uberAgent ESA

Helge Klein, vast limits

Security visibility through Windows endpoint analytics



Why uberAgent – in one word

uberAgent ESA

visibility



Why uberAgent – in one slide

- Everybody monitors servers
  - But what about the end user‘s device?
- Organizations realize **they need visibility** for:
  - Performance, app usage, security, compliance, ...
- uberAgent covers it all
  - **Quality** metrics
  - **Easy** to deploy
  - Proven **scalability** to 100,000s of endpoints



uberAgent UXM

User Experience Monitoring



UXM – applications

- Automatic **app identification**
  - Process (iexplore.exe) -> app (Internet Explorer)
  - Works with all apps out of the box
- Application **startup**
  - Process creation
  - Startup duration

UXM – applications

- Application **performance**
  - Resource utilization for entire apps or individual processes
  - CPU, RAM, disk IO, network, GPU, ...
- Application **errors**
  - Crashes & hangs
  - UI unresponsiveness



UXM – applications

- Application **inventory**
  - What sits on disk?
- Application **usage**
  - What is running?
- **Foreground** application
  - What is the user interacting with?

UXM – applications

- Any kind of **native apps**
  - Win32, UWP, Java, App-V, ...
- **Web apps**, too!
  - All major browsers
  - **No changes** to website code required
- The browser has become an **OS for web apps**
  - uberAgent shows you what's going on inside

UXM – users

- **Logon**/logoff activity
- **Session** start/end
  - uberAgent generates a unique ID per session
- User **account** of process/app events
  - Optional anonymization
- User **metadata** can be read from:
  - AD, registry, environment variables



UXM – machines

- Rich **inventory** info
  - AD, HW, OS, Citrix, VMware, ...
- On/off transitions
  - **Startup**/shutdown/suspend/resume
- Machine **metadata** can be read from:
  - AD, registry, environment variables

UXM – networking

- All **network connections**
  - OS level
  - In the browser
- Network **activity per application** & user
  - Data volume, latency, count, ...
  - Successful & failed connections
- **WiFi SSID**, network type & IP address

UXM – summary

- Sounds like a **cool security tool**, right?
  - But wait – it's getting a lot better!
- All of this is part of our **existing product**
  - UXM = user experience monitoring
- Now we are really getting serious with security
  - **ESA = endpoint security analytics**

