this new feature. Spartan stores its search queries inside the database file we covered in section 4.1. The container name is 'DependencyEntry_5'(see figure 4.9).

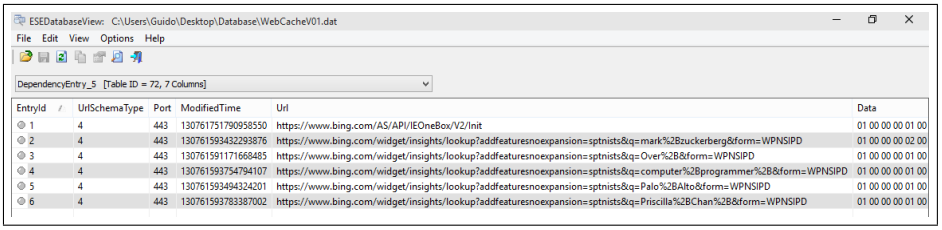


Figure 4.9: Cortana search artefacts that are stored inside the database file, viewed with ESEDatabaseView.

## 4.9 Reading list

The reading list is stored inside a separate database, also separate for each user. We added a web page to the reading list, which could be found inside the database when we opened the database with ESEDatabaseView (see figure4.10).

Project Spartan's Reading List database location

```
%LocalAppData%\Packages\Microsoft.Windows.Spartan_{PackageID}\
AC\Spartan\User\Default\DataStore\Data\nouser1\120712-0049\
DBStore\spartan.edb
```

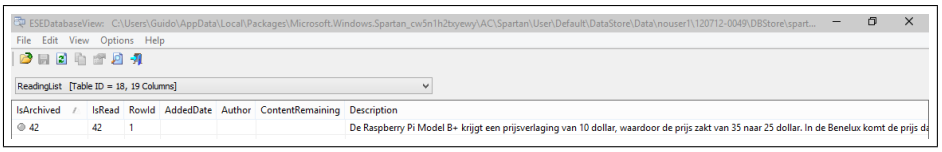


Figure 4.10: Reading list artefacts that are stored inside the database file, viewed with ESEDatabaseView.

## 4.10 Tiles

Since Windows 8, tiles are available and can be modifiable. This feature is included in Project Spartan as well. It consists of fonts, colours and interface elements for applications. The tiles are not stored in the ESE database but can be found on the disk. They are located at:

Project Spartan's tiles location

```
%LocalAppData%\Packages\Microsoft.Windows.Spartan_{PackageID}\
AC\#!001\Spartan\User\Default\Tiles
```

## 4.11 Private browsing

For analysing artefacts for InPrivate browsing, we needed to upgrade our system with a newer version of Windows 10 (build 10122), which had the new Spartan Browser that supported InPrivate browsing. In order to retrieve the InPrivate pages visited, we used a tool created by Chivers, named ESECarve. This tool was intended to retrieve InPrivate browsing artefacts from the IE 10 browser. Due to incompatibility of the software on Windows 10, it was necessary to move the folder containing the database files (.chk, .log and WebCacheV01.dat) to an earlier version of Windows. We were successfully able to recover Project Spartan InPrivate pages with Windows 7 and the ESECarve tool (see figure 4.11).

The life cycle of InPrivate logs described in [6] was verified with Project Spartan. Indeed the InPrivate history could be recovered from the same session with ESEDatabaseView (see figure 4.12) but as soon as we cleared the cache and restarted the browser these entries disappeared from the container. However the entries were recovered using the ESECarve tool that uses the .log and .chk files to recover information about the InPrivate browsing.

[0]We would like to thank H. Chivers to make his tool available for our research.

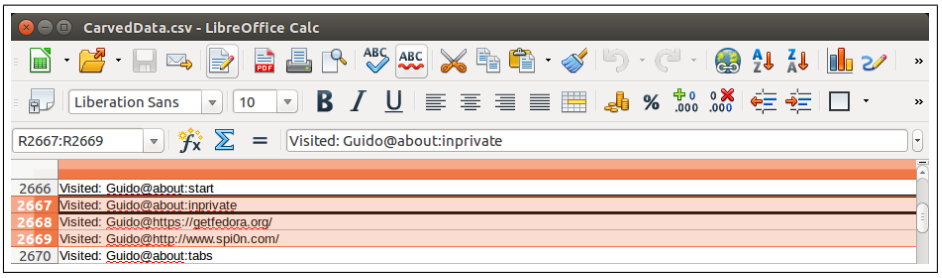


Figure 4.11: InPrivate browsed URLs screenshot of a CSV file produced by ESECarve, which can still be carved after clearing the cache.

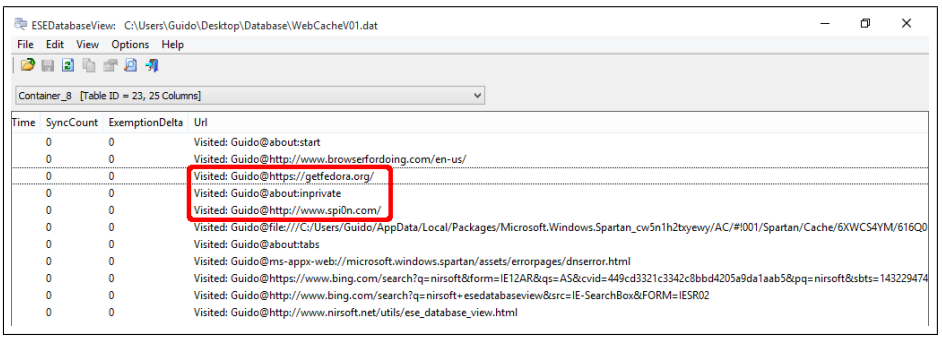


Figure 4.12: InPrivate browsed URLs screenshot in ESEDatabaseView, which are erased after clearing the cache, but can still be carved with ESECarve.

