# CATCH ME IF YOU CAN!
# DETECTING SANDBOX EVASION TECHNIQUES

Francis Guibernau & Ayelen Torello

Enigma 2020 – San Francisco, CA, US



ENIGMA

# ABOUT US



**Francis Guibernau**
@OutrageousLynx

Security Researcher at Deloitte



**Ayelen Torello**
@TorelloAyelen

Security Researcher at Deloitte

# OVERVIEW - AGENDA

1 **Environment Awareness**
Definition of the 'Environment Awareness' master technique and the purpose behind it

2 **Identified Techniques**
List of techniques and the different categories defined within Environment Awareness

3 **In the wild Examples**
Multiple malware examples from each identified category.

4 **APT Insight & Conclusion**
How we use what we learned in order to profile and track APT groups.



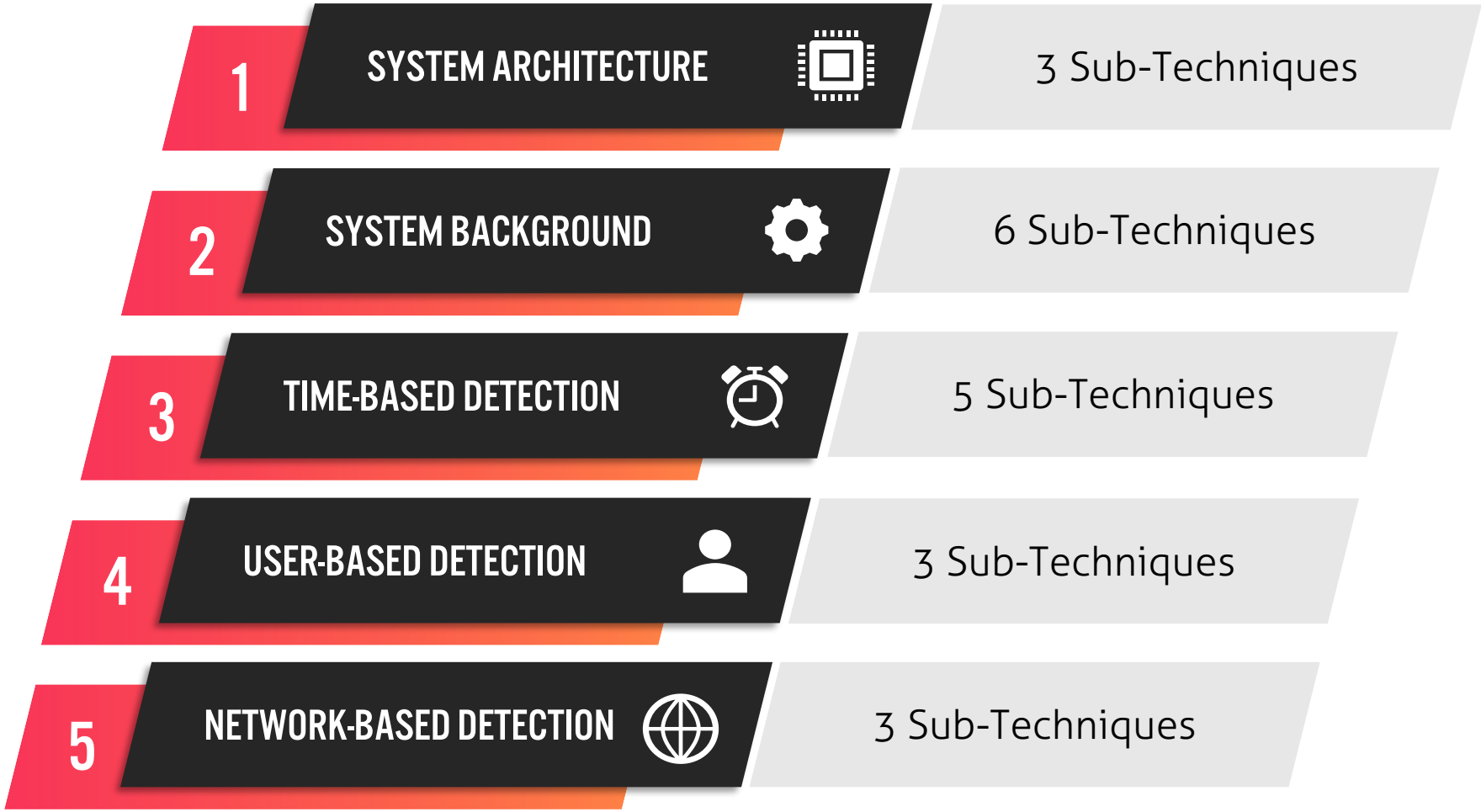
# ENVIRONMENT AWARENESS

**Environment Awareness** is the name provided to the set of high-level techniques used by attackers to attempt to detect the **sandboxing environments**, **virtual machines** or the presence of **forensic tools**.





# TECHNIQUES


1
SYSTEM ARCHITECTURE
3 Sub-Techniques
2
SYSTEM BACKGROUND
6 Sub-Techniques
3
TIME-BASED DETECTION
5 Sub-Techniques
4
USER-BASED DETECTION
3 Sub-Techniques
5
NETWORK-BASED DETECTION
3 Sub-Techniques

# SYSTEM ARCHITECTURE

## SYSTEM SPECIFICATIONS

- CPU Core Count
- System Memory
- Disk Properties

## SYSTEM FOOTPRINT

- BIOS
- UEFI
- EFI

## HARDWARE COMPONENTS

- Thermal Check
- Peripheral Check
- Hardware IDs

# SYSTEM ARCHITECTURE

```
using (ManagementObjectCollection.ManagementObjectEnumerator enumerator = new
    ManagementObjectSearcher("root\\WMI", "select * from MSAcpi_ThermalZoneTemperature").Get().
    GetEnumerator())
{
    if (enumerator.MoveNext())
    {
        float arg_4D_0 = float.Parse(enumerator.Current["CurrentTemperature"].ToString(),
            CultureInfo.InvariantCulture.NumberFormat) / 10f;
