Ex160 - F4rmC0rpPenTestReport

Simon Tobon

2021-07-08

Contents

Executive Summary

Background

Over the course of the past few months, I, as a member of Pr0b3, was tasked to perform an extensive penetration test on the f4rmc0rp.com domain. Throughout the course of this penetration testing many vulnerabilites were identified as well as exploited. Lots of sensitive data was exfiltrated from the f4rmc0rp domain and was properly reported to the respective persons. This document will outline these vulnerabilities and the results of the penetration test from start to finish.

Overall Posture

Based on the outline of the Pre-Engagement Penetration test agreement, the success and effectiveness of the penetration test was high. The following items were checked off the list Matt Mason provided in our pre-engagment meeting:

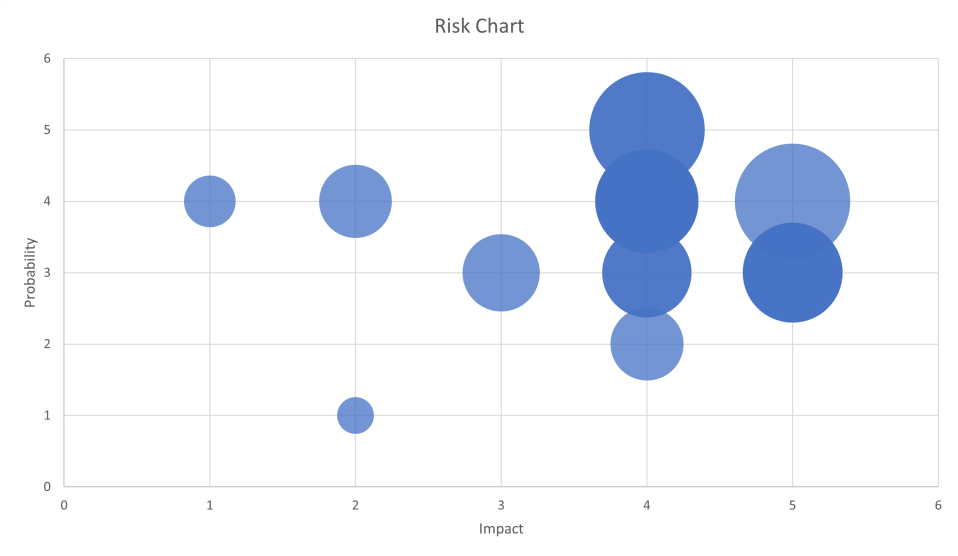
- **The F4rmC0rp network underwent exploitation testing to identify its security.**
- **Linux development servers, Windows Domain Controllers, and Windows Desktops used by employees were all tested against.**
- **Bypassed the "impenetrable" firewall.**

Risk Ranking/Profile

The risk ranking as a qualitative summary of all the vulnerabilities found and exploited can be categorized as Extreme. This is can be claimed as a result of the risks found having **Major** consequences and a **Likely** likelihood of being identified and attacked against.

General Findings

There were many security issues found in the f4rmc0rp domain to represent these issues and their severity's a Risk chart was formulated, the bigger the bubble, the greater the Risk Rating. This was calculated using the Risk matrix approach.



The following risks were measured with respective Risk Ratings (Probability * Impact):

- DNS Misconfiguration - 4
- OpenSSH 7.9 - 2
- Apache 2.4.38 - 15
- vsftpd Backdoor - 16
- Buffer Overflow - 16
- pfsense Default Cred - 20
- BITS service - 15
- Firewall Evasion (Veil) - 16
- Webserver Access Via Pivot - 9
- Remote Desktop Trespass - 16
- SSLStrip - 15
- SU-DOH - 20
- Responder - 12
- BeefHooking - 8
- PHP Injection - 20
- APK Reverse Engineering - 8
- SMB Remote Code Execution - 12

Recommendation Summary

A great way for F4rmC0rp to keep all their software and firmware up to date. In the world of cybersecurity patches are the things that keep things secure. It is almost guaranteed that there will be some exploitation discovered within a patch after a matter of sometime, it is inevitable, but usually they are mitigated swiftly and fully removed by the next patch cycle. By simply keeping things up to date things will be significantly more secure. Also closing ports that are common entries for exploitation is also recommended, if the port is required close it, otherwise try to redirect traffic from these ports in a secure manner. Another recommendation would be to enforcing high quality passwords, and avoid common ”dictionary” passwords.

