# Executive Summary

This report contains the findings of the limited penetration test (pentest) conducted against the Safe Bank primary website and corporate domain environment, conducted between 1st and 2nd Aug 2024. The assessment comprised of both application and infrastructure testing.

The assessment discovered a number of deviations from known good security practice at time of testing. Whilst the environment and systems were found to be up-to-date and patched against the latest security threats, a number of logical flaws and deviations from modern security practices led to a payload being executed by a Safe Bank user, giving the tester remote access to the Safe Bank environment and conditions were enumerated which could allow an attacker full control of the Safe Bank domain.

In its current state, the Safe Bank domain is assessed likely at significant risk of widespread compromise should an attacker gain a foothold in the environment.

The assessment performed was limited in nature and in addition to the technical recommendations already laid out in this report, a more complete and comprehensive assessment of a number of key areas is recommended.

# Technical Summary

A path to full domain compromise against the safebank.local Active Directory Domain was discovered during the assessment. This was achieved by delivery of a Macro enabled Word document via the Safe Bank web application.

This was in turn opened and executed by a Domain user resulting in the successful delivery of a Meterpreter implant in memory of a user’s computer which called back to an attacker-controlled Command and Control (C2) server. This attack was able to avoid all existing security apparatus in the Safe Bank environment and no indication of a successful detection was encountered during the assessment.

In addition to gaining access to the environment, it was discovered that the user who had executed the initial access payload was also a member of a highly privileged built-in security group of the Safe Bank domain, which would also afford an attacker the ability to compromise a Domain Controller and gain access to credential material which could be used for full compromise of the domain.

The time limited nature of the assessment means not all paths for exploitation were enumerated during the testing window and it is strongly recommended that a more complete, bespoke assessment of the domain and the security apparatus present in the environment is perform.

# Strategic Summary

A number of the flaws discovered and exploited in the Safe Bank environment come from either logical misconfiguration of the Active Directory environment or the lack of privileged account segregation that appears to be occurring. Additionally, it appeared that existing Antivirus and XDR solutions in use were ineffective at both detecting and blocking the threat posed by a would-be attacker.

Additional, more complete testing should be performed to discover other flaws and a planning for significant remediation work considered to bring the environment in line with modern security practice.