        bool result = false;
        return result;
    }
}
```

HARDWARE COMPONENTS

↓

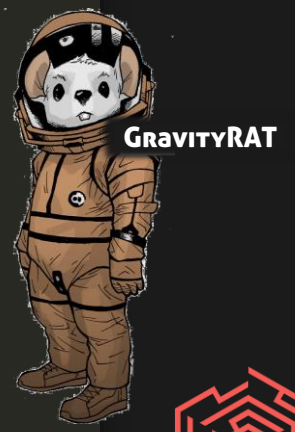
Thermal Check

HARDWARE COMPONENTS

↓

Hardware IDs

```
string queryString = "SELECT ProcessorId FROM Win32_Processor";
try
{
    using (ManagementObjectCollection.ManagementObjectEnumerator enumerator = new
        ManagementObjectSearcher(queryString).Get().GetEnumerator())
    {
        if (enumerator.MoveNext())
        {
            ManagementObject managementObject = (ManagementObject)enumerator.Current;
            bool result;
            if ((string)managementObject["ProcessorId"] == null)
            {
                string host = (string)managementObject["DeviceID"];
                string machineName = (string)managementObject["SystemName"];
                virtualMachine = new VirtualMachine(host, machineName);
                flag = true;
                result = flag;
                return result;
            }
        }
    }
}
```






Reference: https://blog.talosintelligence.com/2018/04/gravityrat-two-year-evolution-of-apt.html

# SYSTEM ARCHITECTURE

**SYSTEM SPECIFICATIONS**

CPU Core Count

```
using (ManagementObjectCollection.ManagementObjectEnumerator enumerator = new
    ManagementObjectSearcher("Select * from Win32_Processor").Get().GetEnumerator())
{
    if (enumerator.MoveNext())
    {
        ManagementBaseObject current = enumerator.Current;
        num += int.Parse(current["NumberOfCores"].ToString());
        bool result;
        if (num == 1)
        {
            string host = current["DeviceID"].ToString();
            string machineName = current["SystemName"].ToString();
            virtualMachine = new VirtualMachine(host, machineName);
            flag = true;
            result = flag;
            return result;
        }
    }
}
```



