HackDefense

Testrapport

pentest OSV2020 TK - Module PP

Kiesraad

versie 1.0 - definitief

3 augustus 2023

Deze test is uitgevoerd conform het Keurmerk Pentesten van het Centrum voor Criminaliteitspreventie en Veiligheid (CCV).

HackDefense BV

Postbus 3025
2301 DA Leiden

(071) 204 0101

`https://hackdefense.nl/`

Project

Projectnaam	pentest OSV2020 TK - Module PP
Opdrachtgever	Kiesraad
Rapport voor	Kiesraad
Projectnummer	PR23057

Documentgeschiedenis

Versie	*Datum*	*Omschrijving*
0.1	27-Jul-2023	eerste concept
0.2	01-Aug-2023	wijzigingen na interne review
1.0	03-Aug-2023	definitieve versie na review opdrachtgever

Managementsamenvatting

De Kiesraad heeft HackDefense gevraagd om een pentest uit te voeren van de kandidaatstellingssoftware voor politieke partijen (OSV2020 TK - Module PP), en om naar aanleiding daarvan aanbevelingen te doen.

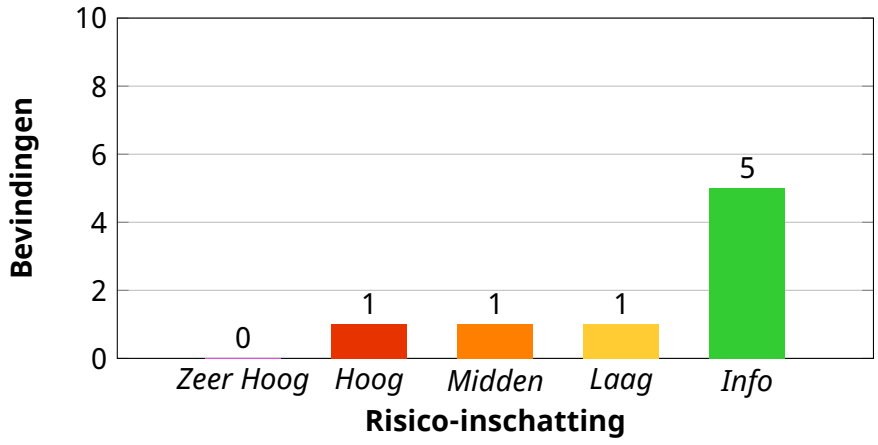
We hebben geprobeerd om kwetsbaarheden te vinden in een werkende installatie van de applicatie. Ook hebben we een configuratiereview uitgevoerd van de installatiebestanden en de bestanden die na de installatie zijn aangemaakt.

We zijn over het geheel genomen positief over de beveiliging van de applicatie. Er is duidelijk aandacht besteed aan de veiligheid. Wel zijn er twee belangrijke verbeterpunten:

- We hebben een functie gevonden die ooit ontwikkeld is, maar niet actief gebruikt wordt voor verkiezingen. Deze zat als het ware "weggestopt", maar kon nog wel worden gebruikt. Deze bleek het uploaden van onveilige bestanden toe te staan. Daarmee ontstaat het risico op ongeautoriseerde toegang tot de software.
- De software maakt een gebruikersaccount aan in Windows, met een wachtwoord dat door anderen te herleiden is. Daardoor zou een aanvaller kunnen inloggen op de computer waarop de software geïnstalleerd is. Hiervoor is een ander type account veiliger, te weten een *service account*.

Daarnaast doen we nog zes aanbevelingen die geen, of slechts een klein, risico oplossen, maar die naar onze mening wel een verbetering zouden zijn.

In dit rapport vindt u de details van ons onderzoek en onze bevindingen, en geven we technische aanbevelingen. De technische bevindingen en aanbevelingen zijn, conform uw verzoek, in het Engels opgesteld.



Inhoudsopgave

Hoofdstuk 1

Hoe hebben we getest

1.1 Uitgevoerde tests

We hebben een webapplicatietest uitgevoerd op de kandidaatstellingssoftware voor politieke partijen (OSV2020 TK - Module PP). Daarnaast hebben we een configuratiereview gedaan op de bijhorende installatie- en configuratie-bestanden.

Daarbij heeft de Kiesraad een aantal kaders meegegeven, die overeenstemmen met de aanpak van HackDefense voor het uitvoeren van beveiligingstests van webapplicaties. De enige aanvullende elementen zijn:

- een check of de aangeleverde software vrij is van bekende malware
- een extra check op eventuele *hard coded* wachtwoorden of cryptografische *salts*, met name in de installers

In bijlage B op pagina 24 staat uitgebreid beschreven hoe we elk onderdeel hebben getest.

