Nothing to see here!

On the awareness of and preparedness and defenses against cloaking malicious web content delivery

Pinoy, J.

Award date:
2021

Link to publication

Open University of the Netherlands

Faculty of Science
Master Computer Science

Nothing to see here!

On the awareness of and preparedness and defenses against cloaking malicious web content delivery

Author:
Jeroen Pinoy

Chair(wo)man:
dr. Fabian van den Broek
Primary Supervisor:
dr. ir. Hugo Jonker

August 26, 2021

Course code:
IM990C

Open
Universiteit

Acknowledgements

During the execution of this research I have received a lot of support from multiple people.

Firstly, I would like to thank my primary supervisor, dr. ir. Hugo Jonker, for allowing me to work on this topic. The expertise and endless enthusiasm you brought to the table helped me immensely. You also gave me the freedom to steer my own course to some extent, which I appreciate a lot. Working with you was a true pleasure and made the experience not just interesting, but also fun.

Secondly, I would like to thank Benjamin Krumnow for keeping me on my toes. Your critical questions and comments definitely resulted in an overall improvement of my work.

Thirdly, I would like to thank my manager, Irving Mereau. Your support and feedback throughout my career and during this research have been invaluable. Because of you I managed to find a balance between my day job and my studies and for that I will be forever grateful.

I would also like to thank all my other colleagues and friends who supported me during this process. I would like to thank Toon Van Rompaey in particular, for being kind enough to provide review comments on this document.

I would also like to specifically thank my dear friend Matthew Keay. Your messages with relevant interesting links and articles, the many talks and discussions we had on the research topic, your feedback on my thesis drafts and your passion for everything IT security related, were truly one of the highlights of this entire process for me. I cannot thank you enough for your kindness and friendship.

Next, I want to thank my family and fiancée Zofia Jelenkiewicz for their love and support, even during the times when I could not spend as much time with them as I would have liked to.

Last but not least, I dedicate this work to my grandfather Georges Rottie, who sadly passed away unexpectedly during the preparation phase of this research. You were one of the few people I met with whom I could truly agree to disagree, the advice you gave me throughout my life will stay with me forever.

Contents

Chapter 1

Introduction

Website cloaking is a technique that enables websites to deliver different content to different clients, with the goal of hiding particular content from certain clients. Website cloaking is based on client detection, which is achieved via browser fingerprinting. In an attempt to hide their malicious web pages from detection, cyber criminals use cloaking. They use vulnerability detection to only target clients that seem vulnerable. On top of that, they also provide benign content in case they suspect someone or something is trying to detect them.

On the other hand, security analysts use security web crawlers, automated tools that crawl web pages and analyze them, for example to find malicious web pages. One example of such tools are honeyclients, also known as client honeypot web bots. Honeyclients are browser clients that are purposefully left vulnerable or that emulate vulnerable browsers. They are the client equivalent of a so-called server honeypot [QH10; QZ11], a server that is left vulnerable on purpose to lure in attackers, thus distracting and detecting them. The goal of a honeyclient is to detect webpages delivering malicious code. They are a potential counter to cloaking. While there is prior research into bot detection and browser fingerprinting [JKV19], it is currently not clear to what extent security web crawlers are distinguishable from regular clients, and thus whether cybercriminals can avoid sending malware to such clients by using generic cloaking techniques. It is also not clear to what extent cyber security professionals and their organisations are aware of and prepared for web based attacks using cloaking, or how their awareness and preparedness could be improved.

In this work, we investigate to what extent security web crawlers can be detected by browser fingerprinting techniques, and provide suggestions for how to improve them to better hide from those techniques. We survey security analysts and analyse a set of threat intelligence sharing communities, to gauge awareness of cloaking as an available detection evasion method for cybercriminals. Finally, we investigate one final technique, the use of Cache-Control: no-store, which an attacker might be able to use to thwart forensic analysis.

Contributions

- We present a review of prior research into the efficacy of a low interaction honeyclient in bypassing targeting techniques.
- We present a new, expanded and actualized, set of targeting techniques to test the efficacy of security web crawlers.
- We develop a new cloaking demonstration site implementing those techniques.
- We determine the ability of four publicly available security web crawlers to bypass the aforementioned targeting techniques using the created demonstration site.
- We propose potential improvements of security web crawlers to make them more effective at bypassing cloaking techniques.
- We present what we believe to be the first survey of cyber security professionals to gauge awareness and preparedness of cyber security professionals and their organisations to deal with cloaking web attacks.
- We examine tooling support to share cloaking related threat intelligence, in an open source threat intelligence sharing tool with a large user base, called MISP.
- We show that rudimentary awareness with regards to web cloaking attacks exists in some MISP threat intelligence sharing communities.
- We show that there is a significant difference in browser forensic artifacts for Chrome, Firefox and IE11 if malicious content was delivered with the cache-control: no-store header set.

Chapter 2

Problem statement and research questions

2.1 Problem statement

Malicious content delivery via the web remains a prevalent threat for individuals and organisations. Some aspects, such as the effectiveness of security web crawlers (for example honey clients) in dealing with generic cloaking techniques, and the awareness and preparedness of organisations to deal with such attacks, have not been discussed in prior research.

2.2 Research questions

- **RQ1** How effective are commonly used security web crawlers, one of the tools used to counter malicious web content delivery, at bypassing targeting techniques commonly used in cloaking and how could their efficacy in bypassing these techniques be improved?
  - **RQ1-1** How effective are publicly available security web crawlers, one of the tools used to counter malicious web content delivery, at bypassing cloaking techniques?
  - **RQ1-2** How could security web crawlers' efficacy at bypassing cloaking techniques be improved?
- **RQ2** How aware and prepared are cyber security professionals and their organisations when it comes to cloaking attacks, and how can their awareness and preparedness be improved?
  - **RQ2-1** How aware are cyber security professionals and their organisations of cloaking techniques?

  - **RQ2-2** How prepared are organisations for dealing with malicious web content delivery attacks using cloaking techniques?
  - **RQ2-3** What additional steps can organisations take to improve awareness and preparedness to deal with malicious web content delivery attacks using cloaking techniques?

- **RQ3** What other techniques can attackers use to thwart analysis of malicious web content delivery attacks?

Part I

Browser fingerprinting overview

Chapter 3

Browser fingerprinting: collection, uses and prevention

Browser fingerprinting refers to the process of collecting information through a web browser and web server, to build a *fingerprint* of a device or user, as well as the process of using a previously determined browser fingerprint to re-identify a user or device. Such a *browser fingerprint* is a set of attributes, attribute values, and behaviours of a browser and its underlying system, collectable via the browser, which together uniquely distinguish this browser from any other.

Browser fingerprinting techniques form the basis of cloaking. The cloaking logic uses the client's browser fingerprint to determine which content should be delivered. We focus on browser fingerprinting rather than the default identification technique on the web, cookies, because it has capabilities that can not be achieved with cookies, such as determining the browser family and version. A cookie can be part of a browser fingerprint however.

We intentionally limit our concept of browser fingerprint to the application layer of the OSI model. This excludes some attributes that could identify a browser by other means, such as network packet information captured on the wire [WG13].

3.1 Fingerprinting process

In this section we explain the main fingerprinting process, which entities can be fingerprinted and the difference between active and passive fingerprinting.

3.1.1 Fingerprinting and re-identification

The core fingerprinting process boils down to two main steps. In the first step, the fingerprint of a particular client is collected or determined. The second step consists of matching the fingerprint of a client, against a list of known fingerprints, to re-identify the client on each visit. The end goal is to vary the content that is sent to the client based on the identity of the user.

We can draw an analogy with the physical world. Imagine you own a coffee shop. You might want to give different service to people you know frequent your shop. ”Do you want to have the usual?” would not be something you say to a new customer for example. It is your knowledge about this particular client that allows you to vary your service.

There are different ways to collect or determine fingerprints to use for matching later. Some examples:

- Collect fingerprints of clients whenever a page you host is visited and store the information.
- Use research to determine what you expect certain attributes of the particular client to be. This could even be information gained from pictures such as one showing a politician using a fairly uncommon mobile phone with a particular browser.

3.1.2 Which entities are fingerprinted?

Browser fingerprinting techniques can be used to fingerprint different types of entities. We introduce the following concept:

Web identity The identity of any entity that accesses the web, either directly or indirectly.

We can distinguish between the following types of web identities:

- **User** A human user
- **Browser** A particular browser instance running on a specific system
- **Device** Any system, such as a smartphone, tablet or PC

We note that in some cases, a group of web identities is targeted. This could for example be web identities that belong to a certain organisation, or all smartphones running a particular version of the Android operating system. For an attacker to be able to do this, there needs to be some commonality between the fingerprints of the web identities.

3.1.3 Active vs passive fingerprinting

As Torres, Jonker and Mauw mentioned, a distinction can be made between active and passive fingerprinting [TJM15].

Active fingerprinting – probing Active fingerprinting or probing uses client side scripts (JavaScript, Flash, Java) to gather browser attributes.

Passive fingerprinting Passive fingerprinting uses attributes that are sent by the client in scope of its original request to the server. This can include attributes such as the request headers.

3.2 Fingerprintable attributes

3.2.1 categorisation

A shortcoming was observed in existing literature by Rik Dolfing and his thesis supervisor Hugo Jonker [Dol19]. It was noted that the existing categorisations of browser fingerprinting techniques such as the ones mentioned by Nikiforakis et al. [Nik+13] and Upathilake, Li and Matrawy [ULM15] are insufficient and need to be revised. For example, Nikiforakis et al. [Nik+13] categorised both Font Detection (detected fonts could be considered a better term) and User-agent under Operating System & Applications. In our opinion, there are very clear differences between the two, specifically if a side channel is used to probe for fonts. A major difference is that font probing checks behaviour of a browser, while the User-agent string is just a property. Discussions with peers and a review of categorisations in other research areas mentioning identification, led to an initial refinement of a new categorisation.

We believe that having a clear categorisation scheme or taxonomy can be helpful for multiple reasons. Firstly, categorizing things helps humans make sense of them [MR03] and can be an essential part of learning [Har17]. Categorisations also play a significant role in human communication [MR03]. We believe categorising fingerprinting techniques makes it easier to reason about groups of techniques, for example when determining measures against those techniques. Lastly, we hope that thinking about browser fingerprinting techniques in a different way sparks new ideas for potential techniques or techniques that might already be used but have not been described yet. Below we introduce the taxonomy for fingerprinting techniques proposed by Dolfing and Jonker [Dol19]. This new taxonomy includes techniques based on behaviour of the web identity.

The categorisation

System properties All properties of the underlying system on which the browser is running. These properties are independent of the properties of the user or browser. Examples of system properties are the operating system and the installed model of graphics card. Properties that depend on the user, such as the list of installed custom fonts, are excluded and considered part of the user properties.

System behaviour All behaviour that is specific to the underlying system on which the browser is running. An example of this is the behaviour of the system under certain types of load, for example using a benchmarking tool. It is important to note that many browser fingerprinting techniques tracking system behaviour, are used to determine system properties. One such case is determining whether the system's CPU supports the extended instructions of AES-NI [Sai+16]. These instructions provide hardware acceleration for encryption and decryption using the AES cryptographic algorithm. The visible behaviour is a difference in processing time when enabling the feature during benchmarking. The actual underlying property that is determined is whether AES-NI extensions are part of the CPU instruction set.

Browser properties All properties of the browser that is used to connect to the server. An example is the User-Agent. Properties that depend on the user, such as the list of installed extensions, are excluded and considered part of the user properties.

Browser behaviour All behaviour that is specific to a particular browser. Examples are the browser fingerprint and the ability to pass certain standardised JavaScript engine tests.

User properties All properties that are directly influenced by the user of the client that is connecting to the server. This includes installed plugins, extensions, set cookies and installed custom fonts. While it might be possible to use browser fingerprinting techniques to determine other personal user properties such as sex, these properties are usually not used in a fingerprint.

User behaviour All behaviour of the human user (if there is any) that is connecting to the server. Examples include mouse movement and typing speed.

Examples of categorised fingerprintable attributes can be found in Table 3.1.

	Properties	Behaviour
System	Screen width	Ability to handle multiprocessor tasks
System+Browser	User-Agent	The way fonts are displayed, Canvas fingerprint
Browser	Browser family and version	JavaScript engine conformance test performance
User	Cookies, Plugins, Installed custom fonts	Mouse movements, typing speed

Table 3.1: Examples of categorised fingerprintable attributes

Discussion

The current taxonomy is a proposal, which can be tested in future research. The completeness of the proposed taxonomy should be verified. Every known fingerprintable attribute should fall under one of the proposed categories, otherwise the taxonomy is not complete.

Aside from that, we realized that the proposed taxonomy leads to cases where a fingerprintable attribute ends up in a combined category, meaning the value is determined by multiple layers, for example system and browser in the case of a canvas fingerprint. This example feels rather intuitive. There are others that require some more thought. The list of all installed default fonts can be seen as a system property. Which custom fonts are installed is entirely determined by the user however. While a User-Agent often contains system information, it is entirely up to the browser to set it and not controlled by the system at all.

Furthermore, we want to point out that in some cases behaviour is used to determine a property. As a result, we feel it is very important to mention what the goal is of the fingerprinting technique and what is actually being fingerprinted. This can result in one technique both fingerprinting behaviour and a property.

Finally, one of the major reasons for introducing this new taxonomy differentiating between behaviour and properties, was the realization that a property fingerprint can be captured fully, while this is practically impossible for behaviour. For example, it is possible to capture all the elements in the DOM [Dol19]. The same can not be said for getting all installed custom fonts using font probing. Font probing checks whether a font in a predetermined list is installed. If the user installed a custom font the fingerprinter doesn't know, it is not checked for. The list of custom fonts that can be installed can be practically infinite. This leads to specific problems. It is practically impossible for a fingerprinter to fingerprint all behaviour and vice versa it is practically impossible for browsers or users to protect against all forms of behavioural fingerprinting.

3.2.2 Identifiable elements and identifiable behaviour

An *identifiable element* is an attribute or set of attributes that can be used to re-identify a web identity at a later time. An analogy can be made to identifying suspects in a bank robbery investigation. Both the color of a person's eyes (an attribute) as well as the fact the person had a goth look based on dark clothing and a choker around neck (combined attributes) can be seen as identifiable elements.

Below we provide a non exhaustive list of identifiable elements that have been described in prior research. We chose the below examples because they either get mentioned in related work frequently, or because we deemed them to be fairly creative and surprising options. As a result, the below list should give the reader a sense of which types of attributes can be fingerprinted. A more extensive set of examples can be found at `https://amiunique.org/fp`.

User-Agent The User-Agent depends on the browser vendor and version, by design[1]. An example of a User-Agent string is: "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36".

[1] `https://tools.ietf.org/html/rfc1945#page-46`

HTML5 Canvas The HTML5 canvas element[2] can be used to programmatically draw images on a web page. The rendering of this image is dependent on the hardware of the system. As a result, and because the pixel information can be gathered via an API, it is possible to use a canvas element to create an identifiable element. Mowery and Shacham were the first people to describe this technique [MS12] in 2012.

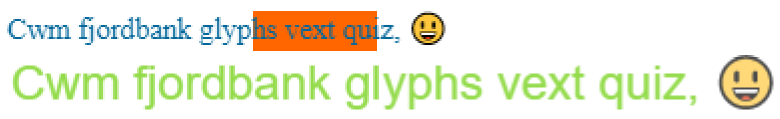

Cwm fjordbank glyphs vext quiz,
Cwm fjordbank glyphs vext quiz,

Figure 3.1: HTML5 canvas image

JavaScript Engine Compliance This technique was introduced by Mulazzani et al [Mul+13] in 2013. They showed that the results of a browser in JavaScript Engine Conformance tests can be used to determine which browser and browser version is connecting to the server. JavaScript code to run a set of tests is sent to the client and is then run client-side. These tests check for compliance to the ECMAScript standard[3]. The results are sent back to the server. Assuming the only browsers connecting to our server are the ones in the table below, we can uniquely identify the browser by only running 2 tests, for example tests 2 and 3. If both tests passed, the client is using browser 2, version Z. In the other cases the failing test immediately leads to one of the other two options. This process can be optimized by using a decision tree.

	JavaScript engine conformance tests			
Browser	*Test 1*	*Test 2*	*Test 3*	*Test 4*
Browser 1, version X	√	Fail	√	√
Browser 1, version Y	√	√	Fail	√
Browser 2, version Z	√	√	√	Fail

Table 3.2: Illustration of JavaScript Conformance tests differences

Enabled plugins The list of enabled plugins could be enumerated. As users can install different plugins, this list can be distinct for a particular browser. The list could be accessed using the navigator.plugins object for example[4]. Due to privacy concerns, this function has become deprecated.

System fonts System fonts can be enumerated with Flash, Java and JavaScript [Eck10].

[2]`https://www.w3.org/TR/2011/WD-html5-20110525/the-canvas-element.html#the-canvas-element`

[3]`https://www.ecma-international.org/ecma-262/11.0/index.html#title`

[4]`https://developer.mozilla.org/en-US/docs/Web/API/NavigatorPlugins/plugins`

Font metrics In 2015, Fifield and Egelman described a novel method for fingerprinting based on font metrics [FE15]. It appears some browsers render the same character differently. This leads to variations in the size of the rendered character. By setting a large font-size, the effect of these variations on the size increases as well, making them more easy to detect. This technique does not use the HTML5 canvas element and can work even when a browser has protection against canvas fingerprinting.

Battery Status In 2015, Olejnik et al. showed that certain battery status details can be used for fingerprinting [Ole+15]. These details are exposed via the battery status API. The researchers submitted a bug report to Firefox for one of the issues with the implementation of this API, which made it possible to determine the battery capacity of the connecting device with a relatively high accuracy. Mozilla implemented a fix as a result. Regardless, it is still possible to use the battery status API for short term fingerprinting purposes.

CPU features In 2016, Saito et al. [Sai+16] showed that it is possible to estimate whether the CPU of the system on which the client application is installed has certain hardware features. This allows detection of AES-NI and Intel Turbo Boost Technology support. These properties can then be used as an extra identifiable element.

Audio device features In 2014, Zhou et al. [Zho+14] showed that it is possible to use high frequency audio output of speakers as a feature, by playing a crafted audio fragment and recording it with an Android phone's microphone.

3.3 Characteristics of a fingerprint

Building on the knowledge about the fingerprinting process and fingerprintable attributes, we can now discuss some characteristics of a fingerprint. A browser fingerprint is in essence a set of current values for certain attributes. The two main characteristics of a fingerprint are its uniqueness and stability, these concepts are explained in the following subsections. The effectiveness and stability of the fingerprints used relies on the selected attributes and how the values for them are determined. An example of potential differences between fingerprints can be seen in the table below.

	Attributes used in fingerprint			
Site	*User-Agent*	*Font list*	*Plugin list*	*Canvas*
Site 1	√	√	√	
Site 2	√	√	√	√[1]
Site 3	√	√	√	√[1]

[1] site 2 and 3 use different canvas fingerprinting techniques

Table 3.3: Illustration of different fingerprint implementations – selected attributes

3.3.1 Uniqueness

To be able to use a specific fingerprint to identify a particular web identity, the fingerprint of that web identity has to be unique. The less common the values of the web identities' fingerprint are, the more unique the overall fingerprint is. The amount of information expressed by a certain identifiable attribute is commonly measured in entropy.

For a more mathematical analysis of the uniqueness of a fingerprint in a particular set of collected fingerprints, and a discussion of how this might be extrapolated to the global set of fingerprints, we refer to the paper by Eckersley [Eck10].

3.3.2 Stability

As the goal of browser fingerprinting is to match a collected fingerprint against a selected fingerprint, a fingerprint should remain relatively stable for this to be possible. Some web identities' fingerprint are less stable than others. Examples of reasons for this are more frequent updating of the browser, changing network environments and extensions that are designed to break stability of the fingerprint.

3.4 Use cases

In this section we discuss different use cases for the browser fingerprinting. It is important to note that the web server is always the fingerprinter and thus has control over which use cases are implemented. While some of these use cases are clearly meant to improve the service for the users, others are there solely to protect the web server or to provide benefits to the owner of the service. In multiple scenarios, there is a trade-off between a positive effect for one of the two parties and a negative effect on the other.

Risk-based authentication Passwords are an imperfect solution for authentication [Bon+15]. In an attempt to decrease the chances of someone else logging in to your account without authorization, websites have started implementing multi-factor authentication mechanisms. Such mechanisms ensure that at least two authentication factors have to be provided before you are successfully authenticated. A site might request you to provide a password and a code sent to your phone by SMS for example. While adding multiple authentication factors can increase security, it also has a negative impact on usability.

Risk-based authentication is the modification of the authentication mechanism based on the perceived risk of the new authentication. In the context of web applications, attributes of the browser can be used to estimate the risk of the new authentication and modify the authentication mechanism accordingly. In practice, this might mean you have to provide a code sent to your phone by SMS the first time you log in from a device you hadn't used to connect to that site before, but not when connecting from a trusted browser. Technically, browser fingerprinting techniques can also be used to counter session hijacking attacks. The goal of risk-based authentication is to increase security while limiting the impact on usability of the service.

Alerting of potentially unauthorized login Instead of denying access to a service, or requesting an extra authentication factor based on the risk of the authentication request, a web application can also notify the legitimate user of a potentially unauthorized login[5], by sending an alert via a predetermined communication channel such as email. While this does not completely block the login, it allows the legitimate user to take action in case the account is indeed compromised. This alerting can be used in situations where the perceived risk of the login being unauthorized, is relatively low.

Targeted advertising Advertisers pay money to advertisement platforms or websites to host their ads. To increase the value advertisers get for their money, many advertisement platforms allow advertisers to choose who their ads are showed to by setting certain parameters. Possible parameters include IP, browser, operating system, gender and social class, amongst others[6]. While receiving more relevant ads might be convenient for users as well, in many cases the means that are used to achieve this targeting raise major privacy concerns.

Collection of personal information to sell to third parties Unbeknownst to many users, online service providers collect and sell personal information [FH18]. These providers often defend this practice by saying the data is anonymized[7]. Unfortunately, anonymized data can often be de-anonymized by combining it with other data [HL21].

Impersonation detection Some people impersonate someone else online, enabling them to cause major issues for other users. An example of this is male sexual predators pretending to be younger or a different gender. Browser fingerprinting techniques can potentially be used to detect impersonation. Van Balen, Wang and Ball [BBW17] showed that it is possible to determine the gender of a user by tracking certain mouse movements, which might be possible via a browser as well. Pusara and Brodley showed that user re-authentication via mouse movements in browsers is possible [PB04], making the proposed use case plausible as well.

Bot detection Web bots are automated clients running tasks on the web. Usage of this technology can help improve the quality of the service. An example of this is automated web testing, which can increase the effectiveness and efficiency of testing. Unfortunately web bots are also used for less honorable practices such as ad-fraud, intellectual property theft (via web crawling) and credential stuffing. Browser fingerprinting techniques can be used to detect and block incoming connections of web bots [JKV19]. Anti-scraping is a specific form of bot detection, aimed at preventing scraping of web pages.

[5]`https://www.zdnet.com/article/google-now-gives-you-android-notifications-when-new-devices-log-into-your-accounts/`

[6]`https://www.advertising.dpgmedia.be/en/advertise/digital/data-targeting`

[7]`https://support.unroll.me/hc/en-us/articles/360004026232-Unroll-Me-Privacy-Policy`

Region-based content locking Due to regulatory or licensing reasons, a website might choose not to allow users from certain regions to access particular content. This can be achieved with the use of browser fingerprinting. An example of such a scenario are video streaming platforms that only allow visitors from certain countries to watch particular videos[8].

Web tracking Since browser fingerprinting allows re-identification of a client, it can be used [Eck10] to track that client's actions on the web on each site that includes the fingerprinters' code.

Vulnerability detection (both for offensive and defensive purposes) Browser fingerprinting techniques can be used to detect vulnerabilities of the browser [SLG19][Lap+20] and the system the browser is running on. This can allow a malicious actor to craft a specific attack against the vulnerable system either directly or at a later time, using the acquired knowledge. It can also be used by defenders to detect weaknesses in systems to patch.

Cloaking In an attempt to hide their malicious pages from detection, cyber criminals use browser context information to vary the content they send to visiting clients [Zha+21]. In case they suspect someone or something is trying to detect them, they provide benign content. Aside from that, they also use vulnerability detection to only target vulnerable clients.

Cookie regeneration Browser fingerprinting techniques can be used to regenerate cookies after deletion [Lap+20]. As a result, even if the user thinks she is no longer tracked because she deleted tracking cookies, this is not necessarily the case.

3.5 Browser fingerprinting adoption

In 2013, Nikiforakis et al. [Nik+13] showed that at least 0.4% of the Alexa top 10,000 sites were using fingerprinting scripts from at least one of three commercial companies: BlueCava, Iovation and ThreatMetrix. Since they only crawled 20 pages for each site and did not access any as registered users, this result is a lower bound.

In 2014, Acar et al. [Aca+14] found that at least 5.5% of the Alexa top 100,000 included a canvas fingerprinting script on their home pages. 20 different fingerprinting provider domains were identified. 9 of those were inhouse, first party domains. 11 were third party providers.

In 2016, Englehardt and Narayanan found that 7% of the Alexa top 10,000 sites used a canvas fingerprinting script[EN16] on their homepage. They also discovered that usage

[8]`https://help.crunchyroll.com/hc/en-us/articles/204878816-Which-shows-are-available-in-my-country-`

of canvas fingerprinting is more prevalent amongst the top sites compared to the lower ranked sites.

In 2019, Jonker et al. [JKV19] showed that 12.8% of the Alexa Top 1 Million sites showed indications of bot detection using fingerprinting techniques.

3.6 Countermeasures

3.6.1 Generic types of defenses

A distinction can be made between several types of defenses, which we present below. Each of these has their own scope and merits, a comprehensive defensive strategy should use a combination of the below types.

Prevention Prevent a compromise or attack altogether. A firewall rule could block a connection to a specific system for example.

Detection Detect when an attack happened. An alert could be raised when a system attempts to initiate a connection it shouldn't, for example.

Mitigation A mitigation doesn't prevent an attack from succeeding, but reduces the impact when it does. An example is the limit most people have on the amount of money they can send per day using their bank details. Even if someone gets hold of a user's debit or credit card and pin code, they can only use it to send a limited amount of money.

Physical Physical protection, such as the door and lock of a house to prevent people from entering.

Logical, technical A technical protection such as a firewall rule.

Administrative Administrative protection such as a policy or law, such as as policy stating corporate devices can not be utilized for personal use.

3.6.2 High level strategies against browser fingerprinting

Limit exposure of browser details

As mentioned earlier, one of the main reasons browser fingerprinting is possible, is the fact that browsers simply expose a lot of information, for example through APIs. One strategy is to simply limit the information available through the APIs. Depending on the extent to which information is limited, this can prevent fingerprinting using the particular attribute or reduce the effectiveness of the fingerprinting technique.

To illustrate this strategy, we refer back to the battery API issue mentioned earlier. On Linux, the precision of the returned battery status value was a 64 bit double-precision floating-point number [Ole+15], which is unnecessarily precise. Rounding the value to two significant digits makes it much harder to abuse.

Increase uniformity → break uniqueness

If you want to hide in the crowd, you have to make sure you don't stand out in the crowd like the yellow chair in Figure 3.2. This strategy works by adopting a browser configuration that is exactly the same or very similar to that of other users. In effect, this renders it impossible for a fingerprinter to distinguish your browser from others. In the example picture below, the yellow chair is very easy to distinguish from the others. The others all look alike however, and it would be much harder to identify one particular green one.

An example use of this strategy could be an organisation only allowing internet access with a browser from a virtual environment hosted on shared hardware. If the configuration of the browser cannot be changed by the end users, all system and browser level attributes would be the same. As a result it would be more difficult to use browser fingerprinting to identify a particular user within the organisation.

Figure 3.2: Unique stadium chair, yellow, is easy to pinpoint

Break linkability

A second possible strategy is to make sure your fingerprint changes often over time, in such a way that it is impossible to link your new fingerprint to the older one. An illustrative example with a real life criminal is the convicted serial killer Theodore Robert Bundy, more commonly known as Ted Bundy. Ted changed his appearance over time in an attempt to make it harder for police to track him down, as can be seen in the different mugshots in Figure 3.3. The same strategy to counter tracking can be applied to web browsers. Increasing randomness of a browser fingerprint is one way of breaking linkability.

A specific example of this strategy is the 'farbling' or randomization the Brave browser does for the canvas image[9].

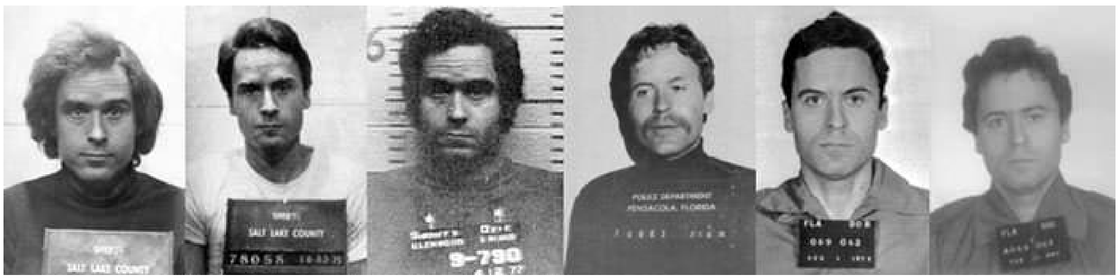


Figure 3.3: Breaking linkability example – Ted Bundy mugshots

3.6.3 Particular tool implementations and ideas

In this section we present a non exhaustive list of tools that have been designed to prevent or mitigate browser fingerprinting attempts.

Script-blocking tools These tools completely block certain scripts from running. Examples include No-Script, Ghostery and Privacy badgers. They all use the *limit exposure of browser details* strategy.

Tor browser The Tor browser[10] is a browser designed to access an uncensored web privately. It has built-in protections against browser fingerprinting[11]. Tor browser has chosen to mainly use the *increasing uniformity* and *limiting exposure* strategies.

Firegloves A Firefox plugin to "impede fingerprinting-based tracking while maintaining browsing experience"[12] [Aca+13]. It uses the *breaking linkability* and *limiting exposure* strategies. This project is no longer under active development.

FP-Block FP-Block is a Firefox plugin that uses the *breaking linkability* strategy [TJM15]. It blocks cross-domain tracking by creating a new fingerprint for each domain.

[9]`https://brave.com/privacy-updates-4/`
[10]`https://www.torproject.org/about/history/`
[11]`https://2019.www.torproject.org/projects/torbrowser/design/#fingerprinting-linkability`
[12]`https://fingerprint.pet-portal.eu/?menu=6`

It still allows tracking within one domain. As a result, the impact on usability is minimized. The creators paid extra attention to making sure the spoofed fingerprint does not have contradictory attributes as well, since these can lead to more uniqueness.

FPGuard Is a browser extension and instrumented Chromium browser [FZW15]. It uses the *breaking linkability* and *limiting exposure* strategies. It also includes fingerprinting detecting logic, that can be used to keep a blacklist of fingerprinters.

Blink Blink is a prototype solution that aims to break linkability by automatically reconfiguring browser environments, including virtualized operating systems [LRB15]. It does allow users to configure some elements that should be kept static for each session, to increase usability. Blink uses the *breaking linkability* strategy.

PriVaricator PriVaricator is an enhancement of the Chromium private browsing mode [NJL15]. It uses the *breaking linkability* strategy. It adds randomness to the return values of some browser attributes.

FPRandom FPRandom is a modified version of Firefox [LBM17]. It aims to apply the *breaking linkability* strategy by making the implementation of some browser functionality less deterministic, while limiting the impact on the user experience. It mainly does this by modifying some browser functions that do not have to be deterministic according to the ECMAScript specification. This last point is a key difference with PriVaricator, since PriVaricator also adds randomness to functions that should be deterministic.

Browser isolation systems Browser isolation systems isolate the browser environment from the system the end user is browsing on [Cro17]. The browser software is run in a local sandboxed, virtualized environment or on a completely different system. We believe it is plausible that these systems could be used to protect against fingerprinting by making it easier to increase uniformity of browsers, if customization options of the browsers are limited. This would be at the cost of usability, but we believe some organisations would and end users would prefer this option. A potential risk to selecting one browser profile for all users of a particular organisation could be that it makes the organisation's web identity easy to fingerprint.

Brave Browser A Browser implementation based on the Chromium browser[13]. It includes protections against fingerprinting in the form of randomizing certain attributes[14]. It uses the *breaking linkability* strategy.

[13]`https://brave.com/`

[14]`https://brave.com/privacy-updates-4/`

A collaborative strategy Gómez-Boix et al. pitched the idea of a collaborative strategy for mitigating tracking through browser fingerprinting [Góm+19]. The browser fingerprints of participants would be collected. Then a clustering algorithm is applied to divide the participants in a predefined number of clusters. Each member of a particular cluster then configures their browser the same way. The different clusters still allow for differentiation of the configuration. As a result the impact on the usability for each user can be limited, depending on the amount of clusters. This idea uses the *increasing uniformity* strategy.

Part II

Malicious content delivery via the web and honeyclients as a protection mechanism

Chapter 4

Malicious content delivery via the web

This section is intentionally kept relatively short, as its only goal is to give the reader the knowledge required to understand the uses of honeyclients and other similar crawlers in the context of malicious content delivery via the web. For a list of commonly known countermeasures against malicious web content, we refer to Appendix A.

4.1 High level targeting paths

4.1.1 Waterhole

The attacker compromises a site that is frequently or mostly visited by the target audience. An example of this would be an attacker compromising a forum for Pokemon card collectors, if the goal is to find collectors with high value cards. The name is derived from waterholes animals tend to visit. An analogy can be made to a crocodile waiting in the water for animals to become thirsty and coming to drink.

4.1.2 Malvertisement

Advertisers pay money to advertisement platforms or websites to host their ads. To increase the value advertisers get for their money, many advertisement platforms allow advertisers to choose who their ads are showed to by setting certain parameters. Possible parameters include IP, browser, operating system, gender and social class, amongst others[1]. If an attacker manages to get their malicious content delivered by an advertisement network, they can use these specific targeting parameters as well.

[1] `https://www.advertising.dpgmedia.be/en/advertise/digital/data-targeting`

4.2 Cloaking delivery flows

As mentioned earlier, website cloaking is a technique that enables websites to deliver different content to different clients, with the goal of hiding particular content from certain clients. Website cloaking is based on client detection, which is achieved via browser fingerprinting. This browser fingerprinting can be done on the server, using server technologies, or on the client, using JavaScript for example. Below we describe four different strategies that can be used to deliver web content. Note that all of the below strategies can be combined with a phased fingerprinting approach. Using such an approach, the different fingerprint checks are done in separate phases. The first server or script might check one fingerprintable attribute and then redirect to another page or server, or deliver another fingerprinting script. This could then check another fingerprintable attribute and do the same thing. The end result is that part of the cloaking stack is cloaked as well, making potential investigations harder.

4.2.1 No cloaking

With no cloaking, the delivered content does not depend on the browser fingerprint. As a result the same content is delivered to all visitors. The client submits a request for a page, the server sends the content without applying any filtering.