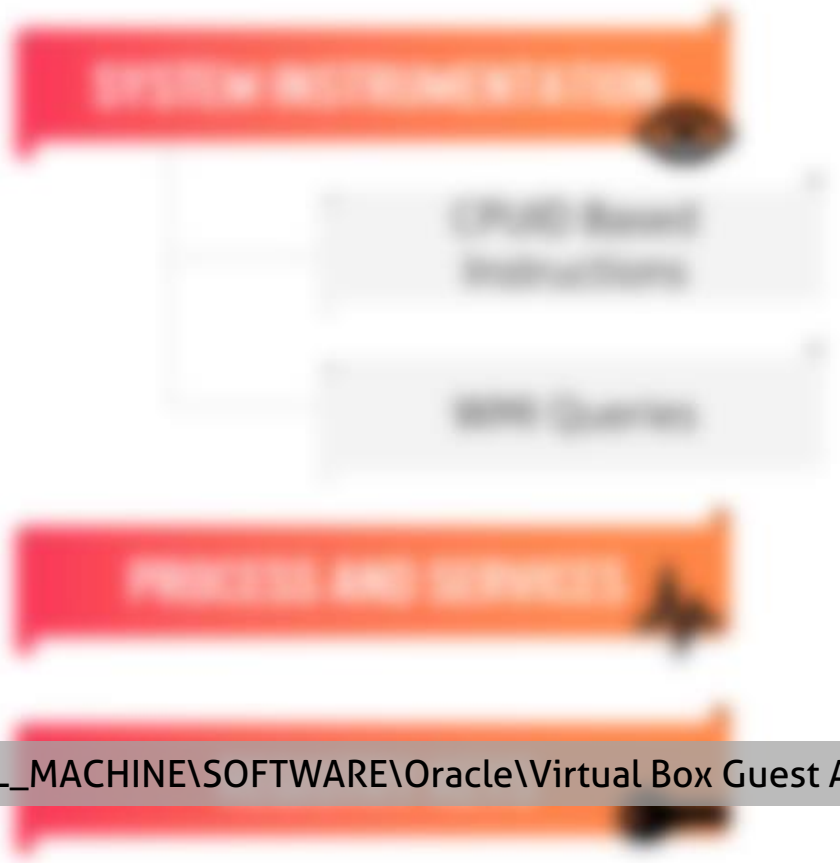



Reference: https://blog.talosintelligence.com/2018/04/gravityrat-two-year-evolution-of-apt.html

# SYSTEM BACKGROUND

- SYSTEM INSTRUMENTATION
  - CPUID Based Instructions
  - WMI Queries
- PROCESS AND SERVICES
- REGISTRY KEYS
- MAC ADDRESSES
- SYSTEM FINGERPRINTING
- ARTIFACTS PRESENCE

# SYSTEM BACKGROUND




00:05:69:xx:xx:xx VMWare
00:0C:29:xx:xx:xx VMWare
00:1C:14:xx:xx:xx VMWare
00:50:56:xx:xx:xx VMWare
00:15:5D:xx:xx:xx Hyper V
00:16:3E:xx:xx:xx Xen
54:52:00:xx:xx:xx KVM

HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\Virtual Box Guest Additions

C:\windows\System32\Drivers\VBoxMouse.sys
C:\windows\System32\Drivers\VBoxGuest.sys
C:\windows\System32\Drivers\VBoxSF.sys
C:\windows\System32\Drivers\VBoxVideo.sys

# SYSTEM BACKGROUND

```
using (ManagementObjectSearcher managementObjectSearcher = new ManagementObjectSearcher("select *
    from Win32_ComputerSystem"))
{
    using (ManagementObjectCollection managementObjectCollection = managementObjectSearcher.Get())
    {
        foreach (ManagementBaseObject current in managementObjectCollection)
        {
            string text = current["Manufacturer"].ToString().ToLower();
            if ((text == "microsoft corporation" && current["Model"].ToString().ToUpperInvariant().
                Contains("VIRTUAL")) || text.Contains("vmware") || current["Model"].ToString() == "
                VirtualBox")
            {
                virtualMachine.Host = current["Model"].ToString();
                virtualMachine.MachineName = current["Manufacturer"].ToString();
                result = true;
                return result;
            }
        }
    }
}
```

