Project Luna

Red teaming

Results report

MindBytes GmbH | Probststraße 15 | 70567 Stuttgart | Germany

+49 711 20709567 | hallo@mind-bytes.de

Represented by Christian Stehle, Nina Wagner, Simon Holl
HRB 790784 | Local Court: Stuttgart

Version 1.0
Confidential
Contact: hallo@mind-bytes.de

Sample Company GmbH

Contents

1 Management summary

Subject of the test: Entire company **Need for action:** Urgent

A realistic attack was carried out on Sample Company to gain insights into attack detection, defense, and potential vulnerabilities. This project consisted of two parts. The goal of the first part was to penetrate the internal company network from the outside, and the goal of the second part was to gain control over the internal IT environment (Active Directory) from an internal system.

One of the two goals, penetrating the internal company network from outside, could be achieved. We rate the security level of the IT environment and the attack detection measures as above average.

During the assessment, we gathered important insights, some of them with high risk. Some of the vulnerabilities that we identified allow taking over some OT systems in London, Paris, and Tokyo. Further findings are about the circumvention of implemented protective mechanisms, gaps in attack monitoring, and the disclosure of information. This shows that certain security mechanisms were effective but also that there is room for improvement.

Based on what we found, we recommend the following:

1. Review, assess, and address the identified findings (see 4 Findings).
2. Repeat the red teaming at a later date, but with a larger timeframe for the preparation of the red team. This will allow us to prepare domains for use in the project earlier, which technically reduces suspicions against them, and we will be better able to analyze and circumvent the EDR system.

Overall risk compared to other companies[1]**:** Better than average

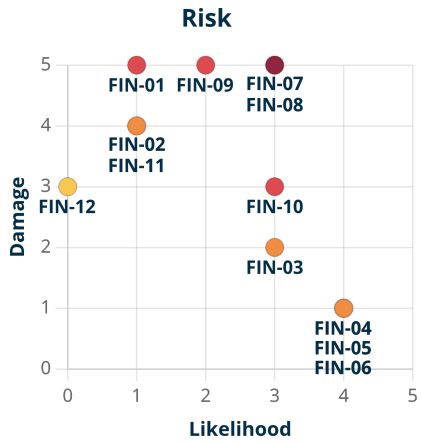


Figure 2 - Distribution according to damage and likelihood

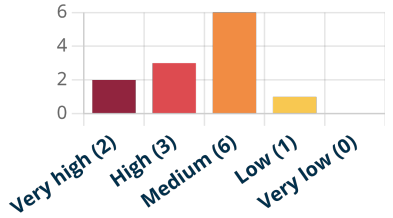


Figure 3 - Distribution according to risk

[1] This is a rating in comparison to other companies and does not allow any conclusions to be drawn about the existing risk in general.

1.1 Recommended actions

The estimation for the remediation is based on our experience and should be validated internally. In general, successful attacks result from a combination of several vulnerabilities, which is why we recommend that all findings are rectified. When implementing measures, it is important not to view vulnerabilities as individual cases, but to work on the cause in order to prevent similar vulnerabilities in the future.

Action	Remediation	Notes on remediation	Findings
Separate networks	Urgent Weeks No	▪ The OT environment should be network-separated from the office systems, such as the Citrix environment, to prevent attacks on these sensitive systems	4.7 FIN-07: Internal: OT systems reachable from Citrix
Remove group policy	Urgent Weeks No	▪ The group policy should be disabled and affected systems should be cleaned	4.8 FIN-08: Internal: Usage of password from group policy
Update components	Medium-term Weeks Probably not	▪ Outdated software in use should be regularly updated	4.2 FIN-02: External: Usage of outdated software
Extend monitoring	Medium-term Weeks Probably	▪ The monitoring should include more components. ▪ The monitoring should include more components ▪ Expanding the monitoring may involve structural adjustments and/or the introduction of new tools	4.1 FIN-01: External: Breached credentials 4.4 FIN-04: External: Blind spot: web application monitoring 4.12 FIN-12: Internal: C2 channel establishment
Reduce information disclosure	Medium-term Weeks Probably	▪ Information disclosures should be regularly monitored and reduced to the technically necessary minimum	4.3 FIN-03: External: Metadata in documents 4.5 FIN-05: External: Disclosure of internal hostnames 4.6 FIN-06: External: Mail addresses verifiable 4.10 FIN-10: Internal: Sensitive information on network shares

Action	Remediation	Notes on remediation	Findings
Harden client	⌚ Medium-term ⏳ Weeks 💰 Probably	▪ The elimination of the findings contributes to making the (unnoticed) execution of malicious programs more difficult	4.9 FIN-09: Internal: Weaknesses in AppLocker configuration 4.11 FIN-11: Internal: Usage of outdated CrowdStrike software

⌚ Priority: Urgent / Medium-term / Long-term | ⏳ Estimated remediation time per finding: Hours / Days / Weeks | 💰 Cost: No / Probably (not) / Yes | ⚡ Quick win

2 Technical summary

2.1 Table of findings

Finding	Risk	Damage	Likelihood
4.1 FIN-01: External: Breached credentials 💡 Regular checks for newly published access credentials	High	Very high	Low
4.2 FIN-02: External: Usage of outdated software 💡 Use the latest version of the JavaFy framework	Medium	Very high	Low
4.3 FIN-03: External: Metadata in documents 💡 Remove metadata from documents before publishing	Medium	Medium	High
4.4 FIN-04: External: Blind spot: web application monitoring 💡 Implement comprehensive monitoring	Medium	Low	Very high
4.5 FIN-05: External: Disclosure of internal hostnames 💡 Use internal CA, avoid error messages	Medium	Low	Very high
4.6 FIN-06: External: Mail addresses verifiable 💡 Harden mail servers	Medium	Low	Very high
4.7 FIN-07: Internal: OT systems reachable from Citrix 💡 Introduce and enforce network segmentation to isolate particularly critical networks	Very high	Very high	High
4.8 FIN-08: Internal: Usage of password from group policy 💡 Deactivate or delete the group policy	Very high	Very high	High
4.9 FIN-09: Internal: Weaknesses in AppLocker configuration 💡 Clean up rules, implement WDAC	High	Very high	Medium
4.10 FIN-10: Internal: Sensitive information on network shares 💡 Clean up shares	High	High	High

Finding	Risk	Damage	Likelihood
4.11 FIN-11: Internal: Usage of outdated CrowdStrike software 💡 Update/Upgrade to CrowdStrike Agent 6.0 or higher	Medium	Very high	Low
4.12 FIN-12: Internal: C2 channel establishment 💡 Implement detection for common C2 channels	Low	High	Very low

Details for each of the findings are described in section 4 Findings.

The following files are attached to this report:

🔎 RedTeamDatabase.xlsx: Among other information, it contains technical information (domains found, accessible services) and information about employees (names, email addresses) from the reconnaissance phase. It also includes the Red Team Activity Log, which records the actions we performed, including the time, source and target systems, users affected, and commands executed.

2.2 Next steps

n/a

2.3 Starting point in the project

Information provided	Test scope	Approach	Starting point
no (Black-Box)	**complete**	**hidden (Red Teaming)**	**from outside**
some (Grey-Box)	limited	obvious (Pentest)	from inside
comprehensive (White-Box)	focused		

2.4 Project limitations

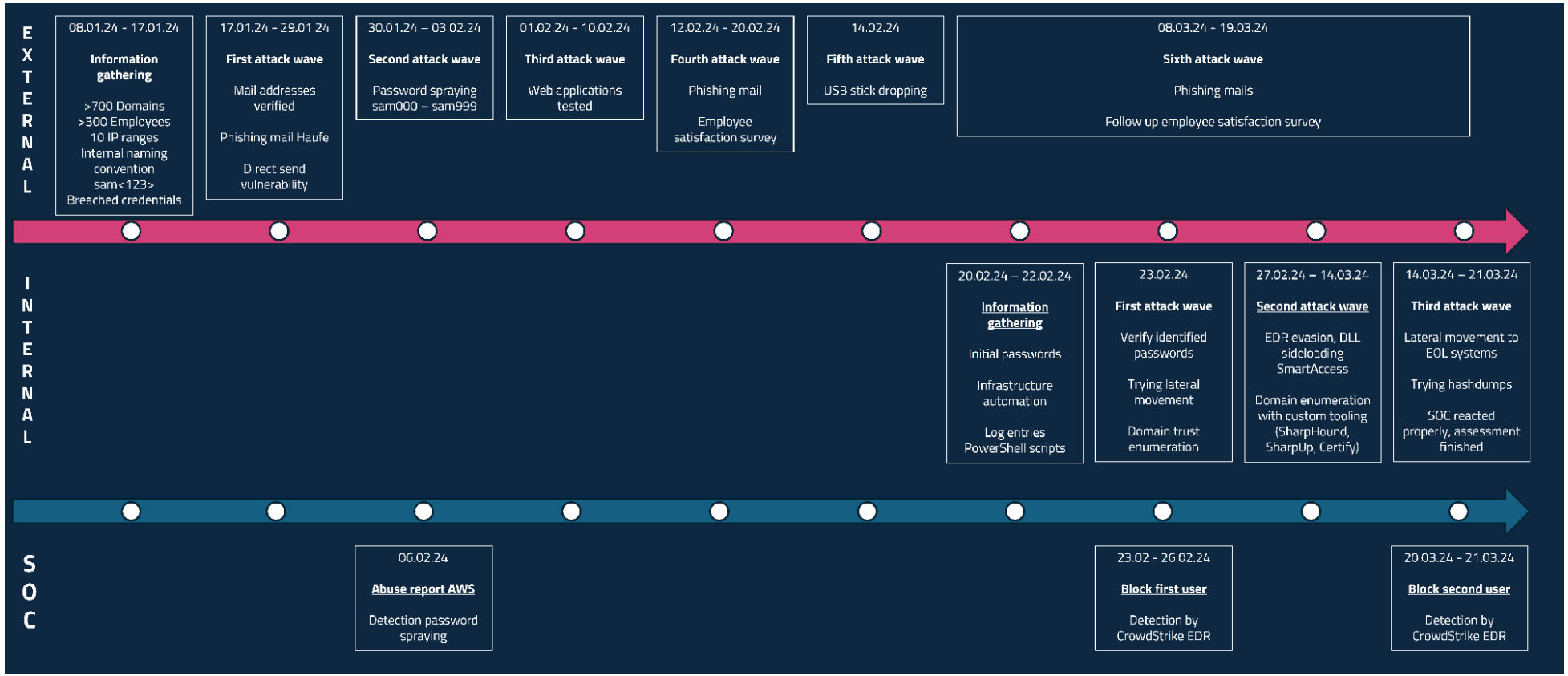
n/a

2.5 Postprocessing

n/a

3 Attack narrative

This chapter chronologically presents the actions taken, our findings and assumptions. The vulnerabilities identified during the assessment are described in the Findings chapter.