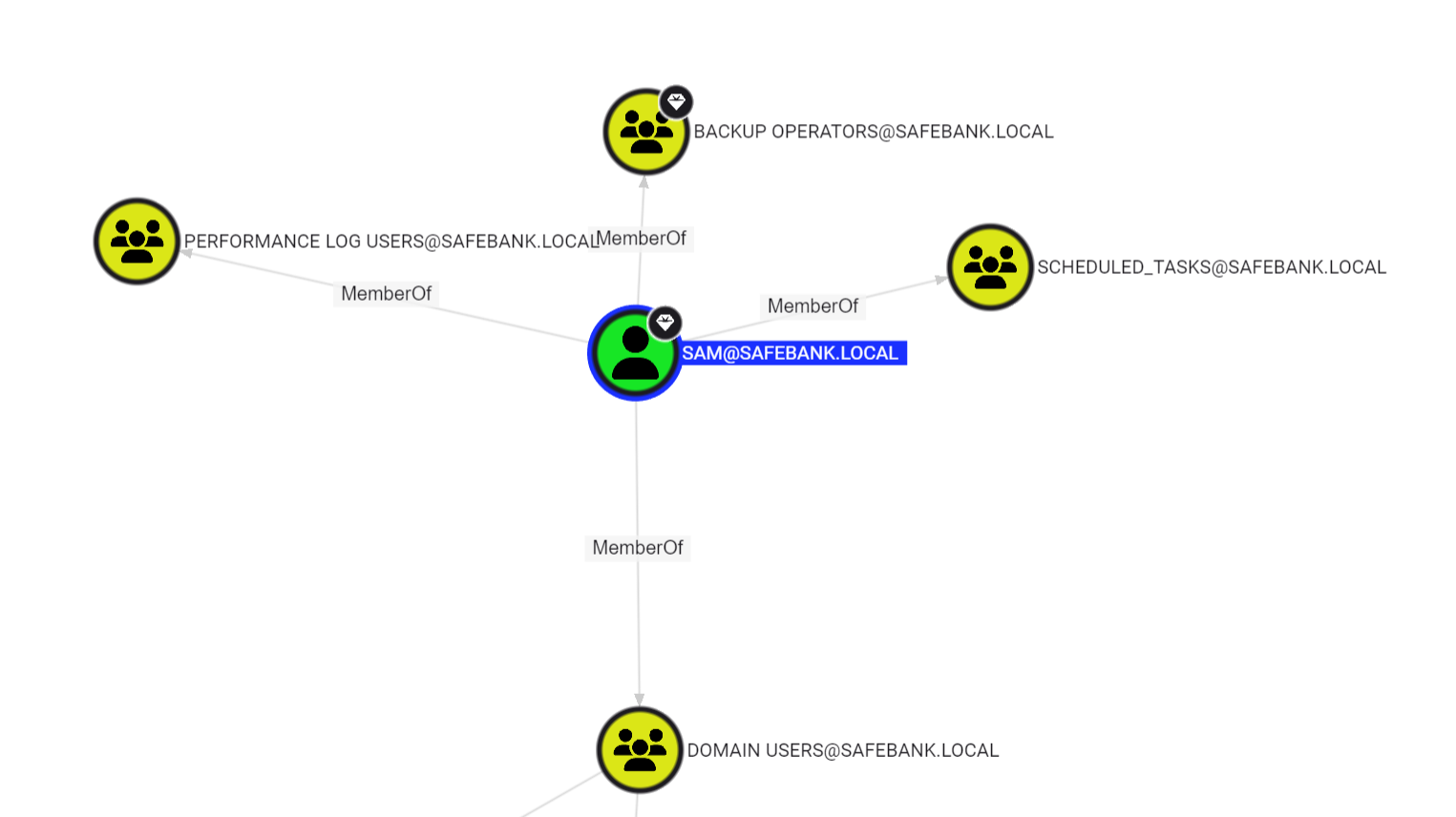
# Issues

## 001-Path to Domain Admin (Risk: Critical)

### Description

During the assessment, a payload introduced to the environment via the Safe Bank web application, in conjunction with the 003-Unrestricted File Upload issue, was opened and executed by a domain user with Backup Operator permissions in the domain.

This security group is considered a high value target for attackers as it allows for retrieval of Security, System and SAM registry hives from local and remote computers and can facilitate lateral movement and privilege escalation inside the domain, including retrieval of sensitive password hashes from the Domain Controller.



### Recommendation

High privileged groups, such as Backup Operator, should be restricted to administrative accounts and a degree of separation applied to their usage. I.e. They should not be used for day-to-day tasks and ideally only perform administrative actions from workstations setup to perform privileged actions, also known as Privileged Access Workstations (PAWs), and hardened in a manner commensurate with the level of administrative permission they support in order to make them resilient to attack.

It is recommended that a full Active Directory review is performed to confirm if other user accounts are in daily use that also are members of sensitive security groups or has unnecessary permissions assigned which could be useful to an attacker.

Further advice on hardening Active Directory is available from Microsoft[[1]](#footnote-0).

## 002-Lack of Effective EDR / Monitoring (Risk: High)

### Description

Whilst it was noted that up-to-date Antivirus (AV) software in the form of Microsoft Defender was running in the Safe Bank environment, as well as Wazuh, an Open Source XDR solution – it was noted that no security products in use blocked or detected the malicious Office Macro payload used to gain initial access in the Safe Bank domain. No intervention was performed by the Safe Bank security team, suggesting the attack itself was not detected.

