|  |
| --- |
| ASP.NET Vulnerability Assessment  Part 1: Software Security  Mikolaj M. Mroz  2003114  BSc (Hons) Ethical Hacking, 2023  CMP417: Engineering Resilient Systems 1 |

|  |
| --- |
| School of Design and Informatics  Abertay University |

Contents

[Context ii](#_Toc129724265)

[Denial of Service Attacks ii](#_Toc129724266)

[Operational Risks ii](#_Toc129724267)

[Recommended Action: Defence in Depth iii](#_Toc129724268)

[What is Defence in Depth? iii](#_Toc129724269)

[Effectiveness iv](#_Toc129724270)

[Team Size iv](#_Toc129724271)

[Team Experience iv](#_Toc129724272)

[Critical Comparison v](#_Toc129724273)

[How can it be implemented? vi](#_Toc129724274)

[Physical Security Controls vi](#_Toc129724275)

[Technical Security Controls vi](#_Toc129724276)

[Administrative Security Controls vi](#_Toc129724277)

[Prevention vii](#_Toc129724278)

[Technical Security – Updates vii](#_Toc129724279)

[Technical Security - Zero Trust Method viii](#_Toc129724280)

[Administrative Security - Backups ix](#_Toc129724281)

[Administrative Security – Security Awareness Training ix](#_Toc129724282)

[References x](#_Toc129724283)

# Context

The team’s implementation of an ASP.NET-based web frontend, commonly used within the client company’s Human Resources (HR) department to manage staff account details, is open to a variety of vulnerabilities listed in the Common Vulnerability and Exposures (CVE) online database.

ASP.NET is an open-source framework developed by Microsoft for use in developing web applications and services utilising Active Server Pages (ASPs), which allow web developers to make their websites more dynamic and interactive. On top of this, the framework is frequently updated and documented in depth making it a very popular choice for web application development and other services such as login systems as demonstrated by ScottishGlen (The Client).

## Denial of Service Attacks

One of the most common issues with the ASP.NET core upon which the staff management frontend is based on is its vulnerability to Denial of Service (DoS) attacks. These attacks occur when a software system is crashed or otherwise rendered inaccessible by an attacker. These attacks paired with a poor mitigation strategy may result in much lengthy downtime or even damage being done to the data stored on the targeted server particularly if a buffer overflow attack vector is utilised.

## Operational Risks

Should an attacker exploit this vulnerability, the client risks losing access to its staff management system for an undefined length of time. Any other aspects of the frontend written in ASP.NET may also be targeted and affected by the DoS such as external facing sites and services, particularly if they are stored on the same server.

The vulnerabilities occur in a variety of locations throughout the ASP.NET code and to varying degrees of severity. Each of the following may result in a DoS if exploited by an attacker:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CVE Identifier | Affected Versions | Description | CVE  Severity Rating | Red Hat Severity Rating |
| 2022-38013 | < .NET Core 3.1.28  < .NET 6.0.8 | A customised payload can trigger a stack overflow during the model binding process. | 5.0 | 5.9 |
| 2021-1723 | < .NET Core 3.1.10  < .NET 5.0.1 | A vulnerability in the way ASP.NET Kestrel parses HTTP2 requests results in the system running call-backs outside of locks. | 5.0 | 7.5 |
| 2020-1597 | < .NET Core 3.1.6  < .Net Core 2.1.20 | An issue with how ASP.NET Core handles client disconnects can result in specific packages consuming an unnecessary amount of processing power and storage space. | 5.0 | 7.5 |
| 2020-1161 | < .NET Core 3.1.3 | Microsoft.AspNetCore.App has a flaw that allows for an attacker to trigger an infinite loop. | 5.0 | 7.5 |

*Table 1 – A list of ASP.NET DoS vulnerabilities from CVE.*

# Recommended Action: Defence in Depth

Fundamentally, the current implementation of the ASP.NET system lacks a defence in depth implementation approach leading to issues which threaten company systems.

## What is Defence in Depth?

Defence in Depth is a commonly used cybersecurity approach in which the user implements a series of layered systems and mitigations to reduce the threat and likelihood of a cyberattack and can be utilised in several different situations ranging from software development to networking. For example, a company may choose to use a software firewall installed on a device despite already having an active physical firewall on the network perimeter. This allows for any attacks designed to infiltrate the outer firewall to be caught by its internal counterpart and vice versa, as the chances of an attacker slipping past each additional security measure should theoretically shrink with each correctly implemented layer (Mell, et al., 2016).

### Effectiveness

Defence in Depth’s biggest strength comes from the various ways software and hardware vendors from which security solutions are received structure their applications - it is highly unlikely that two pieces of unrelated software will use precisely the same approach against a common vulnerability resulting in improved protection against hacks. However, care must be taken to reach an ideal middle-ground in building a Defence in Depth system as introducing too many layers may cause a greatly diminished return in effectiveness potentially opening even more dangerous attack vectors for the attacker to exploit.

