CrowdStrike Channel File 291 Incident Analysis

Christian J. Rudder, Ivan Khramtchenko, Ethan Machleder, Chris Min, Christian Rudder

November 2024

Contents

This page is left intentionally blank.

1.1 Channel File 291 Incident

Friday July 19, 2024, CrowdStrike, a leading cybersecurity firm, released an update to its Falcon Sensor software designed for Windows-based systems. The update, intending to enhance threat detection, single-handedly disrupted the global IT infrastructure, causing widespread system crashes across various industries (e.g., airlines, healthcare, and banking) [10].

The outage costed fortune **500 companies approximately $5.4 billion** in damages, after an estimated **8.5 million devices** were struck by the blue screen of death (BSoD) [24][14].

1.2 Context and Background

CrowdStrike's Founder and CEO George Kurtz, addressed the public on live TV, stating "That we're deeply sorry for the impact we've caused" [23]. Though this isn't the first time George Kurtz has been caught in the crossfire of a cybersecurity incident. In 2010, George Kurts was the Executive Vice President and Chief Technology Officer at McAfee when they released a faulty update, also causing a BSoD. The update, mistakenly identified a critical Windows system file (svchost.exe) as malware, causing the system to endlessly loop [25].

In a blog posted by CrowdStrike, "as of 8:00 p.m. EDT on July 29, 2024, 99% of Windows sensors were back online" [8]. The incident occurred due to a bug which expected 20 input fields instead of 21, causing the software to crash. Channel File 291 was identified as the culprit and was removed from the software [8].

1.3 Modern Endpoint Security

CrowdStrike, as a cybersecurity company, offers endpoint security in the form of their Falcon solution. Falcon is a lightweight agent installed on endpoint devices to monitor and record system activity. Securing **endpoints** means securing devices that connect to a network, such as laptops, desktops, and mobile devices, and thus, this is known as **endpoint security**. 90% of successful breaches and 70% of data breaches originate at an endpoint, costing companies millions [12].

Since many large scale companies have moved to the cloud, and many employees are working from home, ever more devices are connecting to sensitive data. This means many endpoint solutions have to constantly monitor and record system activity to detect and prevent threats. This in itself utilizes cloud-based systems to access threat intelligence, providing autonomous real-time protection [3]

Many original anti-viruses signature based, protecting endpoints on a device by scanning for known malware—**Indicators of Attack (IOA)**. This relied on a database of known malware signatures. CrowdStrike and many others have opted to use **next-gen anti-virus (NGAV)** machine learning technology, to aid in detecting newer types of IOA vectors that are often fileless, by monitoring system memory [13].

1.4 EPP & EDR Anti-virus Solutions

Endpoint Protection Platforms (EPP) like CrowdStrike may include the following features as defined by IBM [12]:

- **Web control and content filtering**: protects against malicious code in websites and user downloaded content. While providing a whitelist of approved websites.
- **Data classification and data loss prevention (DLP)**: identifies and classifies sensitive data, preventing unauthorized access and data loss.
- **Firewall**: monitors and controls incoming and outgoing network traffic based on configured security rules.
- **Email Gateway**: scans incoming email attachments and links for malicious content.
- **Application control**: restricts the programs that users can run on their devices.

CrowdStrike's **Endpoint Detection and Response (EDR)** solution Falcon Sensor, part of the Falcon Platform, is at the root of the issue. EDRs are a class of security tools that go beyond known threats, to monitor files entering and applications running on a system. These **Correlate Indicators of Compromise (IOC)** systems, aggregate data from various sources—network traffic, unusual user behavior, inconsistent permissions, system configuration changes, unverified software or domains, repetitive file access, and more [16].

These systems aren't just designed to detect threats, but respond to them while an attack is in progress. This incurs IOCs many log based solutions such as **extended detection and response (XDR)**, and **security information and event management (SIEM)** systems to mitigate and isolate threats, going as far as to shut down a system if necessary. These systems typically rely on AI to establish a baseline of normal activity and detect deviations from it [16].

