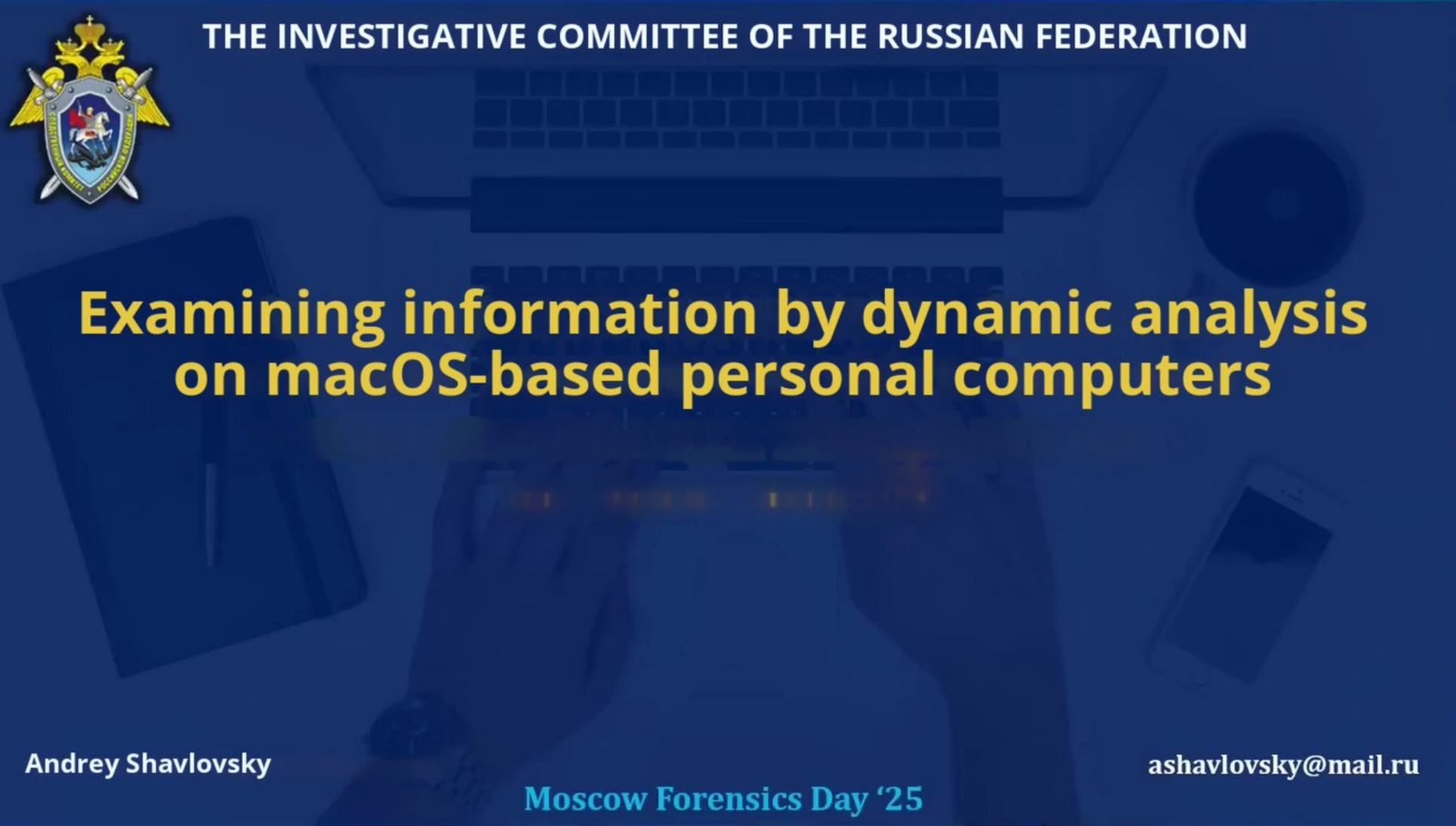
MOSCOW FORENSICS DAY '25



ANDREY SHAVLOVSKY

"EXAMINING INFORMATION BY DYNAMIC ANALYSIS ON MACOS-BASED PERSONAL COMPUTERS"

MFD



THE INVESTIGATIVE COMMITTEE OF THE RUSSIAN FEDERATION

Examining information by dynamic analysis on macOS-based personal computers

Andrey Shavlovsky

ashavlovsky@mail.ru

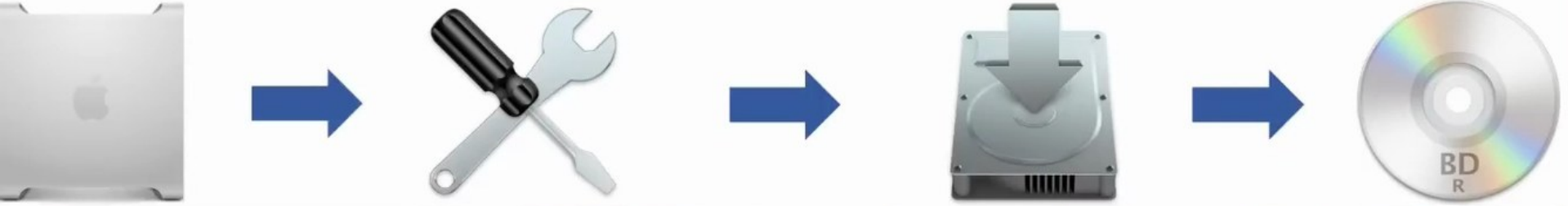
Moscow Forensics Day '25

General approaches in digital forensics

Inspection and photographic recording of the object's appearance

- Preserving the integrity and immutability of the object's information content
- Diagnostics of the object's condition
- Creating an exact copy of the object's information content
- Data analysis (file system, file structure, OS service files, etc.)
- Searching for cryptographically protected files, recovering deleted information
- Establishing details of information objects
- Data interpretation

Saving the results obtained





Static / dynamic analysis

Static Analysis –

examination of the information of a “non-running” computer system / electronic storage medium. The properties of the object (a file, code, the whole computer system) are examined without running it.