Strategic Roadmap

1. Update software and firmware.
2. Close unused ports.
3. Redirect traffic on commonly exploited ports.
4. Strengthen user passwords.

Technical Report

Finding #1: DNS Misconfiguration

Risk Rating

- **Damage Potential:** 4
- **Reproducibility:** 10
- **Exploitability:** 7
- **Affected Users:** 7 (hosts)
- **Discoverability:** 10
- **Total Risk Rating (DREAD):** 38 - High

Vulnerability Description

DNS servers hold all the names of servers, IP addresses for domains, and will report these to anyone who queries for it. DNS's can also be manipulated this allows for cache poisoning. Finally DNS release information from internal servers to outside servers, this data could be exfiltrated.

Confirmation method

It is relatively simple to check whether the vulnerability still exists. A quick scan with fierce will reveal any server names that may be released by the DNS misconfiguration. This can be performed with a simple command: **fierce -dns f4rmc0rp.com.** The Fierce domain scanner is a very robust tool, and usually finds many hostnames with its built-in wordlist, but you could also expand your search horizon, by generating a supplemental wordlist with CeWL.

Mitigation or Resolution Strategy

This security problem can be mitigated by keeping resolvers private, keeping name servers up to date, separate authoritative functions from resolving functions by using different servers, also try to use PKI with certificates for authentication, finally always keep track of your name servers, check if there is any unexpected behavior, or status changes.

Finding #2: OpenSSH 7.9

Risk Rating

- **Damage Potential:** 4
- **Reproducibility:** 7
- **Exploitability:** 4
- **Affected Users:** 7
- **Discoverability:** 10
- **Total Risk Rating (DREAD):** 32 - High

Vulnerability Description

In OpenSSH 7.9 there was an issue discovered that allowed for directory traversal. A SCP client performs only cursory validaton of object names, therefore a Man-in-The-Middle attacker can overwrite arbitrary files in the scp target directory, and potentially sub directories too.

Confirmation method

It is relatively simple to check whether the vulnerability still exists. This process could even be automated with some shell scripting. All you have to do is perform an nmap version scan against the host in question. This will list all the services running on the host and their versions, if you apply the –script vuln parameter it will also list any CVE's it has in it's database for this particular version of your service.

Mitigation or Resolution Strategy

This vulnerability in specific can be easily mitigated by simply keeping the OpenSSH service up to date. This sort of exploitation only existed in version 7.9. Another way to mitigate it would be to close SSH ports if not needed, or by simply disabling the OpenSSH service if it is not needed on that host.

Finding #3: Apache 2.4.38

Risk Rating

- **Damage Potential:** 10
- **Reproducibility:** 10
- **Exploitability:** 7
- **Affected Users:** 7
- **Discoverability:** 10
- **Total Risk Rating (DREAD):** 44 - Critical

Vulnerability Description

In Apache server 2.4 releases 2.4.17 to 2.4.38, code executing in less-privileged child processes or threads (including scripts executed by an in-process scripting interpreter) could execute arbitrary code with the privileges of the parent process (usually root) by manipulating the scoreboard. This can lead to easy privilege escalation, and there are many exploits prewritten in exploit databases.

Confirmation method

It is relatively simple to check whether the vulnerability still exists. This process could even be automated with some shell scripting. All you have to do is perform an nmap version scan against the host in question. This will list all the services running on the host and their versions, if you apply the –script vuln parameter it will also list any CVE's it has in it's database for this particular version of your service.

Mitigation or Resolution Strategy

This vulnerability in specific can be easily mitigated by simply keeping the Apache service up to date. Another way to completely resolve this issue would just be to completely disable the Apache HTTP server service if it is not needed.

Finding #4: vsftpd Backdoor

Risk Rating

- **Damage Potential:** 10
- **Reproducibility:** 10
- **Exploitability:** 4
- **Affected Users:** 7
- **Discoverability:** 10
- **Total Risk Rating (DREAD):** 41 - Critical

Vulnerability Description

A vsftpd Backdoor allows an attacker to remotely gain access to a machines file system, allowing them to read and write to it.

Confirmation method

You can check if this vulnerability exists with some scanning software, the software employed in the penetration test was that of OpenVAS. In OpenVAS you can specify the host you want to attack and it will automatically scan that host for vulnerabilities and provide you a list of useful ones.

Mitigation or Resolution Strategy

This vulnerability can also be easily mitigated by simply avoiding vsFTPD 2.3.4. This can be done by keeping vsftpd up to date, or reverting to a previous version where there were less severe or no exploitations associated.