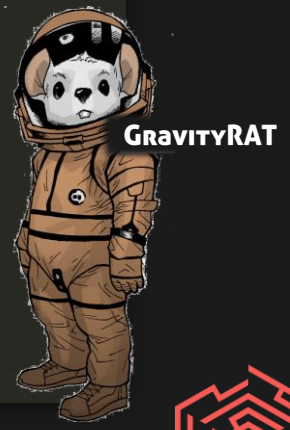
**SYSTEM INSTRUMENTATION**

↓

WMI Queries

```
ManagementObjectCollection.ManagementObjectEnumerator enumerator = new ManagementObjectSearcher("
    select * from Win32_BIOS").Get().GetEnumerator();
if (!enumerator.MoveNext())
{
    throw new Exception("Unexpected WMI query failure");
}

string text = enumerator.Current["version"].ToString();
enumerator.Current["SerialNumber"].ToString();
string[] array = new string[]
{
    "VMware",
    "Virtual",
    "XEN",
    "Xen",
    "A M I"
};
```



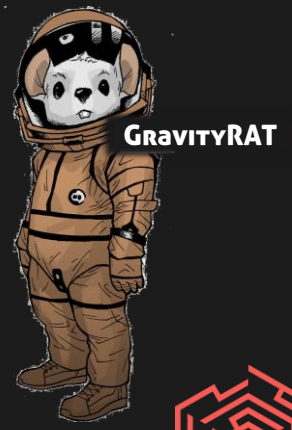



Reference: https://blog.talosintelligence.com/2018/04/gravityrat-two-year-evolution-of-apt.html

# SYSTEM BACKGROUND

MAC ADDRESSES

```
string mac = Identification.Instance.Macid;
if (new string[]
{
    "00:50:56 / VMware, Inc.",
    "00:0C:29 / VMware, Inc.",
    "00:05:69 / VMware, Inc.",
    "08:00:27 / PCS Systemtechnik GmbH (VirtualBox)",
    "00:1C:42 / Parallels, Inc.",
    "00:16:3E / Xensource, Inc."
}.FirstOrDefault((string c) => c.Contains(mac)) != null)
{
    string host = Environment.UserName.ToString();
    string machineName = Environment.MachineName.ToString();
    virtualMachine = new VirtualMachine(host, machineName);
    return true;
}
```



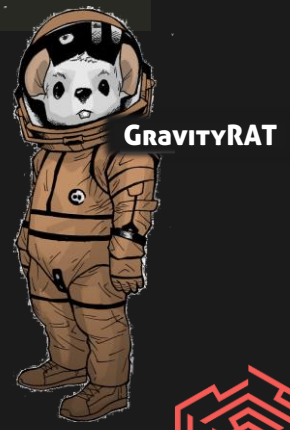



Reference: https://blog.talosintelligence.com/2018/04/gravityrat-two-year-evolution-of-apt.html

# SYSTEM BACKGROUND

```
RegistryKey registryKey = Registry.LocalMachine.OpenSubKey("SOFTWARE\\Microsoft\\Virtual Machine\\
    Guest\\Parameters");
if (registryKey != null)
{
    virtualMachine = new VirtualMachine(registryKey.GetValue("HostName").ToString(), registryKey.
        GetValue("VirtualMachineName").ToString());
    result = true;
}
```

REGISTRY KEYS






Reference: https://blog.talosintelligence.com/2018/04/gravityrat-two-year-evolution-of-apt.html

# TIME-BASED DETECTION

EXTENDED SLEEP



SCHEDULED DOWNLOAD



TIME BOMB



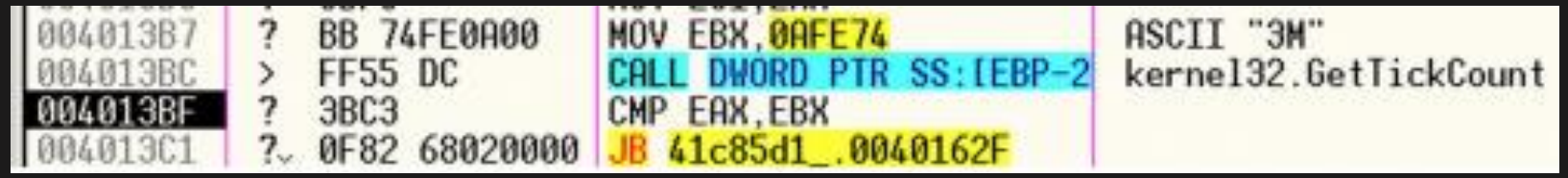
SYSTEM UPTIME



SYSTEM EVENTS



# TIME-BASED DETECTION




SYSTEM UPTIME

UPATRE



Reference: https://unit42.paloaltonetworks.com/ticked-off-upatre-malwares-simple-anti-analysis-trick-to-defeat-sandboxes/