Dynamic Analysis –

examination of a running object (a file, code, the whole computer system); its properties are examined while it is being run.

Static / dynamic analysis

	Static analysis	Dynamic analysis
+	1. Data integrity and immutability are preserved 2. Results are reproducible and repeatable	1. Access to data in protected areas of the information content is possible (in some cases) 2. Analysis of running processes and network connections
−	1. Data from protected areas of the information content may be missed 2. No access to processes	1. Problem of data integrity and immutability 2. Problem of reproducibility and repeatability



Data storage organization in macOS

macOS uses the APFS (Apple File System) file system to store data, introduced in macOS High Sierra (10.13) and current as of 2025. AES-256 is used to encrypt the content.

Keychain – a store of credentials and passwords. Access is only possible after authentication with the account password, Touch ID or Face ID.

Data storage organization in macOS

/Users/<username>/ – the user's "home" directory holding user data. Includes the subdirectories "Documents", "Downloads", "Desktop" and others. Created when an account is registered.

/Users/<name>/Library/ – user data: application history, plist files, SQLite databases.

Directory	Purpose
/System/Library/	Operating system files
/Library/	Application settings, keychains
/Users/	User directories
/private/var/	Temporary files, logs, cache
/etc/ /private/etc/	Configuration files
/Volumes/	Mounted media (volumes)
/Applications/	Installed software

About PLIST files

Property List (plist) — a data representation format in macOS, iPadOS and iOS. Used to store application settings, system configurations, user accounts, application activity logs and more.

Main plist file formats. Plist files support several data representation formats:

XML – a text format based on XML. Used for editing and debugging.

Binary – a machine-oriented format. Unreadable without conversion.

Most system plists (for example, in /Library/Preferences/ or /var/db/dslocal/) are stored exactly this way.

About PLIST files



Fragment of the contents of a binary-format plist file



```
                <key>full name</key>
                <string>nikman198903@icloud.com</string>
                <key>name</key>
                <string>com.apple.idms.appleid.prd.000401-08-611b2d2c-4e38-45c9-a498-ebda7de04e49</
                <key>principal</key>
                <string>com.apple.idms.appleid.prd.000401-08-611b2d2c-4e38-45c9-a498-ebda7de04e49</
                <key>subject dn</key>
                <string>CN=com.apple.idms.appleid.prd.000401-08-611b2d2c-4e38-45c9-a498-ebda7de04e49
                <key>timestamp</key>
                <date>2021-12-06T06:22:23Z</date>
            </dict>
        </array>
    </dict>
</dict>
</plist>
</string>
    </array>
    <key>ShadowHashData</key>
    <array>
        <data>
        YnBsaXN0MDDSAQIDCl8QNlNSUC1SRkM1MDU0LTQwOTYtU0hBNTEyLVBCSORG
        Ml8QFFNBTFRFRC1TSE1MTItUEJLREYy0wQFBgcICVh2ZXJpZmllclRzYWx0
        Wml0ZXJhdGlvbnNPEQIAuNCn0qCScFgI9lkVmeLr0t7/aZKCWs2LQrztDYDE
        GlV8dUOP9LvKqyTg6P+137WnbtCEvq910Fh3RppFw6cEtPYAKK/Z0xKZ9cHL
        LkVy7RWXa4wDDR0hk2RXbaukj63gy0Buis0PTJDXPUNbmPwR3wWV9eUFFJVT
        LdeIlCdjOmBTDYsmCQc1G0NJ9UtlrWhsvaGE1cWGn5UMU+nglDRA+HRp2rEW
        iQLQWWpiyy2xzUko7wUEtM2Us0qlmvVm0irSS7KdAMhFvTtThspnIjFo4MLH
        xRypCzsQ4glGS/23Uv4Ow3k89H0t5Y+c4KAiLip6ku6ShqST+3NM+UJYIjYx
        DyClWFO/aWVV+nBp3hGB5jY+wFElnPj+gZOMazQKE34r8D7afN6V9S5e4fc/
        l8uUHEJg08KXxCXHoga38m5C3ksOs3dOfKSCfsF3koy1k5iaojaXmY7ysYla
        oSQ4wnv+LmCJDWF3cyDB/ZhFnkP7Eb8nzokQKvtzLBFSolWOfkGxu5O4/NUf
        sTQjvgozuKCvQYCds3EQgPPXsaIXacscUE7sYZFwoq73IYis2NhrdyxR4GhN
        EXuxTHGH4UlWr9EuPHS+hyQ4WWF1nbV3nxEq1hZ7O/sjGYcCDx3gfBvm1sry
        dwxNS8c3rmCegNBJGkBDB0vxi5ttEfb39Z6JLU9ivXpPECDLXYNPkja4JEtO
        cpW6N9uPdcXQSamYjntPM6PYFIMAyhH0JN8LBQYMDQ5XZW50cm9weU8QgN4N
        fsj0ywRDZBzKiv+iWEvwVKdYr00rK/OlV53xmZJgyStlT+8+CDQfF0qXq7wg
        OUJNVThS196fpxIbUQguQTbiYwnUYP4EDzEYE9xlFhw4iuWjiCpIfm/YM0Cf
        UDdfyTImFb6eRnGiB0fjVG9QJwn3DoxDe0HNzEWjIt5DSM9DTxAgGqnSYdN3
        kxjGCh9+lUiYARFxLPf/Tu1T37xhvNLeKlwR8p8ACAANAC4ARQBMAFUAWgBl
        AmkCjAKPApYCngMhA0QAAAAAAAACAQAAAAAAAAAPAAAAAAAAAAAAAAAAAAAD
        Rw==
        </data>
    </array>
```