## 4.12 Features not (yet) integrated in Project Spartan

The version of Project Spartan available in the latest Windows 10 build (10122) does not include all the features that should be present on a browser. New features are awaited such as the password storage or extensions capability. IThome [2], leaked some screenshots of an unreleased build of Windows 10 (10123). These screenshots show new features implemented in Project Spartan such as:

- **Credential storage** As of yet, Project Spartan does not enable users to store their credentials whenever they login to a certain website.
- **Forms storage** As of yet, Project Spartan does not enable users to store forms whenever a user fills in a digital form. However, we found the following key that's similar to a IE 11 registry key (albeit still empty) it is thus likely that the forms will be stored there:

  Project Spartan's IntelliForms registry key

  ```
  HKEY_USERS\{User-SID}\SOFTWARE\Classes\Local Settings\Software\Microsoft\Windows\Curr
  ```

- **New features in Cortana** Current features, such as Cortana, may change over time or have added features, which is also interesting for future research.

These features could not be investigated as the 10123 release was not available at the time of writing.

Other potential features that are currently not part of Project Spartan:

- **Synchronisation** Current popular browsers are currently offering synchronisation of passwords, bookmarks and such. It would not be a far fetched idea that Microsoft may implement such a feature in later development versions of Project Spartan, or final versions of EDGE.

# Chapter 5

# Results

This section presents what are the results of our investigation on the Project Spartan browser. First the similarities and differences found between Project Spartan and IE v10+ will be described. This is followed by a description of the automated tool created to find the missing artefacts that are not documented in the ESE database.

## 5.1 Project Spartan vs. Internet Explorer (similarities and differences)

The investigation performed in this research, highlighted the Project Spartan artefacts. This section compares the artefacts found in Project Spartan with the latest versions of IE. This comparison is done as the artefacts created by the two browsers are extremely similar. At the time of writing not all the features of Project Spartan are available. It is thus difficult to deduce all the similarities and differences.

First of all, it is worth mentioning that the back end of Project Spartan is really similar to IE v10+. They both use the same database engine, named ESE database, in order to store information about user activity and to provide a way of recovering crashes occurring in software. It is understandable that these two browsers use the same database engine as the ESE database is used as the core system of many Windows-like features such as Microsoft Exchange Server, Active Directory and Desktop Search [1]. As a result, the structure of the Project Spartan database is also really similar to the latest versions of IE v10+. These allow most of the software created to find artefacts in the ESE database to work in with Project Spartan. However some tweaks need to be implemented to make them work with Windows 10. [1]

However, new features have been introduced with Project Spartan. These features introduces new artefacts that can be of considerable importance in forensic investigations. The new features have been documented in Chapter 4. As an example, the information stored by Cortana can be valuable for an investigator as it stores the values that are searched using the engine. In this

[1]The tweaks needed to run ESECarve for Project Spartan on Windows 10 have been forwarded to Chivers

database suggestions made to the user (based on its profile) by the engine are also stored. Other new features such as the reading list or the Web Notes are likely to be of great interest to an investigator.

To conclude, the structure of Project Spartan is in the end similar to the latest versions of IE. New artefacts appeared as the browser offers features that were not implemented on IE. This artefacts have been documented and the upcoming section presents a proof of concept reuniting the artefacts that were not found in the ESE database.

## 5.2 Automated tool

Not all the artefacts are stored in the ESE database, that is why the authors created a proof of concept able to retrieve the missing artefacts. The script does not retrieve the artefacts present in the database as this database can be read with ESEDatabaseView or with the ESECarve tool. The goal was not to reinvent the wheel but to complete the tool present in the open source community. This tool (named SpartanLeftovers) can be run next to ESECarve to retrieve the most valuable artefacts from the Project Spartan browser. The script is written in PowerShell 3.0 and allows an investigator to easily summarise the location of the missing artefact in clear and readable `csv` format. SpartanLeftovers is open source and available in appendix C. The artefacts that are targeted are the favourites, the web notes, the stored tiles and the last unexpectedly closed tabs. Figure 5.1 shows an output of the script. The script lists all the files present in the related directories with their path, creation time, last accessed time, last modification time, owner of the file, attributes and size. From a forensic standpoint the tool can be run on an mounted disk and does not write on the targeted disk. It has been chosen not to access and carve the files in order not to change the access time values this is why the tool only provides the location of the files. The following figure shows the hash difference created using `FTK`:

`hash difference before/after running the tool`

```
Hashes created before running the tool
 MD5 checksum:    8713ad582467c4239402afd0cf055c32
 SHA1 checksum:   bb775dbb75f2ac06ab0c6a870a334d2542d776d7
Hashes created after running the tool
 MD5 checksum:    8713ad582467c4239402afd0cf055c32
 SHA1 checksum:   bb775dbb75f2ac06ab0c6a870a334d2542d776d7
```

It is however advisable to use a write blocker to prevent the connection from the disk to the forensic station to change the disk image and thus the hash.