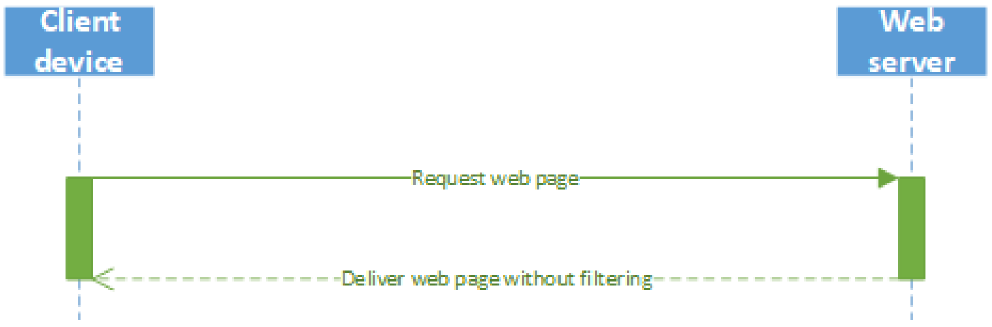


Figure 4.1: No cloaking

Server-side cloaking In server-side cloaking, the cloaking logic is entirely on the server. The client submits a request to the server, the server uses some of the details of the request to determine whether it should send specific content or not. Examples of this type of cloaking are the blocking of certain IP addresses or User-Agent strings from seeing the cloaked content.

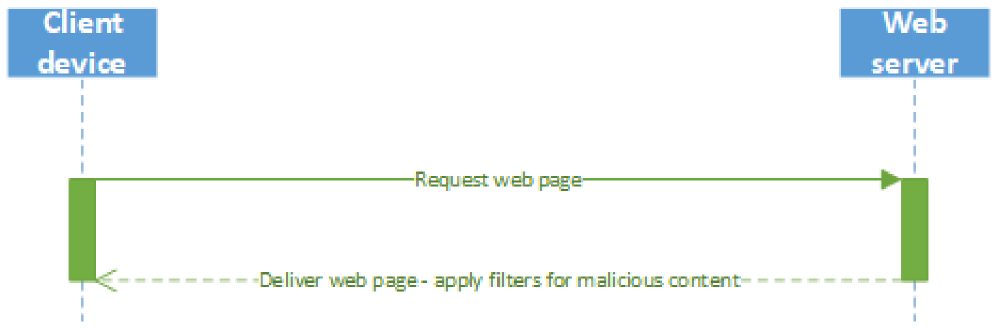


Figure 4.2: Server-side cloaking

4.2.2 Client-side cloaking

In client-side cloaking, the cloaking logic is sent to the client and executed inside the browser client, for example by using JavaScript. An example of this is a script that checks if the user visited other pages before the requested one in this browser session (tab) or not, before requesting the cloaked content.

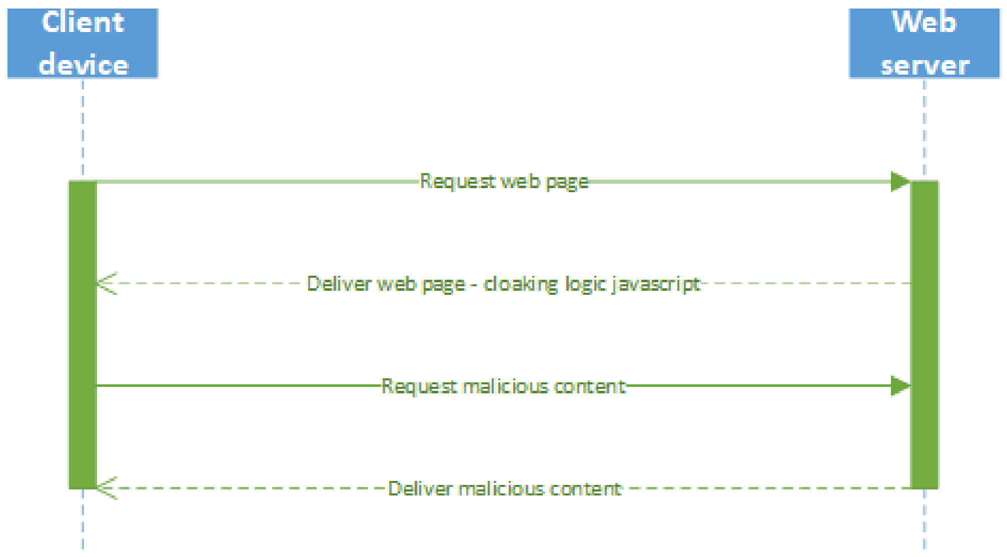

Client device
Web server
Request web page
Deliver web page - cloaking logic javascript
Request malicious content
Deliver malicious content

Figure 4.3: Client side cloaking

4.2.3 Client-side cloaking with server side validation of results

In this case, client-side technologies such as JavaScript are used to gather certain browser attributes, which are then sent back to the server. The server then uses that data to decide whether it should sent the cloaked content or not.

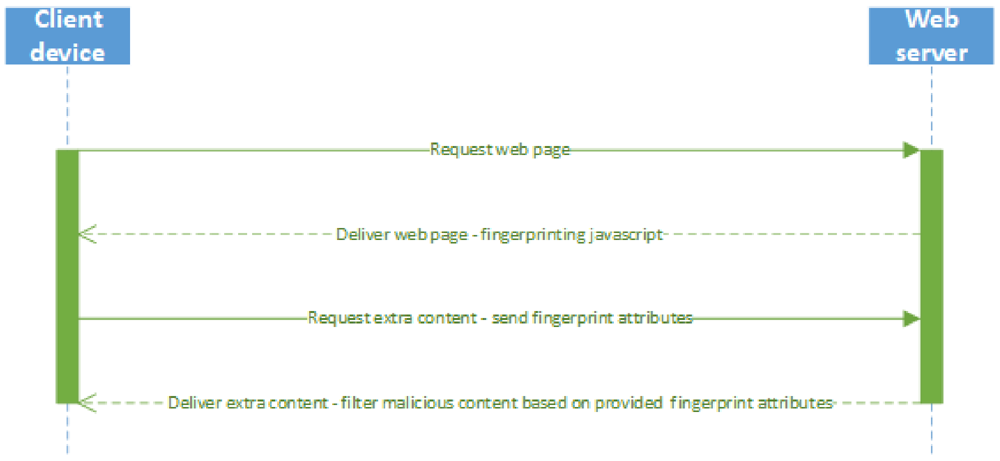

Client device
Web server
Request web page
Deliver web page - fingerprinting javascript
Request extra content - send fingerprint attributes
Deliver extra content - filter malicious content based on provided fingerprint attributes

Figure 4.4: Client side cloaking – fingerprint information sent back to server

4.3 Generic targeting and bot detection techniques used for cloaking

Below we provide a non exhaustive list of potential targeting techniques for cloaking during malicious content delivery. The core of this list was built by combining techniques mentioned in prior work, in particular the papers of Brendan Dolan-gavitt and Yacin Nadji [DN10] and Zhang et al. [Zha+21]. A small set of other ideas was added based on our own knowledge of honeyclients and browser fingerprinting. Finally, we also analyzed the MDN Web API documentation[2], to find other potentially interesting APIs to use for cloaking purposes. We found one, Window.opener, of particular interest. For the specifics, we refer to 4.3. We chose to focus on techniques that are commonly mentioned, cover both properties and behavioural attributes, we also chose examples of newer and deprecated techniques. We excluded potential techniques with particularly complex implementations.

Please note that we also performed a specific review of the techniques mentioned by Brendan Dolan-gavitt and Yacin Nadji [DN10] as part of this research. We believe this is valuable due to the fact that the mentioned paper dates back to ten years ago, meaning some of the techniques might not be relevant anymore. The results can be found in Subsection 6.4.1.

Geolocation – IP based IP addresses can be linked to specific geographic locations, like countries. In this check, the server validates that the request comes from an IP linked to a specific country or set of countries. This way, attacks can be targeted towards users likely originating in a specific set of countries, and analysis from crawlers or security analysts from other countries is thwarted.

[2]`https://developer.mozilla.org/en-US/docs/Web/API`

Referer The Refererer header indicates from which other url the current one was called. As an example, the referer header is set to *https://www.google.com/* when a user clicks on a google search result to navigate to it. In this check, the server validates the request includes the expected value for the referer header. Having this header set gives the server some indication that the user performed certain actions to get to the page.

User-Agent "*The User-Agent request header is a characteristic string that lets servers and network peers identify the application, operating system, vendor, and/or version of the requesting user agent.*" [Moze]. In this check, the server validates whether the User-Agent header is set to an expected value. This is used to target specific browser family versions for example.

Cookie The cookie headers are used to make a HTTP connection stateful. A cookie could be set to indicate a specific page was visited for example. In this check, the server validates that an expected cookie value is set. The malicious payload is only delivered in case the user executed the necessary actions to set the required cookie value.

System date An attacker could assume that real users (humans) would make sure their system time is relatively correct. This might not be the case for automated crawlers, which might restore to a virtual machine image in between consecutive runs as well, leading to an out of sync system time. In this check, the web application triggers the client to send its system time to the server. The server then verifies if that time is not significantly different from its own time. While we do not believe this test will be useful in a lot of cases, it can still work as designed at the time of writing.

Immediate browser history Browsers keep track of which pages were visited during a session, to enable going back and forward through those pages. The History web API gives access to the number of entries in the history [3]. In this check, the web application triggers the client to send the history length to the server. The server then verifies if that length is higher than a specified number. As a result, the malicious content is only delivered for connections from a session that went through the expected number of pages to get to the malicious web page.

Opener The Window.opener property contains a reference to the window that opened the current window[4]. In this check, the web application triggers the client to send the opener url to the server. The server then verifies if that value matches the expected opener. This way, the server validates the new window was opened from a specific url.

User-Agent consistency check The User-Agent header can be easily spoofed by changing the header. The value of the User-Agent is also accessible via the Navigator

[3]`https://developer.mozilla.org/en-US/docs/Web/API/History/length`

[4]`https://developer.mozilla.org/en-US/docs/Web/API/Window/opener`

Web API however[5]. If this value is not spoofed as well, it is fairly obvious the client is lying. In this check, the web application triggers the client to send the navigator.userAgent value to the server. The server then verifies if that value matches the value provided in the User-Agent request header. We note here that there are other ways to check if the User-Agent is being spoofed. Since the User-Agent indicates which browser family and version is used, the server could check for specific functionality or bugs only present in the specific browser version. If it is not present, the server could also assume the User-Agent is spoofed.

Mouse movement While the vast majority of human users probably performs mouse movements when visiting a web page, this might not be the case for web crawlers. One option for crawlers to counter this, is to trigger some random mouse movements after the initial page load. In this check, the web application only triggers the malicious payload if mouse movement is detected after a set timeframe. This would still block analysis by crawlers that only simulate mouse movements during a smaller timeframe.

Captcha "*A captcha is a program that can generate and grade tests that: (A) most humans can pass, but (B) current computer programs can't pass. Such a program can be used to differentiate humans from computers.*" [Ahn+03]. As per the definition, captchas are specifically designed to differentiate humans from computers. If a captcha is not complex enough, it can be bypassed however. In this check, the web application uses a captcha and only delivers the malicious payload to clients that pass the test.

Alert The Window.alert Web API[6] opens a window with a given text and waits until the user clicks ok. Some crawlers might not have the functionality to perform this action. In this check, the web application only delivers the malicious payload to clients that close the alert. This test does require user interaction and could arouse suspicion.

Confirm The Window.confirm Web API[7] opens a window with a given text and waits until the user clicks ok or cancel. Some crawlers might not have the functionality to perform this action or default to one of the two option. In this check, the web application only delivers the malicious payload to clients that click the expected option. This test does require user interaction and might arouse suspicion.

Notification The Notification.requestPermission Web API[8] prompts the user for permission to send notifications later on. It is fairly new with support only having been implemented in certain browsers such as Chrome, Edge and Firefox in 2015 and 2016.

[5]`https://developer.mozilla.org/en-US/docs/Web/API/Navigator/userAgent`
[6]`https://developer.mozilla.org/en-US/docs/Web/API/Window/alert`
[7]`https://developer.mozilla.org/en-US/docs/Web/API/Window/confirm`
[8]`https://developer.mozilla.org/en-US/docs/Web/API/Notification/requestPermission`

The permissions API also plays a role in this process[9]. In this check, the web application prompts the user for permission to send notifications and only delivers the malicious payload to clients that grant permission. This filters out potential crawlers that do not support Notification.requestPermission or default to rejecting the request.

Microphone and camera access The MediaDevices.getUserMedia Web API[10] prompts the user for access to the microphone and webcam, or one of those. Just like Notification.requestPermission, it is fairly new with support only having been implemented in certain browsers such as Chrome, Edge and Firefox in 2015 and 2016. The permissions API also plays a role in this process[11]. In this check, the web application prompts the user for permission to the webcam and microphone and only delivers the malicious payload to clients that grant permission. This filters out potential crawlers that do not support MediaDevices.getUserMedia or default to rejecting the request.

Presence of webcam (video input device) – without access granted The MediaDevices.enumerateDevices Web API requests the list of media input and output devices connected to the browser device. [12]. In some browsers, this returns a list of device info even if permission to access devices was not yet granted. In our control tests, both chromium and firefox reported the presence of a webcam when it was connected, and did not report a connected videoinput device when there was no webcam connected. In this check, the server instructs the browser to only load the malicious content if a video input device is detected. This test is still relevant at the time of writing, even though it does require a different technical implementation than the usage of Flash, which was mentioned by Brendan Dolan-Gavitt & Yacin Nadji [DN10].

Geo API location access The Geolocation.getCurrentPosition Web API[13] prompts the user for access to the location of the device. The permissions API also plays a role in this process[14]. In this check, the web application prompts the user for permission to the device location and only delivers the malicious payload to clients that grant permission. This filters out potential crawlers that do not support Geolocation.getCurrentPosition or default to rejecting the permission request.

The following techniques have been discussed in prior research, but we do not consider them to be very relevant anymore or believe them to only be useful in specific scenarios, for various reasons that are explained per technique.

[9]`https://developer.mozilla.org/en-US/docs/Web/API/Permissions_API`
[10]`https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia`
[11]`https://developer.mozilla.org/en-US/docs/Web/API/Permissions_API`
[12]`https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/enumerateDevices`
[13]`https://developer.mozilla.org/en-US/docs/Web/API/Geolocation/getCurrentPosition`
[14]`https://developer.mozilla.org/en-US/docs/Web/API/Permissions_API`

Window beforeunload "*The beforeunload event is fired when the window, the document and its resources are about to be unloaded. The document is still visible and the event is still cancelable at this point*" [Mozg]. In this check, the web application only delivers the malicious payload when this event is triggered and the client closes the related popup in a predetermined time. However, the Mozilla Developer Network (MDN) documentation indicates that the conditions for this event to trigger are different depending on the browser [Mozg], meaning you can not rely on it being triggered whenever the web page is closed. In some browsers, the commands that can be run when this event is triggered, are also limited. On top of that, triggering this event relies on the web page managing to make the user close the window in the first place. Because of the above limitations, we decided not to include this technique in this research. While technically possible to implement, we believe that whether or not this test is relevant is debatable.

Connection speed In this check, the web application triggers a client-side check of the connection speed and only delivers the payload if the client's connection speed falls in a certain range. The reasoning of this check is that organisations running security web crawlers like honeyclients might have a significantly higher connection speed than normal users [DN10]. We do not believe this is a reliable check, since even consumer grade subscriptions offer 1Gbps connection speeds in certain countries [Tel]. organisations might also throttle their speeds for single connections, to limit excessive bandwidth usage by each browser client. As a result, we decided not to implement this technique for this research. While technically possible to implement, we believe that whether or not this test is relevant is debatable.

Plugins The netscape Plugin Application Programming Interface (NPAPI), and the related Pepper Plugin API are used for the creation of browser plugins. There are existing JavaScript libraries such as PluginDetect [Ger] that detect installed plugins for certain browser versions. In this check, the web application determines if certain plugins are installed and only delivers the malicious payload in that case. The assumption is that human users would have certain plugins or a certain amount of plugins installed. However, all major browser vendors have removed support for NPAPI plugins in the last six years [Mica][Mozc][Chrb]. One exception is Flash, which is no longer enabled or supported in the latest versions of browsers released in 2021 or is planned to be completely removed later [Mozc][Chra]. Because of the above, we decided not to implement this technique for this research. We believe this test is not as relevant anymore for modern browsers.

Presence of speakers (audio output devices) – without access granted This check is similar to that of **presence of webcam**, but detects audio output devices. The control tests we performed to validate this technique led to inconsistent results however.

Presence of microphone (audio input devices) – without access granted This check is similar to that of **presence of webcam**, but detects audio input devices. The control tests we performed to validate this technique led to inconsistent results however.

Browser history In 2006, Jeremiah Grossman published about a CSS hack allowing a web application to detect if a client browser had visited a specified url in the past[Gro]. In this check, the web application verifies if the client has visited a given set of urls, and only delivers the malicious payload if that is the case. Browsers have since continued to include patches for such browser history leaks however [Mozb][Mozd]. While new leaks have been discovered over time [Smi+18][Moza], we do not believe this is a reliable technique any longer. We believe this test is not relevant anymore for modern browsers.

Chapter 5

Introduction to honeyclients

Honeyclients are browser clients that are purposefully left vulnerable or that emulate vulnerable browsers. They are the client equivalent of a so-called server honeypot [QH10; QZ11]. A server that is left vulnerable on purpose to lure in attackers, thus distracting and detecting them. The goal of a honeyclient is to detect webpages delivering malicious code and as such are a potential counter to cloaking. One distinct difference between the client honeypot and server honeypot techniques is that the use of a server honeypot is a passive approach. A honeypot owner waits until an attacker falls into their trap. When using a honeyclient however, the web is actively crawled. Usually some sort of pre-filtering of web pages is done, to reduce the amount of pages that have to be visited to those that are the most likely to be malicious. This reduces the amount of resources used, which is necessary since crawling the entire internet is not feasible for most research groups. Since there is an infinite amount of possible browser configurations, it is also impossible to crawl all web pages with all possible browser configurations.

While traditional scrapers could probably be retrofitted to work as a honeyclients, we note that to be of use, they would have to be modified to be able to act as a vulnerable browser, and to perform automatic analysis of the crawled content to spot potential malicious activity.

Prior research using honeyclients showed that the amount of existing exploit sites was higher for older versions of a specific Windows version [Wan+05]. Existing research mainly focused on describing particular honeyclients [IHF08; DS14; MWF14], honeyclient frameworks [QH10; OS10; PCS16], honeyclient usage strategies [QZ11; Kim+12] or honeyclient analysis evasion [Kap+11].

5.1 Honeyclient types

In related literature, a distinction is made between two types of honeyclients. The distinction between these two types is important to highlight as each type has its own specific advantages and disadvantages, which also play a potentially important role in the results of this research.

High-interaction honeyclients use real browsers on top of a specific operating system environment [QH10]. A master system controls the, usually virtualized slave system, which is running the vulnerable browser. This slave system is used to visit selected web pages. After a web page is visited, the state of the slave system is analyzed to detect traces of potentially malicious activity. If this type of activity is detected, the web page is considered to be malicious. High-interaction honeyclients contain all functionality of a real browser. Using high-interaction honeyclients is very resource intensive as a real system is needed for each browser configuration to be tested. A second disadvantage of a high-interaction client is that since they are fully functional systems that are designed to become compromised, there is a danger that an infection of the slave system could lead to an infection of the master system in case the malware is capable of doing so. In case of using a virtualized environment, the malware might be able of breaking out of the guest host. A popular open source example of a high-interaction honeyclient using virtualization is *cuckoo sandbox*.

Low-interaction honeyclients use emulated browsers. The functionality of different browsers and browser versions is emulated, as well as their vulnerabilities [QH10]. One advantage of low-interaction honeyclinets is their lightweight footprint. Since it is possible to emulate multiple browser configurations, only one real system is needed to test a webpage using multiple configurations. A second advantage is that while they emulate vulnerable functionality, they do not allow the actual exploitation of those vulnerabilities, thus leading to less chance of the test system becoming compromised. A disadvantage of low-interaction honeyclients is their lack of full browser functionality, making it easy to detect them. A popular open source example of a low-interaction honeyclient is *thug*.

5.2 Built-in analysis and enrichment tools

Honeyclients do not just behave like vulnerable clients. They usually also include a toolset to detect potentially suspicious behaviour, and log or store artifacts that could be used by an analyst to investigate what happened while visiting a specific web page. Two examples of this are Thug's capability of submitting samples to VirusTotal and logging the result, and its ability to also log potential tracking cookie information[1]. Similarly, Lookyloo also performs specific analysis and shows the results. Aside from the main use cases, it includes integrations with third party tools to perform additional analysis. Examples of this are performing WHOIS lookups of domains and IPs found during the Lookyloo analysis, as well as submitting samples to VirusTotal and providing more context about content received in case it was known, legitimate software[2].

[1]`https://thug-honeyclient.readthedocs.io/en/latest/usage.html`
[2]`https://www.lookyloo.eu/docs/main/lookyloo-integration.html`

Chapter 6

Analysis of the extent to which security web crawlers can be blocked by generic targeting and bot detection techniques

6.1 Introduction

Security web crawlers, such as honeyclients, are used to analyze web pages with the goal of detecting malicious behaviour. Cyber criminals on the other hand want to prevent this detection from succeeding. One possibility to achieve this might be to fingerprint individual security web crawlers, and to hide the malicious content if the fingerprint is detected. This would potentially require analyzing each individual crawler however. Doing so could be labour intensive, and in some cases, not practically feasible though. Another option might be to implement more generic targeting and bot detection techniques. Implementing such techniques could potentially prevent analysis by a whole set of tools, without the need to individually fingerprint each, leading to a high benefit at a fairly low cost. In this part of the research we focus on such techniques, as they could be considered the low hanging fruit. Specifically, we analyze the extent to which security web crawlers can be blocked by generic targeting and bot detection techniques.

6.2 Methodology

This section provides a more in depth overview of the research methods and data that were used to reach the final contributions as well as the design decisions that were made while developing this part of the research. The books 'Designing a Research Project – Second edition' [VD10] and 'Onderzoeksvaardigheden' [NVV18] were used as reference for a theoretical background in how to create a research plan for this and all other methodologies. The authors of the former describe five possible research strategies. For each contribution, the different potential research strategies are discussed and the reasons for choosing particular ones are given. Grounded theory is not a valid research strategy for any of the contributions in this research, as none of them has developing a theory as a goal. This leaves the following research strategies to discuss: survey, experiment, case study and desk research. For all desk research, Hugo Jonker's approach to finding literature[1] will be used. This approach consists of searching for key terms, performing a second selection to filter out sources that are not relevant and then snowballing by looking at citations, authors and venues related to previously selected articles.

Possible research strategies for this part of the research are described below.

- **Survey** A survey is more tailored for analysis in breadth rather than in depth analysis, as a result it is not a suitable approach for this contribution.
- **Experiment** A quasi-experimental approach is ideal for this contribution and in particular to verify whether certain modifications to generic targeting and bot detection tools, for example adding or removing particular functionality, lead to changes in the efficacy with which selected popular honeyclients avoid detection by generic bot detection techniques. Performing such modifications is only possible in case the research team has access to the source code of the tool. Having an implementation of a set of generic targeting and bot detection techniques is a requirement to be able to perform these experiments. As it seems there is no open source example of such an implementation, this will have to be created.
- **Case study** A comparative case study can be used to compare different security web crawlers. In particular, security web crawlers could be used to attempt to access a particular page which has a specific, generic bot detection technique, protecting it. A security web crawlers would be tested this way against a selected set of generic targeting and bot detection techniques. These results can then be compared.
- **Desk research** A desk research research method can not be used to create new empirical data, but it could be used to discover generic targeting and bot detection techniques described in prior research. Those detection techniques could then potentially be implemented. Desk research can also be used to determine potential countermeasures and improvements for security web crawlers. This can also be useful in situations where creating a specific implementation is not feasible within the scope of this research, due to time or resource constraints.

The chosen research methods for this part of the research are experiment, comparative case study and desk research. Two open-source honeyclients have been selected and installed for testing. The first is Thug [Thub], an open-source low-interaction honeyclient under active development. The second is Cuckoo Sandbox [Sana], an open-source high-interaction honeyclient. Note that according to a notice published on Cuckoo Sandbox' github page [Sanb], Cuckoo Sandbox 2.x is currently unmaintained as a full rewrite

[1]https://www.open.ou.nl/hjo/writing/literature.htm

of cuckoo is ongoing. In addition, a cloud-based honeyclient-like analysis services was chosen as well. Lookyloo [Loo] is an open source project which has a public instance running at `https://lookyloo.circl.lu/`. "*Lookyloo is a web interface that captures a webpage and then displays a tree of the domains, that call each other.*" The last crawler chosen for this part of the research is VirusTotal [Virb]. VirusTotal is a well-known tool in the IT security community. When a file or url is submitted to VirusTotal, it is inspected by a large set of antivirus scanners and URL/domain blocklisting services [Vira]. The analysis results are returned to the submitter.

In the initial phase, a set of generic targeting and bot detection techniques were chosen and presented. This set was be created by combining techniques specified or mentioned in multiple academic papers, with potentially new techniques found by analyzing the Mozilla Developer Network Web APIs documentation [Mozf]. In this phase, we also reviewed particular techniques mentioned by Brendan Dolan-Gavitt & Yacin Nadji [DN10].

In a second phase, the chosen techniques were implemented in a new web application that can be used for testing. The implementation was validated manually by simulating the different connection scenarios. We note here that the application includes explanations of how a specific technique works on each page. While this does not contribute to the main research question that is being tackled with this part of the research, it could potentially be useful in spreading awareness of cloaking usage later on. As such, this contribution could also play a role for RQ2, which relates to the awareness of cyber security professionals about this topic. Having a concrete implementation allows a teacher or presenter to use the concreteness fading teaching strategy. With this strategy, a concept is presented with gradually increasing levels of abstraction. This has been proven to be an effective teaching strategy [Fyf+14].

In the third phase, we tested the chosen crawlers using the implemented web applications.

6.3 Implementation

In this subsection we present the design decisions made when creating the web application implementation of the techniques mentioned above.

6.3.1 Choice of generic targeting and bot detection techniques

The techniques mentioned and that we deemed to still be relevant in Section 4.3 formed the basis of our web application. We chose to focus on techniques that are commonly mentioned, cover both properties and behavioural attributes and provide both examples of newer and deprecated techniques. The full list of implemented techniques can be found below.

- Geolocation – IP based
- Referer

- User-Agent
- Cookie
- System date
- Immediate browser history
- Opener
- User-Agent consistency check
- Fingerprint using fingerprintjs. FingerprintJS offers an open source [Fin] as well as a professional [fA], commercial version of a browser fingerprinting API. While determining the fingerprintability of particular security web crawlers is not the main goal of this part of the research, we still find it interesting to include this particular functionality in the test web application. The reason for that is twofold. Firstly, it does give us some indication of the fingerprintability of each crawler and secondly, it gives us a concrete example of fingerprintability to show during potential awareness trainings or presentations.
- Mouse movement
- Captcha
- Alert
- Confirm
- Notification
- Microphone and camera access
- Presence of webcam (video input device) – without access granted
- Geo API location access

6.3.2 Web application implementation

We chose the Django Python web framework[2] for the implementation of the web application. Django is inspired by the model-view-controller (MVC)[3] architecture. It does have a slightly different architecture however, called model, template, view (MTV). The Django framework was chosen because it is written in Python, one of top five most popular programming languages [OGr]. Another reason is the fact that Django follows the MVC architecture, which should make the developed web application fairly easy to understand for developers with knowledge of the MVC framework. The application can

[2]`https://www.Djangoproject.com/`

[3]`https://folk.universitetetioslo.no/trygver/themes/mvc/mvc-index.html`

be run in two modes. In eicar mode, a download of the anti malware testfile EICAR is triggered[4]. In the other mode, a specific picture with a danger sign is displayed instead. The code is stored on github at `https://github.com/Wachizungu/cloakingDemoSite`. The site is currently run at `https://cloaking.jeroenpinoy.website/cloakingsite/`. It is hosted with NGINX and apache.

As can be seen in Figure 6.1, the navigation menu was divided in three sub menus, *client side – browser state*, *client side – browser behaviour* and *Server side – browser state*. This is mainly for convenience. The names of the menus also indicate whether the check is performed entirely on the server or if there is also a client side component. We do not make the distinction between browser, user and system attributes in the menu names here. This would have been a possibility as well. The choice is mostly a design decision.

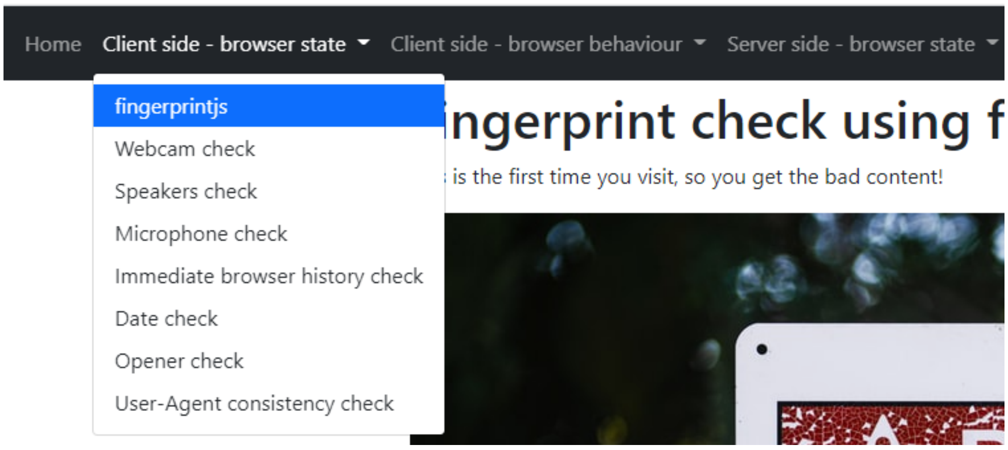

Home
Client side - browser state
Client side - browser behaviour
Server side - browser state
fingerprintjs
Webcam check
Speakers check
Microphone check
Immediate browser history check
Date check
Opener check
User-Agent consistency check
ingerprint check using f
is the first time you visit, so you get the bad content!

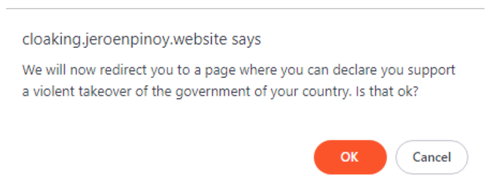
Figure 6.1: Cloaking website navbar

Where possible, we reused existing libraries and Django apps to avoid mistakes by re-implementing already available functionality, and to reduce workload. We used the latest version of the popular bootstrap CSS and jQuery as well. The IP location mapping is done using the Maxmind free geolocation data[5].

For two of the techniques mentioned earlier, we decided to include several variations. For *confirm*, we included one test that expects the user to click ok and another one that expects the user to click cancel. An attacker can tell a user which response they expect, so a crawler should test both options. We also included a third, custom confirm-like prompt using sweetalert2 [Swe]. The reason is that the default confirm window is limited and doesn't allow you to change the text for the responses. An attacker is not limited to using the default either, the potential lack of support in targeted browser version for the custom JavaScript code might be a concern to them however. Screenshots of how both options are rendered in Brave browser can be seen in Figure 6.2. In a

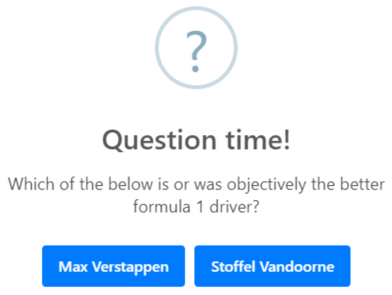
[4]`https://www.eicar.org/?page_id=3950`
[5]`https://dev.maxmind.com/geoip/geolite2-free-geolocation-data`

similar fashion, we implemented three separate captcha pages. Two requiring human interaction, the Django simple captcha [con] app and Google reCAPTCHA v2 [Goo]. The final one, Google reCAPTCHA v3 does not require human interaction, instead, it sends a score back to the server. The score indicates how likely it is that the client is human. Screenshots of the simple captcha and reCAPTCHA v2 can be found in Figure 6.3 and 6.4. While reCAPTCHA v3 does not require interaction, it is not completely transparent to the user. A small widget, as seen in Figure 6.4, is automatically added on the page containing it.


cloaking.jeroenpinoy.website says

We will now redirect you to a page where you can declare you support a violent takeover of the government of your country. Is that ok?

(a) regular


Question time!

Which of the below is or was objectively the better formula 1 driver?

(b) sweetalert2

Figure 6.2: Confirm window

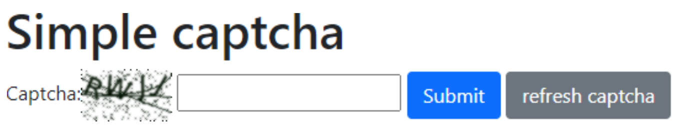
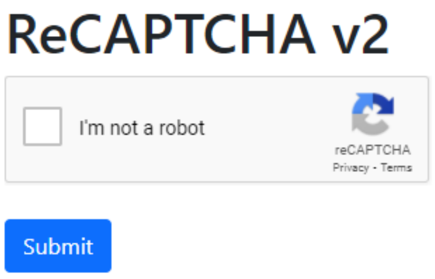
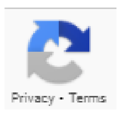

Simple captcha

Figure 6.3: Simple Captcha


ReCAPTCHA v2

(a) reCAPTCHA v2



(b) reCAPTCHA v3

Figure 6.4: Google reCAPTCHA

6.4 Results

6.4.1 Review of See No Evil: Evasions in HoneyMonkey Systems

In Table 6.1, we present the results of our review of the techniques mentioned in “*See No Evil: Evasions in HoneyMonkey Systems*” [DN10].

Attack (technique) mentioned in paper	**Our review**
Alert	This test is still relevant at the time of writing, it does require user interaction and might arouse suspicion however.
Camera	As mentioned in Subsection 4.3, this test is still relevant at the time of writing, even though it does require a different technical implementation than the one mentioned by Brendan Dolan-Gavitt & Yacin Nadji [DN10].
CAPTCHA	This test is still relevant at the time of writing. There are multiple options to implement a CAPTCHA, including Google's reCAPTCHA.
Connection Speed	As discussed in Subsection 4.3, while technically possible to implement, we believe that whether or not this test is relevant is debatable.
Browser History	As discussed in Subsection 4.3, we believe this test is not relevant anymore for modern browsers.
Date	We call this technique *System date*. While we do not believe this test will be useful in a lot of cases, it can still work as designed at the time of writing.
Immediate History	We call this technique *Immediate browser history*. This test is still relevant at the time of writing, the functionality used is still available today.
Message Box	We call this technique *Confirm*. This test is still relevant at the time of writing, it does require user interaction and might arouse suspicion however.
Microphone	Called *Presence of microphone (audio input devices) – without access granted* by us, this check is similar to that of *presence of webcam*, but detects audio input devices. The control tests we performed to validate this technique led to inconsistent results however. We note that this check is still possible by requesting microphone access however, in that case it does require user interaction and might arouse suspicion however.
onMouseMove	We call this technique *Mouse movement*. This test is still relevant at the time of writing and the related functionality has not changed.
onBeforeUnload	We call this technique *Window beforeunload*. As discussed in Subsection 4.3, we believe this technique is unreliable.
Plugins	As discussed in Subsection 4.3, we believe this technique is not as relevant anymore for modern browsers.
Referrer	This test is still relevant at the time of writing, the functionality of the Referer header has not changed.