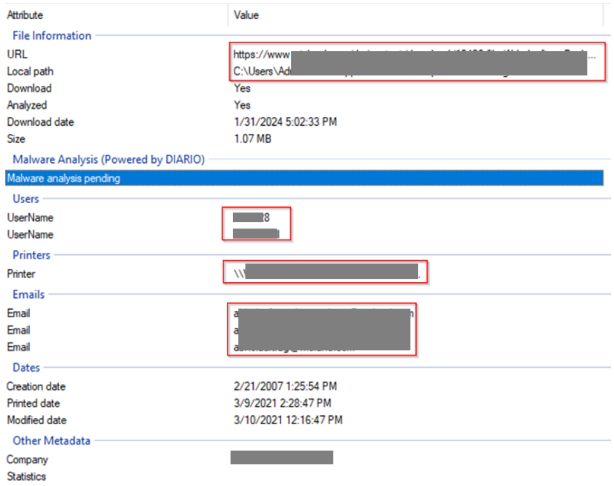
3.1 First part: External – initial access

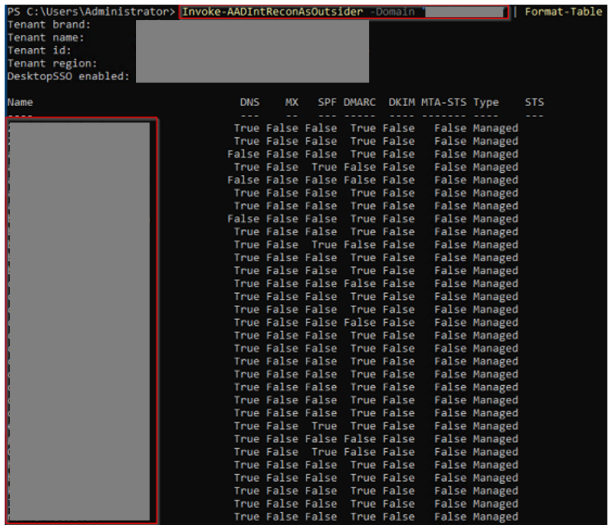
3.1.1 Information gathering: Determining the attack surface (08.01.2024-17.01.2024)

In the first step, we gathered as much information as possible to identify potential attack points. We began with Open Source Intelligence (OSINT), collecting all publicly available information about the client.

We compiled over 700 domains, 10 network ranges, and 336 employees with first and last names. A closer examination of the systems revealed that **M365 logins were frequently used**. We also suspect that the network blocks in **London belong to on-premise networks** and have a connection to the internal office network. Other networks at external hosts were not considered as they, from our perspective, would not lead to the target – the internal network.

Using the FOCA tool, we analyzed metadata from published documents. Various search engines were used to find documents in specific formats (.pdf, .docx, ...) and then automatically download and analyze them. In the metadata, we found valuable information: various employee names and an **ID in the format sam000**. We suspect this ID represents an internal naming scheme, such as a user ID in Active Directory.





An examination of the M365 tenant revealed many more linked domains, which we included in our further research. This information can be read without authentication based on the domain "samplecorp.com."

We also searched for so-called Breached Credentials. These are published login credentials data from websites that have been compromised. We used the service dehashed.com and found **200 published access credentials** for the email domains samplecorp.de and samplecorp.com. It was noticeable that several passwords followed the pattern "FirstnameSamplecorp." We suspect this could be an initial password. We tested the credentials on the publicly accessible M365 login service but had **no success**. As we later found out, the login does not use the email address with the first and last name, but the internal naming scheme (sam<3-digit number>). The Breached Credentials contained no such email addresses (samxxx@samplecorp.com). In the network blocks, we checked the active services and found almost exclusively web applications. It was noted that some of them were self-developed using an outdated framework (JavaFy 7.7.17).

After having an **overview**, the **following attack vectors** emerged:

1. Phishing
2. Attacks on web applications
3. Password spraying
4. USB stick dropping

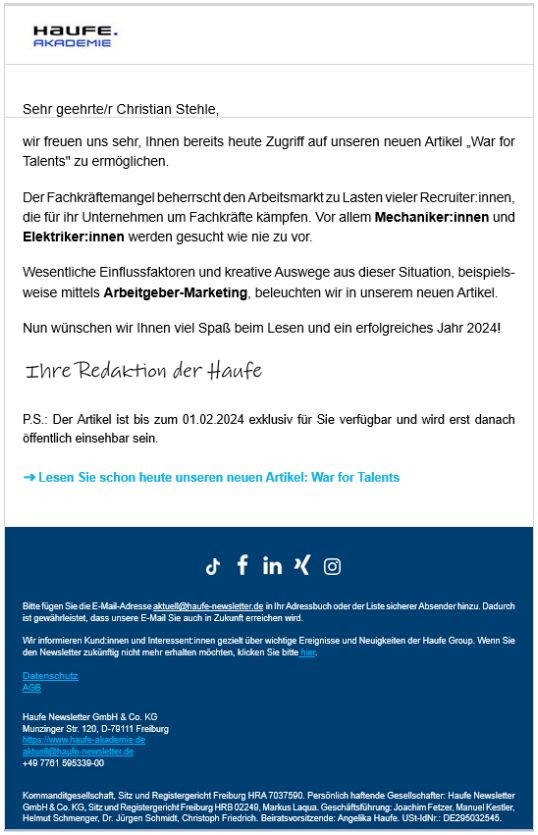
Relevant APT groups (Advanced Persistent Threat, designation for an organized attacker group) rely on phishing as an effective method for initial access to a company. Therefore, phishing was also our first attack vector. In our research, we increasingly encountered prefaced Microsoft logins, so the goal of our phishing campaigns was to obtain Microsoft credentials. We suspected that this could enable the use of various remote access solutions for the internal network, such as VPN and Citrix endpoints, or access to sensitive information.

3.1.2 First attack wave (17.01.2024-29.01.2024)

First, we tried to find valid email addresses. Previous research revealed that the **email format is Firstname.Lastname@sample.com**. Having identified over 300 individuals with first and last names during the information gathering phase, we could also infer the missing email addresses. Since some of these individuals might

no longer work at SampleCompany, we tried to **directly verify the email addresses at the mail server**. This succeeded, as in 4.6 FIN-06: External: Mail addresses verifiable.

For the phishing scenario, we chose a **newsletter article relevant to HR staff**. Since job advertisements frequently sought mechanics and electricians, we based a phishing email on this. We **imitated the style of Haufe**, a well-known provider in the HR area, which also has a newsletter. Addressed employees might already be familiar with legitimate Haufe emails and thus might question our phishing email less. The created phishing email is shown in the figure.


HAUFE. AKADEMIE

Sehr geehrte/r Christian Stehle,

wir freuen uns sehr, Ihnen bereits heute Zugriff auf unseren neuen Artikel „War for Talents" zu ermöglichen.

Der Fachkräftemangel beherrscht den Arbeitsmarkt zu Lasten vieler Recruiter:innen, die für ihr Unternehmen um Fachkräfte kämpfen. Vor allem **Mechaniker:innen** und **Elektriker:innen** werden gesucht wie nie zu vor.

Wesentliche Einflussfaktoren und kreative Auswege aus dieser Situation, beispielsweise mittels **Arbeitgeber-Marketing**, beleuchten wir in unserem neuen Artikel.

Nun wünschen wir Ihnen viel Spaß beim Lesen und ein erfolgreiches Jahr 2024!

Ihre Redaktion der Haufe

P.S.: Der Artikel ist bis zum 01.02.2024 exklusiv für Sie verfügbar und wird erst danach öffentlich einsehbar sein.

➔ Lesen Sie schon heute unseren neuen Artikel: War for Talents

Bitte fügen Sie die E-Mail-Adresse aktuell@haufe-newsletter.de in Ihr Adressbuch oder der Liste sicherer Absender hinzu. Dadurch ist gewährleistet, dass unsere E-Mail Sie auch in Zukunft erreichen wird.

Wir informieren Kund:innen und Interessent:innen gezielt über wichtige Ereignisse und Neuigkeiten der Haufe Group. Wenn Sie den Newsletter zukünftig nicht mehr erhalten möchten, klicken Sie bitte hier.

Datenschutz
AGB

Haufe Newsletter GmbH & Co. KG
Munzinger Str. 120, D-79111 Freiburg
https://www.haufe-akademie.de
aktuell@haufe-newsletter.de
+49 7761 595339-00

Kommanditgesellschaft, Sitz und Registergericht Freiburg HRA 7037590. Persönlich haftende Gesellschafter: Haufe Newsletter GmbH & Co. KG, Sitz und Registergericht Freiburg HRB 02249, Markus Laqua. Geschäftsführung: Joachim Fetzer, Manuel Kestler, Helmut Schmenger, Dr. Jürgen Schmidt, Christoph Friedrich. Beiratsvorsitzende: Angelika Haufe. USt-IdNr.: DE298032545.

The email enticed recipients with an exclusive, newly published article. This provided a reason why **viewing the article initially required a Microsoft authentication**. Clicking the included link led the recipient to a page we controlled, which mimicked the Microsoft login and performed a real login in the background. If a multi-factor login was configured, the required token was also queried, and a real Microsoft session was generated. If successful, we would have both the username and password as well as a valid Microsoft session, which we could likely use at various services used by Sample-Company. To circumvent internal restrictions on internet access, the created phishing page was delivered via the Azure CDN (azureedge.net).

However, when sending the emails, we found that the **mail server rejected us**. The exact error message is listed below.

Several circumstances, such as a **set A-record** for the phishing domain, a **change of the mail provider**, and the **age of the phishing domain** being more than 30 days, led in the further course of the Red Teaming to the mail server accepting the phishing emails. However, we received **no reactions from the recipients, neither visits to our link nor reports of phishing**. After there was no response to the phishing emails, we sent an application from a seemingly private address (x.y@z.de), but the mail server also rejected this email. We investigated the mail system further and found out that the mail security solution Cisco IronPort was used. The reputation of an email could be checked on the Talos Intelligence website by Cisco.

