Veilige weergave van boefjes-data in de KAT web-interface

Aanleiding

De KAT web-interface moet in een oogopslag alle benodigde data tonen over een gedane observatie (bevinding/kwetsbaarheid). Een gebruiker kan hierdoor snel en adequaat de output van boefjes analyseren om te bepalen of een observatie legitiem is. Dit vereist dat de orginele (RAW) data van boefjes in de web-interface beschikbaar is. [1]

De weergave van deze boefjes-data wordt momenteel beschikbaar gesteld via een los ZIP bestand. Dit bestand kan door de gebruiker worden gedownload voor analyse.
De wens is om deze data (deels) direct in de web-interface beschikbaar te maken om het uitvoeren van extra handelingen te beperken. Het gaat hier om het downloaden en uitpakken van het bestand, bij voorkeur binnen een afgeschermde omgeving. Deze wens brengt diverse beveiligingsvraagstukken met zich mee.

Het risico is dat in de output van boefjes mogelijk kwaadaardige bestanden zitten. De vraag is dan ook hoe de inhoud met een zo laag mogelijk risico toegankelijk gemaakt kan worden.

Dit document onderzoekt een aantal mogelijke oplossingen om de output van boefjes zonder onnodig risico voor de gebruiker en het systeem weer te geven. Hierbij wordt gebruik gemaakt van de security best practices.

Wensen & speelveld

Er zijn meerdere belangen waar rekening mee moet worden gehouden. De bruikbaarheid van KAT door gebruikers, de kernwaarden van KAT en de security best practices. Elk van deze invalshoeken heeft wensen en beperkingen die van invloed zijn op de keuze van weergave voor boefjes-data.

Gebruikersperspectief

- Weinig extra klikken om de benodigde informatie op te vragen.
- Duidelijk overzicht van alle data, met weinig afleiding.
- Juiste en correcte data beschikbaar voor analyse.
- Eenvoud in bruikbaarheid.

KAT perspectief

- De software moet schaalbaar blijven, zowel als er gebruik wordt gemaakt van 1 server, als meerdere servers.
- Het moet een standalone oplossing zijn, die onafhankelijk van de situatie of context bruikbaar is.
- Integratie in de bestaande User Interface (UI) van OpenKAT.
- KAT instantie draait binnen een netwerk of een omgeving.
- Betaalbare oplossing, die bij voorkeur zoveel mogelijk werkt zonder licenties en additionele kosten.
- De omgeving moet beheerbaar blijven voor de eigenaar van het systeem (systeembeheerders).
- Security best practices worden gevolgd en geïmplementeerd.[2]

Security perspectief

- Zero trust, vertrouw andere systemen niet.[3]
- Invoer uit externe bronnen (systemen) is onveilig.
- Vertrouw je eigen gebruikers niet; lever de output zo onschadelijk mogelijk aan zodat een onwetende gebruiker moeite moet doen om het aan de gang te krijgen.

---

1 Github, Ministerie VWS, nl-kat-rocky, ‘As a user I would like to see the content of the logs without downloading it’ Issue: #152. *https://github.com/minvws/nl-kat-rocky/issues/152*

2 OWASP Foundation, OWASP Secure Coding Practices-Quick Reference Guide, *https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/*

3 Cloudflare, What is Zero Trust Security, *https://www.cloudflare.com/learning/security/glossary/what-is-zero-trust/*

- Gebruikers zijn een zwakke schakel; beheerders maken onbedoelde configuratiefouten in de KAT installatie, waardoor onbedoelde kwetsbaarheden in de serverconfiguratie ontstaan.
- Privacy by design – data wordt niet naar externe partijen verstuurd, behalve om via een API-key gebruik te maken van de diensten van het betreffende boefje.
- Security by design – neem security best practices mee bij het ontwikkelen van software.

Het is van belang dat er een balans is tussen het gebruikersgemak van de web-interface en de risico's binnen beveiliging.

Technische mogelijkheden voor data weergave

De volgende mogelijkheden worden onderzocht op de geschiktheid binnen de KAT web-interface.

- **Download als zip –** eventueel als password-protected ZIP downloads.
- **Weergave op de pagina als plaintext –** alle boefjes-data wordt opgeslagen in tekst formaat (.txt) en kan vervolgens als plaintext worden ingeladen in de web-interface. Dit maakt uitvoerbare bestanden onuitvoerbaar, maar wel leesbaar.
- **Weergave op de pagina ingeladen vanaf een ander domein –** een extra server binnen het netwerk wordt gebruikt om potentiële kwaadaardige data in te laden.
- **Inline weergave voor afbeeldingen** – afbeeldingen van onbetrouwbare bronnen moeten op een veilige manier worden ingeladen in de KAT web-interface.

Voor elk van bovenstaande mogelijkheden wordt in de volgende sectie kort de risico's en technische implementatie besproken die hierbij van toepassing zijn en hoe risico's mogelijk gemitigeerd kunnen worden.

