Notes over MOVEit Data Breach

New York Times

https://www.nytimes.com/2023/06/15/us/politics/russian-ransomware-cyberattack-clop-moveit.html

- accessed federal agencies (incl energy dept)
- objective to steal and sell back users' data (?)
- described as 'opportunistic'
- **carried out by 'Clop' a Russian ransomware gang**
- exploited a vulnerability in the MOVEit software used by 'an array of local governments, universities, and corporations'
- other victims that came out in June:
    - public officials in Illinois, Nova Scotia, and London
    - British Airways
    - BBC
    - John Hopkins University
    - University System of Georgia
    - Shell (oil)
- anonymous senior CISA official stated about the private sector that 'at least several hundred companies and organizations had been affected'
- GovSpend collected data showing the following government agencies purchased MOVEit software:
    - NASA
    - Treasury Dept
    - Dept of Health and Human Services
    - arms of Defense Dept
- Clop claimed they had "no interest" in data stolen from gov/police and deleted it (LMFAO); only wanted stolen business info
- Progress Software identified the vulnerability **in May and issued a patch withc CISA adding it to the known vulnerabilites online catalog**

Mandiant/Google

https://cloud.google.com/blog/topics/threat-intelligence/zero-day-moveit-data-theft

- **Progress Software Corporation announced vulnerability on May 31, 2023**
- CVE-2023-34362
- Mandiant's intial response engagements shows **earliest evidence on May 27, 2025**
    - earliest evidence shows vulnerability deploying web shells and engaging in data theft
    - "in some instances, data theft has occurred within minutes of the deployment of web shells"
- **Jun 6, 2023, CLOP^_-LEAKS data leak site post claimed responsibility + threatened to post stolen data if victims did not pay an extortion fee**
- `LEMURLOOT` is tailored to execute on a system using MOVEit
- generate commands to enumerate files and folders
- retrieve configuration information

- create or delete a user with a hard-coded name
- `LEMURLOOT` analysis
    - authenticates incoming connections via hard-coded password
    - run commands that download files from the MT system
    - extract its Azure system settings
    - retriece detailed record information
    - create and insert a particular user, or delete the same user
    - data returned to the system interacting with `LEMURLOOT` is gzip compressed
- How the attack goes:
    - CLOP uses vulnerability to access MOVEit software
    - deploys `LEMURLOOT` web shell and uses file names found in MOVEit Transfer (MT) software such as `human.aspx`
    - sends several POST requestes to the legitimate `guestaccess.aspx` file before interacting with `LEMURLOOT`; indicates SQL injection attacks were directed towards `guestaccess.aspx`
    - `LEMURLOOT` checks incoming HTTP requests for a header field containing `X-siLock-Comment` and a corresponding 36-character GUID-formatted value
        - the GUID-formatted value will the be the password between the attacker and the web shell
        - this hides the malware from tools, scanners, and users that are not the attackers
    - with the correct header contents, `LEMURLOOT` responds with `X-siLock-Comment` and value comment to indicate success
        - at this point, the backdoor is active and `LEMURLOOT` is accepting commands and won't generate errors or logs
    - malware then reads the MT server's internal config (incl DB username/pass)
    - malware logs into MT database
    - malware then processes data from attacker via HTTP header files like before: `X-siLock-Step1`, `X-siLock-Step2`, `X-siLock-Step3`
    - `X-siLock-Step1 = -1`
        - `LEMURLOOT` retrieves and returns AZURE system settings which allows it to then perform SQL queries to retrieve files, file size, folders, file owners, and institution name; this is exported to the attacker in a gzip compressed folder or file (?)
    - `X-siLock-Step1 = -2`
        - LEMUTLOOT deletes user account with LoginName and RealName set to "Health Check Service"
            - `Delete FROM users WHERE RealName='Health Check Service'`
    - `X-siLock-Step1 != -1 || -2`
        - malware parses values from the header fields `X-siLock-Step2` and `X-siLock-Step3` to store in variables named *fileid* and *folderid*
            - if `fileid` and `folderid` != NULL, malware retrieves file within the fields, gzip compresses it, and returns it to attacker
            - if `fileid` and `folderid` = NULL, `LEMURLOOT` attempts to find an existing account with permission level "30" and `InstID` = [value set from `X-siLock-Step1`]
                - if unsuccessful it creates a new account with a randomly generated username and with LoginName and RealName as 'Health Check Service'; account is then inserted into an active MT application session