Fragment of the contents of an xml-format plist file

User accounts in macOS

Information about registered user accounts (local users) is held in plist files: /var/db/dslocal/nodes/Default/users/<username>.plist. These files contain various details of the account, including a section with “ShadowHashData” data.



convert

```
</string>
    </array>
    <key>ShadowHashData</key>
    <array>
        <data>
        YnBsaXN0MDDSAQIDCl8QHlNSUC1SRkM1MDU0LTQwOTYtU0hBNTEyLVBCS0RG
        Ml8QFFNBTFRFRC1TSEE1MTItUEJLREYy0wQFBgcICVh2ZXJpZmllclRzYWx0
        Wml0ZXJhdGlvbnNPEQIAuNCn0qCScFgI91kVmeLr0t7/aZKCWs2LQrztDYDE
        GlV8dUOP9LvKqyTg6P+137WnbtCEvq9l0Fh3RppFw6cEtPYAKK/Z0xKZ9cHL
        LkVy7RWXa4wDDR0hk2RXbaukj63gy0Buis0PTJDXPUNbmPwR3wWV9eUFFJVT
        LdeI1CdjOmBTDYsmCQc1G0NJ9Ut1rWhsvaGE1cWGn5UMU+nglDRA+HRp2rEW
        iQLQWWpiyy2xzUko7wUEtM2Us0qlmvVm0irSS7KdAMhFvTtThspnIjPo4MLH
        xRypCzsQ4glGS/23Uv4Ow3k89H0t5Y+c4KAiLip6ku6ShqST+3NM+UJYIjYx
        DyClWFO/aHVV+nBp3hGB5jY+wFE1mPj+gZOMazQKE34r8D7afN6V9S5e4fc/
        l8uUHEJg08KXxCXHoga38m5C3ksOs3dOfKSCfsF3koy1k5iaojaXmY7ysY1a
        oSQ4wnv+LmCJDWF3cyDB/ZhFnkP7Eb8nzokQKvtzLBFSolWOfkGxu5O4/NUf
        sTQjvgozuKCvQYCds3EQgPPXsaIXacscUE7sYZPwoq73IYis2NhrdyxR4GhN
        EXuxTHGH4UlWr9EuPHS+hyQ4WWF1nbV3nxEq1hZ7O/sjGYcCDx3gfBvm1sry
        dwxNS8c3rmCegNBJGkBDB0vxi5ttEfb39Z6JLU9ivXpPECDLXYNPkja4JEtO
        cpW6N9uPdcXQSamYjntPM6PYFIMAyhH0JNMLBQYMDQ5XZW50cm9weU8QgN4N
        fsj0ywRDZBzKiv+iWEvwVKdYr00rK/OlV53xmZJgyStlT+8+CDQfF0qXq7wg
        OUJNVThS196fpxIbUQguQTbiYwnUYP4EDzEYE9xlFhw4iuWjiCpIfm/YM0Cf
        UDdfyTImFb6eRnGiB0fjVG9QJwn3DoxDe0HNzEWjItSDSM9DTxAgGqnSYdN3
        kxjGCh9+lUiYARFxLPf/Tu1T37xhvNLeKlwR8p8ACAANAC4ARQBMAFUAWgBl
        AmkCjAKPApYCngMhA0QAAAAAAAACAQAAAAAAAAAPAAAAAAAAAAAAAAAAAAAD
        Rw==
        </data>
    </array>
```

Extracting the password hash

1. Gaining access to the file:

Location: /var/db/dslocal/nodes/Default/users/<username>.plist

2. Decoding its contents from binary into xml format:

[bash] > plutil -convert xml1 <username>.plist -o user.xml or the use of specialized software products / scripts



convert



Extracting the password hash

1. Gaining access to the file:

Location: /var/db/dslocal/nodes/Default/users/<username>.plist

2. Decoding its contents from binary into xml format:

[bash] > plutil -convert xml1 <username>.plist -o user.xml or the use of specialized software products / scripts