| Name | CreationTime | CreationTimeUtc | LastWriteTime | LastWriteTimeUtc | LastAccessTime | LastAccessTimeUtc | IsReac | Mode | Length | Attributes | Owner | Exists | DirectoryName |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Spartan-Web-Notes-1028x4523 | 06/05/2015 07:37 | 06/05/2015 05:37 | 06/05/2015 07:37 | 06/05/2015 05:37 | 06/05/2015 07:37 | 06/05/2015 05:37 | False | -a--- | 2E+06 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Spartan-Web-Notes-Hotmail--n. | 06/05/2015 07:37 | 06/05/2015 05:37 | 06/05/2015 07:37 | 06/05/2015 05:37 | 06/05/2015 07:37 | 06/05/2015 05:37 | False | -a--- | 703 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Spartan-Web-Notes-1028x4463 | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | False | -a--- | 1E+06 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Spartan-Web-Notes-471x357-5( | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | False | -a--- | 158654 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Spartan-Web-Notes-Hotmail--n( | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | False | -a--- | 2E+06 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Spartan-Web-Notes-1028x4468 | 12/05/2015 04:15 | 12/05/2015 02:15 | 12/05/2015 04:15 | 12/05/2015 02:15 | 12/05/2015 04:15 | 12/05/2015 02:15 | False | -a--- | 1E+06 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Spartan-Web-Notes-Hotmail--n( | 12/05/2015 04:15 | 12/05/2015 02:15 | 12/05/2015 04:15 | 12/05/2015 02:15 | 12/05/2015 04:15 | 12/05/2015 02:15 | False | -a--- | 2E+06 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Spartan-Web-Notes-1024x978-( | 12/05/2015 04:16 | 12/05/2015 02:16 | 12/05/2015 04:16 | 12/05/2015 02:16 | 12/05/2015 04:16 | 12/05/2015 02:16 | False | -a--- | 192784 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Spartan-Web-Notes-SNEOS3-Hot | 12/05/2015 04:16 | 12/05/2015 02:16 | 12/05/2015 04:16 | 12/05/2015 02:16 | 12/05/2015 04:16 | 12/05/2015 02:16 | False | -a--- | 668 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Spartan-Web-Notes-1024x978-( | 12/05/2015 04:16 | 12/05/2015 02:16 | 12/05/2015 04:16 | 12/05/2015 02:16 | 12/05/2015 04:16 | 12/05/2015 02:16 | False | -a--- | 192784 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Spartan-Web-Notes-SNEOS3-Hot | 12/05/2015 04:16 | 12/05/2015 02:16 | 12/05/2015 04:16 | 12/05/2015 02:16 | 12/05/2015 04:16 | 12/05/2015 02:16 | False | -a--- | 668 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Bing.url | 24/04/2015 05:19 | 24/04/2015 03:19 | 24/04/2015 05:19 | 24/04/2015 03:19 | 24/04/2015 05:19 | 24/04/2015 03:19 | False | -a--- | 208 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| os3.url | 06/05/2015 04:26 | 06/05/2015 02:26 | 06/05/2015 04:26 | 06/05/2015 02:26 | 06/05/2015 04:26 | 06/05/2015 02:26 | False | -a--- | 147 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| desktop.ini | 06/05/2015 06:18 | 06/05/2015 04:18 | 06/05/2015 06:18 | 06/05/2015 04:18 | 06/05/2015 06:18 | 06/05/2015 04:18 | False | #NOM? | 80 | Hidden, Sy | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Web Notes on Hotmail nieuws | 12/05/2015 04:15 | 12/05/2015 02:15 | 12/05/2015 04:15 | 12/05/2015 02:15 | 12/05/2015 04:15 | 12/05/2015 02:15 | False | -a--- | 246 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| Hotmail nieuws video sport ge | 12/05/2015 04:15 | 12/05/2015 02:15 | 12/05/2015 04:15 | 12/05/2015 02:15 | 12/05/2015 04:15 | 12/05/2015 02:15 | False | -a--- | 101 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| {CE32C15F-F898-11E4-A072-080 | 12/05/2015 04:20 | 12/05/2015 02:20 | 12/05/2015 04:20 | 12/05/2015 02:20 | 12/05/2015 04:20 | 12/05/2015 02:20 | False | -a--- | 4608 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| RecoveryStore.{739C0908-F898 | 12/05/2015 04:20 | 12/05/2015 02:20 | 12/05/2015 04:20 | 12/05/2015 02:20 | 12/05/2015 04:20 | 12/05/2015 02:20 | False | -a--- | 5120 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| {739C090A-F898-11E4-A072-08( | 12/05/2015 04:44 | 12/05/2015 02:44 | 12/05/2015 04:44 | 12/05/2015 02:44 | 12/05/2015 04:44 | 12/05/2015 02:44 | False | -a--- | 302080 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| {C0F169D8-F898-11E4-A072-08( | 12/05/2015 04:44 | 12/05/2015 02:44 | 12/05/2015 04:44 | 12/05/2015 02:44 | 12/05/2015 04:44 | 12/05/2015 02:44 | False | -a--- | 209920 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| logoimage.png | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | False | -a--- | 2284 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| hires.png | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | False | -a--- | 3218 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| msapplication.xml | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | False | -a--- | 639 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| msapplication.xml | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | 12/05/2015 04:14 | 12/05/2015 02:14 | False | -a--- | 643 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| msapplication.xml | 12/05/2015 04:15 | 12/05/2015 02:15 | 12/05/2015 04:15 | 12/05/2015 02:15 | 12/05/2015 04:15 | 12/05/2015 02:15 | False | -a--- | 509 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |
| msapplication.xml | 12/05/2015 04:15 | 12/05/2015 02:15 | 12/05/2015 04:15 | 12/05/2015 02:15 | 12/05/2015 04:15 | 12/05/2015 02:15 | False | -a--- | 500 | Archive | BUILTIN\Admin | True | C:\Users\user\appdata\local\packages\Microsoft.Windows.Spartan_cw |