Table 6.1: Our review of attacker techniques mentioned by [DN10]

6.4.2 Lookyloo results

Lookyloo provides a web interface that allows you to trigger a crawl of a webpage, which then analyses which other urls are redirected to or called and displays the results in a graphical user interface [Loo]. Lookyloo can be used by cyber security professionals to investigate a webpage to determine if it is performing suspicious activities. An example of a Lookyloo analysis result page can be seen in Figure 6.5. Lookyloo does not manage to handle the eicar download, as can be seen in the following analysis result: `https://lookyloo.circl.lu/tree/366d23f1-680d-407b-9930-0107ca287a9a`. Lookyloo analysis results include a screenshot of the page however. As a result, the tests were carried using the *bad picture* mode.

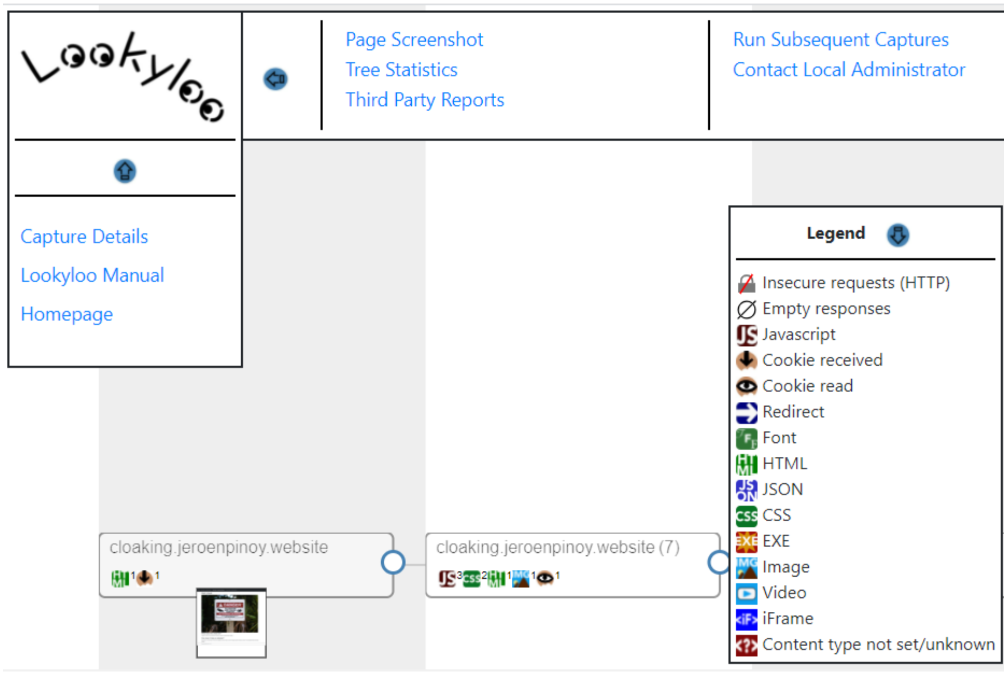


Figure 6.5: Lookyloo analysis result

Lookyloo managed to trigger the malicious content delivery for the *referer*, *user-agent*, *alert*, *confirm – expect cancel* and *cookie* tests. Passing the *referer*, *geolocation – IP based* and *cookie* tests requires using specific configuration parameters when executing the analysis.

The used Lookyloo instance was fingerprintable using FingerprintJS. Subsequent visits, even with a different User-Agent, led to the same visitor ID and thus, more benign content was provided on the second and third visits.

6.4.3 Thug results

Thug is an open source low interaction honeyclient [Thub]. Depending on the configuration, it logs analysis results to files or to a MongoDB[6] instance. Thug also includes functionality to take a screenshot at the end of the analysis [Thua]. After the execution of our control test with the test web application in eicar mode, we found that the following query can be used on the database to determine if the malicious content was successfully triggered.

```
db.behaviors.find({description: /.*44d88612fea8a8f36de82e1278abb02f.*/})
```

Thug only managed to trigger the payload in two of our test scenarios. The first being *referer*, the second *geolocation – IP based.* For many of the other scenarios, the following error was displayed.

```
[handle_events] Event onload not properly handled
```

We opened a discussion on the Thug project Github repository to ask if this was expected behaviour or a known issue[7]. The core developer of Thug confirmed there is indeed an issue, according to him Thug is not able to handle bootstrap v5 at the time of writing. This issue is considered to be low priority by the core developer of Thug since bootstrap v5 officially dropped support for IE 10 and 11. We redid our tests without bootstrap v5. Even then, Thug only managed to trigger the payload in the two test scenarios mentioned before: *referer* and *geolocation – IP based.*

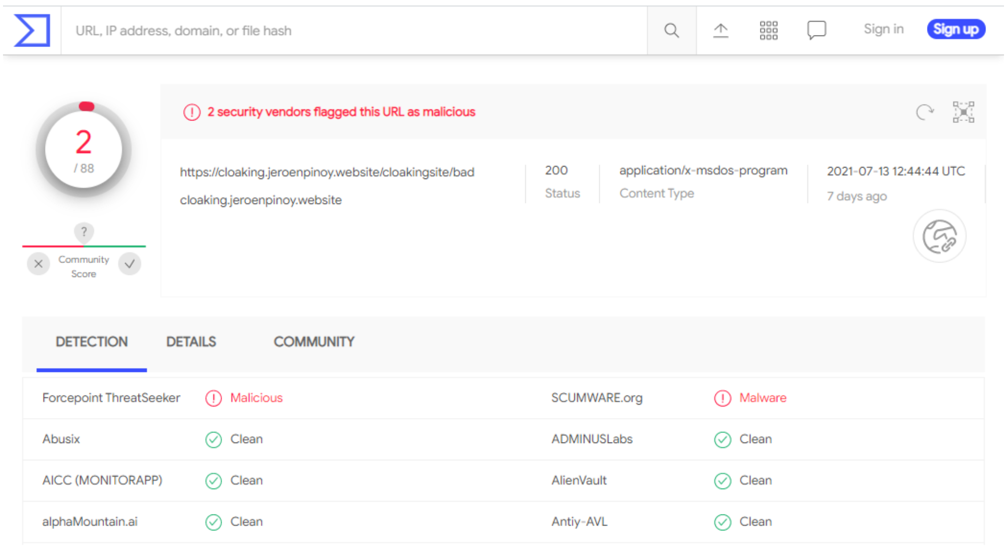
6.4.4 VirusTotal results

VirusTotal (VT) is a well-known tool in the IT security community. When a file or url is submitted to VirusTotal, it is inspected by a large set of antivirus scanners and URL/domain blocklisting services [Vira]. The analysis results are returned to the submitter. An example of a VirusTotal url analysis result page can be seen in Figure 6.6. VirusTotal analysis is a sort of black box, as the inner workings are not clear from the outside. The results can change over time as the tools or reputation engines used are updated. Surprisingly, the results for all of our main test pages were "*No security vendors flagged this URL as malicious*". The control test with the eicar download url gives "*2 security vendors flagged this URL as malicious*" at the time of writing, which is low. This is especially strange considering the fact the analysis at `https://www.virustotal.com/gui/url/bf2664ff2c75022faa70a24bef3af86e04b87b5227e00ea49fb3bc66d58b7ef9/details` does provide the resulting SHA-256 hash of eicar, which is:

```
275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f
```

Entering this hash in VirusTotal gives "*62 security vendors flagged this file as malicious*" at the time of writing (2021-07-27).

[6] `https://www.mongodb.com/`
[7] `https://github.com/buffer/thug/discussions/322`


URL, IP address, domain, or file hash

Sign in Sign up

2 / 88

Community Score

2 security vendors flagged this URL as malicious

https://cloaking.jeroenpinoy.website/cloakingsite/bad
cloaking.jeroenpinoy.website

200 Status

application/x-msdos-program Content Type

2021-07-13 12:44:44 UTC
7 days ago

DETECTION DETAILS COMMUNITY

Forcepoint ThreatSeeker	Malicious	SCUMWARE.org	Malware
Abusix	Clean	ADMINUSLabs	Clean
AICC (MONITORAPP)	Clean	AlienVault	Clean
alphaMountain.ai	Clean	Antiy-AVL	Clean

Figure 6.6: VirusTotal analysis result

6.4.5 Cuckoo Sandbox results

Cuckoo Sandbox is an open source malware sandbox [Sana], which can also serve as a high interaction honeyclient.

We installed Cuckoo on an Ubuntu host, with a Windows 7 ultimate guest system. We chose Windows 7 because the Cuckoo Sandbox documentation lists it as one of the recommended operating systems for the guest system guest[8]. We installed the latest available version of Internet Explorer 11 and Mozilla Firefox on the system. Our control tests showed that Mozilla Firefox was not executed by Cuckoo Sandbox properly. We then rolled back the Firefox version to version 65. This version is orchestrated properly by Cuckoo Sandbox. We ran all our tests with Cuckoo sandbox using both Internet Explorer 11 and Mozilla Firefox 65. We also enabled the *simulate human interaction* mode of Cuckoo Sandbox when executing our tests.

Internet Explorer 11 Cuckoo Sandbox managed to trigger the malicious payload for the *alert*, *confirm – expect cancel* and *mouse movement* tests. We believe it should also be possible to pass the *geolocation – IP based* check, since Cuckoo Sandbox has an option to set a specific network route or vpn to use for a specific analysis. Unfortunately, we were unable to set up the routing rules to test this within the scope of this research due to time limitations. Interestingly, Cuckoo Sandbox fails for some tests which Lookyloo and Thug can pass. This is mainly because of the lack of configuration options that allow you to set the Referer header for example.

[8]`https://cuckoo.readthedocs.io/en/latest/installation/guest/creation/`

Mozilla Firefox 65 Cuckoo Sandbox managed to trigger the malicious payload for the *date* and *mouse movement* tests. Interestingly, there are differences when compared to our runs with Internet Explorer 11. This seems to be due to a lack of support for some of our web application's functionality in Internet Explorer 11, and a lack of support of some of the *simulate human interaction* mode for Mozilla Firefox. Our guest system running Firefox was fingerprintable using FingerprintJS. Subsequent visits, even with a different User-Agent, led to the same visitor ID and thus, more benign content was provided on the second and third visits.

6.4.6 Summary of all crawler results

We present a summary of our findings in Table 6.2. A $\surd$ in the table indicates that the crawler managed to trigger the malicious content. A X means the crawler did not manage to trigger the malicious content. A ? indicates that we believe the crawler should be able to trigger the payload if specific configuration is used, but which we did not manage to test.

The FingerprintJS technique is a special case. Ideally, a crawler should not be fingerprintable. Crawlers that were specifically fingerprintable using FingerprintJS are indicated by $\surd$/X. The other crawlers never managed to trigger the malicious payload on the FingerprintJS check, not even on the first pass.

Cloaking technique	*Lookyloo*	*Thug*	*VT*	*Cuckoo (IE)*	*Cuckoo (FF)*
Geolocation (IP based)	√	√	X	?	?
Referer	√	√	X	X	X
User-Agent	√	X	X	X	X
Cookie	√	X	X	X	X
System date	X	X	X	X	√
Immediate browser history	X	X	X	X	X
Opener	X	X	X	X	X
User-Agent consistency	X	X	X	X	X
FingerprintJS	√/X[1]	X	X	X	√/X[1]
Mouse movement	X	X	X	√	√
CAPTCHAs					
Simple captcha	X	X	X	X	X
reCAPTCHA v2	X	X	X	X	X
reCAPTCHA v3	X	X	X	X	X
client-side					
Alert	√	X	X	√	X
Confirm: expect ok	X	X	X	X	X
Confirm: expect cancel	√	X	X	√	X
Confirm: sweetalert2	X	X	X	X	X
Microphone and camera access	X	X	X	X	X
Notification	X	X	X	X	X
Presence of webcam	X	X	X	X	X
Geo API location access	X	X	X	X	X

√	success: malicious content triggered (note that the crawlers are trying to acquire the malware for analysis)
X	failed: malicious content withheld
?	failed, should be possible to pass with proper configuration, which we were not able to test.
1	First visit receives malicious content, subsequent visits do not.

Table 6.2: Summary of security crawler test results

6.4.7 Proposals for improvement

Generic Most of the tested crawlers lack support for newer technologies such as the notification API, or even the latest browser versions. This results in failing tests across the board. Implementing support for newer technologies or browser versions could improve results. We do note however that focusing on older browser versions makes sense, since they are more likely to have known vulnerabilities and thus are probably targeted more often by attackers. Having configurable parameters for certain browser attributes such as Referer header or cookies can also improve results.

Lookyloo Potential improvements are support for newer technologies such as the notification API and a configuration parameter to click ok instead of cancel when asked to confirm.

Thug Improved support for newer technologies such as the notification API or bootstrap v5. Another potential improvement is the inclusion of a configuration parameter to provide cookies to use when connecting to the url to analyze.

VirusTotal A potential improvement is clarification of how the URL analysis works. Additionally, some content analysis could be triggered for visited urls. The possibility to request a screenshot of the analyzed url would be a nice addition as well.

Cuckoo Sandbox Cuckoo Sandbox lacks analysis parameters to set certain values such as the Referer header or User-Agent header. Adding these could potentially be achieved by using extensions for the installed browsers and creating a custom analysis module. Implementing support for newer operating system and browser versions would be an improvement as well. Finally, we believe it should be possible to add more human interaction simulation and if possible, provide parameters for them. One example could be that instead of clicking cancel on a confirm prompt, the user of Cuckoo Sandbox has the option to configure Cuckoo to click ok.

6.5 Discussion

Some of the results are as expected, with Cuckoo Sandbox managing to simulate certain human interactions, specifically mouse movement, while low interaction honeyclients failed to do so. Many of the results were surprising however, we expected these crawlers to perform better overall. The fact that VirusTotal did not report any pages as suspicious is a disappointment as well, considering the popularity of the service among cyber security professionals.

These results can be used to determine potential areas of improvement. They also serve as a good reminder that you can't rely on tools like this to guarantee that a web page is not malicious.

One interesting avenue for future research with security web crawlers, could be the usage of other crawlers like OpenWPM, used in fingerprinting related research, to develop a honeyclient capable of more human like interaction. Combining the malware analysis functionality of an open source tool like Cuckoo Sandbox, with the more extensive crawling capabilities of OpenWPM, might lead to a system capable of bypassing more advanced cloaking techniques.

Part III

An analysis of the awareness and preparedness of cyber security professionals with regards to cloaking

Chapter 7

Introduction

A survey performed amongst IT leaders of European organisations in 2017, showed that cyber security was one of their top 5 concerns [Kap+19]. In some cases, having certain security controls in place is mandated by a supervisor or certifying body. This is often the case for banks and payment processors. Examples of required compliancy are the payment card industry data security standard (PCI-DSS)[1] and the Society for Worldwide Interbank Financial Telecommunication's Customer Security Programme (CSP)[2]. Considering the increased reliance of organisations on IT infrastructure to perform their business activity, it is now in their best interests to have an organisation wide cyber security strategy. As part of an organisation's cyber strategy, a computer security incident response team (CSIRT) can be founded. Carnegie Mellon University published a whitepaper discussing services that can be offered by such a CSIRT [3]. Relevant services include incident handling, security-related information dissemination, configuration and maintenance of security tools, risk analysis, awareness building, education/training and business continuity and disaster recovery planning.

We argue that to offer these particular services, awareness of attacker's tactics and techniques is vital. Spreading knowledge of this type can be done using shared online repositories. An example of this is the frequently used MITRE ATT&CK® framework. MITRE ATT&CK® is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations [MITb]. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community [MITb]. We noted that while browser fingerprinting is a known attack pattern to MITRE [MITa], it is not part of the ATT&CK knowledge base and related tactics such as reconnaissance, initial access and defense evasion, as well as related techniques such as drive-by-compromise and masquerading do not mention it. This fact, combined with experiences of the researchers, led to the hypothesis that awareness of this technique as well as understanding of how to deal with them, might be lacking amongst cyber security

[1]`https://www.pcisecuritystandards.org/`

[2]`https://www.swift.com/myswift/customer-security-programme-csp`

[3]`https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=53046`

professionals.

Below, we state the research questions related to this part:

- **RQ2-1** How aware are cyber security professionals and their organisations of cloaking techniques?
- **RQ2-2** How prepared are organisations for dealing with malicious web content delivery attacks using cloaking techniques?
- **RQ2-3** What additional steps can organisations take to improve awareness and preparedness to deal with malicious web content delivery attacks using cloaking techniques?

To answer these questions, we combine three main research strategies. We start by performing a case study of two cyber threat intelligence sharing communities, which use MISP as a sharing platform. We follow this up with a survey of cyber security professionals. The results of both are then compared and combined. Desk research is used as a supporting research strategy, to gain the necessary knowledge with regards to prior knowledge and MISP, and to formulate an answer to RQ2-3 by discussing countermeasures and related tools proposed in academic literature.

Chapter 8

Research methodology

This section provides a more in depth overview of the research methods and data that were used to reach the final contributions as well as the design decisions that were made while developing this part of the research. For more information about which resources were used to create the below section, please refer to 6.2.

Possible research strategies for this part of the research are described below.

- **Survey** A survey could be used to gauge the awareness of cyber security professionals with regards to cloaking techniques. Without limiting the scope, this approach would take too much time. Even if only US commercial banks were targeted, 12,000 organisations would have to be contacted [FDI]. If this strategy is chosen, the scope should be limited.

- **Experiment** While it is technically possible to perform attacks against a random set of European financial institutions and assess their performance. This would be against the law in case these institutions did not provide permission. It seems unlikely that permission to perform these types of exercises at financial institutions is often granted and even if it is, it could possibly take a long time for the appropriate approvals to be granted and proper agreements signed. For this reason, the research team believes it is unreasonable to assume this type of experiment can be performed, even at a small group of institutions, within the time frame allocated for this research.

- **Case study** A case study of a specific organisation is not a suitable research strategy for this contribution, for reasons similar to those mentioned under the experiment section. A case study of MISP communities based on data in selected community instances, is possible however. By analysing the built in capabilities of MISP for contextualisation of threat intelligence data, as well as the data shared within those communities, we can potentially draw conclusions with regards to research question 3 and in particular RQ2-1. If there are structured ways of indicating the usage of browser fingerprinting techniques during attacks, or this information was added in an unstructured way to data, this means there is at least some awareness of these techniques amongst cyber professionals. Having a structured, simple way to share threat intelligence with regards to incidents including cloaking, can also contribute to being prepared to deal with them. As a result, this study could also aid in responding to RQ2-2.

- **Desk research** To our knowledge there is no prior research about the topic of RQ2-1. Another possibility is to search documentation of shared online repositories for threat intelligence related information. An example is the MITRE ATT&CK® framework mentioned earlier. The options for the desk research strategy with regards to this particular research question are limited, a simple analysis of this repository was already done prior to starting our more in depth research as well. This strategy can play an important role when determining the answer to RQ2-2 and RQ2-3 however, since prior research potentially analysed countermeasures against cloaking. Note that papers with regards to countermeasures of browser fingerprinting and cloaking were already reviewed in scope of the earlier sections of this research.

The main research strategies for this contribution will be a case study and survey, complemented with desk research. Combining the case study and survey strategy should allow for comparison and triangulation. Triangulation in this context is the usage of different methods to study a particular phenomenon, resulting in a broader and more correct view [Run+09]. A survey will be sent to a select group of organisations from different sectors the author has a connection to. This survey will serve as initial exploratory research with a fairly limited target group. Follow up research could be performed later using the same or an updated set of questions. Cyber security professionals are targeted as the research question directly relates to them. The survey questions will be based on desk research and the outcome of the case study of MISP communities. These questions will be reviewed and tested by a selected small group of participants at the author's institution first, before sending them to other participants.

Data collection for this contribution will be done via the selected MISP communities, and a by the student university's approved web based survey tool, Limesurvey. The survey will be added as addendum to this report.

Chapter 9

Case study: Analysis of MISP communities

We refer back to research question 3 and its secondary questions:

- **RQ2:** How aware and prepared are cyber security professionals and their organisations when it comes to cloaking attacks, and how can their awareness and preparedness be improved?
- **RQ2-1:** How aware are cyber security professionals and their organisations of cloaking techniques?
- **RQ2-2:** How prepared are organisations for dealing with malicious web content delivery attacks using cloaking techniques?
- **RQ2-3:** What additional steps can organisations take to improve awareness and preparedness to deal with malicious web content delivery attacks using cloaking techniques?

The goal of this case study is to contribute to answering RQ2. To this end we will examine contextualisation capabilities in an open source threat intelligence platform with a large user base in the security community: the MISP platform. We will also analyze the data shared within the two case study communities, to determine if data with regards to cloaking incidents is shared. If there are structured ways of indicating the usage of browser fingerprinting techniques during attacks, or this information was added in an unstructured way to data, there is at least some awareness of these techniques amongst cyber professionals. Having a structured, simple way to share threat intelligence with regards to incidents including cloaking, can also contribute to being prepared to deal with them. Threat intelligence platforms such as MISP allow collaboration on investigations, sharing of indicators that could be used to enhance protections and creating human readable reports for analysts and managers. If organisation A found a webpage that seems to be spreading malicious content using cloaking, they can create an event in a MISP community, add whatever information they have and share it. Other organisations can then take action based on this information, for example by blocking connections to the malicious web page or performing extra analysis of the page themselves. They can then potentially share back their findings to the community as well. Use cases such as this lead us to believe organisations actively using these tools in such a way could be considered more prepared. As a result, this study could also aid in responding to RQ2-2 and RQ2-3.

In the remainder of this chapter, we first explain what MISP and MISP communities are, as well as how contextualisation can be added in MISP and how this contextualisation can be searched. We then present the results of our analysis and discuss these results.

9.1 MISP and MISP communities

MISP is an open source threat intelligence platform for sharing, storing and analysing indicators of compromise (IOCs) such as IP addresses, malware samples, financial fraud information or even human readable reports. MISP was originally an acronym for 'Malware Information Sharing Project'. MISP has evolved into an generic purpose intelligence sharing platform however. As a result, MISP is now just called MISP. The core developers of MISP are working for the Computer Incident Response Center Luxembourg (CIRCL), CIRCL is a governmental organisation and is the computer emergency response team (CERT) for the private sector in Luxembourg.

MISP also allows users to build MISP communities. A MISP community is composed of the local organisations on a MISP server and the remote organisations connected by the sync users. MISP has a community-based distribution scheme, using community distributions is the recommended way of using MISP. A graphical presentation of how this distribution works can be seen in Figure 9.1. All main data can be flagged for a particular distribution. The available distributions are *your organisation only*, *this community only*, *connected communities*, *all communities* or a *sharing group*. Sharing groups on the other hand allow you to specify a specific group of organisations and instances to share data with. Due to the complexity of this mechanism, using it is only recommended for use cases where community distribution can not be used. In simple words, a MISP community can be seen as a group of users and organisations, sharing and creating cyber threat intelligence with one another in a collaborative manner.

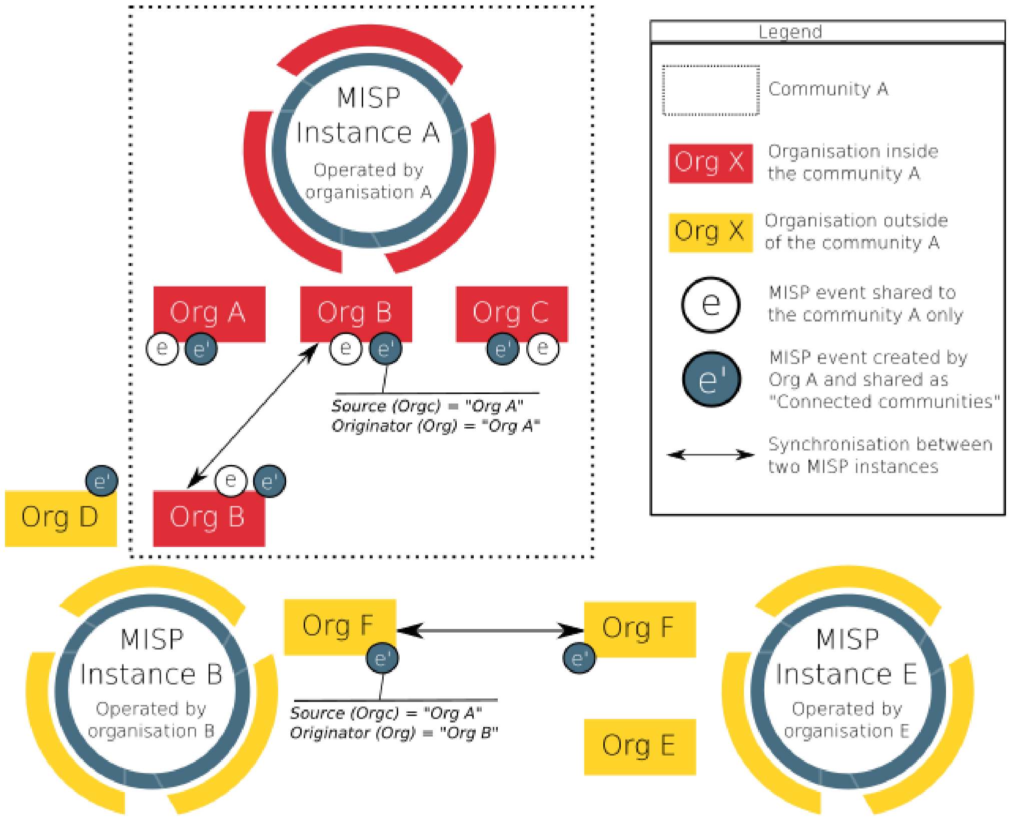
A subset of the default MISP communities list mentioned in the platform [MISa], showing that MISP communities cover a diverse range of topics and members, can be found below :

- **MISPPRIV** CIRCL Private Sector Information Sharing Community
- **CIRCL financial information sharing community**
- **X-ISAC sharing community** X-ISAC (pronounced cross-ISAC) is the supporting Information Sharing and Analysis Center for other ISACs, information sharing

communities or CSIRT networks which provides core software, cross-sector threat intelligence, taxonomies and open standards.

- **COVID-19 MISP community** A community for tracking both health and cyber threat related information around COVID-19

The MISP communities in scope of the case study are the MISPPRIV and COVID-19 MISP community. We chose to only include these to keep the scope withing reasonable bounds. Most of the other communities also have particular requirements to join, making it more complicated for researchers to get access to them.



MISP Instance A
Operated by organisation A
Org A
Org B
Org C
e
e'
Source (Orgc) = "Org A"
Originator (Org) = "Org A"
Org D
Org B
MISP Instance B
Operated by organisation B
Org F
Org F
Source (Orgc) = "Org A"
Originator (Org) = "Org B"
Org E
MISP Instance E
Operated by organisation E
Legend
Community A
Org X
Organisation inside the community A
Org X
Organisation outside of the community A
MISP event shared to the community A only
MISP event created by Org A and shared as "Connected communities"
Synchronisation between two MISP instances

Figure 9.1: MISP community sharing overview, image from the public MISP documentation [MISb]

9.2 Contextualisation in MISP

The MISP platform allows users to automate several business processes. Data in it becomes more valuable as better metadata or contextualisation is added. A relevant selection of MISP's defined user stories, available in the public MISP documentation at [Prob], can be found below in Table 9.2:

User story
As a **cybersecurity specialist**, I want to investigate threats so that I can remediate and prevent cyber attacks
As a **SOC analyst**, I want to share real-time information pertaining to new or existing cases/observables to team members so that we can collaborate on investigations simultaneously
As a **lead threat intelligence analyst**, I want to convert threat data into actionable threat intelligence so that I can improve security posture.
As a **risk analyst**, I want to identify and predict risks to my organisation so that I can improve the organisation's security posture and situational awareness
As a **risk analyst**, I want to present risk data to stakeholders in various formats (depending on their technical ability), so that I can justify the need for risk-mitigating strategies
As a **security analyst**, I want to automate repetitive tasks related to data normalization, importation, aggregation and enrichment so that I can have more time to put into threat analysis efforts

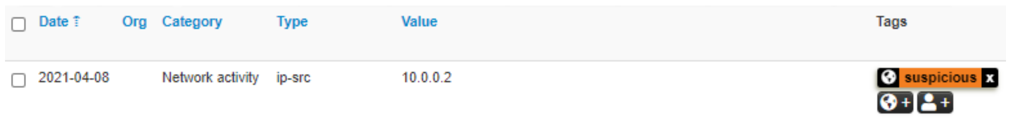
Table 9.1: Relevant MISP user stories [Prob]

Note that all of these use cases are relevant to analysis and risk management of threats such as the ones posed by the potential use of cloaking techniques. Adding metadata in a structured way is necessary for all of the processes related to these use cases to work smoothly and correctly. For example, organisation B could automatically trigger analysis with a honeyclient on potentially malicious web pages. Organisation A might have mentioned that the page uses cloaking and only delivers the malicious content to Belgian IP addresses, but added this information in an unstructured way. Unless organisation B's honeyclient happens to use a Belgian IP address, the honeyclient will not receive the malicious content. If the same information was added in a structured way, organisation B could use it and choose an appropriate proxy to run its honeyclient analysis of the webpage. Having metadata added in a structured way also helps for our research use case, as will become clear later.

Since one of the goals of this case study is to determine whether cyber security professionals are aware of cloaking techniques, and this type of information is usually added as metadata, it is necessary to understand the tools MISP provides for contextualisation and how to search through that data. There are currently six different ways of adding generic metadata or contextualisation to the event data in a MISP. This section gives an overview of those six. While there are other ways of adding contextualisation, those are used to add specific information such as an indication of whether an attribute is likely a false positive. As such, they are not particularly relevant for this research.

9.2.1 Taxonomies and tags

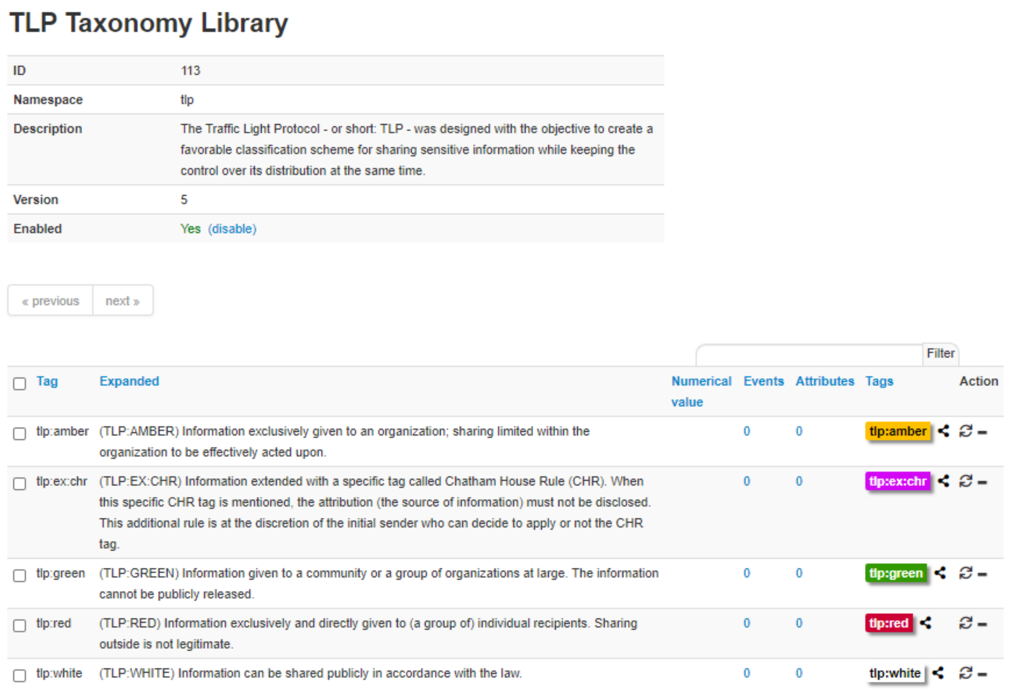
Tags are simple labels that can be added to MISP data such as events and attributes. They can be managed in a structured or unstructured way. The unstructured way is creating separate custom tags on a MISP instance and then subsequently using these. In the example below you can see an example of this. The *suspicious* tag was created as a separate tag and then added to an attribute.



Date	Org	Category	Type	Value	Tags
2021-04-08		Network activity	ip-src	10.0.0.2	suspicious x

Figure 9.2: MISP tagged attribute

The structured approach to tagging is grouping tags in taxonomies. The MISP project already has a set of taxonomies that can be enabled by users. It is also possible to create custom taxonomies using a simple json structured file. An example of a taxonomy can be found below.



TLP Taxonomy Library

ID	113
Namespace	tlp
Description	The Traffic Light Protocol - or short: TLP - was designed with the objective to create a favorable classification scheme for sharing sensitive information while keeping the control over its distribution at the same time.
Version	5
Enabled	Yes (disable)

« previous next »

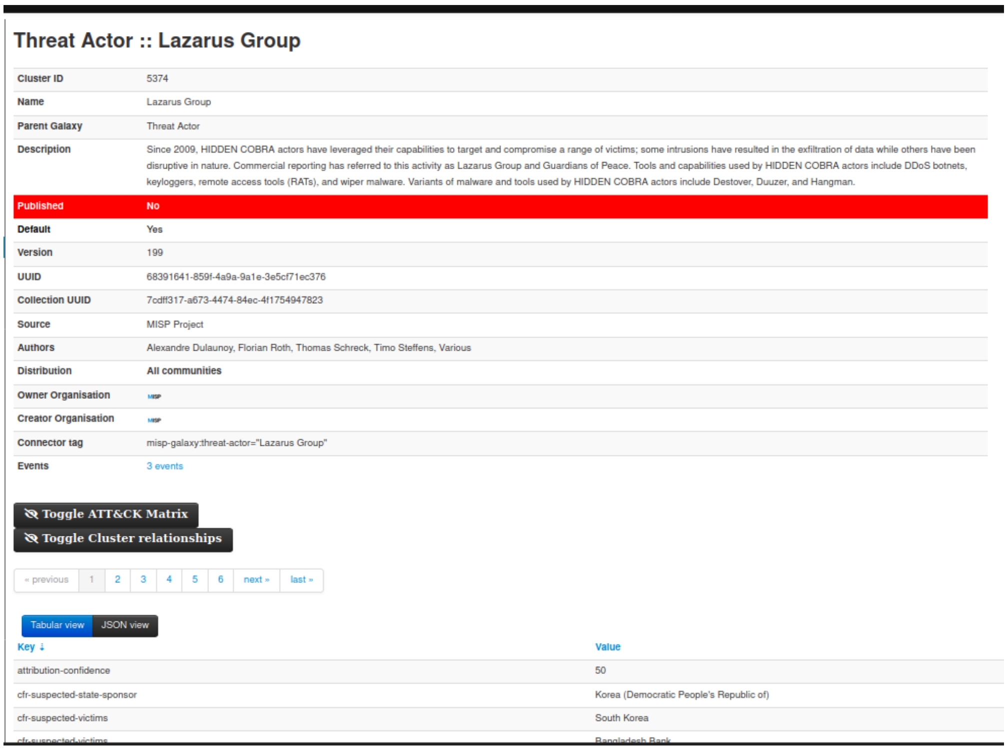
Filter

Tag	Expanded	Numerical value	Events	Attributes	Tags	Action
tlp:amber	(TLP:AMBER) Information exclusively given to an organization; sharing limited within the organization to be effectively acted upon.		0	0	tlp:amber	
tlp:ex:chr	(TLP:EX:CHR) Information extended with a specific tag called Chatham House Rule (CHR). When this specific CHR tag is mentioned, the attribution (the source of information) must not be disclosed. This additional rule is at the discretion of the initial sender who can decide to apply or not the CHR tag.		0	0	tlp:ex:chr	
tlp:green	(TLP:GREEN) Information given to a community or a group of organizations at large. The information cannot be publicly released.		0	0	tlp:green	
tlp:red	(TLP:RED) Information exclusively and directly given to (a group of) individual recipients. Sharing outside is not legitimate.		0	0	tlp:red	
tlp:white	(TLP:WHITE) Information can be shared publicly in accordance with the law.		0	0	tlp:white	

Figure 9.3: MISP tlp taxonomy

9.2.2 Galaxies

A MISP galaxy is a simple, structured method to express a set of metadata called a cluster that can be attached to MISP events or attributes. Galaxies allow you to add a group of linked metadata with just one action, making it easier to provide context, with that context being readily available for access by other analysts as well. In the screenshots below you can find an example of a galaxy cluster and the usage of such a cluster in a MISP event. In the example, the threat actor Lazarus Group cluster was added, indicating the event can be linked to this group. The entry in the MISP event can be hovered over to get extra information.



