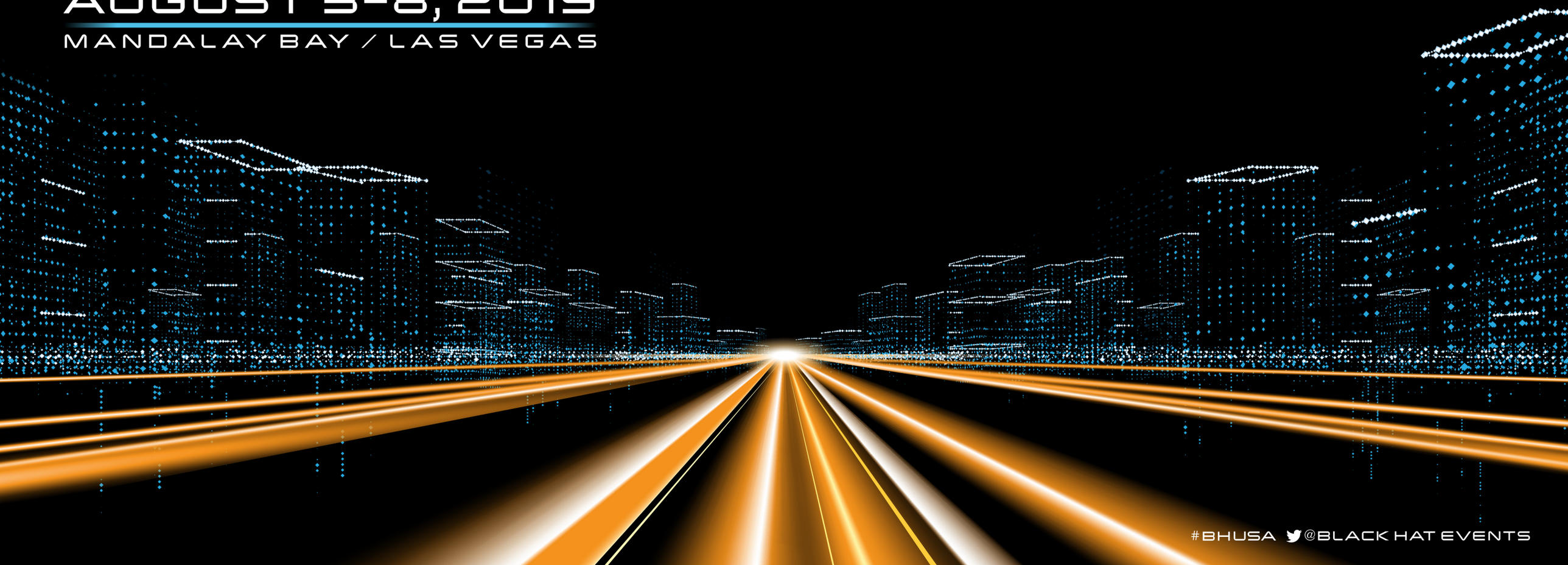
black hat®
USA 2019

AUGUST 3–8, 2019

MANDALAY BAY / LAS VEGAS



#BHUSA @BLACK HAT EVENTS

SysmonX: An Augmented and Community-Driven Drop-In Replacement of Sysmon

Project Goals:

- More visibility of threat activity. Extend the Sysmon data collection sources and create new security events
- Additional extensible threat detection. Extend the Sysmon ability to correlate events. Effectively enabling new logical operations between events and the creation of advanced detection capabilities
- Improved signal to noise. Enable the false positive reduction by narrowing down suspicious events through dedicated scanners
- More resilient. React to known subversion and evasion techniques that impact Sysmon
- Community driven. Lever open source and community to improve security outcomes

Why SysMonX?

SysMon is widely used for monitoring and threat visibility, with numerous solutions designed around its events

SysMon has gaps in visibility and functionality

- Ability to easily add new event visibility (ETW, etc)
- Ability to add metadata to and filter on events
- Ability to correlate multiple events together
- Poor signal-to-noise in captured events
- Weak against multiple subversion techniques

Rapidly responding to and resolving these challenges in a dynamic threat landscape is ready made for an open-source community driven approach