Figure 5.1: Output of the PowerShell script when dumping all missing artefacts location

# Chapter 6

# Conclusion

Currently the way in which and the location where Project Spartan stores its artefacts is very similar to previous versions of Internet Explorer. The browser relies heavily on the ESE database structure, which makes current ways of collecting artefacts not much harder. Most artefacts of features have been analysed that are part of the current development builds of Project Spartan and we suspect that current forensic toolkits that also harvest artefacts of IE will not need to drastically alter their harvesting techniques to also gather artefacts from Project Spartan. Toolkit developers are advised to use the path locations specified in this paper to acquire the artefacts of Project Spartan. The new features such as the Web Notes or Cortana integration can also give insight into the digital footprint a user can leave on a system. These new features should also be added to existing forensic toolkits as well. It should be noted that Project Spartan is still in development and artefacts may change over time (see chapter 7 for more on future work considerations).

The authors also developed a tool which gathers some information analysed in an automated way. The tool is open source and has been designed for forensic/research purposes. It provides a way of recovering the artefacts, left behind by the Project Spartan browser, that are not stored in the ESE database and/or that cannot be retrieved with the ESECarve tool developed by Chivers. The source code (Appendix C) is open to any improvements.

# Chapter 7

# Future work

This research outlines some new artefacts that can be gathered within the current development versions of Project Spartan. However, there are a couple of elements to be considered for future work.

This research should be reviewed whenever Microsoft releases a final and stable version of Edge. This research only focused on the development versions of Project Spartan. Current features that have been analysed during this research may change over time, as well as new features that might be added in the future, which we already outlined in section 4.12. Features like a credential manager, forms storage, synchronisation of connected device are features that would be very interesting subjects for research once they have been implemented.

Currently, the ESE database structure has not been greatly researched, and this also differs per implementation that uses the ESE database structure, such as IE, Exchange and now Project Spartan.

Also, as InPrivate (private browsing) artefacts can still be harvested from the ESE database, it would be good to see Microsoft fix this and perform a similar project as done by Chivers[6] to see if these artefacts can still be harvested. However questions arise if this possibility to harvest such information, with the right forensics skills, was made intentionally for forensics purposes.

# Bibliography


[1] Extensible storage engine. Microsoft Developer Network, 2012.

[2] Exclusive broke the news: Win10 preview version 10123, edge browser new change. IT House Original, 2015.

[3] Forensically interesting spots in the windows 7, vista and xp file system and registry. irongeek, 2015.

[4] Ie passview. Nirsoft, 2015.

[5] Openvpn. OpenVPN Technologies, Inc, 2015.

[6] Howard Chivers. Private browsing: A window of forensic opportunity. 2013.

[7] Jens Lorenz. Notepad++ Plugins - Browse Files at SourceForge.net. `http://sourceforge.net/projects/npp-plugins/files/`, 2015.

[8] Bonnie Malmström and Philip Teveldal. Forensic analysis of the ese database in internet explorer 10. 2013.

[9] Joachim Metz. Extensible storage engine (ese) database file (edb) format specification. 2010.

[10] Nir Sofer. ESEDatabaseView - View/Open ESE Database Files (Jet Blue / .edb files). `http://www.nirsoft.net/utils/ese_database_view.html`, 2015.

[11] Junghoon Oh, Seungbong Lee, and Sangjin Lee. Advanced evidence collection and analysis of web browser activity. *digital investigation*, 8:S62–S70, 2011.

[12] Huwida Said, Noora Al Mutawa, Ibtesam Al Awadhi, and Mario Guimaraes. Forensic analysis of private browsing artifacts. In *Innovations in information technology (IIT), 2011 International conference on*, pages 197–202. IEEE, 2011.

[13] Jason Weber. Project spartan and the windows 10 january preview build. Microsoft IE, 2015.

# Appendices

# Appendix A

# Spartan's WebCache database

As previously mentioned before, Microsoft Project Spartan uses the same Extensible Storage Engine (ESE) database structure as previous versions of IE. The IE 10 ESE database structure has been researched in-depth by Malmström and Teveldal. [8]

When opening this file with a hex editor, we can see that this version still uses the same format that previous versions use:

Sample of a Spartan WebCache hexdump

```
gkroon@desktop-41:~\$ xxd -c 16 -g 4 WebCacheV01.dat | head -n 20
0000000: 8c45e204 efcdab89 20060000 00000000  .E...... .......
0000010: 1bce0700 00000000 453ea999 1f0f0b17  ........E>......
0000020: 04730b00 00000000 00000000 00000000  .s..............
0000030: 00000000 03000000 03032800 44010000  ..........(.D...
0000040: 1e360e0f 05739b0e 37330e0f 0573770e  .6...s..73...sw.
0000050: 68022600 43010000 1e360e0f 05739b0e  h.&.C....6...s..
0000060: 03032800 44010000 01000000 d1515f91  ..(.D........Q_.
0000070: 1e0f0b17 0473470c 00000000 00000000  .....sG.........
0000080: 00000000 00000000 00000000 00000000  ................
0000090: 00000000 00000000 00000000 00000000  ................
00000a0: 00000000 00000000 00000000 00000000  ................
00000b0: 00000000 00000000 00000000 00000000  ................
00000c0: 00000000 00000000 00000000 00000000  ................
00000d0: 00000000 49000000 0a000000 00000000  ....I...........
00000e0: 5a270000 00000000 14000000 00800000  Z'..............
00000f0: 00000000 00000000 00000000 00000000  ................
0000100: 00000000 00000000 00000000 00000000  ................
0000110: 00000000 00000000 00000000 00000000  ................
0000120: 00000000 00000000 00000000 00000000  ................
0000130: 00000000 00000000 00000000 00000000  ................
```

The above hex dump of the database headers can be interpreted as follows [8]:

| Hex offset | Hex value | Description |
|---|---|---|
| 000 - 003 | 8c45e204 | Database checksum |
| 004 - 007 | efcdab89 | File signature |
| 008 - 011 | 20060000 | File format version |
| 010 - 017 | 1bce0700 00000000 | Database time |
| 018 - 033 | 453ea999 [...] 00000000 | Database signature |
| 034 - 037 | 03000000 | Database state |
| 0e8 - 0eb | 14000000 | File format revision |
| 0ec - 0ef | 00800000 | Page size in bytes |

Table A.1: Planning.