1.2 Scope

De Kiesraad heeft HackDefense voor onderzoek het ZIP-bestand `Zonder source module PP.7z`[1] aangeleverd. Dit ZIP-bestand bestaat uit de volgende bestandsmappen:

- `configuration`: configuratie-bestanden die nodig zijn voor de installatie, bestaande uit:
  - `vapp-metdata.zip`
  - `vapp.properties`
- `documentation`: handleidingen over de installatie en werking van de applicatie, bestaande uit 15 PDF-documenten van versie `1.9.0`.
- `installer`: installatiebestand samen met verificatiehashes in SHA256- en SHA512-formaat.

[1] SHA256-hash: `CDD0674F54AD59C247ADBDAC27AD51053E69E65708BA3AC23ED6EEC1579E84DC`

    - `nl-installer-wvp-1.9.0-OSV2020-PP-installer.zip`

- `testdata`: export- en import-bestanden die gebruikt kunnen worden tijdens het testen van de applicatie, bestaande uit vier export-bestanden en twee import-bestanden.

Het installatiebestand bestaat uit drie *installers* voor Windows, Mac OS en Linux. We hebben de software geïnstalleerd op een systeem met Windows 10, een systeem met Mac OS 13.4.1 en een systeem met Debian GNU/Linux 12.1. Daarbij zijn de instructies die waren inbegrepen in de aangeleverde installatie-handleidingen exact gevolgd.

De test is vervolgens uitgevoerd op de Windows-versie.

1.3 Aanvalsperspectief

De beveiligingstest is uitgevoerd als een zogenaamde *white box*-pentest. Dat wil zeggen dat alle informatie over de systemen voor de testers beschikbaar is. Een configuratiereview van de installatie- en configuratie-bestanden was eveneens onderdeel van het onderzoek. Code review was echter geen onderdeel, maar de applicatie is wel *decompiled* om een beter beeld van de werking van de applicatie te krijgen.

De beveiliging is getest vanuit het perspectief van de *insider* (een aanvaller met fysieke toegang tot het systeem waar de software op draait). Doordat de applicatie geen authenticatie en/of autorisatie kent zijn er geen testaccounts gebruikt.

1.4 Tijdstippen en locaties

Tests en reviews zijn uitgevoerd tussen 24 juli en 28 juli 2023 binnen de lab-omgeving van HackDefense.

1.5 Onderzoeksvraag

De in dit onderzoek te beantwoorden onderzoeksvragen luidden als volgt:

1. Welke kwetsbaarheden en risico's op het gebied van informatiebeveiliging zijn te onderkennen in OSV2020 TK - Module PP?
2. In hoeverre zijn de IT-componenten waarvan OSV2020 TK - Module PP gebruikmaakt (te weten: de applicatieserversoftware en databaseserver) gehardend conform Industry Best Practices?
3. Welke maatregelen kunnen worden getroffen om de geconstateerde risico's te mitigeren?

1.6 Normenkader

In dit rapport gaan we nader in op de controls van de OWASP Top 10 (waar van toepassing in de bevindingen)[2].

Bij het testen gaan we er vanuit dat de applicatie op één systeem wordt gebruikt, zonder dat andere systemen in hetzelfde netwerk of vanaf het internet verbinding kunnen maken.

Dit is immers de gebruikssituatie bij een normale installatie. Om als netwerk-verbonden applicatie te worden gebruikt zou de gebruiker speciaal de installatie moeten aanpassen om dit mogelijk te maken.

Dat betekent o.a. dat bij de risicobepaling op basis van CVSS de *Attack Vector* (`AV`) op *Local* (`L`) is gesteld, waar dit bij een internet-verbonden applicatie *Network* (`N`) zou zijn, met bijbehorende hogere score.

[2] De status per categorie is te vinden in bijlage D op pagina 27.

Hoofdstuk 2

Onze bevindingen

2.1 Webapplicatietest

We zijn over het geheel genomen positief over de beveiliging van de applicatie. Het belangrijkste aandachtspunt heeft betrekking op het kunnen uploaden van gevaarlijke bestanden. Deze bestanden kunnen resulteren in een malware-infectie als een gebruiker deze downloadt en uitvoert[1], maar ook in onbedoelde acties van een gebruiker binnen de applicatie via een zogenoemde *Cross-Site Scripting* (XSS)-aanval. Een aanvaller dient dan wel al toegang te hebben tot de applicatie en dus ook het systeem van de applicatie aangezien deze lokaal draait.