Whilst the payload in of itself made use of heavy obfuscation in order to execute an implant in memory and bypass modern AV solutions, it still makes use of suspicious Win32 API calls in order for that execution to occur, including VirtualAlloc, RtlMoveMemory and CreateThread. All of which should be observable and provide a suitable Indicator of Compromise (IoC) for a defensive team to act upon. The payload can be analyzed using oletools.py to confirm the use of these API’s:

| +----------+--------------------+---------------------------------------------+  |Type |Keyword |Description |  +----------+--------------------+---------------------------------------------+  |AutoExec |AutoOpen |Runs when the Word document is opened |  |Suspicious|Open |May open a file |  |Suspicious|Application.Visible |May hide the application |  |Suspicious|CreateObject |May create an OLE object |  |Suspicious|Lib |May run code from a DLL |  |Suspicious|Chr |May attempt to obfuscate specific strings |  | | |(use option --deobf to deobfuscate) |  |Suspicious|Hex Strings |Hex-encoded strings were detected, may be |  | | |used to obfuscate strings (option --decode to|  | | |see all) |  |Suspicious|Base64 Strings |Base64-encoded strings were detected, may be |  | | |used to obfuscate strings (option --decode to|  | | |see all) |  |Suspicious|VBA obfuscated |VBA string expressions were detected, may be |  | |Strings |used to obfuscate strings (option --decode to|  | | |see all) |  |IOC |OleAut32.dll |Executable file name |  |Hex String|ThisDoc |54686973446f63 |  |Hex String|ument.DefineBarter |756d656e742e446566696e65426172746572 |  |Hex String|licat |6c69636174 |  |Hex String|**kerne** |6b65726e65 |  |Hex String|**rtualAlloc** |727475616c416c6c6f63 |  |Hex String|**nel32**  |6e656c3332 |  |Hex String|**VirtualAl**  |5669727475616c416c |  |Hex String|**kernel**  |6b65726e656c |  |Hex String|**RtlMove** |52746c4d6f7665 |  |Hex String|**Memory** |4d656d6f7279 |  |Hex String|**Crea**  |43726561 |  |Hex String|**teThread** |7465546872656164 |  |Base64 |'..&j' |LoadLibraryA |  |String | | |  |VBA string|b'ThisDoc'b'ument.De|DisplayLbo("54686973446f63") & DisplayLbo("75|  | |fineBarter' |6d656e742e446566696e65426172746572") |  |VBA string|b'Wor' |DisplayLbo("576f72") |  |VBA string|b'd.A'b'pp' |DisplayLbo("642e41") & DisplayLbo("7070") |  |VBA string|b'licat'b'ion' |DisplayLbo("6c69636174") & |  | | |DisplayLbo("696f6e") |  |VBA string|b'**kerne**'b'**l32**' |DisplayLbo("6b65726e65") & |  | | |DisplayLbo("6c3332") |  |VBA string|b'**Vi**'b'**rtualAlloc**' |DisplayLbo("5669") & |  | | |DisplayLbo("727475616c416c6c6f63") |  |VBA string|b'**ker**'b'**nel32**' |DisplayLbo("6b6572") & |  | | |DisplayLbo("6e656c3332") |  |VBA string|b'**VirtualAl**'b'**loc**' |DisplayLbo("5669727475616c416c") & |  | | |DisplayLbo("6c6f63") |  |VBA string|b'**kernel**'b'**32**' |DisplayLbo("6b65726e656c") & |  | | |DisplayLbo("3332") |  |VBA string|b'**RtlMove**'b'**Memory**' |DisplayLbo("52746c4d6f7665") & |  | | |DisplayLbo("4d656d6f7279") |  |VBA string|b'**Crea**'b'**teThread**' |DisplayLbo("43726561") & |  | | |DisplayLbo("7465546872656164") |  +----------+--------------------+---------------------------------------------+ |
| --- |

### Recommendation

Consideration should be given to performing an assessment of the Safe Bank EDR and SIEM solutions to confirm that common vectors of attack can be both detected and intervention performed within a timely fashion.