Finding #5: Buffer Overflow

Risk Rating

- **Damage Potential:** 7
- **Reproducibility:** 3
- **Exploitability:** 4
- **Affected Users:** 7
- **Discoverability:** 4
- **Total Risk Rating (DREAD):** 25 - High

Vulnerability Description

Attackers can exploit buffer overflow by overwriting the memory of applications. This even enables them to overwrite areas of executable code and replace entire code blocks, allowing for unexpected inputs and such bypasses.

Confirmation method

You can identify if this vulnerability exists by taking a look at the source code and looking for susceptible buffers, that could be overflowed. C programs are very susceptible to it. You may also be able to see how many bytes are needed to overflow by simply looking at the source code.

Mitigation or Resolution Strategy

The best way to prevent buffer overflow is to use a programming language that does not allow for them. C easily allows buffer overflow because its direct access to memory. If you cannot change languages then write more secure code for example, rather than using strcopy or strcat use strncpy and strncat. These are more secure since they only write to the max size of the buffer.

Finding #6: pfSense Default Credentials/RDP

Risk Rating

- **Damage Potential:** 10
- **Reproducibility:** 10
- **Exploitability:** 10
- **Affected Users:** 10
- **Discoverability:** 7
- **Total Risk Rating (DREAD):** 47 - Critical

Vulnerability Description

If the admin username and password are not changed from the defaults, i.e., admin/pfsense then it is VERY easy for someone to log in to your host. If RDP connections are not properly secured it is fairly easy to inject ransomware.

Confirmation method

This vulnerability can be checked for existence by simply attempting to log in to a pfsense console with the default credentials, admin/pfsense. RDP connections are usually easy to get into if you have prior knowledge of the system and credentials. It is important to note that you can RDP into a host from the exterior because you were able to login to their innerouter and portforward port 3389 to an external address.

Mitigation or Resolution Strategy

This kind of exploit can be mitigated by changing the default login credentials for pfsense on the router. This will make it much harder for someone to login and have full control over the system. It can also be mitigated by blocking traffic on port 3389 with a firewall.

Finding #7: PowerUP and The BITS service

Risk Rating

- **Damage Potential:** 10
- **Reproducibility:** 10
- **Exploitability:** 10
- **Affected Users:** 10
- **Discoverability:** 4
- **Total Risk Rating (DREAD):** 44 - Critical

Vulnerability Description

By using PowerUp I identified that the vulnerable service on this Windows machine was the BITS service. This vulnerability can be exploited to elevate privileges by creating an Administrator account.

Confirmation method

You can check if this vulnerability persists by deploying PowerUp on the machine of interest. PowerUp is a Powershell module that identifies Windows host misconfigurations.

Mitigation or Resolution Strategy

There are several ways this vulnerability can be mitigated and even resolved:

- The first would be to change the innerrouter's credentials from the default credentials to something more complex.
- You could also block all traffic on port 3389, rendering port forwarding for RDP impossible, or disable portforwarding all together if possible.
- Another thing that could be done is disabling the BITS service.
- Finally, restrict administrative privileges to a single, secure account.

Finding #8: Firewall Evasion via Veil

Risk Rating

- **Damage Potential:** 10
- **Reproducibility:** 7
- **Exploitability:** 4
- **Affected Users:** 10
- **Discoverability:** 7
- **Total Risk Rating (DREAD):** 38 - High

Vulnerability Description

The Veil evasion framework creates deployable payloads that can inject into a host under an antivirus' nose. They are hard to detect and very effective. In this case we used a reverse_tcp payload which allows us to create a TCP shell.

Confirmation method

To confirm this vulnerability exists one could use netcat to attempt to connect to TCP ports on a server name. If it is possible, then a reverse_tcp payload can be snook under the antivirus with Veil.

Mitigation or Resolution Strategy

This sort of attack could be mitigated by disabling RDP, and also closing ports 443 and 4444 (TCP) to not allow outside traffic and therefore prevent attacks.

Finding #9: Webserver Access via NetCat Pivot

Risk Rating

- **Damage Potential:** 10
- **Reproducibility:** 7
- **Exploitability:** 4
- **Affected Users:** 4
- **Discoverability:** 4
- **Total Risk Rating (DREAD):** 29 - High

Vulnerability Description

A web server can be accessible from an outside machine via a Netcat Pivot. This enables attackers to exfiltrate potentially disclosed information, and learn more about the network.