1.5 CrowdStrike Falcon Sensor Distinctions

The immediate reason why CrowdStrike's Falcon Sensor update caused such a widespread outage was due to the software living on at kernel level. These differ from traditional user-mode applications that crash in isolation. Kernel-level applications are more privileged living at the heart of the operating system—if it crashes, the whole system crashes. This process is known as **kernel panic**, which stops the system from potentially corrupting beyond repair [1].

CrowdStrike notes three main kernel-level component features that it complies with from the Microsoft's anti-virus kernel APIs [13]:

- **Kernel Patch Protection (KPP)**: also known as **PatchGuard**, prevents third-party software from modifying the Windows kernel. Available on 64-bit (x64) Window systems, but by-passable on 32-bit (x86) systems, CrowdStrike opts to never patch the kernel. Other anti-virus choose this route, which can lead to system instability if not done correctly [4].
- **Kernel-Mode Code Signing (KMCS)**: CrowdStrike complies with Microsoft's KMCS requirements, which ensures that all kernel-mode code obtains a **Extended Validation (EV) Code Signing Certificate** from a trusted **Certificate Authority (CA)** [17][21].

- **Object Callbacks:** a feature that allows CrowdStrike to subscribe to various kernel events, such as file creation, registry access, and network activity. Instead of **kernel hooking**, which intercepts system calls [15].

CrowdStrike further justifies its kernel presence to protect against for **Early Boot Protection (EBP)** and which Microsoft supports with its **Early Launch Anti-Malware (ELAM)** driver [13]. This protects against rootkits, which are malware that can hide from the operating system, by loading before the operating system itself. Having this protection in place, stops attackers from sticking USBs into airport kiosks or hotel computers [2].

1.6 Faulty Code and Production Updates:

Despite CrowdStrike Falcon Sensor passing **Windows Hardware Compatibility Program (WHCP)** certifications, and validations through **Windows Hardware Lab Kit (HLK)** testing, the update still slipped through [18]. However, anti-virus software only needs to pass the WHCP and HLK tests once. They only certify that the driver running on the system is stable. Once the driver is installed, CrowdStrike can push updates—**without re-certification**—in a cloud-based process they call **Rapid Response Content (RRC)** [9].

This process entirely relies on CrowdStrike's internal deploy and testing pipelines. CrowdStrike's testing negligence, did not thoroughly test the update, during their RRC update to its Falcon Sensor software. Typically a software companies employ **Continuous Integration/Continuous Deployment (CI/CD)** pipelines to catch these issues. CI/CD pipelines are characterized by multiple rounds of manual written tests, peer reviewed code, and automated stress testing to ensure the software compiles safely before deployment [11]. These tests comprise of **unit tests**, **integration tests**, and **end-to-end (E2E) tests** to ensure the software is functioning as expected. Unit tests test individual components of the software, integration tests test how the components interact, and E2E, running the application in production-like environments (e.g., test and development environments before production) in a suite of integration tests to ensure stability. CrowdStrike lacked the CI/CD security to catch Channel File 291, a bug that effectively threw an out-of-bounds error.

1.7 CrowdStrike Incident Report

Before jumping into CrowdStrike's incident analysis, we must first understand the technical terms used in the analysis [9]:

1. **Falcon OverWatch® and Falcon Complete™**: OverWatch is a 24/7 managed threat hunting service led by human intelligence. Complete is a **managed detection and response (MDR)** service, which is a suite of tools and services including OverWatch, to detect and remediate threats [5][6][7].

2. **Security Telemetry & Graph Store**: Telemetry is the aggregation of data from various sources (e.g., endpoints, servers, network devices) [20]. These analytics are are stored locally on the Falcon Sensor's sensors in a graph database. Graph databases store information in nodes and edges to represent the relationship between data points [19].

Falcon Sensor's interprets relationships in flowing memory to detect possible IOAs.