Het probleem bevindt zich in een functie die niet gebruikt lijkt te worden tijdens de normale *workflow* van de applicatie, maar die wel in de browser bereikbaar is door de URL van de functie direct in te voeren in de adresbalk.

Hiervan hebben we meerdere voorbeelden aangetroffen: denk bijvoorbeeld aan het uploaden van een logo, en de mogelijkheid om in te loggen en het wachtwoord te wijzigen. Deze functies bestaan niet in de huidige versie van de applicatie maar zijn wel bereikbaar als je de URL weet. We hebben deze functies gevonden door het decompileren van de Java-code.

We zijn wel positief over het verwerken van gebruikersinvoer waardoor het niet mogelijk is om schadelijke code te injecteren in invoervelden. Daarnaast maakt de applicatie geen gebruik van een SQL-database en wordt alle data weggeschreven naar JSON-bestanden die geëxporteerd en geïmporteerd kunnen worden. Het is ons dan ook niet gelukt om JSON-injecties uit te voeren net zoals het niet mogelijk was om CSV-injecties uit te voeren bij de export van CSV-bestanden.

Ten slotte hebben wij een aantal bevindingen gedaan met een risico-score van Laag of Info. Deze bevindingen hebben betrekking op het ontbreken van twee securityheaders en het onnodig weggeven van informatie via metadata in bestanden en de hostnaam van het systeem in een sessiecookie.[2]

[1] Zie bevinding A.1
[2] Zie bevindingen A.3, A.4, A.5 en A.6

Een punt om te overwegen is om een *lockout password* te implementeren, dat automatisch vereist wordt na een periode van inactiviteit (bevinding A.8). Hiermee mitigeer je het riscio dat een gebruiker van de applicatie wegloopt zonder het systeem af te sluiten of te vergrendelen: iemand anders (een mogelijke aanvaller met fysieke toegang) heeft op dat moment volledig toegang tot de applicatie en de data daarin.

2.2 Configuratiereview

Wij hebben gezocht naar risicovolle data binnen installatie-bestanden en bestanden die aangemaakt zijn na de installatie. Wij zijn positief over de bestanden: wij hebben geen hardcoded wachtwoorden of *salts* aangetroffen. Verder zijn wij op zoek gegaan naar de bestanden genaamd `vars` en `dynvariables` maar zijn deze ook niet tegengekomen.

Enige punten van aandacht hebben betrekking op het feit dat de installer een lokale gebruiker toevoegt aan de Windows-installatie. Hierbij wordt gebruik gemaakt van een onveilige *Random*-methode, waardoor een aanvaller met kennis van de installatietijd het wachtwoord kan namaken en hergebruiken.[3]

Wij raden dan ook aan om gebruik te maken van lokale built-in *Service accounts* van Windows. Deze accounts hebben minder rechten, en er hoeft ook geen wachtwoord gegenereerd te worden.[4]

2.3 Overige opmerkingen

Installatie op Windows 11 leek niet te werken. Daarom is de test uitgevoerd op Windows 10.

[3] Zie bevinding A.2
[4] Zie bevinding A.7

Hoofdstuk 3

Conclusies en aanbevelingen

3.1 Conclusies

Dit project had ten doel de volgende onderzoeksvragen te beantwoorden:

1. **Welke kwetsbaarheden en risico's op het gebied van informatiebeveiliging zijn te onderkennen in OSV2020 TK - Module PP?**
   We hebben één kwetsbaarheid geïdentificeerd met een risico-score van Hoog. Het is namelijk mogelijk om allerlei bestanden te uploaden naar de applicatie zonder enige restricties. Dit kan een risico vormen voor de gebruikers van de applicatie doordat de gebruikers malware kunnen downloaden of ongewenste handelingen binnen de applicatie uitvoeren. Verder doen wij in Bijlage A zes overige bevindingen met bijbehorende aanbevelingen.
2. **In hoeverre zijn de IT-componenten waarvan OSV2020 TK - Module PP gebruikmaakt (te weten: de applicatieserversoftware en databaseserver) gehardend conform Industry Best Practices?**
   Deze module (OSV2020-PP) omvat geen applicatie- of database-servers, dus deze vraag is niet van toepassing op het onderzoeksobject.
3. **Welke maatregelen kunnen worden getroffen om de geconstateerde risico's te mitigeren?**
   In de volgende paragraaf vindt u onze aanbevelingen.

3.2 Aanbevelingen

Elke technische aanbeveling in Bijlage A gaat vergezeld van een concrete aanbeveling.[1] Samengevat is de belangrijkste daarvan de volgende.