This can be verified when analysing the database with `esentutl`, which is installed by default on every Windows system. Note that the database is using little endian, so when comparing with a hex dump, every byte range needs to be read in reverse order. For example, the page size is 0x00800000 which we need to reverse in Endianess, so that gives us 0x00008000, which is 32768 in decimal, which means it is 32768 bytes, or 32 KiB per page. Every page's offset starts at 32 KiB increments, which is offset 0x8000 when exploring in a hex dump. If we go to this offset, we can see the start of the first page. The second starts at 64 KiB, and so on.

Database information using esentutl

```
C:\WINDOWS\system32>esentutl -mh WebCacheV01.dat
```

```
Extensible Storage Engine Utilities for Microsoft(R) Windows(R)
Version 10.0
Copyright (C) Microsoft Corporation. All Rights Reserved.

Initiating FILE DUMP mode...
    Database: C:\Users\Guido\Desktop\Database\WebCacheV01.dat


DATABASE HEADER:
Checksum Information:
Expected Checksum: 0x04e2458c
  Actual Checksum: 0x04e2458c

Fields:
        File Type: Database
         Checksum: 0x4e2458c
   Format ulMagic: 0x89abcdef
   Engine ulMagic: 0x89abcdef
 Format ulVersion: 0x620,20
 Engine ulVersion: 0x620,20
Created ulVersion: 0x620,20
     DB Signature: Create time:04/23/2015 13:15:31.005
                    Rand:2578005573
Computer:
         cbDbPage: 32768
           dbtime: 511515 (0x7ce1b)
            State: Clean Shutdown
     Log Required: 0-0 (0x0-0x0)
    Log Committed: 0-0 (0x0-0x0)
   Log Recovering: 0 (0x0)
  GenMax Creation: 00/00/1900 00:00:00.000
         Shadowed: Yes
       Last Objid: 73
     Scrub Dbtime: 0 (0x0)
       Scrub Date: 00/00/1900 00:00:00
     Repair Count: 0
      Repair Date: 00/00/1900 00:00:00.000
 Old Repair Count: 0
  Last Consistent: (0x144,28,303)  05/15/2015 16:54:30.973
      Last Attach: (0x143,26,268)  05/15/2015 16:51:55.955
      Last Detach: (0x144,28,303)  05/15/2015 16:54:30.973
    Last ReAttach: (0x0,0,0)  00/00/1900 00:00:00.000
             Dbid: 1
    Log Signature: Create time:04/23/2015 13:15:30.803
                    Rand:2438943185
Computer:
       OS Version: (10.0.10074 SP 0 NLS ffffffff.ffffffff)
```

# Appendix B

# Download history

The download history is part of the ESE database. This is an example of :