### Team Size

Taking the small size of the development team into account, the selected solutions were selected as to not require any additional manpower outside of the defined team of 6 company developers. With smaller teams, it is important to keep in mind that variables such as installation time and budgets require the given recommendations to take a trade-off approach where the implemented security measures cannot be not so cheap and simple to set up that they are ineffective but also not so expensive that they stretch outside the available funds.

### Team Experience

The team can take a number of different approaches depending on their expertise with networking and an understanding cybersecurity. In order to provide the best possible security however, the most effective Defence in Depth measures require a moderate amount of experience with cybersecurity and networking. Based off the brief’s information regarding the fact that all company systems were customized to a certain extent, it is implied that the developers have adequate knowledge of networking, programming, and relevant security practices.

### Critical Comparison

Implementing a Defence in Depth approach offers the best protection against DoS attacks on ASP.NET, particularly when contrasted against alternative solutions. The first solution that may be considered by an uninformed developer is using a different development framework due to its vulnerability to these attacks. However, ASP.NET is perfectly suitable as a development framework and is favoured highly among top developers due to its reliability in spite of the vulnerabilities that can be found in the CVE database – it continues to be the 2nd most popular server-side programming language after PHP (W3Techs, 2023). This is because DoS attacks should not cause any issues if the relevant Defence in Depth measures are taken as they should prevent attackers from being able to reach those code-related vulnerabilities in the first place.

Another potential approach could be implementing security checks into the Continuous Integration process of software development, such as the OWASP Secure Coding Practices guide or manual review checklists. This is a perfectly valid strategy to implement, particularly if the software used is often customised as noted in the brief as it tackles the issue of dealing with security issues throughout the entire development process. However, this stage can easily be incorporated into an effective Defence in Depth plan which makes using it as a standalone solution obsolete.

Finally, it may be incorrectly assumed that simply educating the development team on secure coding practices will be enough to prevent DoS attacks in the future. As mentioned previously, it can be assumed that the development team is already knowledgeable on different forms of software security. However according to Gordon Fleisch Company datacentre expert Bambulas (2022), it is recommended that development teams have a refresher security exam every four to six months to maintain a consistent level of knowledge and experience with secure coding practices. Many different forms of education exist online for situations such as this and at different price points to suit the team’s needs, though solely implementing this solution and securing ASP.NET net code with this new knowledge will not prevent DoS attacks from taking place on other sections of the network – this is one of the biggest advantages of implementing a defence in depth strategy as it has the potential to protect much more of the network, more effectively.

## How can it be implemented?

Defence in Depth is a broad term used to describe layered security and therefore varies greatly between businesses. However, Cloudflare, the leading company in DoS protection services, segment the necessary network controls into three categories (2023):

* Physical Security Controls
* Technical Security Controls
* Administrative Security Controls

### Physical Security Controls

These protect physical assets from attacks such as break-ins and theft. While these are not particularly relevant to the issue of DoS attacks, making use of secure access control systems and CCTV surveillance has the can help prevent reconnaissance from taking place, helping to deter attackers and protect valuable resources.

### Technical Security Controls

Much more relevant to preventing DoS attacks, Technical Security Controls require effective configurations of the hardware and software in a network to block incoming data breaches and other attacks that utilise networks and software.

### Administrative Security Controls

Finally, Administrative Security Controls are maintained by the network administrators to ensure users do not have access to unauthorized materials, keeping sensitive data stored correctly, and avoiding exposing unnecessary parts of the network to the wider internet. It is particularly vital for the client’s case, as the ASP.NET system is configured to store and maintain user data.

## Prevention

Based on the previous information, the following Defence in Depth measures should be adopted to improve preparedness and security in the event of a DDoS attack against the company network. CVE2022-38013 has been used as the example vulnerability as it is one of the more recent vulnerabilities discovered for the ASP.NET framework, showcasing the fact that just having a schedule set for the newest software updates is not enough to prevent attackers from exploiting company systems. It also one of the more severe vulnerabilities listed on the Red Hat vulnerability database, using stack overflows to bring down and otherwise affect the networking equipment and its data respectively.

### Technical Security – Updates

Investigating ASP.NET vulnerabilities on CVE often leads to the same solution – update the program to the newest available version. By creating a frequent update schedule, vulnerabilities such as CVE2022-38013 can be quickly mitigated before attackers are able to take advantage of it. The frequency of updates available to the team is mostly impacted by the timeslots at which the network is being utilised – updates should not be installed if they have the potential to impact employee workflow or customer availability, and should therefore be checked and downloaded at the end of each possible working day.

