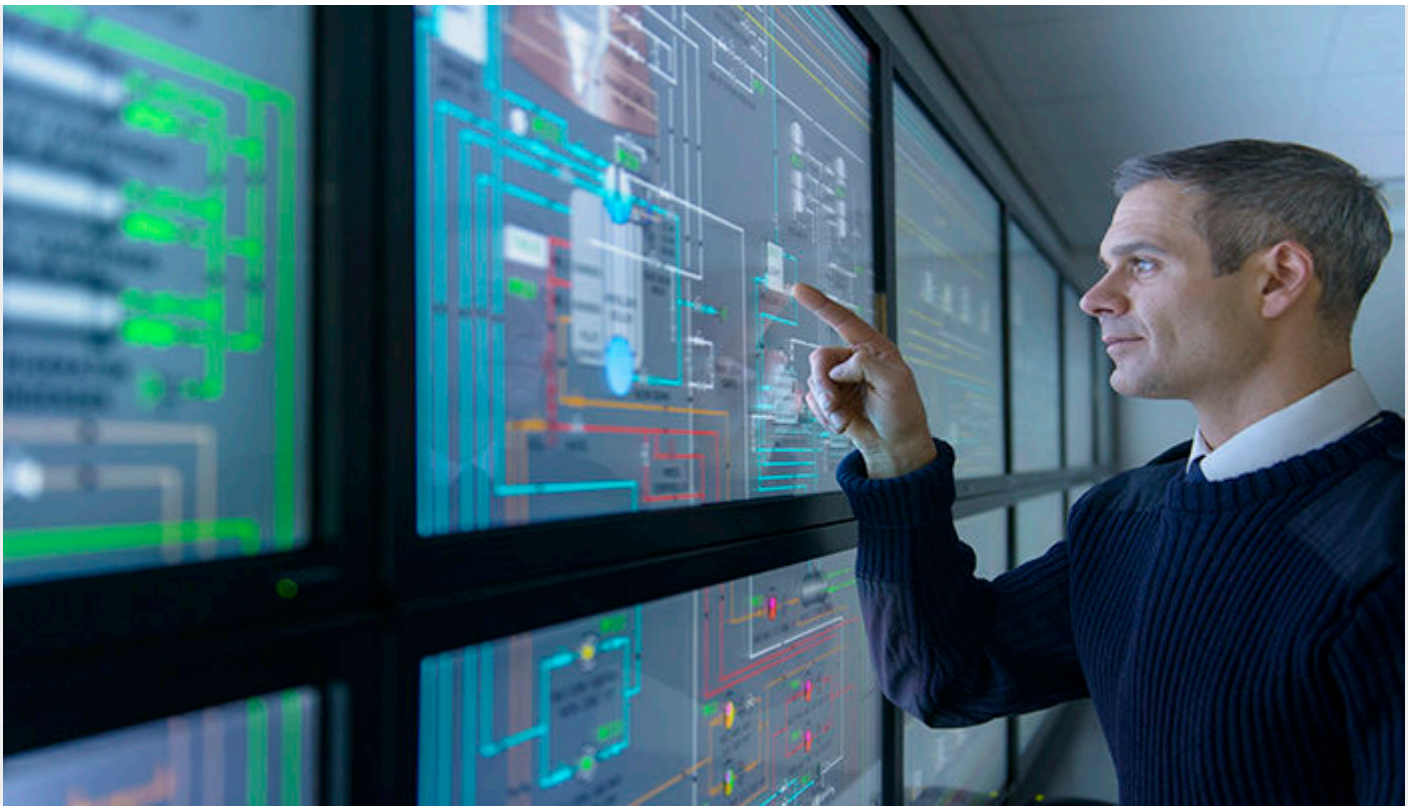
TOP 6 WAYS A HACKER WILL GAIN ACCESS TO YOUR MAINFRAME



Analyzing your mainframe attack surface

In my last blog I questioned whether your mainframe was your weakest link[1]. In order to answer this question effectively, you first need to understand your mainframe's vulnerabilities that a threat actor can leverage to compromise your system. In this blog, we are going to focus on your mainframe's attack surface, which is the summation of all potential points a hacker could exploit to gain initial access to a system. More clearly, these are the most common ways someone can hack into your mainframe:

1. An emulated Telnet 3270 (TN3270) terminal is the most common way for a system programmer to connect to their mainframe for entering commands and running programs. This functionality makes it one of the first targets for a hacker looking to gain access to the system. Many successful hackers are able to gain their initial entry using a technique called password spraying[2] which is a modern adaptation of brute force login. Instead of trying a million passwords against a single user, which usually locks out after 5, the hacker will try a single password against every user on the system. Since mainframes often have hundreds of thousands of users, the hacker just needs to guess one or two commonly used passwords against all these accounts. While this method is not precise, the hacker only needs to find one user using a weak password to successfully gain that initial access to the mainframe.