Project Spartan's Download history raw hex value

```
8A 00 00 00 0B 00 00 00 00 00 00 00 00 00 00 00 B0 04 00 00
CA 8C C2 EC 11 FB E4 11 A0 74 08 00 27 AF 21 F0 16 C3 48 E1
1E 8F D0 01 00 00 00 00 91 01 00 00 00 00 00 00 01 00 00 00
01 00 00 00 00 00 00 00 01 00 00 00 88 89 43 00 00 00 00 00
75 05 25 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00
00 00 00 00 01 00 00 00 00 00 00 00 06 00 00 00 00 00 00 00
98 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01 00 00 00 00 00 00 00 00 00 01 00 01 00 00 00 01 00 00 00
02 00 00 00 6C A8 D0 CE 00 00 00 00 00 00 00 00 A0 8B 1E 50
5E 00 00 00 02 00 00 00 00 00 00 00 00 EE 09 53 5E 00 00 00
DA D9 CA 7F FD 7F 00 00 04 00 00 00 5E 00 00 00 01 00 00 00
FD 7F 00 00 30 DB 16 41 56 00 00 00 00 00 00 00 9A 09 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 B0 EE 09 53
00 00 00 00 C8 EE 09 53 5E 00 00 00 C0 08 65 4C 5E 00 00 00
01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 50 00 69 00 72 00 69 00 66 00 6F 00 72 00 6D 00
20 00 4C 00 74 00 64 00 00 00 68 00 74 00 74 00 70 00 3A 00
2F 00 2F 00 66 00 69 00 6C 00 65 00 68 00 69 00 70 00 70 00
6F 00 2E 00 63 00 6F 00 6D 00 2F 00 64 00 6F 00 77 00 6E 00
6C 00 6F 00 61 00 64 00 5F 00 72 00 65 00 63 00 75 00 76 00
61 00 2F 00 64 00 6F 00 77 00 6E 00 6C 00 6F 00 61 00 64 00
2F 00 37 00 63 00 63 00 38 00 32 00 37 00 34 00 37 00 31 00
35 00 63 00 36 00 34 00 62 00 64 00 36 00 66 00 33 00 35 00
30 00 62 00 65 00 64 00 64 00 35 00 39 00 62 00 65 00 30 00
33 00 62 00 32 00 2F 00 00 00 61 00 70 00 70 00 6C 00 69 00
63 00 61 00 74 00 69 00 6F 00 6E 00 2F 00 78 00 2D 00 6D 00
73 00 64 00 6F 00 77 00 6E 00 6C 00 6F 00 61 00 64 00 00 00
43 00 3A 00 5C 00 55 00 73 00 65 00 72 00 73 00 5C 00 47 00
75 00 69 00 64 00 6F 00 5C 00 41 00 70 00 70 00 44 00 61 00
74 00 61 00 5C 00 4C 00 6F 00 63 00 61 00 6C 00 5C 00 50 00
```

```
61 00 63 00 6B 00 61 00 67 00 65 00 73 00 5C 00 6D 00 69 00
63 00 72 00 6F 00 73 00 6F 00 66 00 74 00 2E 00 77 00 69 00
6E 00 64 00 6F 00 77 00 73 00 2E 00 73 00 70 00 61 00 72 00
74 00 61 00 6E 00 5F 00 63 00 77 00 35 00 6E 00 31 00 68 00
32 00 74 00 78 00 79 00 65 00 77 00 79 00 5C 00 41 00 43 00
5C 00 23 00 21 00 30 00 30 00 31 00 5C 00 53 00 70 00 61 00
72 00 74 00 61 00 6E 00 5C 00 43 00 61 00 63 00 68 00 65 00
5C 00 30 00 4F 00 43 00 45 00 57 00 4B 00 43 00 37 00 5C 00
72 00 63 00 73 00 65 00 74 00 75 00 70 00 31 00 35 00 32 00
5B 00 31 00 5D 00 2E 00 65 00 78 00 65 00 00 00 68 00 74 00
74 00 70 00 3A 00 2F 00 2F 00 66 00 73 00 33 00 34 00 2E 00
66 00 69 00 6C 00 65 00 68 00 69 00 70 00 70 00 6F 00 2E 00
63 00 6F 00 6D 00 2F 00 36 00 38 00 39 00 33 00 2F 00 38 00
35 00 62 00 37 00 32 00 38 00 31 00 36 00 34 00 66 00 32 00
65 00 34 00 31 00 35 00 38 00 62 00 39 00 35 00 34 00 37 00
37 00 30 00 37 00 31 00 38 00 38 00 38 00 34 00 34 00 64 00
33 00 2F 00 72 00 63 00 73 00 65 00 74 00 75 00 70 00 31 00
35 00 32 00 2E 00 65 00 78 00 65 00 00 00 43 00 3A 00 5C 00
55 00 73 00 65 00 72 00 73 00 5C 00 47 00 75 00 69 00 64 00
6F 00 5C 00
```

If one were to convert this directly to ASCII, one would get a similar text like this:

Project Spartan's Download history raw ASCII value

```
??????????????????????????????????????????????'??!?????H??
????????????????????????????????????????????????C????????
u??%????????????????????????????????????????????????????
????????????????????????????????????????????????????????
??????l?????????????????????????^????????????????????^?????
??????????????????^??????????????????O?????V???????????????
????????????????????????????????????????^????????e??^?????
????????????????????????????????????????????????????????
????????????????????????????????P??i??r??i??f??o??r??m??
??L??t??d?????h??t??t??p??:??/??/??f??i??l??e??h??i??p??p??o
??.??c??o??m??/??d??o??w??n??l??o??a??d??_??r??e??c??u??v??a
??/??d??o??w??n??l??o??a??d??/???7??c??c??8??2???7??4??7??1??5
??c??6??4??b??d??6??f??3??5??0??b??e??d??d??5??9??b??e??0??3
??b??2??/?????a??p??p??l??i??c??a??t??i??o??n??/??x??-??m??s
??d??o??w??n??l??o??a??d?????C??:??\??U??s??e??r??s??\??G??u
??i??d??o??\??A??p??p??D??a????????????????????????????????
????????????????????????????????????????????????????????
????????????????????????????????????????????????????????
????????????????????????????????????????????????????????
????????????????????????????????????????????????????????
????????????????????????????????????????????????????????
????????????????????????????????????????????????????????
????????????????????????????????????????????????????????
????????????????????????????????????????????????????????
????????????????????????????????????????????????????????
```

```
??????????????????????????????????????????
```

This is not very helpful yet as some character cannot be converted to ASCII. If one were to omit all the unnecessary signs (here question marks) one would get the following text:

Project Spartan's Download history trimmed ASCII value

```
'!HCu%l^^^0V^e^Piriform Ltdhttp://filehippo.com/download_rec
uva/download/7cc8274715c64bd6f350bedd59be03b2/application/x-
msdownloadC:\Users\Guido\AppDa
```

One can derive from this string that Piriform Recuva from `filehippo.com` has been downloaded with Project Spartan.