This should demonstrate whether the existing solutions are performing effectively and inform decision making as to whether to invest in another commercial solution.

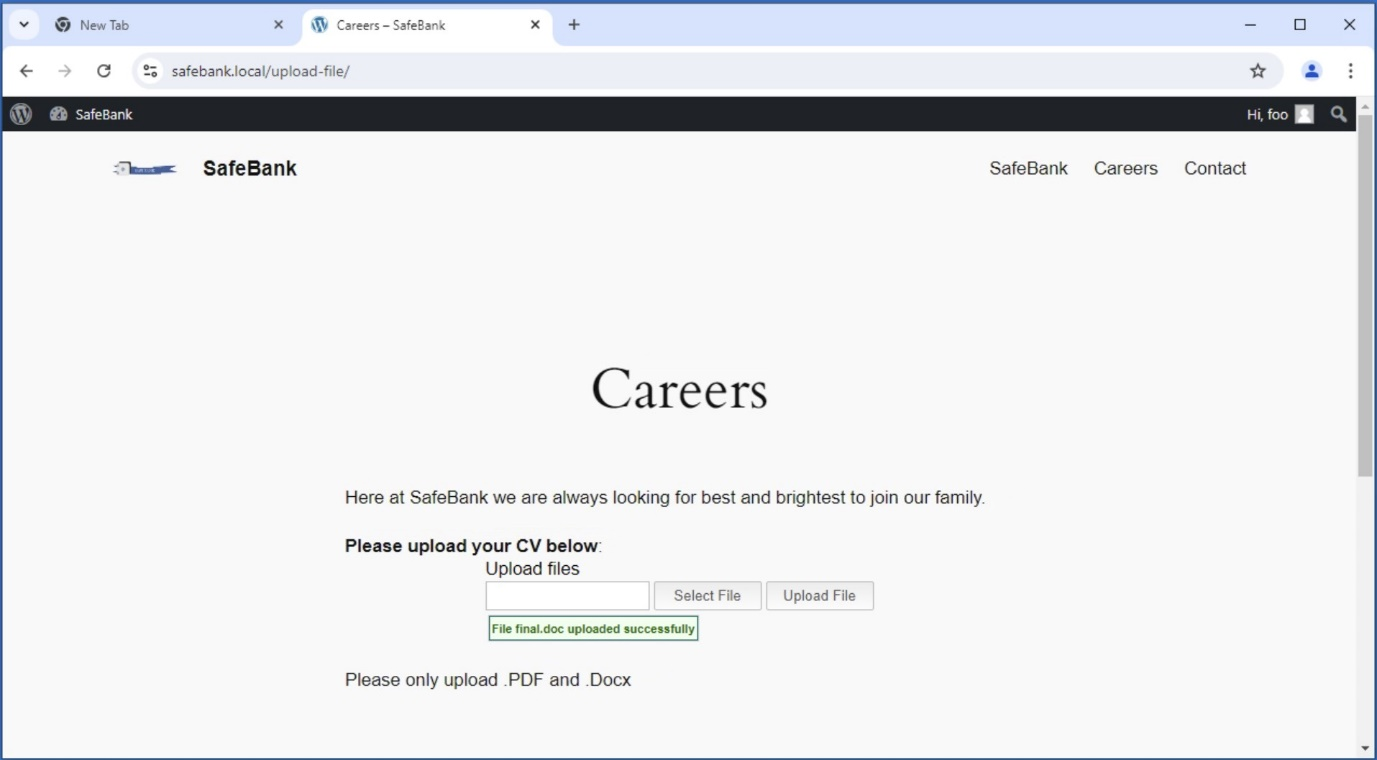
## 003-Unrestricted File Upload (Risk: Medium)

### Description

The Safe Bank web application allowed attackers to upload Office documents with Macro’s enabled and served as the path for delivery of an effective payload for initial access in the Safe Bank domain environment.