New Detection

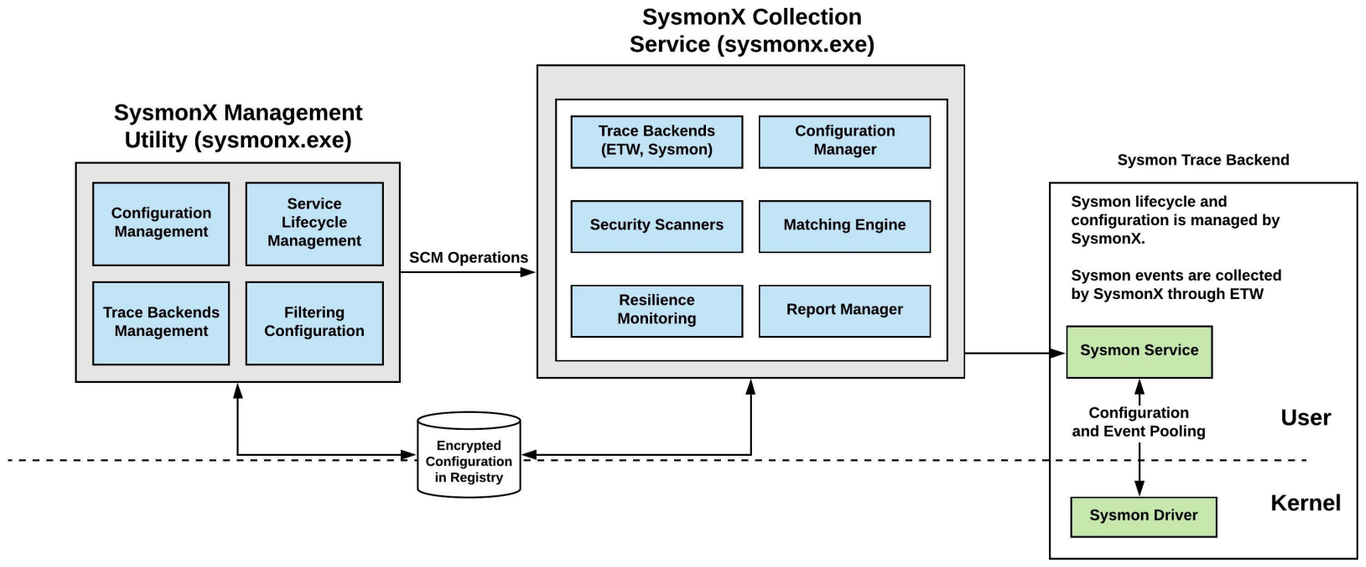
- Cmdline and Parent Process Spoofing detection
- Ability to detect userspace injection techniques (eventing + memory inspection through built in scanner modules)

Extensible

- Ability to perform regex over security event fields
- Open Source

More visibility

- WMI calls over all the namespaces, not just root:subscription
- Ability to collect authentication information
- Ability to collect PowerShell events



Demo: SysmonX Capabilities

New Process Attributes

Will show visibility of process injection and attribute against injected process with a new filtering condition

New Regex capability

Demonstrate alerting on a suspicious command-line with PowerShell

New Event Types

Demonstrate new events from PowerShell logging with an unmanaged PowerShell that executes a malicious script and bypasses AMSI

```
<SysmonX schemaversion="4.1">
    <HashAlgorithms>*</HashAlgorithms>
    <CheckRevocation/>
    <EventFiltering>

     <ProcessCreate onmatch="include" scanners="hollow_modules">
            <CommandLine condition="regex_match" name="id=T1055”>.*-
            EXECUTIONPOLICY.*BYPASS.*IEX.*</CommandLine>
       </ProcessCreate>

       <PowershellEvent onmatch="include">
            <ScriptBlock condition="regex_match”
            name=”Unmanaged PS with AMSI
bypass">.*calc.exe.*</ScriptBlock>
       </PowershellEvent>

       <FileCreateTime onmatch="exclude"/>
       <NetworkConnect onmatch="exclude"/>
       <ProcessTerminate onmatch="exclude"/>
       <DriverLoad onmatch="exclude"/>
       <ImageLoad onmatch="exclude"/>
       <CreateRemoteThread onmatch="exclude"/>
       <RawAccessRead onmatch="exclude"/>
       <ProcessAccess onmatch="exclude"/>
       <FileCreate onmatch="exclude"/>
       <RegistryEvent onmatch="exclude"/>
       <FileCreateStreamHash onmatch="exclude"/>
       <PipeEvent onmatch="exclude"/>
       <WmiEvent onmatch="exclude"/>
  </EventFiltering>
</SysmonX>
```

SysmonX Github Repository

https://github.com/marcosd4h/sysmonx

Utilized open source solutions:

- Blackbone memory access library
  https://github.com/DarthTon/Blackbone
- KrabsETW library
  https://github.com/Microsoft/krabsetw
- Demos include modified source from
  https://www.fuzzysecurity.com
  https://github.com/leechristensen/UnmanagedPowerShell

https://www.blackhat.com/us-19/arsenal/schedule/index.html#sysmonx-an-augmented-and-community-driven-drop-in-replacement-of-sysmon-17021