We also **tried to deliver emails in another way**. During information gathering, we identified the M365 tenant samplecompany and the Exchange Online Gateway at samplecompany-com.mail.protection.outlook.com. This can often be used to send (phishing) emails past the pre-set mail security gateway. However, **this attempt failed** because the Exchange Online Gateway only accepted emails from selected systems.

3.1.3 Second attack wave (30.01.2024-03.02.2024)

Next, we conducted a **password spraying** attack. This involves trying a password at several accounts to avoid lockouts due to multiple failed attempts at the same account. At that time, we had identified several blocks of digits in the internal naming scheme. We tried to guess valid accounts by incrementing the digits. The Microsoft login screen also confirmed that the login at M365 did not occur via the email address but by entering the internal account ID.

For the execution, we rented an AWS instance and used the tool Invoke-MSOLSpray.

We checked all theoretically possible usernames with the **password "SampleCompany2023!"** from the following identified user blocks:

- **sam000 - sam999**

To make the attacks less obvious, a pause of 30 seconds was maintained after each attempt. We could not detect any **locking of our IP address** during the approximately 3 days of password spraying with a total of 1000 usernames tried. Based on the feedback from the login attempts, we could identify approximately **500 valid user accounts**, but none of them had the password we tried. Six days after the start of the password spraying, we received an **abuse report from AWS**, as the password spraying had been noticed in the monitoring. This was **the first reaction of the SOC that we observed**.

3.1.4 Third attack wave (01.02.2024-10.02.2024)

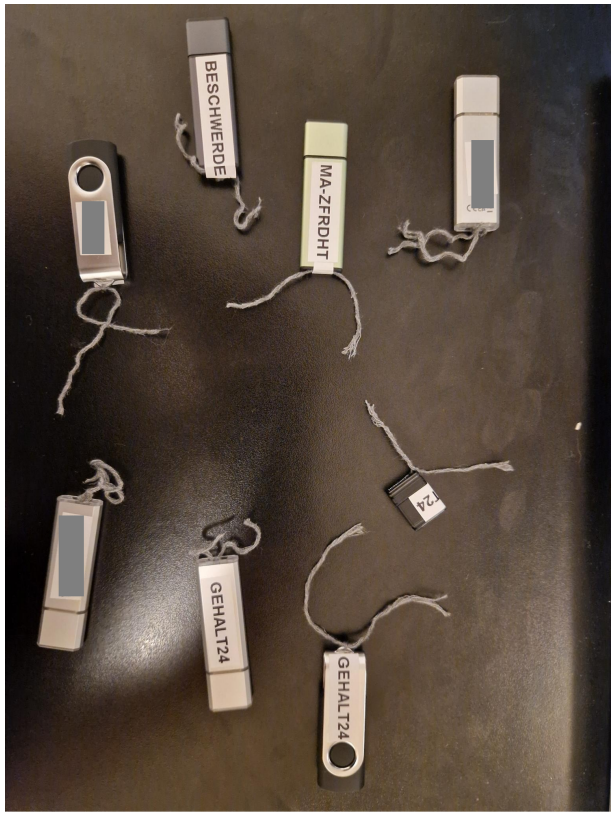
Next, we turned our attention to the **web applications in the relevant network blocks**. Initially, we examined these carefully and inconspicuously. After we **detected no restrictions in the form of a protection system**, we conducted increasingly aggressive and obvious tests. We suspect that there was no protection system that would have detected and repelled attacks of this kind. During the examination of the applications, we obtained various **information about internal hostnames and the internal Microsoft environment**, see 4.5 FIN-05: External: Disclosure of internal hostnames and 4.2 FIN-02: External: Usage of outdated software. However, we had no success in compromising a system via a web application.

3.1.5 Fourth attack wave (12.02.2024-20.02.2024)

Since all previous phishing attempts had been unsuccessful, we started a **new phishing campaign**. This time, however, we targeted a broader audience and chose a satisfaction survey as the scenario. We imitated a fictitious internal person, Barbara Rhubarb, who called for participation in the survey. Our goal was still to obtain M365 logins. Relevant email elements, such as the signature, were obtained by making an anonymous contact request through a web application and receiving a response from there. The mail was sent by the sender barbara.rhubarb@samplecompany.com.

3.1.6 Fifth attack wave (14.02.2024)

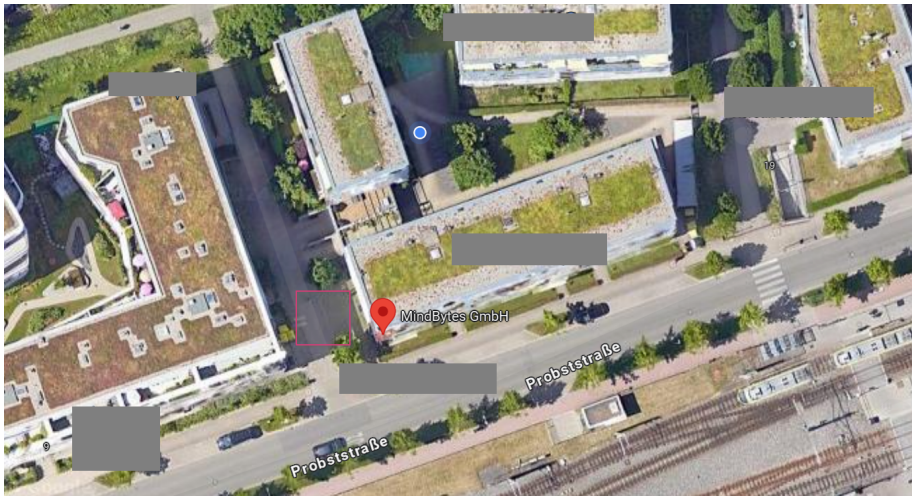
We decided to use **USB sticks** placed in the parking lots of the locations as the next attack vector. The goal was to pique the curiosity of employees, so they would plug the USB sticks into the internal systems and run our software. Therefore, we labeled the USB sticks with intriguing texts, such as "Salary" and "Complaint". We also adorned the USB sticks with torn threads, making it look as if they had fallen from someone.



Through information gathering, we had some information about the internal conditions, such as the Active Directory domain INTERN.SAMPLE.COM. We prepared the **payload to only execute if it was in this domain**. Otherwise, it would terminate immediately. This ensured that our malware would not run on private devices or on devices of employees from other companies. Additionally, this made detection more difficult when examining the malware in a separate environment.

The payload was a shortcut file (Salaries2024.lnk) that executed legitimate, signed .exe file, also provided on the USB stick. Our actual malicious payload, in the form of a DLL, was loaded when the .exe file was executed. This was because we suspected various protection mechanisms, such as AppLocker. AppLocker allows the execution of files that are signed by trusted parties by default. The payload prompted the user for an M365-Login including 2FA login, and once it obtained a valid session, the session token was sent to our server. This way we could reuse the authenticated session. The communication was intended to be encrypted via HTTPS through the Azure CDN. This was supposed to prevent internal communication restrictions to the internet. The choice of the Microsoft CDN was based on the information gathered in the information gathering phase that M365 was used. The .exe and .dll files were marked as "hidden," so they were not visible by default in Windows Explorer. Therefore, only a shortcut file appeared when the stick was plugged in, which had the icon of a text editor.

A total of 7 USB sticks were placed in the employee parking lots at the locations London, Paris, and Tokyo. The locations of the 4 USB sticks placed at the London site are shown in the figure.



This attack vector also **did succeed**, and we gained 2 sets of credentials. This marks the success of the first project phase, gaining internal access, as we could reuse the authenticated session to log in remotely into Citrix.

3.1.7 Sixth attack wave (08.03.2024-19.03.2024)

To create more backup footholds into the internal network, **isolated further phishing campaigns** were conducted. Existing campaigns were repeated or slightly modified. However, these did not have success.

3.2 Second part: Internal — escalate privileges

On February 20th, in consultation with our contact person, we decided to move on to the **second part of the project**. We used our successfully gained initial access vectors from 3.1.5 Fourth attack wave (12.02.2024-20.02.2024) , which we used to log in to Citrix remotely.

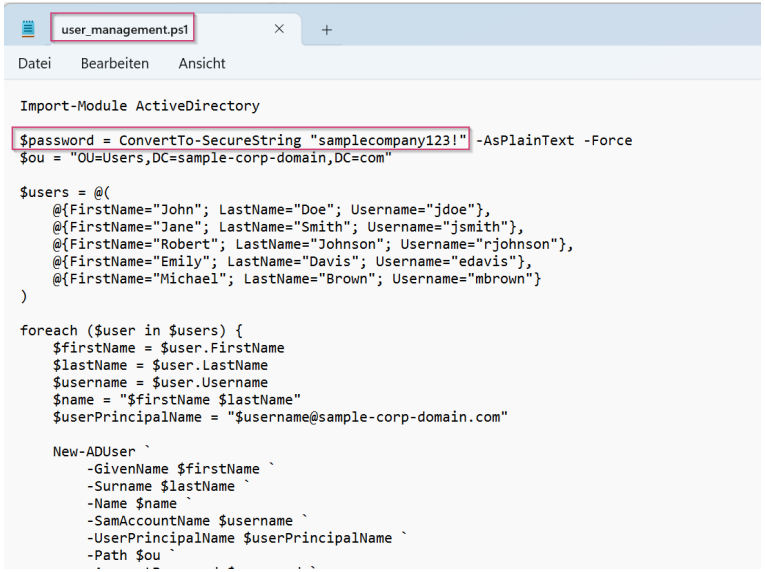
3.2.1 Information gathering: determining the attack vectors (20.02.2024-22.02.2024)

Initially, we examined the local system manually. We found that CrowdStrike is used as EDR, there were various applications, such as SAP and Microsoft Office, and AppLocker was present.

We tried to install a CrowdStrike EDR instance in our lab to analyze the behavior of the EDR and find possible circumventions. We contacted the manufacturer, distributors, and partner companies, but it was not possible for us to obtain a demo instance. This significantly complicated obfuscating our payloads and did not allow us to test possible detection of our payload by CrowdStrike in our own lab environment.

To remain **inconspicuous**, we manually searched network shares, especially the NETLOGON share. Here, we identified some **information that could be useful for attacks**, see 4.10 FIN-10: Internal: Sensitive information on network shares. This included plaintext passwords in scripts that bind SMB shares and encrypted passwords in automation scripts. We also identified interesting files on the exchange drive `X:\Exchange`, such as backups and private keys.