# USER-BASED DETECTION

USER INTERACTION



SOFTWARE AND APPLICATIONS

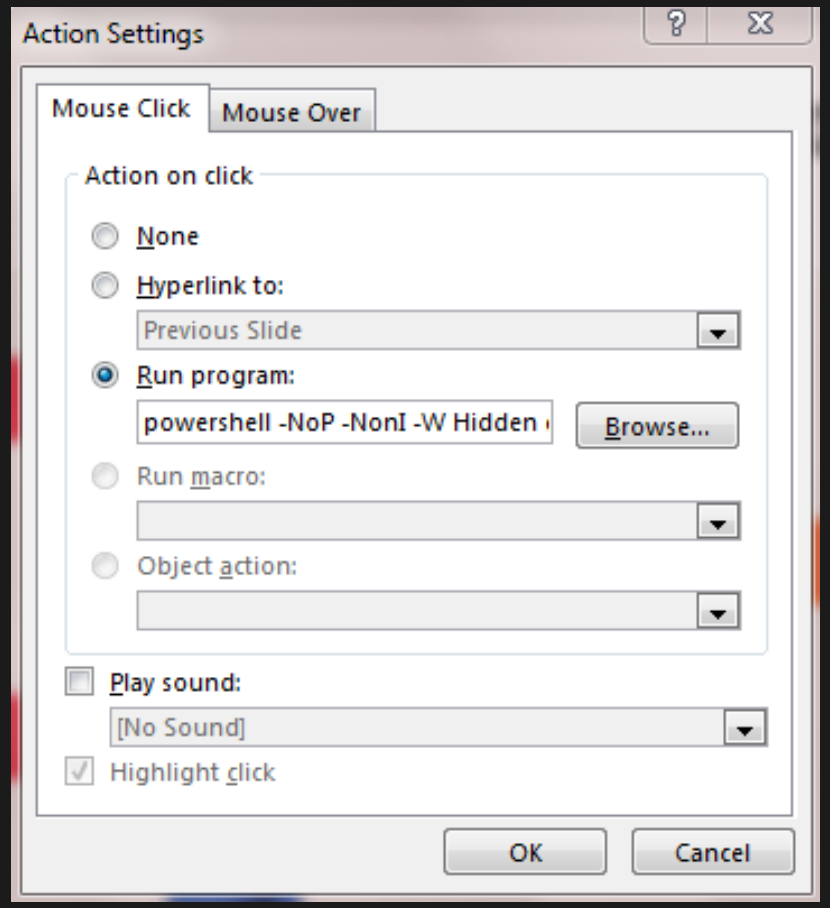




USER PROPERTIES AND CONFIGURATIONS



# USER-BASED DETECTION




USER INTERACTION

# USER-BASED DETECTION




USER INTERACTION

FIN7

# NETWORK-BASED DETECTION

CONNECTIVITY AVAILABILITY



OPEN PORTS



NETWORK CHECK







# NETWORK-BASED DETECTION

```
(Empire: NNHBPD2DFLPURGF4) > usemodule situational_awareness/network/userhunter
(Empire: situational_awareness/network/userhunter) > execute
(Empire: situational_awareness/network/userhunter) >
Job started: Debug32_4hspx

TargetUser      Computer                      IP              SessionFrom     LocalAdmin
----------      --------                      --              -----------     ----------
Administrator   WINDOWS3.dev.testlab.local    192.168.52.205
Administrator   SECONDARY.dev.testlab.local   192.168.52.105  192.168.52.206
Administrator   SECONDARY dev.testlab.local   192.168.52.105  192.168.52.205
Administrator   WINDOWS4.dev.testlab.local    102.168.52.206



Invoke-UserHunter completed
```