Download als zip[4]

Risico's:

- Bestanden die als ZIP bestand worden gedownload via de KAT web-interface bevatten mogelijk kwaadaardige software. In een ideale situatie wil je de inhoud van dit ZIP bestand altijd in een afgeschermde omgeving openen voor analyse. Dit mitigeert de impact op bijvoorbeeld de confidentialiteit, integriteit of beschikbaarheid van eventuele data doordat kwaadaardige code minder kans heeft om impact te veroorzaken op andere data en/of systemen. Enkel is deze mitigatie volledig afhankelijk van het gedrag van de gebruiker.
- Impact op de KAT installatieserver bij misconfiguratie van de server. Afhankelijk van de configuratie kan een gebruiker onbedoeld ZIP bestanden beschikbaar maken ter download aan andere onbedoelde gebruikers.
- Het lokaal gedownloade bestand kan impact geven op de computer van de gebruiker. Dit is afhankelijk van de mate van isolatie in de omgeving waarbinnen het ZIP bestand wordt geopend en geanalyseerd.
- Potentiële impact op de netwerkomgeving waarbinnen de gebruikerscomputer zich bevindt. Dit is afhankelijk van netwerksegmentatie en de mate van isolatie in de omgeving waarbinnen kwaadaardige bestanden worden geanalyseerd.

Technische implementatie & risicobeperking:

- Bestanden worden als ZIP bestand aangeleverd zodat kwaadaardige code zoals JavaScript, executables of JPG bestanden niet direct aan te roepen zijn op het moment dat het ZIP bestand extern (onbedoeld) bereikbaar is gemaakt via de KAT instantie.
- Bestanden kunnen password-protected worden aangeboden zodat virusscanners het bestand niet direct verwijderen en het per ongeluk uitvoeren van kwaadaardige code uit te voeren wordt voorkomen.
- Bestanden die in een .ZIP bestand worden opgeslagen kunnen de extra extensie .txt krijgen op alle bestanden, om te zorgen dat deze niet per ongeluk worden uitgevoerd.

4 ***Opmerking**: Elke keer als hieronder wordt verwezen naar een ZIP bestand wordt hiermee bedoeld dat de inhoud van dit ZIP bestand potentieel kwaadaardige code en/of bestanden bevat. Deze data kan schade aanrichten op het systeem of de (netwerk)omgeving.*

Weergave op de pagina als plaintext [5]

Risico's:

- Geen, mits de implementatie in de KAT broncode op de juiste manier wordt uitgevoerd om uitvoerbaarheid van kwaadaardige code te beperken.
- Gebruik van de *InnerHTML()* tag moet op alle fronten voorkomen worden in verband met de mogelijkheid tot Cross-Site-Scripting aanvallen.
- Het gebruik van dubbele bestaande bestandsextensies (bestand.js.txt of bestand.exe.txt) moet worden voorkomen om eventuele fouten (en dus beveiligingsproblemen) in de parsing van het bestand te voorkomen.[6]

Technische implementatie & risicobeperking:

- Voor alle bestandsformaten wordt input onuitvoerbaar gemaakt door het weg te schrijven naar tekstbestanden, zowel voor leesbare (bijv. broncode) als onleesbare (bijv. afbeeldingen) data. De tekstbestanden kunnen vervolgens in de web-interface worden ingeladen.
- Voor onleesbare bestanden kan de data worden weergegeven in HEX-formaat.
- Directe weergave werkt alleen voor data die 'leesbaar' is via plaintext, bijv. JavaScript code. Onleesbare bestandsformaten, zoals afbeeldingen, zullen minder baat hebben bij deze weergave.
- Als extra beveiligingsmaatregel dienen de volgende HTTP headers gebruikt te worden. Dit minimaliseert de kans dat de data alsnog uitvoerbaar gemaakt wordt binnen de browser. [7] [8]
  - Content-Type: plain/text
  - X-Content-Type-Options: nosniff
- Gebruik van de *.innerText()* of *.textContent()* functie is vereist (NIET *.InnerHTML()*, ivm XSS kwetsbaarheden) in combinatie met non-executable tags (e.g. divs, spans, buttons, etc.). [9] [10] [11]

Weergave op de pagina ingeladen vanaf een ander domein

Risico's:

- Vereist extra tijd, complexiteit en resources voor het opzetten van een extra KAT server binnen het domein. Dit brengt ook meer mogelijkheden op misconfiguraties met zich mee.
- Plaatsing en segmentatie van de losse server kan nog steeds impact geven op het netwerk waarbinnen dit draait, indien er misconfiguraties plaatsvinden. Denk aan onvoldoende firewalling en segmentatie, maar ook aan verkeerde rechten geven aan mappen op de server. Als deze server (onbedoeld) direct aan het internet wordt gekoppeld is de data die hierop aanwezig is potentieel extern benaderbaar en valt dan ook te misbruiken door aanvallers vanaf het internet.
- In het geval dat deze extra server gecompromitteerd wordt, kunnen hiermee ook aanvallen uitgevoerd worden tegen andere systemen in dit netwerk en mogelijk tegen de KAT instantie. In het ergste geval kan de integriteit van KAT data niet meer worden gewaarborgd. Dit vereist wel een samenloop van meerdere omstandigheden en misconfiguraties die plaatsvinden.