A large number of PowerShell scripts were located under `C:\ProgramFiles\SampleCompany\` and `\\sampleshareserver\Scripts\`. We suspected that the domain administration was mainly conducted with PowerShell. Additionally, these PowerShell scripts revealed some information about the internals of the domain, user management, and deployed software. At several points, log entries for the executed scripts were found, allowing us to track well when and in what context the scripts were executed.



A **password of the user _SampleManagement could be decrypted**, as described in 4.10 FIN-10: Internal: Sensitive information on network shares. Later, we found out that the password had been changed, and the corresponding script was therefore outdated.

To query details about the domain, we used the tool dsa.msc available on the system. This tool allows listing users and computers of the Active Directory. Thus, details about users and computers, such as group memberships, the date of the last password change (`pwdLastSet`), or the last login date of computers (`lastLogon`), could be queried.

3.2.2 First attack wave (23.02.2024)

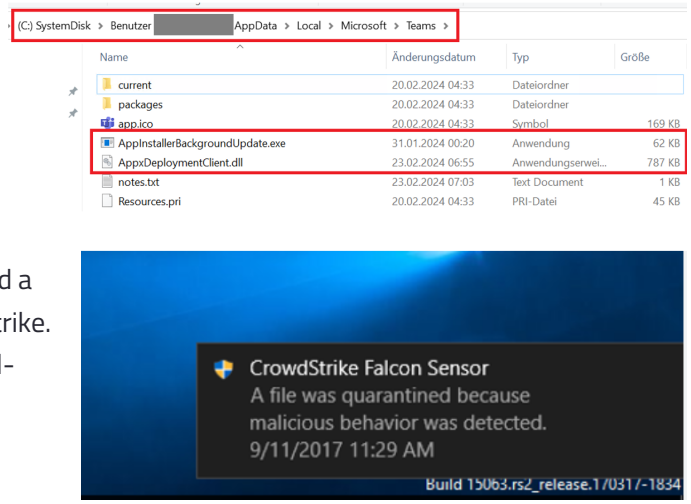
After we found that our Citrix system seemed to be well hardened, we tried to move to another system. The **goal was to access a system that was less hardened** and especially not in the group "sample_server_ENHANCEDSECURITY." We therefore tried to log in to another Citrix system. Typically, Citrix users are also assigned RDP rights on the terminal servers. However, this was not the case, and we could not move laterally to less monitored Citrix systems in this way.

Instead of lateral movements between systems, we now tried to jump to another user. To do this, we attempted a login with the previously obtained credentials of the user _SampleManagement. To do this inconspicuously and detached from our previous user, we used the Windows login screen presented to us when connecting to Citrix. The error message revealed that the password used was not correct. We could later determine that our information from which we extracted the password was outdated and the password had since been changed.

Then we examined the **other domains** that had caught our attention (**sample-subcompany1, sample-subcompany2**). Since the client had acquired and integrated many companies, we suspect that domains from a trust relationship (Trust Relation in Active Directory) are less monitored and controlled. Our research began here

again with the NETLOGON shares of these domains. The provided user had the permissions to view and search them. Here, too, we found some interesting information, but nothing that would extend our rights.

To proceed more efficiently using tools, the next goal was to bypass the deployed EDR. For this, several prerequisites were checked first. We could successfully copy files from our system to the Citrix system. This ensured the transfer of our payload. We could successfully start PowerShell. This allowed us to read the AppLocker rules. Furthermore, we could **identify gaps in the AppLocker rules** that allowed the execution of our own programs, see also 4.9 FIN-09: Internal: Weaknesses in AppLocker configuration. The call to an azureedge.net domain in the web browser was successful. Therefore, communication to the system we controlled on the internet could probably also be successfully established.



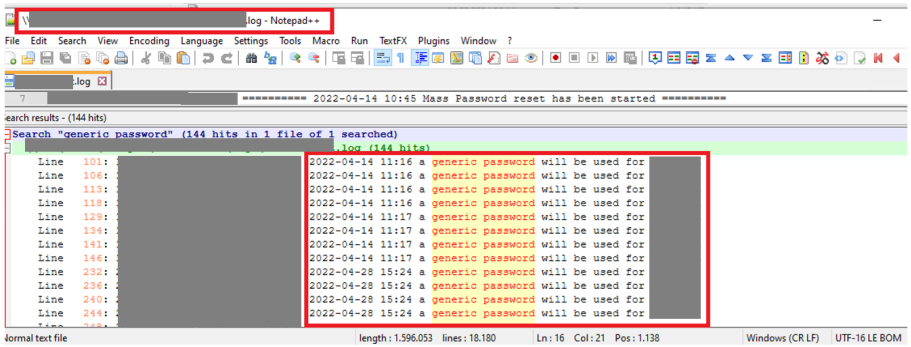
After these prerequisites were met, we copied our **prepared payload** in an encrypted file (sample-notes.zip) into the Citrix environment. We used the same principle of payload as with the USB sticks: Microsoft-signed .exe file with DLL sideloading. We unpacked the transferred files in the path excluded by the AppLocker rules C:\Users\johndoe\AppDate\Local\Microsoft\Teams. We found that Crowdstrike initially did not classify the file as harmful, and our payload remained available in the folder. **When executing the payload, however, the EDR intervened and reported malicious activity**. As we later learned from our contact person, CrowdStrike recognized the combination of a Microsoft-signed .exe file and a non-Microsoft-signed .dll file. However, the malicious code in the .dll file itself was not detected by CrowdStrike. Nevertheless, we could not bypass CrowdStrike in our first attack round and could not establish a command-and-control channel.

Finally, we tried to **establish alternative communication paths** to the internet. To do this, we attempted to establish ssh connections to a system we controlled. To rule out the blocking of typical ports, we used several ports. On ports 22, 25, 80, 443, and 12345, TCP connections were apparently successfully established, but **no SSH session was ever established**. We suspect that the firewall inspects the traffic, classifies it regardless of the port, and SSH connections to the internet are not allowed.

We suspected that the **CrowdStrike alarm triggered several actions in the SOC**. Among other things, our user account was locked. This did not happen immediately. We noticed the lockout on the next login attempt a few days later on 26.03.2024. The SOC contacted our internal contact person, who had requested the accounts, and demanded an explanation. To be able to **continue the Red Teaming**, the matter was internally clarified by our contact person so that we could continue using the

other remaining user. We were not involved in this communication. We continued the Red Teaming with the second provided user. Real attackers can also have access to multiple user accounts, so this approach is legitimate.

3.2.3 Second attack wave (27.02.2024-14.03.2024)

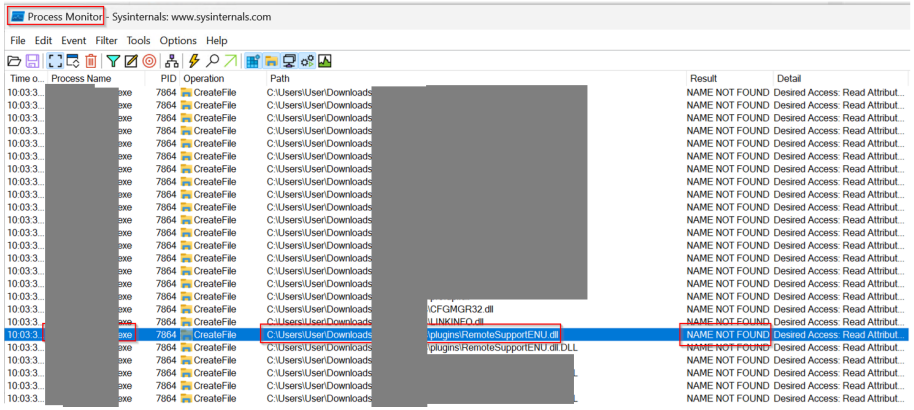


To make sure that any further actions could not be traced back to our user account, **we tried to obtain other Active Directory user accounts**. We used the findings from the information gathering phase regarding passwords of users. With an LDAP query, we searched for user accounts in the Active Directory that might still have the password assigned by the administrators. We compared the attribute pwdLastSet with the times revealed in the log entries. Although several user accounts still had the password set exactly at the time from the log entries, **no login was successful**. We conducted our login attempts partly at office.com, but also at the Windows login window when connecting to Citrix. We suspect that the generic password was adjusted when performing the mass reset in the graphical interface and thus no longer corresponded to the predefined password from the PowerShell script.

As a next step, we dared a **next attempt to bypass the EDR**. For this, we used the previously gained insights and tried to execute our payload in the name of an existing, legitimate application that was not signed by Microsoft or another manufacturer. Our payload was a beacon of the command-and-control framework Cobalt Strike. Since our beacon regularly established HTTPS connections to an Azure CDN, we looked for an application that itself regularly generated network traffic to Microsoft. We examined several applications, and the choice fell on SmartAccess, as it logs in with the M365 account and thus already generates some legitimate connections to Microsoft services.

Examination of SmartAccess in the Lab

We copied the SmartAccess application split into several encrypted ZIP archives from the Citrix environment to our lab. The split was done to avoid alarms caused by copying too large amounts of data. All part-archives were less than 100 MB in size. We then examined the application for possible vulnerabilities that could enable "DLL sideloading." Should the application attempt to reload a DLL that is not present, this is a potential approach to introduce a malicious DLL. For the analysis, the tool Process Monitor from Microsoft's Sysinternals collection was used.



Finally, we identified a vulnerable DLL named "RemoteAccessENU.dll" in the plugins folder, which suited our purposes. In our lab environment, we were able to successfully start a beacon by running SmartAccess.

Execution on the Citrix system

First, we tried to place the DLL in the installation path of the program. However, we did not have write permissions there. To bypass AppLocker, see 4.9 FIN-09: Internal: Weaknesses in AppLocker configuration, we copied the entire SmartAccess folder to `C:/Users/sampleuser/AppData/Roaming/Citrix/SelfService/`, placed our DLL in the plugins folder, and started SmartAccess.

The beacon successfully established the connection, and we successfully established our command-and-control channel.

This connection allowed us to download and execute further tools.