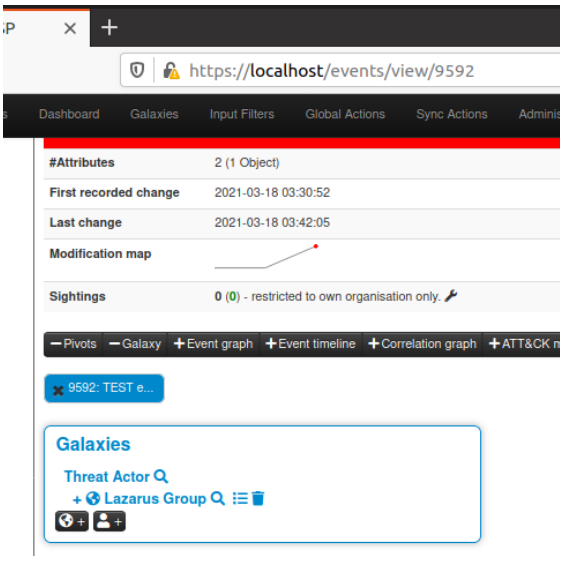
Threat Actor :: Lazarus Group

Cluster ID	5374
Name	Lazarus Group
Parent Galaxy	Threat Actor
Description	Since 2009, HIDDEN COBRA actors have leveraged their capabilities to target and compromise a range of victims; some intrusions have resulted in the exfiltration of data while others have been disruptive in nature. Commercial reporting has referred to this activity as Lazarus Group and Guardians of Peace. Tools and capabilities used by HIDDEN COBRA actors include DDoS botnets, keyloggers, remote access tools (RATs), and wiper malware. Variants of malware and tools used by HIDDEN COBRA actors include Destover, Duuzer, and Hangman.
Published	No
Default	Yes
Version	199
UUID	68391641-859f-4a9a-9a1e-3e5cf71ec376
Collection UUID	7cdff317-a673-4474-84ec-4f1754947823
Source	MISP Project
Authors	Alexandre Dulaunoy, Florian Roth, Thomas Schreck, Timo Steffens, Various
Distribution	All communities
Owner Organisation	MISP
Creator Organisation	MISP
Connector tag	misp-galaxy:threat-actor="Lazarus Group"
Events	3 events

Toggle ATT&CK Matrix
Toggle Cluster relationships
« previous 1 2 3 4 5 6 next » last »
Tabular view
JSON view

Key	Value
attribution-confidence	50
cfr-suspected-state-sponsor	Korea (Democratic People's Republic of)
cfr-suspected-victims	South Korea
cfr-suspected-victims	Bangladesh Bank

Figure 9.4: MISP galaxy



#Attributes	2 (1 Object)
First recorded change	2021-03-18 03:30:52
Last change	2021-03-18 03:42:05
Modification map	
Sightings	0 (0) - restricted to own organisation only.

Galaxies

Threat Actor

+ Lazarus Group

Figure 9.5: MISP event with galaxy

9.2.3 Object relationships

Object relationships allow you to name a relationship between two objects or an object and an attribute. A simple example can be found below. This example shows a file that connects to a specific ip[1].

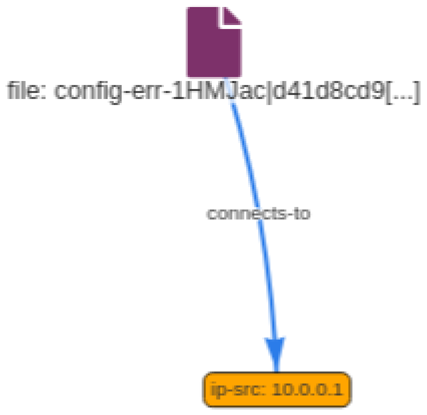


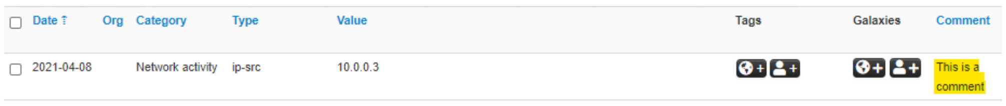
Figure 9.6: MISP object relationship

9.2.4 Object and attribute comments

It is also possible to add simple comments to attributes and objects, as can be seen below. Again, an example is provided below. The comment field is a free text field and

[1]`https://github.com/MISP/misp-objects/blob/main/relationships/definition.json`

leads to unstructured data as a result.

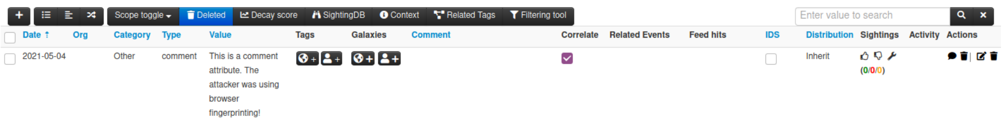


Date	Org	Category	Type	Value	Tags	Galaxies	Comment
2021-04-08		Network activity	ip-src	10.0.0.3			This is a comment

Figure 9.7: MISP attribute with a comment

9.2.5 Comment and text attributes

It is also possible to add comment or text attributes to an event. Comment and text attributes are free text and lead to unstructured data as a result.



Date	Org	Category	Type	Value	Tags	Galaxies	Comment	Correlate	Related Events	Feed hits	IDS	Distribution	Sightings	Activity	Actions
2021-05-04		Other	comment	This is a comment attribute. The attacker was using browser fingerprinting!				✓				Inherit	(0/0/0)		

Figure 9.8: MISP comment attribute

9.2.6 Event reports

Event reports are a fairly new functionality. They are intended to duplicate the information available in an event, presenting it in a more human-readable form. Analysts can use markdown to write these reports and link to event data in them. Event reports behave the same way as event attributes when sharing with others [Proa].

9.3 Searching through MISP contextualisation

The index and detail pages of taxonomies, events and event reports are particularly useful for browsing through the data and performing an in depth review.

String based searches can be performed fairly easily on all types of contextualisation mentioned above. The following specific methods were identified:

- GUI based search functionalities, in particular the event search and event report search. The former allows you to do string based searches for events. According to the documentation[2], the searchall parameter works in the following way: "*Search for a full or a substring (delimited by % for substrings) in the event info, event tags, attribute tags, attribute values or attribute comment fields.*" Unfortunately, the event search with searchall parameter does not match for values in event reports. This is understandable however, searching through event reports can severely impact performance and if best practices are followed, the data contained in the event report should also have been added to the event in other ways, for example in the form of attributes and objects.

[2] `https://www.circl.lu/doc/misp/automation/`

- REST API based search, in particular the /events/restSearch API. The results returned when using the searchall parameter are essentially the same as on the above mentioned GUI based string search for events.
- Extracting the event data in a specific format such as json and then performing data processing on the extract.
- SQL queries using the database. The MISP database schema can be found in the MISP github repository[3]. This gives you direct access to the full data, allowing for more granular searches, including on potentially soft-deleted data.
- In the case of galaxies and taxonomies, string searches can also be done using the github search functionality in the respective git repositories[4][5], this can not always be relied on due to github not searching certain files if they are too large. An alternative could be cloning the git repository and performing string searching on the files.

9.4 Results

We gained access to two MISP communities, MISPPRIV and COVID-19 MISP. The data in the respective community instances was pulled in to two separate local MISP instances by executing a pull sync with a local user.

Different search strategies were used for specific types of MISP data. In some cases, DB queries were executed directly. This potentially leads to different results as when using MISP functionality, since deleted or archived entries might not be returned by MISP in certain scenarios. We argue that this difference is not of significant importance, unless no results are found at all. This is because our goal is to determine whether analysts are aware of browser fingerprinting techniques and if so, whether they add contextualisation in a structured way when mentioning its use. We argue that if any results are found, there is at least some awareness within the community.

9.4.1 Default MISP taxonomies

String searches for the terms *fingerprint*, *browser* and *cloaking* were performed on the MISP project taxonomies repository, which contains the default MISP taxonomies[6], using a locally cloned copy of commit e4930e6c0eb80e8d6d8adbd021ece77c35d5ca35. The following relevant results were found. The **fingerprint** tag from the cycat taxonomy, which has the following description: "*Code to uniquely identify specific cybersecurity-relevant patterns. Fingerprints can be expressed in different formats such as ja3, ja3s,*

[3]`https://github.com/MISP/MISP/blob/2.4/db_schema.json`
[4]`https://github.com/MISP/misp-galaxy`
[5]`https://github.com/MISP/misp-taxonomies`
[6]`https://github.com/MISP/misp-taxonomies`

hassh, jarm or favicon-mmh3.”[7]. In the maec-malware-capabilities taxonomy, which is based on the MAEC 5 specification[8], the following tags were found:

- **compare-host-fingerprints** “*Compares a previously computed host fingerprint to one computed for the current system on which the malware instance is executing, to determine if the malware instance is still executing on the same system.*”
- **fingerprint-host** “*Creates a unique fingerprint for the system on which the malware instance is executing, e.g., based on the applications that are installed on the system.*”
- **environment-awareness** “*Indicates that the malware instance or family can fingerprint or otherwise identify the environment in which it is executing, for the purpose of altering its behavior based on this environment.*”

This result leads to a first observation: There are existing default taxonomy tags that when used, could indicate usage of cloaking. These tags lack granularity however. They can be used for many types of fingerprinting. As a result it is not possible to automatically determine whether they are related to an attack involving cloaking as defined in this research or not.

9.4.2 Galaxies

String searches for the terms *fingerprint*, *browser* and *cloaking* were performed on the MISP project galaxy repository[9], which contains the default MISP galaxies, using a locally cloned copy of commit 6c8949caa920296b8d6d10065d9521d3e33bf185.

The only relevant hits not related to specific tools or attacker groups are the deprecated *Unconditional client-side exploitation/Injected Website/Driveby – PRE-T1149* entry from the mitre-pre-attack-attack-pattern cluster and *Drive-by Compromise* from the mitre-ics-techniques cluster. There are several other MITRE related attack techniques that can be performed using browser fingerprinting, but they are not specific enough.

The tool cluster includes an entry for RICECURRY with the following description: “*RICECURRY is a Javascript based profiler used to fingerprint a victim’s web browser and deliver malicious code in return. Browser, operating system, and Adobe Flash version are detected by RICECURRY, which may be a modified version of PluginDetect.*”.

We note that the latest version of the MITRE ATT&CK framework, released end of April 2021[10] has a newly defined technique that is very similar to cloaking as defined in this research called *Drive-by Target*[11]. The emphasis in their description is on detecting vulnerabilities or performing profile by using a waterhole attack though. There is still

[7]https://github.com/MISP/misp-taxonomies/blob/main/cycat/machinetag.json

[8]https://maecproject.github.io/documentation/maec5-docs/#behaviors

[9]https://github.com/MISP/misp-galaxy

[10]https://attack.mitre.org/resources/updates/updates-april-2021/

[11]https://attack.mitre.org/techniques/T1608/004/

no specific mention of using fingerprinting to actively avoid detection by researchers or analysts. This version has now been integrated with MISP but was not yet part of the platform when we performed our full analysis.

This result leads to a similar observation as for taxonomies: There are several existing default galaxy clusters that when used, could indicate usage of cloaking. These clusters still lack some granularity however. The most specific one is the Drive-by Target technique from the MITRE Attack Pattern galaxy, but it does not necessarily indicate detection evasion using cloaking

Keeping track of adoption of usage of this new galaxy cluster in MISP communities, could be interesting follow up research.

9.4.3 Object relationships

The default MISP object relationships are defined in the misp-objects repository on github[12].

We performed searches for the terms *fingerprint*, *browser* and *cloaking* on the file. This was followed up by a manual review of the different relationships. Potentially relevant relanship types are *properties-queried* and *properties-queried-by.* The following SQL query was run to determine which relationship types were used in each community.

```
SELECT distinct relationship_type from object_references
INTO OUTFILE '/tmp/relationship_types.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

The results for both the COVID-19 and MISPPRIV communities are displayed in Appendix B. None of the relationship types are likely to be related to browser fingerprinting.

9.4.4 Tags

For this analysis, the tags were exported to json for future reference. After that, string searches for the terms *fingerprint*, *browser* and *cloaking* were performed on the MISP instance on which the data was imported. In a final phase the researcher manually reviewed all tags and checked whether the name could be a synonym for browser fingerprinting. In both MISP communities, a subset of the tags were actually galaxy tags. We decided to include them in the analysis anyway as they might have been created for custom galaxies that are not part of the default MISP galaxies. In that case, these tags would not have been found in the generic galaxies analysis described earlier.

COVID-19 MISP At the time of the analysis, 705 tags were used in the COVID-19 MISP. None of these, except for the the cycat fingerprint tag, have a reasonable chance to potentially be related to browser fingerprinting.

[12]https://github.com/MISP/misp-objects/blob/main/relationships/definition.json

MISPPRIV At the time of the analysis, 3,294 tags were used in the COVID-19 MISP. None of these, except for the the cycat fingerprint tag, have a reasonable chance to potentially be related to browser fingerprinting.

While there are no tags that directly indicate browser fingerprinting was used, there are some tags that are more likely to be added in situations where browser fingerprinting might have happened as well, *exploit kit* is an example of this. Nevertheless, this result indicates that no tags and galaxy clusters directly related to browser fingerprinting, were used in these MISP communities.

9.4.5 Objects, attributes, event info

For this analysis, the restSearch API was used, with variations of the following request for the strings *fingerprint*, *browser* and *cloaking*.

