Introduction

In July 2021 Microsoft patched 117 CVEs in Windows related applications[6], where 13 of these were rated critical, 103 rated important and only one rated moderate in severity. According to Edgescans Vulnerability Statistics Report from 2021[4], the average Mean Time to Remediate (MTTR) for all severities of vulnerabilities is 60.3 days. That is, on average it takes two months between the patch being publicly available until it is installed, while the longest time to patch was a total of 309 days. With these numbers in mind, there is a lot of time for a malicious actor to analyze the patch, create an exploit for the vulnerability and use it to attack companies.

Even if a company has a good patch policy and applies available patches within a short amount of time, as many as 29%[12] of an organizations business-critical systems rely on legacy systems. Such systems are especially critical in terms of ensuring that a patch does not break functionality. This is also the reason that, as of september 2020, 59% of all uniquely observed instances of Windows Server had reached end-of-life[11].

With ransomware attacks growing year by year[10], a rule of thumb has surfaced, in that you should consider your company compromised in some way or another. This might be an employee whose company email account has been phished, leaked credentials from a third party website, malware on an employee PC or the exploitation of a vulnerability.

The main issues with applying patches is the requirement to uphold the uptime for the affected machine. In some cases the machine might stop working as expected, while in other cases a required restart is not feasible. In these instances companies are reluctant to update immediately until the patch is either tested properly in a test environment or the patch has been battle tested on less critical machines.

Once a patch is released for a Microsoft product, it can be analyzed by anyone including malicious actors, who can then develop a so-called N-day exploit for the vulnerability. This has been an ongoing issue with CVE-2017-11882[17] which is a vulnerability in Microsoft Office's Equation Editor. This vulnerability has been, and still is used to deliver malware to unpatched machines.

Whatever the case is, some machines cannot or will not be updated immediately after a patch release, and it is these machines we will focus on in this project. If a company were able to be notified of an exploitation attempt at the time between a patch has been released and the patch is applied, they would be more secure while still upholding the uptime for business-critical systems.

1.1 Purpose

Purpose

1.2 Thesis overview

Chapter 2: Tracing and logging. In this chapter we present the theory behind tracing, logging and telemetry methods and tools within the Windows operating system.

Chapter 3: Vulnerability analysis. In this chapter we present the analysis of CVE-2021-24086 that lead to the root-cause of the vulnerability and a fully working Proof of Concept (PoC). Furthermore, the chapter also contains a primer on IPv6 headers needed to exploit CVE-2021-24086.

Chapter 4: Detection. In this chapter we analyze CVE-2021-24086 in regards to the detection methods discussed in **chapter 3: Vulnerability analysis**. As a part of this analysis we showcase an application that is able to detect exploitation attempts of CVE-2021-24086.

Chapter 5: Discussion. In this chapter we discuss and compare the differences between the detection methods of **chapter 3: Vulnerability analysis**. We also discuss the work needed to create an automated and scalable solution that can use patch information to detect vulnerability exploitation attempts.

Chapter 6: Conclusion. In this chapter we resent the general conclusion for the work done.

1.3 Related work

Using host-based telemetry gathering methods to detect exploitation of vulnerabilities, both known and unknown, has been widely discussed before[9][3]. One example hereof is DACODA[3], which traces a network packet through the relevant processes until an unintended memory state happens, such as jumping to an address present in the packet.
Many Network Detection and Response (NDR) products claim to be able to detect exploitation of both known vulnerabilities and zero-days[13][15], but are mostly limited to behavior analysis of the network using proprietary machine learning models. Some work, such as ZeroWall[14] exists for detection of vulnerabilities in web based applications.
Many security software vendors claim to have developed models and techniques to detect against Zero-day attacks[2][1][8], most of the work is proprietary and not available for study. Furthermore, based on publicly available information

none of the vendors explain what telemetry is used other than *"host and network based data"*.
As with any other computer science field, machine learning has also been applied to exploitation of vulnerabilities. One example hereof is FastEmbed[5] which attempts to use machine learning to predict the number of exploits present in the wild, but is not related to detection of exploitation attempts.
To our knowledge, not a lot of research can be found on using information gathered from patches to detect known vulnerabilities. Most research in this area revolves around using patch information to discover similar vulnerabilities[16][7]