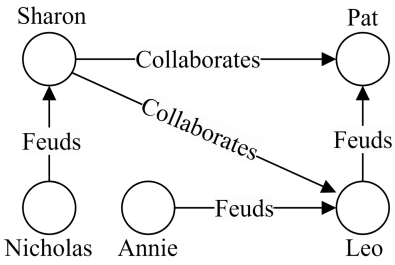

Figure 1.1: Graph Example Depicting Relationships Between Co-workers

3. **Template Types**: Template types push telemetry data to a regular-expression (RegEx) engine Content Interpreter. Template types outline predefined schemas of security criteria to evaluate. **Template Instances** are live configurations, defining parameters, thresholds, and conditions to trigger alerts.

4. **Channel Files**: Each Falcon sensor defines a channel file, which is configured with a template type. RRCs then push threat intelligence to channel files. Template instances then compare the security telemetry to determine IOAs.

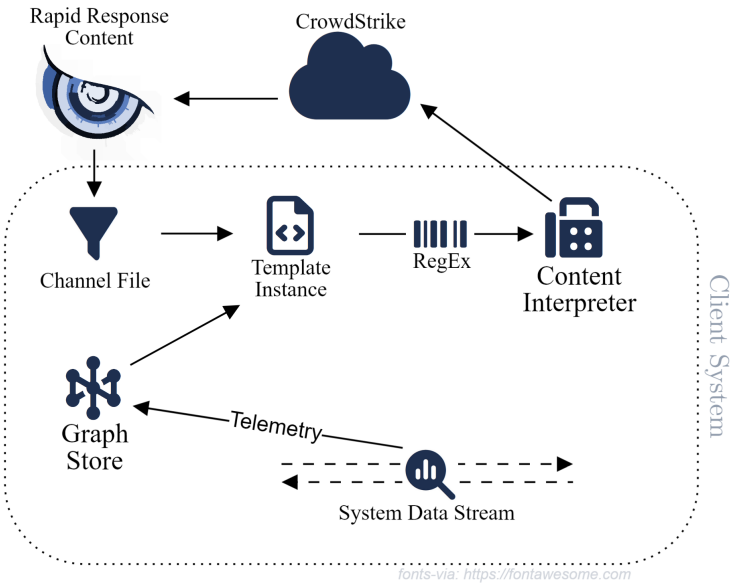

Figure 1.2: RRC & Telemetry Graph Store Data Aggregation for IOA Content Interpreter.

February 2024, CrowdStrike released sensor version 7.11, introducing a new template type. The template type target a new IOA vector relating to **Windows Interprocess Communication (IPC)** mechanisms—notably **Named Pipes**. Named pipes facilitate communication between processes on the same or different machines. The exploit enabled an adversary to execute code and impersonation leading to privilege escalation [22].

This release included a new IPC template type. RRC delivers IPC template instances to **Channel File 291**. This new template type defined 21 input fields, while the Content Interpreter still expected 20. This evaded detection as the Content Interpreter identified files based on a wildcard matching pattern.

Listing 1.1: Wildcard Pattern Matching Example

```
*: Matches any sequence
?: Matches any single character
Input: txt = ``abcdef'', pattern = ``a?c*''
Output: true
Reason: `?' matches with `b' and `*' matches with ``def''.
```

(As CrowdStrike has mentioned their use of RegEx before, it is likely that the Content Interpreter used a RegEx pattern to match fields.)

On July 19, 2024, the RRC push two new IPC template instances to Channel—one of which dropped wildcard matching. This required the Content Interpreter to check the 21st field from Channel File 291. However, the Content Interpreter expected 20 fields, causing an out-of-bounds error, crashing the Falcon Sensor. This error caused the BSoD that Friday afternoon affecting 8.5 million devices.

1.8 Windows Kernel Crash Dump Analysis

David Weston, Vice President, Enterprise and OS Security at Microsoft, posted in an incident the kernel crash dump from Channel File 291 [26]. The Microsoft team's **Windows Error Reporting (WER)** kernel crash dumps analysis involved **WinDBG Kernel Debugger**, and several other freely accessible debugging extensions.

Code Dump Continued on Next Page.