```
curl \
-d '{"returnFormat":"json","searchall":"%cloaking%"}' \
-H "Authorization: [API_KEY]" \
-H "Accept: application/json" \
-H "Content-type: application/json" \
-X POST https://localhost/attributes/restSearch
```

COVID-19 MISP The search for cloaking yielded no results. The searches for fingerprint and browser had 91 and 111 hits for attributes, but none of the attributes or related events were clearly linked to browser fingerprinting.

MISPPRIV There were 2 hits for cloaking, neither are related to browser fingerprinting.

There were 876 attribute hits for fingerprint and 734 attribute hits for browser. The relevant results are presented below. We always include the event UUID and the content of the event info field.

59f01bb9-c1f8-487f-8bee-3adfac12042b2, Diskcoder.D/BadRabbit Outbreak String matches for `%fingerprint%` on several ip-dst attributes with comment *Victim fingerprinting and malware delivery.* The following is mentioned in a linked report [13]: "*Before the actual malicious file was downloaded a POST request was observed to a static IP address (185.149.120[.]3). This request was found to be posting to a static path of "/scholasgoogle" and provided the user agent, referring site, cookie, and domain name of the session. After the POST the dropper was downloaded from two different paths from 1dnscontrol[.]com, /index.php and /flash_install.php.*" The event also contains four ip-dst attributes with comment "*Victim fingerprinting and malware delivery*"

[13] `https://blog.talosintelligence.com/2017/10/bad-rabbit.html`

54c2050d-cc18-4fa5-b6e2-45bf950d210b, OSINT – The Waterbug attack group String matches for `%fingerprint%` on several hostname attributes with comment *Used for fingerprinting.* The following is mentioned in a linked report [14]: "*However, unlike traditional watering-hole attacks, where all visitors to a particular website are targeted indiscriminately, in the case of the Venom network used by the Waterbug group, the attackers use a more deliberate approach. This is done in a multi-staged fashion by firstly redirecting visitors to another malicious server. On the malicious server, a fingerprinting script is executed and this extracts configuration information from the user's computer related to installed bowser plugins (Adobe Reader, Silverlight, Java, Flash etc.). The attackers also collect basic system and network information, such operating system version, type, browser version, and internet protocol (IP) address. At this point, the attackers have enough information to determine if the visitor is of further interest. When an IP address of interest is identified, such as one associated with a government institution, they proceed to create a rule specific to that IP address. This rule ensures that the next time the visitor arrives on the compromised website their computer may be sent a malicious payload instead of just being fingerprinted.*"

564a5a82-8aa0-419f-b8c4-4a17950d210b, OSINT Pinpointing Targets: Exploiting Web Analytics to Ensnare Victims – WitchCoven by FireEye String match for both `%fingerprint%` and `%browser%` on an md5 attribute which has the following text as a contextual comment: "*JavaScript file to fingerprint browser and plugins, then redirect to*". Aside from that, a comment attribute in the event contains the following text: "*The script contains the jQuery JavaScript library that is obfuscated to remove variable names. The full script is quite lengthy but is largely comprised of identical or slightly modified versions of publicly available scripts, including evercookie, PluginDetect and the detect Office module from the Browser Exploitation Framework Project.*".

5be9c820-b558-4494-a96e-4a4d0a021402, OSINT: Sensitive Data Exposure via Battery Information Broadcasts in Android OS [CVE-2018-15835] String match for both `%fingerprint%` and `%browser%` on a text attribute, with the following relevant content: "*System broadcasts by the Android operating system expose detailed information about the battery. Prior research has demonstrated that same charging information – when exposed via browser battery status API – can be used to uniquely identify and track users.*".

5c461015-56a4-40e0-a12a-1f540a021402, OSINT: Fallout Exploit Kit now includes exploit for CVE-2018-15982 Flash zero-day String match for both `%fingerprint%` and `%browser%` on a text attribute, with the following relevant content: " *Experts at Malwarebytes observed a new version of the Fallout Exploit kit that include the code to exploit a recently discovered Flash zero-day vulnerability. The Fallout Exploit kit was discovered at the end of August by the threat analyst nao_sec, at the time it was used to*

[14]`https://docs.broadcom.com/doc/waterbug-attack-group`

distribute the GandCrab ransomware and other malicious codes, including droppers and potentially unwanted programs (PUPs). First detailed in September 2018, the toolkit was observed delivering malware families ranging from ransomware to backdoors, but also fingerprinting the browser profile to identify targets of interest."

5c3f64cc-d508-4afd-9b6e-43e30a021402, OSINT: New Magecart Attack Delivered Through Compromised Advertising Supply Chain String match for both `%fingerprint%` and `%browser%` on a text attribute, with the following relevant content: "*Part of its fingerprinting routine includes checking if the script is running on a mobile device (by checking the browser User-Agent) and if there are handlers that check if the browser debugger is on. The fingerprinting routines are done to confirm that the browser session is from an actual consumer.*"

5ee7402e-22b8-4409-9675-546dac13a7a7, Earth Empusa Uses Phishing Attacks to Deploy New ActionSpy Malware String match for `%browser%` on a comment attribute, with the following relevant content: "*First is the injection in the ScanBox framework to collect information from a website's visitors by using JavaScript to record keypresses and harvest the profiles of the OS, browser, and browser plugins from the client environment.*"

57e95d7a-c1a4-4ca2-8e79-e730646d1a36, Phishing – Browser Reconnaissance String match for `%browser%` on a comment attribute, with the following relevant content: "*Sends user to savethechildren[.]org if the browser does not meet certain criteria.*"

56ca3897-a370-42f8-9899-4bb19062e56a, WitchCoven: Exploiting Web Analytics to Ensnare Victims String match for `%browser%` on a url attribute pointing to https://panopticlick.eff.org/browser-uniqueness.pdf, note that was likely a reference to a research paper. The event also references an external analysis report from FireEye, describing a browser profiling script they called witchcoven[15]

[15] `https://www2.fireeye.com/rs/848-DID-242/images/rpt-witchcoven.pdf`

Related event UUID	Search string hit
59f01bb9-c1f8-487f-8bee-3adfac12042b	fingerprint
54c2050d-cc18-4fa5-b6e2-45bf950d210b	fingerprint
564a5a82-8aa0-419f-b8c4-4a17950d210b	fingerprint, browser
5be9c820-b558-4494-a96e-4a4d0a021402	fingerprint, browser
5c461015-56a4-40e0-a12a-1f540a021402	fingerprint, browser
5c3f64cc-d508-4afd-9b6e-43e30a021402	fingerprint, browser
5ee7402e-22b8-4409-9675-546dac13a7a7	browser
57e95d7a-c1a4-4ca2-8e79-e730646d1a36	browser
56ca3897-a370-42f8-9899-4bb19062e56a	browser

Table 9.2: Relevant events based on string searches

9.4.6 Event reports

String searches for the terms *fingerprint*, *browser* and *cloaking* were performed using the eventReports search functionality of the MISP GUI.

COVID-19 MISP At the time of the analysis, the COVID-19 MISP community did not have any event reports.

MISPPRIV Three hits were returned when searching for *browser*, but none of those had content related to browser fingerprinting.

9.5 Discussion

We have seen that multiple events in the MISPPRIV MISP community had data related to usage of cloaking techniques. This indicates that at least some members of this community were aware of these techniques. Based on the above results, the answer to RQ2-1 is that there is at least some awareness of browser fingerprinting techniques amongst security professionals. Nevertheless, the amount of events found is fairly limited.

We also found multiple structured methods for contextualisation, in particular object relationships, taxonomy tags and galaxy clusters, that could be used to indicate usage of cloaking. We consider none of the existing object relationships, taxonomy tags or galaxy clusters to be granular enough to indicate usage of certain cloaking techniques however. We consider it likely that a more structured method would have been added if the awareness of these techniques were higher. The user stories mentioned earlier are all hindered by the lack of structured contextualization methods for incidents using cloaking:

> **As a cybersecurity specialist, I want to investigate threats so that I can remediate and prevent cyber attacks** If I want to perform such an investigation into cloaking as a cybersecurity specialist, I need to sift through

a lot of potentially unrelated information, because I can not easily search for events or attributes due to a lack of granularity in the contextualisation options, and the ways in which other analysts entered their data.

As a SOC analyst, I want to share real-time information pertaining to new or existing cases/observables to team members so that we can collaborate on investigations simultaneously Effective collaboration during a new or existing case relies on a common understanding of certain concepts and terminology.

As a lead threat intelligence analyst, I want to convert threat data into actionable threat intelligence so that I can improve security posture. This is similar to the previous user story. If I want to perform such a conversion as a threat intelligence analyst specialist, I might not be able to determine proper actions due to a lack of granularity in the contextualisation added.

As a risk analyst, I want to identify and predict risks to my organisation so that I can improve the organisation's security posture and situational awareness As a risk analyst, I could use trend analysis or general statistics to determine how frequently cloaking is used and if this frequency is increasing. This could be an input for my risk assessment or something to report to management. The lack of decent, granular contextualisation makes this a lot more cumbersome though.

As a risk analyst, I want to present risk data to stakeholders in various formats (depending on their technical ability), so that I can justify the need for risk-mitigating strategies The issue here is essentially the same as for the previous user story.

As a security analyst, I want to automate repetitive tasks related to data normalization, importation, aggregation and enrichment so that I can have more time to put into threat analysis efforts Automated enrichment of bad urls could be influenced by having a tag or cluster for browser fingerprinting. For example, if something is tagged with 'browser-fingerprinting:detection-evasion' and PAP allows it, we could make MISP trigger enrichment using services that have anti-cloaking, such as using spoofed browser profiles, multiple machines with significantly different setups, or even just different IPs. Again, due to the lack of structured contextualisation for browser-fingerprinting, this is not currently possible in a simple way.

Before continuing, we repeat the main research questions related to this part of the research:

- **RQ2:** How aware and prepared are cyber security professionals and their organisations when it comes to cloaking attacks, and how can their awareness and preparedness be improved?
- **RQ2-1:** How aware are cyber security professionals and their organisations of cloaking techniques?
- **RQ2-2:** How prepared are organisations for dealing with malicious web content delivery attacks using cloaking techniques?
- **RQ2-3:** What additional steps can organisations take to improve awareness and preparedness to deal with malicious web content delivery attacks using cloaking techniques?

All of the above cases raise the question of how good the data quality of the data in our MISP communities is. For browser fingerprinting this seems to be low. According to the Data Management Association International (DAMA) UK, there are six main data quality dimensions: Completeness, uniqueness, timeliness, validity, accuracy and consistency [Ask+13]. We noted that the proper use of some of the above structured techniques to add contextualisation, could lead to higher data quality. This clearly has advantages when attempting to analyse or use the data. Using structured methods of adding metadata likely improves the consistency of the data. Using galaxies adds to completeness as more related information is automatically linked. Aside from the tools for contextualisation, other functionality of a threat intelligence tool such as MISP also directly helps in maintaining data quality. Having built in deduplication of events can improve uniqueness of data entries for example. Finally, the analyst still has a big responsibility in making sure the data quality is high. In the end, it is the analysts job to make sure data is entered completely, accurately, in a timely manner and using the best practices.

Using a tool such as MISP also likely contributes in being prepared to deal with attacks, including those using browser fingerprinting. A first example use case is the collaboration during an incident. Multiple analysts can investigate the same incident or similar incidents at the same time and share information about those incidents with each other in a structured way. A second example is the automation of repetitive tasks. A url attribute having a tag mentioning it is using location based cloaking and targeting users from Belgium, might trigger an automated analysis using a Belgian proxy. This is relevant for RQ2-2, although we note that the lack of a good, granular, structured contextualisation option for browser fingerprinting makes MISP less effective for this than it could be. We argue that improving these contextualisation options would benefit the preparedness of organisation to deal with cloaking attacks. As such, this result also relates to RQ2-3.

Follow up research that could be done is using the /events/restSearch endpoint to search the event info field. We believe this is unlikely to give much extra hits however, as we expect that events related to cloaking would also contain attributes mentioning it. Nevertheless, we already ran the relevant queries and have the data ready for analysis.

In a similar vein, it is possible there are other search terms we are unaware of, which might be used as synonym to cloaking. It is perhaps possible to review events one by one for potentially relevant data. This would be very time consuming however. Finally, keeping track of adoption of the new Drive-by Target MITRE galaxy cluster in MISP communities, could be interesting follow up research as well.

Chapter 10

Survey of cyber security professionals

10.1 Survey design

The books 'The art of asking questions' [Pay51], 'Onderzoeksvaardigheden' [NVV18] and 'Fragebugen' [Por14] were used as reference for a theoretical background in how to create a survey design.

The variables of the relevant research questions are cyber security professionals, organisations, awareness of cloaking techniques and preparedness to defend against drive-by-download attacks using cloaking. Note that drive-by-download is slightly different from the term malicious web content delivery we used in earlier parts of this research. When preparing this survey, we focused on drive-by-download attacks. In a drive-by-download attack, a malicious payload is delivered and executed on an unsuspecting user's system, while they are browsing the web in a normal way. After executing this survey, we broadened the scope of our research to also include other types of malicious web content delivery such as phishing content on phishing sites. For our survey, we translated the aforementioned variables to the following components:

- Cyber security professionals
  - Years of cyber security related experience
  - Current job role
  - Job roles performed in career
- Organisation
  - Name
  - Sector
  - Industry
  - Size (in terms of employees)

  - Age (how long the organisation exists)
- Awareness of cloaking techniques
  - Knowledge of a situation or fact
  - Perception of a situation or fact
- Preparedness to deal with drive-by download attacks using cloaking
  - Implemented defenses
  - Usage of strategies to investigate drive-by download attacks
  - Collection of threat intelligence

All three of the sources mentioned earlier indicate that for gathering information and determining awareness, open questions are the best option. As a result, open questions and matrices were used for this purpose. To measure the preparedness of organisations to deal with drive-by download attacks using cloaking, a combination of open questions and multiple choice questions was added. For perception of cloaking techniques, inspiration was gained from the structure of the quintamensional design mentioned by Payne [Pay51], we combined open knowledge questions, an attitude question, a reason-why question and an intensity question. To limit the influence of provided information, knowledge questions are asked first and participants can not go back to change their answers. The questions are also split into several pages so specific topics are mentioned one by one. In some cases, we deliberately forced a respondent to provide an answer other than *I don't know*. No questions related to the contextualisation of threat intelligence were asked as this does not directly relate to the research questions and we did not want to overload the respondents with too many questions. While we did do our best to use simple wording, the nature of the research questions is such that specific terms had to be used. Since the target group is cyber security professionals, we also preferred using well-known cyber industry lingo. A specific concern for the survey was the potential unwillingness of participants to share the name of their organisation and their personal details. While being able to group responses by organisation is of major importance for this research, we did not want to force respondents to provide this. For this reason, we added a specific question to ask respondents if they are willing to provide this information. If they indicated they were, the name was subsequently asked for. Survey results were anonymized otherwise, meaning it is impossible for us to link specific responses to individuals. The welcome page of the survey contains generic information about the research team and contact information. The final page thanks the respondents and repeats contact information. A *final comments* question was added to give respondents the possibility to mention comments or concerns about the subject of the survey or the survey itself. The survey questions were tested with two candidates and modified afterwards based on feedback. The final version of the survey can be found in Appendix D.

Potential survey respondents were contacted via different channels.

10.2 Results

10.2.1 Background of respondents & general statistics

35 fully completed surveys were collected from respondents from at least 6 different organisations. 15 respondents were unwilling to provide the name of their organisation. It is likely at least some of those belonged to different organisations than the 6 mentioned by others. For most of the 6 organisations, the amount of respondents is very low. This means it is not possible to draw conclusions per organisation in a statistically significant way. Potential respondents were contacted either directly by the author of this thesis, or by a contact of the author that was willing to ask other people working as cyber security specialist at the contact's organisation. The vast majority (96.67%) of respondents for which the organisation size was known, worked for an organisation with more than 250 employees. The same can be said for the age of the organisations, with (96.67%) of the respondents for which the organisation age was known working for an organisation that has existed for at least 10 years. As can be seen in Figure 10.1, the respondents were distributed among the public, semi-public, private and not-for-profit sectors fairly equally. With a slightly lower representation of the semi-public sector. Figure 10.2 shows that most of the respondents indicated they work in the *Finance & Insurance* industry. Different experience levels are represented as well, as can be seen in Figure 10.4. The bulk of the respondents were members of the blue team, as shown in Figure 10.3.

Sector	*Amount of respondents*	*% of respondents*
Public	9	31.03%
Semi-public	3	10.34%
Private	10	34.48%
Not-for-profit	7	24.14%

Table 10.1: Sector the respondents work in

Industry	*Amount of respondents*	*% of respondents*
Finance & Insurance	24	68.6%
Telecommunications	9	25.7%
Government & Public Administration	1	2.9%
Education	6	17.1%

Table 10.2: Industry the respondents work in

Role	*Amount of respondents*	*% of respondents*
cyber analyst (blue team)	22	62.86%
red team member	3	8.57%
IT security manager (middle management)	3	8.57%
chief information security officer (CISO)	1	2.86%
security system engineer	1	2.86%
security researcher/analyst	1	2.86%
cyber threat intelligence analyst	1	2.86%
security architect	1	2.86%
systems manager	1	2.86%

Table 10.3: Current job roles of the respondents

Experience	*Amount of respondents*	*% of respondents*
0-2 years	10	28.57%
3-5 years	10	28.57%
6-10 years	8	22.86%
>10 years	7	20.00%

Table 10.4: Years of experience in cyber security of the respondents

10.2.2 Awareness of cloaking (targeting techniques)

When asked about potential targeting techniques, the vast majority of respondents failed to mention any techniques that are used when performing cloaking on the web. This can be seen in 10.5. About 20% of the respondents do mention some sort of generic browser fingerprinting techniques and 11.43% mention IP based targeting. Most of the other techniques mentioned rely on other services: ads, e-mail, compromising a site that the targets tend to visit (waterhole attack). When asked *How should an analyst investigate a drive-by compromise attack to account for potential defense evasion techniques used during the drive-by compromise?*, none of the investigation techniques mentioned are specifically designed to detect cloaking, see Figure 10.6. Finally, only 8.57% of respondents heard or read about a successful drive-by compromise attack that was performed in the last two years, which used some sort detection evasion other than vulnerability detection (cloaking).

Based on all of the above results, we consider the awareness of cloaking (targeting techniques) to be very low.

Targeting technique	*Number of mentions*	*% of respondents*
Waterhole attack	11	31.43%
Generic browser fingerprinting	7	20.00%
Create a site that is likely to attract target audience	6	17.14%
Use ad (networks) and their targeting options	5	14.29%
IP based (verify connecting IP belongs to target)	4	11.43%
E-mail / phishing via e-mail	3	8.57%
Language or country	3	8.57%
Target specific vulnerabilities	1	2.86%

Table 10.5: Targeting techniques mentioned

Investigation technique for attack which potentially uses cloaking
Monitor log files (network, servers,. . .)
Forensic analysis of impacted system
Anomaly detection (network flows,...)

Table 10.6: Mentioned investigation techniques for attacks which potentially use cloaking

10.2.3 Perceived risk of malicious web content delivery attacks using cloaking techniques compared to attacks not using cloaking

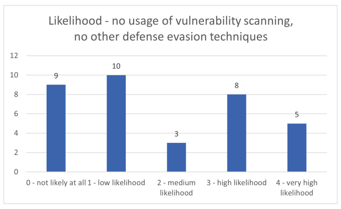
We asked survey respondents for an assessment of the likelihood and impact of a drive-by-target attack against their organization using specific techniques. This allows us to calculate a perceived risk score (likelihood x risk). The possible answers for each question ranged from 0 to 4. For likelihood the possible answers were *0 – not likely at all*, *1 – low likelihood*, *2 – medium likelihood*, *3 – high likelihood*, *4 – very high likelihood*. For impact the possible answers were *0 – no impact*, *1 – low impact*, *2 – medium impact*, *3 – high impact*, *4 – very high impact*. In the survey we made a distinction between *vulnerability scanning* for targeting and *other defense evasion techniques*. While both of these are targeting techniques for cloaking, we believe checking if the browser should have the specific vulnerability an exploit would work for, the bare minimum for an attacker to implement. Other defense evasion techniques in this case means mean any other targeting techniques used for cloaking such as targeting clients from certain countries.

As can be seen in Table 10.7, the perceived risk is higher for attacks using more advanced cloaking techniques (*usage of vulnerability scanning and other defense evasion techniques*). However, even for this case the risk is still lower than *high*, which would be 9 (3 x 3). Some respondents noted that it is hard for them to judge how likely such attacks are to happen, partly due to a lack of awareness around this topic. Others did not see a reason for a difference in impact based on the use of cloaking.

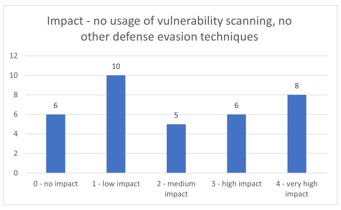
A more granular overview of the perceived likelihood, impact and risk, can be found in Figures 10.1, 10.2, 10.3 and 10.4. Unfortunately, due to the fact that many respondents did not give extensive motivations for their choices and the somewhat random distribution of the results, it is hard to draw specific conclusions based on these results.

	no other defense evasion	other defense evasion
no vuln. scanning	3.43	4.65
vuln. scanning	4.96	6.24

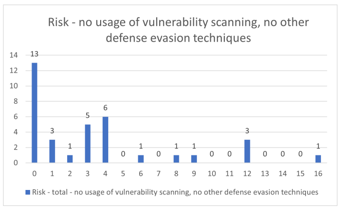
Table 10.7: Perceived risk



(a) likelihood

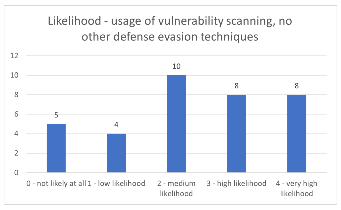


(b) impact

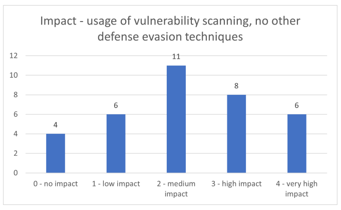


(c) risk

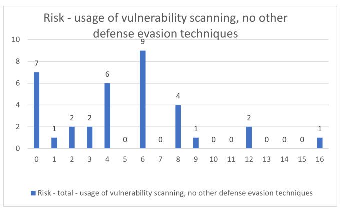
Figure 10.1: Perceived likelihood, impact and risk for drive-by-download attacks using no vulnerability scanning and no other defense evasion techniques



(a) likelihood



(b) impact



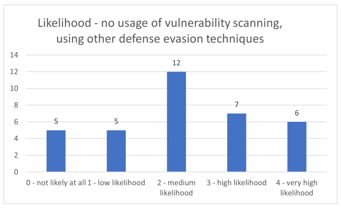
(c) risk

Figure 10.2: Perceived likelihood, impact and risk for drive-by-download attacks using vulnerability scanning and no other defense evasion techniques

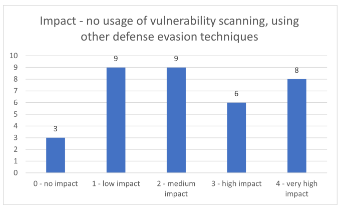
We believe this question should also be asked in future surveys. If the perceived risk is very low, the interest of organizations to implement controls against these techniques will be low as well. The outcome of our survey gives some indication that organizations would have some interest in this topic.

10.2.4 Preparedness for dealing with malicious web content delivery attacks using cloaking techniques

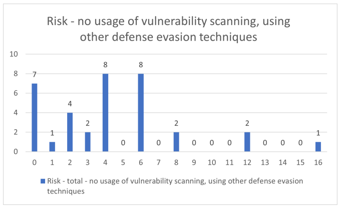
When asked for defenses against drive-by-compromise attacks, none of the respondents gave a response that can clearly be considered a defense against attacks using cloaking.



(a) likelihood

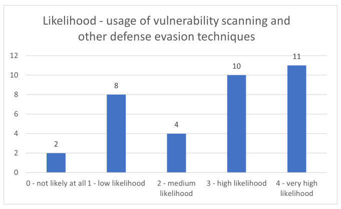


(b) impact

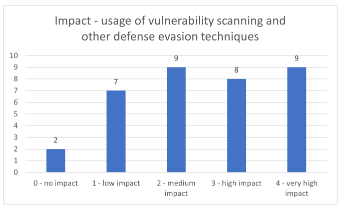


(c) risk

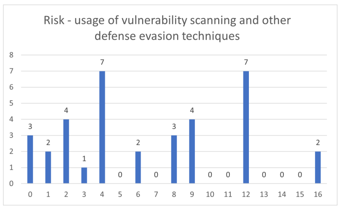
Figure 10.3: Perceived likelihood, impact and risk for drive-by-download attacks using no vulnerability scanning and other defense evasion techniques



(a) likelihood



(b) impact



(c) risk

Figure 10.4: Perceived likelihood, impact and risk for drive-by-download attacks using vulnerability scanning and other defense evasion techniques

That doesn't mean the mentioned defenses aren't effective of course, since the controls mentioned are designed to protect against generic drive-by-compromise attacks. The consolidated answers can be found in Figure 10.8. When asked about how a potential drive-by-compromise attack should be investigated, the respondents also mentioned some clear techniques to investigate potential drive-by-compromise attacks, see Figure 10.11. The majority of the respondents indicated their organisation implemented several of the defenses against drive-by-compromise we mentioned ourselves, see Figure 10.9. Additionally, the amount of respondents that indicated their organisation uses anti-browser-fingerprinting strategies is very low, as can be seen in Figure 10.10. Lastly, 18 out of the 35 respondents, or 51.43%, say they think their organisation should implement additional defensive strategies against drive-by-compromise attacks. This number is significantly lower for the organisation with the highest number of respondents however, at 36.71%.

Based on all of the above results, we consider the preparedness of organisations to deal with drive-by-compromise attacks to be considerable. There is still room for improvement however, especially when it comes to implementing anti-fingerprinting strategies. We note that browser isolation should be a fairly effective strategy to limit the potential impact of a compromise. Depending on the solution used, it can also make it harder to fingerprint individual individual users' browsers or even organisations' browsers, since the browsers can be made uniform and potentially be shared between organisations if browser isolation as a service is used.

Defense technique	*Number of mentions*	*% of respondents*
Use proxy server with filter and content analysis	12	34.29%
Browser isolation	11	31.43%
Fast patching	11	31.43%
Blacklist known bad urls/domains	9	25.71%
User awareness training	6	17.14%
Restrict web content (adblock, javascript)	5	14.29%
Use endpoint detection tools	5	14.29%
Use intrusion detection system	4	11.43%
Use antivirus software on endpoints	3	8.57%
Use threat intelligence	3	8.57%
Least/low privilege on devices	2	5.71%
Use threat hunting	1	2.86%
Apply zero trust policy / strategy	1	2.86%

Table 10.8: Mentioned potential defense techniques for drive-by-compromise attacks

Defense technique	*Number of mentions*	*% of respondents*
Browser isolation	22	62.86%
Antivirus on systems	30	85.71%
Regular patching of browser software	29	82.86%
Disabling javascript	7	20.00%
Web proxy with category based filter	29	82.86%
Web proxy with content analysis	21	60.00%

Table 10.9: Implemented defense techniques against drive-by-compromise attacks

Defense technique	*Number of mentions*	*% of respondents*
Uniform browser fingerprint	3	8.57%
Randomized browser fingerprint	4	11.43%
Anti-fingerprinting browser extension	3	8.57%

Table 10.10: Implemented defense techniques against fingerprinting

Investigation technique	*Number of mentions*	*% of respondents*
Investigate log data (network, proxy, Firewall)	14	40.00%
Perform forensic analysis of impacted system	7	20.00%
Analyze the malicious website	6	17.14%
IOC based search on network	2	5.71%
Anomaly detection in network traffic	2	5.71%
Use crawlers to analyze malicious website	2	5.71%
Determine scope of the compromise	1	2.86%
Isolate compromised system	1	2.86%

Table 10.11: Mentioned investigation techniques for drive-by-compromise attacks

10.2.5 MITRE ATT&CK® drive-by target

MITRE introduced the *Drive-by Target* technique to the ATT&CK® framework fairly recently (March 2021) [MITc]. We believe the naming of techniques is important and found the name of this technique confusing considering the description. We added a question to our survey to determine if our respondents would provide a definition that matches the one given by MITRE, when given the name *drive-by target*. Only 1 out of 35 respondents (that is, 2.86%), provided a definition that is remotely similar to that of MITRE ATT&CK. This strengthens our belief that the name or description of this technique should change in the future.

10.3 Discussion

Our survey results indicate that the awareness of cloaking techniques among cyber security professionals is fairly low. Nevertheless, the preparedness of the respondents' organisations is considerable. We would recommend organisations to look at specific defensive strategies like the ones mentioned in Appendix A, and to perform a risk analysis to determine if the effort required to implement them is worth it. Unfortunately, a lack of awareness and tooling support we discovered in our research, makes it harder to assess the risk, since it is not possible to easily create complete, consolidated reports of the prevalence of cloaking attacks on organisations.

Our survey has clear limitations. The amount of respondents is limited. For some organisations there are only a few respondents and almost half of the respondents were not willing to indicate which organisation they work for, making it impossible to have significant results grouped by organisation.

Chapter 11

Comparison of results and discussion

In this chapter, we compare the results of the prior two sections and discuss potential future research.

11.1 Comparison of results

The results of both the MISP community analysis and survey indicate that while there is certainly awareness of cloaking among cyber security professionals, the overall awareness of these techniques is fairly limited. Improvements could be made by providing user awareness training, sharing more information with regards to cloaking techniques in MISP haring communities and adding structured contextualisation options for cloaking related metadata in MISP.

The results with regards to RQ2-2 and RQ2-3 are complementary. While the survey results indicate that organisations are fairly well prepared to deal with malicious web pages, there is room for improvement there as well, especially when it comes to investigating these attacks and countering fingerprinting. Adding structured contextualisation options for cloaking related metadata in MISP could potentially help with that to some extent. Other blue team tools, such as browser isolation, can be implemented as well.

11.2 Discussion

Future research could include designing and performing a red team exercise involving a cloaking attack at a set of organisations. This would be a true test of the preparedness of those organisation and would lead to a more concrete assessment. Finding willing participants could be a hindrance however.

An extension of the survey would increase the validity of the research as well. One of the limitations is that we can not guarantee respondents did not fill in the survey multiple times or that they did not use external resources while answering the questions.

Performing the survey in an offline, exam type setting, would lead to more valid results. Those results could also be combined with in person interviews, to apply the principle of triangulation.

Part IV

Concerns for forensic analysis when investigating a potential compromise as a result of malicious web content delivery

Chapter 12

Analysis of differences in browser cache forensic artifacts on usage of Cache-Control: no-store header

Considering the arms race between attackers and defenders, thinking about countermeasures and analysis techniques also makes us reflect about ways in which attackers could react to those countermeasures. Our final research question handles other concerns with regards to investigating a potential compromise as a result of malicious web content delivery.

RQ3 What other techniques can attackers use to thwart analysis of malicious web content delivery attacks?

One of the actions that blue team members can take after a potential attack using cloaking, is performing forensic analysis of the device on which the browser was running. During usage of a device, traces of the activities performed on it are often left on disk and in memory. There are three particular artifacts of interest in this case. The browser history, cache and cookies. The browser history lists pages visited, the cache is used to store re-usable files and cookies are used to make HTTP connections stateful. The browser cache is intended to speed up the loading of web pages and to reduce bandwidth usage. If allowed, a browser stores a version of received content for potential future use. When the same content is requested again, the browser either serves the stored version, or requests a new version from the server, based on several rules. This presents an opportunity to investigators. If the malicious content was cached, it can be retrieved from the device and analyzed. An attacker can potentially counter this however. The proposed HTTP 1.1 caching standard, RFC 2616 [IET], includes the Cache-Control mechanism directive no-store. The purpose is explained in the RFC [IET] and can be found below.

"*The purpose of the no-store directive is to prevent the inadvertent release or retention of sensitive information (for example, on backup tapes). The no-store directive* *applies to the entire message, and MAY be sent either in a response or in a request. If sent in a request, a cache MUST NOT store any part of either this request or any response to it. If sent in a response, a cache MUST NOT store any part of either this response or the request that elicited it. This directive applies to both non- shared and shared caches. "MUST NOT store" in this context means that the cache MUST NOT intentionally store the information in non-volatile storage, and MUST make a best-effort attempt to remove the information from volatile storage as promptly as possible after forwarding it. Even when this directive is associated with a response, users might explicitly store such a response outside of the caching system (e.g., with a "Save As" dialog). History buffers MAY store such responses as part of their normal operation.*"

In effect, an attacker might consider adding the header Cache-Control: no-store header when delivering malicious content. If browsers follow the standard, this will prevent them from storing a local copy in the cache, complicating the work of a security analyst. In this part of the research, we present an analysis of browser cache forensic artifacts on usage of Cache-Control: no-store header, to determine if there are truly differences when the header is set. With this research, we contribute to answering RQ3, albeit to a fairly limited extent.

12.1 Research methodology

This section provides a more in depth overview of the research methods and data that were used to reach the final contributions as well as the design decisions that were made while developing this part of the research. For more information about which resources were used to create the below section, please refer to 6.2.

Possible research strategies for this part of the research are described below.

- **Survey** A survey is not a valid strategy for this part of the research.
- **Experiment** An experiment could be performed, delivering web content with different values set for the Cache-Control header, to determine potential differences in the resulting forensic artifacts.
- **Case study** A case study is not a valid strategy for this part of the research.
- **Desk research** To the best of our knowledge, there is no prior research on this topic. Browser documentation and standards documentation could be reviewed. Implementations do not always comply with specifications however, so they can not be used to determine how a browser behaves in practice.

We chose for the experiment strategy, as it seems the most suitable for this part of the research. Two separate files were set up to be delivered. The first is delivered with no Cache-Control header set. The second is delivered with Cache-Control: no-store.

For the client machine, we focus on the operating system with the biggest market share on desktop systems, which is Windows. The choice was made to focus on the four browsers with the biggest market share according to the latest statistics from NetMarketshare [net]. Those are Chrome, Edge, Firefox and Internet Explorer. We installed the latest available version of those browsers on a newly installed virtual machine running the latest version of Windows 10. We then created a snapshot of the virtual machine in the clean state. From this stage, we then visited the two respective URLs to fetch the web content with each browser. For each test, we started from the clean state, to make sure they did not influence each other. For each experiment, we created a snapshot after the test and collected a memory and disk image. We then performed forensic analysis using a copy of the forensic disk image.

The SANS 500 course material [LT20] was used to gain a solid understanding of browser forensic artifacts. Please note that the cited book mentions the Edge browser would become Chromium based in the future, but that was not the case yet when the book was written. As a result there was no specific information about the location of the current forensic artifact locations for the Edge browser.

12.2 Results

Screenshots of all core evidence can be found in Appendix C.

12.2.1 Chrome

Chrome is a Chromium based browser, developed by Google®. On Windows 10, all browser artifacts for Chrome can be found at the locations specified below.

- **Forensic artifacts root** %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default
- **History** %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\History
- **Cache** %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\Cache

The history database is stored in SQLite format. SQLite is a small and simple SQL database engine [Conb]. We used DB Browser for SQLite [Cona] to view the database entries. Entries of interest are the visited urls in the urls table. These show that our test urls were visited. The cache is stored in a different format. We used NirSoft's ChromeCacheView[1] to parse the cache database entries.

The evidence in Appendix C.2 shows that the urls were visited. There is an entry in the cache database for the file delivered without Cache-Control header set. There is no entry in the cache database for the file delivered with Cache-Control: no-store however.

[1]`https://www.nirsoft.net/utils/chrome_cache_view.html`

12.2.2 Edge

The edge browser is developed by Microsoft®. Since version 79, released in January 2020, the Edge browser is based on Chromium [Micb][Micc]. As a result, the forensic artifacts have the same structure as those of the Chrome browser. On Windows 10, they can be found at the locations below.

- **Forensic artifacts root** %USERPROFILE%\AppData\Local\Microsoft\Edge\User Data\Default
- **History** %USERPROFILE%\AppData\Local\Microsoft\Edge\User Data\Default\History
- **Cache** %USERPROFILE%\AppData\Local\Microsoft\Edge\User Data\Default\Cache

The evidence in Appendix C.3 shows that the urls were visited. There is an entry in the cache database for the file delivered without Cache-Control header set. Once again, there is no entry in the cache database for the file delivered with Cache-Control: no-store.

12.2.3 Firefox

Firefox is a free, open source browser maintained by the Mozilla Foundation [2]. Firefox has different browser artifacts than Chrome and Edge. places.sqlite contains the history of the browser. We used DB Browser for SQLite [Cona] to view the database entries. We also used Nirsoft's BrowsingHistoryView[3] for a different view of the data. Entries of interest are the visited urls in the moz_places table. These show that our test urls were visited. The cache is stored in a different format. We used NirSoft's MZCacheView[4] to parse the cache database entries. The forensic artifacts for Firefox can be found at the below locations.

- **Main forensic artifacts root (excluding cache)** %USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<random text>.default
- **History** %USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<random text>.default\places.sqlite
- **Cache** %USERPROFILE%\AppData\Local\Mozilla\Firefox\User Data\Default\Cache

The evidence in Appendix C.4 shows that the urls were visited. There is an entry in the cache database for the file delivered without Cache-Control header set. Again, there is no entry in the cache database for the file delivered with Cache-Control: no-store however.

[2]https://foundation.mozilla.org/en/who-we-are/

[3]https://www.nirsoft.net/utils/browsing_history_view.html

[4]https://www.nirsoft.net/utils/mozilla_cache_viewer.html

12.2.4 Internet Explorer 11 (IE11)

The Internet Explorer browser is developed by Microsoft®. Version is the latest and final version. Microsoft plans to stop support of Internet Explorer 11 on the 15th of June 2022. IE11 uses an Extensible Storage Engine database[5] to store metadata, such as cache metadata and browser history. Cached files are stored directly in one of two possible folders. The forensic artifacts for IE11 can be found at the below locations.

- **History** %USERPROFILE%\AppData\Local\Microsoft\Edge\User Data\Default\History
- **Cache location 1** %USERPROFILE%\AppData\Local\Microsoft\InetCache\IE
- **Cache location 2** %USERPROFILE%\AppData\Local\Microsoft\InetCache\Low\IE

The results for IE11 are the same as for the other browsers. The evidence in Appendix C.5 shows that the urls were visited. There is an entry in the cache database for the file delivered without Cache-Control header set. Again, there is no entry in the cache database for the file delivered with Cache-Control: no-store.

12.3 Discussion

In conclusion, all of the browsers investigated did not have any entries in the cache database for content delivered with cache-control: no-store header set, adhering to the standard of RFC 7234 [IET]. The file was not cached. The results are summarized in Figure 12.1.

Browser	*cache-control: no-store set*	*No cache-control header set*
Chrome 91.0.4472.124	No trace in cache	File cached
Edge 91.0.864.59	No trace in cache	File cached
Firefox version 89.0.2	No trace in cache	File cached
IE11 11.789.19041.0	No trace in cache	File cached

Table 12.1: Browser cache forensic analysis results

The results show that cyber criminals could set this header as an anti-forensic practice. Potential follow up research could be to analyze how often this header is set for malicious content in the wild. If this is often the case, it could be a good feature for potential machine learning models classifying malicious content. Crawlers searching for malicious web pages could also be configured to scrutinize content delivered with this header more than other content. The same logic could be applied to some security tools as well. An example of this might be browser isolation systems. According to SANS'

[5] `https://techcommunity.microsoft.com/t5/ask-the-directory-services-team/ese-deep-dive-part-1-the-anatomy-of-an-ese-database/ba-p/400496`

browser isolation buyer's guide [Bro20], one criteria for a browser isolation is the implementation of threat mitigation technologies, such as code analysis. Extra analysis could be triggered on certain types of files if they are delivered with the Cache-Control: no-store header set.

Part V

Final conclusions and reflection

Chapter 13

Answers to the research questions and final conclusions

Specific conclusions and recommendations were provided in the separate discussion sections of the different parts of this research. In this chapter, we provide our answers to the research questions.

13.1 Answers to the research questions

RQ1-1: How effective are publicly available security web crawlers, one of the tools used to counter malicious web content delivery, at bypassing cloaking techniques? As expected, high interaction honeyclients such as Cuckoo Sandbox seem to perform better when it comes to simulating human interaction. Overall, the effectiveness of publicly available security web crawlers at bypassing cloaking techniques is low. None of the tested crawlers managed to trigger the malicious payload for at least 50% of the tests.

RQ1-2: How could security web crawlers' efficacy at bypassing cloaking techniques be improved? Most of the tested crawlers lack support for newer technologies such as the notification API, or even the latest browser versions. This results in failing tests across the board. Implementing support for newer technologies or browser versions could improve results. We do note however that focusing on older browser versions makes sense, since they are more likely to have known vulnerabilities and thus are probably targeted more often by attackers. Having configurable parameters for certain browser attributes such as Referer header or cookies can also improve results.

Lookyloo could implement improved support for newer technologies such as the notification API. Thug could implement improved support for newer technologies such as the notification API or bootstrap v5. Another potential improvement would be the addition of a configuration parameter to provide cookies to use when connecting to the url to analyze.

Proposed improvements for VirusTotal are to clarify how the URL analysis work, trigger some content analysis for visited urls and if possible, include a screenshot of the visited url.

Finally, a first proposed improvement for Cuckoo Sandbox is to provide analysis parameters to set certain values such as the Referer header or User-Agent header. This could potentially be achieved by using extensions for the installed browsers and creating a custom analysis module. Another potential improvement is the implementation of support for newer operating system and browser versions. A final proposal for development is to implement more human interaction simulation and if possible, provide parameters for them. One example could be that instead of clicking cancel on a confirm prompt, the user of Cuckoo Sandbox has the option to configure Cuckoo to click ok.

RQ1: How effective are commonly used security web crawlers, one of the tools used to counter malicious web content delivery, at bypassing targeting techniques commonly used in cloaking and how could their efficacy in bypassing these techniques be improved? Overall, the effectiveness of publicly available security web crawlers at bypassing cloaking techniques is low. None of the tested crawlers managed to trigger the malicious payload for at least 50% of the tests. There are several potential improvements for each crawler. High interaction and low interaction honeyclients each have their own particular, expected, strengths and weaknesses. There are clear potential improvements for each of the researched security web crawlers. As most of these tools have a specific, specialized scope, it is not clear if these proposed improvements would be considered to be implemented.

RQ2-1: How aware are cyber security professionals and their organisations of cloaking techniques? Both our MISP community analysis and survey results showed that the overall awareness of cyber security professionals and their organisations of cloaking techniques is low.

RQ2-2: How prepared are organisations for dealing with malicious web content delivery attacks using cloaking techniques? While the survey results indicate that organisations are fairly well prepared to deal with malicious web pages, there is room for improvement there as well, especially when it comes to investigating these attacks and countering fingerprinting.

RQ2-3: What additional steps can organisations take to improve awareness and preparedness to deal with malicious web content delivery attacks using cloaking techniques? Adding structured contextualisation options for cloaking related metadata in MISP could potentially help with that to some extent. Other blue team tools, such as browser isolation, can be implemented as well. Security teams can also add specific analysis steps to cover cloaking. Finally, dedicated security awareness training on the topic of cloaking attacks can be given to employees.

RQ2: How aware and prepared are cyber security professionals and their organisations when it comes to cloaking attacks, and how can their awareness and preparedness be improved? The overall awareness of the topic is low. Preparedness on the other hand is considerable, mainly due to generic defensive measures that also cover cloaking attacks. Multiple improvements can be made, as discussed above.

RQ3: What other techniques can attackers use to thwart analysis of malicious web content delivery attacks? Attackers can use the Cache-Control: no-store header as an anti-forensic option.

13.2 Final conclusions

Our main goal with this research was to explore the subject of malicious web content delivery using cloaking. Our initial hypothesis that awareness of this topic amongst cyber security professionals was low, was confirmed by our research. We believe this situation should be remediated and hope that this research can help kickstart efforts in this regard. By discussing not only this problem, but also managerial, anti-forensic and tooling related concerns, we covered multiple areas that can be explored more in depth later on. Finally, the implementation of our demonstration cloaking website is an added tool to aid in spreading awareness of cloaking techniques and can be used during presentations and trainings.

Chapter 14

Reflection

In this research we explored the topics of cloaking and malicious content delivery via the web in breadth rather than in depth. Focusing on the concerns of organisations and cyber security professionals. As a result there is a lack of depth in some of the results, when comparing it to research with a more limited scope. While this was a conscious decision, it is still clearly a concern. Follow up research could explore each of the topics handled by the different parts of this research in more depth.

Aside from that, the total amount of respondents for the survey conducted as part of this research, 35, is relatively low. This is partly the result of the COVID-19 pandemic, which caused problems for our survey, since we planned to use networking at conferences to gather more volunteers. In the early phases of the pandemic, a lot of organisations' IT teams also seemed to be focused on making the necessary changes to allow employees to work from home in a secure way, making us reluctant to reach out to them at that time. Regardless, better results might have been achieved by more assertively reaching out to more organisations. To get more representative results, a larger group of employees should be included for each of the organisations. Informal interviews with some of the respondents also showed that doing in person interviews as well as questionnaires might lead to more complete results and allows for clarification of the questions when necessary. These are our key learnings when it comes to the survey process.

Out of ethical considerations, we did not explore certain potential research avenues. In particular, we decided not to use real malware on our cloaking website, even though this might have potentially triggered more VirusTotal hits. While we do mention some techniques that could be used by cyber criminals, we believe they are not game changers and publishing this research will not damage the community. We hope that this research will benefit the IT security community by increasing awareness of certain risks instead.

Finally, some life events and in particular COVID-19, resulted in a temporary slower pace of research. While this might be understandable, some practices could be used to prevent this to some extent. In particular, having more regular sync meetings with the research team and forming a study group with other students, might have aided in keeping up the pace.

Bibliography

[Aca+13] Gunes Acar et al. "FPDetective: dusting the web for fingerprinters". In: *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013.* Ed. by Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung. ACM, 2013, pp. 1129–1140. DOI: `10.1145/2508859.2516674`. URL: `https://doi.org/10.1145/2508859.2516674`.

[Aca+14] Gunes Acar et al. "The Web Never Forgets: Persistent Tracking Mechanisms in the Wild". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security.* CCS '14. Scottsdale, Arizona, USA: ACM, 2014, pp. 674–689. ISBN: 978-1-4503-2957-6. DOI: `10.1145/2660267.2660347`. URL: `http://doi.acm.org/10.1145/2660267.2660347`.

[Ahn+03] Luis von Ahn et al. "CAPTCHA: Using Hard AI Problems for Security". In: *Advances in Cryptology — EUROCRYPT 2003.* Ed. by Eli Biham. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 294–311. ISBN: 978-3-540-39200-2.

[Ask+13] Nicola Askham et al. *THE SIX PRIMARY DIMENSIONS FOR DATA QUALITY ASSESSMENT.* Tech. rep. DAMA United Kingdom, Oct. 2013. URL: `https://www.dama-uk.org/resources/Documents/DAMA%5C%20UK%5C%20DQ%5C%20Dimensions%5C%20White%5C%20Paper2020.pdf`.

[BBW17] Nicolas Van Balen, Christopher Ball, and Haining Wang. "Analysis of Targeted Mouse Movements for Gender Classification". In: *EAI Endorsed Transactions on Security and Safety* 4.11 (Dec. 2017). DOI: `10.4108/eai.7-12-2017.153395`.

[Bon+15] Joseph Bonneau et al. "Passwords and the Evolution of Imperfect Authentication". In: *Commun. ACM* 58.7 (June 2015), pp. 78–87. ISSN: 0001-0782. DOI: `10.1145/2699390`. URL: `https://doi.org/10.1145/2699390`.

[Bro20] Matt Bromiley. "All Roads Lead to the Browser. A SANS Buyer's Guide to Browser Isolation". In: (2020).

[Chra] Chromium. *Flash Roadmap.* (Accessed: 2021-07-20). URL: `https://www.chromium.org/flash-roadmap`.

[Chrb] Chromium. *The Final Countdown for NPAPI.* (Accessed: 2021-07-20). URL: https://blog.chromium.org/2014/11/the-final-countdown-for-npapi.html.

[Cona] SQLite Consortium. *DB Browser for SQLite.* (Accessed: 2021-07-04). URL: https://sqlitebrowser.org/.

[Conb] SQLite Consortium. *SQLite Home Page.* (Accessed: 2021-07-04). URL: https://www.sqlite.org/index.html.

[con] Django simple captcha contributors. *Django Simple Captcha.* (Accessed: 2021-07-20). URL: https://github.com/mbi/django-simple-captcha.

[Cro17] Penny Crosman. *Why JPMorgan, Amex, HSBC are backing 'isolation' web browsing.* English. Name - JPMorgan Chase & Co; American Express Co; Copyright - Copyright SourceMedia Dec 12, 2017; Last updated - 2020-11-20. Dec. 2017. URL: https://www.americanbanker.com/news/why-jpm-amex-hsbc-backing-isolation-web-browsing.

[DN10] Brendan Dolan-gavitt and Yacin Nadji. *See No Evil: Evasions in Honeymonkey Systems.* Tech. rep. 2010.

[Dol19] Rik Dolfing. "Research Internship: Towards a fingerprint surface". MA thesis. Radboud University, 2019, p. 21.

[DS14] Laurent Delosières and Antonio Sánchez. "DroidCollector: A Honeyclient for Collecting and Classifying Android Applications". In: *Information Sciences and Systems 2014 - Proceedings of the 29th International Symposium on Computer and Information Sciences, ISCIS 2014, Krakow, Poland, October 27-28, 2014.* 2014, pp. 175–183. DOI: 10.1007/978-3-319-09465-6_19. URL: https://doi.org/10.1007/978-3-319-09465-6%5C_19.

[Eck10] Peter Eckersley. "How Unique Is Your Web Browser?" In: *Privacy Enhancing Technologies, 10th International Symposium, PETS 2010, Berlin, Germany, July 21-23, 2010. Proceedings.* Ed. by Mikhail J. Atallah and Nicholas J. Hopper. Vol. 6205. Lecture Notes in Computer Science. Springer, 2010, pp. 1–18. DOI: 10.1007/978-3-642-14527-8_1. URL: https://doi.org/10.1007/978-3-642-14527-8%5C_1.

[EN16] Steven Englehardt and Arvind Narayanan. "Online Tracking: A 1-Million-Site Measurement and Analysis". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.* CCS '16. Vienna, Austria: Association for Computing Machinery, 2016, pp. 1388–1401. ISBN: 9781450341394. DOI: 10.1145/2976749.2978313. URL: https://doi.org/10.1145/2976749.2978313.

[fA] FingerprintJS Pro - Browser fingerprinting and fraud detection API. *Browser Fingerprinting API.* (Accessed: 2021-07-20). URL: https://fingerprintjs.com/.

[FDI] FDIC. *Statistics at a glance.* (Accessed: 2019-11-11). URL: `https://www.fdic.gov/bank/statistical/stats/2019jun/fdic.pdf`.

[FE15] David Fifield and Serge Egelman. "Fingerprinting Web Users Through Font Metrics". In: *Financial Cryptography and Data Security - 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers.* Ed. by Rainer Böhme and Tatsuaki Okamoto. Vol. 8975. Lecture Notes in Computer Science. Springer, 2015, pp. 107–124. DOI: `10.1007/978-3-662-47854-7\_7`. URL: `https://doi.org/10.1007/978-3-662-47854-7%5C_7`.

[FH18] Casey Fiesler and Blake Hallinan. ""We Are the Product": Public Reactions to Online Data Sharing and Privacy Controversies in the Media". eng. In: CHI '18. ACM, 2018, pp. 1–13. ISBN: 9781450356206.

[Fin] FingerprintJS. *FingerprintJS.* (Accessed: 2021-07-20). URL: `https://github.com/fingerprintjs/fingerprintjs`.

[Fyf+14] Emily R Fyfe et al. "Concreteness Fading in Mathematics and Science Instruction: a Systematic Review". eng. In: *Educational psychology review* 26.1 (2014), pp. 9–25. ISSN: 1040-726X.

[FZW15] Amin FaizKhademi, Mohammad Zulkernine, and Komminist Weldemariam. "FPGuard: Detection and Prevention of Browser Fingerprinting". In: *Data and Applications Security and Privacy XXIX.* Ed. by Pierangela Samarati. Cham: Springer International Publishing, 2015, pp. 293–308. ISBN: 978-3-319-20810-7.

[Ger] Eric Gerds. *Browser Plugin Detection with PluginDetect.* (Accessed: 2021-07-20). URL: `http://www.pinlady.net/PluginDetect/`.

[Góm+19] Alejandro Gómez-Boix et al. "A Collaborative Strategy for Mitigating Tracking through Browser Fingerprinting". In: *Proceedings of the 6th ACM Workshop on Moving Target Defense, MTD@CCS 2019, London, UK, November 11, 2019.* Ed. by Zhuo Lu. ACM, 2019, pp. 67–78. DOI: `10.1145/3338468.3356828`. URL: `https://doi.org/10.1145/3338468.3356828`.

[Goo] Google. *reCAPTCHA v2.* (Accessed: 2021-07-20). URL: `https://developers.google.com/recaptcha/docs/display`.

[Gro] Jeremiah Grossman. *I know where you've been.* (Accessed: 2021-07-20). URL: `https://blog.jeremiahgrossman.com/2006/08/i-know-where-youve-been.html`.

[Har17] Stevan Harnad. "Chapter 2 - To Cognize is to Categorize: Cognition is Categorization". eng. In: *Handbook of Categorization in Cognitive Science.* Second Edition. Elsevier Ltd, 2017, pp. 21–54. ISBN: 9780081011072.

[HL21] Ron S Hirschprung and Ori Leshman. "Privacy disclosure by de-anonymization using music preferences and selections". eng. In: *Telematics and informatics* 59 (2021), p. 101564. ISSN: 0736-5853.

[IET] Internet Engineering Task Force (IETF). *Hypertext Transfer Protocol (HTTP/1.1): Caching*. (Accessed: 2021-07-04). URL: `https://datatracker.ietf.org/doc/html/rfc7234`.

[IHF08] Ali Ikinci, Thorsten Holz, and Felix C. Freiling. "Monkey-Spider: Detecting Malicious Websites with Low-Interaction Honeyclients". In: *Sicherheit 2008: Sicherheit, Schutz und Zuverlässigkeit. Konferenzband der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI), 2.-4. April 2008 im Saarbrücker Schloss*. Ed. by Ammar Alkassar and Jörg H. Siekmann. Vol. 128. LNI. GI, 2008, pp. 407–421.

[JKV19] Hugo Jonker, Benjamin Krumnow, and Gabry Vlot. "Fingerprint surface-based detection of web bot detectors". In: *Proc. 24th European Symposium on Research in Computer Security Part II (ESORICS'19)*. Vol. 11736. LNCS. Springer, 2019, pp. 586–605. DOI: `10.1007/978-3-030-29962-0_28`.

[Kap+11] Alexandros Kapravelos et al. "Escape from Monkey Island: Evading High-Interaction Honeyclients". In: *Detection of Intrusions and Malware, and Vulnerability Assessment - 8th International Conference; DIMVA 2011, Amsterdam, The Netherlands, July 7-8, 2011. Proceedings*. Ed. by Thorsten Holz and Herbert Bos. Vol. 6739. Lecture Notes in Computer Science. Springer, 2011, pp. 124–143. DOI: `10.1007/978-3-642-22424-9\_8`. URL: `https://doi.org/10.1007/978-3-642-22424-9%5C_8`.

[Kap+19] Leon Kappelman et al. "A study of information systems issues, practices, and leadership in Europe". eng. In: *European journal of information systems* 28.1 (2019), pp. 26–42. ISSN: 0960-085X.

[Kim+12] HongGeun Kim et al. "Efficient Detection of Malicious Web Pages Using High-Interaction Client Honeypots". In: *J. Inf. Sci. Eng.* 28.5 (2012), pp. 911–924. URL: `http://www.iis.sinica.edu.tw/page/jise/2012/201209%5C_06.html`.

[Lap+20] Pierre Laperdrix et al. "Browser Fingerprinting: A Survey". eng. In: *ACM transactions on the web* 14.2 (2020), pp. 1–33. ISSN: 1559-1131.

[LBM17] Pierre Laperdrix, Benoit Baudry, and Vikas Mishra. "FPRandom: Randomizing Core Browser Objects to Break Advanced Device Fingerprinting Techniques". In: *Engineering Secure Software and Systems - 9th International Symposium, ESSoS 2017, Bonn, Germany, July 3-5, 2017, Proceedings*. Ed. by Eric Bodden, Mathias Payer, and Elias Athanasopoulos. Vol. 10379. Lecture Notes in Computer Science. Springer, 2017, pp. 97–114. DOI: `10.1007/978-3-319-62105-0\_7`. URL: `https://doi.org/10.1007/978-3-319-62105-0%5C_7`.

[Loo] Lookyloo. *Lookyloo*. (Accessed: 2021-06-28). URL: `https://www.lookyloo.eu/docs/main/index.html`.

[LRB15] Pierre Laperdrix, Walter Rudametkin, and Benoit Baudry. "Mitigating Browser Fingerprint Tracking: Multi-level Reconfiguration and Diversification". In: *10th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2015, Florence, Italy, May 18-19, 2015*. Ed. by Paola Inverardi and Bradley R. Schmerl. IEEE Computer Society, 2015, pp. 98–108. DOI: 10.1109/SEAMS.2015.18. URL: https://doi.org/10.1109/SEAMS.2015.18.

[LT20] Rob Lee and Chad Tilbury. *Windows Forensic Analysis V: Web Browser Forensics - Firefox, Internet Explorer, and Chrome*. Maryland: SANS Institute, 2020.

[Mica] Microsoft. *Netscape-Style Plug-ins Do Not Work After Upgrading Internet Explorer*. (Accessed: 2021-07-20). URL: https://web.archive.org/web/20071023233219/http://support.microsoft.com/kb/303401.

[Micb] Microsoft. *New year, new browser – The new Microsoft Edge is out of preview and now available for download*. (Accessed: 2021-07-04). URL: https://blogs.windows.com/windowsexperience/2020/01/15/new-year-new-browser-the-new-microsoft-edge-is-out-of-preview-and-now-available-for-download/.

[Micc] Microsoft. *Upgrading to the new Microsoft Edge*. (Accessed: 2021-07-04). URL: https://blogs.windows.com/msedgedev/2020/01/15/upgrading-new-microsoft-edge-79-chromium/.

[MISa] MISP. *community-metadata/defaults.json*. (Accessed: 2021-08-23). URL: https://github.com/MISP/MISP/blob/2.4/app/files/community-metadata/defaults.json.

[MISb] MISP. *MISP compliance iso concepts*. (Accessed: 2021-08-23). URL: https://github.com/MISP/misp-book/blob/main/sharing/figures/misp-compliance-iso-concepts.png.

[MITa] MITRE. *CAPEC-472: Browser Fingerprinting*. URL: https://capec.mitre.org/data/definitions/472.html (visited on 05/03/2021).

[MITb] MITRE. *MITRE ATT&CK*. URL: https://attack.mitre.org (visited on 05/03/2021).

[MITc] MITRE. *MITRE ATT&CK*. URL: https://attack.mitre.org/techniques/T1608/004/ (visited on 07/22/2021).

[Moza] Mozilla. *Firefox 88 combats window.name privacy abuses*. (Accessed: 2021-07-20). URL: https://blog.mozilla.org/security/2021/04/19/firefox-88-combats-window-name-privacy-abuses/.

[Mozb] Mozilla. *Plugging the CSS History Leak*. (Accessed: 2021-07-20). URL: https://blog.mozilla.org/security/2010/03/31/plugging-the-css-history-leak/.

[Mozc] Mozilla. *Plugin Roadmap for Firefox.* (Accessed: 2021-07-20). URL: https://developer.mozilla.org/en-US/docs/Plugins/Roadmap.

[Mozd] Mozilla. *Privacy and the :visited selector.* (Accessed: 2021-07-20). URL: https://developer.mozilla.org/en-US/docs/Web/CSS/Privacy_and_the_:visited_selector.

[Moze] Mozilla. *User-Agent.* (Accessed: 2021-07-18). URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent.

[Mozf] Mozilla. *Web APIs.* (Accessed: 2021-07-18). URL: https://developer.mozilla.org/en-US/docs/Web/API.

[Mozg] Mozilla. *Window: beforeunload event.* (Accessed: 2021-07-20). URL: https://developer.mozilla.org/en-US/docs/Web/API/Window/beforeunload_event.

[MR03] Arthur B Markman and Brian H Ross. "Category Use and Category Learning". eng. In: *Psychological bulletin* 129.4 (2003), pp. 592–613. ISSN: 0033-2909.

[MS12] Keaton Mowery and Hovav Shacham. "Pixel Perfect: Fingerprinting Canvas in HTML5". In: *Proceedings of W2SP 2012.* Ed. by Matt Fredrikson. IEEE Computer Society. May 2012.

[Mul+13] Martin Mulazzani et al. "Fast and Reliable Browser Identification with JavaScript Engine Fingerprinting". In: 2013.

[MWF14] Masood Mansoori, Ian Welch, and Qiang Fu. "YALIH, Yet Another Low Interaction Honeyclient". In: *Twelfth Australasian Information Security Conference, AISC 2014, Auckland, New Zealand, January 2014.* 2014, pp. 7–15. URL: http://crpit.com/abstracts/CRPITV149Mansoori.html.

[net] netmarketshare. *Browser Market Share.* (Accessed: 2021-07-04). URL: https://netmarketshare.com/browser-market-share.aspx.

[Nik+13] Nick Nikiforakis et al. "Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting". In: *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013.* IEEE Computer Society, 2013, pp. 541–555. DOI: 10.1109/SP.2013.43. URL: https://doi.org/10.1109/SP.2013.43.

[NJL15] Nick Nikiforakis, Wouter Joosen, and Benjamin Livshits. "PriVaricator: Deceiving Fingerprinters with Little White Lies". eng. In: WWW '15. ACM, 2015, pp. 820–830. ISBN: 1450334695.

[NVV18] Caroline Neckebroeck, Inge Vanderstraeten, and Mieke Verhaeghe. *Onderzoeksvaardigheden. Voor onderwijs, zorg en welzijn. Tweede editie.* Wommelgem: VAN IN, 2018.

[OGr] Stephen O'Grady. *The RedMonk Programming Language Rankings: January 2021.* (Accessed: 2021-07-17). URL: https://redmonk.com/sogrady/2021/03/01/language-rankings-1-21/.

[Ole+15] Lukasz Olejnik et al. “The Leaking Battery - A Privacy Analysis of the HTML5 Battery Status API”. In: *Data Privacy Management, and Security Assurance - 10th International Workshop, DPM 2015, and 4th International Workshop, QASA 2015, Vienna, Austria, September 21-22, 2015. Revised Selected Papers.* Ed. by Joaquín García-Alfaro et al. Vol. 9481. Lecture Notes in Computer Science. Springer, 2015, pp. 254–263. DOI: `10.1007/978-3-319-29883-2\_18`. URL: `https://doi.org/10.1007/978-3-319-29883-2%5C_18`.

[OS10] T. J. O’Connor and Benjamin Sangster. “honeyM: a framework for implementing virtual honeyclients for mobile devices”. In: *Proceedings of the Third ACM Conference on Wireless Network Security, WISEC 2010, Hoboken, New Jersey, USA, March 22-24, 2010.* Ed. by Susanne Wetzel, Cristina Nita-Rotaru, and Frank Stajano. ACM, 2010, pp. 129–138. DOI: `10.1145/1741866.1741888`. URL: `https://doi.org/10.1145/1741866.1741888`.

[Pay51] Stanley L Payne. *The art of asking questions.* Princeton: University Press, 1951.

[PB04] Maja Pusara and Carla Brodley. “User re-authentication via mouse movements”. eng. In: VizSEC/DMSEC ’04. ACM, 2004, pp. 1–8. ISBN: 1581139748.

[PCS16] Jin-Hak Park, Jang Won Choi, and Jung-Suk Song. “How to Design Practical Client Honeypots Based on Virtual Environment”. In: *11th Asia Joint Conference on Information Security, AsiaJCIS 2016, Fukuoka, Japan, August 4-5, 2016.* IEEE Computer Society, 2016, pp. 67–73. DOI: `10.1109/AsiaJCIS.2016.19`. URL: `https://doi.org/10.1109/AsiaJCIS.2016.19`.

[Por14] Rolf Porst. *Fragebogen. Ein Arbeitsbuch.* Fourth. Wiesbaden: Springer VS, 2014.

[Proa] MISP Project. *Event Report - A convenient mechanism to edit, visualize and share reports.* URL: `https://www.misp-project.org/2020/10/08/Event-Reports.html` (visited on 05/04/2021).

[Prob] MISP Project. *MISP User Stories.* URL: `https://github.com/MISP/misp-book/tree/main/user-stories` (visited on 08/23/2021).

[QH10] M. T. Qassrawi and Hongli Zhang. “Client honeypots Approaches and challenges”. In: *4th International Conference on New Trends in Information Science and Service Science.* May 2010, pp. 19–25.

[QZ11] Mahmoud T. Qassrawi and Hongli Zhang. “Detecting Malicious Web Servers with Honeyclients”. In: *JNW* 6.1 (2011), pp. 145–152. DOI: `10.4304/jnw.6.1.145-152`. URL: `https://doi.org/10.4304/jnw.6.1.145-152`.

[Run+09] Per Runeson et al. “Guidelines for conducting and reporting case study research in software engineering”. eng. In: *Empirical software engineering : an international journal* 14.2 (2009), pp. 131–164. ISSN: 1382-3256.

[Sai+16] Takamichi Saito et al. “Estimating CPU Features by Browser Fingerprinting”. eng. In: IEEE, 2016, pp. 587–592. ISBN: 9781509009848.

[Sana] Cuckoo Sandbox. *Cuckoo Sandbox.* (Accessed: 2021-06-28). URL: `https://cuckoosandbox.org/`.

[Sanb] Cuckoo Sandbox. *Cuckoo Sandbox.* (Accessed: 2021-06-28). URL: `https://github.com/cuckoosandbox/cuckoo`.

[SLG19] Michael Schwarz, Florian Lackner, and Daniel Gruss. “JavaScript Template Attacks: Automatically Inferring Host Information for Targeted Exploits”. In: *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019.* The Internet Society, 2019. URL: `https://www.ndss-symposium.org/ndss-paper/javascript-template-attacks-automatically-inferring-host-information-for-targeted-exploits/`.

[Smi+18] Michael Smith et al. “Browser history re:visited”. In: *12th USENIX Workshop on Offensive Technologies (WOOT 18).* Baltimore, MD: USENIX Association, Aug. 2018. URL: `https://www.usenix.org/conference/woot18/presentation/smith`.

[Swe] Sweetalert2. *Sweetalert2.* (Accessed: 2021-07-20). URL: `https://sweetalert2.github.io/`.

[Tel] Telenet. *Wat bepaalt de snelheid van je internetverbinding?* (Accessed: 2021-07-20). URL: `https://www2.telenet.be/nl/business/klantenservice/wat-bepaalt-de-snelheid-van-uw-internetverbinding/`.

[Thua] Thug. *https://thug-honeyclient.readthedocs.io/en/latest/usage.html.* (Accessed: 2021-07-22). URL: `https://thug-honeyclient.readthedocs.io/en/latest/`.

[Thub] Thug. *Thug code repository.* (Accessed: 2019-11-10). URL: `https://github.com/buffer/thug`.

[TJM15] Christof Ferreira Torres, Hugo L. Jonker, and Sjouke Mauw. “FP-Block: Usable Web Privacy by Controlling Browser Fingerprinting”. In: *Computer Security - ESORICS 2015 - 20th European Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceedings, Part II.* Ed. by Günther Pernul, Peter Y. A. Ryan, and Edgar R. Weippl. Vol. 9327. Lecture Notes in Computer Science. Springer, 2015, pp. 3–19. DOI: `10.1007/978-3-319-24177-7\_1`. URL: `https://doi.org/10.1007/978-3-319-24177-7%5C_1`.

[ULM15] R. Upathilake, Y. Li, and A. Matrawy. “A classification of web browser fingerprinting techniques”. In: *2015 7th International Conference on New Technologies, Mobility and Security (NTMS).* 2015, pp. 1–5. DOI: `10.1109/NTMS.2015.7266460`.

[VD10] Piet Verschuren and Hans Doorewaard. *Designing a Research Project. Second edition.* The Hague: Eleven International Publishing, 2010.

[Vira] VirusTotal. *How it works - VirusTotal.* (Accessed: 2021-07-18). URL: `https://support.virustotal.com/hc/en-us/articles/115002126889-How-it-works`.

[Virb] VirusTotal. *VirusTotal.* (Accessed: 2021-07-18). URL: `https://www.virustotal.com/`.

[Wan+05] Yi-Min Wang et al. *Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities.* Tech. rep. MSR-TR-2005-72. Aug. 2005, p. 12. URL: `https://www.microsoft.com/en-us/research/publication/automated-web-patrol-with-strider-honeymonkeys-finding-web-sites-that-exploit-browser-vulnerabilities/`.

[WG13] Tao Wang and Ian Goldberg. "Improved website fingerprinting on Tor". In: *Proceedings of the 12th annual ACM Workshop on Privacy in the Electronic Society, WPES 2013, Berlin, Germany, November 4, 2013.* Ed. by Ahmad-Reza Sadeghi and Sara Foresti. ACM, 2013, pp. 201–212. DOI: `10.1145/2517840.2517851`. URL: `https://doi.org/10.1145/2517840.2517851`.

[Zha+21] Penghui Zhang et al. "CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing". In: *Proceedings of the IEEE Symposium on Security and Privacy.* Best Student Paper Award. May 2021.

[Zho+14] Zhe Zhou et al. "Acoustic Fingerprinting Revisited: Generate Stable Device ID Stealthily with Inaudible Sound". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security.* CCS '14. Scottsdale, Arizona, USA: Association for Computing Machinery, 2014, pp. 429–440. ISBN: 9781450329576. DOI: `10.1145/2660267.2660300`. URL: `https://doi.org/10.1145/2660267.2660300`.

Appendices

Appendix A

Commonly known countermeasures against malicious web content delivery

The below countermeasures are based on prior academic research, professional experience of the research team and the results of the survey performed as part of this research.

Usage of a proxy server with filter and content analysis A forward proxy can be used for connections to the internet. Proxy servers can block certain content based on website reputation or categorization. A proxy can also perform content analysis (anti-virus scans), before potentially delivering the content to the client. These technologies block users from accessing certain content which is deemed dangerous or inappropriate.

Browser isolation See subsection 3.6.3.

Fast patching As software becomes older, more vulnerabilities are usually discovered. For active software projects, maintainers publish patches to fix those vulnerabilities. The longer a system remains unpatched, the higher the chances someone will exploit the vulnerability. As a result, patching often and keeping an eye on published vulnerabilities and exploits, is recommended.

Blacklisting of known bad urls and domains Threat intelligence communities such as the MISP communities mentioned in this research are sharing information with regards to known bad urls and domains. Organisations and users can use this type of information to block access to these urls, thus preventing a potential compromise.

User awareness training Users can be trained to look out for suspicious activity and potential attacks. The goal in this particular case is for users to avoid performing actions that could lead to a compromise, and to report any suspicious web pages they ended up visiting.

Restriction of allowed web content (usage of adblock, blocking JavaScript) Some attacks rely on advertisement networks or JavaScript. Using ad blockers or blocking JavaScript from executing can prevent such attacks from succeeding.

Usage of endpoint detection tools Endpoint detection tools allows monitoring of endpoints, and triggering of alerts and automated actions, based on rules to detect potentially suspicious activity.

Usage of intrusion detection systems Somewhat similar to the above, but often include monitoring network packets for potentially anomalous behaviour.

Usage of antivirus software on endpoints Antivirus software detects known malware, often based on a database of known malware signatures.

Usage of threat intelligence Threat intelligence, such as the information shared in the MISP communities we discussed as part of this research, can be used to assess risk, choose which protective measures to implement, and to automatically block or look for traces of known malicious behaviour.

Usage of least privilege principles on devices A mitigation strategy. If a component or user of the device is compromised, the attacker will have only those privileges the component or user had.

Threat hunting A proactive practice, which involves regularly looking for different types of potential compromises, which might not be detected or prevented by the currently implemented defenses.

Application of zero trust policy A policy that involves verifying any request, even those from historically vetted sources.

Vulnerability scanning Defenders can also scan their own systems for vulnerabilities, with the goal of fixing them as soon as possible after discovery.

Usage of security web crawlers Security web crawlers and in particular, honeyclients, can be used to crawl the web and search for malicious web pages. Due to their ability to act as a vulnerable browser, they should be more likely to trigger exploit delivery, even on malicious web pages using cloaking.

Appendix B

Relationship types used in the COVID-19 and MISPPRIV communities

Relationship type			
abuses	analysed-with	analyzed-with	child-of
connects-to	contained-within	contains	created
derived-from	drops	hosted-in	included-in
includes	opened	owner-of	part-of
read-from	references	related-to	same-as
targets			

Table B.1: Relationship types used in the COVID-19 MISP community

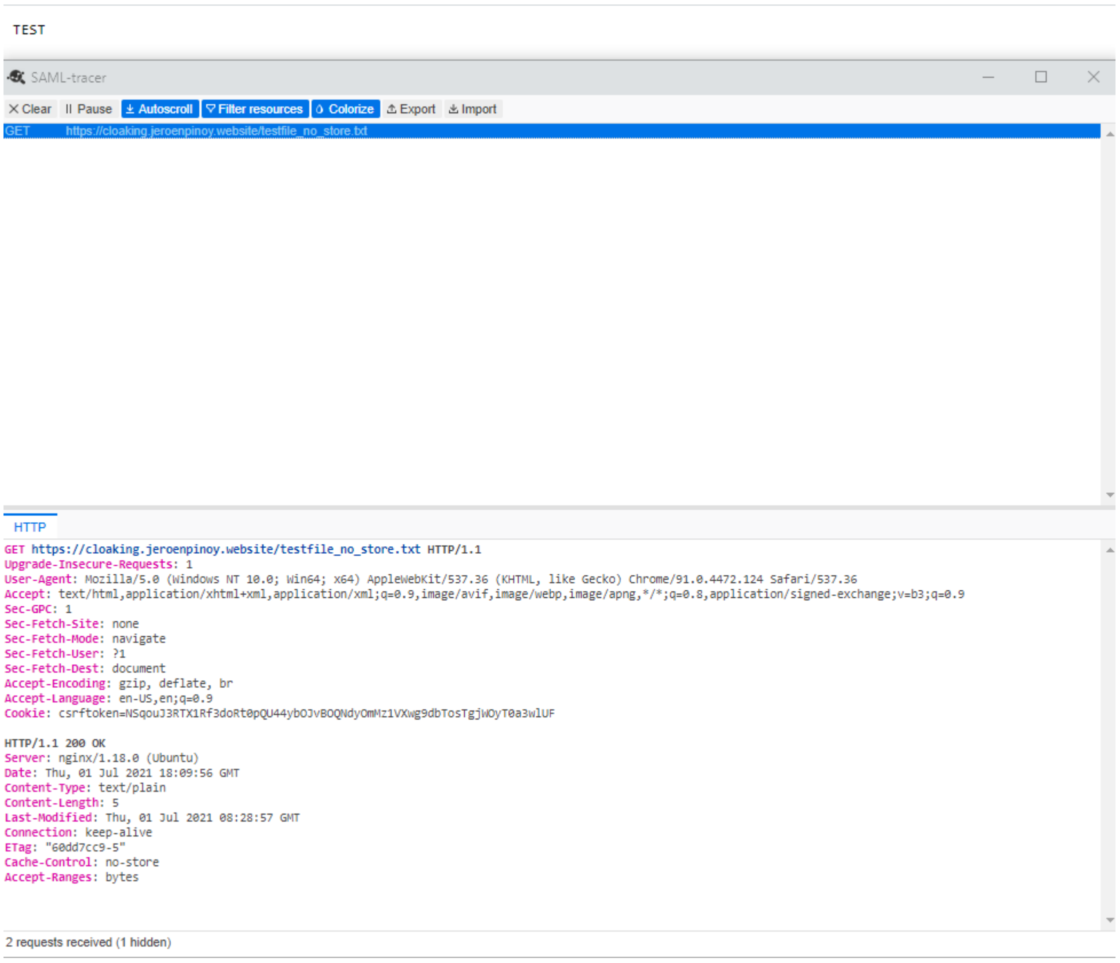
Relationship type			
abuses	analysed-with	analyzed-with	annotates
attributed-to	beacons-to	calls	captured-by
Characterized_By	characterized-by	child-of	communicates-with
connected-from	connected-to	connects-to	contained-by
contained-within	contains	cooperates-with	created
deleted	delivered-by	derived-from	detected-as
detected-with	downloaded	downloaded-by	downloaded-from
downloads	downloads-from	dropped	dropped-by
drops	executed-by	executes	exfiltrates-to
exploits	extracted-from	header-of	hosted-in
impersonates	included-in	includes	indicates
is-in-relation-with	linked-to	loaded-by	located
mentions	modified-properties-of	moved	moved-from
moved-to	opened	owner-of	part-of
read-from	received-from	redirects-to	references
related-to	relevant-to	rendered-as	renders-as
resolved-to	resumed	same-as	screenshot-of
sends	sends-to	signed-by	subdomain
targeted-by	targets	triggers	uploads
used-by	uses	variant-of	Vulnerability abused
weakened-by	writes	wrote-to	

Table B.2: Relationship types used in the MISPPRIV MISP community

Appendix C

Browser forensics evidence

C.1 Test file pages

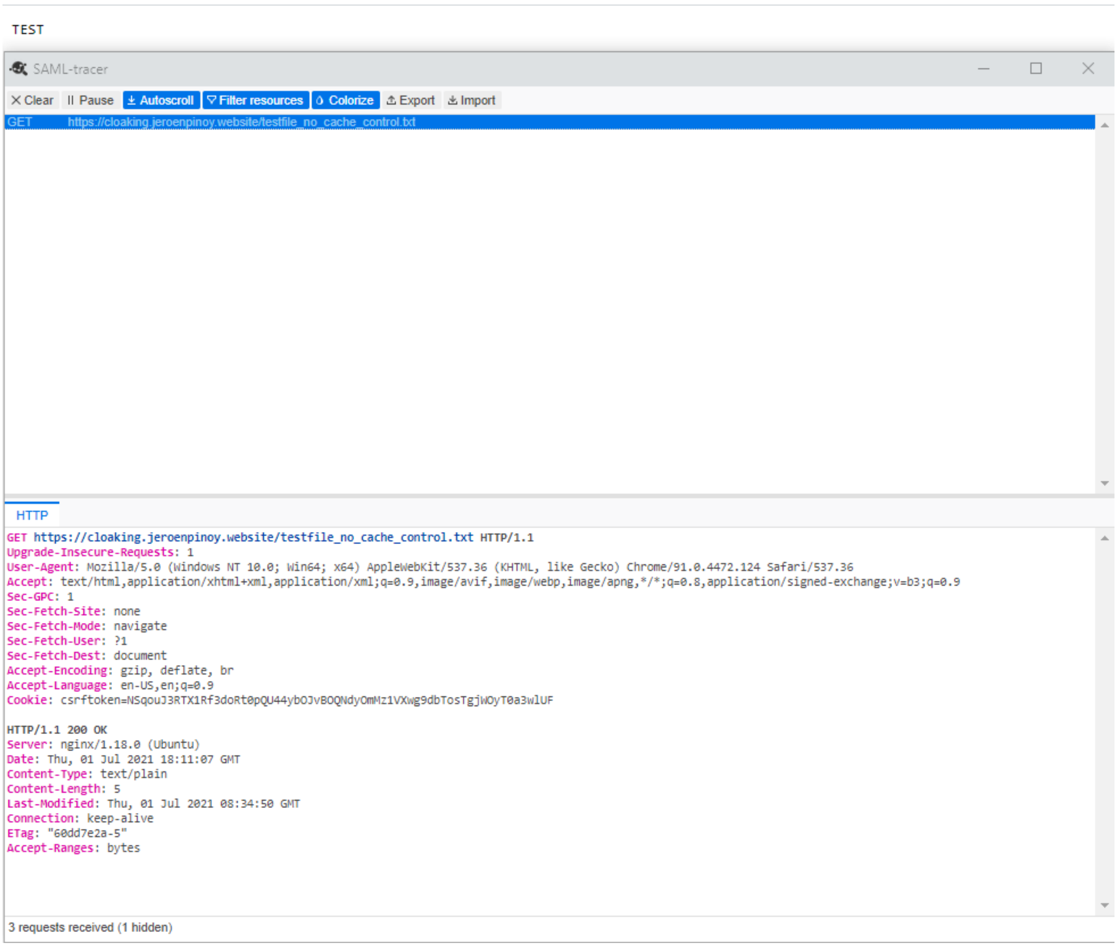


