CTCH 605 Project 3
Introduction to Cybersecurity

Security Event and Incident Report

Prepared By: Shomoi Francis

Version 2.0

Introduction

The goal of this investigation was to investigate suspicious behaviors reported at FICBANK, such as file renaming, system reboots, and slowdowns, to see if a security compromise happened. The goal was to identify potential Indicators of Compromise (IoCs) by analyzing digital artifacts, network traffic, and file integrity with forensic tools such as TrID and Wireshark, uncover any malware or unauthorized access, and ensure the investigation was in line with industry best practices and regulatory requirements. Finally, this procedure helps secure FICBANK's systems and sensitive data from future threats while also ensuring compliance with cybersecurity regulations.

Objectives

Identify potential Indicators of Compromise (IoCs) in FICBANK’s systems architecture.
Conduct a preliminary investigation using fundamental digital forensics techniques.
Document your findings using the Security Event and Incident Report Template.

Definitions

Ransomware: Malware that encrypts files and demands payment to restore access.
Indicators of Compromise (IoCs): Signs of a potential security breach, such as unusual file changes or network activity.
Malware: Malicious software designed to damage or exploit systems, including ransomware and viruses.
Denial of Service (DoS)/DDoS: Attacks that overwhelm systems, causing slowdowns or making them unavailable.

Packet Capture (PCAP): A file format used to capture and analyze network traffic.
Command and Control (C2): A method used by attackers to remotely manage infected machines.

Predictions

What types of indicators of compromise do you predict might be found during the investigation? Particularly ransomware, the enigmatic renaming of files could point to illegal access or viruses, either of which could compromise files. Random system reboots can indicate hardware modification, malware, or denial-of-service attacks compromising the system. Dramatic slowdowns in processing might point to the presence of resource-intensive malware including a botnet or crypto miner. Additionally expected to be installed on the systems are unidentified or rough apps.

What tools, techniques, and practices do you predict will be most useful in identifying the root cause of the reported events? Wireshark is one of the tools, methods, and instruments I think would be rather helpful in determining the underlying cause of the documented incidents. Highly helpful for spotting suspicious outbound connections or malicious network activities, Wireshark is a network traffic analysis tool that records and examines packet level data. Some excellent practices would be tracking user behavior particularly for attempts at illegal access or privilege escalations. To find any illegal changes, be sure to do baseline checks between the present situation of systems and files versus known good baseline. To conduct thorough forensic investigation, grab disk pictures and memory dumps from impacted computers; this will highlight either concealed or erased malicious artifacts.

What do you expect the outcomes to be from these tools, techniques, and practices? The outcome I expect is the identification of malicious activity, detection of network anomalies, forensic evidence collection, mitigation of threats, enhanced security awareness and improved incident response preparedness.

Based on your knowledge of common security incidents, what do you predict might have occurred at FICBANK? Several possible security events might have happened based on the odd activity recorded by FICBANK: irregular system reboots, unexplained file renaming, and major slowdowns. A virus infection—probably ransomware—would explain the file renaming, or a Trojan or rootkit generating system instability. Another possibility is illegal access—from an outside assailant or a malevolent insider—using compromised credentials to carry out illegal activities, hence perhaps slowing down systems. Botnet activity or cryptojacking could potentially be involved; systems being utilized for illegal activities like bitcoin mining causes performance degradation. By overwhelming the systems, a Denial of Service (DoS) or Distributed DoS (DDoS) attack could also be accountable for the reboots and slowdowns. Furthermore, systems could have been compromised by misconfiguration or vulnerability exploitation, therefore enabling attackers to enter and cause disturbance of operations. Finally, data exfiltration is a possibility as financial institutions are often targets for data theft and the unusual activity could be an attempt to hide continuous data breaches. Common in security events, these situations fit the criteria documented.

Events and Indicators of Compromise (IoC) Work Products

Dradis:
Using Dradis, I appreciated how easy it was to create an issue, but at first it was a little learning curve; I'm a new user and I think it will require time before I master it. I loved the tagging system as it helps prioritize the most critical issues. I also explored a little bit and saw that it integrates with popular forensic and security tools like Nessus, Metasploit, Burp Suite, and others.
Attached Screen Shots:

Wireshark:

Figure 1 The traffic sample reveals a potentially malicious interaction where a client makes a small POST request to upload a file, and the server responds with what is supposed to be a JPEG image but is actually a Windows executable, as indicated by the "MZ" signature in the binary data. Despite the response being labeled as an image, the presence of executable content disguised in this way strongly suggests a malware delivery attempt. The use of Russian language headers in the request may provide additional context regarding the origin of the activity. This technique of masquerading an executable as an image is a common tactic used in cyberattacks to evade detection and compromise systems. Given these indicators, the traffic likely represents part of a broader attack chain aimed at distributing malicious software.

Figure 3 : The traffic sample shows the transfer of a Windows Portable Executable (PE) file, confirmed by the "This program cannot be run in DOS mode," along with the "MZ" signature common to EXE or DLL files, therefore indicating its executable character. This revelation raises questions as, particularly in cases with no official justification for their distribution, executable files in network traffic sometimes indicate harmful behavior. Such a file points to it maybe being part of a malware distribution, maybe hidden as a picture or other innocuous information to go undetectable. Considering the background of the previous HTTP stream, which also included binary data beginning with "MZ," it seems this executable might have been sent via an assault using a standard method of concealing harmful payloads inside apparently authorized downloads. This calls serious questions regarding the possibility of malware spread and calls for more research.