[1] We geven zo concreet mogelijke aanbevelingen om u zo goed mogelijk op weg te helpen met het oplossen van specifieke risico's. We kunnen echter nooit uitsluiten dat een door ons gedane aanbeveling technisch niet exact werkt in uw omgeving. Verifieer altijd (door een hertest of eigen tests) of het gerapporteerde issue is opgelost na doorvoering van onze technische aanbeveling.

Pas een filtermechanisme toe die de geüploade bestanden filtert op bestandsextensie, *Content Type* en zogenaamde *magic bytes* van het bestand.[2]

Of, als deze functie niet noodzakelijk is, verwijder de gehele functie.

Voor meer details, en voor de aanbevelingen ten aanzien van de bevindingen met een lager risico verwijzen we de geïnteresseerde lezer naar de specifieke bevindingen in Bijlage A.

3.3 Mogelijk vervolgonderzoek

De limitatie in tijd (”time box”) van deze opdracht was voldoende om een goede test te kunnen uitvoeren. Elke beveiligingstest heeft ruimte voor meer onderzoek, maar in dit geval zijn we van mening dat een goede analyse heeft kunnen plaatsvinden en dat het onwaarschijnlijk is dat meer onderzoekstijd meer zinvolle informatie zou hebben opgeleverd.

Een Secure Code Review was geen onderdeel van het onderzoek en zou nog zinvolle informatie kunnen opleveren. Door het decompileren van de applicatie is het (doordat het een Java-applicatie betreft) echter wel mogelijk om veel inzicht in de code te krijgen. Daaruit zijn ook de belangrijkste bevindingen voortgekomen.

[2] Zie bevinding A.1 op pagina 13.

Bijlage A

Technische bevindingen

In deze bijlage vindt u onze specifieke bevindingen ten aanzien van het onderzoeksobject. Hierop zijn de algemene conclusies en aanbevelingen van HackDefense gebaseerd. Elke bevinding gaat gepaard met een risico-inschatting en een concreet technisch advies.

Risico-inschattingen zijn ingedeeld op basis van de volgende algemene werkwijze[1]:

- **Zeer Hoog** – Er bestaat een direct risico op verlies van systeem- of data-integriteit. We raden aan om direct actie te ondernemen om dit issue te verhelpen.
- **Hoog** – Het risico van een inbraak of lek is significant maar niet acuut; een hacker zou in het algemeen nog één element nodig hebben om tot een volledige inbraak te komen. We adviseren om zo snel mogelijk actie te ondernemen.
- **Midden** – Er is sprake van een risico, maar er is geen direct inbraakgevaar. Desondanks is sprake van een belangrijke verbetering van de beveiliging en we adviseren een relevante wijziging door te voeren bij de eerstvolgende gelegenheid voor onderhoud.
- **Laag** – Een kans om de algemene robuustheid en beveiligingsniveau van het onderzoeksobject te verbeteren. Hierbij adviseren we om een oplossing voor het issue mee te nemen in een volgende release of ander majeur onderhoudsmoment.
- **Info** – Er is geen direct beveiligingsrisico, maar we willen onze constatering wel graag met u delen. Ook kan er sprake van zijn dat een bepaalde nieuwe beveiligingsoptie niet wordt ingezet op het onderzoeksobject, en willen we u de suggestie doen om deze optie in te zetten.

We baseren onze inschatting op de meest recente versie van het *Common Vulnerability Scoring System* (CVSS) zoals dat te vinden is op `https://first.org/cvss/`.

Daarbij geldt de volgende inschaling:

[1] Ondanks het hierboven beschreven systeem en onze best mogelijke inschatting is het vaststellen van zakelijke risico's formeel geen onderdeel van ons onderzoek. We bevelen dan ook aan om uw eigen risico-inschatting te maken voordat u prioriteiten bepaalt voor het oplossen van de door ons gedane bevindingen.

CVSS-score	*CVSS-categorie*	*Onze categorie*
9,0 t/m 10,0	Critical	**Zeer Hoog**
7,0 t/m 8,9	High	**Hoog**
4,0 t/m 6,9	Medium	**Midden**
0,1 t/m 3,9	Low	**Laag**
0,0	None	**Info**

U vindt hieronder onze bevindingen in detail. Om het intern distribueren van individuele bevindingen mogelijk te maken start elke bevinding op een aparte pagina.

A.1 No upload restriction

No upload restrictions apply when uploading a logo.

Risk estimate

7,8 – High
CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

It is possible to upload any type of file through this form, including executables or HTML files. By navigating to the uploaded file, it is possible to execute the file in the context of the application which can result in a *Cross-Site Scripting* (XSS) attack. In addition, a user can also download and execute the uploaded file. This creates the possibility of the affected user's computer being completely taken over.

