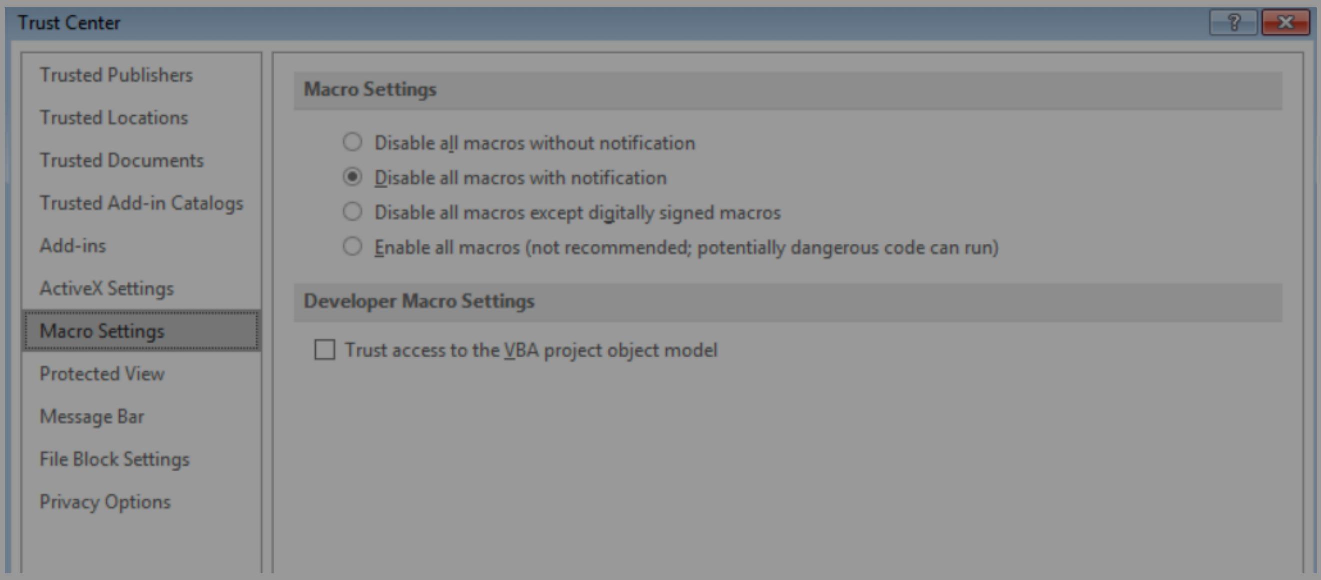
Hunting for evil: detect macros being executed

Pieter Ceelen | January 16, 2018

In many of our red teaming and incident response engagements, we encounter the abuse of MS Office macros as a vector to drop a remote access trojan and thereby gain initial foothold. From many discussions with our clients we have learned that macros are hard to secure and often a necessity for business operations. In this blog I'll share a rough approach for detection of evil macros. In parallel, we are working on a more robust solution that can fully manage the macro problem (we hope to share some news on this later).

Once discussing the subject of macro security, clients indicate that there is no insight into what office macros are being used/executed with a legit business purpose. In the majority of cases businesses leave Macro settings untouched, rendering the effective settings "macro disabled with notification".



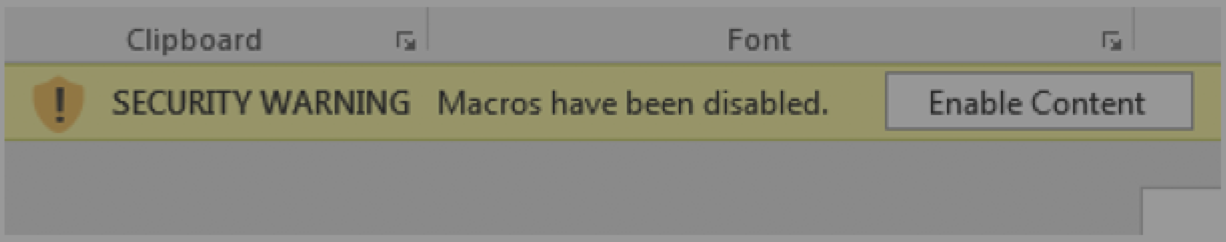
Macro Settings

- Disable all macros without notification
- [✓ Disable all macros with notification]
- Disable all macros except digitally signed macros
- Enable all macros (not recommended; potentially dangerous code can run)

Developer Macro Settings

- ☐ Trust access to the VBA project object model

If macro security is configured like this, it only takes one user to click on the yellow warning button and voila, a new ransomware infection or remote access trojan has entered your network…



SECURITY WARNING Macros have been disabled.