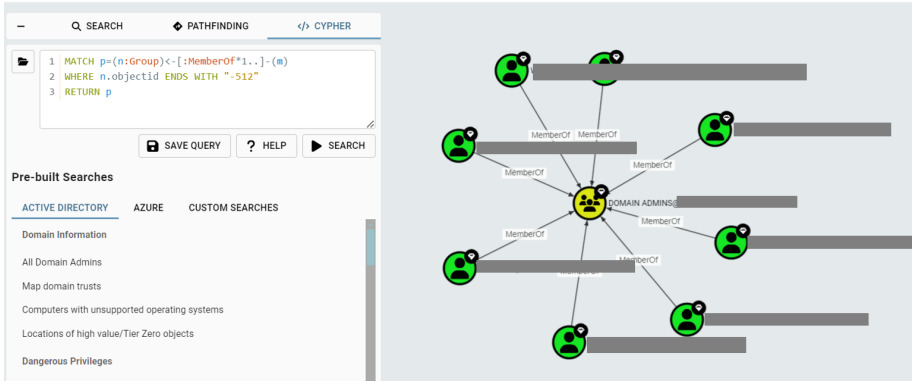
First, we determined which monitoring methods the EDR employed. With so-called hooks, EDR systems can monitor the actions performed and react accordingly. With the command hooks list of the beacon, we determined that the CrowdStrike EDR used this functionality. **With the command hooks clean, we could remove this monitoring undetected.**

```
[14.3.2024 05:23:16] (finished)[redacted]> hooks list
[!] Possible function hook found in module: ntdll.dll
    -> Function: LdrQueryImageFileExecutionOptionsEx
    -> Address:  0x77D1A1C0

[!] Possible function hook found in module: ntdll.dll
    -> Function: NtCreateFile
    -> Address:  0x77D24320

[!] Possible function hook found in module: ntdll.dll
    -> Function: NtOpenFile
    -> Address:  0x77D24100
```

As the first tool for scouting the environment, we used Bloodhound, a tool for enumerating the Active Directory. For this purpose, the collector called SharpHound was first started on the Citrix system. This **collected comprehensive information from the Active Directory** in the context of our user. The data could then be downloaded and processed and analyzed offline.



We **looked for attack paths in the Active Directory**. An initial analysis revealed several Kerberoastable accounts. No permissions were assigned to our user himself that would have helped us further. The success of a Kerberoasting attack depends on the complexity of the password of the vulnerable account. We wanted to conduct this attack in the third attack wave. After our user account was locked due to previous attacks, we could not perform the attack. Due to the many strong passwords observed at various points, we estimated the chances of success as rather low.

Since the **Active Directory Certificate Services (ADCS)** service often contains critical misconfigurations that allow users to escalate privileges, we examined this service next. For this purpose, we chose the tool certify, which enumerated the deployed CAs and certificate templates. Here, too, **no attack opportunities** emerged. Only 3 certificate templates were available, and these could only be used by domain administrators.

With the tool SharpUp, various possibilities **to extend user rights on the local system** were checked. Here **no direct possibility** was found. **However, an outdated Group Policy with a stored password was still active**, see also 4.8 FIN-08: Internal: Usage of password from group policy . We tried the **password on the Citrix system**, but this attempt **failed**. We also saw that the system was managed using LAPS, making it unlikely that the password would work on this system. An analysis

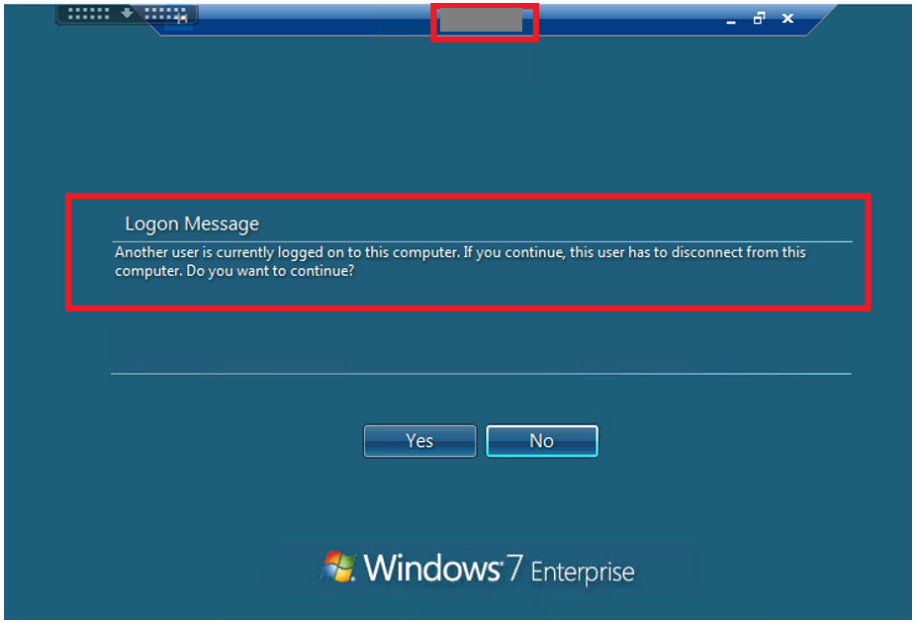
with the previously collected information from SharpHound revealed that the GPO apparently affected the entire Organization Unit "sample-hardware." This included over 10,000 systems. We **suspected that there might still be some systems on which the password was set by this Group Policy and noted this as an attack vector for later.**

3.2.4 Third attack wave (14.03.2024-21.03.2024)

We found a **large number of outdated systems in the Active Directory**, with operating systems that are no longer supported ("End of Life"). To exclude deactivated systems, we first filtered for systems that had logged in during the last 7 days.

Outdated systems are an attractive target because they lack the latest security features. We suspect that many of these systems belong to production. Therefore, we first checked the network reachability via port scans. We checked whether port 445/TCP was reachable. We found that **almost all systems were reachable**, see also 4.7 FIN-07: Internal: OT systems reachable from Citrix.

We tried to log in to the systems with the **password identified in the Group Policy**, as these systems were not managed by LAPS. To do this, we logged in via RDP with the user _Sample and had **success on some systems**. According to Active Directory descriptions, these were located in London, Paris, and Tokyo. When logging in, we found that **other users were active** there. Due to the locations and age of the systems, we suspected that these were OT systems. **As we wanted to avoid disruptions, we consulted with our contact person on how to proceed at this point.**



We were named the system SAMPLE012, which we were allowed to investigate further.

Since we had local administrator rights on this system, we tried to read detailed information about the deployed EDR CrowdStrike. This was done by downloading debug files that CrowdStrike had created. This is described in 4.11 FIN-11: Internal: Usage of outdated CrowdStrike software.

We found a configured exclusion for monitoring by CrowdStrike and prepared a payload accordingly. This payload attempted to read the cached login credentials. These credentials are stored in the memory of the "lsass.exe" process. To access

this process memory, a so-called handle is needed. Creating such a handle is very conspicuous. Therefore, we used the tool nanodump and prepared the execution so that an existing handle to the lsass process was cloned. This should reduce the chances of detection by CrowdStrike.

The **payload was tested in our lab** on a system of the same type to avoid unwanted side effects. Here, the stored login credentials could be **successfully** read.

The **execution on the system SAMPLE012 failed**, however, as CrowdStrike recognized the behavior and blocked it. Since we again suspected a lockout of the user account, we tried to generate a dump of the lsass process through the Task Manager. This is an obvious malicious method and is usually detected. This time the EDR SentinelOne also present on the system sounded the alarm and blocked the action.

We **then logged off and tried to erase the traces**, so we could later attempt another attack on a different system. As we found out the next day, the **SOC also correctly classified and acted on this incident**. **Our second user account was now also locked. This concluded the assessment.**

4 Findings

4.1 FIN-01: External: Breached credentials

Affected:

- Different user accounts with email address

Risk: **High**
Damage: **Very high**
Likelihood: **Low**

4.1.1 Summary

During the initial information gathering, some access credentials were identified that could be associated with service or user accounts of the client. These credentials originate from so-called breaches, i.e., websites whose user databases have been attacked. The identified credentials could not be used for a successful login. Likely, the passwords have been changed in the meantime, or the client regularly checks for such specific credentials. In the case that the client is already performing these checks, this situation should be considered null.

Possible consequences of successful exploitation

- With valid access credentials, an attacker can gain access to internal systems if a second factor does not protect the login

Examples of prerequisites for exploitation

- The credentials can be easily obtained
- The credentials must be valid
- If users reuse passwords, the affected credentials may also be valid for other applications
- None of the found credentials were valid

4.1.2 Recommendation

- Regularly check for newly published access credentials, e.g., using services like haveibeenpwned.com, if this is not already being done
- Employees should be made aware of using different passwords
- Employees should not use their business user accounts for registering with services for personal use

4.1.3 Technical Details

For the domains *samplecompany.de* and *samplecompany.com*, leaked credentials from various sources were analyzed. In total, 200 user accounts including passwords were identified. However, it was not possible to log in with the credentials.

As we later found out, a different naming scheme was used for remote access. Instead of the email address, an internal naming identifier was used. No credentials for such accounts were found in breaches. The credentials were queried with the DeHashed service which compiles login credentials from various sources, including the largest databases (“Breach Compilation”).

4.2 FIN-02: External: Usage of outdated software

Affected:

- https://oldapp.samplecompany.com
- https://evenolderapp.samplecompany.com

Risk: **Medium**
Damage: **Very high**
Likelihood: **Low**

4.2.1 Summary

Several web applications were developed using the JavaFy framework in a version which is outdated and in an end-of-life state. None of the publicly known vulnerabilities could be exploited.

Possible consequences of successful exploitation

- Known vulnerabilities allow attacks on other users
- Newly discovered vulnerabilities are not fixed by the manufacturer except with paid support

Examples of prerequisites for exploitation

- Exploitation depends on the vulnerability itself
- Some publicly known vulnerabilities are exploitable without authenticating, but they require interaction with a user

4.2.2 Recommendation

- Use the latest version of the JavaFy framework

4.2.3 Technical Details

During the examination of the web applications, it was found that several applications were developed with the JavaFy framework. It was noted that an outdated version was used. The information is beig disclosed in the source code. The deployed versions contain various vulnerabilities that could impair the availability of the service. Since our goal was to penetrate the internal network, we made no attempts to exploit the vulnerabilities. At least the following applications are affected:

- https://oldapp.samplecompany.com
- https://evenolderapp.samplecompany.com

4.3 FIN-03: External: Metadata in documents

Affected:

- Various office documents among the identified domains

Risk: **Medium**
Damage: **Medium**
Likelihood: **High**

4.3.1 Summary

In the metadata of publicly available documents, information is disclosed that can assist an attacker in further actions. Specifically, full names of employees, deployed software including version numbers, and a possible naming scheme for internal user accounts were identified.