Both attacks are less severe in nature because no sessions or authentication are used and because the application runs locally.

Affects the URL

`https://tk-pp.osv2020.local/wvp-nl/anlage/logo-upload.xhtml`

Observation

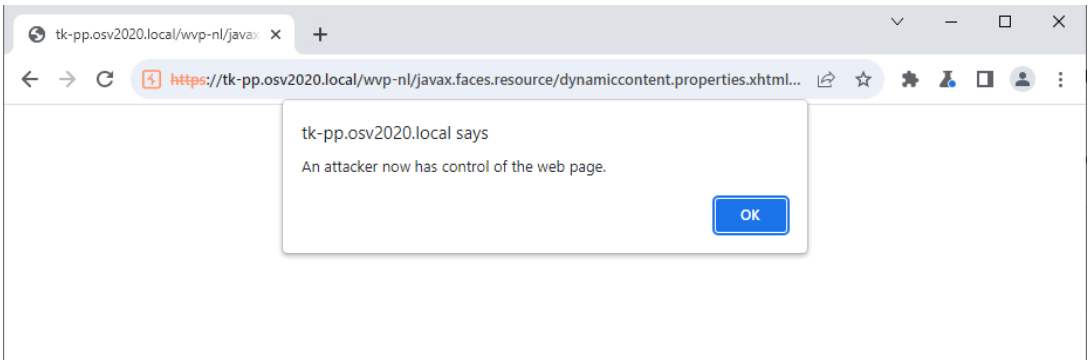
By navigating to the URL above, there is the option to upload a file with one of the following four file extensions: `.svg`, `.jpg`, `.png` en `.jpeg`.

By catching the upload request using a proxy, it is possible to change the content and *Content Type* of the file.

```
…
Content-Disposition: form-data; name="anwenderDatenEditForm:datei";
filename="test.svg"
Content-Type: text/html

<script>alert("An attacker now has control of the web page.")
</script>
------WebKitFormBoundaryzPQty9wh0Np9fA4a--
…
```

Next, you navigate to the uploaded file and because the application does not use the file extension but the provided *Content Type* as its own, the application executes the provided JavaScript code:



Recommendation

We recommend applying a filtering mechanism that filters uploaded files by file extension, *Content Type* and so called *file magic bytes* of the file.

Finally, we recommend banning the file extension *.svg* as well, since it also interprets HTML and JavaScript, thus enabling an XSS attack.

For more information, we refer to the OWASP *File upload cheat sheet.*[2]

[2] `https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html`

A.2 Use of insecure random function

When creating a user during Windows installation, an insecure random function is used to generate a password.

Risk estimate

4,9 – Medium
CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:L

When using an insecure random function that calculates the string based on time and date, it is possible to guess the password of the created user. An attacker with knowledge of the installation date and time can generate and use the same password.

Affects the file

`tools\security\addUser.bat`

Observation

The code below shows some lines from the above file. The `%RANDOM%` variable is used to grab a random character from a list of characters. However, the `%RANDOM%` variable is a so-called pseudo-random variable. In the case of CMD, this random number is derived from the current time in seconds. This means that if you know what time the script started (or approximately what time it started), you can derive the password by repeating the same algorithm with that time as the start number.

```
SET _RNDLength=14
SET _Alphanumeric=
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789~!@#$
%%*+-.:?{}_
SET _Str=%_Alphanumeric%987654321

:_LenLoopServicePassword
IF NOT "%_Str:~14%"=="" SET _Str=%_Str:~9%& SET /A _Len+=9& GOTO
_LenLoopServicePassword
SET _tmp=%_Str:~9,1%
SET /A _Len=_Len+_tmp
SET _count=0
SET _RndAlphaNum=

:_loopServicePassword
SET /a _count+=1
SET _RND=%Random%
SET /A _RND=_RND%%%_Len%
SET _RndAlphaNum=!_RndAlphaNum!!_Alphanumeric:~%_RND%,1!
IF !_count! lss %_RNDLength% goto _loopServicePassword
```

Due to the limited time for the test, we did not attempt to guess the password, thus only theoretically confirming the finding.

Note that the attribute `PasswordLastSet` is freely available in Windows, giving a good indication of the time the password was generated, so we expect that an attack on these generated passwords is very possible.

Recommendation

As with finding A.7, we recommend using a local service account. When using a local service account, there is no need to generate a password.

If a user does need to be created, then we recommend using a more secure random function instead of the `%random%` function.

A.3 Metadata in files

Files downloadable from the application contain the name and version of the software used to generate the document.

Risk estimate

2,8 – Low
CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:L/I:N/A:N

Disclosing metadata is unnecessary and makes it easier for an attacker or malicious user to detect any vulnerabilities in the specific versions of the software.