Macro signing is hard!

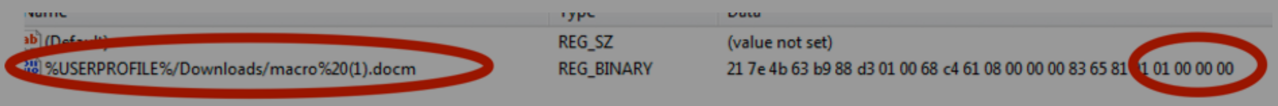
When diving deeper into this, it turns out that it is quite a challenge to deploy macro signing. Amongst others the following issues exist:

- Lack of central insight in macros that exist and are executed (although the office telemetry agent should be able to collect insights on VBA use)
- No warnings are provided when macros are disabled when signing is enforced, there are no visual clues for an end user that functionality is being suppressed.
- Complete lack of key management and insights into documents and keys used. We have encountered production macro documents signed with an expired/revoked key. Should we really trust all macros signed with a revoked key?

Let's take this subject step by step. First, create insight into macros executed in a corporate environment so that the information can be used for hunting or alerting. For that we will use the 'trustRecord' registry keys used by MS Office.

About trustrecords

Once macros in an Office document are enabled by the end-user, Office 'caches' this approval and the next time the exact same document is opened the user is no longer asked for approval as the document is auto-approved. The cache of approved files is maintained in a trustrecord, for more info see this blogpost.



```
(Default)                                  REG_SZ        (value not set)
%USERPROFILE%/Downloads/macro%20(1).docm   REG_BINARY    21 7e 4b 63 b9 88 d3 01 00 68 c4 61 08 00 00 00 83 65 81 01 01 00 00 00
```

Note that this only applies when the macro settings is configured to be "disable macros and show notification" (default setting).

As can be seen in the screenshot the full filepath is used as the registry value name and a binary blob is used for maintaining the exact trust. There are 2 events that add new items here:

- Mark a protected document for editing, indicated with the 01 00 00 00 suffix
- Enable macros indicated with FF FF FF 7F suffix

