CrowdStrike IT outage affected 8.5 million Windows devices, Microsoft says

20 July 2024

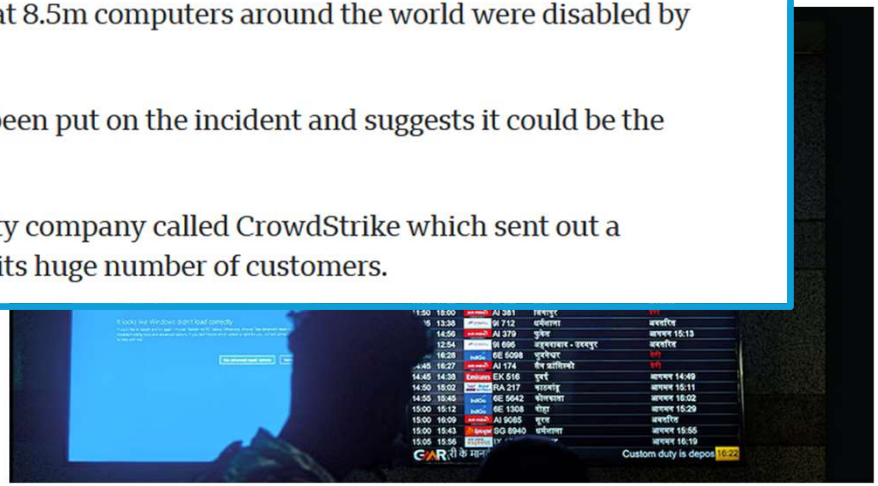
Share Save

Joe Tidy
Cyber correspondent, BBC News

Microsoft says it estimates that 8.5m computers around the world were disabled by the global IT outage.

It's the first time a figure has been put on the incident and suggests it could be the worst cyber event in history.

The glitch came from a security company called CrowdStrike which sent out a corrupted software update to its huge number of customers.



What does CrowdStrike do?

Security systems primer

- **Intrusion detection**: Raises alerts on suspicious activity
  - Network-based: Based on network traffic
    - Malicious packets
    - Exfiltration activity
    - Command & Control (C2) presence
  - Host-based: Based on scanning/monitoring systems
    - Anti-viruses based on scanning files are the most basic kind
- **Intrusion prevention**: Drops malicious network packets
  - Firewalls are a basic kind of network filter

Endpoint Detection & Response

- Developed from anti-viruses
- **Telemetry**: instead of only scanning system files, also includes:
  - Registry changes
  - Process memory, including DLLs
  - Function hooking
  - Connections, ports
  - etc.
- Sends data from endpoints to **centralized processing**
- Provides visualization tools
- Gives alert score; can be configured to automatically respond
- **Live** instead of requiring regular scans

Endpoint Detection & Response

Function Hooking

- Redirecting a library function (usually syscall) to your program so that *all calls to the function go to your program first*
  - Use an event hook API, or just overwrite the library function
- Example:
  - Malware: Reads the process memory of another process to steal a password
  - If we hook the ReadProcessMemory function, we can catch this behavior
- Userland hooking can be **bypassed**… (Hitchins MalwareTech blog)
  - Malware can simply revert what you wrote (but you can scan files to detect it)
  - Malware can perform the syscall or the library function itself (but you can scan process memory to detect it)
  - Malware can call your hook in a clever way to jump over your code

Function Hooking

- Simply overwrite machine instructions with a jump to your code, then re-align the assembly code
- Replicate necessary functionality in your own code

Cat and mouse security game

- Key issue: malware and userland EDRs live on the **same (escalated) privilege**
  - They can overwrite each other's memory, including library calls
- However, EDR is the mouse…
  - EDR must be a *publicly* sold product
  - Malware is written *privately* and often only used once, especially in APTs
  - Malware author can test their code against known EDR, but EDR cannot test their code against future malware
- How can we break the cat and mouse game?

Kernel Mode EDR

- Instead of hooking functions from userland, EDRs can live in the **kernel** directly
  - Much harder to unhook/corrupt EDR functionality
- Write EDR as a kernel driver
  - Windows: since 2022, Early Launch Antimalware (ELAM) drivers can be run as Protected Process Light to ensure code integrity
- Downsides:
  - Application crashes become *system crashes*
  - EDRs also want to be **boot-start**, so a restart causes a permanent crash…
  - *What if the EDR is malware?*

Kernel Mode EDR, but Agile!

- ELAM drivers require a lengthy testing/certification process
  - WHQL release signature required for any **patching**
- **Agile** workaround for patching:
  - Write machine instructions (**p-code**) into configuration files
  - Our EDR is effectively just an interpreter for our configuration file p-code, so it does not need to be patched
  - We can arbitrarily change our EDR's behavior by changing configuration files
- We have circumvented lengthy kernel mode testing, with no potential downsides at all

Disaster

- Analysis according to Dave Plummer:
  - Channel File 0000291
  - Instruction: Read memory pointer in register r8, write to r9
  - Memory pointer was *corrupted*, appears to be caused by a file type mismatch
- Simple workaround: boot into **safe mode** and delete the channel file
  - But safe mode requires physical access, and nowadays a lot of IT support is done remotely…

Why CrowdStrike?

- IDC Reports of market cap and market share:
  - 2020: 756m, 9.2%
  - 2021: 1302m, 12.6%
  - 2022: 1527m, 17.7%
  - 2023: 2279m, 18.1%
    - Overtaken by Microsoft
- CrowdStrike was the first to utilize Microsoft's PPL service to write their product as an ELAM kernel driver

Could this have happened to anyone else?

- This did not happen in macOS
  - Apple Endpoint Security Framework offers API to EDRs, does not allow them to run in kernel
- This *did* happen in Linux
  - kernel panic in RedHat, due to an error in the eBPF
  - eBPF loads programs from user mode to extend kernel capabilities
    - So they don't have to run as kernel modules
  - This requires tight input validation
    - And the input validation had a bug

Kernel Mode EDR why?

- Both Linux and macOS keep third-party kernel drivers at bay - *but why not Windows*?
  - PatchGuard in 2005 prevents patching the kernel, so Microsoft clearly understands that kernel drivers are dangerous
  - Windows does offer similar solutions, e.g. Windows Defender Application Control
- Microsoft software licensing expert Rich Gibbons: 2009 EU anti-competition ruling is to blame
  - *"Microsoft shall ensure that third-party software products can interoperate with Microsoft's Relevant Software Products using the same Interoperability Information on an equal footing as other Microsoft Software Products."*
- Crowdstrike claims scalability issues