Affects the documents

all `.pdf` files

Observation

With a program like exiftool, it is possible to read metadata from documents or images. For example, in the file `Concept_H3_1_MachtigingPlaatsenAanduiding_TK2023_Nederland.pdf` we find the following:

```
ExifTool Version Number         : 12.57
Directory                       : .
File Size                       : 173 kB
File Modification Date/Time     : 2023:07:31 12:40:55+02:00
File Access Date/Time           : 2023:07:31 12:40:55+02:00
File Inode Change Date/Time     : 2023:07:31 12:40:55+02:00
File Permissions                : -rw-r--r--
File Type                       : PDF
File Type Extension             : pdf
MIME Type                       : application/pdf
PDF Version                     : 1.5
Linearized                      : No
Modify Date                     : 2023:07:31 12:40:55+02:00
Create Date                     : 2023:07:31 12:40:55+02:00
Producer                        : OpenPDF 1.3.30
Page Count                      : 3
```

Recommendation

Implement a feature that removes metadata from documents before they are downloaded through Web applications.

A.4 Hostname in cookie

The web application uses the hostname at the end in the value of the cookie.

Risk estimate

0,0 – Info

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N

Disclosing the hostname is unnecessary and gives an attacker or malicious user additional information about the system for further attacks.

Because the application runs on the system itself and a possible attacker can already request the hostname itself, this poses no risk.

Affects the site

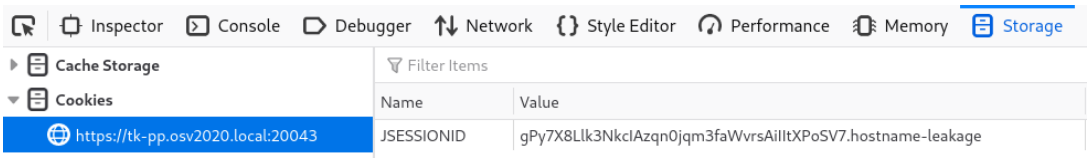
`tk-pp.osv2020.local`

Observation

The value of the cookie `JSESSIONID` ends with the hostname of the server. The hostname is added to the end of the cookie. As an example, we have changed our hostname to hostname-leakage.

The hostname will be included in the cookie:

```
hackdefense@debian:~/Downloads$ hostname
hostname-leakage
```



Recommendation

Make sure that the hostname of the system is no longer used in the value of the cookie.

A.5 No Permissions Policy

The web server does not define which browser features may or may not be used.

Risk estimate

0,0 – Info
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N

The `Permissions-Policy` defines which browser features may be used on the site. For example, it is possible to block microphone access. This allows a developer to protect the privacy of its end users. In addition to protecting privacy, a properly tuned Permissions-Policy can also block external *iframes* preventing attacks such as *Clickjacking*.

Affects the site

`tk-pp.osv2020.local`

Observation

Responses from the web server are not preceeded by a header called `Permissions-Policy`. This means that it is left up to the visitor's browser which browser features are used.

Recommendation

Add a header called `Permissions-Policy` to all web server responses. Specify which browser features the site may not use. Some examples are:

- `microphone=(), camera=()` – disables the ability to access the microphone and camera
- `microphone=(*), camera=(*)` – allows the camera and microphone to be used by the current page and any nested pages
- `microphone=(self), camera=(self)` – the camera and microphone may be used by the current page and any nested pages if they are on the same site

There are many other browser features that can be added. A complete overview can be found at `https://www.w3.org/TR/permissions-policy-1/`

A.6 No Referrer Policy

The web server does not define whether the browser may pass the page address to subsequent pages.

Risk estimate

0,0 – Info
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N

The `Referrer-Policy` determines whether and how subsequent pages may be notified that the request is coming from this page.

Affects the site

`tk-pp.osv2020.local`

Observation

Responses from the web server are not preceeded by a header called `Referrer-Policy`. This means that it is left up to the visitor's browser to determine what information is passed to subsequent pages in the `Referrer` header.

Recommnendation

Add a header called `Referrer-Policy` to all responses from the web server. For the value, you can choose from:

- `same-origin` – forward the URL of this page only to pages within the same site
- `no-referrer-when-downgrade` – never forward the URL of this page to pages that are not secured with HTTPS
- `no-referrer` – never forward the URL of this page

There are several other options that can be chosen as values. A full list can be found at `https://www.w3.org/TR/referrer-policy/`.

A.7 Incorrect use of service account

The application creates a local user account instead of using a built-in local service account.