NETWORK CHECK

POWERSHELL EMPIRE



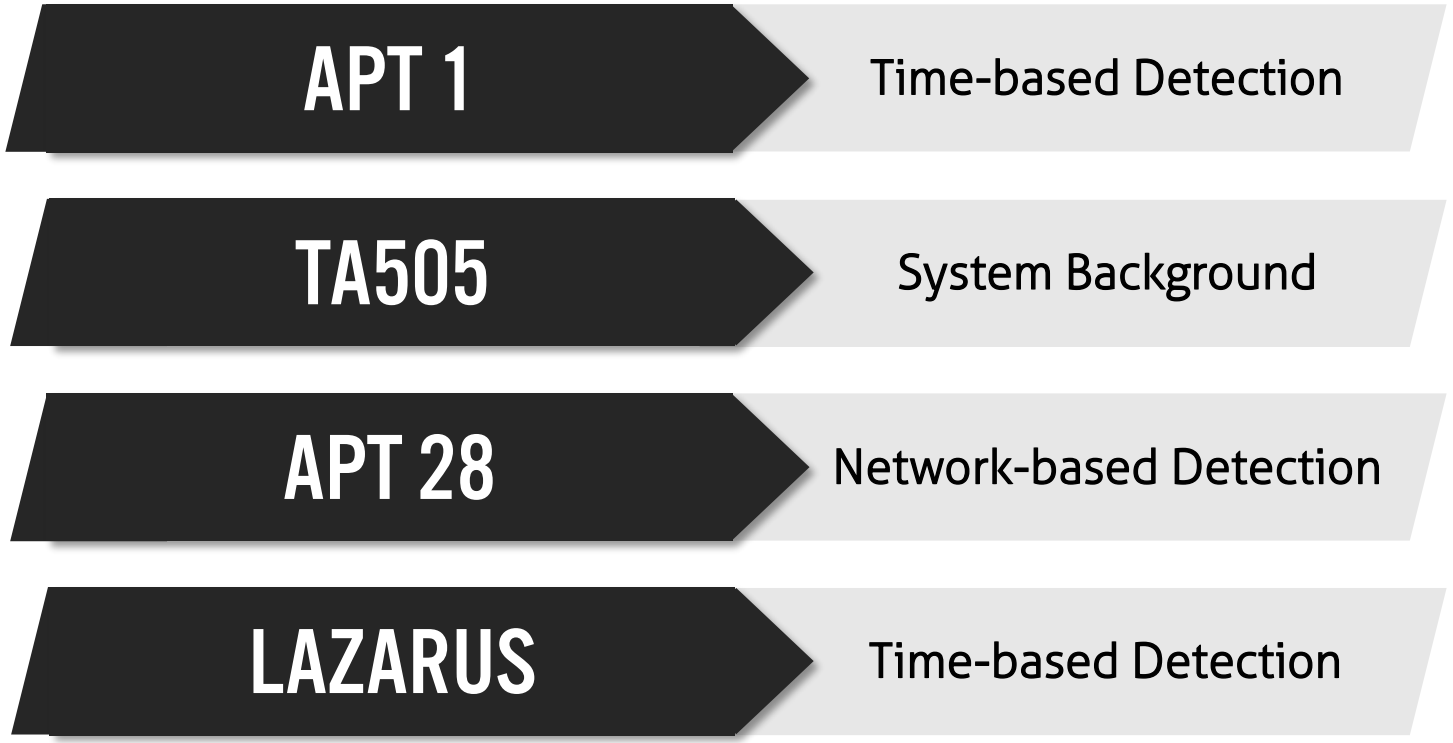


# RESULTS – APT TRACKING




According to Vendor's Group Attribution

# RESULTS – APT TRACKING




According to Vendor's Group Attribution

# RESULTS & INSIGHT

Mitre ATT&CK™ Framework Updated (2019)

T1497 – VIRTUALIZATION/SANDBOX EVASION

Improvement on APT Insight

APT OVERLAPPING

EVASION TECHNIQUES KNOWLEDGE





# CLOSING REMARKS

- Evasion techniques are constantly evolving.
- Use different profiles for your Sandbox and avoid generic ones.
- Keep all your systems up-to-date to avoid Malware exploiting known vulnerabilities.





# QUESTIONS





ENIGMA