TrID:
Index
.EXE : Executable File
.DLL : Dynamic Link Library (for executable code)
1.jpg: Identified as a Microsoft Visual C++ built executable (32.2% confidence) as well as other executable file types (.EXE and.DLL). This indicates that the file is likely malware masquerading as an

innocuous image file.

2.jpg: is classified as a Win64 executable (40.3% confidence) and other executable formats. This confirms the discovery that this file is likewise an executable disguised as a JPEG image.

3.jpg: is classified as a Win64 executable (40.3% confidence) and other executable formats. This confirms the discovery that this file is likewise an executable disguised as a JPEG image.

4.jpg: Identified as a Win32 executable with 47.3% confidence. Other executable types include Win64 and DLLs. This means that 4.jpg is most likely a malicious file masquerading as an image.

5.jpg: This file is classed as a Win64 executable (32.2% confidence) with a high likelihood of being a DLL (20.1%). Like the preceding files, 5.jpg is most likely malware disguised as a.jpg image.

6.jpg: Identified largely as a Win64 executable (32.2% confidence), with.DLL components. This suggests a potentially hazardous program masquerading as an image.
7.jpg: Detected as a Win32 executable (47.3% confidence), indicating that it is another malicious executable masquerading as a.jpg file.

Log Analysis:
The investigation of the log file and the given artifacts suggests that the assault started from multiple dubious IP addresses, most famously 61.161.130.169. These IP addresses repeatedly sought to take advantage of web server weaknesses. Specifically, the 61.161.130.241 request reveals indications of a Bash injection effort most likely connected to the Shellshock vulnerability. Suggesting that the assailant was trying to take over the server by running remote code, this exploit comprised a command to download a file from a distant server (61.160.212) and run a malicious script called "China.Z-ionw."

Other log entries show that the assailant was also searching for weak files, including XMLrpc.php, a typical target on Linux sites. Frequent404 (Not Found) mistakes point to the attacker most likely searching for holes to take advantage of, thereby indicating known vulnerabilities. There also were requests from many other IP addresses that seem to be part of a bigger, scattered effort to test other system access points.

From IPs 61.161.130 and 169.50.3.171, the Wireshark grab displays direct traffic. The data in the logs makes it abundantly evident that the assailant was trying to hack the system by using recognized

weaknesses and running hostile scripts. This conduct suggests a deliberate attempt to either compromise the server or create endurance for next assaults. These results ought to be included in a more general security analysis and reaction plan.

Reflections

General:
What digital forensic tools did you use during the investigation? The digital Forensics tools I used during the investigation is Dradis, Wireshark, and TrID.

What was your approach to identifying potential IoCs? I initially looked at the HTTP traffic structure and headers to find possible IoCs, noting variations such a JPEG image file providing binary executable content identified by the “MZ” signature, common of Windows Portable executable (PE) files. Acknowledging this disparity caused one to be suspicious about virus distribution. Additional background was given by my analysis of the short POST size of the request and the Russian language headers. This traffic was identified as possibly hostile since the executable passed for an image fit for known attack strategies applied in malware dissemination.

How did you verify the credibility of the artifacts from FICBANK's employees? Several methods were taken to ensure the authenticity of the artifacts provided by FICBANK employees. First, system and network logs were inspected to ensure that reported events, such as Wireshark were utilized to detect any unusual traffic or malware communication. I also used the TrID tool to determine the exact file types causing the suspicious file renaming occurrences. This helped determine whether the renamed files were legitimate or if they had been tampered with or camouflaged with malware.

What challenges did you face during the investigation? Several problems arose while investigating FICBANK's reported occurrences. First, the lack of detail in initial reports from employees made it impossible to quickly identify the source of the anomalous events, such as file renaming and system slowdowns. This necessitated considerable log analysis to reconcile reported occurrences with real system activity. Second, employing TrID for file identification posed difficulties when files were modified with or renamed by malware, necessitating thorough examination to discover their real nature. Furthermore, determining whether the suspicious files were part of a broader attack required cross-referencing file hashes with known malware databases. Another problem was examining network traffic for evidence of command-and-control communication, which took time and required complex tools like Wireshark to filter through massive amounts of data for potential Indicators of Compromise (IoCs). Finally, resource limits at FICBANK constituted a problem, since limited infrastructure necessitated efficient use of existing tools and skills.

How did you ensure that the investigation met industry standards? To ensure that the inquiry adhered to industry standards, I followed chain of custody protocols to protect the integrity of all digital evidence. Logs, files, and network captures were securely captured and stored to maintain legal and regulatory compliance, which is especially important for financial institutions such as FICBANK. I also used recognized forensic tools, such as TrID for file identification and Wireshark for network analysis. These instruments are industry-approved, ensuring the investigation's accuracy. Furthermore, I followed a defined incident response framework based on NIST principles, ensuring that each stage of the inquiry was methodical and in accordance with best practices. Finally, I cross-referenced findings with current threat intelligence and malware databases to ensure that Indicators of Compromise (IoCs) were examined using the most recent cybersecurity standards, thereby confirming the investigation.