Confirmation method

To confirm this vulnerability exists one could use netcat to attempt to connect to HTTP ports (80) on a server name. If that is possible and the network is not tightly monitored, it is fairly simple to set up a Netcat pivot with a FIFO pipe. It is also important to note, not much data could be exfiltrated if the webserver required an administrative login once accessed.

Mitigation or Resolution Strategy

This attack can be mitigated by closing port 80. It can also be mitigated by locking all web pages under an administrative login.

Finding #10: Remote Desktop Trespassing via Sticky Keys

Risk Rating

- **Damage Potential:** 10
- **Reproducibility:** 10
- **Exploitability:** 7
- **Affected Users:** 1
- **Discoverability:** 7
- **Total Risk Rating (DREAD):** 35 - High

Vulnerability Description

The sticky keys executable is ran by the **NT AUTHORITY/System** user. This can lead to some problems if you can either change the **sethc.exe** programs binary or the registry entry for the sticky keys. There are also requirements for modifying this binary executable, however, **cmd.exe** happens to fulfill all of them, meaning you can essentially run an Administrative Command Prompt with this method.

Confirmation method

This vulnerability is not very common place. The only reason something like this would be employed by someone other than the penetration tester would be if a user forgot their login information to the Windows machine. You can verify the vulnerability exists by pressing the **Shift** key 5 times. If the command prompt opens up rather than the usual Sticky Keys prompt then someone may have forgot their password and used this to reset their credentials and forgot to change the binary executable back to its intended state.

Mitigation or Resolution Strategy

Never overwrite the binary or registry entry for the Sticky Keys executable. Or if you **DO** be sure to change the executable to its original state.

Finding #11: SSLStrip/Arpspoof

Risk Rating

- **Damage Potential:** 7
- **Reproducibility:** 6
- **Exploitability:** 5
- **Affected Users:** 4
- **Discoverability:** 8
- **Total Risk Rating (DREAD):** 30 - High

Vulnerability Description

SSLStrip is used to downgrade HTTPS to HTTP. If a HTTPS service is found to be susceptible to SSLStrip we can use this to gain lots of information in regards to the webpage and possibly exfiltrate sensitive data. Arpspoof is used to get in the middle of HTTPS packets and SSLStrip.

Confirmation method

You can verify that SSLStrip is possible by using Wireshark to sniff HTTPS packets, that is packets from a TCP Source Port 443. Also verify that you can utilize ArpSpoof and IP forwarding.

Mitigation or Resolution Strategy

This attack can be mitigated, by disabling and not allowing IP forwarding to be performed on the system. You can also encrypt HTTP traffic making it harder for SSLStrip. Also, if possible, have your entire web application under HTTPS, rather than a landing page in HTTP.

Finding #12: SU-DOH

Risk Rating

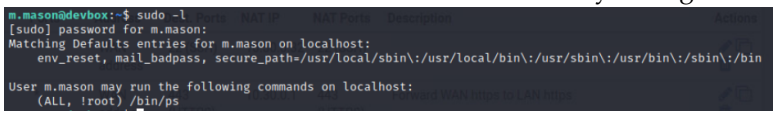
- **Damage Potential:** 10
- **Reproducibility:** 8
- **Exploitability:** 7
- **Affected Users:** 6
- **Discoverability:** 7
- **Total Risk Rating (DREAD):** 38 - High

Vulnerability Description

In sudo versions before 1.8.28, there is a vulnerability where we can bypass a deny on a command as root. By running the command as the kernel, i.e., -1. We can by pass the restriction and run the command as "root."

Confirmation method

To verify that this exploit is possible, first you must check that your sudo version is a version 1.8.27 or earlier, as this was patched in sudo 1.8.28, and onward. This can be done by running the following command: **sudo -V.** You must check the sudoers file to see if there is any configuration as follows:



If this is the case then you can rewrite the binary of the executable to whatever you'd like, i.e., a root shell.

Mitigation or Resolution Strategy

This attack can be easily mitigated by keeping your sudo version up to date. As mentioned earlier, this vulnerability only exists in versions of sudo prior to 1.8.28.