Possible consequences of successful exploitation

- The disclosure of information itself is not a vulnerability
- Damage can occur if information is useful for attacks
- Example 1: Information about deployed software and versions helps to specifically search for publicly known vulnerabilities in the component
- Example 2: Technical information is helpful in preparing targeted social engineering attacks by designing legitimately sounding hooks

Examples of prerequisites for exploitation

- The documents are publicly available
- Viewing and collecting the information is possible with simple means

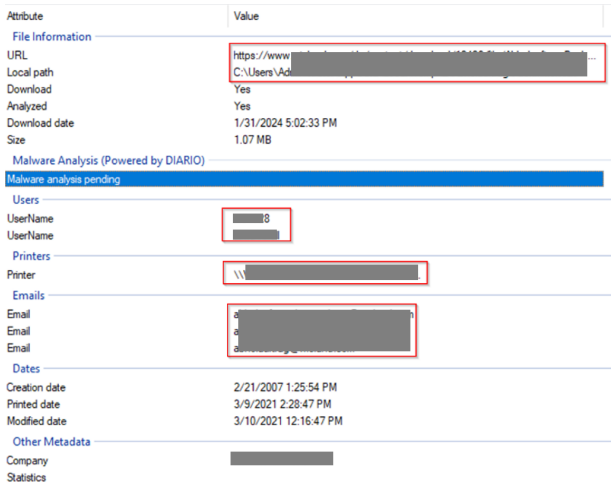
4.3.2 Recommendation

- Remove metadata from documents before publishing

4.3.3 Technical Details

Using the open-source tool FOCA, publicly available Office documents and PDF files were collected, downloaded, and automatically evaluated using search engines such as Bing and Google. The following image shows some identified information as an example.

Additionally, we suspected that the internal naming scheme was thereby revealed. To identify the author, entries with the structure “sam123” were used in documents, where only the 3 digits varied. Our assumption that these are the internal user accounts in Active Directory was confirmed during the course of the Red Teaming.



4.4 FIN-04: External: Blind spot: web application monitoring

Affected:

- All publicly accessible web applications

Risk: **Medium**
Damage: **Low**
Likelihood: **Very high**

4.4.1 Summary

Attacks were carried out on various externally accessible web applications. From what we could see from outside, those attacks seemed not to blocked or detected by any protection system.

We believe this represents a blind spot. We suspect that the situation was either not caught by blue team monitoring or went unnoticed.

Possible consequences of successful exploitation

- An attacker can discreetly identify and exploit even hard-to-detect vulnerabilities
- Depending on the nature of the vulnerability, an attacker may gain access to the internal network or collect valuable information on compromised systems

Examples of prerequisites for exploitation

- Vulnerabilities must be present
- The web servers were publicly accessible, no further steps were necessary

4.4.2 Recommendation

- Include publicly accessible web servers in monitoring
- Implement additional protective measures such as a Web Application Firewall (WAF) to detect and block attackers

4.4.3 Technical Details

Assessing the attack surface from the outside, we identified several web applications. First we inconspicuously examined them manually. Later on, we also used vulnerability scanners which send a large number of requests and check the applications for many different vulnerabilities.

During the examination of the applications, we noted that no counteraction occurred and that we were even able to examine the applications automatically without any restrictions. For example, a directory brute force was carried out to guess possible URLs, and we inserted payloads such as for discovering SQL injection into various input fields. This generated several requests per second over several minutes. The servers under investigation responded reliably and the requests all seemed to be carried out. Furthermore, even after extensive enumeration, our IP addresses were not blocked.

This indicates that the applications don't have any upstream protection system and are thus not included in monitoring.

Possible interesting points of attack could be the applications that use outdated software, see also 4.2 FIN-02: External: Usage of outdated software.

4.5 FIN-05: External: Disclosure of internal hostnames

Affected:

- Various internal and external systems, see description

Risk: **Medium**
Damage: **Low**
Likelihood: **Very high**

4.5.1 Summary

At various points that are accessible over the internet, information about internal systems is being disclosed. The information can be useful for more targeted attacks.

Possible consequences of successful exploitation

- The information can be useful for further attacks such as phishing

Examples of prerequisites for exploitation

- The information is freely available on the internet

4.5.2 Recommendation

- If possible, don't request public TLS certificates for internal systems
  - Use an internal CA instead
- Avoid detailed error messages.
  - Instead, a generic error code can be generated, which can be associated with error details on the server side

4.5.3 Technical Details

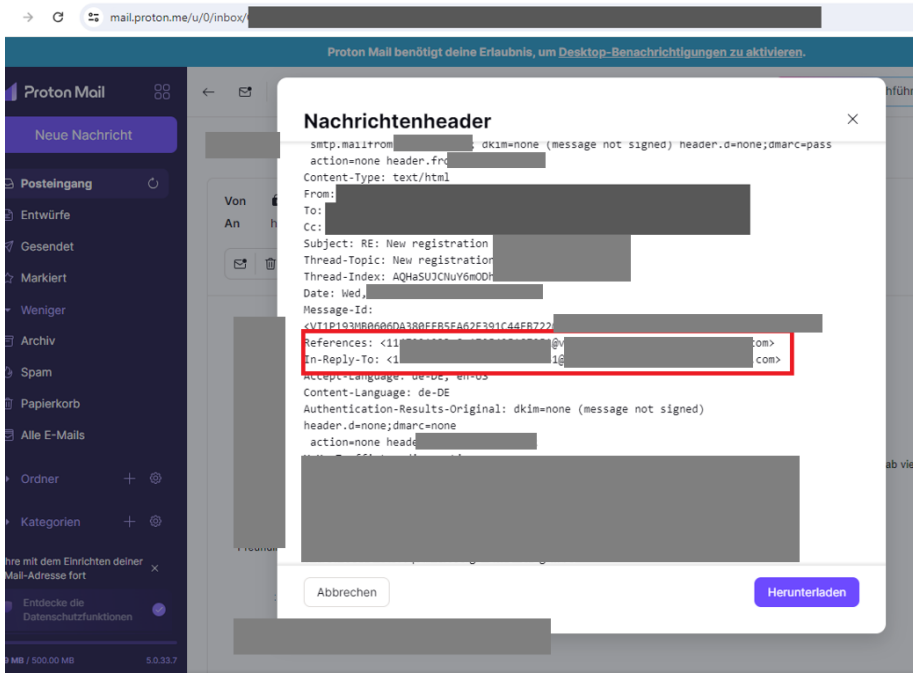
The following sections explain where technical information is being disclosed.

All of the information listed suggests that internally, the domain *samplecompany.com* is used.

Information disclosure through TLS certificates

When searching for domains, we found the subdomain *intern.samplecompany.com*. A search for possible requested TLS certificates with the service *crt.sh* revealed many more internal domains. The newest entries were a few days old. Therefore, we assume that the names belong to existing internal systems.

Information disclosure through mail headers



After registering a user with the web application *sampleapp*, an email was sent asking for the reason for registration. The headers of this email reveal internal hostnames at several points.

Information disclosure through website

When calling a non-existent URL, the internal hostname is being disclosed in the error message.

4.6 FIN-06: External: Mail addresses verifiable

Affected:

- Mail servers: mx1.samplecompany.com, mx2.samplecompany.com

Risk: **Medium**
Damage: **Low**
Likelihood: **Very high**

4.6.1 Summary

The affected mail servers allowed for the verification of the validity of email addresses without the need to send emails. Previously obtained email addresses could thus be verified and used in subsequent attacks.

Possible consequences of successful exploitation

- Previously obtained email addresses can be checked for validity to avoid hard bounces, so that the reputation of the sender's domain will not be jeopardized
- Combinations of popular first and last names can be tried to guess valid employees
- Follow-up attacks such as phishing that use valid email addresses only are less conspicuous

Examples of prerequisites for exploitation

- An SMTP connection has to be established with the mail server
- Verifying email addresses could be easily automated
- After 20 attempts, we had to wait a few minutes before verifying email addresses again.

4.6.2 Recommendation

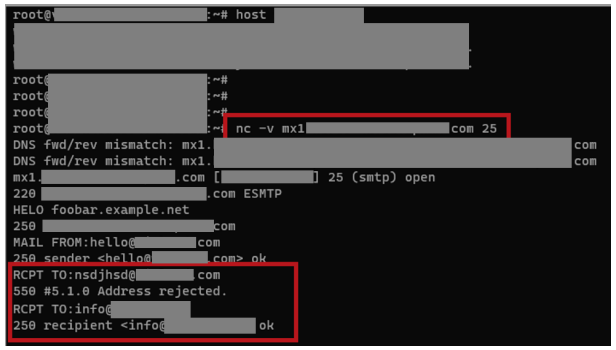
- The response from the mail servers should not reveal the validity of email addresses before an email is sent

4.6.3 Technical Details

The two mail servers *mx1.samplecompany.com* and *mx2.samplecompany.com* were examined for common methods to verify email addresses.

It was noted that with the SMTP command `RCPT TO`, both mail servers disclose whether the specified recipient exists or not. As seen in the figure, the mail server responded with the status code `250 sender _mail_@samplecompany.com ok` for a valid email address. For an invalid email address, the status code `550 #5.1.0 Address rejected.` was returned.

During verification, we noted that the mail server always responds with the status code `452 Too many recipients received this hour` after about 20 attempts. This temporarily prevented further verification of email addresses. However, this block was lifted after a few minutes.



4.7 FIN-07: Internal: OT systems reachable from Citrix

Affected:

- Internal networks of Sample Company

Risk: **Very high**
Damage: **Very high**
Likelihood: **High**

4.7.1 Summary

From the Citrix work system we could access systems that looked like OT systems. We assume that the network is inadequately segmented or no separation between different segments is enforced.