Technische implementatie & risicobeperking:

- Vereist dat de KAT instantie gebruik maakt van browser policies om data op een veilige manier van externe bronnen te downloaden.

---

5 Hong J, Jeong D, Kim S-W. Classifying Malicious Documents on the Basis of Plain-Text Features: Problem, Solution, and Experiences. *Applied Sciences*. 2022; 12(8):4088. *https://doi.org/10.3390/app12084088*

6 OWASP, OWASP Cheat Sheet Series, File Upload Cheat Sheet *https://cheatsheetseries.owasp.org/cheatsheets/File_Upload_Cheat_Sheet.html*

7 Mozilla MSDN Docs, Content-Type, *https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Type*

8 Mozilla MSDN Docs, Content-Type-Options, *https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options*

9 OWASP, DOM based XSS Prevention Cheat Sheet, *https://cheatsheetseries.owasp.org/cheatsheets/DOM_based_XSS_Prevention_Cheat_Sheet.html*

10 EduCBA, InnerText vs InnerHTML, *https://www.educba.com/innertext-vs-innerhtml/*

11 Dev.to, JavaScript innerHTML, innerText, and textContent, *https://dev.to/4myc/javascript-innerhtml-innertext-and-textcontent-ih*

- Vereist dat resources en calls naar externe bronnen op het internet actief geblokkeerd worden.
- Risicobeperking is lastig aangezien elke KAT installatie op een andere omgeving draait, met andere segmentaties. Hierdoor is het complex om een generieke oplossing te ontwikkelen.

Inline weergave voor afbeeldingen

Risico's:

- Boefjes kunnen random afbeeldingen (o.a. GIF, PNG, BMP, JPEG) verzamelen als bewijslast. Deze afbeeldingen kunnen in de KAT web-interface worden ingeladen. Het kan zijn dat een afbeelding kwaadaardige code bevat. Door het tonen van deze afbeeldingen in de KAT web-interface kan er potentieel kwaadaardige code in de browser van de gebruiker worden uitgevoerd.
- Kwaadaardige afbeeldingen zijn veelal 'dropper' files, die gebruikt worden om de malware te downloaden. Veelal wordt kwaadaardige code aan het einde van een afbeelding toegevoegd. Detectie is in deze gevallen relatief simpel door het zoeken op specifieke byte-codes in de afbeelding. EXIF tags worden ook regelmatig gebruikt voor kwaadaardige code. [12]
- Obfuscatietechnieken kunnen het detecteren van kwaadaardige code lastig maken.
- Een afbeelding kan ongezien externe calls naar domeinen/IP-adressen op het internet bevatten.

Technische implementatie & risicobeperking

- Boefjes verzamelen plekken van diverse plekken informatie, waarbij afbeeldingen ook kwaadaardige code kunnen bevatten. Dit vereist dat browsers altijd op de meest recente versie zijn om eventuele kwetsbaarheden (zero-days) in de browser sandbox te mitigeren. Dit is enkel lastig omdat eindgebruikers niet altijd invloed hebben op het patch-beleid van een organisatie, danwel slecht zijn in het installeren van hun updates.
- Blokkeer standaard externe calls naar het internet en sta enkel geautoriseerde domeinen/IP-adressen toe.

Advies

Het aanbieden van data via een extra server in het netwerk brengt extra beheertaken en potentiële beveiligingsincidenten met zich mee. De kans is reëel dat er onbedoelde fouten worden gemaakt in de serverconfiguratie of bij het afschermen en segmenteren van het netwerk. Dit kan nadelige gevolgen hebben en beveiligingsrisico's met zich mee brengen voor de betreffende organisatie.

De oplossing met de minste impact en kans op (misconfiguratie)fouten is het exporteren van boefjes-data naar plaintext om te zorgen dat eventuele kwaadaardige code onuitvoerbaar wordt gemaakt. Dit werkt met name goed voor bestandsextensies waarvan de inhoud leesbaar is. Voor een onleesbare bestandsextensies kan dit worden opgelost via een weergave in HEX-formaat, danwel in combinatie met het blijven aanbieden van de boefjes-data in een los ZIP bestand.

Voor een gebruiker blijft het advies om, uit voorzorgsmaatregelen, analyse in een afgeschermde omgeving uit te voeren. Hiervoor is het raadzaam om de boefjes-data beschikbaar te houden via de KAT web-interface. Hier blijft het risico bestaan dat een analist onzorgvuldig met potentiële kwaadaardige code omgaat. Hiervoor dient een organisatie zelf maatregelen te nemen om deze risico's zoveel mogelijk te beperken.

Vanuit de KAT web-interface kan een configuratiekeuze worden aangeboden om te kiezen of boefjes-data standaard als plaintext formaat, danwel als ZIP bestand wordt aangeboden.

12 ReversingLabs, Malware in Images: When you can't see "the whole picture", *https://www.reversinglabs.com/blog/malware-in-images*