# Appendix C

# Powershell script

SpartanLeftovers source code

```
<#
.SYNOPSIS
    Carving tool of the Spartan artefacts. UvA - SNE 2015
.NOTES
    File Name      : spartanforensic.ps1
    Author         : J Gratchoff (james.gratchoff@os3.nl)
    Prerequisite   : PowerShell V3
    Copyright 2015
#>

$drive = Read-Host " `n Welcome in SpartanLeftovers `n  `n Pleas
e enter the drive letter you wish to analyse"
If ((Test-Path $drive":"))
    {
                        $dest = Read-Host " Enter the destinatio
n drive you wish to store the information"
                        If (($drive -eq $dest))
                        {
                                echo "`n `n In order to be foren
sically sound please choose another path to store your informati
on on. Use c or C for the C:/ drive"
                                exit
                        }
                        else
                        {
                        echo " `n This utility will not store on
the targeted drive however a write blocker is highly recommended
to prevent the connection to modify the targeted device."
                        $username = Read-Host " Enter the usern
ame you wish to investigate"
                        If ((Test-Path $drive":"\Users\$usernam
e))
                        {
                                $spartanname = Get-ChildItem $dr
```

```
ive":"\Users\$username\appdata\local\packages\microsoft.windows.
spartan*
                                $path =  $spartanname.Fullname
                                echo " This tool can be used in
conbination with ESEcarve (written by Chivers) to locate the art
efacts that are not present in the database and new to Microsoft
Edge. `n "
                                $name = Read-Host " Select artef
act and press enter `n 1: Favorites `n 2: Recovery tabs `n 3: We
b Notes `n 4: Tiles `n 5: Dump all the missing artefacts! `n all
this commands output to a csv file located on the desired direct
ory"

                                If($name -eq 1)
                                 {
                                        Get-ChildItem -Recurse $
path\AC\Spartan\User\Default\Favorites -force| ForEach-Object {$
_ | add-member -name "Owner" -membertype noteproperty -value (ge
t-acl $_.fullname).owner -passthru} | Sort-Object LastAccessTime
| Select Name,CreationTime,CreationTimeUtc,LastWriteTime,LastWri
teTimeUtc,LastAccessTime,LastAccessTimeUtc,IsReadOnly,Mode,Lengt
h,attributes,Owner,Exists,DirectoryName | Export-Csv -Force -NoT
ypeInformation ${dest}:\Favorites.csv
                                        echo "`n `n Dumped the f
avourites locations on:" ${dest}":\Favorites.csv"
                                        exit
                                  }
                                elseIf($name -eq 2)
                                 {
                                        Get-ChildItem -Recurse $
path\AC\Spartan\User\Default\Recovery\Active -force| ForEach-Obj
ect {$_ | add-member -name "Owner" -membertype noteproperty -val
ue (get-acl $_.fullname).owner -passthru} | Sort-Object LastAcce
ssTime | Select Name,CreationTime,CreationTimeUtc,LastWriteTime,
LastWriteTimeUtc,LastAccessTime,LastAccessTimeUtc,IsReadOnly,Mod
e,Length,attributes,Owner,Exists,DirectoryName | Export-Csv -For
ce -NoTypeInformation ${dest}:\Recoverydirectory.csv
                                        echo "`n `n Dumped the r
ecovery tabs locations on:" ${dest}":\Recoverydirectory.csv"
                                        exit
                                 }
                                elseIf($name -eq 3)
                                 {
                                        Get-ChildItem -Recurse $
path\AC\"#!001"\Spartan\History -force| ForEach-Object {$_ | add
-member -name "Owner" -membertype noteproperty -value (get-acl $
_.fullname).owner -passthru} | Sort-Object LastAccessTime | Sele
ct Name,CreationTime,CreationTimeUtc,LastWriteTime,LastWriteTime
Utc,LastAccessTime,LastAccessTimeUtc,IsReadOnly,Mode,Length,attr
ibutes,Owner,Exists,DirectoryName | Export-Csv -Force -NoTypeInf
```

```
ormation ${dest}:\Historydirectory.csv
                                        echo "`n `n Dumped the w
eb notes location on: " ${dest}":\Historydirectory.csv"
                                        exit
                                    }
                                   elseIf($name -eq 4)
                                    {
                                                Get-ChildItem -R
ecurse $path\AC\"#!001"\Spartan\User\Default\Tiles -force| ForEa
ch-Object {$_ | add-member -name "Owner" -membertype notepropert
y -value (get-acl $_.fullname).owner -passthru} | Sort-Object La
stAccessTime | Select Name,CreationTime,CreationTimeUtc,LastWrit
eTime,LastWriteTimeUtc,LastAccessTime,LastAccessTimeUtc,IsReadOn
ly,Mode,Length,attributes,Owner,Exists,DirectoryName | Export-Cs
v -Force -NoTypeInformation ${dest}:\tilesdirectory.csv
                                        echo "`n `n Dumped all t
he tiles locations on: " ${dest}":\tilesdirectory.csv"
                                        exit
                                    }
                                   elseIf($name -eq 5)
                                    {
                                        Get-ChildItem -Recurse $
path\AC\"#!001"\Spartan\History -force| ForEach-Object {$_ | add
-member -name "Owner" -membertype noteproperty -value (get-acl $
_.fullname).owner -passthru} | Sort-Object LastAccessTime | Sele
ct Name,CreationTime,CreationTimeUtc,LastWriteTime,LastWriteTime
Utc,LastAccessTime,LastAccessTimeUtc,IsReadOnly,Mode,Length,attr
ibutes,Owner,Exists,DirectoryName | Export-Csv -Force  -NoTypeIn
formation ${dest}:\Allartefacts.csv
                                        Get-ChildItem -Recurse $
path\AC\Spartan\User\Default\Favorites -force| ForEach-Object {$
_ | add-member -name "Owner" -membertype noteproperty -value (ge
t-acl $_.fullname).owner -passthru} | Sort-Object LastAccessTime
| Select Name,CreationTime,CreationTimeUtc,LastWriteTime,LastWri
teTimeUtc,LastAccessTime,LastAccessTimeUtc,IsReadOnly,Mode,Lengt
h,attributes,Owner,Exists,DirectoryName | Export-Csv -Force  -Ap
pend ${dest}:\Allartefacts.csv
                                        Get-ChildItem -Recurse $
path\AC\Spartan\User\Default\Recovery\Active -force| ForEach-Obj
ect {$_ | add-member -name "Owner" -membertype noteproperty -val
ue (get-acl $_.fullname).owner -passthru} | Sort-Object LastAcce
ssTime | Select Name,CreationTime,CreationTimeUtc,LastWriteTime,
LastWriteTimeUtc,LastAccessTime,LastAccessTimeUtc,IsReadOnly,Mod
e,Length,attributes,Owner,Exists,DirectoryName | Export-Csv -For
ce -Append ${dest}:\Allartefacts.csv
                                        Get-ChildItem -Recurse $
path\AC\"#!001"\Spartan\User\Default\Tiles -force| ForEach-Objec
t {$_ | add-member -name "Owner" -membertype noteproperty -value
(get-acl $_.fullname).owner -passthru} | Sort-Object LastAccessT
ime | Select Name,CreationTime,CreationTimeUtc,LastWriteTime,Las
```

```
tWriteTimeUtc,LastAccessTime,LastAccessTimeUtc,IsReadOnly,Mode,L
ength,attributes,Owner,Exists,DirectoryName | Export-Csv -Force
-Append  ${dest}:\Allartefacts.csv

                                        echo " `n `n  Dumped all
the missing artefacts locations on:  "${dest}":\Allartefacts.csv
"
                                        exit
                                 }
                                else
                                {
                                        echo " Input not valid.
Please try again"
                                        exit
                                }
                        }
                        else
                        {
                                echo " Username not existing. Pl
ease enter a valid username"
                                exit
                        }
                        }
        }
        else
        {
                echo " Drive is not existing/available. Please e
nter a valid drive letter (e.g: For the C:\ drive enter c or C)"
                exit
        }
```