Listing 1.2: WinDBG Kernel Crash Dump

```
1  FAULTING_THREAD:  ffffe402fe868040
2  READ_ADDRESS:  ffff840500000074 Paged pool
3  MM_INTERNAL_CODE:  2
4  IMAGE_NAME:  csagent.sys
5  MODULE_NAME: csagent
6  FAULTING_MODULE: fffff80671430000 csagent
7  PROCESS_NAME:  System
8
9  TRAP_FRAME:  ffff94058305ec20 -- (.trap 0xffff94058305ec20)
10 .trap 0xffff94058305ec20
11 NOTE: The trap frame does not contain all registers.
12 Some register values may be zeroed or incorrect.
13 rax=ffff94058305f200 rbx=0000000000000000 rcx=0000000000000003
14 rdx=ffff94058305f1d0 rsi=0000000000000000 rdi=0000000000000000
15 rip=fffff806715114ed rsp=ffff94058305edb0 rbp=ffff94058305eeb0
16  r8=ffff840500000074  r9=0000000000000000 r10=0000000000000000
17 r11=0000000000000014 r12=0000000000000000 r13=0000000000000000
18 r14=0000000000000000 r15=0000000000000000
19 iopl=0         nv up ei ng nz na po nc
20 csagent+0xe14ed:
21 fffff806`715114ed 458b08 mov r9d,dword ptr [r8] ds:ffff8405`00000074=???...
22 .trap
23 Resetting default scope
24
25 STACK_TEXT:
26 ffff9405`8305e9f8 fffff806`5388c1e4...: nt!KeBugCheckEx
27 ffff9405`8305ea00 fffff806`53662d8c...: nt!MiSystemFault+0x1fcf94
28 ffff9405`8305eb00 fffff806`53827529...: nt!MmAccessFault+0x29c
29 ffff9405`8305ec20 fffff806`715114ed...: nt!KiPageFault+0x369
30 ffff9405`8305edb0 fffff806`714e709e...: csagent+0xe14ed
31 ffff9405`8305ef50 fffff806`714e8335...: csagent+0xb709e
32 ffff9405`8305f080 fffff806`717220c7...: csagent+0xb8335
33 ffff9405`8305f1b0 fffff806`7171ec44...: csagent+0x2f20c7
34 ffff9405`8305f430 fffff806`71497a31...: csagent+0x2eec44
35 ffff9405`8305f5f0 fffff806`71496aee...: csagent+0x67a31
36 ffff9405`8305f760 fffff806`7149685b...: csagent+0x66aee
37 ffff9405`8305f7d0 fffff806`715399ea...: csagent+0x6685b
38 ffff9405`8305f850 fffff806`7148efbb...: csagent+0x1099ea
39 ffff9405`8305f980 fffff806`7148edd7...: csagent+0x5efbb
40 ffff9405`8305fac0 fffff806`7152e681...: csagent+0x5edd7
41 ffff9405`8305faf0 fffff806`53707287...: csagent+0xfe681
42 ffff9405`8305fb30 fffff806`5381b8e4...: nt!PspSystemThreadStartup+0x57
43 ffff9405`8305fb80 00000000`00000000...: nt!KiStartSystemThread+0x34
```

[Modified to fit page with '...' usage.]

WER shows a compressed crash dump view, so prior calls leading to the crash aren't shown.