This upload mechanism did not perform any form of Antivirus scanning or sandbox execution to determine if uploaded files are safe for viewing.

In this instance, it likely provides an effective bypass of traditional security products that are used to protect users from files downloaded from the internet (Mark of the Web) as well as other email protection suites.



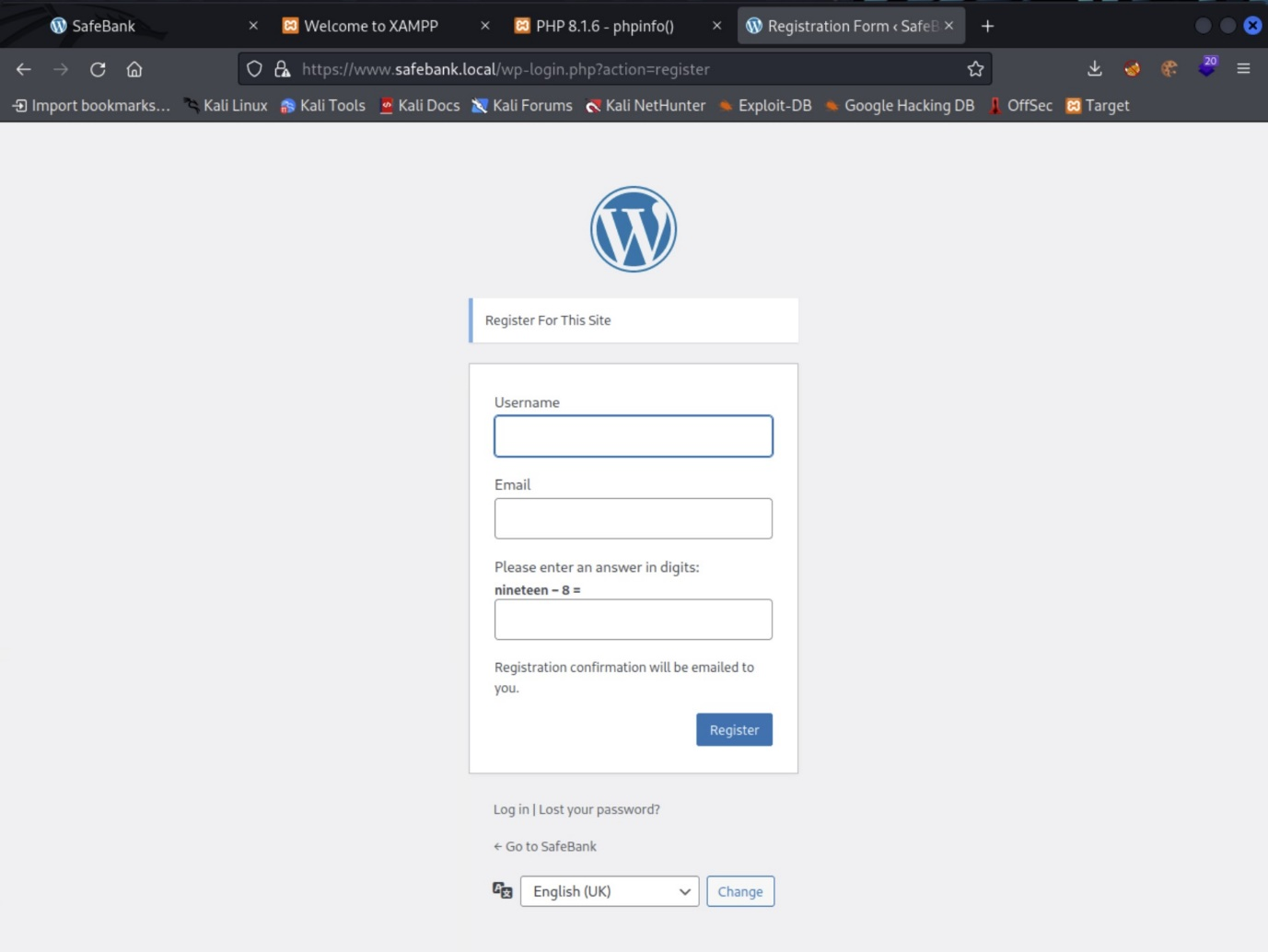
### Recommendation

As referenced in issue 002-Lack of Effective EDR / Monitoring, a more in-depth assessment should be made of existing security tooling to determine if the current solution can be improved to detect potential malicious payloads, ideally before the point of execution, or if a commercial solution should be considered.