Possible consequences of successful exploitation

- Attackers compromise OT systems and disrupt parts of the production or expand their rights in the domain
- Attackers can more easily identify and exploit vulnerabilities, as OT systems are typically more vulnerable

Examples of prerequisites for exploitation

- An attacker needs access to a Citrix system
  - To get access, they must compromise a valid user account, for example through phishing

4.7.2 Recommendation

- Introduce and enforce network segmentation to separate particularly critical networks as much as possible
- Prevent regular workstations from accessing OT systems

4.7.3 Technical Details

During our vulnerability search, we first scouted the systems with an unsupported Windows operating system. These systems are often more vulnerable because they lack various security features of newer versions. Using a password of a local administrator user that we obtained earlier, as described in 4.8 FIN-08: Internal:

Usage of password from group policy, we first determined the network accessibility of systems via SMB. Here, we found hardly any restrictions. Almost all the systems we checked were accessible from the Citrix work system:

- Sample123
- Sample134
- Sample133
- Sample132
- Sample136
- Sample131
- Sample130

Furthermore, we tried to log in using RDP and also found hardly any restriction there. We suspect that some target systems were OT systems, as there were, for example, shared users named sampleshareduser (producing_terminal) logged in.

4.8 FIN-08: Internal: Usage of password from group policy

Affected:

- Group Policy "SAMPLE_GPO_LOCAL_ADMIN" of the domain samplecompany.local

Risk: **Very high**
Damage: **Very high**
Likelihood: **High**

4.8.1 Summary

Through a group policy, the password for two local administrator users was set on multiple systems in the Active Directory. The password could be read. The users were active on several systems, particularly OT systems.

Possible consequences of successful exploitation

- Local administrator rights on various systems, especially critical OT systems
- Taking over other systems with elevated rights (“lateral movement”)
- Reading cached session information on the systems that the admin users can access

Examples of prerequisites for exploitation

- Any domain user can read the password from the group policy with the appropriate tools

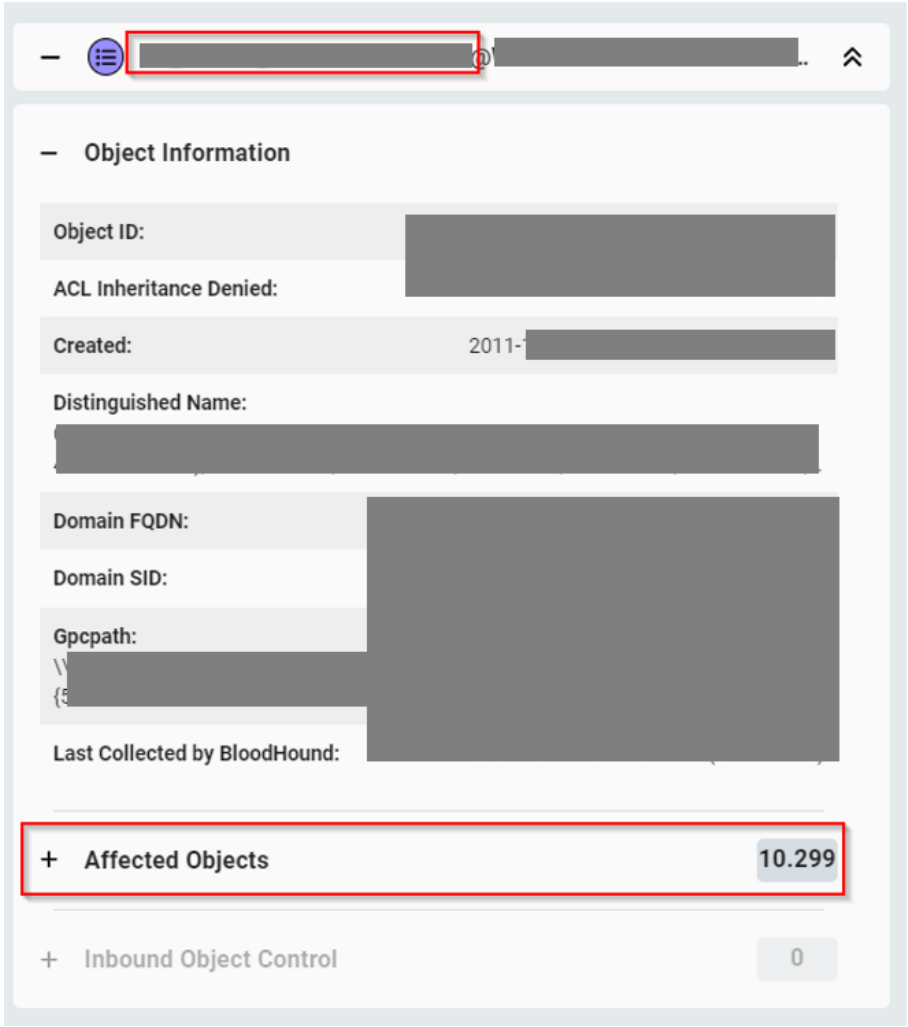
4.8.2 Recommendation

- Deactivate or delete the group policy.
- Check where the users in question still use the password, and change it

4.8.3 Technical Details

Through a group policy, two local administrator users (*Administrator* and *Client*) were deployed across multiple systems. The password was in encrypted form in the group policy itself and could be viewed by any user. Microsoft published the key for this specific encryption, making it easy to obtain the plain text password.

The group policy was created in 2011 and affects the entire Organization Unit "Hardware" of the domain *samplecompany.local*. This includes about 10,200 systems.



4.9 FIN-09: Internal: Weaknesses in AppLocker configuration

Affected:

- Citrix clients, e.g., SAMPLE012345

Risk: **High**
Damage: **Very high**
Likelihood: **Medium**

4.9.1 Summary

AppLocker is configured on the Citrix clients to prevent the execution of unwanted programs. However, we found several ways to bypass this and were able to execute our own programs.

Possible consequences of successful exploitation

- Attackers can execute malware, for example to establish a command-and-control channel

Examples of prerequisites for exploitation

- AppLocker rules must be known
  - Can usually be read by all users via a PowerShell command
- Once a loophole is recognized, an attacker needs the ability to place and execute files at the corresponding locations
  - This usually requires interactive remote access, for example using Citrix

4.9.2 Recommendation

- Use Windows Defender Application Control (WDAC) instead of AppLocker
- If you want to continue using AppLocker, improve the configuration with regard to the following points:
  - Check all placeholders in allowed paths: Users should not have permission to create or modify files affected by a wildcard rule
  - For administrative reasons, don't assign rules to individual users and avoid duplicates of rules
  - Restrict the execution of so-called LOLBAS files as much as possible

- Periodically evaluate necessary exceptions

4.9.3 Technical Details

The AppLocker configuration was extracted and analyzed on the Citrix system SAMPLE012345 with the PowerShell command `Get-AppLockerPolicy -Effective -Xml`.

We noted that some exclusions could be abused because they contained wildcards and our user (with normal privileges) was able to create files in the affected paths. The AppLocker policy included more than 700 rules (276 of them with wildcards) and therefore could not be fully analyzed. Below is an example.

In the following rule, all users (`C:\Users\*`) are allowed to execute any files and subfolders if they are located in the user folder under `AppData\Roaming\Citrix\SelfService\`. By default, users have write permissions in their own subdirectories, including the `AppData` folder.

During the assessment, we used this to execute our own tools to establish a command-and-control channel. Other affected rules:

- `C:\Users\*\APPDATA\LOCAL\TEMP\TEAMVIEWER\*.EXE`
- `C:\USERS\*\APPDATA\LOCAL\TEMP\TEAMS\*.EXE`
- `C:\SAMPLE\*`

4.10 FIN-10: Internal: Sensitive information on network shares

Affected:

- Network shares of the domain samplecompany.com, see technical details

Risk: **High**
Damage: **High**
Likelihood: **High**

4.10.1 Summary

Users in the Active Directory (AD) are able to access sensitive data on network shares. These contain personal data as well as credentials for highly privileged users and can be useful for further attacks.

Possible consequences of successful exploitation

- Multiple passwords can be identified, both from normally privileged and highly privileged user accounts
- The log files provide lots of information on how the scripts operate, enabling more targeted attacks

Examples of prerequisites for exploitation

- An AD user is needed
- Useful information must be found among the large number of files
- There are tools that automate this process

4.10.2 Recommendation

- Change disclosed credentials
- Evaluate whether stored information is still needed
- Internally review the network shares based on the attached list

4.10.3 Technical Details

Network shares often contain information that are useful for attacks, such as passwords in scripts or files. Therefore, we investigated the accessible network shares from this point of view. Below are the shares that contain sensitive information.

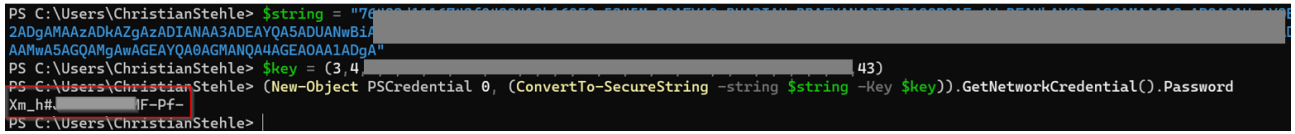
\\SHARE1\

The network drive `\\\\SHARE1\\` is mapped for the provided users. This share is apparently being used internally to exchange files. Other domain users were adding new files on a daily basis. Some of these were backup files that might contain sensitive information. An attacker with more time could monitor this drive and wait for sensitive information to be placed there.

\\samplecompany.intern\netlogon

We found credentials in plain text in multiple scripts. Specifically, these were scripts that automatically map network drives. The affected files were:

- FILE1.cmd
- FILE2.cmd
- FILE3.cmd
- FILE4.cmd
- FILE5.cmd



In the script `FILE1.cmd`, there was a reference to another script (`sample_office.ps` 1), to which a key was passed as a parameter. The script `sample_office.ps1` loaded a SecureString from a text file and decrypted it with the key. We were able to use this to obtain the password of the user *_sample*.

We were able to read the SecureString as it was present on the file system of the Citrix system. This allowed us to decrypt the plaintext password of the user. Later, we found that the password no longer worked.

4.11 FIN-11: Internal: Usage of outdated CrowdStrike software

Affected:

- Deployed CrowdStrike EDR solution

Risk: **Medium**
Damage: **Very high**
Likelihood: **Low**

4.11.1 Summary

The version of the CrowdStrike EDR that is being used allows reading exclusions as well as the password hash of the supervisor user. We were unable to obtain the password during the project period.