2. The File Transfer Protocol (FTP) method for uploading, downloading, and managing files is well known among distributed system administrators but FTP is uniquely powerful on z/OS because it has the ability to submit commands to the mainframe through the Job Command Language. This enables a hacker who finds login credentials to have remote code execution to encrypt files, steal information, or build a robust shell to gain persistence over the machine. Unfortunately, relying on credentials in RACF, ACF2, or Top Secret is insufficient. We are decades away from a system programmer walking to the data center and plugging in directly to a machine. Today, system programmers use Linux or Windows personal computers and log in remotely. A hacker who successfully gains access to a victim system programmer could drop a keylogger[3] and wait for the system programmer's next day at work where they log into their machine. The hacker now has active privileged credentials and can submit any commands to the mainframe through FTP. Thanks to the necessary permissions granted to system programmers to effectively do their job, this gives the hacker almost complete control over the mainframe.

3. Another similarity between the mainframe and distributed systems is the commonality of running websites over the ubiquitous Hyper Text Transfer Protocol (HTTP). Modern websites host a growing suite of features designed to improve the user experience. The increasing code and functionality increase the chance that the website contains a vulnerability that would enable remote code execution through common techniques like remote file inclusions[4], SQL Injection[5] , or broken authentication and session management. The hacker can even leverage publicly available or 0day vulnerabilities in the web server like apache or tomcat itself. Thanks to most website being public facing, hackers have decades of experience building tools and penetration testing this service. Those skills can be directly applied with equal success against the mainframe as an initial attack vector.

4. The mainframe's Customer Interaction Control System (CICS) Transaction Server is unique to the mainframe and offers a front end for customers to interact directly with the mainframe. Similar to the early days of breaking websites on HTTP Servers, hackers are starting to build automated tools to enumerate as much of the CICS Transaction Server so they can to identify one of the numerous misconfigurations that would enable them to bypass authentication or brute force an account log in. One open source tool, named CICSpwn, will automatically retrieve the security settings running on the underlying z/OS operating system, read available files, enumerate system naming conventions, and even remotely execute code [6]. Thanks to the automation in the tool, an experienced hacker who is new to the mainframe will find it similar to the rest of their arsenal and find a lower learning curve for successfully gaining initial access to the mainframe through the CICS Transaction Server.

5. The Network Job Entry (NJE) is another feature unique to the mainframe that enables one trusted mainframe to send a job to any other mainframe its connected to. This is extremely powerful and mainframe penetration testers have found significant success through NJE to either spoof a known mainframe to submit a job and gain access or to laterally transfer between mainframes. Lateral transfers are especially dangerous because many companies with constrained resources will only protect their production machines while failing to monitor their development machines. If a hacker gains access to a development machine that is trusted it can easily leverage the NJE protocol for immediate access to any other mainframe.

6. Finally, the Secure Shell (SSH) protocol is also available on the mainframe. Hackers are very successful in pilfering credentials and using the native features in SSH to propagate through a network by simply logging in with SSH. Today, one of the most common techniques is called credential stuffing, as opposed to spraying, which involves taking the usernames and password of old breaches and applying them to new systems[7]. Since users have to remember dozens of passwords they inevitably begin reusing them which enables a breach in one company to have cascading effects throughout organizations that have shared customers.

Whether you are a system programmer or a member of the enterprise security team, you need to consider that the software on your mainframe has many of the same vulnerabilities as the distributed systems that comprise the majority of the organization. This means that the advanced persistent threats which have significantly matured over the past few decades will be extremely adept at gaining access to the device running a company's most critical programs and manipulating the most critical data.

To adequately defend this critical system, organizations need to treat the mainframe like the rest of their enterprise. This requires real-time visibility into their mainframe's security events, employment of an intelligent Endpoint Detection and Response (EDR) solution and testing their defenses with professional penetration testers. These basic measures, already applied throughout the rest of the enterprise, are only the first step in developing a holistic security plan and avoiding the severe repercussions of a major breach.

[1] https://www.bmc.com/blogs/are-mainframes-your-weakest-link/

[2] https://attack.mitre.org/techniques/T1110/

[3] https://www.veracode.com/security/keylogger

[4] https://kellgon.com/own-a-server-with-a-remote-file-inclusion/

[5] https://www.owasp.org/index.php/SQL_Injection/

[6] https://github.com/ayoul3/cicspwn/

[7] https://www.owasp.org/index.php/Credential_stuffing/