uberAgent ESA

Endpoint Security Analytics

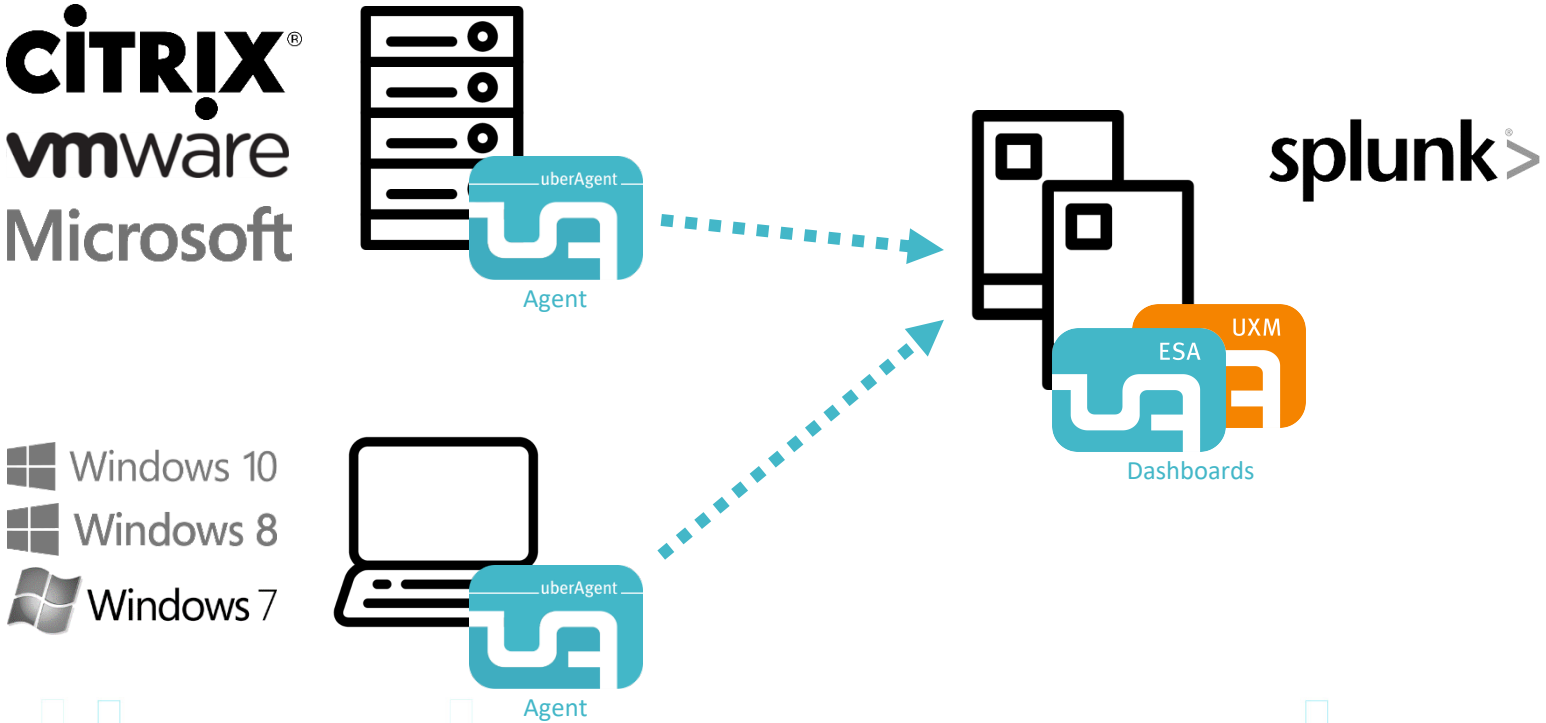


ESA in a nutshell

- UXM provides rich **context & metadata**
- ESA adds deep **security visibility**
- **One agent** for UX, performance & security
  - Small footprint, proven reliability
- Optimized for **physical & virtual**
- Windows client & server
  - Soon **macOS**, too



ESA – architecture



uberAgent ESA

Features



ESA – process tagging

- Goal: **identification of risky processes**
- Matching processes get
  - **Tag** (any string)
  - **Risk score** (any number)
- Dashboard visualizes findings

ESA – process tagging

- Powerful **rule definition language**
- Regular expressions everywhere
- Built-in extension: PATH_REGEX
  - Combination of **environment variables & regex**
  - Env var is evaluated first, resulting regex second
  - Example: `^%ProgramFiles%\\Windows Defender\\.+\.exe$`



ESA – process tagging

- **Reusable rule blocks**
- E.g. define how to detect MS Office parent processes:

```
[ConfigBlockDefine name=ParentIsMsOffice]
Parent.Name = ^excel\.exe$
Parent.Name = ^msaccess\.exe$
Parent.Name = ^onenote\.exe$
Parent.Name = ^outlook\.exe$
Parent.Name = ^powerpnt\.exe$
Parent.Name = ^winword\.exe$
Parent.Company = ^Microsoft.*
```

ESA – process tagging

- Insert blocks in rules:

```
[ProcessTaggingRule]
RuleName = Detect script child processes of MS Office apps
EventType = Process.Start
@ConfigBlockInsert ParentIsMsOffice
Process.Name = ^cmd\.exe$
Process.Name = ^powershell\.exe$
Process.Name = ^cscript\.exe$
Process.Name = ^wscript\.exe$
Process.Name = ^ftp\.exe$
Tag = proc-start-msoffice-child
RiskScore = 100
```

ESA – process tagging

- **Detection elements**
  - Process & parent properties
  - Name, user, path, command line
  - **Application name, version**
  - Company, elevation status
  - Session ID
  - Directory permissions