3. Decoding (base64) “ShadowHashData”

[bash] > **echo "*ShadowHashData*" | base64 --decode**

```
</string>
    </array>
    <key>ShadowHashData</key>
    <array>
        <data>
        YnBsaXN0MDDSAQIDCl8QHlNSUC1SRkMlMDU0LTQwOTYtU0hBNTEyLVBCS0RG
        Ml8QFFNBTFRFRC1TSEE1MTItUEJLREYy0wQFBgcICVh2ZXJpZmllclRzYWx0
        Wml0ZXJhdGlvbnNPEQIAuNCn0qCScFgI91kVmeLr0t7/aZKCWs2LQrztDYDE
        GlV8dUOP9LvKqyTg6P+137WnbtCEvq910Fh3RppFw6cEtPYAKK/Z0xKZ9cHL
        LkVy7RWXa4wDDR0hk2RXbaukj63gy0Buis0PTJDXPUNbmFwR3wWV9eUFFJVT
        LdeIlCdjOmBTDYsmCQc1G0NJ9Ut1rWhsvaGE1cWGn5UMJ+nglDRA+HRp2rEW
        iQLQWWpiyy2xzUko7WUEtM2Us0qlmvVm0irSS7KdAMhFvTtThspnIjPo4MLH
        xRypCzsQ4glGS/23Uv4Ow3k89H0t5Y+c4KAiLip6ku6Shq5T+3NM+UJYIjYx
        DyClWFO/aHVV+nBp3hGB5jY+wFElmPj+gZOMazQKE34r8D7afN6V9S5e4fc/
        l8uUHEJg08KXkCXHoga38m5C3ksOs3dOfKSCfsF3koy1k5iaojaXmY7ysY1a
        oSQ4wnv+LmCJDWF3cyDB/ZhFnkP7Eb8nzokQKvtzLBFSolWOfkGxu5O4/NUf
        sTQjvgozuKCvQYCds3EQgFPXsaIXacscUE7sYZPwoq73IYis2NhrdyxR4GhN
        EXuxTHGH4UlWr9EuPHS+hyQ4WWFlnbV3nxEq1hZ7O/sjGYcCDx3gfBvmlszy
        dwxNS8c3rmCegNBJGkBDB0vxi5ttEfb39Z6JLU9ivXpPECDLXYNPkja4JEtO
        cpW6N9uPdcXQ5amYjntPM6PYFIMAyhH0JNMLBQYMDQ5XZN50cm9weU8QgN4N
        fsj0ywRDZBzKiv+iWEvwVKdYr00rK/OlV53xmZJgy5tlT+8+CDQfF0qXq7wg
        OUJNVThS196fpxIbUQguQTbiYwnUYP4EDzEYE9xlFhw4iuWjiCpIfn/YM0Cf
        UDdfyTImFb6eRnGiB0fjVG9QJwn3DoxDe0HNzEWjItSDSM9DTxAgGqnSYdN3
        kxjGCh9+lUiYARFxLPf/TulT37xhvNLeKlwR8p8ACAANAC4ARQBMAFUAWgBl
        AmkCjAKPApYCngMhA0QAAAAAAAACAQAAAAAAAAAPAAAAAAAAAAAAAAAAAAAD
        Rw==
        </data>
    </array>
```

base64 decode

```
{
  "SALTED-SHA512-PBKDF2" => {
    "entropy" => <de0d7ec8 f4cb0443 641cca8a ffa2584b f054a758 af4d2b2b f3a5579d
 f1999260 c92b654f ef3e0834 1f174a97 abbc2039 424d5538 52d7de9f a7121b51 082e413
6 e26309d4 60fe040f 311813dc 65161c38 8ae5a388 2a487e6f d833409f 50375fc9 322615
be 9e4671a2 0747e354 6f502709 f70e8c43 7b41cdcc 45a322d4 8348cf43>
    "iterations" => 62111
    "salt" => <1aa9d261 d3779318 c60a1f7e 95489801 11712cf7 ff4eed53 dfbc61bc d2
de2a5c>
  }
```

Account password hash


```
{
  "SALTED-SHA512-PBKDF2" => {
    "entropy" => <de0d7ec8 f4cb0443 641cca8a ffa2584b f054a758 af4d2b2b f3a5579d
 f1999260 c92b654f ef3e0834 1f174a97 abbc2039 424d5538 52d7de9f a7121b51 082e413
6 e26309d4 60fe040f 311813dc 65161c38 8ae5a388 2a487e6f d833409f 50375fc9 322615
be 9e4671a2 0747e354 6f502709 f70e8c43 7b41cdcc 45a322d4 8348cf43>
    "iterations" => 62111
    "salt" => <1aa9d261 d3779318 c60a1f7e 95489801 11712cf7 ff4eed53 dfbc61bc d2
de2a5c>
  }
```

PBKDF2-SHA512 — a cryptographic algorithm for computing hashes from a password value that uses the PBKDF2 function (Password-Based Key Derivation Function 2) combined with the SHA-512 hash function (described in RFC 2898 and updated in RFC 8018). Some of its features:

“entropy” – the final result of the hash function. Stored in hexadecimal form.

“iterations” – the set number of PBKDF2 iterations (62111 in this case).

“salt” – a unique sequence of bytes added to the password before hashing.