Listing 1.3: Debugging csagent.sys Module (# comments)

```
6: kd> .trap 0xffff94058305ec20 # user debugging command
.trap 0xffff94058305ec20
# Displays the CPU's register states at the time of the crash.

NOTE: The trap frame does not contain all registers.
Some register values may be zeroed or incorrect.
rax=ffff94058305f200 rbx=0000000000000000 rcx=0000000000000003
rdx=ffff94058305f1d0 rsi=0000000000000000 rdi=0000000000000000
rip=fffff806715114ed rsp=ffff94058305edb0 rbp=ffff94058305eeb0
r8=ffff840500000074  r9=0000000000000000 r10=0000000000000000
# `rip`: Faulting instruction pointer address.
# `r8`: Address being accessed; points to `ffff840500000074` (invalid memory).

csagent+0xe14ed:
fffff806`715114ed 458b08 mov r9d,dword ptr [r8]
# Faulting instruction: attempts to read a 32-bit value from address stored in `r8`.

6: kd> !pte ffff840500000074
# Examines the Page Table Entry (PTE) for the address `ffff840500000074.`
# PTEs map virtual addresses to physical memory.

!pte ffff840500000074
VA ffff840500000074
PXE at FFFFABD5EAF57840    PPE at FFFFABD5EAF080A0
PDE at FFFFABD5E1014000    PTE at FFFFABC202800000
contains 0A00000277200863  contains 0000000000000000
pfn 277200     ---DA--KWEV  contains 0000000000000000
not valid
# The PTE indicates that the address is not valid (not mapped in memory).

6: kd> ub fffff806`715114ed
# Unassembles instructions backward from the address `fffff806715114ed`.
# Helps analyze the code that led to the faulting instruction.

csagent+0xe14d9:
fffff806`715114d9 04d8      add     al,0D8h
fffff806`715114db 750b      jne     csagent+0xe14e8 (fffff806715114e8)
fffff806`715114dd 4d85c0    test    r8,r8
# The above checks if `r8` is NULL. The below (je) jumps if `r8` is NULL.
fffff806`715114e0 7412      je      csagent+0xe14f4 (fffff806715114f4)
fffff806`715114e2 450fb708  movzx   r9d,word ptr [r8]
# below is where the (je) jumps to if `r8` is NULL, avoiding the `r8` read above.
fffff806`715114e6 eb08      jmp     csagent+0xe14f0 (fffff806715114f0)
```

```
44
45 6: kd> u fffff806`715114eb
46 # Unassembles instructions forward from the address `fffff806715114eb`.
47 # Provides context for what happens after the faulting instruction.
48
49 csagent+0xe14eb:
50 fffff806`715114eb 7407            je      csagent+0xe14f4 (fffff806715114f4)
51 fffff806`715114ed 458b08          mov     r9d,dword ptr [r8]
52 # Faulting instruction: attempts to read from `r8` (invalid memory).
53 fffff806`715114f0 4d8b5008        mov     r10,qword ptr [r8+8]
54 fffff806`715114f4 4d8bc2          mov     r8,r10
55 fffff806`715114f7 488d4d90        lea     rcx,[rbp-70h]
56 fffff806`715114fb 488bd6          mov     rdx,rsi
57 fffff806`715114fe e8212c0000      call    csagent+0xe4124 (fffff80671514124)
58
59 6: kd> db ffff840500000074
60 # Dumps the raw memory content at the address `ffff840500000074`.
61
62 db ffff840500000074
63 ffff8405`00000074  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??...
64 ffff8405`00000074  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??...
65 ffff8405`00000084  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??...
66 ffff8405`00000094  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??...
67 ffff8405`000000a4  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??...
68 ffff8405`000000b4  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??...
69 ffff8405`000000c4  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??...
70 ffff8405`000000d4  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??...
71 ffff8405`000000e4  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??...
72 # The memory dump shows all `??`, meaning the address is invalid or not mapped.
```

Lines 63-71 show that indeed 'r8' was reading from an invalid memory address, lining up with CrowdStrike's analysis of an out-of-bounds error.

Code Dump Continued on Next Page.

Listing 1.4: Inspecting csagent.sys

```
1  6: kd>!reg querykey \REGISTRY\MACHINE\system\ControlSet001\services\csagent
2
3  Hive           ffff84059ca7b000
4  KeyNode        ffff8405a6f67f9c
5
6  [SubKeyAddr]            [SubKeyName]
7  ffff8405a6f683ac        Instances
8  ffff8405a6f6854c        Sim
9
10 Use '!reg keyinfo ffff84059ca7b000 <SubKeyAddr>' to dump the subkey details
11
12 [ValueType]    [ValueName]         [ValueData]
13 REG_DWORD      Type                2
14 REG_DWORD      Start               1
15 REG_DWORD      ErrorControl        1
16 REG_EXPAND_SZ  ImagePath           C:\Windows\system32\drivers\CrowdStrike\csagent.sys
17 REG_SZ         DisplayName         CrowdStrike Falcon
18 REG_SZ         Group               FSFilter Activity Monitor
19 REG_MULTI_SZ   DependOnService     FltMgr\0
20 REG_SZ         CNFG                Config.sys
21 REG_DWORD      SupportedFeatures   f
```

Upon inspecting csagent.sys, we verify that csagent.sys is a **File System Filter Driver (FLT)** (line 18-19). FLT drivers are often used by anti-virus software to receive file system event notifications. This also verifies CrowdStrike usage of object callbacks to subscribe to kernel events. Additionally the FLT API provides support for IPC monitoring, which aligns with CrowdStrike's new template type.