Another thing to consider is whether to update to long-term application versions or nightly releases – nightly releases will require more frequent updates and potentially cause problems as they are less tested, and long-term updates will provide a much more reliable stability at the cost of becoming outdated. Companies often release emergency updates if a particularly dangerous vulnerability is discovered, so the appropriate update framework will have to be determined based on what works best for the team. In the case of CVE2022-3813, Microsoft promptly put out an update once it was discovered.

Diagram

Description automatically generated

*Figure 1 – A robust technology update framework.*

*[Source: www.warrenaverett.com/insights/technology-upgrades/]*

*(Jones, 2023)*

### Technical Security - Zero Trust Method

When setting up firewalls and access controls, ensure that all incoming data is declined by default – this allows network administrators to slowly open the network up to new services, IP addresses, and applications carefully and minimises the chances of an attacker accessing or exploiting any of the systems on the network. In the case of CVE2022-38013, this will also prevent the attacker from triggering a stack overflow DoS as the crafted packets they send will not make it to the ASP.NET framework for execution, being stopped by an intermediary network application instead.

### Administrative Security - Backups

Ensure a frequent backup schedule is applied to all company systems and the data stored on them. The volume of backup storage and capability greatly depends on the budget available to the client, though Acronis Cyber Protect is often regarded as one of the best solutions for small businesses and should therefore be investigated by the client. As for backup network devices, in the event a server is brought offline by a DoS attack, it is considered good practice to back up services on separate servers to prevent downtime or data being overwritten in the event of a stack overflow exploit. Both hardware and software firewalls should be utilised to protect the network from the outside world and from any internal security issues as well.

Again, the solutions mostly vary on the given budget, though if the team has any experience with Microsoft Azure or AWS cloud services, they are both very effective ways of continuing data flow in the event of an attack. Smaller companies may also choose to host their entire server contents in the cloud as it reduces many of the costs and issues that arise with physical servers and on-premises datacentres (Amazon Web Services, 2023).

### Administrative Security – Security Awareness Training

Although a development team may be well-versed in good security practices, it is recommended that the team is thoroughly trained and examined on topics such as antipatterns, latest software security features, vulnerability prevention methods, application testing methods, and attack mitigation procedures. Implementing an effective training regimen may greatly reduce the chances of vulnerabilities such as CVE2022-38013 being implemented and pushed to the wider public, particularly if using customised open-source software such as ASP.NET.

# References

Amazon Web Services, 2023. *AWS Cloud Value Framework.* [Online]   
Available at: https://aws.amazon.com/economics/  
[Accessed 14 March 2023].

Bambulas, N., 2022. *How Often Should You Do Cybersecurity Awareness Training?.* [Online]   
Available at: https://www.gflesch.com/elevity-it-blog/how-often-should-you-do-cybersecurity-awareness-training  
[Accessed 14 March 2023].

CloudFlare Inc., 2023. *What is defense in depth? | Layered security.* [Online]   
Available at: https://www.cloudflare.com/en-gb/learning/security/glossary/what-is-defense-in-depth/  
[Accessed 14 March 2023].

CVE, 2020. *Vulnerability Details : CVE-2020-1161.* [Online]   
Available at: https://www.cvedetails.com/cve/CVE-2020-1161/  
[Accessed 14 March 2023].

CVE, 2020. *Vulnerability Details : CVE-2020-1597.* [Online]   
Available at: https://www.cvedetails.com/cve/CVE-2020-1597/  
[Accessed 14 March 2023].

CVE, 2021. *Vulnerability Details : CVE-2021-1723.* [Online]   
Available at: https://www.cvedetails.com/cve/CVE-2021-1723/  
[Accessed 14 March 2023].

ITGovernance, 2018. *Common causes of data breaches.* [Online]   
Available at: https://www.itgovernance.co.uk/blog/4-of-the-5-top-causes-of-data-breaches-are-because-of-human-or-process-error  
[Accessed 14 March 2023].

Jones, E., 2023. *How Often Should a Business Upgrade its Technology? [And Other Questions Answered About Improving Your Systems].* [Online]   
Available at: https://warrenaverett.com/insights/technology-upgrades/  
[Accessed 14 March 2023].

Mell, P., Shook, J. & Harang, R., 2016. Measuring and Improving the Effectiveness of Defense-in-Depth Postures. *ICSS '16: Proceedings of the 2nd Annual Industrial Control System Security Workshop,* 1(1), pp. 15-22.

Microsoft, 2022. *.NET Core and Visual Studio Denial of Service Vulnerability CVE-2022-38013.* [Online]   
Available at: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-38013  
[Accessed 14 March 2023].

W3Techs, 2023. *Historical trends in the usage statistics of server-side programming languages for websites.* [Online]   
Available at: https://w3techs.com/technologies/history\_overview/programming\_language  
[Accessed 14 March 2023].