Microsoft Windows & Apple macOS



mac OS

NT hash, no salt or iterations

PBKDF2-SHA512 hash (salt + iterations)

Preparing for the password brute force

ml<iterations>$<salt-hex>$<entropy-hex>

1. $ml – denotes the Mac OS X Lion (10.8) operating system version and above. From this OS version on, passwords are hashed with the PBKDF2-SHA512 algorithm. By this marker hashcat identifies the PBKDF2-SHA512 algorithm in macOS.
2. $<iterations> – the number of iterations.
3. $<salt-hex> – the “salt” in hex form.
4. 4. $<entropy-hex> – the final hash after computation.



ml62111$1aa9d261d3779318c60a1f7e95489801117l2cf7ff4eed53dfbc61bcd2de2a5c$de0d7ec8f4cb0443641cca8affa2584bf054a758af4d2b2bf3a5579df1999260c92b654fef3e08341f174a97abbc2039424d553852d7de9fa7121b51082e4136e26309d460fe040f311813dc65161c388ae5a3882a487e6fd833409f50375fc9322615be9e4671a20747e3546f502709f70e8c437b41cdcc45a322d48348cf43

The resulting hash for password guessing

Extracting the password hash: scripts



```
$ml$62111$1aa9d261d3779318c60a1f7e954898011171
2cf7ff4eed53dfbc61bcd2de2a5c$de0d7ec8f4cb0443641c
ca8affa2584bf054a758af4d2b2bf3a5579df1999260c92b6
54fef3e08341f174a97abbc2039424d553852d7de9fa712
1b51082e4136e26309d460fe040f311813dc65161c388ae
5a3882a487e6fd833409f50375fc9322615be9e4671a207
47e3546f502709f70e8c437b41cdcc45a322d48348cf43
```

Hash for password guessing

```
Windows PowerShell
PS D:\scripts> py .\script.py .\administrator.plist
$ml$62111$1aa9d261d3779318c60a1f7e9548980111712cf7ff4eed53dfbc61bcd2de2a5c$de0d7ec8
f4cb0443641cca8affa2584bf054a758af4d2b2bf3a5579df1999260c92b654fef3e08341f174a97abb
c2039424d553852d7de9fa7121b51082e4136e26309d460fe040f311813dc65161c388ae5a3882a487e
6fd833409f50375fc9322615be9e4671a20747e3546f502709f70e8c437b41cdcc45a322d48348cf43
```

The script's output

```python
#!/usr/bin/env python3
import plistlib
import io
import sys
from pathlib import Path
def extract_shadowhash(plist_path: Path):
    with open(plist_path, "rb") as f:
        plist_data = plistlib.load(f)
    shadow = plist_data.get("ShadowHashData")
    if not shadow:
        raise ValueError("В файле нет ключа 'ShadowHashData'")
    blob = shadow[0]
    inner = plistlib.load(io.BytesIO(blob))
    pbkdf2 = inner.get("SALTED-SHA512-PBKDF2")
    if not pbkdf2:
        raise ValueError("Нет секции 'SALTED-SHA512-PBKDF2'")
    iterations = pbkdf2["iterations"]
    salt_hex = pbkdf2["salt"].hex()
    entropy_hex = pbkdf2["entropy"].hex()
    hashcat_line = f"$ml${iterations}${salt_hex}${entropy_hex}"
    return hashcat_line
def main():
    if len(sys.argv) < 2:
        print(f"Использование: {sys.argv[0]} user.plist [output.txt]")
        sys.exit(1)
    plist_path = Path(sys.argv[1])
    if not plist_path.exists():
        print(f"Файл не найден: {plist_path}")
        sys.exit(1)
    hash_line = extract_shadowhash(plist_path)
    if len(sys.argv) >= 3:
        out_path = Path(sys.argv[2])
        with open(out_path, "w") as f:
            f.write(hash_line + "\n")
        print(f"[+] Хэш сохранён в {out_path}")
    else:
        print(hash_line)
if __name__ == "__main__":
    main()
```

Extracting the password hash: software



ДАННЫЕ АРТЕФАКТА

Имя пользователя	**administrator**
Пароль или токен	**ml62111 $1aa9d261d3779318c60a1f7e9548980111712cf7ff4eed53dfbc61bcd 2de2a5c $de0d7ec8f4cb0443641cca8affa2584bf054a758af4d2b2bf3a5579df1 999260c92b654fef3e08341f174a97abbc2039424d553852d7de9fa712 1b51082e4136**
Тип	Пароли и токены
ИД элемента	2602254
Исходный артефакт	User Accounts - macOS

ml62111$1aa9d261d3779318c60a1f7e9548980111712cf7ff4eed53dfbc61bcd2de2a5c$de0d7ec8f4cb0443641cca8affa2584bf054a758af4d2b2bf3a5579df1999260c92b654fef3e08341f174a97abbc2039424d553852d7de9fa7121b51082e4136e26309d460fe040f311813dc65161c388ae5a3882a487e6fd833409f50375fc9322615be9e4671a20747e3546f502709f70e8c437b41cdcc45a322d48348cf43

1234 **The missing part of the hash**



Password guessing: hashcat

🔎 hashcat -m 7100 -a 0 hash.txt wordlist.txt

Windows PowerShell

```
PS C:\Portable Programs\hashcat-7.1.2\hashcat-7.1.2> .\hashcat.exe -m 7100 -a 0 .\hashmac.txt .\wordlist.txt
hashcat (v7.1.2) starting
```