```
GET https://cloaking.jeroenpinoy.website/testfile_no_store.txt HTTP/1.1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-GPC: 1
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: csrftoken=NSqouJ3RTX1Rf3doRt0pQU44ybOJvBOQNdyOmMz1VXwg9dbTosTgjWOyT0a3wlUF

HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 01 Jul 2021 18:09:56 GMT
Content-Type: text/plain
Content-Length: 5
Last-Modified: Thu, 01 Jul 2021 08:28:57 GMT
Connection: keep-alive
ETag: "60dd7cc9-5"
Cache-Control: no-store
Accept-Ranges: bytes
```

2 requests received (1 hidden)

Figure C.1: Request and response for file delivered with Cache-Control: no-store



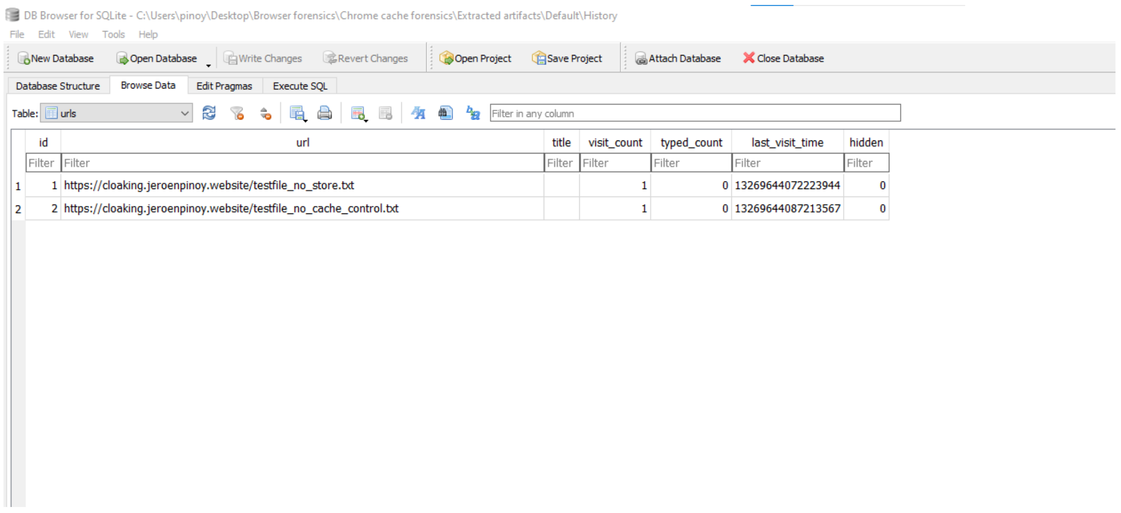
```
GET https://cloaking.jeroenpinoy.website/testfile_no_cache_control.txt HTTP/1.1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-GPC: 1
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: csrftoken=NSqouJ3RTX1Rf3doRt0pQU44ybOJvBOQNdyOmMz1VXwg9dbTosTgjWOyT0a3wlUF

HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 01 Jul 2021 18:11:07 GMT
Content-Type: text/plain
Content-Length: 5
Last-Modified: Thu, 01 Jul 2021 08:34:50 GMT
Connection: keep-alive
ETag: "60dd7e2a-5"
Accept-Ranges: bytes
```

3 requests received (1 hidden)

Figure C.2: Request and response for file delivered without Cache-Control

C.2 Chrome

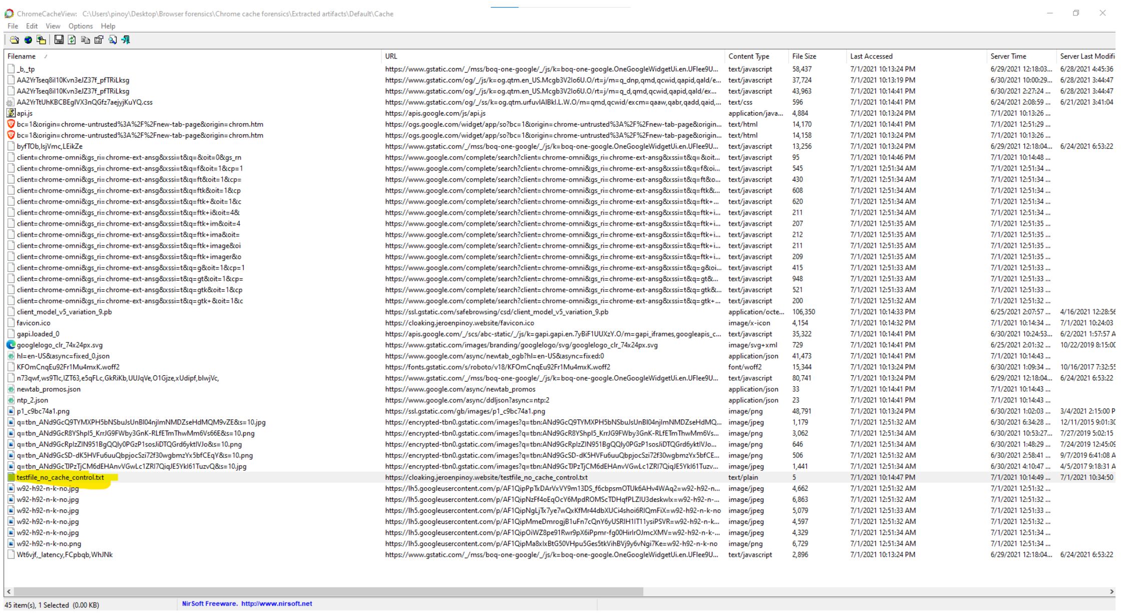


DB Browser for SQLite - C:\Users\pinoy\Desktop\Browser forensics\Chrome cache forensics\Extracted artifacts\Default\History

Table: urls

	id	url	title	visit_count	typed_count	last_visit_time	hidden
1	1	https://cloaking.jeroenpinoy.website/testfile_no_store.txt		1	0	13269644072223944	0
2	2	https://cloaking.jeroenpinoy.website/testfile_no_cache_control.txt		1	0	13269644087213567	0

Figure C.3: Chrome visited URLs history



ChromeCacheView: C:\Users\pinoy\Desktop\Browser forensics\Chrome cache forensics\Extracted artifacts\Default\Cache

Filename	URL	Content Type	File Size	Last Accessed	Server Time	Server Last Modifi
_b,_tp	https://www.gstatic.com/_/mss/boq-one-google/_/js/k=boq-one-google.OneGoogleWidgetUi.en.UFlee9U...	text/javascript	58,437	7/1/2021 10:13:24 PM	6/29/2021 12:18:03...	6/28/2021 4:45:36
AA2YrTseq8il10Kvn3eJZ37f_pfTRiLksg	https://www.gstatic.com/og/_/js/k=og.qtm.en_US.Mcgb3V2Io6U.O/rt=j/m=q_dnp,qmd,qcwid,qapid,qald/e...	text/javascript	37,724	7/1/2021 10:13:19 PM	6/30/2021 10:00:29...	6/28/2021 3:44:47
AA2YrTseq8il10Kvn3eJZ37f_pfTRiLksg	https://www.gstatic.com/og/_/js/k=og.qtm.en_US.Mcgb3V2Io6U.O/rt=j/m=q_d,qmd,qcwid,qapid,qald/ex...	text/javascript	43,963	7/1/2021 10:14:41 PM	6/30/2021 2:27:24 ...	6/28/2021 3:44:47
AA2YrTtUhKBCBEglVX3nQGfz7aejyjKuYQ.css	https://www.gstatic.com/og/_/ss/k=og.qtm.urfuvlAlBkl.L.W.O/m=qmd,qcwid/excm=qaaw,qabr,qadd,qaid,...	text/css	596	7/1/2021 10:14:41 PM	6/24/2021 2:08:59 ...	6/21/2021 3:41:04
api.js	https://apis.google.com/js/api.js	application/java...	4,884	7/1/2021 10:13:24 PM	7/1/2021 10:13:26 ...	
bc=1&origin=chrome-untrusted%3A%2F%2Fnew-tab-page&origin=chrom.htm	https://ogs.google.com/widget/app/so?bc=1&origin=chrome-untrusted%3A%2F%2Fnew-tab-page&origin...	text/html	14,170	7/1/2021 10:14:41 PM	7/1/2021 12:51:29 ...	
bc=1&origin=chrome-untrusted%3A%2F%2Fnew-tab-page&origin=chrom.htm	https://ogs.google.com/widget/app/so?bc=1&origin=chrome-untrusted%3A%2F%2Fnew-tab-page&origin...	text/html	14,158	7/1/2021 10:13:24 PM	7/1/2021 10:13:26 ...	
byfTOb,lsjVmc,LEikZe	https://www.gstatic.com/_/mss/boq-one-google/_/js/k=boq-one-google.OneGoogleWidgetUi.en.UFlee9U...	text/javascript	13,256	7/1/2021 10:13:24 PM	6/29/2021 12:18:04...	6/24/2021 6:53:22
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=&oit=0&gs_rn	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=&oit...	text/javascript	95	7/1/2021 10:14:46 PM	7/1/2021 10:14:48 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=f&oit=1&cp=1	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=f&oi...	text/javascript	545	7/1/2021 12:51:34 AM	7/1/2021 12:51:34 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ft&oit=1&cp=	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ft&o...	text/javascript	430	7/1/2021 12:51:34 AM	7/1/2021 12:51:34 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk&oit=1&cp	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk&...	text/javascript	608	7/1/2021 12:51:34 AM	7/1/2021 12:51:34 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk+&oit=1&c	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk+...	text/javascript	620	7/1/2021 12:51:34 AM	7/1/2021 12:51:34 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk+i&oit=4&	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk+i...	text/javascript	211	7/1/2021 12:51:34 AM	7/1/2021 12:51:34 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk+im&oit=4	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk+i...	text/javascript	207	7/1/2021 12:51:35 AM	7/1/2021 12:51:35 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk+ima&oit=	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk+i...	text/javascript	212	7/1/2021 12:51:35 AM	7/1/2021 12:51:35 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk+image&oi	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk+i...	text/javascript	211	7/1/2021 12:51:35 AM	7/1/2021 12:51:35 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk+imager&o	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=ftk+i...	text/javascript	209	7/1/2021 12:51:35 AM	7/1/2021 12:51:35 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=g&oit=1&cp=1	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=g&oi...	text/javascript	415	7/1/2021 12:51:33 AM	7/1/2021 12:51:33 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=gt&oit=1&cp=	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=gt&...	text/javascript	948	7/1/2021 12:51:33 AM	7/1/2021 12:51:33 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=gtk&oit=1&cp	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=gtk&...	text/javascript	521	7/1/2021 12:51:33 AM	7/1/2021 12:51:33 ...	
client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=gtk+&oit=1&c	https://www.google.com/complete/search?client=chrome-omni&gs_ri=chrome-ext-ansg&xssi=t&q=gtk+...	text/javascript	200	7/1/2021 12:51:32 AM	7/1/2021 12:51:32 ...	
client_model_v5_variation_9.pb	https://ssl.gstatic.com/safebrowsing/csd/client_model_v5_variation_9.pb	application/octe...	106,350	7/1/2021 10:14:33 PM	6/25/2021 2:07:57 ...	4/16/2021 12:28:5
favicon.ico	https://cloaking.jeroenpinoy.website/favicon.ico	image/x-icon	4,154	7/1/2021 10:14:32 PM	7/1/2021 10:14:34 ...	7/1/2021 10:24:03
gapi.loaded_0	https://apis.google.com/_/scs/abc-static/_/js/k=gapi.gapi.en.7yBiF1UUXzY.O/m=gapi_iframes,googleapis_c...	text/javascript	35,322	7/1/2021 10:14:41 PM	6/30/2021 10:24:53...	6/2/2021 1:57:57 A
googlelogo_clr_74x24px.svg	https://www.gstatic.com/images/branding/googlelogo/svg/googlelogo_clr_74x24px.svg	image/svg+xml	729	7/1/2021 10:14:41 PM	6/25/2021 2:01:32 ...	10/22/2019 8:15:0
hl=en-US&async=fixed_0.json	https://www.google.com/async/newtab_ogb?hl=en-US&async=fixed:0	application/json	41,473	7/1/2021 10:14:41 PM	7/1/2021 10:14:43 ...	
KFOmCnqEu92Fr1Mu4mxK.woff2	https://fonts.gstatic.com/s/roboto/v18/KFOmCnqEu92Fr1Mu4mxK.woff2	font/woff2	15,344	7/1/2021 10:13:24 PM	6/30/2021 1:09:34 ...	10/16/2017 7:32:5
n73qwf,ws9Tlc,IZT63,e5qFLc,GkRiKb,UUJqVe,O1Gjze,xUdipf,blwjVc,	https://www.gstatic.com/_/mss/boq-one-google/_/js/k=boq-one-google.OneGoogleWidgetUi.en.UFlee9U...	text/javascript	80,741	7/1/2021 10:13:24 PM	6/29/2021 12:18:04...	6/24/2021 6:53:22
newtab_promos.json	https://www.google.com/async/newtab_promos	application/json	33	7/1/2021 10:14:41 PM	7/1/2021 10:14:43 ...	
ntp_2.json	https://www.google.com/async/ddljson?async=ntp:2	application/json	23	7/1/2021 10:14:41 PM	7/1/2021 10:14:43 ...	
p1_c9bc74a1.png	https://ssl.gstatic.com/gb/images/p1_c9bc74a1.png	image/png	48,791	7/1/2021 10:13:24 PM	6/30/2021 1:02:03 ...	3/4/2021 2:15:00 P
q=tbn_ANd9GcQ9TYMXPH5bNSbuJsUnBl04njlmNMDZseHdMQM9vZE&s=10.jpg	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQ9TYMXPH5bNSbuJsUnBl04njlmNMDZseHdMQ...	image/jpeg	1,179	7/1/2021 12:51:32 AM	6/30/2021 6:34:28 ...	12/11/2015 9:01:3
q=tbn_ANd9GcR8YShpl5_KrrJG9FWby3GnK-RLfETmThwMm6Vs66E&s=10.png	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR8YShpl5_KrrJG9FWby3GnK-RLfETmThwMm6Vs...	image/png	3,062	7/1/2021 12:51:34 AM	6/30/2021 10:53:27...	7/27/2019 5:02:15
q=tbn_ANd9GcRplzZIN951BgQQly0PGzP1sosJiDTQGrd6yktlVJo&s=10.png	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRplzZIN951BgQQly0PGzP1sosJiDTQGrd6yktlVJo...	image/png	646	7/1/2021 12:51:34 AM	6/30/2021 1:48:29 ...	7/24/2019 12:45:0
q=tbn_ANd9GcSD-dK5HVFu6uuQbpjocSzi72f30wgbmzYx5bfCEqY&s=10.png	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSD-dK5HVFu6uuQbpjocSzi72f30wgbmzYx5bfCE...	image/png	506	7/1/2021 12:51:32 AM	6/30/2021 2:58:41 ...	9/7/2019 6:41:08 A
q=tbn_ANd9GcTJPzTjCM6dEHAnvVGwLc1ZRl7QiqJE5Ykl61TuzvQ&s=10.jpg	https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTJPzTjCM6dEHAnvVGwLc1ZRl7QiqJE5Ykl61Tuzv...	image/jpeg	1,441	7/1/2021 12:51:34 AM	6/30/2021 4:10:47 ...	4/5/2017 9:18:31 A
testfile_no_cache_control.txt	https://cloaking.jeroenpinoy.website/testfile_no_cache_control.txt	text/plain	5	7/1/2021 10:14:47 PM	7/1/2021 10:14:49 ...	7/1/2021 10:34:50
w92-h92-n-k-no.jpg	https://lh5.googleusercontent.com/p/AF1QipPpTxDArVxVY9m13DS_f6cbpsmOTUk6AHv4WAq2=w92-h92-n...	image/jpeg	4,662	7/1/2021 12:51:32 AM	7/1/2021 12:51:32 ...	
w92-h92-n-k-no.jpg	https://lh5.googleusercontent.com/p/AF1QipNzFf4oEqOcY6MpdROMScTDHqfPLZlU3deskwlx=w92-h92-n-...	image/jpeg	6,863	7/1/2021 12:51:32 AM	7/1/2021 12:51:32 ...	
w92-h92-n-k-no.jpg	https://lh5.googleusercontent.com/p/AF1QipNgLjTx7ye7wQxKfMr44dbXUCi4shoi6RlQmFiX=w92-h92-n-k-no	image/jpeg	5,079	7/1/2021 12:51:33 AM	7/1/2021 12:51:33 ...	
w92-h92-n-k-no.jpg	https://lh5.googleusercontent.com/p/AF1QipMmeDmrogjB1uFn7cQnY6yUSRlH1IT11ysiPSVR=w92-h92-n-k...	image/jpeg	4,597	7/1/2021 12:51:32 AM	7/1/2021 12:51:32 ...	
w92-h92-n-k-no.jpg	https://lh5.googleusercontent.com/p/AF1QipOiWZ8pe91Rwr9pX6iPpmr-fg00HirlrOJmcXMV=w92-h92-n-k-...	image/jpeg	4,329	7/1/2021 12:51:34 AM	7/1/2021 12:51:34 ...	
w92-h92-n-k-no.png	https://lh5.googleusercontent.com/p/AF1QipMa8xlxBtG50VHpu5Ges5tkVihBVj9y6vNgi7Ke=w92-h92-n-k-no	image/png	6,729	7/1/2021 12:51:34 AM	7/1/2021 12:51:34 ...	
Wt6vjf,_latency,FCpbqb,WhJNk	https://www.gstatic.com/_/mss/boq-one-google/_/js/k=boq-one-google.OneGoogleWidgetUi.en.UFlee9U...	text/javascript	2,896	7/1/2021 10:13:24 PM	6/29/2021 12:18:04...	6/24/2021 6:53:22

45 item(s), 1 Selected (0.00 KB)

Figure C.4: Chrome cache entries

C.3 Edge

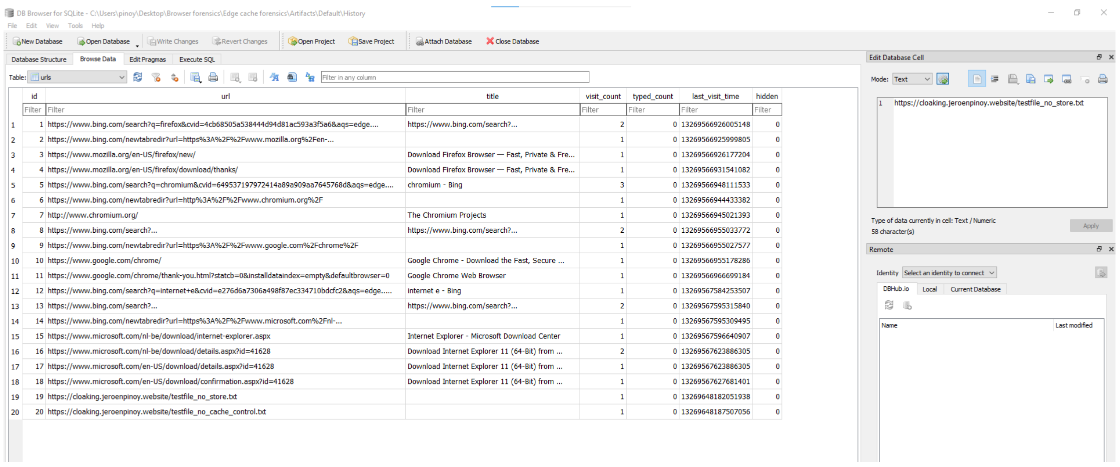

DB Browser for SQLite - C:\Users\pinoy\Desktop\Browser forensics\Edge cache forensics\Artifacts\Default\History

Table: urls

	id	url	title	visit_count	typed_count	last_visit_time	hidden
1	1	https://www.bing.com/search?q=firefox&cvid=4cb68505a538444d94d81ac593a3f5a6&aqs=edge....	https://www.bing.com/search?...	2	0	13269566926005148	0
2	2	https://www.bing.com/newtabredir?url=https%3A%2F%2Fwww.mozilla.org%2Fen-...		1	0	13269566925999805	0
3	3	https://www.mozilla.org/en-US/firefox/new/	Download Firefox Browser — Fast, Private & Fre...	1	0	13269566926177204	0
4	4	https://www.mozilla.org/en-US/firefox/download/thanks/	Download Firefox Browser — Fast, Private & Fre...	1	0	13269566931541082	0
5	5	https://www.bing.com/search?q=chromium&cvid=649537197972414a89a909aa7645768d&aqs=edge....	chromium - Bing	3	0	13269566948111533	0
6	6	https://www.bing.com/newtabredir?url=http%3A%2F%2Fwww.chromium.org%2F		1	0	13269566944433382	0
7	7	http://www.chromium.org/	The Chromium Projects	1	0	13269566945021393	0
8	8	https://www.bing.com/search?...	https://www.bing.com/search?...	2	0	13269566955033772	0
9	9	https://www.bing.com/newtabredir?url=https%3A%2F%2Fwww.google.com%2Fchrome%2F		1	0	13269566955027577	0
10	10	https://www.google.com/chrome/	Google Chrome - Download the Fast, Secure ...	1	0	13269566955178286	0
11	11	https://www.google.com/chrome/thank-you.html?statcb=0&installdataindex=empty&defaultbrowser=0	Google Chrome Web Browser	1	0	13269566966699184	0
12	12	https://www.bing.com/search?q=internet+e&cvid=e276d6a7306a498f87ec334710bdcfc2&aqs=edge.....	internet e - Bing	1	0	13269567584253507	0
13	13	https://www.bing.com/search?...	https://www.bing.com/search?...	2	0	13269567595315840	0
14	14	https://www.bing.com/newtabredir?url=https%3A%2F%2Fwww.microsoft.com%2Fnl-...		1	0	13269567595309495	0
15	15	https://www.microsoft.com/nl-be/download/internet-explorer.aspx	Internet Explorer - Microsoft Download Center	1	0	13269567596640907	0
16	16	https://www.microsoft.com/nl-be/download/details.aspx?id=41628	Download Internet Explorer 11 (64-Bit) from ...	2	0	13269567623886305	0
17	17	https://www.microsoft.com/en-US/download/details.aspx?id=41628	Download Internet Explorer 11 (64-Bit) from ...	1	0	13269567623886305	0
18	18	https://www.microsoft.com/en-US/download/confirmation.aspx?id=41628	Download Internet Explorer 11 (64-Bit) from ...	1	0	13269567627681401	0
19	19	https://cloaking.jeroenpinoy.website/testfile_no_store.txt		1	0	13269648182051938	0
20	20	https://cloaking.jeroenpinoy.website/testfile_no_cache_control.txt		1	0	13269648187507056	0

Edit Database Cell

Mode: Text

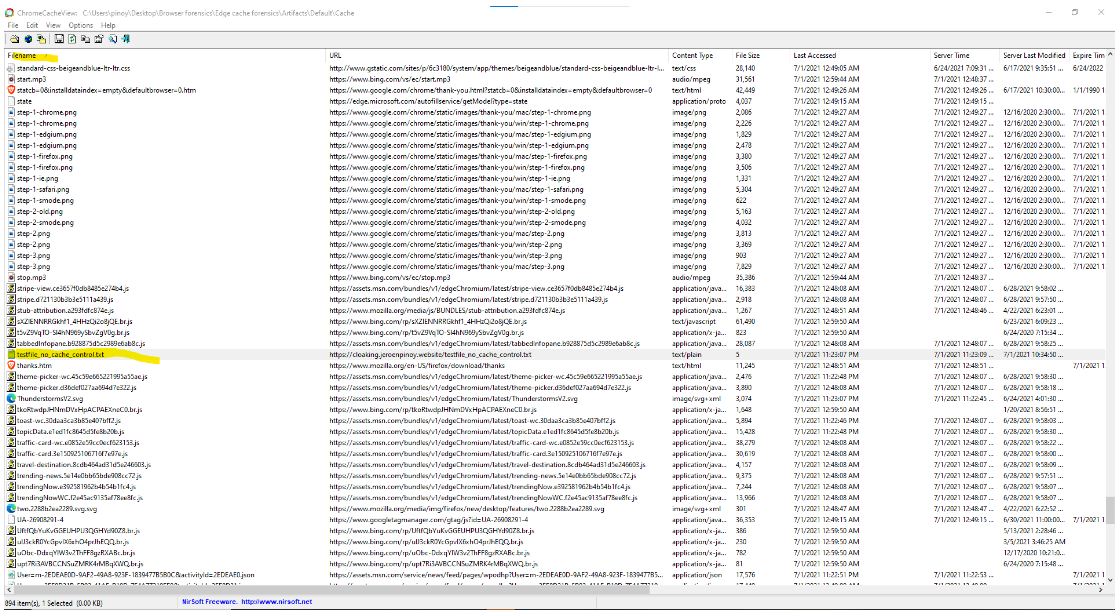
1 https://cloaking.jeroenpinoy.website/testfile_no_store.txt

Type of data currently in cell: Text / Numeric
58 character(s)

Remote

Identity: Select an identity to connect

Figure C.5: Edge visited URLs history

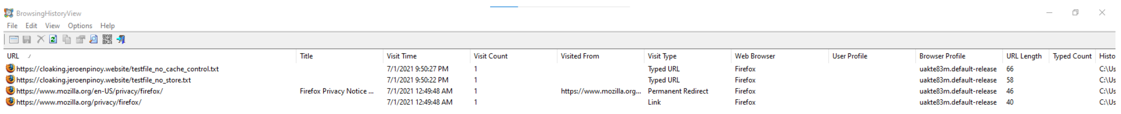

ChromeCacheView: C:\Users\pinoy\Desktop\Browser forensics\Edge cache forensics\Artifacts\Default\Cache

Filename	URL	Content Type	File Size	Last Accessed	Server Time	Server Last Modified	Expire Tim
standard-css-beigeandblue-ltr-ltr.css	http://www.gstatic.com/sites/p/6c3180/system/app/themes/beigeandblue/standard-css-beigeandblue-ltr-l...	text/css	28,140	7/1/2021 12:49:05 AM	6/24/2021 7:09:31 ...	6/17/2021 9:35:51 ...	6/24/2022
start.mp3	https://www.bing.com/vs/ec/start.mp3	audio/mpeg	31,561	7/1/2021 12:59:44 AM	7/1/2021 12:48:37 ...		
statcb=0&installdataindex=empty&defaultbrowser=0.htm	https://www.google.com/chrome/thank-you.html?statcb=0&installdataindex=empty&defaultbrowser=0	text/html	42,449	7/1/2021 12:49:26 AM	7/1/2021 12:49:26 ...	6/17/2021 10:30:00...	1/1/1990 1
state	https://edge.microsoft.com/autofillservice/getModel?type=state	application/proto	4,037	7/1/2021 12:49:15 AM	7/1/2021 12:49:15 ...		
step-1-chrome.png	https://www.google.com/chrome/static/images/thank-you/mac/step-1-chrome.png	image/png	2,086	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-1-chrome.png	https://www.google.com/chrome/static/images/thank-you/win/step-1-chrome.png	image/png	2,226	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-1-edgium.png	https://www.google.com/chrome/static/images/thank-you/mac/step-1-edgium.png	image/png	1,829	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-1-edgium.png	https://www.google.com/chrome/static/images/thank-you/win/step-1-edgium.png	image/png	2,478	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-1-firefox.png	https://www.google.com/chrome/static/images/thank-you/mac/step-1-firefox.png	image/png	3,380	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-1-firefox.png	https://www.google.com/chrome/static/images/thank-you/win/step-1-firefox.png	image/png	3,506	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-1-ie.png	https://www.google.com/chrome/static/images/thank-you/win/step-1-ie.png	image/png	1,331	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-1-safari.png	https://www.google.com/chrome/static/images/thank-you/mac/step-1-safari.png	image/png	5,304	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-1-smode.png	https://www.google.com/chrome/static/images/thank-you/win/step-1-smode.png	image/png	622	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-2-old.png	https://www.google.com/chrome/static/images/thank-you/win/step-2-old.png	image/png	5,163	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-2-smode.png	https://www.google.com/chrome/static/images/thank-you/win/step-2-smode.png	image/png	4,032	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-2.png	https://www.google.com/chrome/static/images/thank-you/mac/step-2.png	image/png	3,813	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-2.png	https://www.google.com/chrome/static/images/thank-you/win/step-2.png	image/png	3,369	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-3.png	https://www.google.com/chrome/static/images/thank-you/win/step-3.png	image/png	903	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
step-3.png	https://www.google.com/chrome/static/images/thank-you/mac/step-3.png	image/png	7,829	7/1/2021 12:49:27 AM	7/1/2021 12:49:27 ...	12/16/2020 2:30:00...	7/1/2021 1
stop.mp3	https://www.bing.com/vs/ec/stop.mp3	audio/mpeg	35,386	7/1/2021 12:59:44 AM	7/1/2021 12:48:37 ...		
stripe-view.ce3657f0db8485e274b4.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/stripe-view.ce3657f0db8485e274b4.js	application/java...	16,383	7/1/2021 12:48:08 AM	7/1/2021 12:48:07 ...	6/28/2021 9:58:02 ...	
stripe.d721130b3b3e5111a439.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/stripe.d721130b3b3e5111a439.js	application/java...	2,918	7/1/2021 12:48:08 AM	7/1/2021 12:48:07 ...	6/28/2021 9:57:50 ...	
stub-attribution.a293fdfc874e.js	https://www.mozilla.org/media/js/BUNDLES/stub-attribution.a293fdfc874e.js	application/java...	1,267	7/1/2021 12:48:51 AM	7/1/2021 12:48:46 ...	4/22/2021 6:23:01 ...	
sXZIENNRRGkhf1_4HHzQi2o8jQE.br.js	https://www.bing.com/rp/sXZIENNRRGkhf1_4HHzQi2o8jQE.br.js	text/javascript	61,490	7/1/2021 12:59:50 AM		6/23/2021 6:09:23 ...	
t5vZ9VqTO-SI4hN969ySbvZgV0g.br.js	https://www.bing.com/rp/t5vZ9VqTO-SI4hN969ySbvZgV0g.br.js	application/x-ja...	823	7/1/2021 12:59:50 AM		6/24/2020 7:15:34 ...	
tabbedInfopane.b928875d5c2989e6ab8c.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/tabbedInfopane.b928875d5c2989e6ab8c.js	application/java...	28,087	7/1/2021 12:48:08 AM	7/1/2021 12:48:07 ...	6/28/2021 9:58:25 ...	
testfile_no_cache_control.txt	https://cloaking.jeroenpinoy.website/testfile_no_cache_control.txt	text/plain	5	7/1/2021 11:23:07 PM	7/1/2021 11:23:09 ...	7/1/2021 10:34:50 ...	
thanks.htm	https://www.mozilla.org/en-US/firefox/download/thanks	text/html	11,245	7/1/2021 12:48:51 AM	7/1/2021 12:48:51 ...		7/1/2021 1
theme-picker-wc.45c59e665221995a55ae.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/theme-picker-wc.45c59e665221995a55ae.js	application/java...	2,476	7/1/2021 11:22:48 PM	7/1/2021 12:48:07 ...	6/28/2021 9:58:30 ...	
theme-picker.d36def027aa694d7e322.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/theme-picker.d36def027aa694d7e322.js	application/java...	3,890	7/1/2021 12:48:08 AM	7/1/2021 12:48:07 ...	6/28/2021 9:58:18 ...	
ThunderstormsV2.svg	https://assets.msn.com/bundles/v1/edgeChromium/latest/ThunderstormsV2.svg	image/svg+xml	3,074	7/1/2021 11:23:07 PM	7/1/2021 11:22:45 ...	6/24/2021 4:01:30 ...	
tkoRtwdpJHNmDVxHpACPAEXneC0.br.js	https://www.bing.com/rp/tkoRtwdpJHNmDVxHpACPAEXneC0.br.js	application/x-ja...	1,648	7/1/2021 12:59:50 AM		1/20/2021 8:56:51 ...	
toast-wc.30daa3ca3b85e407bff2.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/toast-wc.30daa3ca3b85e407bff2.js	application/java...	5,894	7/1/2021 11:22:46 PM	7/1/2021 12:48:07 ...	6/28/2021 9:58:03 ...	
topicData.e1ed1fc8645d5fe8b20b.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/topicData.e1ed1fc8645d5fe8b20b.js	application/java...	15,428	7/1/2021 11:22:48 PM	7/1/2021 12:48:07 ...	6/28/2021 9:58:30 ...	
traffic-card-wc.e0852e59cc0ecf623153.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/traffic-card-wc.e0852e59cc0ecf623153.js	application/java...	38,279	7/1/2021 12:48:08 AM	7/1/2021 12:48:07 ...	6/28/2021 9:58:22 ...	
traffic-card.3e150925106716f7e97e.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/traffic-card.3e150925106716f7e97e.js	application/java...	30,619	7/1/2021 12:48:08 AM	7/1/2021 12:48:07 ...	6/28/2021 9:58:00 ...	
travel-destination.8cdb464ad31d5e246603.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/travel-destination.8cdb464ad31d5e246603.js	application/java...	4,157	7/1/2021 12:48:08 AM	7/1/2021 12:48:07 ...	6/28/2021 9:58:09 ...	
trending-news.5e14e0bb65bde908cc72.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/trending-news.5e14e0bb65bde908cc72.js	application/java...	9,375	7/1/2021 12:48:08 AM	7/1/2021 12:48:07 ...	6/28/2021 9:57:51 ...	
trendingNow.e392581962b4b54b1fc4.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/trendingNow.e392581962b4b54b1fc4.js	application/java...	7,244	7/1/2021 12:48:08 AM	7/1/2021 12:48:07 ...	6/28/2021 9:58:07 ...	
trendingNowWC.f2e45ac9135af78ee8fc.js	https://assets.msn.com/bundles/v1/edgeChromium/latest/trendingNowWC.f2e45ac9135af78ee8fc.js	application/java...	13,966	7/1/2021 12:48:08 AM	7/1/2021 12:48:07 ...	6/28/2021 9:58:07 ...	
two.2288b2ea2289.svg.svg	https://www.mozilla.org/media/img/firefox/new/desktop/features/two.2288b2ea2289.svg	image/svg+xml	301	7/1/2021 12:48:47 AM	7/1/2021 12:48:47 ...	4/22/2021 6:22:52 ...	
UA-26908291-4	https://www.googletagmanager.com/gtag/js?id=UA-26908291-4	application/java...	36,353	7/1/2021 12:49:15 AM	7/1/2021 12:49:15 ...	6/30/2021 11:00:00...	7/1/2021 1
UftfQbYuKvGGEUHPU3QGHYd90Z8.br.js	https://www.bing.com/rp/UftfQbYuKvGGEUHPU3QGHYd90Z8.br.js	application/x-ja...	386	7/1/2021 12:59:50 AM		5/13/2021 2:28:46 ...	
ulJ3ckR0YcGpvlX6xhO4prJhEQQ.br.js	https://www.bing.com/rp/ulJ3ckR0YcGpvlX6xhO4prJhEQQ.br.js	application/x-ja...	230	7/1/2021 12:59:50 AM		3/5/2021 3:46:25 AM	
uObc-DdxqYIW3v2ThFF8gzRXABc.br.js	https://www.bing.com/rp/uObc-DdxqYIW3v2ThFF8gzRXABc.br.js	application/x-ja...	782	7/1/2021 12:59:50 AM		12/17/2020 10:21:0...	
upt7Ri3AVBCCNSuZMRK4rMBqXWQ.br.js	https://www.bing.com/rp/upt7Ri3AVBCCNSuZMRK4rMBqXWQ.br.js	application/x-ja...	81	7/1/2021 12:59:50 AM		6/24/2020 7:15:48 ...	
User=m-2EDEAE0D-9AF2-49A8-923F-1839477B5B0C&activityId=2EDEAE0.json	https://assets.msn.com/service/news/feed/pages/wpodhp?User=m-2EDEAE0D-9AF2-49A8-923F-1839477B5...	application/json	17,576	7/1/2021 11:22:51 PM	7/1/2021 11:22:53 ...		7/1/2021 1

894 item(s), 1 Selected (0.00 KB)

Figure C.6: Edge cache entries

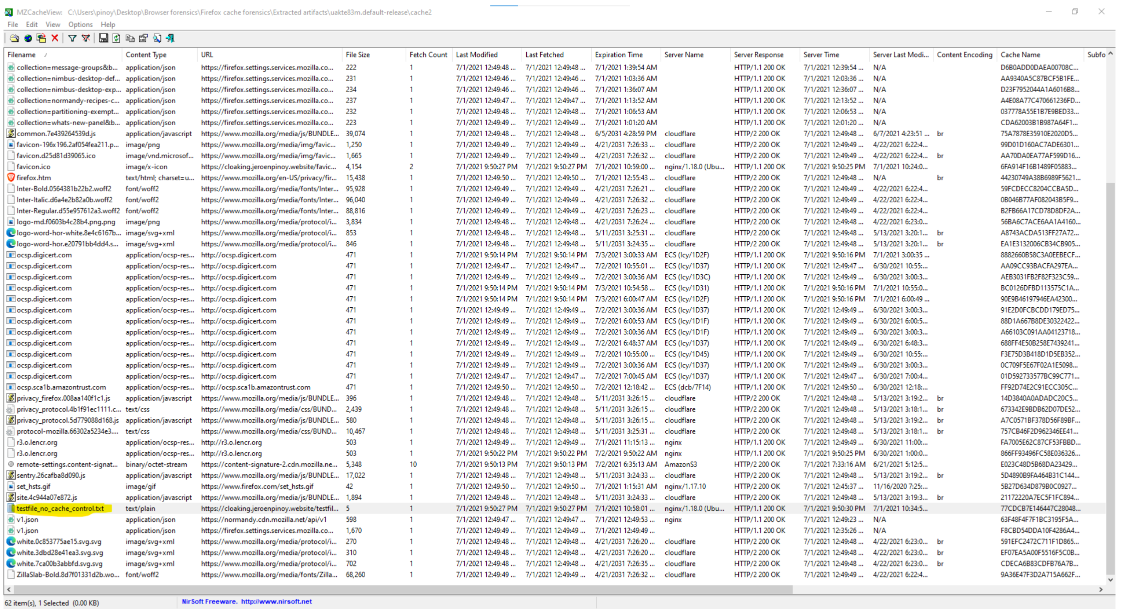
C.4 Firefox


BrowsingHistoryView

File Edit View Options Help

URL	Title	Visit Time	Visit Count	Visited From	Visit Type	Web Browser	User Profile	Browser Profile	URL Length	Typed Count	Histo
https://cloaking.jeroenpinoy.website/testfile_no_cache_control.txt		7/1/2021 9:50:27 PM	1		Typed URL	Firefox		uakte83m.default-release	66		C:\Us
https://cloaking.jeroenpinoy.website/testfile_no_store.txt		7/1/2021 9:50:22 PM	1		Typed URL	Firefox		uakte83m.default-release	58		C:\Us
https://www.mozilla.org/en-US/privacy/firefox/	Firefox Privacy Notice ...	7/1/2021 12:49:48 AM	1	https://www.mozilla.org...	Permanent Redirect	Firefox		uakte83m.default-release	46		C:\Us
https://www.mozilla.org/privacy/firefox/		7/1/2021 12:49:48 AM	1		Link	Firefox		uakte83m.default-release	40		C:\Us

Figure C.7: Firefox visited URLs history


MZCacheView: C:\Users\pinoy\Desktop\Browser forensics\Firefox cache forensics\Extracted artifacts\uakte83m.default-release\cache2

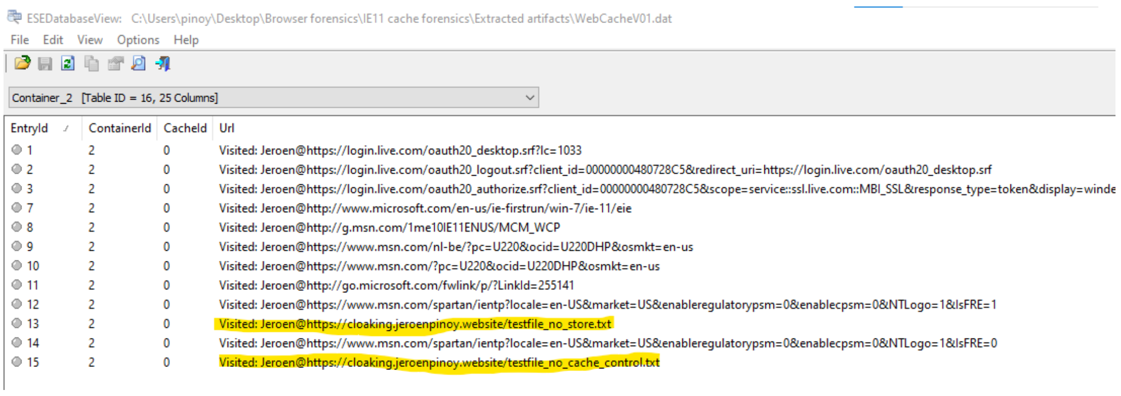
File Edit View Options Help

Filename	Content Type	URL	File Size	Fetch Count	Last Modified	Last Fetched	Expiration Time	Server Name	Server Response	Server Time	Server Last Modi...	Content Encoding	Cache Name	Subfo
collection=message-groups&b...	application/json	https://firefox.settings.services.mozilla.co...	222	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	7/1/2021 1:39:54 AM		HTTP/1.1 200 OK	7/1/2021 12:39:54 ...	N/A		D6B0ADD0DAEA00708C...	
collection=nimbus-desktop-def...	application/json	https://firefox.settings.services.mozilla.co...	231	1	7/1/2021 12:49:46 ...	7/1/2021 12:49:46 ...	7/1/2021 1:03:36 AM		HTTP/1.1 200 OK	7/1/2021 12:03:36 ...	N/A		AA9340A5C87BCF5B1FE...	
collection=nimbus-desktop-exp...	application/json	https://firefox.settings.services.mozilla.co...	234	1	7/1/2021 12:49:46 ...	7/1/2021 12:49:46 ...	7/1/2021 1:36:07 AM		HTTP/1.1 200 OK	7/1/2021 12:36:07 ...	N/A		D23F7952044A1A6016B8...	
collection=normandy-recipes-c...	application/json	https://firefox.settings.services.mozilla.co...	237	1	7/1/2021 12:49:47 ...	7/1/2021 12:49:47 ...	7/1/2021 1:13:52 AM		HTTP/1.1 200 OK	7/1/2021 12:13:52 ...	N/A		A4E08A77C470661236FD...	
collection=partitioning-exempt...	application/json	https://firefox.settings.services.mozilla.co...	232	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	7/1/2021 1:06:53 AM		HTTP/1.1 200 OK	7/1/2021 12:06:53 ...	N/A		037778A55E1B7E9BED33...	
collection=whats-new-panel&b...	application/json	https://firefox.settings.services.mozilla.co...	223	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	7/1/2021 1:01:20 AM		HTTP/1.1 200 OK	7/1/2021 12:01:20 ...	N/A		CDA62003B1B987A64F1...	
common.7e439264539d.js	application/javascript	https://www.mozilla.org/media/js/BUNDLE...	39,074	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	6/5/2031 4:28:59 PM	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	6/7/2021 4:23:51 ...	br	75A7878E35910E2020D5...	
favicon-196x196.2af054fea211.p...	image/png	https://www.mozilla.org/media/img/favic...	1,250	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	4/21/2031 7:26:33 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:49 ...	4/22/2021 6:22:4...		99D01D160AC7ADE6301...	
favicon.d25d81d39065.ico	image/vnd.microsof...	https://www.mozilla.org/media/img/favic...	1,665	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	4/21/2031 7:26:32 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:49 ...	4/22/2021 6:22:4...	br	AA70DA0EA77AF599D16...	
favicon.ico	image/x-icon	https://cloaking.jeroenpinoy.website/favic...	4,154	2	7/1/2021 9:50:27 PM	7/1/2021 9:50:27 PM	7/1/2021 10:59:00 ...	nginx/1.18.0 (Ubu...	HTTP/1.1 200 OK	7/1/2021 9:50:25 PM	7/1/2021 10:24:0...		6FA914F16B1489F05883...	
firefox.htm	text/html; charset=u...	https://www.mozilla.org/en-US/privacy/fir...	15,438	1	7/1/2021 12:49:50 ...	7/1/2021 12:49:50 ...	7/1/2021 12:55:43 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	N/A	br	44230749A38B6989F5621...	
Inter-Bold.0564381b22b2.woff2	font/woff2	https://www.mozilla.org/media/fonts/Inter...	95,928	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	4/21/2031 7:26:21 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:49 ...	4/22/2021 6:22:4...		59FCDECC8204CCBA5D...	
Inter-Italic.d6a4e2b82a0b.woff2	font/woff2	https://www.mozilla.org/media/fonts/Inter...	96,040	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	4/21/2031 7:26:32 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:49 ...	4/22/2021 6:22:4...		0B046B77AF082043B5F9...	
Inter-Regular.d55e957612a3.woff2	font/woff2	https://www.mozilla.org/media/fonts/Inter...	88,816	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	4/21/2031 7:26:23 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:49 ...	4/22/2021 6:22:4...		B2FB66A17CD78D8DF2A...	
logo-md.f0603b4c28b4.png.png	image/png	https://www.mozilla.org/media/protocol/i...	3,834	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	4/21/2031 7:26:24 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	4/22/2021 6:23:0...		56BA6C7ACE6AA1A4160...	
logo-word-hor-white.8e4c6167b...	image/svg+xml	https://www.mozilla.org/media/protocol/i...	853	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	5/11/2031 3:25:31 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	5/13/2021 3:20:1...	br	A8743ACDA513FF27A72...	
logo-word-hor.e20791bb4dd4.s...	image/svg+xml	https://www.mozilla.org/media/protocol/i...	846	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	5/11/2031 3:24:35 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	5/13/2021 3:20:1...	br	EA1E3132006CB34CB905...	
ocsp.digicert.com	application/ocsp-res...	http://ocsp.digicert.com	471	1	7/1/2021 9:50:14 PM	7/1/2021 9:50:14 PM	7/3/2021 3:00:33 AM	ECS (lcy/1D2F)	HTTP/1.1 200 OK	7/1/2021 9:50:16 PM	7/1/2021 3:00:35 ...		8882660B58C3A0EEBECF...	
ocsp.digicert.com	application/ocsp-res...	http://ocsp.digicert.com	471	1	7/1/2021 12:49:47 ...	7/1/2021 12:49:47 ...	7/2/2021 10:55:01 ...	ECS (lcy/1D37)	HTTP/1.1 200 OK	7/1/2021 12:49:47 ...	6/30/2021 10:55:...		AA09CC93BACFA297EA...	
ocsp.digicert.com	application/ocsp-res...	http://ocsp.digicert.com	471	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	7/2/2021 3:00:36 AM	ECS (lcy/1D3C)	HTTP/1.1 200 OK	7/1/2021 12:49:49 ...	6/30/2021 3:00:3...		AEB3031FB2F82F323C59...	
ocsp.digicert.com	application/ocsp-res...	http://ocsp.digicert.com	471	1	7/1/2021 9:50:14 PM	7/1/2021 9:50:14 PM	7/3/2021 10:54:58 ...	ECS (lcy/1D31)	HTTP/1.1 200 OK	7/1/2021 9:50:16 PM	7/1/2021 10:55:0...		BC0126DFBD113575C1A...	
ocsp.digicert.com	application/ocsp-res...	http://ocsp.digicert.com	471	1	7/1/2021 9:50:14 PM	7/1/2021 9:50:14 PM	7/3/2021 6:00:47 AM	ECS (lcy/1D2F)	HTTP/1.1 200 OK	7/1/2021 9:50:16 PM	7/1/2021 6:00:49 ...		90E9B46197946EA42300...	
ocsp.digicert.com	application/ocsp-res...	http://ocsp.digicert.com	471	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	7/2/2021 3:00:36 AM	ECS (lcy/1D37)	HTTP/1.1 200 OK	7/1/2021 12:49:49 ...	6/30/2021 3:00:3...		91E2D0FCBCDD179ED75...	
ocsp.digicert.com	application/ocsp-res...	http://ocsp.digicert.com	471	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	7/2/2021 6:00:53 AM	ECS (lcy/1D1F)	HTTP/1.1 200 OK	7/1/2021 12:49:49 ...	6/30/2021 6:00:5...		88D1A667B8DE30322422...	
ocsp.digicert.com	application/ocsp-res...	http://ocsp.digicert.com	471	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	7/2/2021 3:00:36 AM	ECS (lcy/1D1F)	HTTP/1.1 200 OK	7/1/2021 12:49:49 ...	6/30/2021 3:00:3...		A66103C091AA04123718...	
ocsp.digicert.com	application/ocsp-res...	http://ocsp.digicert.com	471	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	7/2/2021 6:48:37 AM	ECS (lcy/1D37)	HTTP/1.1 200 OK	7/1/2021 12:49:49 ...	6/30/2021 6:48:3...		688FF4E50B258E7439241...	
ocsp.digicert.com	application/ocsp-res...	http://ocsp.digicert.com	471	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	7/2/2021 10:55:00 ...	ECS (lcy/1D45)	HTTP/1.1 200 OK	7/1/2021 12:49:49 ...	6/30/2021 10:55:...		F3E75D3B418D1D5EB352...	
ocsp.digicert.com	application/ocsp-res...	http://ocsp.digicert.com	471	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	7/2/2021 3:00:36 AM	ECS (lcy/1D37)	HTTP/1.1 200 OK	7/1/2021 12:49:49 ...	6/30/2021 3:00:3...		0C709F5E67F02A1E5098...	
ocsp.digicert.com	application/ocsp-res...	http://ocsp.digicert.com	471	1	7/1/2021 12:49:47 ...	7/1/2021 12:49:47 ...	7/2/2021 7:00:45 AM	ECS (lcy/1D37)	HTTP/1.1 200 OK	7/1/2021 12:49:47 ...	6/30/2021 7:00:4...		01D592733577BC99C771...	
ocsp.sca1b.amazontrust.com	application/ocsp-res...	http://ocsp.sca1b.amazontrust.com	471	1	7/1/2021 12:49:50 ...	7/1/2021 12:49:50 ...	7/2/2021 12:18:42 ...	ECS (dcb/7F14)	HTTP/1.1 200 OK	7/1/2021 12:49:50 ...	6/30/2021 12:18:...		FF92D74E2C91ECC305C...	
privacy_firefox.008aa140f1c1.js	application/javascript	https://www.mozilla.org/media/js/BUNDLE...	396	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	5/11/2031 3:26:15 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	5/13/2021 3:19:2...	br	14D3840A0ADADC20C5...	
privacy_protocol.4b1f91ec1111.c...	text/css	https://www.mozilla.org/media/css/BUND...	2,439	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	5/11/2031 3:26:15 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	5/13/2021 3:18:1...	br	673342E9BDB62D07DE52...	
privacy_protocol.5d779088d168.js	application/javascript	https://www.mozilla.org/media/js/BUNDLE...	580	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	5/11/2031 3:26:15 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	5/13/2021 3:19:2...	br	A7C0571BF378D56F89BF...	
protocol-mozilla.66302a5234e3....	text/css	https://www.mozilla.org/media/css/BUND...	10,467	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	5/11/2031 3:25:31 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	5/13/2021 3:18:1...	br	757CB46F2D962346EE41...	
r3.o.lencr.org	application/ocsp-res...	http://r3.o.lencr.org	503	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	7/1/2021 11:15:13 ...	nginx	HTTP/1.1 200 OK	7/1/2021 12:49:49 ...	6/30/2021 11:00:...		FA7005E62C87CF53FBBD...	
r3.o.lencr.org	application/ocsp-res...	http://r3.o.lencr.org	503	1	7/1/2021 9:50:22 PM	7/1/2021 9:50:22 PM	7/2/2021 9:50:22 AM	nginx	HTTP/1.1 200 OK	7/1/2021 9:50:25 PM	6/30/2021 1:00:0...		866FF93496FC58E036326...	
remote-settings.content-signat...	binary/octet-stream	https://content-signature-2.cdn.mozilla.ne...	5,348	10	7/1/2021 9:50:13 PM	7/1/2021 9:50:13 PM	7/2/2021 6:35:13 AM	AmazonS3	HTTP/2 200 OK	7/1/2021 7:33:16 AM	6/21/2021 5:12:5...		E023C48D5B68DA23429...	
sentry.26cafba8d090.js	application/javascript	https://www.mozilla.org/media/js/BUNDLE...	17,022	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	5/11/2031 3:24:33 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	5/13/2021 3:19:2...	br	5D4890B9FA464B31C144...	
set_hsts.gif	image/gif	https://www.firefox.com/set_hsts.gif	42	1	7/1/2021 12:49:50 ...	7/1/2021 12:49:50 ...	7/1/2021 1:15:31 AM	nginx/1.17.10	HTTP/2 200 OK	7/1/2021 12:45:37 ...	11/16/2020 7:25:...		5B27D634D879B0C0927...	
site.4c944a07e872.js	application/javascript	https://www.mozilla.org/media/js/BUNDLE...	1,894	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	5/11/2031 3:24:33 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	5/13/2021 3:19:3...	br	21172220A7EC5F1FC894...	
testfile_no_cache_control.txt	text/plain	https://cloaking.jeroenpinoy.website/testfil...	5	1	7/1/2021 9:50:27 PM	7/1/2021 9:50:27 PM	7/1/2021 10:58:01 ...	nginx/1.18.0 (Ubu...	HTTP/1.1 200 OK	7/1/2021 9:50:30 PM	7/1/2021 10:34:5...		77CDCB7E146447C28048...	
v1.json	application/json	https://normandy.cdn.mozilla.net/api/v1	598	1	7/1/2021 12:49:47 ...	7/1/2021 12:49:47 ...	7/1/2021 12:49:53 ...	nginx	HTTP/1.1 200 OK	7/1/2021 12:49:23 ...	N/A		63F48F4F7F1BC3195F5A...	
v1.json	application/json	https://firefox.settings.services.mozilla.co...	1,670	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...		HTTP/1.1 200 OK	7/1/2021 12:35:26 ...	N/A		F8CBD54DDA10F4286A4...	
white.0c853775ae15.svg.svg	image/svg+xml	https://www.mozilla.org/media/protocol/i...	270	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	4/21/2031 7:26:20 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	4/22/2021 6:23:0...	br	591EFC2472C711F1D865...	
white.3dbd28e41ea3.svg.svg	image/svg+xml	https://www.mozilla.org/media/protocol/i...	310	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	4/21/2031 7:26:20 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	4/22/2021 6:23:0...	br	EF07EA5A00F5516F5C0B...	
white.7ca00b3abbfd.svg.svg	image/svg+xml	https://www.mozilla.org/media/protocol/i...	702	1	7/1/2021 12:49:48 ...	7/1/2021 12:49:48 ...	4/21/2031 7:26:35 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:48 ...	4/22/2021 6:23:0...	br	CDECA6B83CDFB76A7B...	
ZillaSlab-Bold.8d7f01331d2b.wo...	font/woff2	https://www.mozilla.org/media/fonts/Zilla...	68,260	1	7/1/2021 12:49:49 ...	7/1/2021 12:49:49 ...	4/21/2031 7:26:32 ...	cloudflare	HTTP/2 200 OK	7/1/2021 12:49:49 ...	4/22/2021 6:22:4...		9A36E47F3D2A715A662F...	

62 item(s), 1 Selected (0.00 KB)   NirSoft Freeware. http://www.nirsoft.net

Figure C.8: Firefox cache entries

C.5 IE11

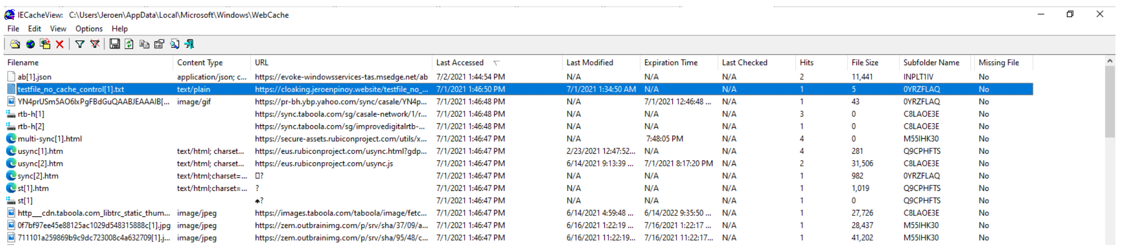


ESEDatabaseView: C:\Users\pinoy\Desktop\Browser forensics\IE11 cache forensics\Extracted artifacts\WebCacheV01.dat

Container_2 [Table ID = 16, 25 Columns]

EntryId	ContainerId	CacheId	Url
1	2	0	Visited: Jeroen@https://login.live.com/oauth20_desktop.srf?lc=1033
2	2	0	Visited: Jeroen@https://login.live.com/oauth20_logout.srf?client_id=00000000480728C5&redirect_uri=https://login.live.com/oauth20_desktop.srf
3	2	0	Visited: Jeroen@https://login.live.com/oauth20_authorize.srf?client_id=00000000480728C5&scope=service::ssl.live.com::MBI_SSL&response_type=token&display=winde
7	2	0	Visited: Jeroen@http://www.microsoft.com/en-us/ie-firstrun/win-7/ie-11/eie
8	2	0	Visited: Jeroen@http://g.msn.com/1me10IE11ENUS/MCM_WCP
9	2	0	Visited: Jeroen@https://www.msn.com/nl-be/?pc=U220&ocid=U220DHP&osmkt=en-us
10	2	0	Visited: Jeroen@https://www.msn.com/?pc=U220&ocid=U220DHP&osmkt=en-us
11	2	0	Visited: Jeroen@http://go.microsoft.com/fwlink/p/?LinkId=255141
12	2	0	Visited: Jeroen@https://www.msn.com/spartan/ientp?locale=en-US&market=US&enableregulatorypsm=0&enablecpsm=0&NTLogo=1&IsFRE=1
13	2	0	Visited: Jeroen@https://cloaking.jeroenpinoy.website/testfile_no_store.txt
14	2	0	Visited: Jeroen@https://www.msn.com/spartan/ientp?locale=en-US&market=US&enableregulatorypsm=0&enablecpsm=0&NTLogo=1&IsFRE=0
15	2	0	Visited: Jeroen@https://cloaking.jeroenpinoy.website/testfile_no_cache_control.txt

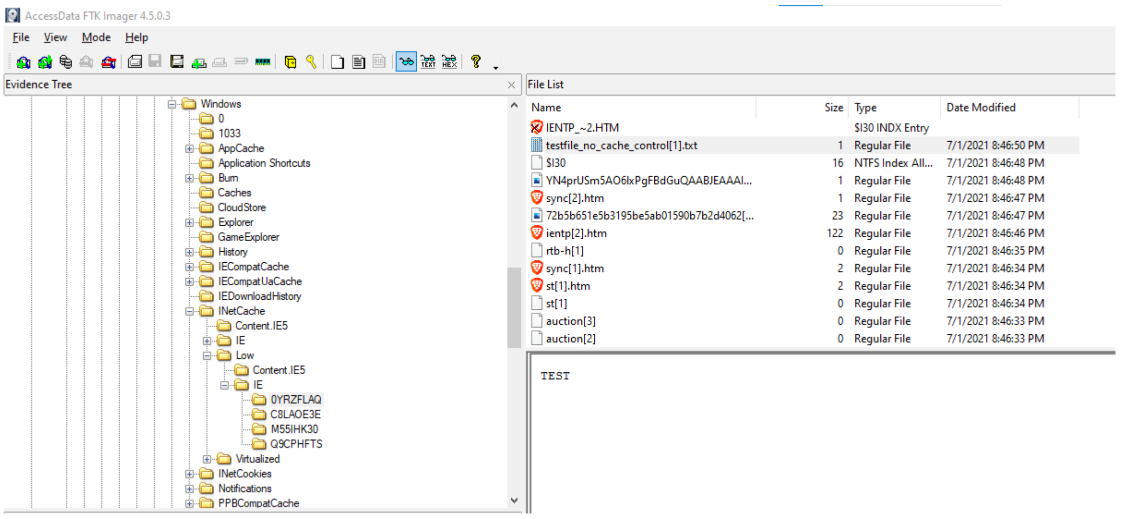
Figure C.9: Internet Explorer 11 visited URLs history



IECacheView: C:\Users\Jeroen\AppData\Local\Microsoft\Windows\WebCache

Filename	Content Type	URL	Last Accessed	Last Modified	Expiration Time	Last Checked	Hits	File Size	Subfolder Name	Missing File
ab[1].json	application/json; c...	https://evoke-windowsservices-tas.msedge.net/ab	7/2/2021 1:44:54 PM	N/A	N/A	N/A	2	11,441	INPLT1IV	No
testfile_no_cache_control[1].txt	text/plain	https://cloaking.jeroenpinoy.website/testfile_no_...	7/1/2021 1:46:50 PM	7/1/2021 1:34:50 AM	N/A	N/A	1	5	0YRZFLAQ	No
YN4prUSm5AO6lxPgFBdGuQAABJEAAAIB[...	image/gif	https://pr-bh.ybp.yahoo.com/sync/casale/YN4p...	7/1/2021 1:46:48 PM	N/A	7/1/2021 12:46:48 ...	N/A	1	43	0YRZFLAQ	No
rtb-h[1]		https://sync.taboola.com/sg/casale-network/1/r...	7/1/2021 1:46:48 PM	N/A	N/A	N/A	3	0	C8LAOE3E	No
rtb-h[2]		https://sync.taboola.com/sg/improvedigitalrtb-...	7/1/2021 1:46:48 PM	N/A	N/A	N/A	1	0	C8LAOE3E	No
multi-sync[1].html		https://secure-assets.rubiconproject.com/utils/x...	7/1/2021 1:46:47 PM	N/A	7:48:05 PM	N/A	4	0	M55IHK30	No
usync[1].htm	text/html; charset...	https://eus.rubiconproject.com/usync.html?gdp...	7/1/2021 1:46:47 PM	2/23/2021 12:47:52...	N/A	N/A	4	281	Q9CPHFTS	No
usync[2].htm	text/html; charset...	https://eus.rubiconproject.com/usync.js	7/1/2021 1:46:47 PM	6/14/2021 9:13:39 ...	7/1/2021 8:17:20 PM	N/A	2	31,506	C8LAOE3E	No
sync[2].htm	text/html;charset=...	??	7/1/2021 1:46:47 PM	N/A	N/A	N/A	1	982	0YRZFLAQ	No
st[1].htm	text/html;charset=...	?	7/1/2021 1:46:47 PM	N/A	N/A	N/A	1	1,019	Q9CPHFTS	No
st[1]		?	7/1/2021 1:46:47 PM	N/A	N/A	N/A	1	0	Q9CPHFTS	No
http___cdn.taboola.com_libtrc_static_thum...	image/jpeg	https://images.taboola.com/taboola/image/fetc...	7/1/2021 1:46:47 PM	6/14/2021 4:59:48 ...	6/14/2022 9:35:50 ...	N/A	1	27,726	C8LAOE3E	No
0f7bf97ee45e88125ac1029d548315888c[1].jpg	image/jpeg	https://zem.outbrainimg.com/p/srv/sha/37/09/a...	7/1/2021 1:46:47 PM	6/16/2021 1:22:19 ...	7/16/2021 1:22:17 ...	N/A	1	28,437	M55IHK30	No
711101a259869b9c9dc723008c4a632709[1].j...	image/jpeg	https://zem.outbrainimg.com/p/srv/sha/95/48/c...	7/1/2021 1:46:47 PM	6/16/2021 11:22:19...	7/16/2021 11:22:17...	N/A	1	41,202	M55IHK30	No

Figure C.10: Internet Explorer 11 cache entries



AccessData FTK Imager 4.5.0.3

Evidence Tree: Windows > 0, 1033, AppCache, Application Shortcuts, Burn, Caches, CloudStore, Explorer, GameExplorer, History, IECompatCache, IECompatUaCache, IEDownloadHistory, INetCache (Content.IE5, IE, Low (Content.IE5, IE (0YRZFLAQ, C8LAOE3E, M55IHK30, Q9CPHFTS)), Virtualized), INetCookies, Notifications, PPBCompatCache

File List

Name	Size	Type	Date Modified
IENTP_~2.HTM		$I30 INDX Entry	
testfile_no_cache_control[1].txt	1	Regular File	7/1/2021 8:46:50 PM
$I30	16	NTFS Index All...	7/1/2021 8:46:48 PM
YN4prUSm5AO6lxPgFBdGuQAABJEAAAI...	1	Regular File	7/1/2021 8:46:48 PM
sync[2].htm	1	Regular File	7/1/2021 8:46:47 PM
72b5b651e5b3195be5ab01590b7b2d4062[...	23	Regular File	7/1/2021 8:46:47 PM
ientp[2].htm	122	Regular File	7/1/2021 8:46:46 PM
rtb-h[1]	0	Regular File	7/1/2021 8:46:35 PM
sync[1].htm	2	Regular File	7/1/2021 8:46:34 PM
st[1].htm	2	Regular File	7/1/2021 8:46:34 PM
st[1]	0	Regular File	7/1/2021 8:46:34 PM
auction[3]	0	Regular File	7/1/2021 8:46:33 PM
auction[2]	0	Regular File	7/1/2021 8:46:33 PM