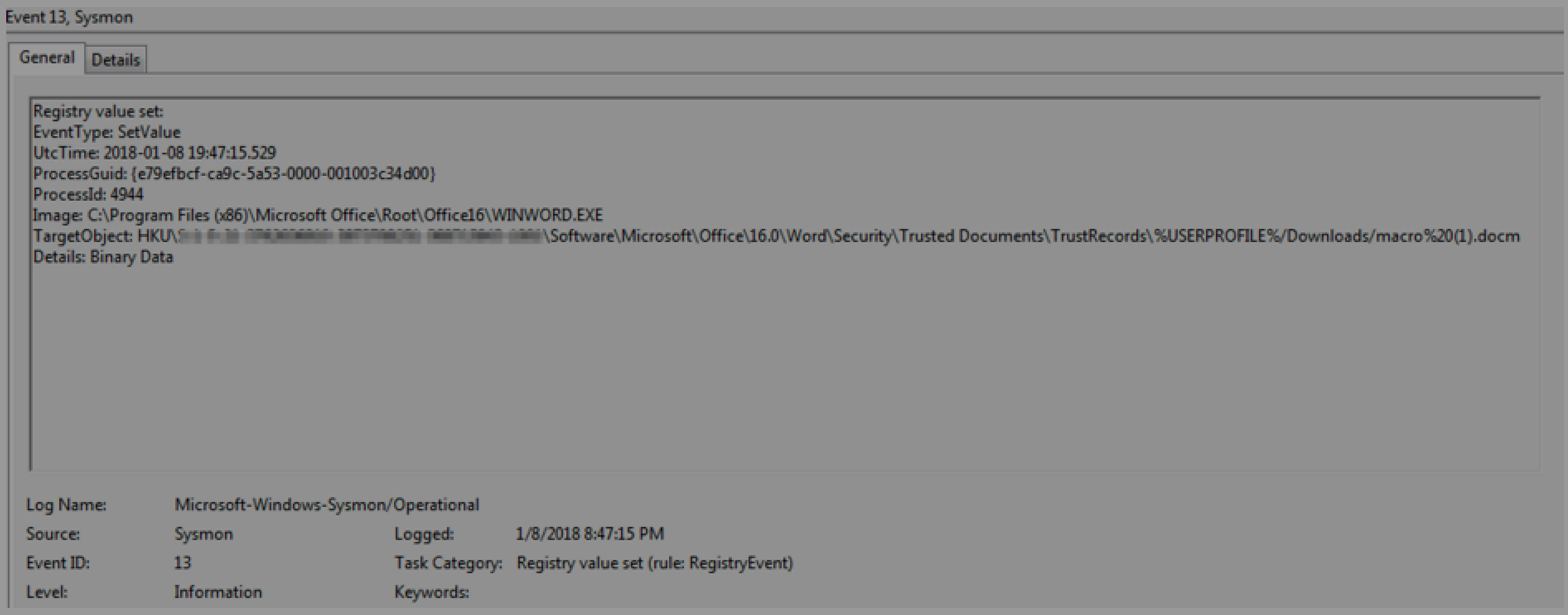
Sysmon to the rescue?

So, lets monitor for registry changes on this registry hive, for this we will use Sysmon (and assume Windows event collection is in place to aggregate these logs from all workstations).

The following basic Sysmon config creates an event at every 'protect document editing' or 'enable macros'

```
<Sysmon schemaversion="4.00">
  <HashAlgorithms>md5,sha256</HashAlgorithms>
  <EventFiltering>
    <RegistryEvent onmatch="include">
      <TargetObject condition="contains>
        Security\Trusted Documents\TrustRecords
      </TargetObject>
    </RegistryEvent>
  </EventFiltering>
</Sysmon>
```

Example event generated



Event 13, Sysmon

```
Registry value set:
EventType: SetValue
UtcTime: 2018-01-08 19:47:15.529
ProcessGuid: {e79efbcf-ca9c-5a53-0000-001003c34d00}
ProcessId: 4944
Image: C:\Program Files (x86)\Microsoft Office\Root\Office16\WINWORD.EXE
TargetObject: HKU\[redacted]\Software\Microsoft\Office\16.0\Word\Security\Trusted Documents\TrustRecords\%USERPROFILE%/Downloads/macro%20(1).docm
Details: Binary Data

Log Name:     Microsoft-Windows-Sysmon/Operational
Source:       Sysmon                 Logged:          1/8/2018 8:47:15 PM
Event ID:     13                     Task Category:   Registry value set (rule: RegistryEvent)
Level:        Information            Keywords:
```

In my limited testing (Win7, Office 2016) I identified the following 3 moments in time when the event fires:

- When end-user marks a protected document for editing
- When end-user enables macros
- When end-user opens a document that already has a trust record

Hunting and alerting

Unfortunately, there are some limitations in Sysmon that prevent us from distinguishing between the ‘allow for editing’ and the ‘enable macros’ actions. In Sysmon configuration we cannot provide a bitmask or filter for the registry data being written and in the resulting event we only see a string ‘Binary data’, not the data itself. As such we cannot alert on the specific event ‘macro enabled’ only.

However, this data can be a significant improvement compared to having no visibility at all. There are various use cases to be thought off:

- Alert when a filename occurs twice in the logs within 10 minutes where the specific filename/path was not seen before. Two events in short time likely indicate a ‘open protected document for editing’ and ‘enable macro’ event consecutively.
- Use the logs on regular executed macros to start projects on macro signing or automated whitelisting in your SIEM.
- In case of malware infections detected via another way (e.g. AV alert, sysmon persistency alert) quickly triage and identify the originating malicious macro infection vector file.

Happy hunting!