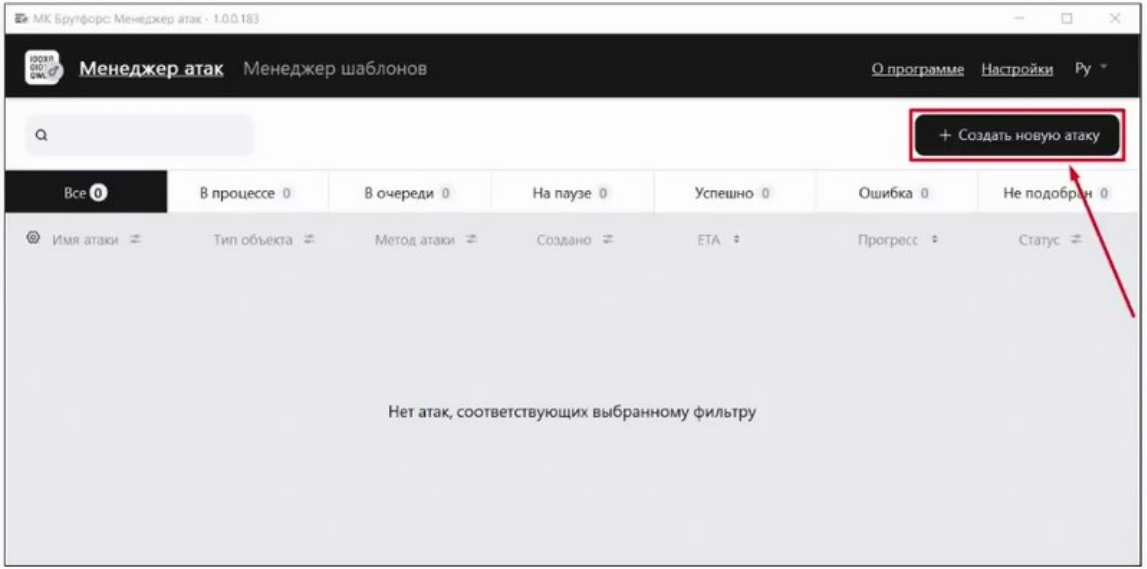
🔄 Running hashcat

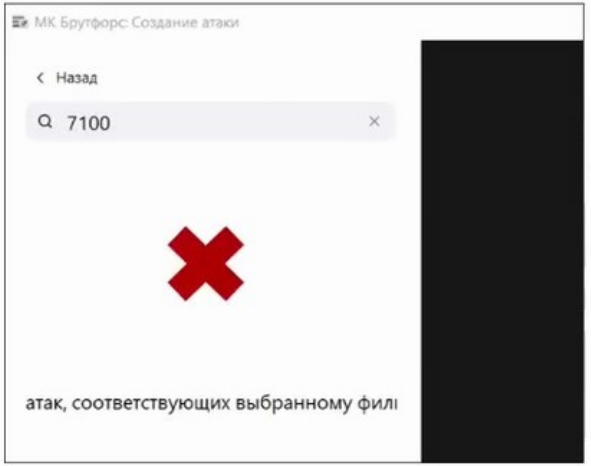
```
$ml$62111$1aa9d261d3779318c60a1f7e9548980111712cf7ff4eed53dfbc61bcd2de2a5c$de0d7ec8f4cb0443641cca8
affa2584bf054a758af4d2b2bf3a5579df1999260c92b654fef3e08341f174a97abbc2039424d553852d7de9fa7121b510
82e4136e26309d460fe040f311813dc65161c388ae5a3882a487e6fd833409f50375fc9322615be9e4671a20747e3546f5
02709f70e8c437b41cdcc45a322d48348cf43:22012

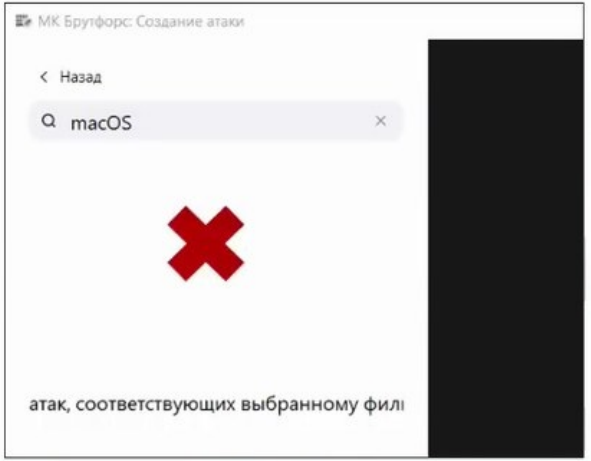
Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 7100 (macOS v10.8+ (PBKDF2-SHA512))
Hash.Target......: $ml$62111$1aa9d261d3779318c60a1f7e9548980111712cf7f...48cf43
```