## 004-Open WordPress Registration (Risk: Low)

### Description

The Safe Bank website is built on a Windows installation of the WordPress Content Management System (CMS). In its current configuration, remote anonymous users can create accounts on the website which in turn presents as a security risk as it exposes functionality in the WordPress application to unverified users and could facilitate further exploitation.



### Recommendation

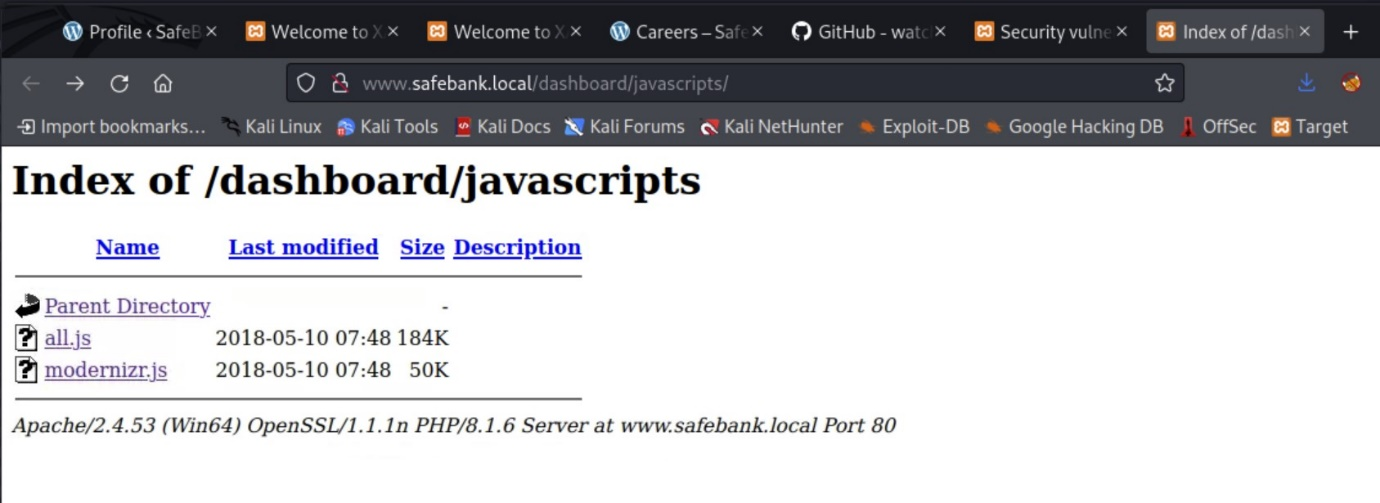
The Safe Bank website did not appear to expose any functionality which should require anonymous users to create accounts and authenticate, therefore it is recommended that this functionality is removed so as not to increase the attack surface of the application.

## 005-WebApp Directory Enumeration (Risk: Low)

### Description

The Safe Bank web application exposes unnecessary information due to directory listing being enabled in the underlying Apache web server.

This could potentially assist a would-be attacker to discover files or directories which may further an attack or attempts to gain more information about the application.