Code Dump Continued on Next Page.

Listing 1.5: Inspecting csagent.sys

```
1  !ca ffffde8a870a8290
2
3  ControlArea  @ ffffde8a870a8290
4  Segment       ffff...0   Flink       ffff...   Blink          ffffde8a870a7d98
5  Section Ref          0   Pfn Ref            b   Mapped Views 0
6  User Ref             0   WaitForDel         0   Flush Count  0
7  File Object   ffff...0   ModWriteCount      0   System Views 0
8  WritableRefs         0   PartitionId        0
9  Flags (8008080) File WasPurged OnUnusedList
10
11 \Windows\System32\drivers\CrowdStrike\C-00000291-00000000-00000032.sys
12
13 1: kd> !ntfskd.ccb ffff880ce06f6970
14 !ntfskd.ccb ffff880ce06f6970
15
16 Ccb: ffff880c`e06f6970
17 Flags: 00008003 Cleanup OpenAsFile IgnoreCase
18 Flags2: 00000841 OpenComplete AccessAffectsOplocks SegmentObjectReferenced
19     Type: UserFileOpen
20 FileObj: ffffde8a879b29a0
21
22 (018) ffff880c`db937370  FullFileName:
23 [\Windows\System32\drivers\CrowdStrike\C-00000291-00000000-00000032.sys]
24 # The above confirms the file path of the Channel File 291.
25 (020) 000000000000004C  LastFileNameOffset
26 (022) 0000000000000000  EaModificationCount
27 (024) 0000000000000000  NextEaOffset
28 (048) FFFF880CE06F69F8  Lcb
29 (058) 0000000000000002  TypeOfOpen
```

Line 23 (modified to fit page) shows the full file path of Channel File 291, confirming the cause of the crash. The dump shows the crash is due to an out-of-bounds error. While the crash dump is not publicly available, the Microsoft team confirms the out-of-bounds error in the given crash dump. Microsoft confirmed the crash to be Crowdstrike's fault early on into the incident due to there being no public facing crash message, only a BSoD, which led to some to believe Microsoft was at fault.

Despite the fact that this is not Kurtz's first significant security mishap, it would be unfair to place responsibility squarely on his shoulders: each Windows sensor release is certified through extensive testing in Microsoft's HLK and WHQL. The lack of comprehensive end-to-end testing introduced a critical error that was not caught by their CI pipelines. Such an erroneous mistake could have been easily caught in a test or development environments, raising substantial concerns. The 12 page incident report duly noted this, mentioning under point 5 of the Findings and mitigations section that CrowdStrike should "expand [validation] to include testing within the Content Interpreter" [9]. Even though customers managed to recover from the incident, but CrowdStrike continues to face severe legal repercussions.

Bibliography

[1] Rahul Awati. Kernel panic. https://www.techtarget.com/searchdatacenter/definition/kernel-panic. Accessed: November 30, 2024.

[2] Kurt Baker. Rootkit malware. https://www.crowdstrike.com/en-us/cybersecurity-101/malware/rootkits/, November 2023. Accessed: November 30, 2024.

[3] Cisco. What is endpoint security? https://www.cisco.com/c/en/us/products/security/endpoint-security/index.html. Accessed: November 30, 2024.

[4] Wikipedia contributors. Kernel patch protection. https://en.wikipedia.org/wiki/Kernel_Patch_Protection. Accessed: November 30, 2024.

[5] Cosive. Crowdstrike falcon complete. https://www.cosive.com/capabilities/crowdstrike-falcon-complete. Accessed: November 30, 2024.

[6] CrowdStrike. Falcon complete: Next-gen managed detection and response (mdr). https://www.crowdstrike.com/services/endpoint-security/falcon-complete-next-gen-mdr/. Accessed: November 30, 2024.