✅ The result – the password value has been determined

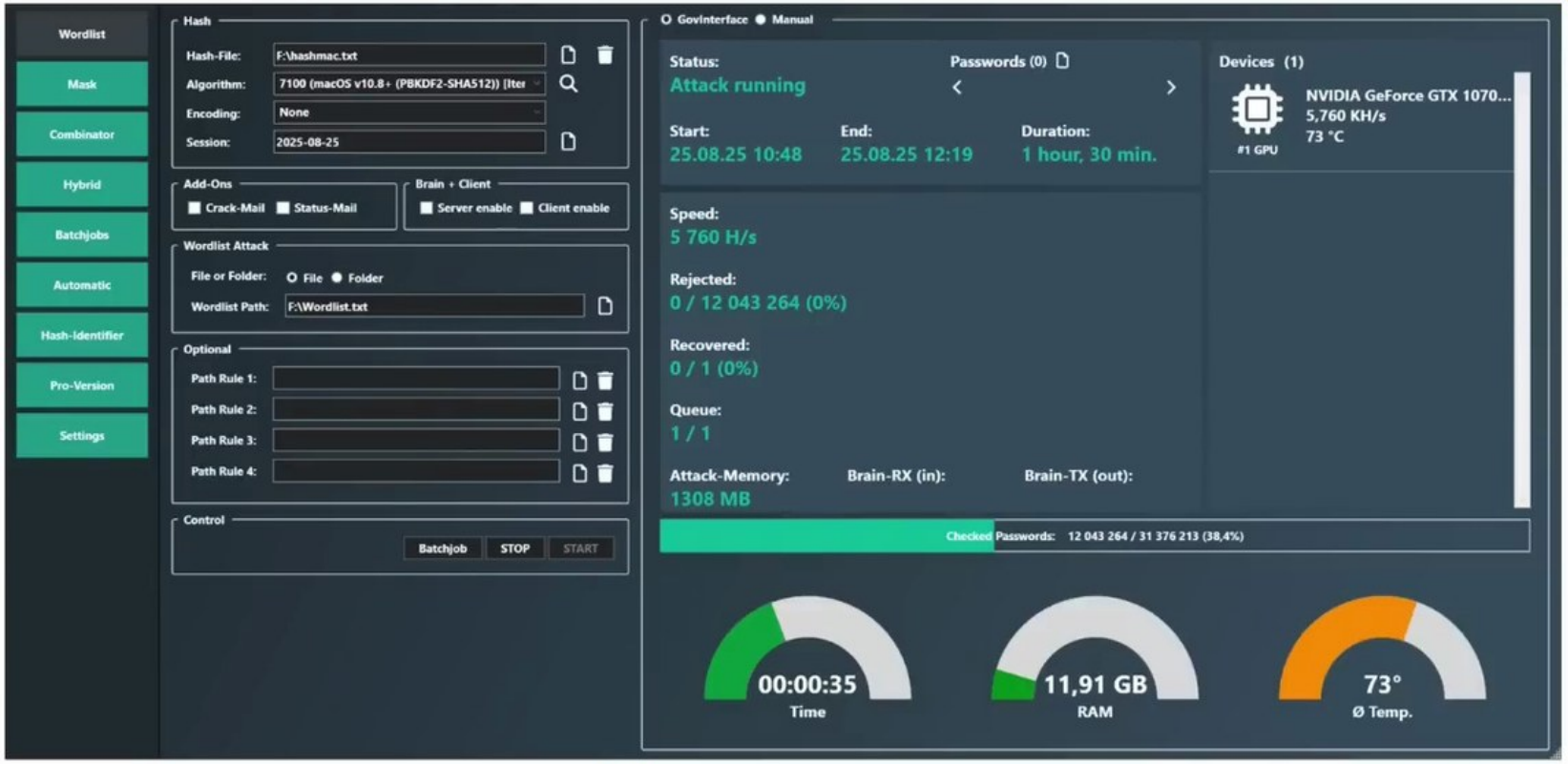
Password guessing: gui built on hashcat



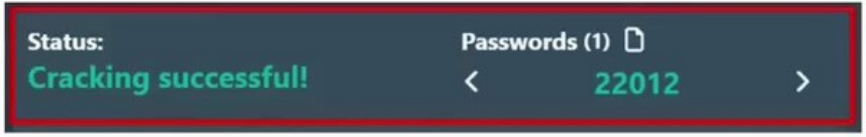




Password guessing: gui built on hashcat



The result – the password value has been determined

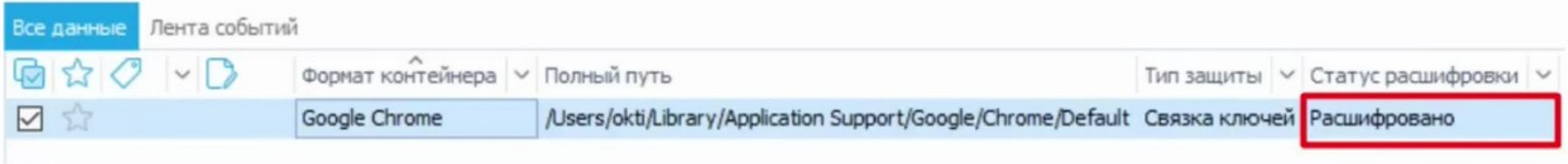


Data analysis and interpretation

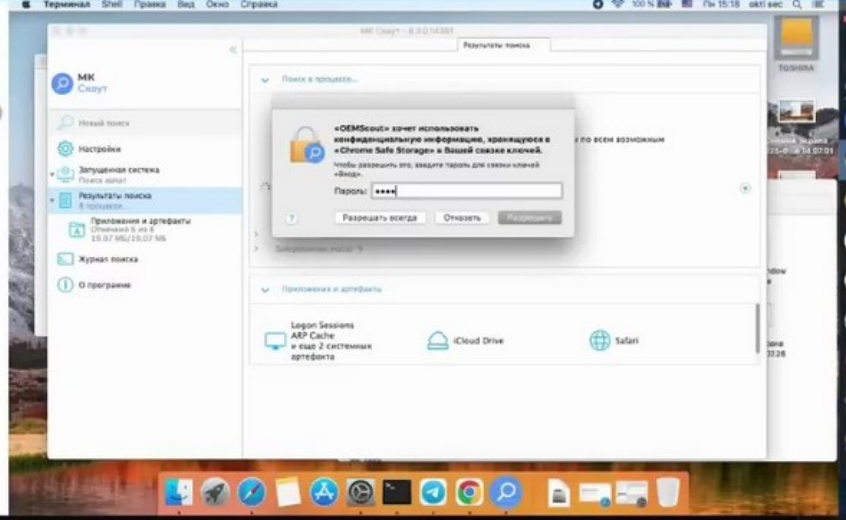
📌 Virtualization of the object and examination of its exact copy.
📌 Examination of the object itself with prior permission to make changes, under Art. 57 pt. 4 cl. 3 of the Russian Code of Criminal Procedure (if this is a forensic examination). ⚖️).