ESA – process tagging

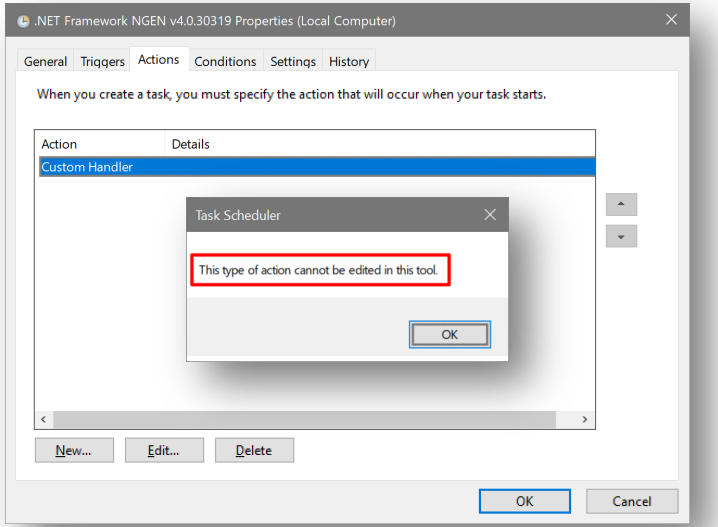
- **Directory permission** detection elements
  - Process.DirectoryUserWriteable
    - Checks if the process' **directory is writeable by the user**
  - Process.DirectorySdSddl
    - Security descriptor in SDDL format
    - **SIDs replaced with names**
      - S-1-5-21-3803133166-2955000686-238773884-1029 -> Corp\User23
    - Permissions converted from **hex access masks to strings**
      - 0x1200a9 -> read_execute

ESA – process tagging

- Predefined rules
  - Process starts from directories with a **low mandatory integrity** label
  - Process starts from directories that are **user-writeable**
  - **Script child processes** of Microsoft Office applications
  - Child processes of the **WMI service**
  - Child processes of **Adobe Reader**
  - **LOLBAS** (various)
  - …

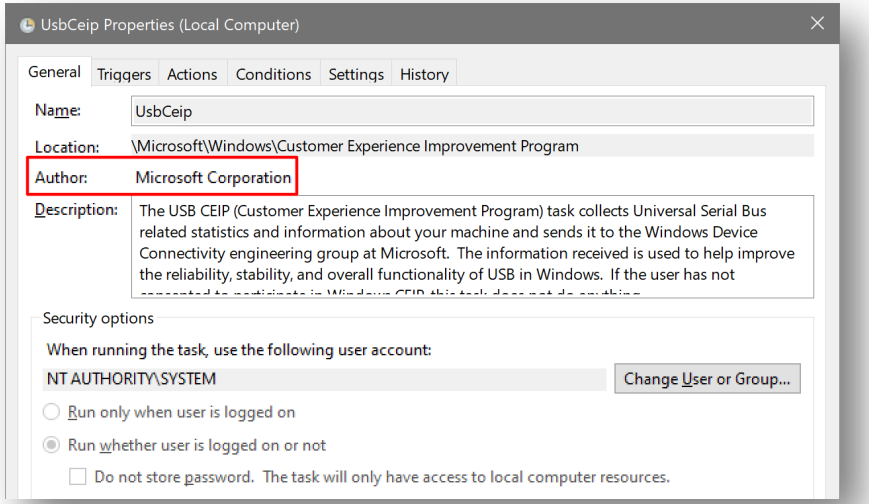
ESA – scheduled tasks

- Scheduled tasks are **fantastic for hiding malware**
- Important properties are **missing from the UI**
  - COM actions, custom triggers
- Huge number of tasks on any system
  - Completely **undocumented**



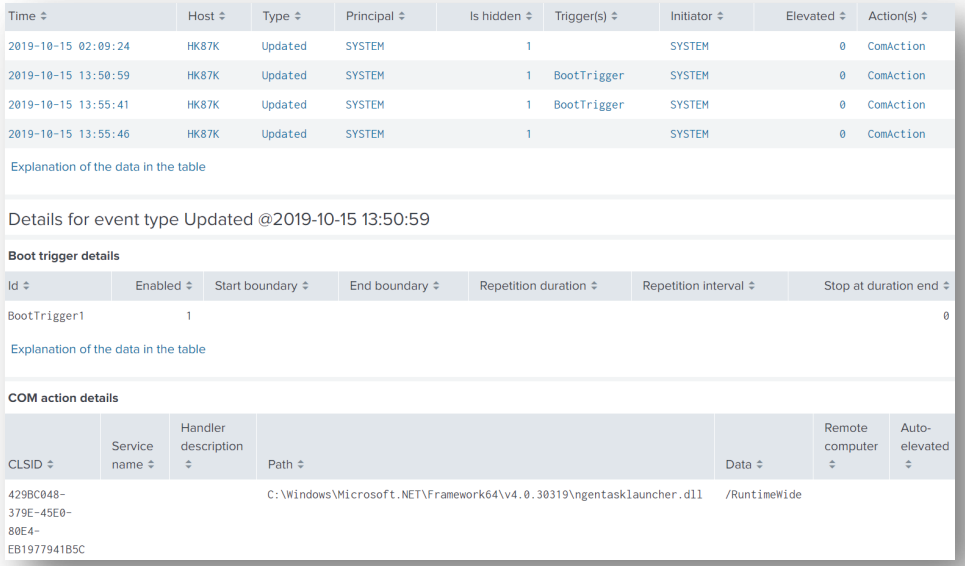
ESA – scheduled tasks

- **No authentication** mechanism for “good” tasks
  - Author can be set to **any value**



