Potential MuddyWater Campaign uses PRB-Backdoor

blog.trendmicro.com/trendlabs-security-intelligence/another-potential-muddywater-campaign-uses-powershell-based-prb-backdoor/

June 14, 2018

The MuddyWater campaign was first sighted in 2017 when it targeted the Saudi government using an attack involving PowerShell scripts deployed via Microsoft Office Word macro. In March 2018, we provided a detailed analysis of another campaign that bore the hallmarks of MuddyWater.

In May 2018, we found a new sample (Detected as W2KM_DLOADR.UHAOEEN) that may be related to this campaign. Like the previous campaigns, these samples again involve a Microsoft Word document embedded with a malicious macro that is capable of executing PowerShell (PS) scripts leading to a backdoor payload. One notable difference in the analyzed samples is that they do not directly download the Visual Basic Script(VBS) and PowerShell component files, and instead encode all the scripts on the document itself. The scripts will then be decoded and dropped to execute the payload without needing to download the component files.

As mentioned earlier, our analysis of the sample revealed characteristics that likely connect it to the MuddyWater campaign, in particular:

- The delivery method, which involves the use of a malicious document with an embedded macro as a lure for potential victims

- The obfuscation method for the macro scripts, which will result in an intended backdoor payload. This method is commonly used in samples that were used in the MuddyWater campaign

Infection chain

Figure 1. Comparison of the infection chains used in the previous and current campaigns

Technical details

The sample we analyzed was a Word document used as a lure for unsuspecting victims. However, unlike the samples from the previous campaigns, the lure document deals with a different subject matter. Instead of using government or telecommunications-related documents, the new lure document presents itself as a reward or promotion, which could indicate that the targets are no longer limited to specific industries or organizations.

Figure 2. Sample lure document used in the new campaign

The document is designed to trick users into enabling the macro to view its full content. However, the macro's true purpose is to allow it to execute malicious routines without the user’s knowledge.

Once the macro is enabled, it will use the **Document_Open()** event to automatically execute the malicious routine if either a new document using the same template is opened or when the template itself is opened as a *document0*.

Figure 3. Executing the malicious routine via Document_Open()

Figure 3. Executing the malicious routine via Document_Open()

The malicious macro's code snippet uses three main functions, specifically:

- The function contained in the RED box is the Document_Open() event, where all the sub-functions will be executed/called.
- The code inside the GREEN box manipulates the images shown in the document's body.
- The code inside the BLUE box constructs the main Powershell commands and scripts. These will be executed to perform the main routine.

Figure 4. A snippet of the malicious macro’s code, marked with colored boxes to show the different functions

Decoding and deobfuscation

Analysis of the code revealed a PowerShell script capable of decoding the contents of the malicious document, which results in the execution of yet another encoded PowerShell script.

Figure 5. The Powershell script contained in the sample's code

Figure 5. The Powershell script contained in the sample's code

Figure 6. The second encoded PowerShell script, which is executed after the first script is decoded

Figure 6. The second encoded PowerShell script, which is executed after the first script is decoded

This will then result in more readable PowerShell scripts capable of dropping various components in the %Application Data%\Microsoft\CLR\* directory. The main PowerShell file *invoker.ps1* uses these components to run the final payload, PRB-Backdoor, previously analyzed by other security researchers in May 2018.

Figure 7: The components dropped in the %Application Data%\Microsoft\CLR\* directory

PRB-Backdoor is a backdoor that takes its name from the function used in the final PowerShell script payload, as seen in the figure below.

Figure 8. The PS function from which PRB-Backdoor takes its name

Figure 8. The PS function from which PRB-Backdoor takes its name

The backdoor communicates with its Command-and-Control (C&C server), hxxp://outl00k[.]net, to send and receive the following commands:

Command	Details
PRB-CREATEALIVE	Initializes connection with the C&C Server
PRB-CREATEINTRODUCE	Registers/introduces the affected machine to the C&C server
PRB-History	Gather browsing histories from different browsers and send it to the C&C server using the "sendfile" function
PRB-PASSWORD	Steals passwords listed or found in the browser histories
PRB-READFILE	Reads files

PRB-WRITEFILE	Writes files
PRB-Shell	Executes shell commands
PRB-Logger	Calls the "Logger" function, used to record keyboard strokes
PRB-Shot	Triggers the SNAP function, used to capture screenshots
PRB-funcupdate	Updates functions
sysinfo	Gathers system information
Start_Dns	Initializes DNS Session/Connection

If these samples are indeed related to MuddyWater, this means that the threat actors behind MuddyWater are continuously evolving their tools and techniques to make them more effective and persistent.

Countermeasures and Trend Micro Solutions

Given the use of lure documents designed with social engineering in mind, it is likely that the attackers use phishing or spam to target users who are unaware of these documents' malicious nature. Awareness can effectively mitigate or stop these kinds of attacks from being successful. The first step is to be able to identify phishing attacks and distinguish legitimate emails from malicious ones. Telltale signs of social engineering include “too-good-to-be-true” offers and messages that lack context. In general, users should always practice caution when it comes to email. This includes avoiding clicking on links or downloading any documents unless certain that these are legitimate.

Trend Micro™ Deep Discovery™ provides detection, in-depth analysis, and proactive response to today’s stealthy malware, and targeted attacks in real time. It provides a comprehensive defense tailored to protect organizations against targeted attacks and advanced threats through specialized engines, custom sandboxing, and seamless correlation across the entire attack lifecycle, allowing it to detect threats even without any engine or pattern update.

Trend Micro™ Hosted Email Security is a no-maintenance cloud solution that delivers continuously updated protection to stop spam, malware, spear phishing, ransomware, and advanced targeted attacks before they reach the network.

Trend Micro™ Deep Discovery™ Email Inspector and InterScan™ Web Security prevent malware from ever reaching end users. At the endpoint level, Trend Micro™ Smart Protection Suites deliver several capabilities that minimize the impact of these attacks.

These solutions are powered by the Trend Micro XGen™ security, which provides a cross-generational blend of threat defense techniques against a full range of threats for data centers, cloud environments, networks, and endpoints. It features high-fidelity machine learning to secure the gateway and endpoint data and applications, and protects physical, virtual, and cloud workloads.

Indicators of Compromise (IoCs)

Detected as W2KM_DLOADR.UHAOEEN
- 240b7d2825183226af634d3801713b0e0f409eb3e1e48e1d36c96d2b03d8836b

APT & Targeted Attacks

We found a new sample that may be related to the MuddyWater campaign. The sample does not directly download the Visual Basic Script and PowerShell component files, and instead encode all the scripts on the document itself.

By: Michael Villanueva, Martin Co June 14, 2018 Read time: (words)

Content added to Folio