### Recommendation

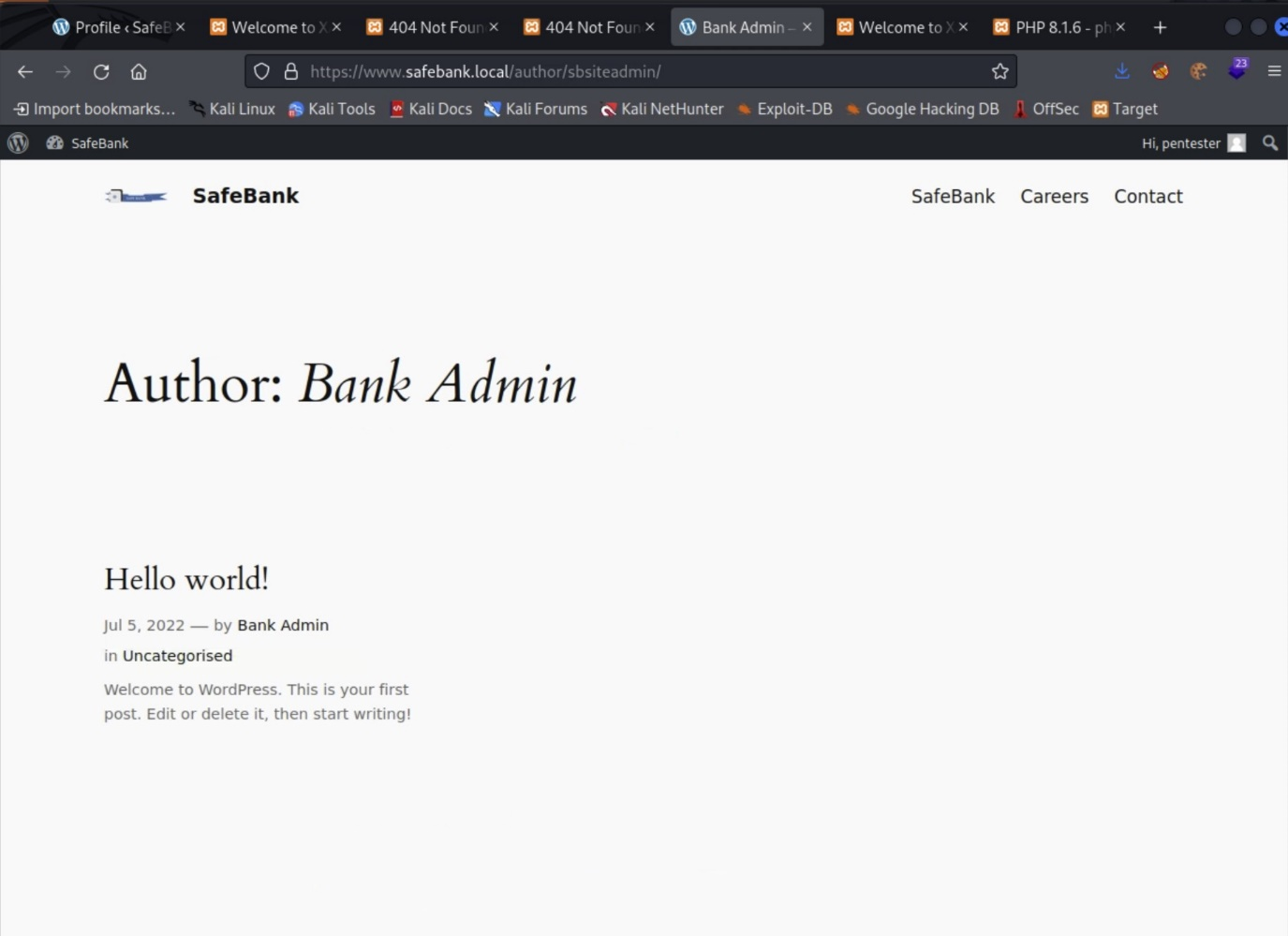
It is recommended that Directory Listing should be disabled in the underlying server.

## 006-Wordpress Site Allows Username Enumeration (Risk: Informational)

### Description

The WordPress CMS which hosts the Safe Bank site exposes user names in its current configuration.

This could facilitate an attacker looking to perform a password brute force against the application.