ESA – scheduled tasks

- uberAgent detects **new or changed** tasks
- Details on all types of
  - **Actions**
    COM, exec, email, message
  - **Triggers**
    Event, time, idle, boot, logon, custom, …



Time	Host	Type	Principal	Is hidden	Trigger(s)	Initiator	Elevated	Action(s)
2019-10-15 02:09:24	HK87K	Updated	SYSTEM	1		SYSTEM	0	ComAction
2019-10-15 13:50:59	HK87K	Updated	SYSTEM	1	BootTrigger	SYSTEM	0	ComAction
2019-10-15 13:55:41	HK87K	Updated	SYSTEM	1	BootTrigger	SYSTEM	0	ComAction
2019-10-15 13:55:46	HK87K	Updated	SYSTEM	1		SYSTEM	0	ComAction

Explanation of the data in the table

Details for event type Updated @2019-10-15 13:50:59

Boot trigger details

Id	Enabled	Start boundary	End boundary	Repetition duration	Repetition interval	Stop at duration end
BootTrigger1	1					0

Explanation of the data in the table

COM action details

CLSID	Service name	Handler description	Path	Data	Remote computer	Auto-elevated
429BC048-379E-45E0-80E4-EB1977941B5C			C:\Windows\Microsoft.NET\Framework64\v4.0.30319\ngentasklauncher.dll	/RuntimeWide		

ESA – process tree dashboard

- Security tools record **every new process**

  👍🎉❓
- That easily amounts to **1,000s of events per minute**

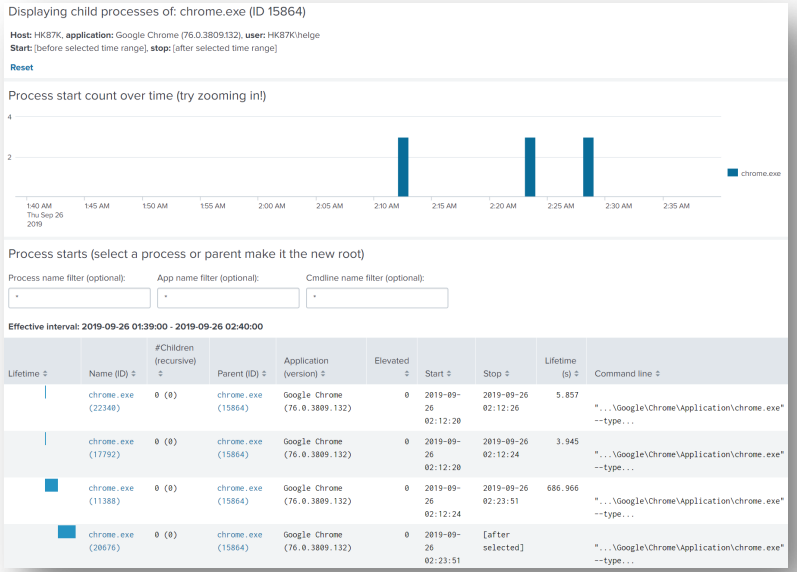
  ❓😨🤯
- How to find the **needle in the haystack?**

  🤔🔍 ❓



ESA – process tree dashboard

- Interactive **navigation through process hierarchies**
  - Filtering by host(s) or any metadata
  - **Process starts over time**
  - Process lifetime
  - Full **command line**
  - Application name & version
  - **Elevation status**



ESA – roadmap

- **Beta** version in Q1/2020
- Planned features
  - New **system services**
  - New local (admin) **users**
  - New TLS root **certificates**
  - New **autoruns**
- Much more in version 1.x

uberAgent ESA

See us at booth #108

Questions?



More information

https://**uberagent.com**

info@uberagent.com

vast LIMITS

We build enterprise-grade software.

Our founder architected what is now Citrix Profile Management.

Our tools Delprof2 and SetACL have been downloaded more than half a million times.