A fragment of the data interpretation without access to the keychain



A fragment of the data interpretation with access to the keychain


МК
Скаут
Настройки
Журнал поиска
О программе
«OEMScout» хочет использовать конфиденциальную информацию, хранящуюся в «Chrome Safe Storage» в вашей связке ключей.
Пароль:
Разрешать всегда
Отказать
Разрешить
Logon Sessions
ARP Cache
и еще 2 системных артефакта
iCloud Drive
Safari

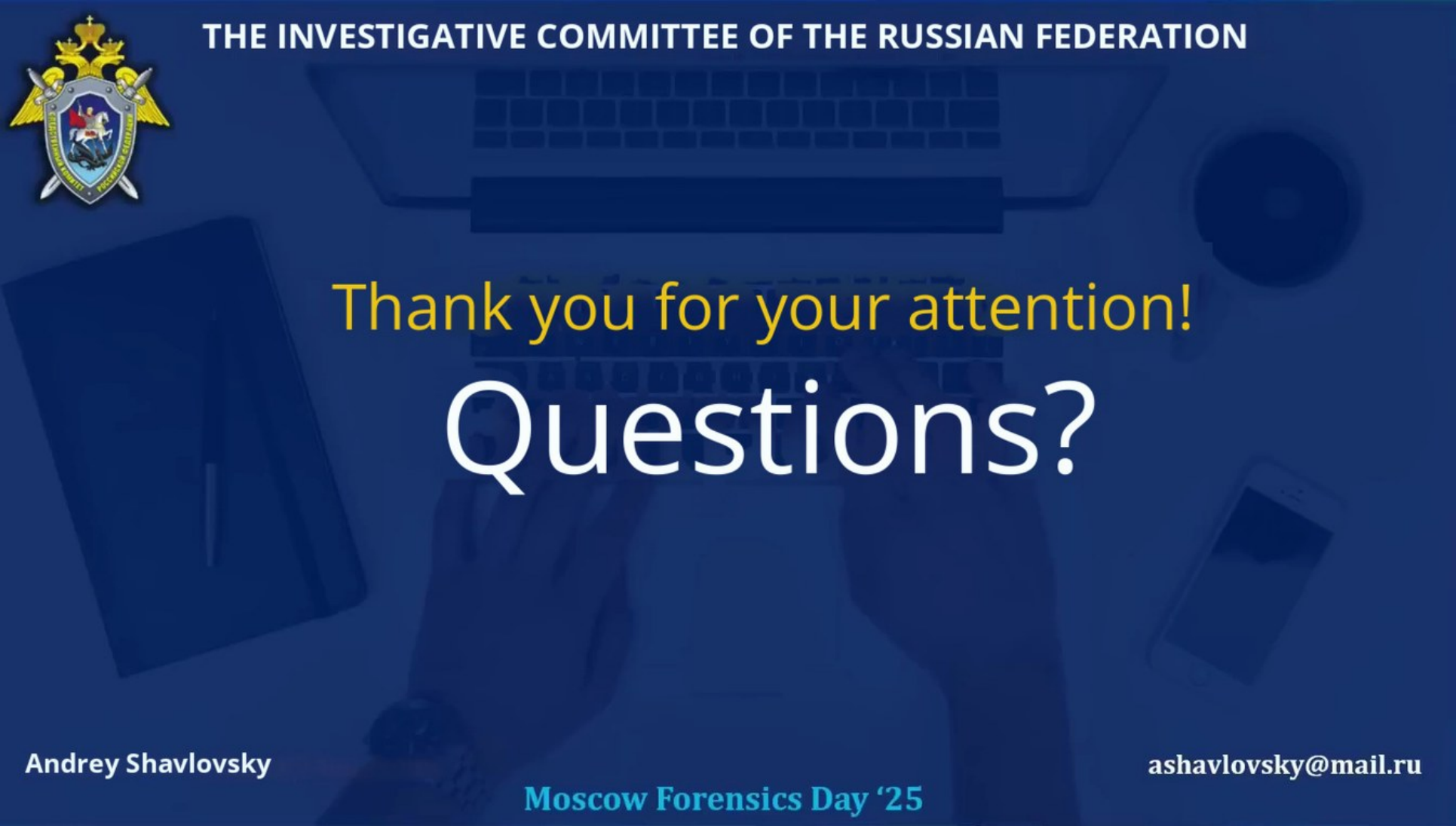

Чаты
Архив





Understanding how traces are formed in the digital environment, to make the results obtained verifiable and reliable

Static and dynamic analysis methods complement each other in the examination, making it complete and comprehensive


THE INVESTIGATIVE COMMITTEE OF THE RUSSIAN FEDERATION
Thank you for your attention!
Questions?
Andrey Shavlovsky
ashavlovsky@mail.ru
Moscow Forensics Day '25