```
TEST
```

Figure C.11: Internet Explorer 11 cache – file entry

Appendix D

Questionnaire

Malware delivery via web: awareness of specific attacker technique

This survey is being performed by Jeroen Pinoy as part of Master thesis research, in scope of the Computer Science graduate programme at the Open University of the Netherlands (Open Universiteit Nederland). The research is supervised by assistant professor Dr. Ir. Hugo Jonker.

This questionnaire is divided into sections (displayed as separate pages), each containing a set of questions. While filling in the questionnaire, it is not possible to go back to previous sections. This is intentional and important for the validity of the results, please keep this in mind. Please answer the questions from memory and do not search for answers using google for example.

I would like to thank you in advance for participating in this survey! Note that this survey is anonymized by default. For part of this research, I need to be able to group responses by organisation however. If possible, please fill in the name of your organisation or the code word I sent with my request if you received one, as answer to the related question in the *your professional background* section.

Feel free to reach out using one of the below channels in case you have any questions or comments, or if you would like to receive a copy of my master thesis once it is finished (public pgp key can be found on the CIRCL key server - https://pgp.circl.lu/).

Preferred
jeroen.pinoy@fortitude.ninja
PGP Fingerprint=2C5B D8EE 4422 243E 10A5 4799 DF33 A50B 8E4E E081

Student email address
j.pinoy@studie.openuniversiteit.be
PGP Fingerprint=1988 D542 2EFF 6BD2 14F0 AF58 5560 1DB9 C9F9 0D01

Academic institution details

Open University of the Netherlands (Open Universiteit)

Valkenburgerweg 177, 6419 AT Heerlen, Netherlands

https://www.ou.nl/

There are 33 questions in this survey.

Awareness - part 1

In the remainder of this survey, questions are always asked in the context of web based cyber attacks

How would you define "Drive-by compromise"?

Please answer *I do not know* in case you do not know.

*

Please write your answer here:

What, if anything, can an organisation do to protect itself against drive-by compromise attacks?

Please answer *I do not know* in case you do not know.

*

Please write your answer here:

Awareness - part 2

Before continuing, please read the below explanation of *drive-by compromise* from the MITRE ATT&CK® framework:

Adversaries may gain access to a system through a user visiting a website over the normal course of browsing. With this technique, the user's web browser is typically targeted for exploitation, but adversaries may also use compromised websites for non-exploitation behavior such as acquiring Application Access Token.

Multiple ways of delivering exploit code to a browser exist, including:

- A legitimate website is compromised where adversaries have injected some form of malicious code such as JavaScript, iFrames, and cross-site scripting.
- Malicious ads are paid for and served through legitimate ad providers.
- Built-in web application interfaces are leveraged for the insertion of any other kind of object that can be used to display web content or contain a script that executes on the visiting client (e.g. forum posts, comments, and other user controllable web content).

How would you define "Drive-by target"?
Please answer *I do not know* in case you do not know.
*

Please write your answer here:

Awareness - part 3

To your knowledge, in what ways can an attacker target a specific set of clients when executing a drive-by compromise attack?

Please answer *I do not know* in case you do not know.

*

Please write your answer here:

Awareness - part 4

What, if anything, can an organisation do to protect itself against *targeted* drive-by compromise attacks? Note that the emphasis in this case is on *targeted* attacks.

Please answer *I do not know* in case you do not know. In case you wouldn't change your answer compared to the first question of this survey, which was "What, if anything, can an organisation do to protect itself against drive-by compromise attacks?", please answer **nothing different from what I mentioned before**.

*

Please write your answer here:

Awareness - part 5

How might an analyst investigate a potential drive-by compromise attack?

Please answer *I do not know* in case you do not know.

*

Please write your answer here:

Does your organisation collect and use cyber threat intelligence, including threat intelligence about drive-by compromise attacks? *

❗ Choose one of the following answers
Please choose **only one** of the following:

◯ Yes
◯ No
◯ I don't know
◯ I prefer not to answer this question

Risk - part 1

Please refer to the following definitions from MITRE ATT&CK©:

- *Defense Evasion* consists of techniques that adversaries use to avoid detection throughout their compromise.
- *Vulnerability Scanning*: Adversaries may scan victims for vulnerabilities that can be used during targeting. Vulnerability scans typically check if the configuration of a target host/application (ex: software and version) potentially aligns with the target of a specific exploit the adversary may seek to use.

We argue that vulnerability scanning can be used to selectively target specific systems by only attempting a compromise if a specific set of vulnerabilities has been detected. This can also be done to decrease the possibility of detection and is as such a form of defense evasion.

Aside from using vulnerability scanning to select potential victims, other defense evasion techniques can be used while performing a drive-by compromise attack. Some examples include:

- Usage of a compromised site which the target group frequently visits, that is, the watering hole technique
- Check the referer header, for example to verify the user was redirected to the page by clicking a search engine result
- Check for normal user behaviour such as clicking, to counter automatic analysis
- Blocking of repeated exploit delivery
  - IP based
- IP based blocklisting, for example using commercial blocklists containing (security related) web bot IPs

How likely do you consider it to be that a *drive-by compromise* attack using the below techniques is performed against your **organisation**? *

Please choose the appropriate response for each item:

	0 - not likely at all	1 - low likelihood	2 - medium likelihood	3 - high likelihood	4 - very high likelihood
no **usage of vulnerability scanning,** ***no*** **other defense evasion techniques**	◯	◯	◯	◯	◯
usage of vulnerability scanning, ***no*** **other defense evasion techniques**	◯	◯	◯	◯	◯
no **usage of vulnerability scanning, using other defense evasion techniques**	◯	◯	◯	◯	◯
usage of vulnerability scanning and other defense evasion techniques	◯	◯	◯	◯	◯

How high would the impact be if your organisation was compromised by a *drive-by compromise* using the below *techniques?*

*

Please choose the appropriate response for each item:

	0 - no impact	**1 - low impact**	**2 - medium impact**	**3 - high impact**	**4 - very high impact**
***no* vulnerability detection, *no* other defense evasion techniques**	○	○	○	○	○
vulnerability detection, *no* other defense evasion techniques	○	○	○	○	○
***no* vulnerability detection, using other defense evasion techniques**	○	○	○	○	○
vulnerability detection and other defense evasion techniques	○	○	○	○	○

Please motivate your choices:

Please write your answer here:

Risk - part 2

Please refer to the following definitions from MITRE ATT&CK©:

- *Defense Evasion* consists of techniques that adversaries use to avoid detection throughout their compromise.
- *Vulnerability Scanning*: Adversaries may scan victims for vulnerabilities that can be used during targeting. Vulnerability scans typically check if the configuration of a target host/application (ex: software and version) potentially aligns with the target of a specific exploit the adversary may seek to use.

We argue that vulnerability scanning can be used to selectively target specific systems by only attempting a compromise if a specific set of vulnerabilities has been detected. This can also be done to decrease the possibility of detection and is as such a form of defense evasion.

Aside from using vulnerability scanning to select potential victims, other defense evasion techniques can be used while performing a drive-by compromise attack. Some examples include:

- Usage of a compromised site which the target group frequently visits, that is, the watering hole technique
- Check the referer header, for example to verify the user was redirected to the page by clicking a search engine result
- Check for normal user behaviour such as clicking, to counter automatic analysis
- Blocking of repeated exploit delivery
  - IP based
- IP based blocklisting, for example using commercial blocklists containing (security related) web bot IPs

Did you hear or read about a successful drive-by compromise attack that was performed in the last two years? *

❗ Choose one of the following answers
Please choose **only one** of the following:

◯ Yes
◯ No
◯ I can't remember

Did one of those attacks you heard or read about use detection evasion other than vulnerability detection? *

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '11 [RMQ3]' (Did you hear or read about a successful drive-by compromise attack that was performed in the last two years?)

❗ Choose one of the following answers
Please choose **only one** of the following:

◯ Yes
◯ No
◯ I can't remember

What type of defense evasion did that attack or those attacks use?

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '12 [RMQ4]' (Did one of those attacks you heard or read about use detection evasion other than vulnerability detection?)

Please write your answer here:

Defenses - part 1

Which of the following strategies against drive-by compromise attacks has *your organisation* implemented? Please use the *other* option if you want to add entries.

Check all that apply
Please choose **all** that apply:

☐ Separation of the browser environment from the organisation network (for example using virtualization), that is, browser isolation

☐ Usage of antivirus software on the client systems

☐ Regular updating of browser software

☐ Disabling javascript

☐ Usage of a web filter proxy that performs category based web filtering (blocking gambling sites for example)

☐ Usage of a web filter proxy that uses antivirus software to perform content analysis

☐ I do not want to answer this question. (Please note that unless you fill in your organisation name later in this survey, your answers are anonymized)

☐ Other:

Which of those defensive strategies do you consider the most important?

Please write your answer here:

Please motivate your choice:

Please write your answer here:

Defenses - part 2

Browser fingerprinting refers to the process of collecting information through a web browser and web server, to build a fingerprint of a device or user, as well as the process of using a previously determined browser fingerprint to re-identify a user or device. Browser fingerprinting is one of the main techniques used for targeting in a drive-by compromise attack.

Which of the following strategies against *browser fingerprinting* has *your organisation* implemented? Please use the *other* option if you want to add entries and note that we did not repeat the strategies mentioned earlier on purpose.

❗ Check all that apply

Please choose **all** that apply:

☐ Usage of a browser that attempts to give the same fingerprint for all users, such as Tor browser

☐ Usage of a browser that adds randomness to the fingerprintable surface, such as Brave

☐ Usage of an anti-fingerprinting browser plugin or extension

☐ Other:

Which of those defensive strategies do you consider the most important?

Please write your answer here:

Please motivate your choice:

Please write your answer here:

Defenses - part 3

According to you, should your organisation implement additional defensive strategies against drive-by compromise attacks? *

❗ Choose one of the following answers

Please choose **only one** of the following:

◯ Yes

◯ No

◯ I do not want to answer this question

Please motivate your choice:

Only answer this question if the following conditions are met:
Answer was 'Yes' *or* 'No' at question '20 [DQ3]' (According to you, should your organisation implement additional defensive strategies against drive-by compromise attacks?)

Please write your answer here:

Investigation - part 1

How should an analyst *investigate* a drive-by compromise attack to account for potential *defense evasion* techniques used during the drive-by compromise?

Please answer *I do not know* in case you do not know.

*

Please write your answer here:

Investigation - part 2

Which of the following strategies to investigate potential drive-by compromise attacks does your organisation use? Please use the *other* option if you want to add entries.

❗ Check all that apply
Please choose **all** that apply:

☐ Usage of a malware analysis sandbox with url analysis capabilities, that is, high-interaction honeyclients such as Cuckoo Sandbox

☐ Usage of a physical malware analysis host configured in a similar way as a normal end user system or the affected system

☐ Usage of vulnerable browser *emulation* software. Emulated browsers do not offer full browser functionality but only implement a limited set of functionality. In that sense they are different from full browsers installed inside a virtual machine. Examples are low-interaction honeyclients such as Thug.

☐ Usage of web based url analysis tools such as VirusTotal and Lookyloo

☐ Forensic analysis of the affected system

☐ Usage of different client IP ranges during analysis with honeyclients

☐ Usage of different operating system and browser combinations during analysis with honeyclients

☐ Forensic analysis on network related data, that is, network forensics

☐ Correlation of gathered data with available threat intelligence

☐ I prefer not to answer this question

☐ Other: ____________________

Your professional background

How many years of cyber security related work or research experience do you have (rounding up to the nearest amount of years)? *

❗ Only numbers may be entered in this field.

Please write your answer here:

Which of the following roles is closest to your current job role? If none of the given roles applies at all, please give your own description. *

❗ Choose one of the following answers

Please choose **only one** of the following:

- ◯ cyber analyst (blue team)
- ◯ red team member
- ◯ IT security manager (middle management)
- ◯ chief information security officer (CISO)
- ◯ Other

Which of the following roles have you performed in your career, please also check the box if you performed a very similar role?

❗ Check all that apply

Please choose **all** that apply:

- ☐ digital forensic investigator / incident responder (this includes SOC analysts)
- ☐ threat intelligence analyst
- ☐ security engineer (designing and implementing defenses such as security tools and firewall rules)
- ☐ red teamer
- ☐ IT security manager (middle management)
- ☐ chief information security officer (CISO)

Is the *organisation* you currently work for part of the public sector, private sector or not-for-profit sector? *

❗ Choose one of the following answers

Please choose **only one** of the following:

- ○ Public sector
- ○ Semi-public sector
- ○ Private sector
- ○ Not-for-profit sector
- ○ I prefer not to answer this question
- ○ Other ____________________

Which of the following categories describe the industry your *organisation* is a part of? If none of the given industries applies, please use the *other* option. *

❗ Check all that apply

Please choose **all** that apply:

☐ Finance and Insurance

☐ Telecommunications

☐ Utilities

☐ Government and Public Administration

☐ Education

☐ Other:

Are you willing to fill in the name of your organisation or the code word I sent with my request if you received one? This would solely be used to group survey responses per organisation. *

❗ Choose one of the following answers

Please choose **only one** of the following:

○ Yes

○ No

Please fill in the name of your organisation or the code word I sent with my request if you received one:

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '29 [YPB2]' (Are you willing to fill in the name of your organisation or the code word I sent with my request if you received one? This would solely be used to group survey responses per organisation.)

Please write your answer here:

How long has your organisation existed so far?

Only answer this question if the following conditions are met:
Answer was 'No' at question '29 [YPB2]' (Are you willing to fill in the name of your organisation or the code word I sent with my request if you received one? This would solely be used to group survey responses per organisation.)

Choose one of the following answers
Please choose **only one** of the following:

- ◯ 0-3 years
- ◯ 4-5 years
- ◯ 6-10 years
- ◯ 10+ years
- ◯ I don't know

How many employees work at your organisation?

Only answer this question if the following conditions are met:
Answer was 'No' at question '29 [YPB2]' (Are you willing to fill in the name of your organisation or the code word I sent with my request if you received one? This would solely be used to group survey responses per organisation.)

Choose one of the following answers
Please choose **only one** of the following:

- ◯ Less than 10
- ◯ Between 10 and 50
- ◯ Between 51 and 250
- ◯ More than 250
- ◯ I don't know

Final comments

That brings us to the end of our survey. Do you have any comments on the subject of our survey or on the survey itself?

Please write your answer here:

Thank you very much for participating in this survey. Your answers are valuable for our research!

Please feel free to reach out using one of the below channels in case you have any questions or comments, or if you would like to receive a copy of my master thesis once it is finished (public pgp key can be found on the CIRCL key server - https://pgp.circl.lu/).

Preferred
jeroen.pinoy@fortitude.ninja
PGP Fingerprint=2C5B D8EE 4422 243E 10A5 4799 DF33 A50B 8E4E E081

Student email address
j.pinoy@studie.openuniversiteit.be
PGP Fingerprint=1988 D542 2EFF 6BD2 14F0 AF58 5560 1DB9 C9F9 0D01

Submit your survey.
Thank you for completing this survey.