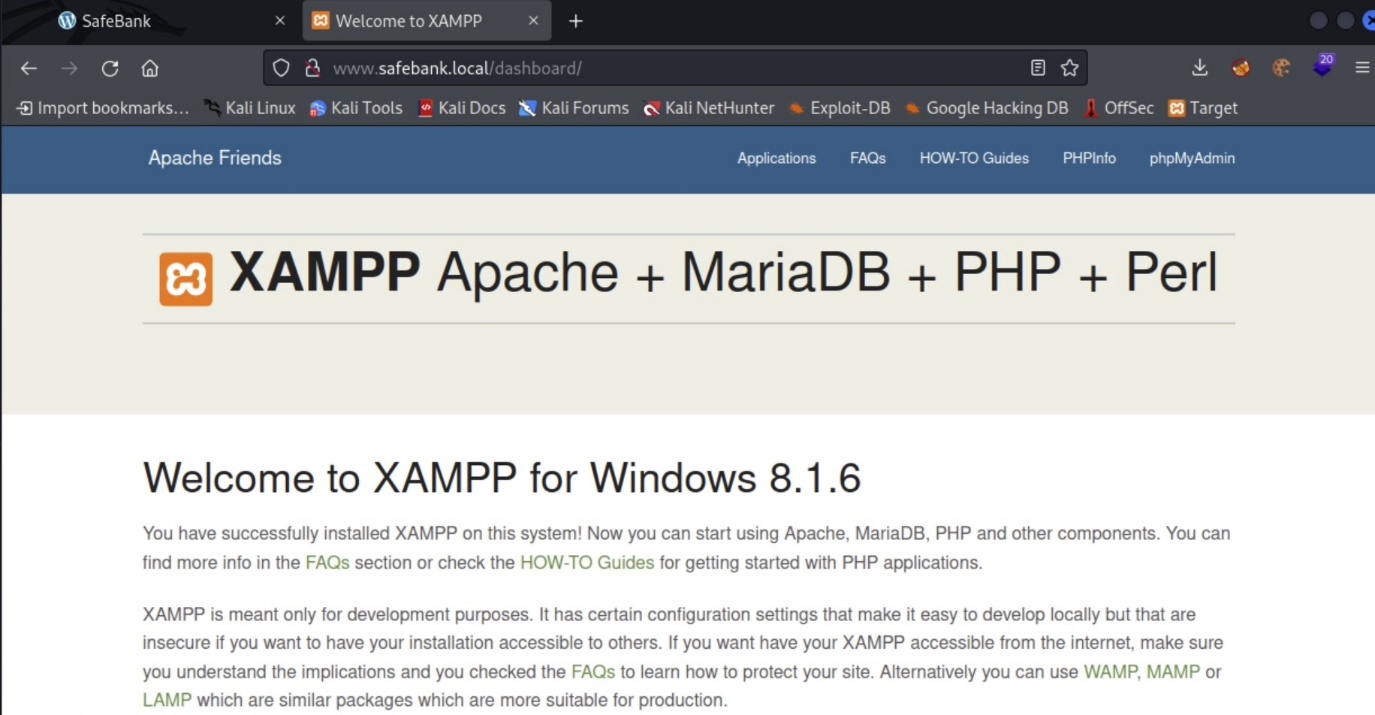
### Recommendation

Posts made on the WordPress CMS shouldn’t expose usernames where possible, or alternative accounts used that don’t have administrative permissions so as to prevent would be attackers from enumerating administrative accounts by which to target by other means.

## 007-WebApp Verbose Version Information (Risk: Informational)

### Description

The Safe Bank web application exposes version information left over from the default software installation which exposes too much information about the technology in use in the underlying application. This could aid an attacker during enumeration and vulnerability exploitation and is not inline with good security practice.



### Recommendation

Dashboards and pages which give away host, operating system and software information should be disabled or removed where possible.

1. <https://learn.microsoft.com/en-us/security/privileged-access-workstations/privileged-access-strategy> [↑](#footnote-ref-0)