Risk estimate

0,0 – Info
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N

Creating a user account broadens the attack surface for an attacker. Combined with the use of an insecure random function when generating the password, this can result in taking over the user account the application is running on.[3]

Affects the file

`tools\security\addUser.bat`

Observation

The code below from the above file shows a user being created. In addition, this user is reflected in the system's home screen.

```
::ECHO add user "%SERVICE_USER%", password "!NEW_SERVICE_PW!"
net user "%SERVICE_USER%" "!NEW_SERVICE_PW!" /ADD

ECHO user "%SERVICE_USER%" successfully added

ECHO set password never expires for user "%SERVICE_USER%"
wmic path Win32_UserAccount where name="%SERVICE_USER%" set
PasswordExpires=false

ECHO set password not changeable for user "%SERVICE_USER%"
wmic path Win32_UserAccount where name="%SERVICE_USER%" set
PasswordChangeable=false
```

Recommendation

We recommend using built-in local service accounts. The text below provides additional explanation of a built-in local service account:

[3] See finding A.2 on page 15

”The built-in Local Service user account has fewer access privileges on the computer than the Network Service user account, and those user privileges are limited to the local computer. Use the Local Service user account if the worker process does not require access outside the server on which it is running.”

A.8 No lockout mechanism

The application does not lock itself after a period of inactivity.

Risk estimate

0,0 – Info

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N

If a user leaves the computer without locking the screen, others may alter data or the process.

Affects the systems

n/a

Observation

The application continues to be usable even after extended periods of inactivity.

Recommendation

Require the user to set a ”lockout password” and automatically switch the screen to a version that requires that password after a period of inactivity of, for example, 2 or 3 minutes.

Alternatively, point out to the user that it is important to always lock access to the desktop when leaving the computer.

Bijlage B

Onze aanpak

B.1 Algemeen

In het algemeen geldt voor de onderzoeken en tests van HackDefense dat uitvoer van tooling voor ons niet leidend is. Tooling is een hulpmiddel, het is het gereedschap van de vakman. Conclusies worden getrokken door de vakmensen zelf, voor wie een goed begrip van de werking van het te testen object het belangrijkste element van een beveiligingstoets is. De uitvoer van de tooling is daarom altijd handmatig geverifieerd. Ook zijn tests die niet geautomatiseerd uitvoerbaar zijn met de hand uitgevoerd.

Conform de handleiding (die zich in het ZIP-bestand bevond dat is aangeleverd) hebben we de software geïnstalleerd op de drie virtuele systemen. De besturingsystemen van deze systemen bestonden uit: Windows 10, Mac OS 13.4.1 en Debian 12.1.

B.2 Webapplicatietest

Allereerst is er een poortscan en een kwetsbaarheidsscan uitgevoerd van de URL's en IP-adressen in scope. Daarbij is gebruik gemaakt van *Nmap*, *Nessus*, *Nikto*, *Gobuster* en de *Active Scan*-component van *BurpSuite Pro*.

Tegelijkertijd hebben we met handmatige tests een beeld gevormd van de werking van de webapplicatie. Onze basistool daarbij is de *intercepting proxy* van *BurpSuite Pro*.

De resultaten van de scanner zijn handmatig geverifieerd. Daarbij zijn ook ondersteunende scan-modules van BurpSuite Pro gebruikt voor de handmatige test (waarbij bijvoorbeeld voor de tester inzichtelijk wordt gemaakt waar gebruikersinvoer terugkomt in de uitvoer), zodat we handmatig ook hebben kunnen testen op kwetsbaarheden die de scanner mogelijk niet heeft gedetecteerd c.q. niet heeft kunnen detecteren. Denk hierbij aan kwetsbaarheden zoals: *Cross-Site Scripting* (XSS), *SQL Injection*, *XML External Entity injection* (XXE) en *Server-Side Template Injection* (SSTI).

Naast de resultaten van de scanner, hebben wij handmatig alle beveiligingsrisico's van een webapplicatie gecontroleerd met als basis de *OWASP Top 10*.[1] Wij zijn via de diverse

[1] `https://owasp.org/www-project-top-ten/`

rollen binnen de applicatie gestart met een uitgebreide analyse op de authenticatie, autorisatie en sessiemanagement. Hierbij is er bijvoorbeeld gekeken of het mogelijk is om als gebruiker ongeautoriseerd toegang te krijgen tot data of functionaliteiten binnen de applicatie of zelfs zonder enige vorm van authenticatie.

Vervolgens hebben wij gecontroleerd of de webapplicatie niet onnodig informatie weggeeft via bestanden, *Stack-traces* of response headers en is er gekeken of de applicatie gebruik maakt van verouderde of kwetsbare software. Verder is er een analyse gedaan op specifiek functionaliteiten binnen de applicatie zoals het uploaden van bestanden, het wijzigen van een wachtwoord en/of e-mail en het gebruik van *WebSockets*.