Huntress by John Hammond ❤️

https://www.huntress.com/blog/moveit-transfer-critical-vulnerability-rapid-response

- Progress brought down MOVEit Cloud after exploitation attempts were discovered/detected (possibly June 1st of 2nd)
- initial phase is through SQL injection which leads to arbitrary code execution
  - leads to intant deploy of ransomware under MOVEit service account 'moveitsvc' which is a local administrators group and they could then disable antivirus protections
- **Cl0p is also called Lace Tempest by Microsoft**
- *omg this is sooo much*

BBC

https://www.bbc.com/news/technology-65814104

- Victims:
  - BBC
    - stolen data incl staff ID numbers, DOBs, addresses, national insurance numbers (?)
  - British Airways
    - some may have bank details stolen
  - Boots
  - Aer Lingus
  - Zellis, payroll services provider
    - data from eight of its client firms had been stolen
  - mostly people in the US

CybersecurityDive

https://www.cybersecuritydive.com/news/moveit-breach-timeline/687417/

- 2,650+ organization impacted
- five additional vulnerabilities were discovered after
- victims:
  - National Student Clearinghouse
  - PBI Research Services
  - TIAA
  - Zellis
- **timeline**

HackTheBox (for understanding)

https://www.hackthebox.com/blog/cve-2023-34362-explained

- Cl0p exploited the CVE 36934

- affected approx 130 victims over 10 days

- the web shell, `LEMURLOOT`, allowed attackers to:

  - enumerate (scan) underlying SQL databse
  - store and retrieve files from MT system

    - create a new administrator privileged acount

- vulnerability is caused by `UserGetUsersWithEmailAddress()` not being cleaned
    - within UserEngine (`UserEngine.UserGetUsersWithEmailAddress()`) defined in `MOVEit.DMZ.Class.Lib`
- The `SILHttpSessionWrapper.SetAllSessionVarsFromHeaders()` function (completely removed in patched MOVEit Transfer versions) allows the caller to set arbitrary session variables from HTTP request headers starting with `X-siLock-SessVar`.
    - called by machine2.aspx's SILMachine2
    - incorrectly parses header with `action=m2` parameter in moveitisapi.dll (accessible from otuside) allows to forward arbitrary data to machine2.aspx which bypasses the localhost restriction
- after session variables have been set, `LoadFromSession()` from SILGuestAccess is called by makiung a request to guestaccess.aspx

  To trigger SQL injection, the payload is first put into the `MyPkgSelfProvisionedRecips` environment variable through the `moveitisapi.dll?action=m2` > SILMachine2 (machine2.aspx) > `SetAllSessionVarsFromHeaders()` path, then copied to this .`SelfProvisionedRecips` via guestaccess.aspx.

  The `SelfProvisionedRecips` value is then parsed as a comma-separated list of email addresses and passed to `UserGetUsersWithEmailAddress()` unsanitized, *to be then inserted into the constructed SQL query as the AND Email='...' value, resulting in the execution of arbitrary queries.*

CVE Details

https://www.cvedetails.com/cve/CVE-2023-36934/

CWE ids for CVE-2023-36934

```
CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL
Injection')
The product constructs all or part of an SQL command using externally-influenced
input from an upstream component, but it does not neutralize or incorrectly
neutralizes special elements that could modify the intended SQL command when it is
sent to a downstream component. Without sufficient removal or quoting of SQL
syntax in user-controllable inputs, the generated SQL query can cause those inputs
to be interpreted as SQL instead of ordinary user data.
Assigned by: nvd@nist.gov (Primary)
```

CWE (Common Weakness Enumeration)

https://cwe.mitre.org/data/definitions/89.html

```
The product constructs all or part of an SQL command using externally-influenced
input from an upstream component, but it does not neutralize or incorrectly
neutralizes special elements that could modify the intended SQL command when it is
sent to a downstream component. Without sufficient removal or quoting of SQL
syntax in user-controllable inputs, the generated SQL query can cause those inputs
to be interpreted as SQL instead of ordinary user data.
```