Finding #13: Link-Local Multicast Name Resolution Exploit (Responder)

Risk Rating

- **Damage Potential:** 10
- **Reproducibility:** 5
- **Exploitability:** 10
- **Affected Users:** 7
- **Discoverability:** 6
- **Total Risk Rating (DREAD):** 38 - High

Vulnerability Description

If a DNS fails to provide a response for a host lookup on a subnet then it will attempt to resolve the name, this can be intercepted by a MiTM attack and potentially exfiltrate user credentials or other sensitive data. Responder is a python program that enables attackers to easily deploy an attack with very little knowledge. This program can receive poisoned answers from the host.

Confirmation method

You can verify that the vulnerability exists if the machine has LLMNR enabled on the DNS client.

Mitigation or Resolution Strategy

This attack can be mitigated by disabling LLMNR with a group policy in the Computer configuration. Deploy some form of Network intrusion detection systems to identify traffic patterns indicative of Man-in-the-Middle attacks. By isolating infrastructure components that do not require broad network access, we can mitigate the attack also.

Finding #14: XSS and Browser Exploitation (Beef)

Risk Rating

- **Damage Potential:** 6
- **Reproducibility:** 7
- **Exploitability:** 5
- **Affected Users:** 7
- **Discoverability:** 6
- **Total Risk Rating (DREAD):** 31 - High

Vulnerability Description

BeEF allows for typical XSS and many other exploit modules such as history gathering, intelligence, network recon, browser plugin detection and cookie withdrawal. It can go as far as hijacking someones entire web browser, allowing for severe exploitation and data breaches.

Confirmation method

If your browser allows for JavaScript on all sites, then a BeEf injection is possible.

Mitigation or Resolution Strategy

This attack can be mitigated by not allowing your browser to run JavaScript on any site, only allow it on trusted sites. It is important to only access sites you can trust, for this reason. It is also important to take measures against XSS on your own web servers. Use HTTPOnly cookies, as they are not accessible by JavaScript. Use a restrictive Content Security Policy.

Finding #15: XSS via PHP Injection

Risk Rating

- **Damage Potential:** 10
- **Reproducibility:** 10
- **Exploitability:** 6
- **Affected Users:** 5
- **Discoverability:** 5
- **Total Risk Rating (DREAD):** 36 - High

Vulnerability Description

The vulnerability here was that there was no server side validation. This means that we can use PHP injection methods to execute shell commands and alter the behavior of functions. With no server side execution we can essentially modify program functions on the web page to do what they were not intended to. For example a site with an image upload function that may check for file extensions such as JPG, PNG, etc, can be easily overwritten to accept any file upload. This can allow for the planting of dangerous files such as a PHP file with shell commands to completely refactor/rewrite a web server file system.

Confirmation method

You can verify the existence of this vulnerability by trying to perform it.

Mitigation or Resolution Strategy

This could be mitigated by having server side validation, not creating PHP scripts that read arbitrary files, and maybe use tokens for images, rather than absolute file paths, finally separate static files from executable files.

Finding #16: APK Reverse Engineering

Risk Rating

- **Damage Potential:** 6
- **Reproducibility:** 8
- **Exploitability:** 9
- **Affected Users:** 3
- **Discoverability:** 3
- **Total Risk Rating (DREAD):** 29 - High

Vulnerability Description

The risk arises from not following the OWASP recommendations for mobile app development. In this particular case, there was crucial data that should not have been included in the application source which led to easy exploitation and a breach of privacy. Specifically, login credentials were included in the source (not raw, but, encrypted).

Confirmation method

By using a program like jadx you can examine an APK's source code. From there you can investigate any potential useful information, and any malpractices, that is things that do not follow the OWASP recommendations for mobile app development.

Mitigation or Resolution Strategy

This could be easily avoided if the credentials were not set inside the application source, perhaps they could be set as an Environmental variable or on the backend, i.e., database, also do not use something as trite and trivial as Base 64 encryption, this can be easily decoded.

Finding #17: SMB Remote Code Execution

Risk Rating

- **Damage Potential:** 10
- **Reproducibility:** 10
- **Exploitability:** 5
- **Affected Users:** 7
- **Discoverability:** 7
- **Total Risk Rating (DREAD):** 39 - High

Vulnerability Description

This vulnerability allows for remote attackers to execute arbitrary code via SMB. Potentially enabling chances for privilege escalation and more. It is a very famous exploit known as Eternal Romance.

Confirmation method

You can verify that the vulnerability exists by checking that port 445 is accessible on the host and if the Microsoft server deployed uses SMBv1.

Mitigation or Resolution Strategy

This can be mitigated by keeping your Microsoft Windows servers up to date, and disabling SMB compression.