[7] CrowdStrike. Falcon overwatch. https://www.crowdstrike.com/platform/threat-intelligence/adversary-overwatch. Accessed: November 30, 2024.

[8] CrowdStrike. Channel file 291 incident - remediation and guidance hub. https://www.crowdstrike.com/falcon-content-update-remediation-and-guidance-hub/, August 2024. Page last updated 2024-08-06 2119 UTC.

[9] CrowdStrike. External technical root cause analysis — channel file 291. https://www.crowdstrike.com/wp-content/uploads/2024/08/Channel-File-291-Incident-Root-Cause-Analysis-08.06.2024.pdf, August 2024. Accessed: November 30, 2024.

[10] CrowdStrike. Technical details: Falcon content update for windows hosts, 2024. Executive Viewpoint, published on July 20, 2024.

[11] Red Hat. What is ci/cd? https://www.redhat.com/en/topics/devops/what-is-ci-cd, December 2023. Accessed: November 30, 2024.

[12] IBM. What is endpoint security? https://www.ibm.com/topics/endpoint-security. Accessed: November 30, 2024.

[13] Alex Ionescu, Milos Petrbok, Martin O'Brien, and Johnny Shaw. Tech analysis: Crowdstrike's kernel access and security architecture. *CrowdStrike Blog*, August 2024. Executive Viewpoint, Accessed: November 30, 2024.

[14] Sean Michael Kerner. Crowdstrike outage explained: What caused it and what's next. *TechTarget*, October 2024.

[15] Microsoft. Obregistercallbacks function (wdm.h). https://learn.microsoft.com/en-us/windows-hardware/drivers/ddi/wdm/nf-wdm-obregistercallbacks. Accessed: November 30, 2024.

[16] Microsoft. What are indicators of compromise (iocs)? https://www.microsoft.com/en-us/security/business/security-101/what-are-indicators-of-compromise-ioc. Accessed: November 30, 2024.

[17] Microsoft. Kernel-mode code signing requirements. https://learn.microsoft.com/en-us/windows-hardware/drivers/install/kernel-mode-code-signing-requirements--windows-vista-and-later-, June 2022. 5 contributors, Accessed: November 30, 2024.

[18] Microsoft. Windows hardware compatibility program certification process. https://learn.microsoft.com/en-us/windows-hardware/design/compatibility/whcp-certification-process, March 2022. Accessed: November 30, 2024.

[19] Oracle. What is a graph database? https://www.oracle.com/autonomous-database/what-is-graph-database/, November 2024. Accessed: November 30, 2024.

[20] Proofpoint. What is telemetry? telemetry cybersecurity explained. https://www.proofpoint.com/us/threat-reference/telemetry#:~:text=Security%20telemetry%20involves%20data%20collection,%2C%20vulnerabilities%2C%20or%20potential%20breaches. Accessed: November 30, 2024.

[21] ReasonLabs. What is kernel-level hooking? https://cyberpedia.reasonlabs.com/EN/kernel-level%20hooking.html. Accessed: November 30, 2024.

[22] Carsten Sandker. Offensive windows ipc internals 1: Named pipes. https://csandker.io/2021/01/10/Offensive-Windows-IPC-1-NamedPipes.html, January 2021. Accessed: November 30, 2024.

[23] Mia Sato. Crowdstrike ceo was working for mcafee in 2010 when there was a global tech outage too. *The Verge*, July 2024. Posted at 9:33 AM EDT, 9 Comments, 9 New.

[24] Joe Tidy. Crowdstrike it outage affected 8.5 million windows devices, microsoft says. *BBC News*, July 2024.

[25] Adrian Volenik. Crowdstrike ceo was working for mcafee in 2010 when there was a global tech outage too. *The Verge*, July 2024. 4 min read.

[26] David Weston. Windows security best practices for integrating and managing security tools. https://www.microsoft.com/en-us/security/blog/2024/07/27/windows-security-best-practices-for-integrating-and-managing-security-tools/, July 2024. 16 min read, Accessed: November 30, 2024.