# Appendix D

# Work separation

| Part | Name | Written by |
|---|---|---|
| Chapter 1 | Introduction | Both |
| Chapter 2 | Related work | James |
| Chapter 3 | Approach | James |
| Chapter 4 | Artefacts Analysis | Both |
| Section 4.1 | Database | Guido |
| Section 4.2 | Cache | Guido |
| Section 4.3 | Cookies | Guido |
| Section 4.4 | Bookmarks | Guido |
| Section 4.5 | Visited URLs | Guido |
| Section 4.6 | Download history | Guido |
| Section 4.7 | Web Notes | Guido |
| Section 4.8 | Cortana | Guido |
| Section 4.9 | Reading list | Guido |
| Section 4.10 | Tiles | James |
| Section 4.11 | Private browsing | Guido |
| Section 4.12 | Features not (yet) integrated in Project Spartan | Guido |
| Chapter 5 | Results | James |
| Chapter 6 | Conclusion | Both |
| Chapter 7 | Future work | Both |
| Appendix A | Spartan's WebCache database | Guido |
| Appendix B | Download history | Guido |
| Appendix C | Powershell script | James |
| Appendix D | Glossary | Guido |

# Glossary

**Active Directory** Microsoft Active Directory is a directory service to keep track of all entities within a domain. It can be compared to a the yellow pages, but then within an IT domain, which keeps track of not only people but also all equipment and such.. 18

**ASCII** American Standard Code for Information Interchange is a character-encoding scheme. 12, v, vi

**Bing** Microsoft search engine for website on the Internet (like the Google search engine, or Yahoo's search engine).. 14

**Cortana** Name of the new Microsoft speech engine, derived from the artificial intelligence character in the Microsoft Halo video game franchise.. 2, 7, 14, 16, 18

**CSV** Comma separated values, used to denote a format structure, which its values are delimited by commas.. 15

**Edge** Name of the new Microsoft web browser engine, which is used in Project Spartan, which is also to be named Edge when it releases.. 2, 4, 17, 22

**Endianess** Ordering of how bytes are read into memory, depending on the computing architecture.. ii

**ESE** Extensible Storage Engine (ESE) is a database structure developed by Microsoft, which they have implemented in some of their product (Internet Explorer, Exchange, Active Directory, Desktop Search, and now also Project Spartan). 5, 6, 7, 15, 18, 19, 21, 22, i, iv

**Exchange** Microsoft Exchange is a mail server, with calendaring and contacting support, which can also be integrated into Microsoft Active Directory.. 18, 22

**HTML** Hyper Text Markup Language, a markup language mostly used to display web page content when using a browser.. 8

**IE** Common abbreviation for Microsoft Internet Explorer.. 2, 3, 4, 5, 6, 8, 15, 16, 18, 19, 21, 22, i

**InPrivate** Microsoft private browsing technology, to browse without leaving traces on the system the browser runs on (like Google Chrome's Incognito, or Firefox's Private Windows.. 15, 22

**JET** Joint Engine Technology, the former name of the ESE database structure.. 5

**Metadata** Term used to denote descriptive information about data. Therefore, metadata is not the actual data itself, but offers some insight into what the actual data may contain.. 7, 11

**PowerShell** A Microsoft scripting language, which offers Cmd-Lets to control the inner working of compatible software as well.. 19

**SpartanLeftovers** A custom designed script developed by the authors to automate some of Project Spartan's artefacts.. 19

**Trident** Name of the Microsoft web browser engine used in Internet Explorer.. 2, 3

**URL** Uniform Resource Locator, used to denote links, or addresses, of remote entities (systems).. 5, 11, 15

**Web Notes** A new feature in Project Spartan one can use to annotate certain things on a web page and then save it for later.. 12, 13, 18