Tot slot hebben wij een cryptografische analyse van de SSL-/TLS-beveiliging uitgevoerd en is er gekeken naar het (correct) gebruik van *Security Headers*.

B.3 Configuratiereview

We voeren een analyse uit van alle installatie- en configuratie-bestanden. Denk hierbij aan de *installers*, bestanden/mappen die aangemaakt zijn tijdens de installatie, maar ook de *decompiled* software. Tijdens deze analyse is er gekeken dat de bestanden `vars` en `dynvariables` niet aanwezig zijn. Verder is er in de bestanden en software gezocht naar hardcoded wachtwoorden of andere gevoelige data zoals *salts*.

Via de *decompiled* software is er ook een duidelijker beeld gecreëerd over de werking van de applicatie, maar is er ook gekeken naar functies en bestanden die niet zichtbaar zijn tijdens de 'normale' flow van de applicatie. Alle functies zijn in kaart gebracht en vervolgens dieper geanalyseerd door te kijken welke beveiligingsmaatregelen getroffen zijn en of deze omzeild kunnen worden. Verder is er ook gekeken naar het gebruik van zwakke hash- en encryptie-methodes.

Voor een complete lijst van alle gecontroleerde configuratiebestanden verwijzen wij naar Bijlage C op pagina 26.

Bijlage C

Lijst configuratiebestanden

Applicatie-bestanden

```
config/custom.cli                              config/elect_config.sh
shortcuts/startBrowser.sh                      tools/chmodFiles.sh
tools/copy_jdk.sh                              tools/copy_metadata.sh
tools/createDirectories.sh                     tools/generateCertificate.sh
tools/security/addAll.sh                       tools/security/addCa.sh
tools/security/addFirefoxPolicies.sh           tools/security/addHost.sh
tools/security/createCredentialStore.sh        tools/security/maskCredentialStore.sh
tools/security/removeAll.sh                    tools/security/removeCASystem.sh
tools/security/removeCAUser.sh                 tools/security/removeHost.sh
tools/security/runScriptByAdministrator.sh     tools/generateCertificate.sh
tools/uninstall/cleanup.sh                     tools/uninstall/cleanup.xml
tools/uninstall/uninstaller.sh                 start.bat
config/elect_config.bat                        config/initservice-configuration.bat
tools/security/addAll.bat                      tools/security/addFirefoxPolicies.bat
tools/security/addUser.bat                     tools/security/createCredentialStore.bat
tools/security/maskCredentialStore.bat         tools/security/removeAll.bat
tools/security/resetACL.bat                    tools/security/runScriptByAdministrator.bat
tools/service/configureService.bat             tools/service/installService.bat
tools/service/removeService.bat                tools/service/service-configuration.bat
tools/service/startService.bat                 tools/service/stopService.bat
tools/uninstall/cleanUpAll.bat                 tools/uninstall/removeData.bat
tools/uninstall/uninstaller.bat                tools/copy_jdk.bat
tools/copy_metadata.bat                        tools/createDirectories.bat
tools/generateCertificate.bat
```

Installatie-bestanden

```
installer-windows.exe  installer-linux-hdpi.sh
installer-linux.sh
```

Decompiled software

```
deployments/nl-wvp-war.war
```

Bijlage D

OWASP top 10

De applicatie is getoetst aan de top 10 OWASP categorieën.

De kolom ”Status” in onderstaande tabel geeft het volgende weer:

x	geen afwijkingen waargenomen
A.x	afwijking waargenomen, met impact (refereert aan nummer bevinding in Bijlage A)
–	niet van toepassing of niet in scope

De tekst van de controls is te vinden op `https://owasp.org/www-project-top-ten/`.

Categorie	*Status*	*Opmerking*
A01:2021-Broken Access Control	-	De applicatie kent geen autorisatiemanagement
A02:2021-Cryptographic Failures	**A.2**	
A03:2021-Injection	**x**	
A04:2021-Insecure Design	**A.1** & **A.3** & **A.4**	
A05:2021-Security Misconfiguration	**A.5** & **A.6** & **A.7**	
A06:2021-Vulnerable and Outdated Components	**x**	
A07:2021-Identification and Authentication Failures	**A.8**	De applicatie kent geen authenticatiemanagement
A08:2021-Software and Data Integrity Failures	**x**	
A09:2021-Security Logging and Monitoring Failures	**x**	
A10:2021-Server-Side Request Forgery	**x**	