Possible consequences of successful exploitation

- Bypassing the EDR by unnoticed execution of malware
- Uninstalling the EDR on a system

Examples of prerequisites for exploitation

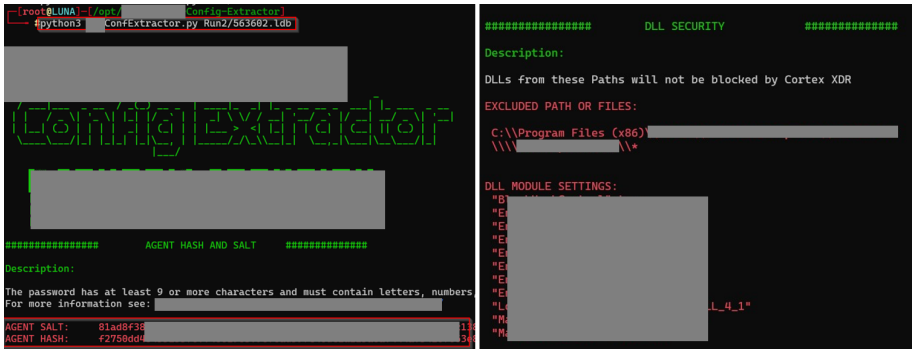
- Local administrative privileges are required to read the exclusions and the password hash

4.11.2 Recommendation

- Update/Upgrade to CrowdStrike Agent 6.0 or higher
- Review and remove exclusions if no longer needed

4.11.3 Technical Details

Local administrators are able to read debug files of the deployed CrowdStrike EDR solution. These files contain some information about the inner workings of the EDR as well as the password hash of the supervisor user. We analyzed these debug files using the tool *CrowdStrike-XDR-Config-Extractor*. The exclusions should be checked to see if they are still relevant.



4.12 FIN-12: Internal: C2 channel establishment

Affected:

- Monitoring internet network traffic

Risk: **Low**
Damage: **High**
Likelihood: **Very low**

4.12.1 Summary

A command-and-control channel (C2 channel) that we established was not detected. Through the channel, we were able to execute commands remotely, and data, such as command outputs and file contents, were transmitted to the server of the red team.

Possible consequences of successful exploitation

- An attacker can execute commands and steal data

Examples of prerequisites for exploitation

- A program must be executed, which requires bypassing protection mechanisms such as AppLocker and antivirus software

4.12.2 Recommendation

- Implement detection for common C2 channels
- Review the effectiveness of the implemented solution

4.12.3 Technical Details

On the system, we initiated a beacon that established an HTTPS connection to a server controlled by us and communicated with it. The beacon received commands that we issued, executed them locally, and returned the results. This is referred to as a command-and-control channel.

We used the Azure CDN domain, which is classified as trustworthy in many proxy solutions and thus was accessible in this case. The beacon sent an HTTPS POST request every 30 seconds with 30% jitter to the address `https://samplecompany.azureedge.net/msupdate/Setup_Install001.cab` and used a base64-encoded payload for data transmission.

5 Project scope

5.1 Persons involved

Name	Role	Mail address
Christian Stehle	Project lead & Pentester	hallo@mind-bytes.de
Nina Wagner	Pentester	hallo@mind-bytes.de
Simon Holl	Pentester	hallo@mind-bytes.de
Anja Neudert	Review	hallo@mind-bytes.de
Max Mustermann	CEO	max.mustermann@samplecompany.com

5.2 Test period

1/1/24–3/31/24

5.3 Provided accounts

No user accounts were provided.

5.4 Provided information

No information was provided.

5.5 Implementation concept

A red team has the task of achieving project goals, such as infiltrating a company's infrastructure, without being detected. Techniques similar to those used by real attackers are used to test both the technical and organizational defense capabilities of a company.

In particular, the effectiveness of the Blue Team's or Security Operations Center's (SOC) defense mechanisms and internal reporting chains are tested under real-world conditions.

During project implementation, the White Team — a group of people within the attacked company who are privy to the exercise — and the Red Team work closely together. The Red Team communicates the project's progress. Decisions on how to proceed under certain circumstances are made jointly. The White Team shares observations made by the Blue Team with the Red Team.

5.6 Rules of Engagement

Before the project begins, the client and the Red Team define the objectives and framework conditions for the implementation of the project in the so-called *Rules of Engagement* (RoE). The following sections provide an overview of the potential approaches in a red teaming project and the rules established for this specific implementation.

5.6.1 Starting point of the red team

What is the starting position from which the red team begins its operation? Like real attackers, the red team uses this starting position to find attack paths to achieve the project goals. As part of an assume-breach approach, it could be specified, for example, that the red team starts with access to a laptop, as employees receive by default. This would be the position an attacker has after a successful phishing attack.

In this project: Both externally and internally. In the first part of the project, attempts were made to penetrate from the outside. In the second part of the project, attempts were made to compromise the internal environment from the inside.

5.6.2 In-Scope Controls

Which parts of the company is the red team allowed to attack?

In this project: External IT infrastructure, internal IT infrastructure, employees

5.6.3 Out-of-Scope Controls

Which parts of the company are explicitly excluded from this project?

In this project: Buildings

5.6.4 Permitted Tactics, Techniques & Procedures (TTPs)

What actions are permitted during project implementation?

In this project:

- Enumerating the company's attack surface using publicly available information
- Exploiting technical vulnerabilities, provided there is no prior suspicion that this will impair the availability of the systems
- Contacting employees in phishing campaigns

5.6.5 Excluded Tactics, Techniques & Procedures (TTPs)

Which actions have been explicitly excluded from project implementation?

In this project:

- Contacting employees in their private sphere
- Intentionally carrying out destructive actions

5.7 Implementation – the phases of red teaming

Depending on the defined project rules, the red team selects suitable attack options and techniques. A red teaming project can be divided into several phases, which are explained below. Due to the individual nature of each project, deviations from the described process are possible. For example, phases can be skipped if projects start at an advanced stage on the assumption that an attacker has reached this starting point.

Reconnaissance (information gathering)

In this phase, information about the company is gathered from sources such as company websites, social media profiles, and websites with compromised access data. This approach is known as “open source intelligence (OSINT)”. In addition, technical sources are evaluated to determine the company's IP addresses and domain names as well as accessible services. If physical intrusion attempts are permitted, the red team can gather information about the conditions and typical daily routines via the internet or through on-site observations, for example.

Initial Compromise (gaining access to the internal network)

In this phase, the collected information is analyzed to find and exploit vulnerabilities. Phishing attacks are often used to trick employees into revealing their login credentials or executing files that give the red team access to the company's internal network.

Establish Persistence

After the red team has gained access to the internal network in the previous phase, this phase ensures persistence. For example, it ensures that if the connection to a compromised server is interrupted, the connection can be restored without re-exploiting the vulnerability. This is usually achieved with a command-and-control framework.

Lateral Movement & Escalate Privileges

The Red Team attacks neighboring systems or users in order to gradually move and spread throughout the environment. In this phase, the Red Team typically attempts to gain administrative access to a system. From new starting positions, information about the environment is collected iteratively in order to identify the next attack opportunities.

Complete Mission (achieve project goals)

Depending on the project goals, data is downloaded or access to a specific system, such as a backup server, is verified.

6 Appendix

6.1 Explanations of rating scales

The findings were rated using a risk-based approach. The focus here is on (potential) damage and likelihood. Damage describes the impact that successful exploitation could have. Likelihood describes how easy it is to exploit a vulnerability. The resulting risk rating takes into account damage, probability, and the importance of the affected components.

The rating represent our opinion. The following gradations are used: Very low, Low, Medium, High, Very high.

6.2 Glossary

Begriff	Beschreibung
Beacon	A program used by the red team to communicate with an external control server (team server) in regular communication intervals. Instructions or data are exchanged between the infected system (victim) and the red team. Overall, beaconing enables efficient and inconspicuous control of infected systems via a secure and concealed communication connection.
Blue Team	The company's defense team, which is responsible for detecting and defending against cyberattacks. This is usually the Security Operations Center (SOC).
Critical Functions (CF)	The core functions of the company, the protection of which has the highest priority. The term originally comes from the TIBER framework. The CF are determined during project preparation between the red team and the client and should represent target objects for the red team.
Command-and-Control (C2)	Software used by the red team to manage connections to compromised systems and execute commands via beacons.
Foothold	First access gained by the red team to the company's infrastructure or system.
In-Scope Control	Components of the company that the Red Team is permitted to involve in the project. These can include buildings, employees, and IT infrastructure.

Begriff	Beschreibung
Out-of-Scope Control	Components of the company that the Red Team is explicitly not allowed to involve in the project.
Indicators of Compromise (IoC)	Serve as forensic evidence of a possible intrusion into a system or network. These artifacts can be used to detect intrusion attempts or other malicious activities.
Initial Compromise	Phase in the red teaming project. This is when the red team gains a foothold in the internal company environment for the first time. See also 5.7 Implementation – the phases of red teaming
Initial Recon/Information Gathering	Phase in the red teaming project, see also 5.7 Implementation – the phases of red teaming
Lateral Movement	Part of a phase in the red teaming project, see also 5.7 Implementation – the phases of red teaming
Open Source Intelligence (OSINT)	Part of a phase in the red teaming project, siehe 5.7 Implementation – the phases of red teaming
Persistence	Phase in the red teaming project, see also 5.7 Implementation – the phases of red teaming
Privilege Escalation	Phase in the red teaming project, see also 5.7 Implementation – the phases of red teaming
Red Team	An independent team that simulates real attacks on the company.
Rules of Engagement	Fixed framework and rules agreed between the client and the red team for the red teaming project.
Tactics, Techniques & Procedures (TTP)	Methods used by attackers.
White Team / White Cell	Group of people from the client's company who are aware of the project. They are the Red Team's point of contact and, importantly, are not part of the Blue Team.

7 List of changes

Version	Date	Change	Who
1.0	22.07.24	Release	Christian Stehle

8 Disclaimer

This project was carried out in order to assess the security of the components in focus and to identify weaknesses.

1. This test is a snapshot and not a continuous security monitoring. The security situation may change over time, for example due to changes to the components, disclosed information, new attack techniques or vulnerabilities.
2. The project was carried out within a limited time frame. This may mean that not all potential vulnerabilities and disclosed information were identified.
3. Even though the project was carried out with great care, false positives cannot be completely ruled out.

9 Legal information

MindBytes GmbH | Probststraße 15 | 70567 Stuttgart | Germany

+49 711 20709567 | hallo@mind-bytes.de | https://mind-bytes.de

Local Court: Stuttgart, HRB 790784 | VAT number: DE363069855

Represented by **Christian Stehle, Nina Wagner, Simon Holl**